

Oracle® Database

Upgrade Guide

11g Release 1 (11.1)

B28300-01

July 2007

Oracle Database Upgrade Guide, 11g Release 1 (11.1)

B28300-01

Copyright © 2002, 2007, Oracle. All rights reserved.

Primary Authors: Viv Schupmann, Craig B. Foch

Contributor: Valarie Moore (graphic artist)

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

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software—Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Preface	xi
Audience	xi
Documentation Accessibility	xii
Related Documentation	xii
Conventions	xiii
 What's New in Oracle Database Upgrade Guide	xv
 1 Introduction to the Database Upgrade Process	
Overview of the Database Upgrade Process	1-1
Oracle Release Numbers	1-4
Running Multiple Oracle Releases	1-5
Using Optimal Flexible Architecture (OFA)	1-6
Converting Databases to 64-bit Oracle Database Software	1-6
Rolling Upgrades	1-7
Moving From the Standard Edition to the Enterprise Edition	1-8
Upgrading from Oracle Database Express Edition to Oracle Database	1-9
 2 Preparing to Upgrade	
Prepare to Upgrade	2-1
Become Familiar with New Oracle Database Features	2-1
Determine the Upgrade Path	2-2
Choose an Upgrade Method	2-3
Choose an Oracle Home Directory	2-5
Develop a Testing Plan	2-5
Prepare a Backup Strategy	2-8
Test the Upgrade Process	2-8
Test the Upgraded Test Database	2-9
 3 Upgrading to the New Release	
System Considerations and Requirements	3-1
Upgrading Oracle Clusterware and Automatic Storage Management (ASM) Databases	3-2
Upgrading with Read-Only and Offline Tablespace	3-3
Upgrading Standby Databases	3-4

Upgrading Your Operating System.....	3-4
Migrating Data to a Different Operating System	3-4
Upgrading Databases That Use Oracle Streams Downstream Capture	3-5
Install the New Oracle Database Software	3-5
Install the Latest Available Patchset Release and Any Required Patches	3-7
Run the Pre-Upgrade Information Tool.....	3-7
Prepare the Database to Be Upgraded.....	3-11
Deprecated CONNECT Role	3-11
Database Links with Passwords	3-11
TIMESTAMP WITH TIME ZONE Data Type.....	3-12
Optimizer Statistics.....	3-12
Disable Oracle Database Vault.....	3-13
Save Oracle Enterprise Manager Database Control Data	3-13
Run the Oracle Net Configuration Assistant	3-15
Upgrade a Database Using Database Upgrade Assistant.....	3-16
Using the DBUA Graphical User Interface.....	3-17
Using DBUA in Silent Mode.....	3-33
Upgrade a Database Manually	3-37
Back Up the Database	3-37
Prepare the New Oracle Home	3-38
Upgrade the Database	3-39
Troubleshoot the Upgrade.....	3-45
Cancel the Upgrade	3-47
Upgrade an ASM Instance with DBUA	3-48
Upgrade an ASM Instance Manually	3-53
Reconfigure Oracle Cluster Synchronization Services	3-53
Prepare the New ASM Oracle Home	3-54
Upgrade ASM.....	3-55
Troubleshoot the ASM Upgrade.....	3-57
Cancel the ASM Upgrade	3-57

4 After Upgrading to the New Release

Required Tasks After Database Upgrades	4-1
Update Environment Variables (Linux and UNIX Systems Only).....	4-1
Upgrade the Recovery Catalog	4-2
Upgrade Statistics Tables Created by the DBMS_STATS Package.....	4-2
Upgrade Externally Authenticated SSL Users.....	4-2
Install Oracle Text Supplied Knowledge Bases	4-2
Update Your Oracle Application Express Configuration	4-3
Configure Fine-Grained Access to External Network Services.....	4-3
Enable Oracle Database Vault.....	4-4
Deploy Database Vault Administrator (DVA).....	4-4
Recommended Tasks After Database Upgrades	4-6
Recommended Tasks After All Database Upgrades.....	4-6
Recommended Tasks After Upgrading a 10g Release 1 (10.1) Database	4-9
Recommended Tasks After Upgrading an Oracle Express Edition Database	4-11
Tasks to Complete Only After Manual Database Upgrades.....	4-11

Change Passwords for Oracle-Supplied Accounts	4-11
Migrate Your Initialization Parameter File to a Server Parameter File	4-11
Upgrade Oracle Text.....	4-12
Upgrade the Oracle Cluster Registry (OCR) Configuration.....	4-13
Adjust the Initialization Parameter File for the New Release	4-13
Configure Enterprise Manager.....	4-13
Set CLUSTER_DATABASE Initialization Parameter.....	4-14
Required Tasks After ASM Upgrades	4-14
Recommended Tasks After ASM Upgrades	4-14
Reset ASM Passwords to Enforce Case-Sensitivity.....	4-14
Advance the ASM and Oracle Database Disk Group Compatibility	4-15
Set Up ASM Preferred Read Failure Groups	4-15
Optional Tasks After ASM Upgrade	4-16
Single-Instance ASM Upgrade	4-16
Cluster ASM Upgrade	4-16
Database Upgrade After ASM Upgrade.....	4-17
Tasks to Complete Only After Manual ASM Upgrades	4-18
Upgrade the Oracle Cluster Registry (OCR) Configuration for ASM.....	4-18
Adjust the ASM Initialization Parameter File for the New Release	4-19
Install and Configure Enterprise Manager Database Control for ASM.....	4-19

5 Compatibility and Interoperability

What Is Compatibility?	5-1
The COMPATIBLE Initialization Parameter.....	5-1
Setting the COMPATIBLE Initialization Parameter	5-3
What Is Interoperability?	5-4
Deprecated Features in 11g Release 1 (11.1)	5-4
Compatibility and Interoperability Issues Introduced in 11g Release 1 (11.1)	5-4
Automatic Maintenance Tasks Management.....	5-5
New SYSASM Privilege and OSASM Group for ASM Administration	5-6
ASM Disk Group Compatibility	5-6
COMPUTE STATISTICS and ESTIMATE STATISTICS Clauses	5-6
Oracle Data Mining Models and the DMSYS Schema Objects.....	5-6
Oracle Data Mining Scoring Engine	5-7
SQL Plan Management and Control of SQL Plan Baselines.....	5-7
Binary XML Support for Oracle XML Database.....	5-8
PL/SQL Native Compilation and Access Control for Network Utility Packages	5-8
PL/SQL Control Parameters	5-9
Change in WebDAV ACL Evaluation Rules in Oracle XML DB	5-10
Summary Management and SQL Access Advisor	5-10
SQL Access Advisor Tasks.....	5-10
Standard Edition Starter Database	5-10
Core Dump Location	5-11
LOG_ARCHIVE_DEST_# Parameter.....	5-11
SHARED_POOL_SIZE Parameter	5-11
JOB_QUEUE_PROCESSES Parameter	5-12
Compatibility and Interoperability Issues Introduced in 10g Release 2 (10.2)	5-12

SQL	5-12
CONNECT Role	5-12
Time Zone Files	5-13
New Limit for FAILED_LOGIN_ATTEMPTS	5-13
Compatibility and Interoperability Issues Introduced in 10g Release 1 (10.1)	5-13
SQL Optimizer	5-13
SQL	5-14
Invalid Synonyms After an Upgrade	5-14
Manageability	5-15
Transaction and Space	5-15
Recovery and Data Guard	5-15
RMAN	5-16
CREATE DATABASE	5-16
Oracle Real Application Clusters	5-16
Materialized Views	5-17
Change Data Capture	5-17
Change in the Default Archival Processing to Remote Archive Destinations	5-17
Limitations on NCHAR Data Types	5-18
PL/SQL Native Compilation	5-18
Evaluation of Numeric Literals	5-18
SHARED_SERVERS Parameter	5-19

6 Upgrading Your Applications

Overview of Upgrading Applications	6-1
Compatibility Issues for Applications	6-1
Upgrading Precompiler and OCI Applications	6-2
Understanding Software Upgrades and Your Client/Server Configuration	6-2
Compatibility Rules for Applications When Upgrading Client/Server Software	6-3
Upgrading Options for Your Precompiler and OCI Applications	6-5
Upgrading SQL*Plus Scripts and PL/SQL	6-7
Evaluation of Numeric Literals	6-7
Upgrading Oracle Forms or Oracle Developer Applications	6-7

7 Downgrading a Database

Supported Releases for Downgrading	7-1
Check for Incompatibilities	7-2
Perform a Full Backup	7-2
Downgrade the Database	7-3
Perform Post-Downgrade Tasks	7-7
Downgrading Oracle Clusterware Configuration	7-7
Restoring Oracle Enterprise Manager	7-8
Enabling Oracle Database Vault	7-12

8 Moving Data Using Data Pump and Export/Import

When to Use Data Pump Export/Import Versus Original Export/Import	8-1
Export and Import Requirements	8-2

Export and Import Requirements for Upgrades	8-2
Export and Import Requirements for Downgrades	8-3
Export/Import Usage on Data Incompatible with a Previous Release.....	8-4
Upgrade the Database Using Export/Import.....	8-4

A Initialization Parameter and Data Dictionary Changes

Initialization Parameter Changes.....	A-1
Deprecated Initialization Parameters.....	A-1
Obsolete Initialization Parameters.....	A-3
Compatibility Issues with Initialization Parameters	A-4
Compatibility Issues Introduced in Oracle Database 11g Release 1 (11.1)	A-4
Compatibility Issues Introduced in Oracle Database 10g	A-4
Static Data Dictionary View Changes	A-7
Deprecated Static Data Dictionary Views	A-7
Obsolete Static Data Dictionary Views	A-8
Static Data Dictionary Views with Dropped Columns	A-8
Dynamic Performance View Changes.....	A-8
Deprecated Dynamic Performance Views.....	A-9
Obsolete Dynamic Performance Views	A-9
Dynamic Performance Views with Renamed Columns.....	A-10
Dynamic Performance Views with Dropped Columns.....	A-10

B Gathering Optimizer Statistics

Collecting Statistics for System Component Schemas	B-1
Creating a Statistics Table.....	B-2

Index

List of Figures

1-1	Upgrade Steps	1-2
1-2	Example of an Oracle Release Number	1-4

List of Tables

2-1	Upgrade Paths	2-2
3-1	Statistics Collection for System Components and Schemas	3-12
3-2	DBUA Command Line options	3-34
3-3	Guidelines for Sizing the SYSAUX Tablespace	3-42
5-1	The COMPATIBLE Initialization Parameter	5-2
8-1	Exporting Data From Releases Earlier Than 11.1 and Importing Into Release 11.1.....	8-2
8-2	Exporting Data From Release 11.1 and Importing Into Earlier Releases.....	8-3
A-1	Dynamic Performance Views with Renamed Columns in Release 9.2.....	A-10

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 to Oracle Database 11g Release 1 (11.1), and important changes in the new 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, with good usability, to the disabled community. 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.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documentation

For more information, see these Oracle resources:

- *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 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

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.
<i>italic</i>	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.

What's New in Oracle Database Upgrade Guide

This preface describes new features in Oracle Database 11g Release 1 (11.1) and provides links to additional information.

■ Upgrade and Downgrade Enhancements

Oracle Database 11g Release 1 (11.1) continues to make improvements to simplify manual upgrades, upgrades performed using Database Upgrade Assistant (DBUA), and downgrades. The following features reduce the complexity of database upgrades and reduce the total amount of time it takes to perform an upgrade:

- **Improved pre-upgrade information**—Additional improvements have been made in the areas of space estimation, initialization parameters, statistics gathering, and easier recovery. See the Pre-Upgrade Information Tool in the steps described in "[Upgrade the Database](#)" on page 3-39.
- **Simplified error management**—Errors are now collected as they are generated during the upgrade and displayed by the Post-Upgrade Status Tool for each component. See the Post-Upgrade Status Tool in the steps described in "[Upgrade the Database](#)" on page 3-39.
- **Patchset upgrade/downgrades**—The `catupgrd.sql` script performs all upgrades and the `catdwgrd.sql` script performs all downgrades, for both patch releases and major releases. See "[Supported Releases for Downgrading](#)" on page 7-1.

■ General DBUA Enhancements

- DBUA will always configure Enterprise Manager Database Control in secure mode during upgrade (even if pre-upgrade is in nonsecure mode).
- The directory that you specify during the interview phase of Oracle Universal Installer (OUI) for `ORACLE_BASE` is stored in the Oracle home inventory. During the DBUA interview phase, `ORACLE_BASE` is displayed (and cannot be changed by the user) as follows:
 - * If the `ORACLE_BASE` environment variable is set, then `ORACLE_BASE` is displayed as the `ORACLE_BASE` environment variable.
 - * If `ORACLE_BASE` exists in the Oracle home inventory, then `ORACLE_BASE` is stored in the Oracle home inventory.
 - * Otherwise, the `ORACLE_BASE` is stored in `ORACLE_HOME`.

During the DBUA interview phase, by default, the `DIAGNOSTIC_DEST` initialization parameter (ADR directory) is displayed (can be changed by user) as `ORACLE_BASE/diag` and is used during subsequent directory creations.

■ **DBUA Enhancements for Single-Instance Databases**

DBUA provides the following enhancements for single-instance databases:

- Support for improvements to the pre-upgrade tool in the areas of space estimation, initialization parameters, statistics gathering, and new warnings.
- The directory that you specify during the DBUA interview phase for `ORACLE_BASE` is stored in the Oracle home inventory as the `DIAGNOSTIC_DEST` parameter and used during subsequent directory creations. (The default is the `ORACLE_BASE/diag/` directory.)

Note: The `DIAGNOSTIC_DEST` initialization parameter replaces the `USER_DUMP_DEST`, `BACKGROUND_DUMP_DEST`, and `CORE_DUMP_DEST` parameters

- Support for upgrading Oracle Database Express Edition (Oracle Database XE) to Oracle Database 11g Release 1 (11.1).
- Support for Automatic Storage Management (ASM)/Oracle File System (OFS) related features, such as moving data files to ASM or OFS or other storage devices such as (Storage Area Networks (SAN) and Network Area Storage (NAS) as part of upgrade.
- Option to make the connection to Enterprise Manager secure during database upgrade
- Automatic upgrade of Application Express

See Also: DBUA online Help system and "[Upgrade a Database Using Database Upgrade Assistant](#)" on page 3-16

■ **DBUA Enhancements for Oracle Real Application Clusters Configurations**

DBUA provides the following enhancements for Oracle Real Application Clusters (Oracle RAC) configurations:

- DBUA uses the value of `DIAGNOSTIC_DEST` parameter as the root directory where it automatically creates all of the relevant Automatic Diagnostic Repository (ADR) subdirectories and metadata files. The names of the subdirectories are based on the user-provided database name and instance name. DBUA constructs the directory structure in a fashion similar to the following:

ADR_BASE_directory/rdbms/db_unique_name/instance_name

- DBUA supports upgrades from Oracle9i Release 2 (9.2.0.4) and beyond, Oracle Database 10g Release 1 (10.1), and Oracle Database 10g Release 2 (10.2).
- Starting with Oracle Database 11g Release 1 (11.1), DBUA supports patchset upgrades of Oracle Database 11g Release 1 (11.1) software, as follows:
 - * Oracle Universal Installer (OUI) places the 11g Release 1 (11.1) Oracle RAC metadata in the Oracle RAC database home.
 - * OUI launches DBUA.

- * DBUA brings up all instances of the Oracle RAC database and collects all necessary information before the upgrade.
- * DBUA shuts down all instances and sets `CLUSTER_DATABASE=FALSE`.
- * DBUA brings up the local instance to run the SQL upgrade scripts.
- DBUA supports Oracle Database 11g Release 1 (11.1) improvements to the pre-upgrade tool in the areas of space estimation, initialization parameters, statistics gathering, and new warnings.
- DBUA creates restore scripts to restore the database, upon a user's request for a backup during the upgrade procedure. This restore script should incorporate necessary logic to update the `/etc/oratab` entries appropriately upon the downgrade.
- DBUA automatically restarts any services that were running prior to upgrade.

See Also: DBUA online Help system and the [Upgrade a Database Using Database Upgrade Assistant](#) on page 3-16

- **SQL Plan Management ensures that plans do not degrade when upgrading to a newer database release**

By maintaining a history of execution plans for a SQL statement, SQL Plan Management can detect a plan change. When SQL Plan Management detects a new plan, it marks the new plan for evaluation and uses the old (current good) plan. During the next maintenance window, SQL Plan Management validates the new plan. If it is found to perform better than the existing plan, then it is used in the future. If the performance has not improved, then the optimizer continues to use the old plan.

See Also: [SQL Plan Management and Control of SQL Plan Baselines](#) on page 5-7

- **Stronger password protection**

If you want to use the new Oracle Database 11g Release 1 (11.1) feature that enforces case-sensitive passwords, then you must reset the passwords of existing users during the database upgrade procedure. Strong password authentication will not be enabled for upgraded databases until each user password has been reset with the `ALTER USER` statement.

Note: If the default Oracle Database 11g Release 1 (11.1) 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: ["Reset Passwords to Enforce Case-Sensitivity"](#) on page 4-6

- **Binary XML support for Oracle XML DB**

When the database is upgraded to Oracle Database 11g Release 1 (11.1), none of the existing user XMLType tables and instances is modified. 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 11g Release 1 (11.1).

See Also: The section titled "[Binary XML Support for Oracle XML Database](#)" on page 5-8 for compatibility information

- **WebDAV ACL Support in Oracle XML DB**

There have been changes to the default behavior of WebDAV ACL (access control list) entries. Beginning with Oracle Database 11g Release 1 (11.1), ACE order is irrelevant. The default behavior is determined only by the first `<allow>` or `<deny>` entry that is encountered. When upgrading to Oracle Database 11g Release 1 (11.1), you might wish to reorder ACLs so that the `<deny>` entry occurs first if you want to retain the same behavior as previous releases.

See Also: "[Change in WebDAV ACL Evaluation Rules in Oracle XML DB](#)" on page 5-10

- **AUM Enabled by Default and Support for Upgrade from Rollback Segments**

With Oracle Database 11g Release 1 (11.1), automatic undo management is now enabled by default. (This is equivalent to setting the initialization parameter `UNDO_MANAGEMENT=AUTO`.) Also, a step-by-step procedure is available to help you upgrade your database from using rollback segments to using automatic undo management. As a part of the upgrade procedure, you can use a new function in that helps you to properly size the `UNDO` tablespace for each individual environment.

See Also: "[Migrate From Rollback Segments to Automatic Undo Mode](#)" on page 4-8

- **Automatic Storage Management (ASM) supports rolling upgrades**

Starting with Oracle Database 11g Release 1 (11.1), you can perform a *rolling upgrade* of Oracle Automatic Storage Management (ASM) software. You can perform rolling upgrades of your ASM software on ASM instances, one at a time, in an Oracle Clusterware environment while continuing to provide high availability.

See Also: [Rolling Upgrades](#) on page 1-7

- **Automatic Storage Management (ASM) Manageability Enhancements**

The new storage administration features for ASM manageability include the following:

- New attributes for disk group compatibility
 - To enable some of the new ASM features, you can use two new disk group compatibility attributes, `COMPATIBLE.RDBMS` and `COMPATIBLE.ASM`. These attributes specify the minimum software version that is required to use disk groups for the database and for ASM, respectively. This feature enables heterogeneous environments with disk groups from both Oracle Database 10g and Oracle Database 11g. By default, both attributes are set to 10.1. You must advance these attributes to take advantage of the new features.
- New ASM command-line utility (ASMCMD) commands and options
 - ASMCMD allows ASM disk identification, disk bad block repair, and backup and restore operations in your ASM environment for faster recovery.
- ASM fast rebalance

Rebalance operations that occur while a disk group is in RESTRICTED mode eliminate the lock and unlock extent map messaging between ASM instances in Oracle RAC environments, thus improving overall rebalance throughput.

See Also: ["Advance the ASM and Oracle Database Disk Group Compatibility"](#) on page 4-15

For complete information about ASM, see *Oracle Database Storage Administrator's Guide*

- **Data Mining Schema Objects**

Data Mining models are now more similar to other database objects. This enables you to assign access privileges, audit operations and, in general, manage Data Mining models as full database objects.

Starting with Oracle Database 11g Release 1 (11.1), the data mining schema is created automatically as part of `catproc.sql` under the `SYS` account. Existing data mining customers are able to upgrade to Oracle Database 11g Release 1 (11.1) from any previous release without major constraints and product compatibility between releases is strictly maintained.

See Also: ["Oracle Data Mining Models and the DMSYS Schema Objects"](#) on page 5-6 for compatibility information

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](#)
- [Using Optimal Flexible Architecture \(OFA\)](#)
- [Converting Databases to 64-bit Oracle Database Software](#)
- [Rolling Upgrades](#)
- [Moving From the Standard Edition to the Enterprise Edition](#)
- [Upgrading from Oracle Database Express Edition to Oracle Database](#)

Overview of the Database Upgrade Process

This section includes an overview of the major steps required to upgrade an existing Oracle database to Oracle Database 11g Release 1 (11.1). These procedures transform an existing Oracle Database system (including associated applications) into an Oracle Database 11g Release 1 (11.1) system. Oracle Database 11g Release 1 (11.1) 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 11g Release 1 (11.1) features and functions.

Oracle Database 11g Release 1 (11.1) 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

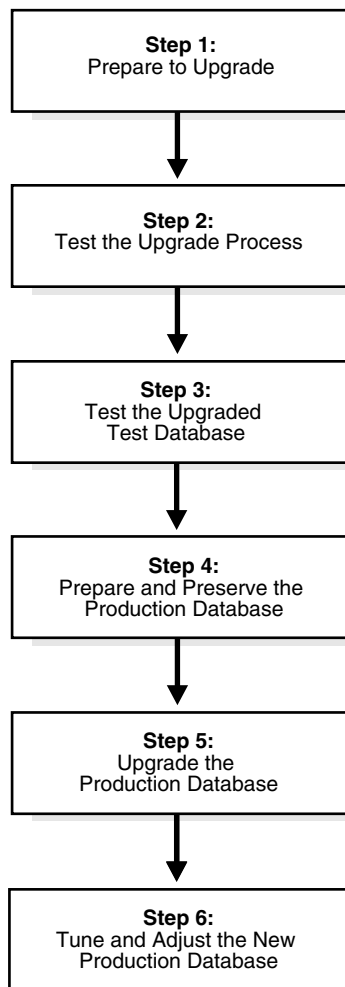
Provides a command-line upgrade of a database, using SQL scripts and utilities.

- **Export and Import utilities**
Uses the Oracle Data Pump Export and Import utilities, available as of Oracle Database 10g 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 11g Release 1 (11.1) 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 11g Release 1 (11.1) 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



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 Oracle Database 11g Release 1 (11.1).
- 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 Oracle Database 11g Release 1 (11.1).
- 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 Oracle Database 11g Release 1 (11.1) 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 Oracle Database 11g Release 1 (11.1).
- 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

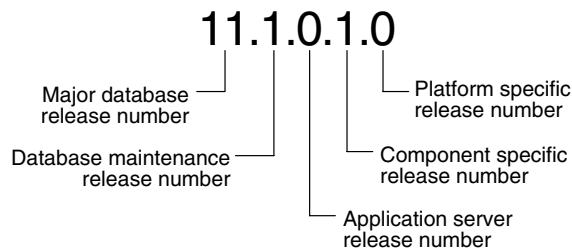
- Tune the Oracle Database 11g Release 1 (11.1) production database. The new production database should perform as well as, or better than, the database prior to the upgrade. [Chapter 4, "After Upgrading to the New Release"](#) describes these adjustments.
- Determine which features of Oracle Database 11g Release 1 (11.1) 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 6, "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 Oracle Database 9i 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 include release 1 (11.1); references to Oracle Database 10g include Oracle Database 10g Release 1 (10.1) and Oracle Database 10g Release 2 (10.2); references to Oracle9i include release 1 (9.0.1) and 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 Oracle Database9i Release 2 (9.2) applies to release 9.2.0.1, release 9.2.0.2, and all other platform-specific releases within 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 1-6 for more information.

Caution: It is not possible to install Oracle Database 11g Release 1 (11.1) 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 Oracle9i and Oracle Database 11g Release 1 (11.1) installed on the same computer, then the Oracle9i database server can access Oracle9i databases but not Oracle Database 11g Release 1 (11.1) databases, and the Oracle Database 11g Release 1 (11.1) database server can access Oracle Database 11g Release 1 (11.1) databases but not Oracle9i databases.

See Also: Your operating system-specific Oracle documentation for more information about running multiple releases of Oracle on your operating system. Restrictions might 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 Oracle8i, Oracle9i, and Oracle Database 10g databases in multiple (separate) Oracle homes on the same computer and have Oracle8i, Oracle9i, and Oracle Database 10g clients connecting to any or all of the databases.

Install Databases in Multiple Oracle Homes on Separate Computers

You can install Oracle8i, Oracle9i, Oracle Database 10g, and Oracle Database 11g Release 1 (11.1) databases in multiple (separate) Oracle homes on separate computers and have Oracle8i, Oracle9i, Oracle Database 10g, and Oracle Database 11g Release 1 (11.1) clients connecting to any or all of the databases.

Upgrade a Database to the Current Release

You can upgrade an Oracle8i, Oracle9i, or Oracle Database 10g database to Oracle Database 11g Release 1 (11.1) and have Oracle8i, Oracle9i, Oracle Database 10g, and Oracle Database 11g Release 1 (11.1) clients connecting to the upgraded database.

Upgrade Clients to the Current Release

You can upgrade any or all of your Oracle8i, Oracle9i, or Oracle Database 10g clients to Oracle Database 11g Release 1 (11.1). The Oracle Database 11g Release 1 (11.1) client can be used to access your Oracle8i, Oracle9i, Oracle Database 10g, and Oracle Database 11g Release 1 (11.1) databases.

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 Release 1 (11.1) 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 Oracle Database 11g Release 1 (11.1).

You must increase initialization parameters affecting the system global area, such as `sga_target` and `shared_pool_size`, to support 64-bit operation.

See Also: Your platform-specific installation guide for additional information

Rolling Upgrades

A rolling upgrade upgrades different databases or different instances of the same database (in an Oracle Real Application Clusters environment) one at a time, without stopping the database. Depending on the method you choose, you can perform a rolling upgrade of the Oracle Database software version, apply patchsets, or apply individual patches (sometimes referred to as one-off patches), incurring little or no database downtime.

Oracle Database 11g Release 1 (11.1) provides the following methods for performing rolling upgrades:

- Oracle Data Guard and logical standby databases

Using SQL Apply and logical standby databases, you can upgrade Oracle Database software and patchsets with little or no database downtime. 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 10g Release 1 (10.1) to Oracle Database 11g Release 1 (11.1).

See Also: ["Upgrading Standby Databases"](#) on page 3-4 for more information.

- Oracle Streams

Using Streams source and destination databases, you can upgrade to a new version 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. These maintenance operations use the features of Oracle Streams to achieve little or no database down time.

See Also: *Oracle Streams Concepts and Administration*

- Oracle Real Application Clusters rolling upgrades with the OPatch Utility

Using the OPatch command-line utility, you can perform rolling patch upgrades with Oracle Real Application Clusters (Oracle RAC), incurring little or no database downtime. Note that the OPatch utility can only be used to apply individual patches, not patchset releases. See *Oracle Universal Installer and OPatch User's Guide* for more information about the OPatch utility.

The advantage of a Oracle RAC rolling upgrade is that it enables at least some instances of the Oracle RAC installation to be available during the scheduled outage required for patch upgrades. Only the Oracle RAC instance that is currently being patched must be brought down; the other instances can continue to remain available. This means that the effect on the application downtime required for such scheduled outages is further minimized. The Oracle OPatch utility enables you to apply the patch successively to the different instances of the Oracle RAC installation.

See Also: *Oracle Database High Availability Best Practices*

- Oracle Clusterware rolling upgrades using the Oracle Universal Installer (OUI)

Using the Oracle Universal Installer (OUI) that is downloaded with the patch update, you can perform rolling upgrades to upgrade Oracle Clusterware or Oracle Cluster Ready Services. You can perform a rolling upgrade to apply patchset releases.

The advantage of an Oracle Clusterware rolling upgrade is that it enables at least some instances of the Oracle RAC installation to be available during the scheduled outage required for patchset upgrades. Only the node that is currently being patched must be brought down; the other instances remain available. This means that the effect on the application downtime required for such scheduled outages is further minimized. OUI enables you to apply the patchset successively to the different instances of the Oracle Clusterware installation.

See Also: The "How to Perform Oracle Clusterware Rolling Upgrades" discussion in your platform-specific Oracle Clusterware installation guides for step-by-step instructions to help you upgrade your Oracle Clusterware. Also, see *Oracle Universal Installer and OPatch User's Guide* for information about using OUI.

- Automatic Storage Management (ASM) Rolling Upgrades

Starting with Oracle Database 11g Release 1 (11.1), you can perform a *rolling upgrade* of Oracle Automatic Storage Management software from Oracle Database 11g Release 1 (11.1) to a later release. ASM rolling upgrades enable you to independently upgrade or patch clustered ASM nodes without affecting database availability, thus providing greater uptime. During a rolling upgrade, all of the features of a clustered ASM environment continue to function even while one or more of the nodes in the cluster are running on different software versions. This feature provides greater availability and enables you to upgrade ASM software from one release to the next more gracefully.

See Also: Your platform-specific Oracle Clusterware installation guide for instructions on how to perform a rolling upgrade, and the *Oracle Database Storage Administrator's Guide* for more information about ASM

Moving From the Standard Edition to the Enterprise Edition

If you have a Standard Edition database at a release prior to Oracle Database 11g Release 1 (11.1), 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 manual.

To change your existing Oracle Database 11g 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.

Upgrading from Oracle Database Express Edition to Oracle Database

In order to upgrade Oracle Database 10g Express Edition (Oracle Database XE) to Oracle Database 11g, you need to install Oracle Database 11g on the same system as Oracle Database XE and use Database Upgrade Assistant to perform the upgrade.

Preparing to Upgrade

This chapter describes the steps to complete before upgrading a database to Oracle Database 11g Release 1 (11.1). 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 Oracle Database 11g Release 1 (11.1). *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 11g Release 1 (11.1) 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 Oracle Database 11g Release 1 (11.1) depends on the release number of your current database. It might not be possible to upgrade directly from your current version of Oracle Database to the latest version. Depending on your current release, you might be required to upgrade through one or more intermediate releases to upgrade to Oracle Database 11g Release 1 (11.1).

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 9.2.0.8 using the instructions in *Oracle9i Database Migration Release 2* (9.2).
3. Upgrade release 9.2.0.8 to Oracle Database 11g Release 1 (11.1) 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	<p>Direct upgrade is <i>not</i> supported. Upgrade to an intermediate Oracle Database release before you can upgrade to Oracle Database 11g Release 1 (11.1), as follows:</p> <ul style="list-style-type: none"> ■ 7.3.3 (or lower) -> 7.3.4 -> 9.2.0.8 -> 11.1 ■ 8.0.5 (or lower) -> 8.0.6 -> 9.2.0.8 -> 11.1 ■ 8.1.7 (or lower) -> 8.1.7.4 -> 9.2.0.8 -> 11.1 ■ 9.0.1.3 (or lower) -> 9.0.1.4 -> 9.2.0.8 -> 11.1 <p>When upgrading to an intermediate Oracle Database release, follow the instructions in the intermediate release's documentation. Then, upgrade the intermediate release database to Oracle Database 11g Release 1 (11.1) using the instructions in Chapter 3, "Upgrading to the New Release".</p>
9.2.0.4 10.1.0.2 10.2.0.1	<p>Direct upgrade to Oracle Database 11g Release 1 (11.1) is supported from 9.2.0.4 or higher, 10.1.0.2 or higher, and 10.2.0.1 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 11g release 1 (11.1). See "Upgrading an Oracle Real Application Clusters (Oracle RAC) Database" on page 3-2.</p> <p>For release 9.2.0.3 you must first upgrade to an intermediate Oracle Database release, as follows:</p> <p>9.2.0.3 (or lower) -> 9.2.0.8 -> 11.1</p> <p>To upgrade to Oracle Database 11g Release 1 (11.1), follow the instructions in Chapter 3, "Upgrading to the New Release".</p>

Note: See [Supported Releases for Downgrading](#) in [Chapter 7](#) 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 Oracle Database 11g Release 1 (11.1):

- [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 Oracle Database 11g Release 1 (11.1). 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) and Automatic Storage Management (ASM).

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

Support for Automatic Storage Management DBUA supports upgrades of databases that use Automatic Storage Management (ASM). If an ASM instance is detected, then you have the choice of updating both the database and ASM or only the ASM instance.

See Also: [Upgrade a Database Using Database Upgrade Assistant](#) on page 3-16

Manual Upgrade

A manual upgrade consists of running SQL scripts and utilities from a command line to upgrade a database to Oracle Database 11g Release 1 (11.1).

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 Upgrade Information Tool is a SQL script that ships with Oracle Database 11g Release 1 (11.1), and must be run in the environment of the database being upgraded.

The Upgrade Information Tool displays warnings about possible upgrade issues with the database. It also displays information about required initialization parameters for Oracle Database 11g Release 1 (11.1).

- 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 Oracle Database 11g Release 1 (11.1), and should be run in the environment of the new release.

See Also: ["Upgrade a Database Manually"](#) on page 3-37

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 10g 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 11g Release 1 (11.1) database. Export/Import can copy a subset of the data in a database, leaving the original database unchanged.

When upgrading from Oracle Database 10g 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 Oracle Database 11g Release 1 (11.1) loads the exported data into a new database. However, the new Oracle Database 11g Release 1 (11.1) database must already exist before it can be loaded from the export dump file.

When importing data from an earlier release, the Oracle Database 11g Release 1 (11.1) Import utility 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 11g Release 1 (11.1) 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 11g Release 1 (11.1) 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 8, "Moving Data Using Data Pump and Export/Import"](#)

Choose an Oracle Home Directory

You must choose an Oracle home directory for Oracle Database 11g Release 1 (11.1) 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 11g patchset release. For a patchset release, you can use the same Oracle Database 11g Release 1 (11.1) 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](#)
- [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 Oracle Database 11g Release 1 (11.1), 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. But 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.

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 11g Release 1 (11.1) 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 Oracle Database 11g Release 1 (11.1), 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 11g Release 1 (11.1) 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 Oracle Database9i Release 2 (9.2). If you are upgrading from Oracle Database9i 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 Beginning with Oracle Database 11g 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 Performance Tuning Guide* for complete information about how to capture and replay workloads

SQL Performance Analyzer Beginning with Oracle Database 11g 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 Performance Tuning Guide* for complete information and examples using the SQL Performance Analyzer to perform what-if analysis on potential database changes

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

performance of corresponding SQL statements, regardless of changes occurring in the system.

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 version 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 6-2 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 Oracle Database 11g Release 1 (11.1). 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 6, "Upgrading Your Applications"](#) for more information on using applications with Oracle Database

Upgrading to the New Release

This chapter guides you through the process of upgrading a database to Oracle Database 11g Release 1 (11.1). 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](#)
- [Prepare the Database to Be Upgraded](#)
- [Run the Oracle Net Configuration Assistant](#)
- [Upgrade a Database Using Database Upgrade Assistant](#)
- [Upgrade a Database Manually](#)
- [Upgrade an ASM Instance with DBUA](#)
- [Upgrade an ASM Instance Manually](#)

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 its Move Database Files option during the upgrade. See "[Upgrade a Database Using Database Upgrade Assistant](#)" on page 3-16 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:

- [Upgrading Oracle Clusterware and Automatic Storage Management \(ASM\) Databases](#)
- [Upgrading with Read-Only and Offline Tablespaces](#)
- [Upgrading Standby Databases](#)
- [Upgrading Your Operating System](#)
- [Migrating Data to a Different Operating System](#)

Upgrading Oracle Clusterware and Automatic Storage Management (ASM) Databases

Prior to Oracle Database 11g Release 1 (11.1), all Oracle software installations were owned either 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`. Beginning with Oracle Database 11g Release 1 (11.1), the same user that owned the Oracle Clusterware 10g software must perform the Oracle Clusterware 11g release 1 (11.1) upgrade. Also, the Oracle Clusterware 11g release 1 (11.1) software must be upgraded in the same directory where the Oracle Clusterware 10g software was installed. Oracle Clusterware was installed in a separate home directory, but Oracle Database installations and ASM installations could be installed in the same Oracle home.

With Oracle Database 11g Release 1 (11.1), to improve data and software security, Oracle Database, ASM, and Oracle Clusterware may each be installed in separate home directories, and access to each of these directories may be controlled by creating separate operating system (OS) users.

Upgrading an Oracle Real Application Clusters (Oracle RAC) Database

If you are upgrading an Oracle RAC database with DBUA, then all necessary configuration is done 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.x is at release 10.2.0.3 (or higher), before you attempt to upgrade it to Oracle Clusterware 11g release 1 (11.1). If this check fails, then you are instructed to apply Oracle Clusterware patchset release 10.2.0.3.0 or later to your existing release 10.2.0.1 or 10.2.0.2 before it can be upgraded. All other upgrade paths and fresh install cycles are unaffected by this prerequisite check.

Upgrading a Database that Uses Automatic Storage Management (ASM)

Beginning with Oracle Database 11g Release 1 (11.1), you should use the `SYSASM` privilege to separate database management and storage management responsibilities.

Also beginning with Oracle Database 11g Release 1 (11.1), 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 Oracle Database 11g Release 1 (11.1), 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 user_password
```

Note: This step is necessary only when upgrading the ASM instance. It is not necessary when upgrading a database to Oracle Database 11g Release 1 (11.1) without upgrading ASM.

If the default Oracle Database 11g Release 1 (11.1) 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.

Each node's ASM instance uses this SQL*Plus statement to update its own password file. The password file itself has to be created by the `ORAPWD` utility. Note that new users have no privileges until they are granted access using the `GRANT system_privilege TO user` statement. This enables OS authentication control to allow any user authenticated with SYSDBA privileges on the database instance to have limited access to certain ASM operations, without automatically granting that user full SYSDBA privileges on the ASM instance. Then, after the upgrade, users are able to connect to the ASM instance using either of the following connection types:

- `CONNECT AS SYSASM` or `SQLPLUS sys/sys_password AS SYSASM`
Connecting as SYSASM grants the user full access to all of the available ASM disk groups.
- `CONNECT AS SYSDBA` or `SQLPLUS sys/sys_password AS SYSDBA`
Connecting as SYSDBA grants the user limited privileges. Every time a user authenticated as SYSDBA executes a command that only a user authenticated as SYSASM is able to do, a warning is written to the alert log files to warn the user to connect as SYSASM.

See Also:

- *Oracle Database Storage Administrator's Guide* for more information about ASM
- *Oracle Database SQL Language Reference* for complete syntax about `CREATE USER`, `ALTER USER`, `DROP USER`, `GRANT system_privilege TO user`, and `REVOKE system_privilege FROM user`

Upgrading with Read-Only and Offline Tablespaces

Oracle Database 11g Release 1 (11.1) 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 11g 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 Standby Databases

To upgrade the Oracle Database software when physical or logical standby databases are present in an Oracle Data Guard configuration, see the *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*.

Starting with Oracle Database 10g Release 1 (10.1.0.3), you can use SQL Apply on a logical standby database to perform a *rolling upgrade* to Oracle Database 11g Release 1 (11.1). During a rolling upgrade, you can run different releases of Oracle Database software on the primary and logical standby databases while you upgrade them, one at a time, incurring minimal downtime on the primary database. See the *Oracle Data Guard Concepts and Administration* documentation for complete information.

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 on 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 Oracle9i database on Solaris to an Oracle Database 11g Release 1 (11.1) 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 Oracle Database 11g Release 1 (11.1) on the current operating system platform following the instructions in this guide.
2. Use the Oracle Database 11g Release 1 (11.1) cross-platform transportable tablespace feature or the Oracle Data Pump Export and Import utilities to migrate the upgraded database to the different operating system.

See Also:

- *Oracle Database Backup and Recovery User's Guide* for more information on transportable tablespaces
- [Chapter 8, "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.

Install the New Oracle Database Software

Installation of the release 11.1 Oracle software involves upgrading Cluster Ready Services (CRS) if necessary, and installing Oracle Database 11g Release 1 (11.1).

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

1. If you are upgrading an Oracle Real Application Clusters (Oracle RAC) database, then you must perform the following steps in the order shown:
 - a. Mount the Oracle Clusterware 11g release 1 (11.1) 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 Clusterware 11g release 1 (11.1).
 - c. If necessary, perform patch upgrades of the previous release Oracle Clusterware or Oracle Cluster Ready Services software to the most recent patch version.

See Also: Your platform-specific Oracle Clusterware Installation Guide for patch upgrade instructions

- d. Run the `preupdate.sh` script, located in the `clusterware/upgrade` directory on the Oracle Clusterware 11g release 1 (11.1) installation media.

This script shuts down the Oracle Clusterware software stack and unlocks the Oracle Clusterware home (CRS home) for upgrading.

If you do not have a shared CRS home, then you must run this script on each node in the cluster that you want to upgrade.

Note: You only need to run this script if upgrading from Oracle Database 10g. You do not need to run this script if upgrading from Oracle Database 9i.

- e. Ensure that you are logged in as the user that you want to own the Oracle Clusterware installation, and run the Oracle Clusterware installation. For example, if the owner you want to use for Oracle Clusterware is `crs`, and the mountpoint for Oracle Clusterware 11g release 1 (11.1) is `/shiphome/clusterware`, then you could use the following commands:

```
> whoami
   crs
> /shiphome/clusterware/runInstaller
```

Provide information as prompted by the installer.

- f. When prompted, open a separate terminal session, log in as root, and run the rootupgrade script.

Note: If you are upgrading from Oracle Database 10g to Oracle Database 11g, you must install Oracle Clusterware into the existing CRS home on the local and remote node subset. When you run the root script, it starts the Oracle Clusterware 11g release 1 (11.1) stack on the subset cluster nodes. For every node except the last, the version change does not take effect. The version change only takes effect on the last node.

For information on how to perform rolling upgrades, see *Oracle Clusterware Installation Guide for Linux*.

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 3-7.) 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 11g Release 1 (11.1) 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 Vault installation guide for more information.

See Also:

- "[Run the Oracle Net Configuration Assistant](#)" on page 3-15
- "[Upgrade a Database Using Database Upgrade Assistant](#)" on page 3-16

Install the Latest Available Patchset Release and Any Required Patches

After you have installed Oracle Database 11g Release 1 (11.1), 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. Patchsets provide bug fixes only; they do not include new features or functions, and do not require certification on the target system.

- 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 likely to be applied, or that are prerequisites for the security fixes.

The latest patchset release and critical patch update for Oracle Database 11g Release 1 (11.1) 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 11g Release 1 (11.1) 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 necessary step if you are upgrading manually. It 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 11g Release 1 (11.1), 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 11g Release 1 (11.1) Oracle home directory.
2. Copy the Pre-Upgrade Information Tool (`utlu111i.sql`) from the Oracle Database 11g Release 1 (11.1) `ORACLE_HOME/rdbms/admin` directory to a directory outside of the Oracle home, such as the temporary directory on your system.
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 files 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> @utlu111i.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 11.1 Pre-Upgrade Information Tool    06-15-2007 09:14:02
.
*****
Database:
*****
--> name:          ORCL
--> version:       10.2.0.1.0
--> compatible:    10.2.0.1
--> 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 11.1 init.ora or spfile]
*****
WARNING: --> "sga_target" needs to be increased to at least 388 MB
.
*****
Renamed Parameters: [Update Oracle Database 11.1 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 11.1 init.ora or spfile]
*****
--> "max_enabled_roles"
--> "remote_os_authent"
--> "background_dump_dest" replaced by "diagnostic_dest"
--> "user_dump_dest" replaced by "diagnostic_dest"
--> "core_dump_dest" replaced by "diagnostic_dest"
```



```

.
*****
Components: [The following database components will be upgraded or installed]
*****
--> Oracle Catalog Views           [upgrade]  VALID
--> Oracle Packages and Types      [upgrade]  VALID
--> JServer JAVA Virtual Machine   [upgrade]  VALID
--> Oracle XDK for Java            [upgrade]  VALID
--> Oracle Workspace Manager       [upgrade]  VALID
--> 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
--> Oracle Ultra Search            [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 an old timezone file version.
... Patch the 10.2.0.1.0 database to timezone file version 4
... BEFORE upgrading the database. Re-run utlu111i.sql after
... patching the database to record the new timezone file version.
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.
... 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.
.

```

Note: Oracle *interMedia* became Oracle Multimedia in Oracle Database 11g 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. New files of at least 4 MB (preferably 10 MB) must be created in the current database. Any redo log files less than 4 MB must be dropped before the database is upgraded.

Tablespaces

This section displays a list of tablespaces in the current database. For each tablespace, the tablespace name and minimum required size is displayed. In addition, a message is displayed if the tablespace is adequate for the upgrade. If the tablespace does not have enough free space, then space must be added to the tablespace in the current database. Tablespace adjustments must be made before the database is upgraded.

Update Parameters

This section displays a list of initialization parameters in the parameter file of the current database that must be adjusted before the database is upgraded. The adjustments must be made to the parameter file after it is copied to Oracle Database 11g Release 1 (11.1).

See Also: [Appendix A, "Initialization Parameter and Data Dictionary Changes"](#) for more information about changes to initialization parameters in Oracle Database 11g Release 1 (11.1)

Deprecated Parameters

This section displays a list of initialization parameters in the parameter file of the current database that are deprecated in Oracle Database 11g Release 1 (11.1).

See Also: ["Deprecated Initialization Parameters"](#) on page A-1 for a list of initialization parameters that are deprecated in Oracle Database 11g Release 1 (11.1)

Obsolete Parameters

This section displays a list of initialization parameters in the parameter file of the current database that are obsolete in Oracle Database 11g Release 1 (11.1). Obsolete initialization parameters must be removed from the parameter file before the database is upgraded.

See Also: ["Obsolete Initialization Parameters"](#) on page A-3 for a list of initialization parameters that are obsolete in Oracle Database 11g Release 1 (11.1)

Components

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

Miscellaneous Warnings

This section 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 Oracle Database 11g Release 1 (11.1). The *SYSAUX* tablespace must be created after the new release is started and *before* the upgrade scripts are invoked.

Prepare the Database to Be Upgraded

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 Oracle Database 11g Release 1 (11.1):

- [Deprecated CONNECT Role](#)
- [Database Links with Passwords](#)
- [TIMESTAMP WITH TIME ZONE Data Type](#)
- [Optimizer Statistics](#)
- [Disable Oracle Database Vault](#)
- [Save Oracle Enterprise Manager Database Control Data](#)

Deprecated CONNECT Role

After upgrading to Oracle Database 11g Release 1 (11.1) from Oracle Database9i Release 2 (9.2) or Oracle Database 10g 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.

Database Links with Passwords

During the upgrade to Oracle Database 11g Release 1 (11.1) from Oracle Database9i Release 2 (9.2) or Oracle Database 10g 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 11g Release 1 (11.1) have been updated from version 2 to version 4 to reflect changes in transition rules for some time zone regions. The changes might affect existing data of `TIMESTAMP WITH TIME_ZONE` data type. If you have not already updated your existing databases to time zone file version 4 (patch 4689959), you must do so prior to upgrading to Oracle Database 11g Release 1 (11.1).

If your existing databases are already using a time zone file version greater than 4, then after installing Oracle Database 11g Release 1 (11.1), you must patch the `ORACLE_HOME` with the appropriate patch for the time zone file version in use with the databases you will be upgrading.

If the time zone file version used by the Oracle Database 11g Release 1 (11.1) server does not match the time zone file version that was used with the existing database being upgraded, then the upgrade script will terminate without upgrading the database. The `TIMESTAMP WITH TIME_ZONE` data stored in the database can be 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

Optimizer Statistics

When upgrading to Oracle Database 11g Release 1 (11.1), optimizer statistics are collected for dictionary tables that lack statistics. This statistics collection can be time consuming for databases with a large number of dictionary tables, 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 10g 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 Oracle Database 9i 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 3–1](#) lists the system components and schemas.

Table 3–1 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 Data Vault	DVSYSS
Oracle Enterprise Manager	SYSMAN
Oracle Expression Filter	EXFSYS

Table 3–1 (Cont.) Statistics Collection for System Components and Schemas

Component Name	Schema
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

Disable 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:

- ["Enable Oracle Database Vault"](#) on page 4-4
- *Oracle Database Vault Administrator's Guide* for instructions on disabling Oracle Database Vault

Save Oracle Enterprise Manager Database Control Data

In order to downgrade Oracle Enterprise Manager Database Control after upgrading to Oracle Database 11g Release 1 (11.1), 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 Oracle Database 11g Release 1 (11.1) home. Before running the utility you must install Oracle Database 11g Release 1 (11.1) and invoke the script from the Oracle Database 11g Release 1 (11.1) home. 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 Oracle Database 11g Release 1 (11.1).
2. Set `ORACLE_HOME` to your old Oracle home.
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 Database 11g Release 1 (11.1) home.
6. Execute one of the following:

- a. 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
```

- b. 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 10g 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/dbcdire5 -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/racdwwgrd_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.

Run the Oracle Net Configuration Assistant

If you are upgrading from Oracle9i and a listener was not configured in the Oracle9i repository, then run Oracle Net Configuration Assistant to configure the listening protocol address and service information for the Oracle Database 11g Release 1 (11.1) database, including a `listener.ora` file. A new version of the listener is required for an Oracle Database 11g Release 1 (11.1) database. Previous versions of the listener are not supported for use with an Oracle Database 11g Release 1 (11.1) database. However, it is possible to use the new version of the listener with previous versions of Oracle Database.

Note: It is important to run Oracle Net Configuration Assistant *before* running DBUA.

If you are upgrading an Oracle Real Applications Cluster (Oracle RAC) database, then you have two options:

- Upgrade the Oracle RAC database with DBUA, which automatically migrates the listener from your 10g release Oracle home to the new Oracle Database 11g Release 1 (11.1) home.
- If you are upgrading from 9i or upgrading manually without using DBUA, 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 new Oracle Database 11g Release 1 (11.1) home 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 11g Release 1 (11.1) 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 3-15

If you are upgrading an existing Oracle Enterprise Manager Database Control or if you are configuring a new Oracle Enterprise Manager Database Control for Oracle Database 11g Release 1 (11.1), 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 with 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 Oracle Database 11g Release 1 (11.1) 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 11g Release 1 (11.1) 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 will return an error asking you to disable Database Vault prior to upgrade. See "[Disable Oracle Database Vault](#)" on page 3-13

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

During the upgrade, DBUA automatically modifies or creates new required tablespaces and invokes the appropriate upgrade scripts. 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 11g Release 1 (11.1) 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**.

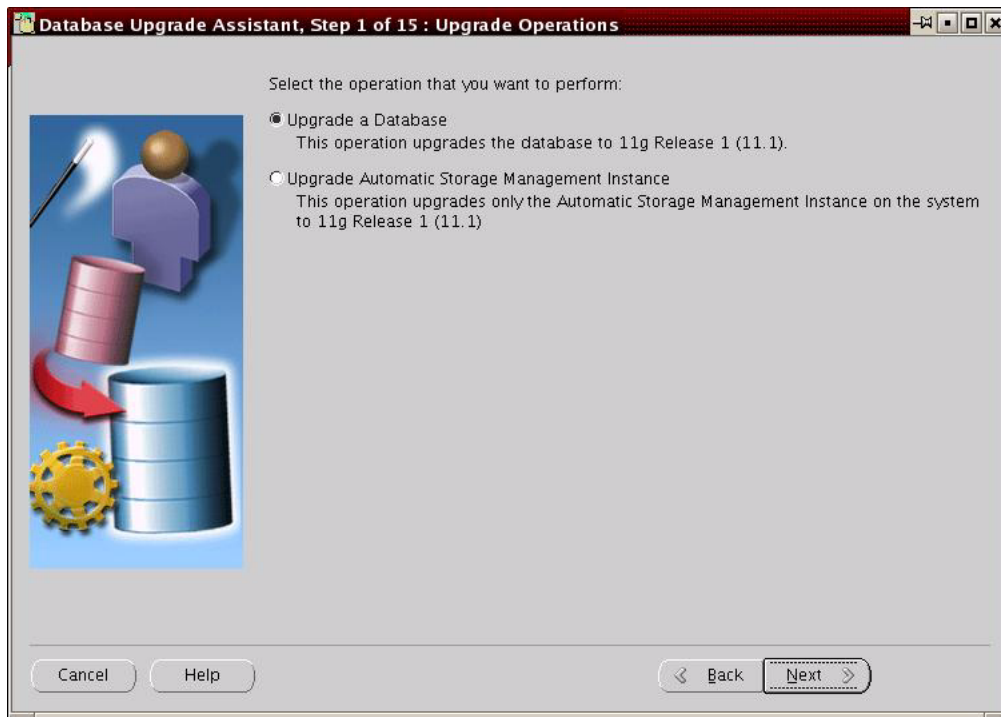
The DBUA Welcome screen appears.



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**.

If an Automatic Storage Management (ASM) instance is detected on the system, then the Upgrade Operations page appears with options to upgrade a database or an ASM instance. If no ASM instance is detected, then Databases screen appears.



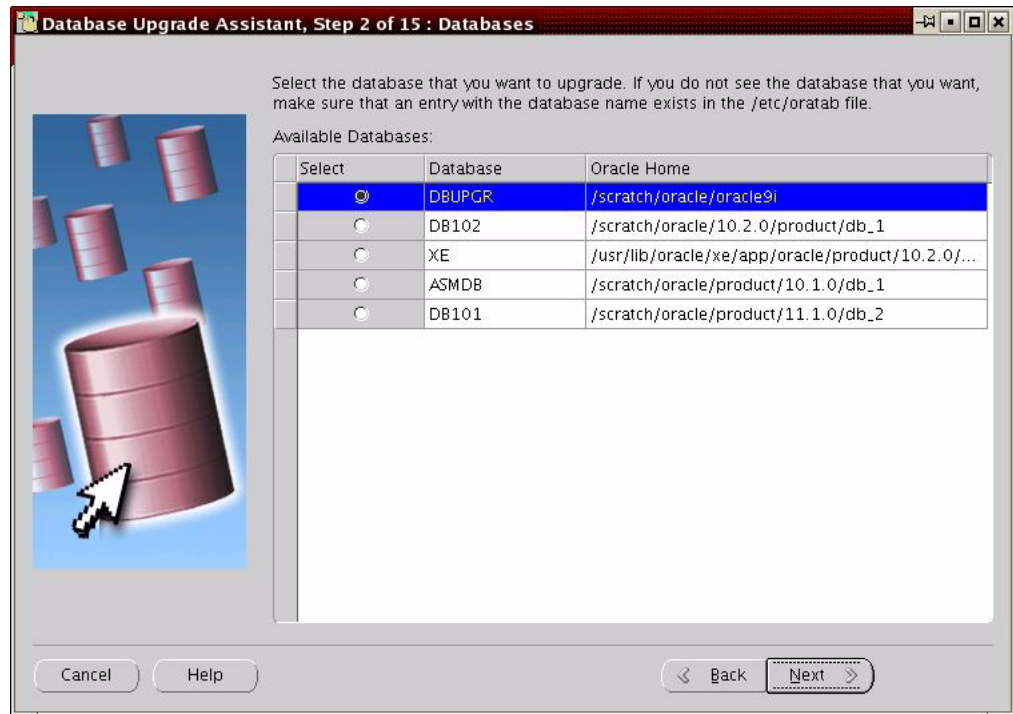
- At the Upgrade Operations page, select **Upgrade a Database**. This operation upgrades a database to Oracle Database 11g Release 1 (11.1).

The other option, **Upgrade Automatic Storage Management Instance**, is discussed in "[Upgrade an ASM Instance with DBUA](#)" on page 3-48.

Oracle recommends that you upgrade the database and ASM in separate DBUA sessions, in separate Oracle homes.

Click **Next**.

The Databases screen appears.



- Select the database you want to upgrade from the **Available Databases** table.

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.

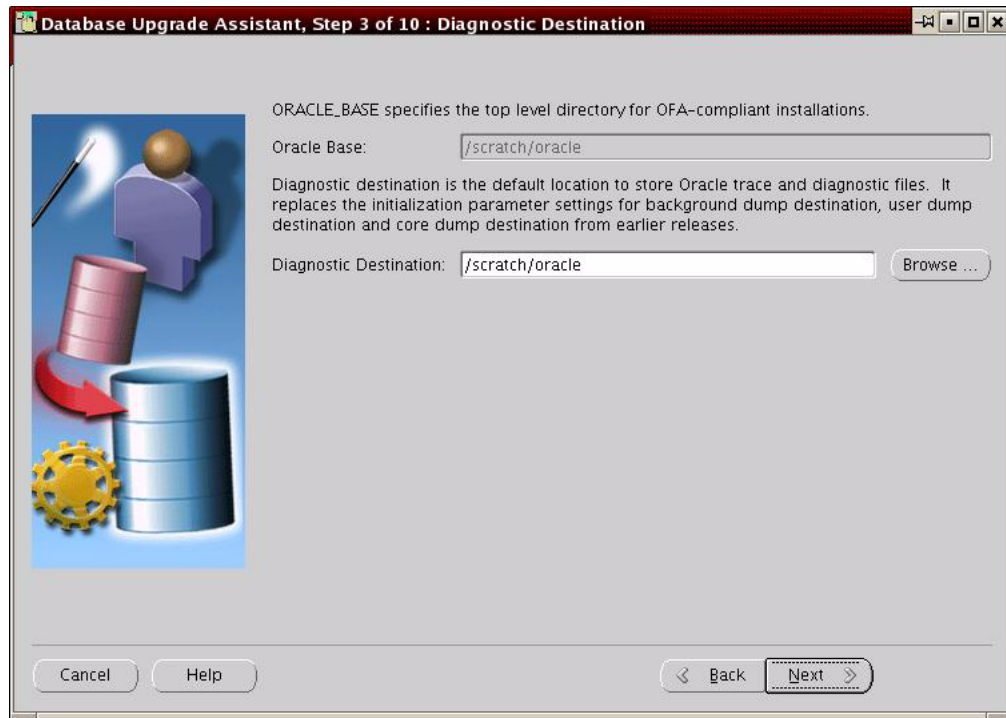
If you are running DBUA from a user account that does not have `SYSDBA` privileges, then you must enter the user name and password credentials to enable `SYSDBA` privileges for the selected database.

Click **Next**.

DBUA analyzes the database, performing the following pre-upgrade checks and displaying warnings as necessary:

- 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 Diagnostic Destination screen appears.



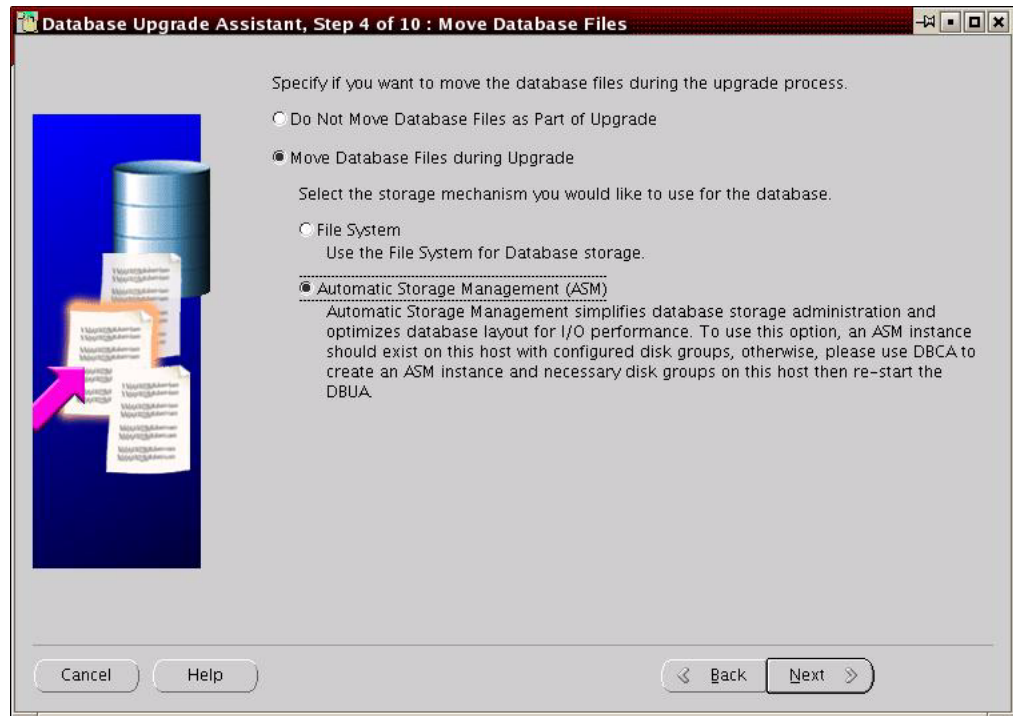
5. Do one of the following:

- Accept the default location for your diagnostic destination
- Enter the full path to a different diagnostic destination in the **Diagnostic Destination** field
- Click **Browse** to select a diagnostic destination

Diagnostic destination is the default location to store Oracle trace and diagnostic files. It replaces the initialization parameter settings for background dump destination, user dump destination, and core dump destination from earlier releases.

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 appear.



6. 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 stored on the host file system.

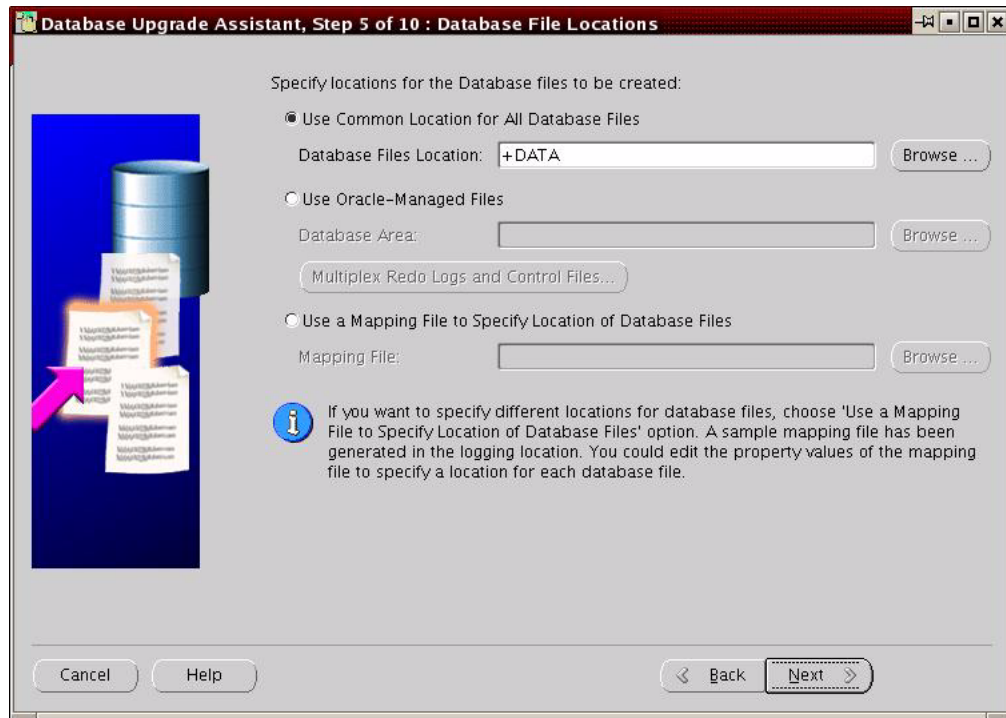
- **Automatic Storage Management (ASM)**

Your database files are stored on 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 Clusterware Installation Guide for more information on installing and configuring ASM

Click **Next**.

The Database File Locations screen appears.



7. Select one of the following options:

- **Use Common Location for All Database Files**

If you choose to have all of your database files in one location, then you must also do one of the following:

- Accept the default location for your database files
- Enter the full path to a different location in the **Database Files Location** field
- Click **Browse** and select a different location for your database files

- **Use Oracle-Managed Files**

If you choose to use Oracle-Managed Files for your database files, then you must also do one of the following:

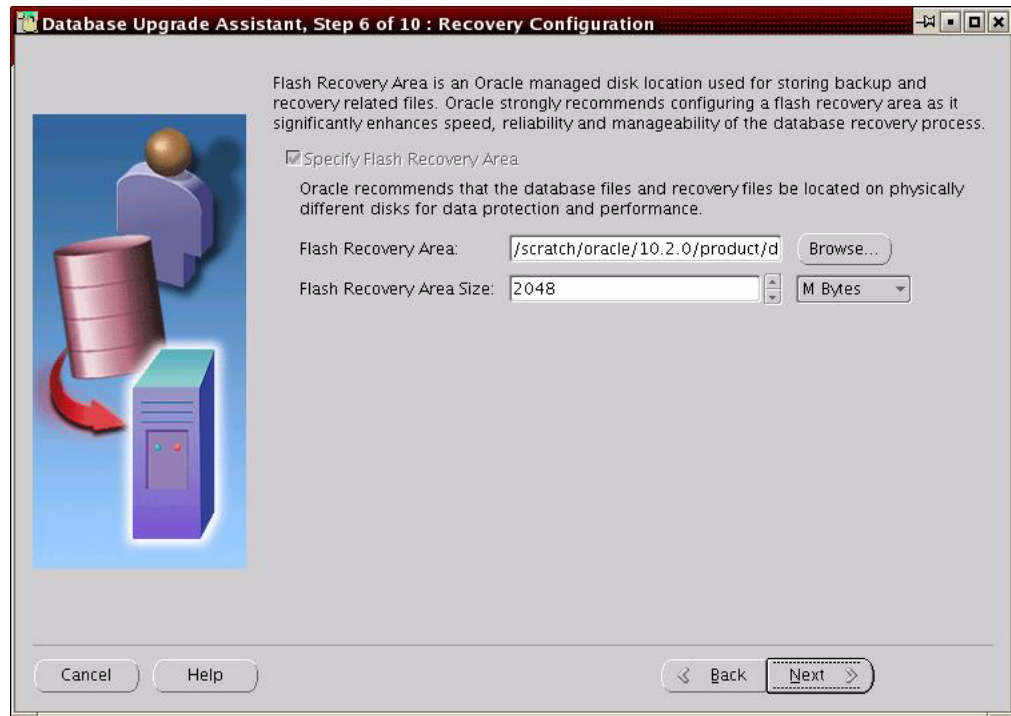
- Accept the default database area
- Enter the full path to a different database area in the **Database Area** field
- Click **Browse** and select a different database area

- **Use a Mapping File to Specify Location of Database Files**

This option enables you to specify different locations for your database files. A sample mapping file is available in the logging location. You can edit the property values of the mapping file to specify a different location for each database file.

Click **Next**.

The Recovery Configuration screen appears, where you can designate a Flash Recovery Area for your database.



Flash Recovery Area is an Oracle managed disk location used for storing backup and recovery related files. Oracle strongly recommends configuring a Flash Recovery Area, because it significantly enhances speed, reliability, and manageability of the database recovery process. This location 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 6, or if an Oracle Express Edition database is being upgraded to Oracle Enterprise Edition, then a Flash recovery Area must be configured. If a Flash Recovery Area is already configured, then current settings are retained but the screen will come up to allow you to override these values.

8. Do one of the following:
 - Accept the default Flash Recovery Area location
 - Enter the full path to a different location in the **Flash Recovery Area** field
 - Click **Browse** and select a different Flash Recovery Area location

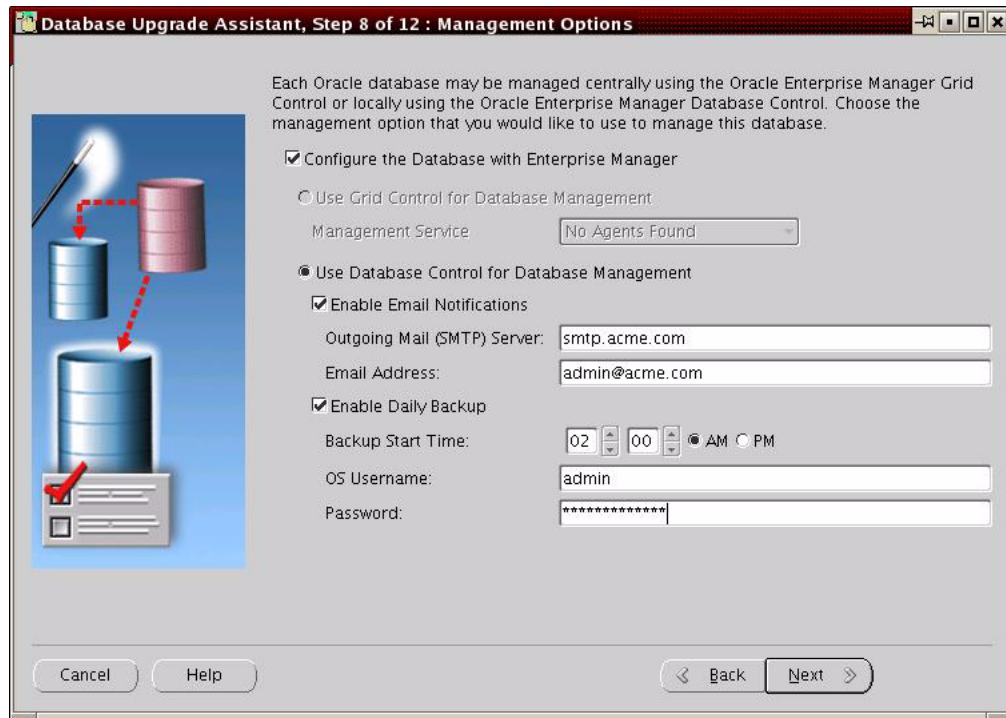
Do one of the following:

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

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

Click **Next**.

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



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.

9. 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:

- **Use Grid Control for Database 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 the Oracle Enterprise Manager Grid Control.

- **Use Database Control for Database 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 Email 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 **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 Backup**
- Enter a start time in the **Backup Start Time** field
- Enter host credentials in the **OS Username** and **Password** fields

If you select this 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.

Click **Next**.

The Database Credentials screen appears.

Database Upgrade Assistant, Step 9 of 12 : Database Credentials

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

Use Different Passwords

User Name	Password	Confirm Password
DBSNMP		
SYSMAN		

Use the Same Password for All Accounts

Password:

Confirm Password:

Cancel Help < Back Next >

10. 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.

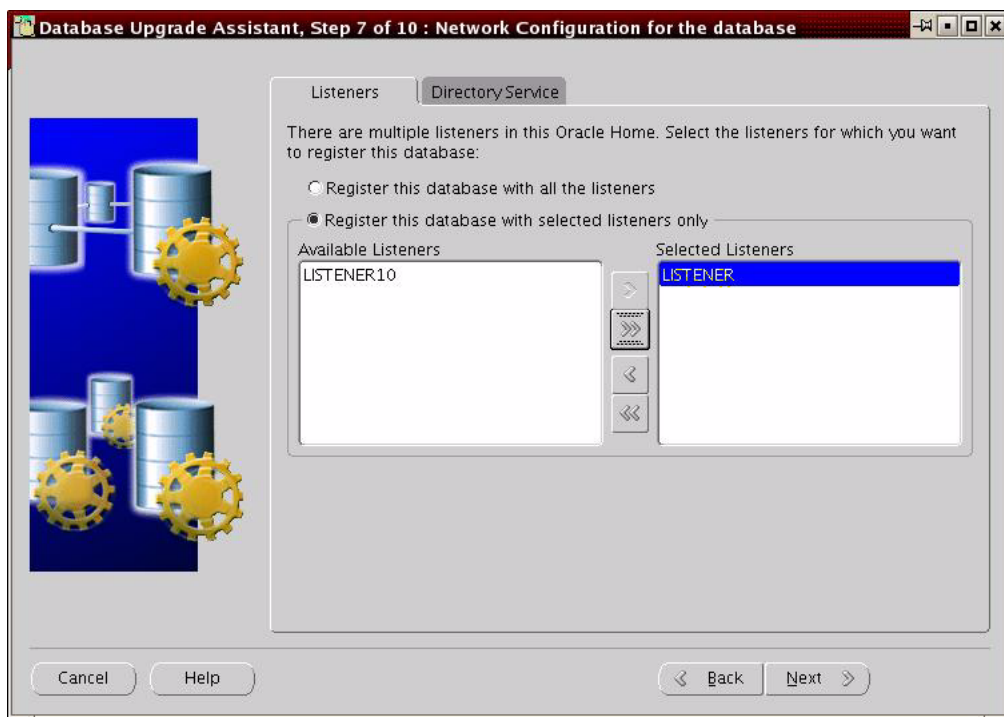
- **Use the Same Password for All Accounts**

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 11g Release 1 (11.1) 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.

Click **Next**.

If DBUA detects multiple listeners are configured in the Oracle Database 11g Release 1 (11.1) home, then the Network Configuration for the Database screen appears.



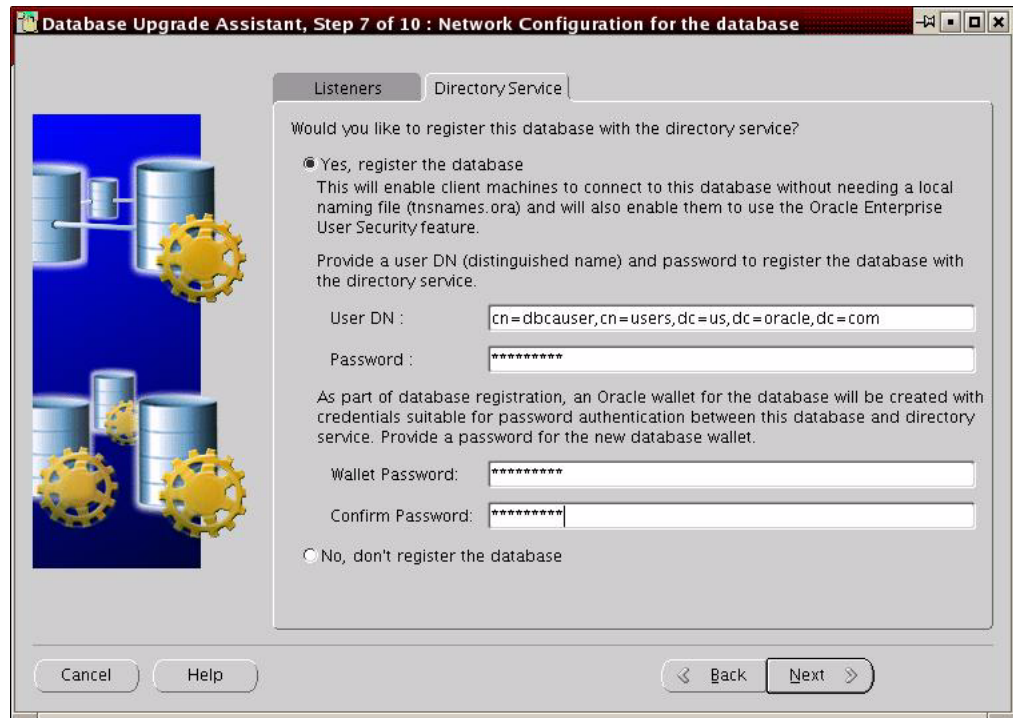
11. The Network Configuration screen has two tabs. The Listeners tab is displayed if you have more than one listener in the Oracle Database 11g Release 1 (11.1) Oracle home. The Directory Service tab shows up if you have directory services configured in the Oracle Database 11g Release 1 (11.1) Oracle home.

On the Listeners tab, select one of the following options:

- **Register this database with all the listeners**
- **Register this database with selected listeners only**

If you choose to register selected listeners only, then you must select the listeners you want in the **Available Listeners** list and use the arrow buttons to move them to the **Selected Listeners** list.

If you want to register your database with a directory service, then click the **Directory Service** tab.



On the Directory Service tab, select one of the following options:

- **Yes, register the database**

Selecting this option enables client computers to connect to this database without a local name file (tnsnames.ora) and also enables them to use the Oracle Enterprise User Security feature.

See Also: *Oracle Database Enterprise User Security Administrator's Guide*

- **No, don't register the database**

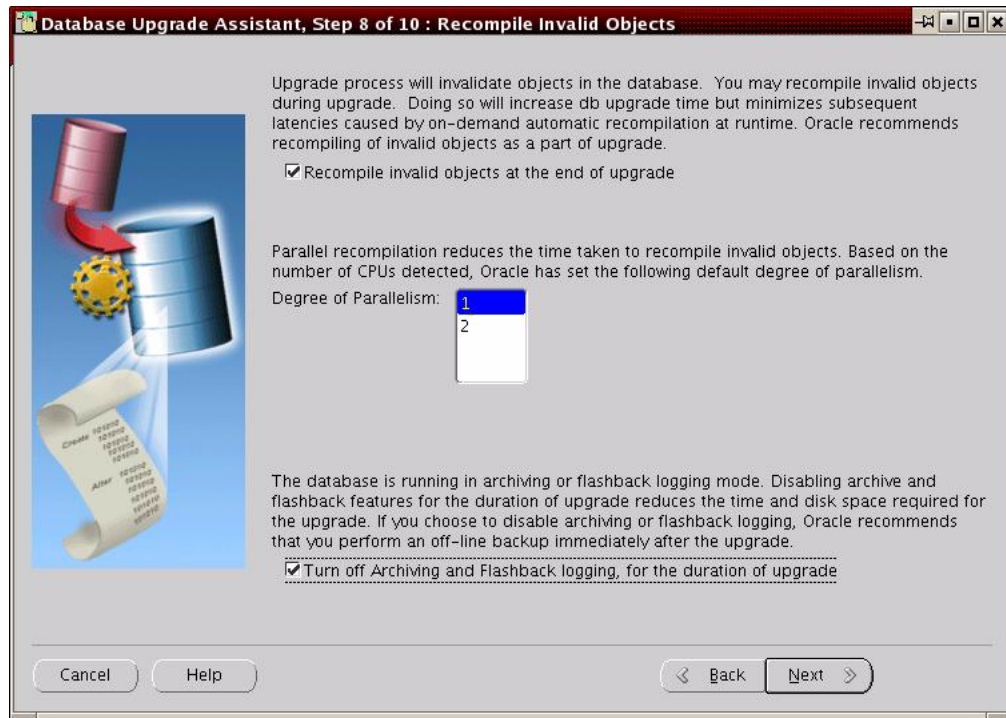
If you choose to register the database, then you must also provide a user distinguished name (DN) in the **User DN** field and a password for that user in the **Password** field.

Note: If the default Oracle Database 11g Release 1 (11.1) 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.

An Oracle wallet is created as part of database registration. It contains credentials suitable for password authentication between this database and the directory service. Enter a password in the **Wallet Password** and **Confirm Password** fields.

Click **Next**.

The Recompile Invalid Objects screen appears.



12. 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 11g Release 1 (11.1), 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.

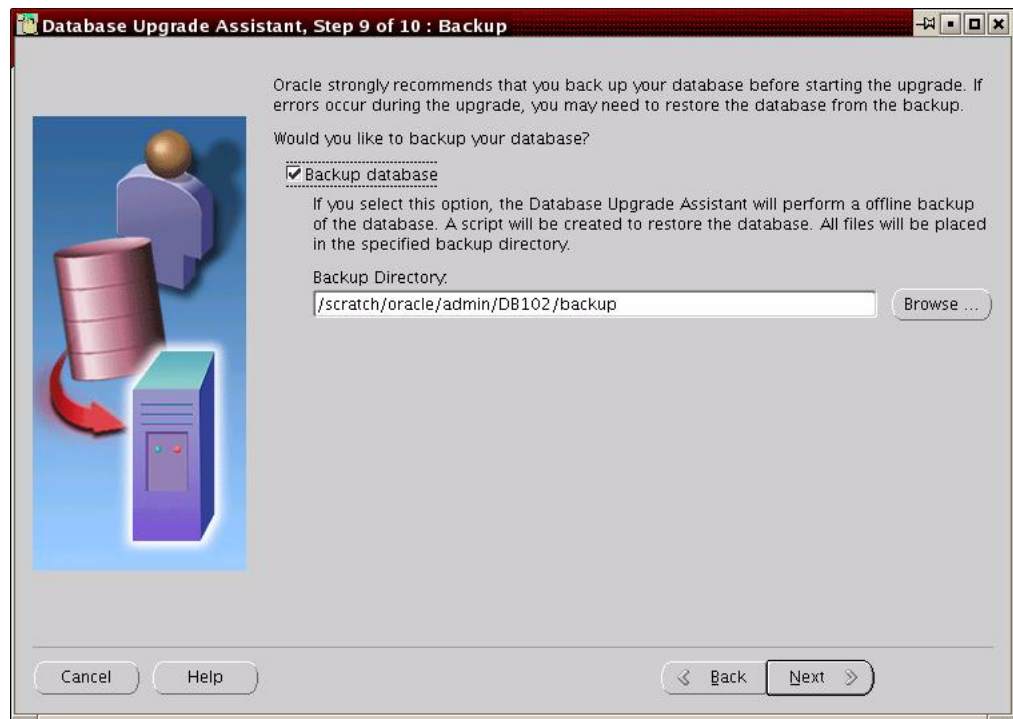
DBUA also provides a recommended degree of parallelism, which determines how many parallel processes are used to recompile your invalid PL/SQL modules. Specifically, DBUA sets the degree of parallelism to one less than the number of CPUs you have available. For example, if you have three CPUs available 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.

Select **Turn off Archiving and Flashback logging for the duration of upgrade** to reduce the time required to complete the upgrade.

If the database is in ARCHIVELOG or flashback logging mode, then DBUA gives you the choice of turning them off for the duration of the upgrade. If you choose this option, Oracle recommends that you perform an offline backup immediately after the upgrade.

Click **Next**.

The Backup screen appears.



13. 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:

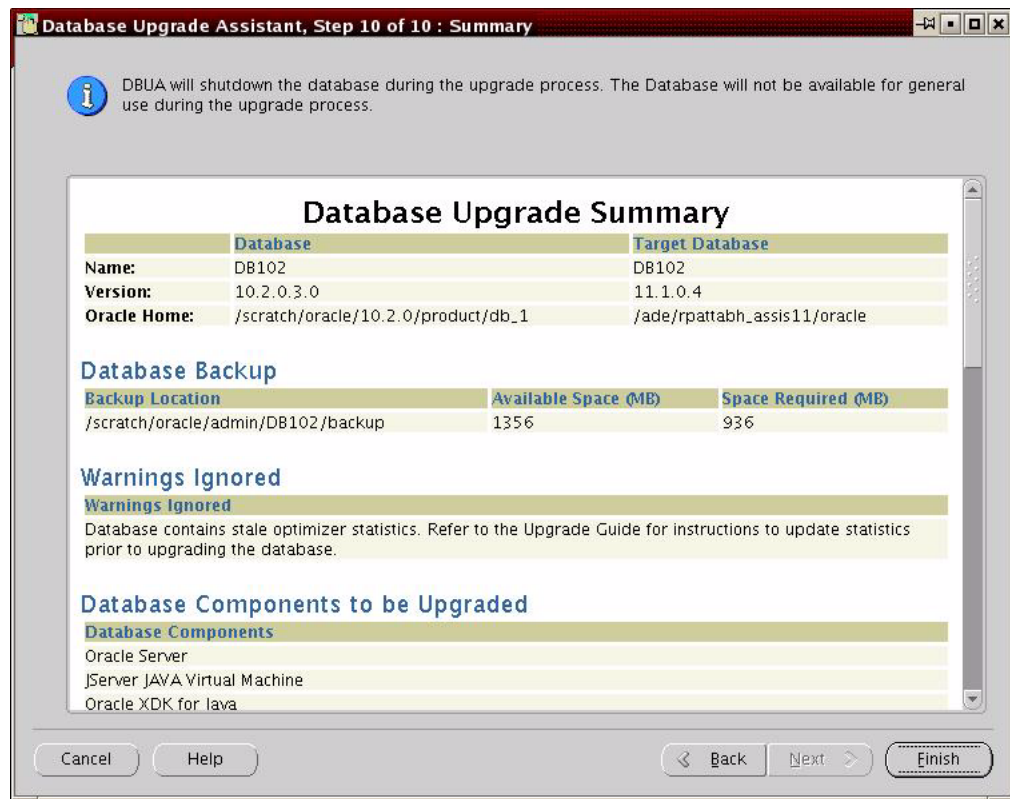
- On Windows operating systems, the file is called `db_name_restore.bat`.
- On Linux or UNIX platforms, the file is called `db_name_restore.sh`.

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**.

The Summary screen appears.



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

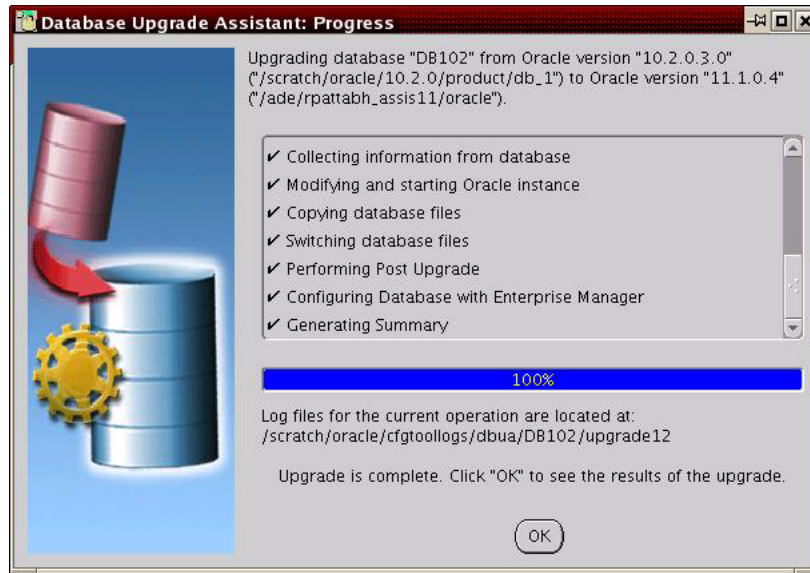
- Name, version, and Oracle home of the old and new databases
- Database backup location, available space, and space required
- Warnings ignored
- Database components to be upgraded
- Initialization parameters changes
- Database files location
- Listener registration

See Also: [Chapter 5, "Compatibility and Interoperability"](#) 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 appears, and DBUA begins the upgrade.



15. You might encounter error messages with Ignore and Abort choices. If other errors appear, then you must address them accordingly. If an error is severe and cannot be handled during the upgrade, then you have the following choices:

- Click **Ignore** to ignore the error and proceed with the upgrade.

This causes DBUA to display the step as skipped and move on to the next step in the upgrade, ignoring this and any dependent steps. The errors ignored are logged and shown later in the summary. After the upgrade is complete, you can fix the problem, restart DBUA, and complete the skipped steps.

- Click **Abort** to terminate the upgrade process.

If Ignore is not presented as a choice in the message, then this is your only option.

If a database backup was taken by DBUA, then it asks if you want to restore the database. After the database has been restored, you must correct the cause of the error and restart DBUA to perform the upgrade again.

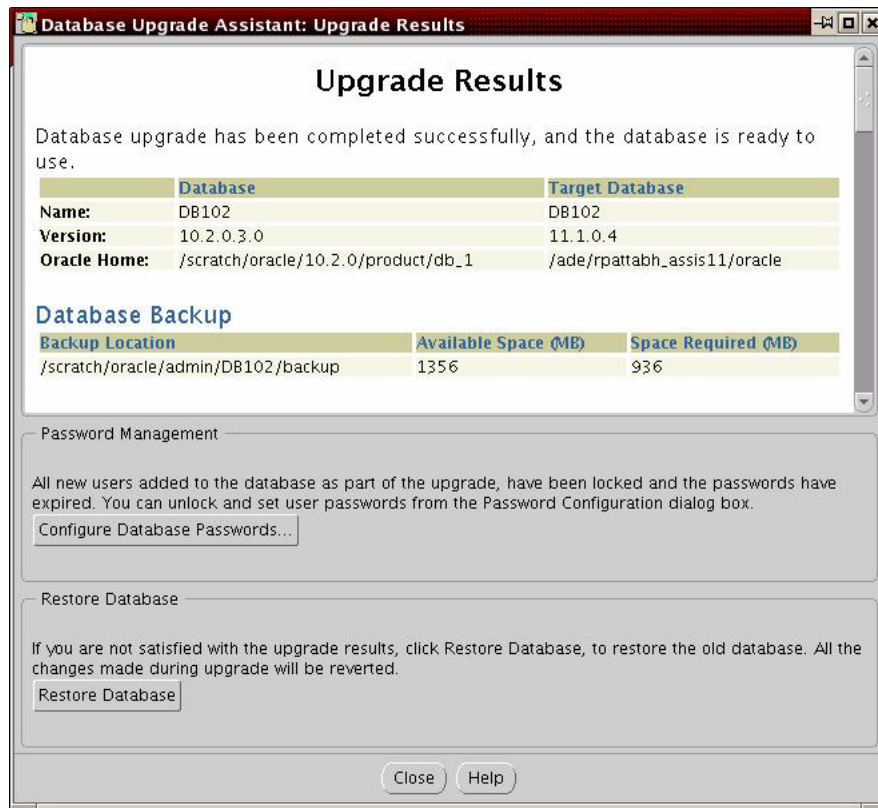
If you do not want to restore the database, then DBUA leaves the database in its present state so that you can proceed with a manual upgrade.

16. 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.

Click **OK**.

The Upgrade Results screen appears.



17. 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 11g Release 1 (11.1) 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/upgraden`, 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 3–2 describes the various options and their parameters that are supported by DBUA.

Table 3–2 DBUA Command Line options

Option	Description
-silent	Specifies that DBUA should operate in silent mode.
-sid <i>SID</i>	Specifies the system identifier (SID) of the database to upgrade
-oracleHome <i>home_name</i>	Specifies the Oracle Database home directory of the database to upgrade
-oracleBase <i>base_name</i>	Specifies the Oracle Database base directory of the database to upgrade
-diagnosticDest <i>diagnostic_destination</i>	Specifies the default location to store Oracle trace and diagnostic files. It replaces the initialization parameter settings for background dump destination, user dump destination, and core dump destination from earlier releases.
-sysDBAUserName <i>SYSDBA_user</i>	Specifies a user with SYSDBA privileges.
-sysDBAPassword <i>SYSDBA_pwd</i>	Specifies the password for <i>SYSDBA_user</i> .
-upgradeASM	This option enables the upgrade of an Automatic Storage Management (ASM) instance.
-autoextendFiles	This option autoextends database files during the upgrade. Data files revert to their original autoextend settings after the upgrade.
-newGlobalDbName <i>db_name</i>	Specifies a new global database name. This option applies only if you are moving data files or upgrading an Oracle XE database.
-newSid <i>new_SID</i>	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 <i>common_files</i>	Specifies a common location to store database files. This option applies only if you are moving data files or upgrading an Oracle XE database.

Table 3–2 (Cont.) DBUA Command Line options

Option	Description
<code>-omfLocation omf_area</code>	Specifies a database area for Oracle-Managed Files. This option applies only if you are moving data files or upgrading an Oracle XE database.
<code>-databaseMapFile map_file_name</code>	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.
<code>-newRecoveryArea recover_area</code>	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.
<code>-newRecoveryAreaSize recover_size</code>	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.
<code>-apexAdminPassword apex_pwd</code>	Specifies the password for the Application Express Administrator.
<code>-disableUpgradeScriptLogging</code>	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.
<code>-backupLocation directory</code>	Specifies a directory to back up your database before the upgrade starts
<code>-sysauxTablespace</code>	This option creates a new SYSAUX tablespace
<code>-datafileName name</code>	Specifies the complete path of the data file for creating the new SYSAUX tablespace
<code>-datafileSize size</code>	Specifies the size of the data file for creating the new SYSAUX tablespace {500M, 10G}
<code>-datafileSizeNext size</code>	Specifies the next extent for the data file
<code>-datafileSizeMax size</code>	Specifies the maximum size of the data file
<code>-postUpgradeScripts script [, script] ...</code>	Specifies a comma-delimited list of SQL scripts. Specify complete path names. The scripts are executed at the end of the upgrade.
<code>-initParam parameter=value [, parameter=value] ...</code>	Specifies a comma-delimited list of initialization parameter values of the form <i>name=value</i>
<code>-disableArchiveLogMode</code>	This option turns off archiving and flashback logging for the duration of the upgrade.
<code>-recompile_invalid_objects true false</code>	When you specify <code>true</code> for this option, DBUA recompiles all invalid PL/SQL modules immediately after the upgrade is performed.
<code>-degree_of_parallelism cpu_number</code>	Specifies the number of CPUs to be used for parallel recompilation.

Table 3–2 (Cont.) DBUA Command Line options

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

Table 3–2 (Cont.) DBUA Command Line options

Option	Description
<code>[-localRacSid SID]</code>	Specifies the local SID of the cluster database if the cluster database is not registered in the Oracle Cluster Registry
<code>-h -help</code>	Displays help for DBUA

Note: If the default Oracle Database 11g Release 1 (11.1) 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 3-7.

- [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 11g Release 1 (11.1) 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 11g Release 1 (11.1).

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 11g Release 1 (11.1) 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 `pwsid.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 11g Release 1 (11.1) by completing the following steps:
 - a. Remove obsolete initialization parameters and adjust deprecated initialization parameters. Certain parameters are obsolete in Oracle Database 11g Release 1

(11.1), while other parameters have become deprecated. Remove all obsolete parameters from any parameter file that starts an Oracle Database 11g Release 1 (11.1) instance. Obsolete parameters might cause errors in Oracle Database 11g Release 1 (11.1). Also, alter any parameter whose syntax has changed in the new release.

See Also: "[Compatibility Issues with Initialization Parameters](#)" on page A-4 for a list of initialization parameters that have changed in ways that raise compatibility issues between different releases of Oracle Database software

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: "[Initialization Parameter Changes](#)" on page A-1 for a list of initialization parameters that have been deprecated or have become obsolete

- b. Make sure the `COMPATIBLE` initialization parameter is properly set for Oracle Database 11g Release 1 (11.1). 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 value indicated 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 `initdb_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 OracleServiceSID 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 11g Release 1 (11.1) service at a command prompt using the ORADIM command of the new Oracle Database release:

```
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
<i>SID</i>	The same SID name as the SID of the database you are upgrading.
<i>PASSWORD</i>	The password for the new Oracle Database 11g Release 1 (11.1) database instance. This is the password for the user connected with SYSDBA privileges. The <code>-INTPWD</code> 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 11g Release 1 (11.1) security settings are in place, then passwords must be at least 8 characters, and passwords such as <code>welcome</code> and <code>oracle</code> are not allowed. See <i>Oracle Database Security Guide</i> for more information.
<i>USERS</i>	The maximum number of users who can be granted SYSDBA and SYSOPER privileges.
<i>ORACLE_HOME</i>	The Oracle Database 11g Release 1 (11.1) Oracle home directory. Ensure that you specify the full path name with the <code>-PFILE</code> option, including the drive letter of the Oracle home directory.

For example, if your *SID* is ORCL, 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 -SID ORCL -INTPWD TWxy5791 -MAXUSERS 10
      -STARTMODE AUTO -PFILE C:\ORACLE\PRODUCT\11.1.0\DB\DATABASE\INITORCL.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 your Oracle Database 11g Release 1 (11.1) Oracle home
 - c. The following environment variables point to the Oracle Database 11g Release 1 (11.1) 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 11g Release 1 (11.1) 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 commands:

```
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 11g Release 1 (11.1) 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 Oracle Database9i Release 2 (9.2).

Create a `SYSAUX` tablespace. In Oracle Database 11g Release 1 (11.1), 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 3-3](#) can be used to determine an optimal size for the `SYSAUX` tablespace.

Table 3-3 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
```

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

The upgrade script creates and alters certain data dictionary tables. It also upgrades or installs the following database components in the new Oracle Database 11g Release 1 (11.1) 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 `utlu111s.sql` to display the results of the upgrade:

```
SQL> @utlu111s.sql
```

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

```
Oracle Database 11.1 Post-Upgrade Status Tool          03-18-2007 22:48:55
.
Component                      Status          Version  HH:MM:SS
.
Oracle Server                    VALID          11.1.0.6.0  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
.
Oracle Ultra Search             VALID          11.1.0.6.0  00:08:24
.
Oracle Expression Filter        VALID          11.1.0.6.0  00:03:09
.
Oracle Rules Manager            VALID          11.1.0.6.0  00:00:38
.
Oracle Application Express       VALID          11.1.0.6.0  00:00:21
.
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 3-45.

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 `utlrlp.sql` (see step 15), if you run it in a new SQL*Plus session.

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

```
SQL> @utlrlp.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;
```

16. Exit SQL*Plus.

Your database is now upgraded to the Oracle Database 11g Release 1 (11.1). 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](#)
- [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.

- Flash Recovery Area

If you are using a Flash 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 Flash Recovery Area prior to starting the upgrade.

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. Rerun `catupgrd.sql`:

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

4. Rerun `utlu111s.sql`:

```
SQL> @utlu111s.sql
```

Cancel the Upgrade

If you completed the steps in "[Back Up the Database](#)" on page 3-37 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 with DBUA

It is not possible to upgrade an ASM instance with DBUA when the source and target ASM Oracle homes are owned by different users. Attempting to do so returns an error similar to the following:

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

Upgrading a single-instance ASM requires that Oracle Cluster Synchronization Services (CSS) first be configured to run from the new Oracle Database 11g Release 1 (11.1) home. If DBUA detects that this has not been done, it will instruct you to do so. See "[Reconfigure Oracle Cluster Synchronization Services](#)" on page 3-53 for more information.

Complete the following steps to upgrade an Automatic Storage Management (ASM) instance to Oracle Database 11g Release 1 (11.1):

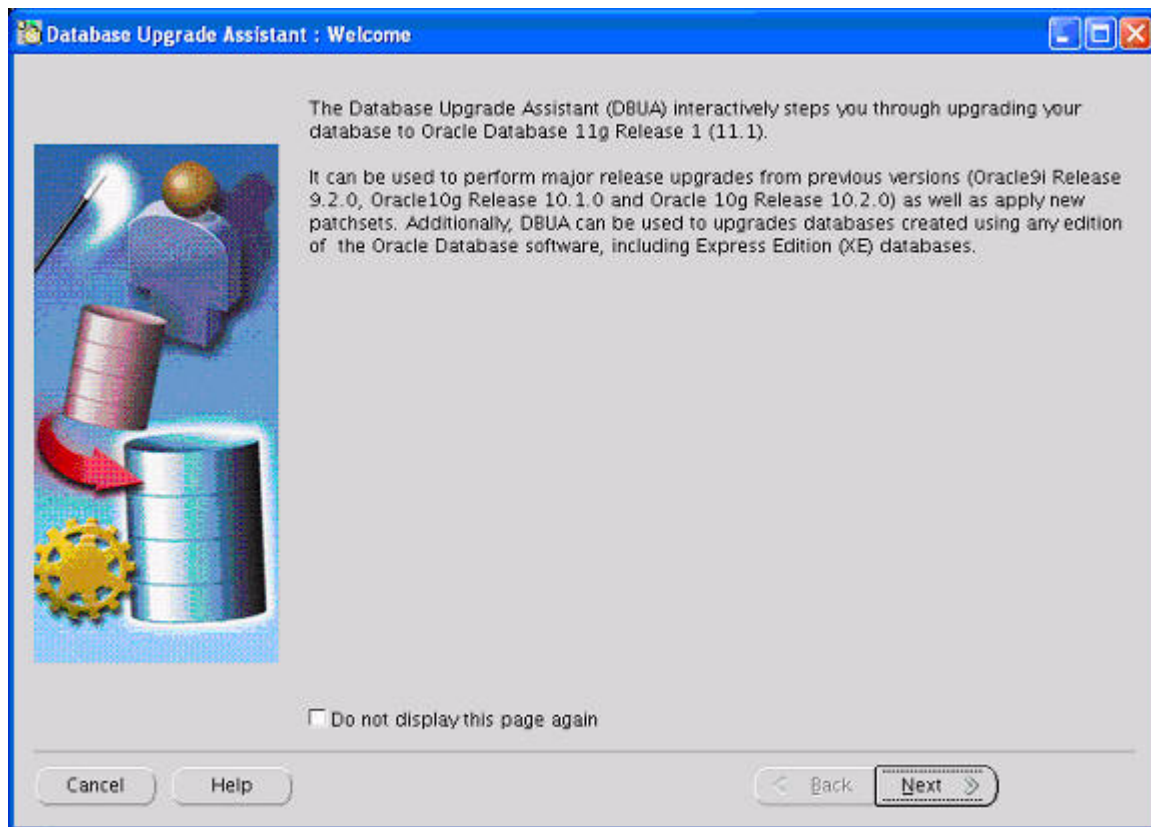
1. Do one of the following to start DBUA:
 - On Linux or UNIX platforms, enter the following command at a system prompt in the Oracle Database 11g Release 1 (11.1) environment:

```
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**.

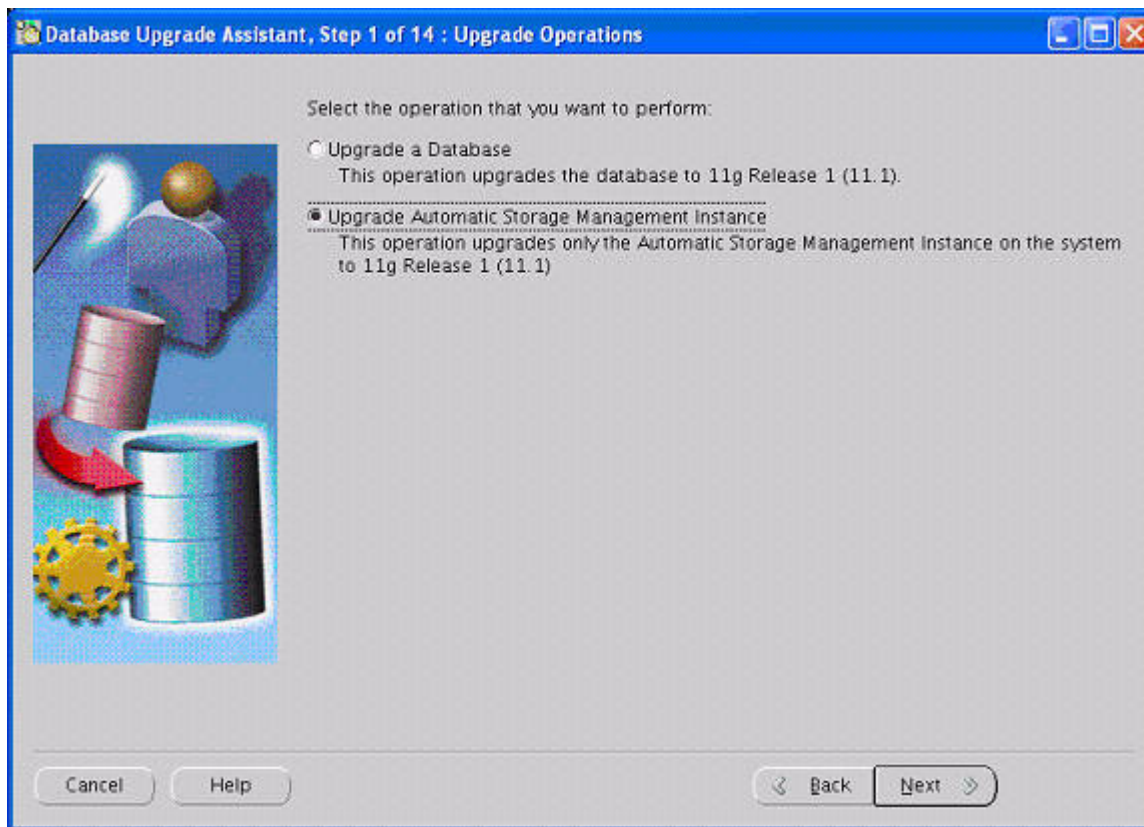
The DBUA Welcome screen appears.



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**.

If an Automatic Storage Management (ASM) instance is detected on the system, then the Upgrade Operations page appears with options to upgrade a database or an ASM instance. If no ASM instance is detected, then Databases screen appears.



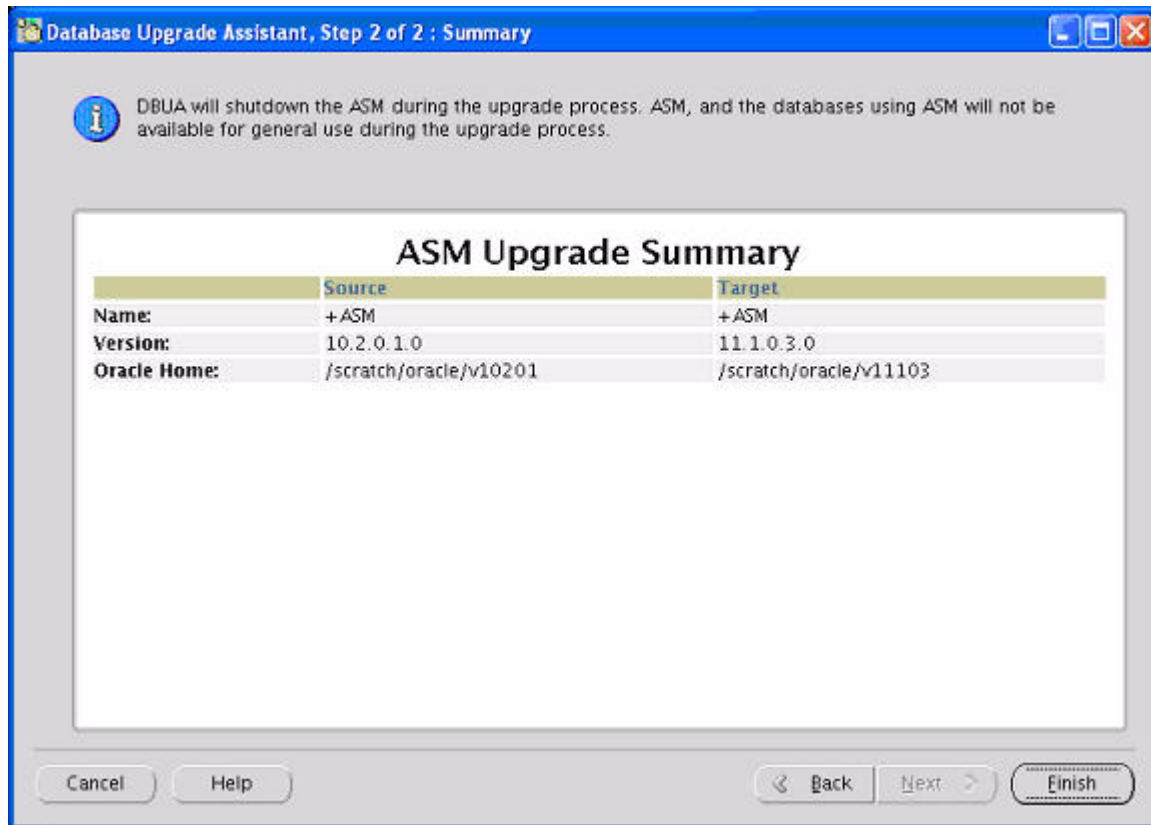
3. At the Upgrade Operations page, select **Upgrade Automatic Storage Management Instance**. This operation upgrades only the Automatic Storage Management instance on the system to 11g Release 1 (11.1).

The other option, **Upgrade a Database**, is discussed in "[Upgrade a Database Using Database Upgrade Assistant](#)" on page 3-16.

Oracle recommends that you upgrade the database and ASM in separate DBUA sessions, in separate Oracle homes.

Click **Next**.

The DBUA Summary screen appears.



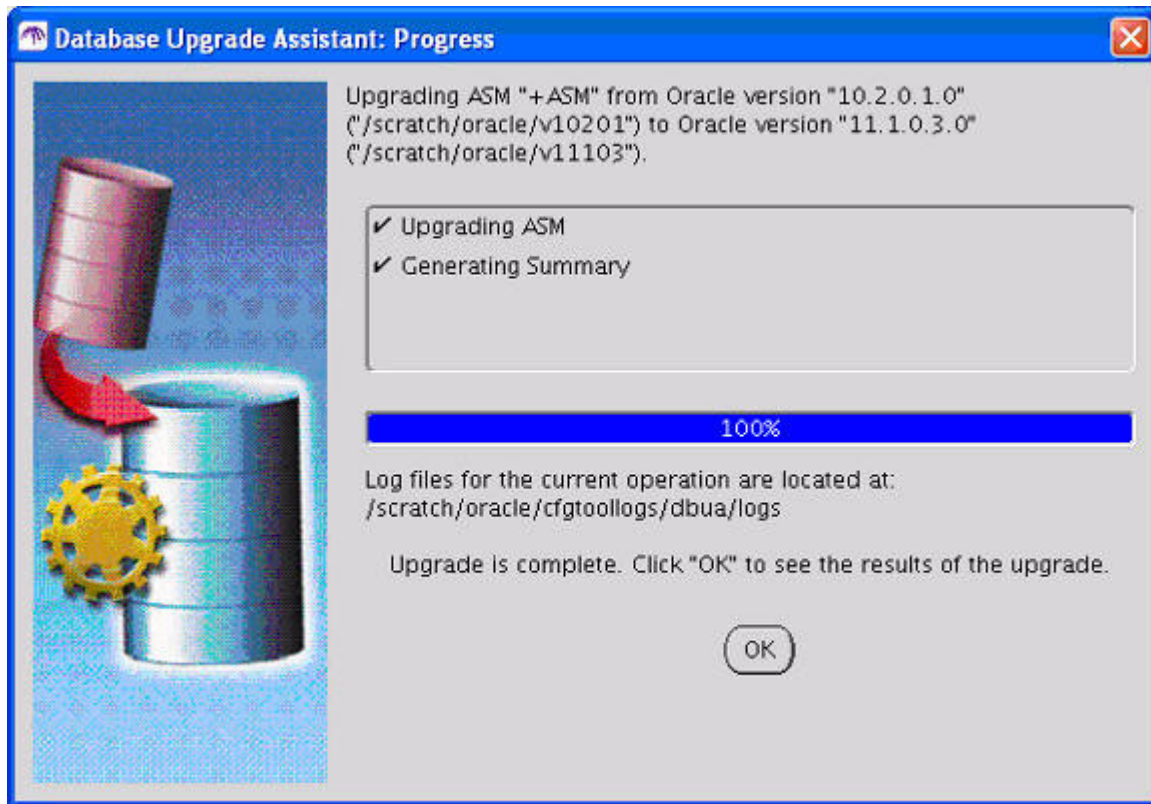
- The Summary screen shows the name, version, and Oracle home of the old and new ASM instances.

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

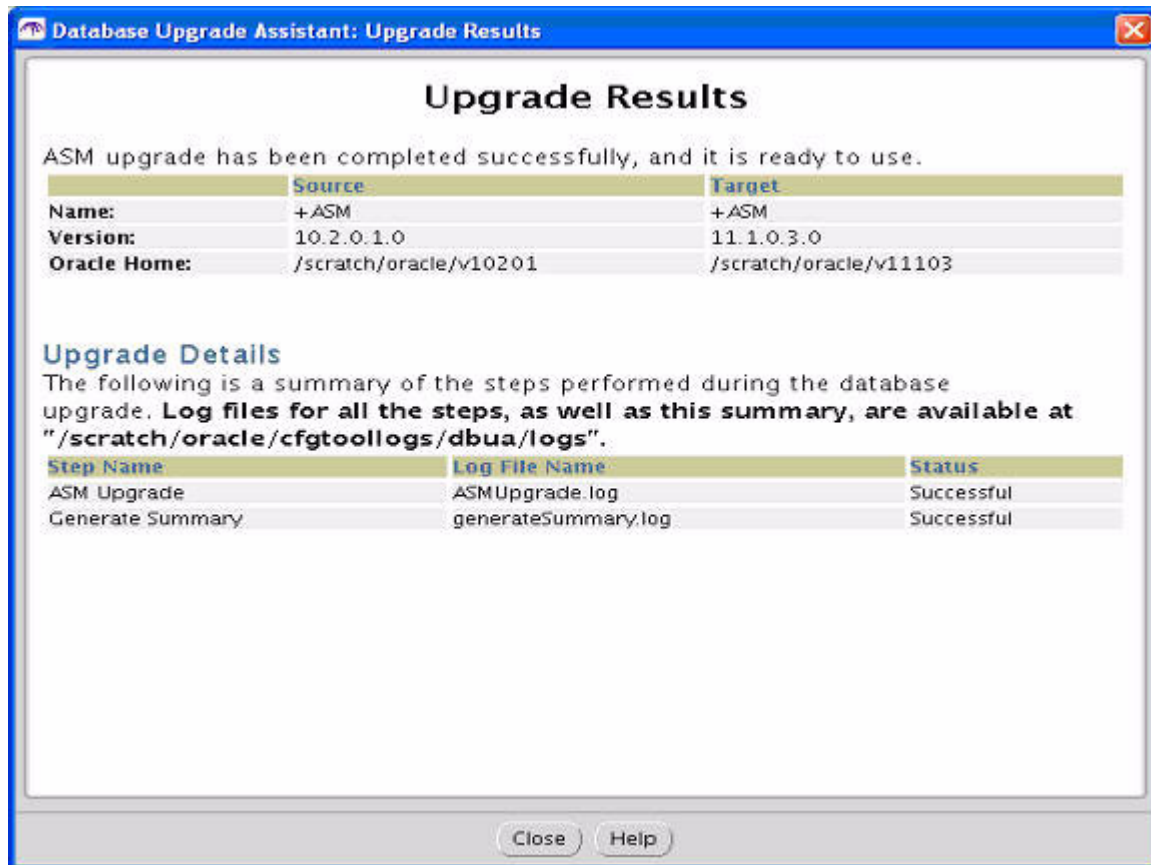
- Click **Back** if anything is incorrect until you reach the screen where you can correct it.
- Click **Finish** if everything is correct.

Note: DBUA shuts down the ASM instance during the upgrade process. Neither it nor the databases using it are available for general use during the upgrade.

The Progress screen appears, and DBUA begins the upgrade.



5. 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.
Click **OK**.
The Upgrade Results screen appears.



6. The Upgrade Results screen displays information about the previous and upgraded ASM instances. It also includes a summary of the steps performed during the ASM upgrade and the location of log files offering more details about individual upgrade steps.

Click **Close** to exit DBUA.

Upgrade an ASM Instance 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 3-7.

- [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 11g Release 1 (11.1) 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 11g Release 1 (11.1) 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 11g Release 1 (11.1).

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 11g Release 1 (11.1) 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 11g Release 1 (11.1) by completing the following steps:
 - a. Remove obsolete initialization parameters and adjust deprecated initialization parameters. Certain parameters are obsolete in Oracle Database 11g Release 1 (11.1), while other parameters have become deprecated. Remove all obsolete parameters from any parameter file that starts an Oracle Database 11g Release 1 (11.1) instance. Obsolete parameters might cause errors in Oracle Database 11g Release 1 (11.1). Also, alter any parameter whose syntax has changed in the new release.

See Also:

- ["Initialization Parameter Changes"](#) on page A-1 for a list of initialization parameters that have been deprecated or have become obsolete
- ["Compatibility Issues with Initialization Parameters"](#) on page A-4 for a list of initialization parameters that have changed in ways that raise compatibility issues between different releases of Oracle Database software

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, "Initialization Parameter and Data Dictionary Changes"](#) for a list of initialization parameters that have been deprecated or have become obsolete

- b. Adjust the values of the initialization parameters to at least the minimum value indicated 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 `OracleASMSERVICE` 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 11g Release 1 (11.1) service at a command prompt using the `ORADIM` command of the new Oracle Database release:

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

This syntax includes the following variables:

Variable	Description
<i>SID</i>	The same SID name as the SID of the ASM instance you are upgrading.
<i>PASSWORD</i>	The password for the new Oracle Database 11g Release 1 (11.1) ASM instance. This is the password for the user connected with <i>SYSDBA</i> privileges. The <i>-INTPWD</i> 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 11g Release 1 (11.1) security settings are in place, then passwords must be at least 8 characters, and passwords such as <i>welcome</i> and <i>oracle</i> are not allowed. See <i>Oracle Database Security Guide</i> for more information.
<i>USERS</i>	The maximum number of users who can be granted <i>SYSDBA</i> and <i>SYSOPER</i> privileges.
<i>ORACLE_HOME</i>	The Oracle home for the new Oracle Database 11g Release 1 (11.1) ASM instance. Ensure that you specify the full path name with the <i>-PFILE</i> 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 -INTPWD 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 11g Release 1 (11.1) ASM instance
 - c. The following environment variables point to the Oracle Database 11g Release 1 (11.1) 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 11g Release 1 (11.1) 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:
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`:

```
SQL> STARTUP
```

```
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 11g Release 1 (11.1). 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 or restarting `DBUA`.

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.

After Upgrading to the New Release

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

- [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 Upgrade](#)
- [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 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](#)
- [Deploy Database Vault Administrator \(DVA\)](#)

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 new Oracle Database 11g Release 1 (11.1) directories:

- ORACLE_HOME
- PATH

Also check that your `oratab` file and any client scripts that set the value of `ORACLE_HOME` point to the Oracle Database 11g Release 1 (11.1) home.

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 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 Oracle Database 9i Release 2 (9.2) or Oracle Database 10g 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 10g 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 11g Release 1 (11.1) products and are not immediately available after an upgrade to Oracle Database 11g Release 1 (11.1). Any Oracle Text features dependent on the supplied knowledge bases which were available before the upgrade do not function after the upgrade. To reenable 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 3.0, there is no additional configuration necessary after upgrading to Oracle Database 11g Release 1 (11.1).

If your database was not an Oracle Express Edition (XE) database, but contained a prior version of Application Express (HTML DB), then version 3.0 is automatically installed during the upgrade. You must complete a series of post-installation steps to configure Application Express Version 3.0 for use with Oracle Database 11g. These steps are described in the section on post-installation tasks in the *Oracle Database Application Express Installation Guide*.

If your database was an Oracle Express Edition (XE) database, it contained a prior version of Application Express tailored for the XE environment. Refer to the OTN document describing the differences between Oracle Database XE Application Express and Application Express 3.0 at:

http://www.oracle.com/technology/products/database/application_express/html/upgrade_apex_for_xe.html

The database administration features available with the XE version of Application Express are not available in version 3.0, 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 11g Release 1 (11.1) 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

If you use Oracle Database Vault, then you were instructed to disable it before upgrading your database. You must now enable Database Vault again.

See Also:

- ["Disable Oracle Database Vault"](#) on page 3-13
- *Oracle Database Vault Administrator's Guide* for instructions on enabling Oracle Database Vault

Deploy Database Vault Administrator (DVA)

You can manually deploy Database Vault Administrator (DVA) to the following Oracle Application Server Containers for J2EE (OC4J) home:

`$ORACLE_HOME/oc4j/j2ee/home`

Use the following steps to manually deploy the DVA application:

1. Edit the file, `$ORACLE_HOME/oc4j/j2ee/home/config/server.xml`. Enter the following line just before the last line that reads, `</application-server>`:

```
<application name="dva" path="$ORACLE_HOME/dv/jlib/dva_webapp.ear"
auto-start="true" />
```

For example:

```
<application name="dva" path="/u00/app/oracle/oracle/product/dv12/dv/jlib/dva_
webapp.ear" auto-start="true" />
```

2. Edit the file, `$ORACLE_HOME/oc4j/j2ee/home/config/http-web-site.xml`. Enter the following line just above the last line that reads, `</web-site>`:

```
<web-app application="dva" name="dva_webapp" root="/dva" />
```

3. Edit the file, `$ORACLE_HOME/oc4j/j2ee/home/config/global-web-application.xml`. Search for

```
<servlet-class>oracle.jsp.runtimev2.JspServlet</servlet-class>
>. Uncomment the following lines after this:
```

```
<init-param>
  <param-name>main_mode</param-name>
  <param-value>justrun</param-value>
</init-param>
```

4. Create the following directory:

```
mkdir -p $ORACLE_HOME/dv/jlib/sysman/config
```

5. Create the database connection configuration file, `emoms.properties`, in the configuration directory that you just created. Add the following lines to the file:

```
oracle.sysman.emSDK.svlt.ConsoleMode=standalone
oracle.sysman.eml.mntr.emdRepRAC=FALSE
oracle.sysman.eml.mntr.emdRepDBName=ORACLE_SID
oracle.system.eml.mntr.emdRepConnectDescriptor=TNS_connection_string
```

Note:

- `oracle.sysman.eml.mntr.emdRepRAC` should be set to `TRUE` for Oracle Real Application Clusters database.
- For `oracle.sysman.eml.mntr.emdRepConnectDescriptor`, you can use an alias from `$ORACLE_HOME/network/admin/tnsnames.ora`. Alternatively, you can use the following syntax:

```
oracle.sysman.eml.mntr.emdRepConnectDescriptor=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
(HOST=HOSTNAME) (PORT=PORT))) (CONNECT_DATA=(SERVICE_NAME=ORACLE_SID)))
```

6. Start OC4J. Before starting OC4J, ensure that the correct environment variables are set. For example:

```
ORACLE_SID=orcl
export ORACLE_SID
ORACLE_HOME=/u00/app/oracle/product/10.2/dv
export ORACLE_HOME
LD_LIBRARY_PATH=$ORACLE_HOME/bin:$ORACLE_HOME/lib:$ORACLE_HOME/jdbc/lib
export LD_LIBRARY_PATH
PATH=$ORACLE_HOME/bin:$ORACLE_HOME/jdk/bin:$PATH
export PATH
```

Note: `LD_LIBRARY_PATH` must be set to use the OCI-based JDBC libraries.

Start OC4J using the following syntax:

```
$ORACLE_HOME/jdk/bin/java -Djava.awt.headless=true -DEMDROOT=$ORACLE_
HOME/dv/jlib -jar $ORACLE_HOME/oc4j/j2ee/home/oc4j.jar -userThreads -config
$ORACLE_HOME/oc4j/j2ee/home/config/server.xml
```

7. You can now access the DVA application. The HTTP port defaults to 8888 for this environment. Use the following URL:

`http://hostname:8888/dva`

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 a 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

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

To 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.

Note: If the default Oracle Database 11g Release 1 (11.1) 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 with DBUA"](#) on page 3-48
- ["Upgrade an ASM Instance Manually"](#) on page 3-53
- ["Required Tasks After ASM Upgrades"](#) on page 4-14
- ["Recommended Tasks After ASM Upgrades"](#) on page 4-14
- ["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 Oracle Database 11g Release 1 (11.1). 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 6, "Upgrading Your Applications"](#) describes ways to enhance your applications so that you can take advantage of new Oracle Database 11g Release 1 (11.1) features. 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 Oracle Database 11g Release 1 (11.1) features, 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 11g 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 11g 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 11g 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 11g 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 11g Release 1 (11.1).
- 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 11g 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 11g 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 10g configuration to Oracle Database 11g 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 Oracle9i 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

Recommended Tasks After Upgrading a 10g Release 1 (10.1) Database

Performing the following tasks is recommended, but not required, after you have upgraded from Oracle Database 10g Release 1 (10.1) or Oracle Database 10g 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

Beginning with Oracle Database 10g 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 Oracle Database9i Release 2 (9.2) system as a source.

See the *Oracle Database Data Warehousing Guide* for complete information about how to upgrade an Oracle Database9i Release 2 (9.2) or Oracle Database 10g Release 1 (10.1) database to Oracle Database 11g Release 1 (11.1) 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 10g 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 10g Release 2 (10.2) or later releases, 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 10g Release 2. 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 Oracle Database 11g Release 1 (11.1), 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 11g Release 1 (11.1) 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 Oracle Database 11g Release 1 (11.1), 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 9i and desupported in Oracle 10g Release 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 11g:

- Use `srvconfig` from the Oracle Database 11g Release 1 (11.1) 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 preface in *Oracle Database Reference* for a list of the new initialization parameters in Oracle Database 11g Release 1 (11.1), and for information about each parameter
- [Appendix A, "Initialization Parameter and Data Dictionary Changes"](#) for lists of obsolete and deprecated initialization parameters in Oracle Database 11g Release 1 (11.1)

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

See Also: ["Setting the COMPATIBLE Initialization Parameter"](#) on page 5-3 for information

Configure Enterprise Manager

If you are not yet using Oracle Enterprise Manager to manage your database, 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 new Oracle Database 11g Oracle Home. When prompted, provide the Oracle home from which the configuration is being upgraded.

See Also: *Oracle Enterprise Manager Advanced Configuration*

Set CLUSTER_DATABASE Initialization Parameter

In ["Prepare the New Oracle Home"](#) on page 3-38, 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

If your operating system is Linux or UNIX, then make sure that the following environment variables point to the new Oracle Database 11g Release 1 (11.1) directories:

- `ORACLE_HOME`
- `PATH`

Also check that your `oratab` file and any client scripts that set the value of `ORACLE_HOME` point to the Oracle Database 11g Release 1 (11.1) home.

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.

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 4-7
- ["Develop New Administrative Procedures as Needed"](#) on page 4-7

Reset ASM Passwords to Enforce Case-Sensitivity

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

To 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 11g Release 1 (11.1) 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

Beginning with Oracle Database 11g 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 `asm_dg2`:

```
ALTER DISKGROUP asm_dg2 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.

Both the ASM clients and ASM require Oracle Database 11g Release 1 (11.1) or higher to use the preferred read feature. The `compatible.asm` and `compatible.rdbms` attributes are the disk group attributes that must be advanced to release 11.1 to use this new feature.

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 Upgrade

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

- [Single-Instance ASM Upgrade](#)
- [Cluster ASM Upgrade](#)
- [Database Upgrade After ASM Upgrade](#)

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 11.1 using OUI and DBUA. The new ASM release 11.1 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 11.1 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. Copy `spfile+ASM.ora` and `init+ASM.ora` to the new OH3/`db`s. Update these two files so that any reference to OH2 is changed to OH3.
9. Copy `orapw+ASM` file from OH2 to OH3.
10. Run EMCP to change the ASM instance path and `connect-string` role.
11. Make sure the disks are owned by `asmuser` and by `OSDBA` for ASM. They should also have the `0660` permission set.
12. As `asmuser`, start the listener.
13. As `asmuser`, start ASM (connect as `SYSASM`).
14. 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 11.1 using OUI and DBUA. The new ASM release 11.1 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 11.1 in a third Oracle Home (OH3). This should be a software-only installation.
4. Execute the following:
 - a. As `orauser`, from 10.2 Oracle home, run:

```
srvctl modify instance -d <db-name> -i <db-inst-sid> -n <node-name> -r
```

b. From CRS home, run:

```
srvctl remove listener -n <node-name>
srvctl add listener -n <node-name> -o <OH3>
srvctl remove asm -n <node-name>
srvctl add asm -n <node-name> -i <asm-sid> -o <OH3>
```

c. As `asmuser`, from 10.2 Oracle home, run:

```
srvctl modify instance -d <db-name> -i <db-inst-sid> -s <asm-sid>
```

Repeat these steps for each node.

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. Copy `spfile+ASM.ora` and `init+ASM.ora` to the new OH3/dbs. Update these two files so that any reference to OH2 is changed to OH3.
8. Copy `orapw+ASM` file from OH2 to OH3.
9. Run EMCP to change the ASM instance path and `connect-string` role.
10. Make sure the disks are owned by `asmuser` and by OSDBA for ASM. They should also have the 0660 permission set.
11. Start ASM and Listener resources from CRS home.
12. As `asmuser`, start ASM (connect as SYSASM).
13. Run the command, `GRANT sysasm TO sys`.

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](#)

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 rather than using DBUA, 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 preface in *Oracle Database Reference* for a list of the new initialization parameters in Oracle Database 11g Release 1 (11.1), and for information about each parameter
- [Appendix A, "Initialization Parameter and Data Dictionary Changes"](#) for lists of obsolete and deprecated initialization parameters in Oracle Database 11g Release 1 (11.1)

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

Compatibility and Interoperability

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

This chapter covers the following topics:

- [What Is Compatibility?](#)
- [What Is Interoperability?](#)
- [Deprecated Features in 11g Release 1 \(11.1\)](#)
- [Compatibility and Interoperability Issues Introduced in 11g Release 1 \(11.1\)](#)
- [Compatibility and Interoperability Issues Introduced in 10g Release 2 \(10.2\)](#)
- [Compatibility and Interoperability Issues Introduced in 10g Release 1 \(10.1\)](#)

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 11.0.0 for Oracle Database 11g Release 1 (11.1). You cannot use new Oracle Database 11g Release 1 (11.1) features that would make your upgraded database incompatible unless the COMPATIBLE initialization parameter is set to this value.

[Table 5–1](#) lists the default, minimum, and maximum values of the COMPATIBLE initialization parameter in Oracle Database 11g Release 1 (11.1) and in each release supported for upgrading to Oracle Database 11g Release 1 (11.1).

Table 5–1 The COMPATIBLE Initialization Parameter

Oracle Database Release	Default Value	Minimum Value	Maximum Value
Oracle Database9i Release 2 (9.2)	8.1.0	8.1.0.0.0	9.2.0.n.n
Oracle Database 10g Release 1 (10.1)	10.0.0	9.2.0.0.0	10.1.0.n.n
Oracle Database 10g Release 2 (10.2)	10.2.0	9.2.0.0.0	10.2.0.n.n
Oracle Database 11g Release 1 (11.1)	11.0.0	10.0.0.0.0	11.0.0.n.n

Downgrading and Compatibility

Before upgrading to Oracle Database 11g Release 1 (11.1), the `COMPATIBLE` initialization parameter must be set to at least `10.0.0`, which is the lowest possible setting for Oracle Database 11g Release 1 (11.1). Only a subset of Oracle Database 11g Release 1 (11.1) features are available while the `COMPATIBLE` initialization parameter is set to this value.

After upgrading to Oracle Database 11g Release 1 (11.1), 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 10g Release 1 (10.1)
- Set to `10.2.0` or earlier if you upgraded from Oracle Database 10g Release 2 (10.2)

See Also: [Chapter 7, "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 11g Release 1 (11.1) database with the `COMPATIBLE` initialization parameter set to `10.1.0`, then it generates database structures on disk that are compatible with Oracle Database 10g 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.

Checking the Current Value of the COMPATIBLE Initialization Parameter

To check the current value of the COMPATIBLE initialization parameter, enter the following SQL statement:

```
SQL> SELECT name, value, description 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 Oracle Database 11g Release 1 (11.1). However, after you do this, the database cannot subsequently be downgraded.

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`.

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.

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.

Deprecated Features in 11g Release 1 (11.1)

This section lists Oracle Database features deprecated in Oracle Database 11g Release 1 (11.1). They are supported in this release for backward compatibility. But 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*

Compatibility and Interoperability Issues Introduced in 11g Release 1 (11.1)

The following sections describe compatibility and interoperability issues introduced in Oracle Database 11g Release 1 (11.1) and actions you can take to prevent problems resulting from these issues.

- [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](#)
- [LOG_ARCHIVE_DEST_n Parameter](#)
- [SHARED_POOL_SIZE Parameter](#)
- [JOB_QUEUE_PROCESSES Parameter](#)

Automatic Maintenance Tasks Management

Automatic Maintenance Tasks Management, a new database component in Oracle Database 11g 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 10g, then the corresponding Automatic Maintenance Tasks Management feature is disabled after upgrading to Oracle Database 11g 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 11g 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 11g 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 a Database that Uses Automatic Storage Management \(ASM\)](#)" on page 3-2.

See Also: *Oracle Database Storage Administrator's Guide* for more information about accessing ASM instances

ASM Disk Group Compatibility

Beginning with Oracle Database 11g 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 10g Release 1 (10.1), Oracle Database 10g Release 2 (10.2), and Oracle Database 11g 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 11g 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 11g 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

Beginning 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 11g Release 1 (11.1), taking the `SYSAUX` tablespace offline can have potential SQL performance consequences.

Backward Compatibility

In Oracle Database 11g 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 11g 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 11g Release 1 (11.1)

When the database is upgraded to Oracle Database 11g 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 11g 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 11g Release 1 (11.1).

PL/SQL Native Compilation

Starting in Oracle Database 11g, 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 10g initialization parameters `PLSQL_Native_Library_Dir` and `PLSQL_Native_Library_Subdir_Count` have no significance in Oracle Database 11g. 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 11g 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 11g 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` \leq 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` \leq 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 11g Release 1 (11.1), a `<deny>` entry always trumped any `<allow>` entry in a given ACL. Beginning with Oracle Database 11g 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 11g 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 10g 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 reexecuting 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. Beginning with Oracle Database 11g 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.

LOG_ARCHIVE_DEST_n Parameter

Beginning with Oracle Database 11g 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 10g 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 10g 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 11g Release 1 (11.1):

```
SQL> SELECT SUM(BYTES) FROM v$sgastat WHERE pool = 'shared pool';
```

In Oracle Database 11g 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

Beginning with Oracle Database 11g Release 1 (11.1), `JOB_QUEUE_PROCESSES` has changed from a basic 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 0-1000. 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 of 1-1000, 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`.

Compatibility and Interoperability Issues Introduced in 10g Release 2 (10.2)

The following sections describe compatibility and interoperability issues introduced in Oracle Database 10g Release 2 (10.2). If you are upgrading to Oracle Database 11g Release 1 (11.1) from a release prior to Oracle Database 10g Release 2 (10.2), then see the following sections for information about actions you can take to prevent problems resulting from these issues:

- [SQL](#)
- [CONNECT Role](#)
- [Time Zone Files](#)
- [New Limit for FAILED_LOGIN_ATTEMPTS](#)

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 10g 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 10g 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 to Oracle Database 10g 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 3-11.

Time Zone Files

The time zone files that are supplied with Oracle Database 10g Release 2 (10.2) have been updated from version 1 to version 2 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 3-12.

New Limit for `FAILED_LOGIN_ATTEMPTS`

As of Oracle Database 10g Release 2 (10.2), the limit for `FAILED_LOGIN_ATTEMPTS` for the `DEFAULT` profile is 10. Prior to Oracle Database 10g Release 2 (10.2), the default was `UNLIMITED`.

Compatibility and Interoperability Issues Introduced in 10g Release 1 (10.1)

The following sections describe compatibility and interoperability issues introduced in Oracle Database 10g Release 1 (10.1). If you are upgrading to Oracle Database 11g Release 1 (11.1) from a release prior to Oracle Database 10g Release 1 (10.1), then see the following sections for information about actions you can take to prevent problems resulting from these issues:

- [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](#)
- [SHARED_SERVERS Parameter](#)

SQL Optimizer

This section describes compatibility and interoperability issues relating to the SQL Optimizer in Oracle Database 10g Release 1 (10.1).

Rule-Based Optimizer Desupported

Starting with Oracle Database 10g 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 10g 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 to Oracle Database 10g Release 1 (10.1), and for newly created Oracle Database 10g Release 1 (10.1) 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 10g 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 10g Release 1 (10.1) and later, 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 10g Release 1 (10.1), `CLOB <-> NCLOB` implicit conversion in SQL and PL/SQL is allowed.

Starting with Oracle Database 10g 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 10g Release 1 (10.1), VPD policies are attached to synonyms rather than the base objects.

Invalid Synonyms After an Upgrade

Starting with Oracle Database 10g 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 to Oracle Database 10g Release 1 (10.1) and for newly created Oracle Database 10g Release 1 (10.1) databases. This data is stored in the `SYS_AUX` 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 10g Release 1 (10.1) to avoid conflict with the AWR.

Transaction and Space

Starting with Oracle Database 10g 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 10g 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 10g 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 10g 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 10g Release 1 (10.1), the `LOG_PARALLELISM` initialization parameter has been deprecated. Log file parallelism is now automatically enabled.

Starting with Oracle Database 10g Release 1 (10.1), the default value for the `RECOVERY_PARALLELISM` initialization parameter now defaults to allow parallel recovery.

Starting with Oracle Database 10g 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 10g Release 1 (10.1), the default buffer size for the `ASYN` attribute of the `LOG_ARCHIVE_DEST_n` initialization parameter has increased from 2,048 blocks to 61,440 blocks.

Starting with Oracle Database 10g 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 10g 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 10g 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 10g 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 10g 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 10g Release 1 (10.1), 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 10g 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 10g 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 10g 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 10g 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 10g 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 10g 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 Oracle9i 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 10g 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 10g Release 1 (10.1). Note that this new default archival processing is relevant only when log transport services are defined to use archiver processes (`ARCn`), 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 10g 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 (`ARCn`), 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

In Oracle Database 10g Release 1 (10.1), 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 10g 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 obsoleted. 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/plsql/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 Oracle Database 9i 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 release 10g 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:

- What's New preface 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

SHARED_SERVERS Parameter

When the initialization parameter SHARED_SERVERS is dynamically changed to 0, no new clients can connect in shared mode, but existing shared server connections can continue to operate. Prior to Oracle Database 10g Release 1 (10.1), existing shared server connections would hang in this situation.

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 Oracle Database 11g Release 1 (11.1). Existing applications running against a new Oracle Database 11g Release 1 (11.1) 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 Oracle Database 11g Release 1 (11.1). 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 Oracle Database 11g Release 1 (11.1)

Compatibility Issues for Applications

There might be compatibility issues between different releases of the 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, "Initialization Parameter and Data Dictionary 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 10g, and Oracle Database 11g Release 1 (11.1) 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 11g Release 1 (11.1) 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 10g Release 1 (10.1) to Oracle Database 10g Release 2 (10.2) is a database maintenance release upgrade.

Note: Starting with Oracle Database9*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 Administrator'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 type 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 6-2.

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 6-4.

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 Oracle9i Release 2 (9.2.0), then it is recommended that your server software be Oracle9i Release 2 (9.2.0) 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 11g Release 1 (11.1), then Oracle recommends upgrading the client software to Oracle Database 11g Release 1 (11.1) 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.

Dynamically-Linked Applications Must Be Relinked When Upgrading to a Major Release Dynamically-linked OCI applications from 10g 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 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), 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.

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 Oracle Database 11g Release 1 (11.1).
- **Option 2:** Precompile or compile and then relink the application using Oracle Database 11g Release 1 (11.1). Application code must be changed if any APIs are deprecated or changed.
- **Option 3:** Change the application code to use new Oracle Database 11g Release 1 (11.1) 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 11g Release 1 (11.1) 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 Oracle Database 11g Release 1 (11.1). In addition, your application cannot leverage some of the possible performance benefits of Oracle Database 11g Release 1 (11.1).

Option 2: Precompile or Compile the Application Using the New Software

When upgrading from a maintenance release to Oracle Database 11g Release 1 (11.1) 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 11g 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 11g Release 1 (11.1) Features

You can make code changes to your application to take advantage of new Oracle Database 11g Release 1 (11.1) 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 Oracle Database 11g Release 1 (11.1). Further, you can add new features to your application that are available only with Oracle Database 11g Release 1 (11.1).

Become familiar with the new Oracle Database 11g Release 1 (11.1) features 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 Oracle Database 11g Release 1 (11.1) features:

1. If you want to take advantage of new Oracle Database 11g Release 1 (11.1) features, 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 Oracle Database 11g Release 1 (11.1) runtime library, `SQLLIB`, which is included with the precompiler.

Changing OCI Applications Complete the following steps to change your OCI application to use Oracle Database 11g Release 1 (11.1) features:

1. Incorporate the new Oracle Database 11g Release 1 (11.1) OCI calls into the existing application.
2. Compile the application.
3. Relink the application with the Oracle Database 11g Release 1 (11.1) runtime library.

Upgrading SQL*Plus Scripts and PL/SQL

To use new Oracle Database 11g Release 1 (11.1) features and functions, change existing SQL scripts to use the new Oracle Database 11g Release 1 (11.1) syntax. Existing SQL scripts run unchanged on Oracle Database 11g Release 1 (11.1), and require no modification, if they do not use new Oracle Database 11g Release 1 (11.1) features and functions.

Note that improved error checking in Oracle Database 11g Release 1 (11.1) 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 10g Release 1 (10.1) use INTEGER arithmetic (approximately 9 significant digits) for some expressions whereas Oracle9i Release 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 release 10g 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:

- What's New preface 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 Oracle9i, Oracle Database 10g, and Oracle Database 11g Release 1 (11.1). However, review the new features described in *Oracle Database New Features Guide* to determine whether any of the new Oracle Database 11g Release 1 (11.1) features would be beneficial to your applications or might otherwise affect them. Information about the ways in which the Oracle Database 11g Release 1 (11.1) features 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.

Downgrading a Database

This chapter guides you through the process of downgrading a database to the previous Oracle Database release. 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 version from which the database was upgraded. Major release downgrades are supported back to 10.2 and 10.1. Patchset downgrades are supported back to all prior 11.1 patch releases.

Note: You cannot downgrade a database that was upgraded from Oracle Database Express Edition.

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 10g Release 1 (10.1) to Oracle Database 11g Release 1 (11.1), then you could not subsequently downgrade to Oracle Database 10g Release 2 (10.2); you could only downgrade to Oracle Database 10g Release 1 (10.1).

If the release number of your Oracle 10g Release 1 (10.1) database is lower than 10.1.0.5, then you should install the latest patch for Oracle Database 10g Release 1 (10.1) prior to downgrading. Similarly, if the release number of your Oracle 10g Release 2 (10.2) database is lower than 10.2.0.3, then you should install the latest patch release for 10.2 prior to downgrading. Note that you can *upgrade* from any patch release of 10.1 or 10.2, but you should install the latest patch release in the Oracle home you intend to use after downgrading.

If you have Database Vault installed with your Oracle 11g Release 1 database, you can downgrade to release 10.2.0.4 only. Downgrade to 10.2.0.3 is not supported with Database Vault.

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. But 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 3-13
- ["Restoring Oracle Enterprise Manager"](#) on page 7-8

In any event, if you have Enterprise Manager configured in your database, then you must drop the `SYSMAN` schema before downgrading the database. Run the following command to drop the `SYSMAN` schema:

```
DROP USER sysman CASCADE
```

Check for Incompatibilities

Check the compatibility level of your database to see if your database might have incompatibilities that prevent you from downgrading. If the compatibility level of your Oracle Database 11g Release 1 (11.1) database is 11.0.0 or higher, then you are not able to downgrade. Your compatibility level is determined by the setting of the `COMPATIBLE` initialization parameter. Check your `COMPATIBLE` initialization parameter setting by issuing the following SQL statement:

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

If you are downgrading to Oracle Database 10g 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 10g Release 1 (10.1), then the `COMPATIBLE` initialization parameter must be set to 10.1.0.

See Also: ["Downgrading and Compatibility"](#) on page 5-2 and ["Compatibility and Interoperability Issues Introduced in 11g Release 1 \(11.1\)"](#) beginning on page 5-4

Perform a Full Backup

Perform a full backup of your Oracle Database 11g Release 1 (11.1) 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 11g Release 1 (11.1) database to a major release or a relevant patchset upgrade:

1. If you have enabled Oracle Database Vault on your database, then you must disable it before downgrading the database.

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

2. Log in to the system as the owner of the Oracle Database 11g Release 1 (11.1) Oracle home directory.

Note: This step is required only if Enterprise Manager Database Control is already configured for the database.

Stop the Database Control, as follows:

- a. Set the `ORACLE_SID` environment variable to the `databaseSid`
- 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 this step should be performed for all the instances.

3. If you are downgrading an Oracle RAC database to 10g 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
```

4. 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` parameter to `false`. After the downgrade, you must set this parameter back to `true`.

5. Start SQL*Plus.
6. Connect to the database instance as a user with `SYSDBA` privileges.
7. 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.

8. Drop the `SYSMAN` schema:

```
DROP USER sysman CASCADE;
```

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

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

10. 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 11g Release 1 (11.1).
- You must run the script in the Oracle Database 11g Release 1 (11.1) environment.
- The script downgrades all Oracle Database components in the database to the major release or 11g 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.

11. 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 9; the suggested name was `downgrade.log`. Correct any problems you find in this file and rerun the downgrade script if necessary.

12. Shut down the instance:

```
SQL> SHUTDOWN IMMEDIATE
```

13. Exit SQL*Plus.

14. 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 11g Release 1 (11.1) installation documents for information about setting other important environment variables on your operating system

15. 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 11g Release 1 (11.1) 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
<i>SID</i>	Same <i>SID</i> name as the <i>SID</i> of the database being downgraded.
<i>PASSWORD</i>	Password for the database instance. This is the password for the user connected with <code>SYSDBA</code> privileges. The <code>-INTPWD</code> option is not required. If you do not specify it, then operating system authentication is used, and no password is required.
<i>USERS</i>	Maximum number of users who can be granted <code>SYSDBA</code> and <code>SYSOPER</code> privileges.
<i>ORACLE_HOME</i>	Oracle home directory of the database to which you are downgrading. Ensure that you specify the full path name with the <code>-PFILE</code> option, including drive letter of the Oracle home directory.

For example, if you are downgrading to Oracle Database 10g 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.

16. 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`.
-
-

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

18. Start SQL*Plus.

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

20. Start up the instance:

```
SQL> STARTUP UPGRADE
```

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

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

22. Run `catrelod.sql`:

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

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

23. If you are downgrading to release 10.1.0.5 and you have XDB in your database, run the following script after running `catrelod.sql`:

```
@dbmsxdbt.sql
```

24. 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 21; the suggested name was `reload.log`. Correct any problems you find in this file and rerun the appropriate script if necessary.

25. 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.

26. Perform this step if the database is configured for Oracle Label Security and you are downgrading from Oracle Database 10g Release 1 (10.1) or Oracle Database 10g Release 2 (10.2). Copy the `olstrig.sql` script from the Oracle Database 11g Release 1 (11.1) Oracle home to the Oracle home of the version to which the database is to be downgraded. Run `olstrig.sql` to re-create DML triggers on

tables with Oracle Label Security policies. (See *Oracle Database Enterprise User Security Administrator's Guide* for more information.)

```
SQL> @olstrig.sql
```

27. 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.

28. 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](#)

Downgrading Oracle Clusterware Configuration

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

- Use `srvconfig` from the Oracle Database 11g Release 1 (11.1) 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: Downgrading the database back to 10g needs to be done 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 3-13

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 11g Release 1 (11.1) 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
```

After completing the `emca -restore` procedure you are ready to restore the Oracle Enterprise Manager Database files and data with the `emdwrgrd` utility. You must run the `emdwrgrd` utility from the Oracle Database 11g Release 1 (11.1) 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 `emdwrgrd.bat` for `emdwrgrd`.

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.
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 11g Release 1 (11.1) 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:

```
emdwrgrd -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:

```
emdwrgrd -restore -tempTablespace TEMP -cluster -sid SID10g -path save_
directory
```

If 10g Oracle home is on a shared device, add `-shared` to the previous command line.

6. Enter the SYS and SYSMAN passwords when prompted by `emdwrgrd`.
7. On a single-instance database, the `emdwrgrd` 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/racdwwgrd_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/racdwwgrd_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 enable Database Vault again.

See Also: *Oracle Database Vault Administrator's Guide* for instructions on enabling Oracle Database Vault

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 10g Release 1 (10.1), are high performance replacements for the original Export and Import utilities. When upgrading from Oracle Database 10g 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 10g 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 10g Release 1 (10.1), then you must use the original Export/Import utilities.
- If you must downgrade backward to a release prior to Oracle Database 10g 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 11g Release 1 (11.1) to Oracle Database 10g 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*, and Oracle Database 10g original Import utility.

Table 8–1 shows which releases to use when exporting data from releases earlier than Oracle Database 11g Release 1 (11.1) and then importing that data into Oracle Database 11g Release 1 (11.1).

Table 8–1 Exporting Data From Releases Earlier Than 11.1 and Importing Into Release 11.1

Export From	Import To	Export Version to Use	Import Version to Use
Release 10.2	Release 11.1	Data Pump Export Release 10.2	Data Pump Import Release 11.1
Release 10.1	Release 11.1	Data Pump Export Release 10.1	Data Pump Import Release 11.1
Release 9.2	Release 11.1	Original Export Release 9.2	Original Import Release 11.1
Release 8.1.7	Release 11.1	Original Export Release 8.1.7	Original Import Release 11.1
Release 8.0.6	Release 11.1	Original Export Release 8.0.6	Original Import Release 11.1
Release 7.3.4	Release 11.1	Original Export Release 7.3.4	Original Import Release 11.1

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 11g Release 1 (11.1), then Oracle Database 10g 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 Oracle9i Export dump file cannot be imported by an Oracle8i Import utility, and an Oracle Database 10g export dump file cannot be imported by an Oracle9i Import utility. The following tables provide specific examples.

See Also: *Oracle Database Utilities* for more information on using different releases and versions of Export

[Table 8–2](#) shows which releases to use when exporting data from Oracle Database 11g Release 1 (11.1) and then importing that data into earlier releases. Major release downgrades are supported to 10.2 and 10.1.

Table 8–2 Exporting Data From Release 11.1 and Importing Into Earlier Releases

Export From	Import To	Export Version to Use	Import Version to Use
Release 11.1	Release 10.2	Data Pump Export Release 11.1 with <code>VERSION=10.2</code>	Data Pump Import Release 10.2
Release 10.2	Release 10.1	Data Pump Export Release 10.2 with <code>VERSION=10.1</code>	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"](#).

Initialization Parameter and Data Dictionary Changes

This appendix lists changes to initialization parameters and the data dictionary across different releases of Oracle Database. This appendix also discusses compatibility issues with certain initialization parameters.

This appendix covers the following topics:

- [Initialization Parameter Changes](#)
- [Compatibility Issues with Initialization Parameters](#)
- [Static Data Dictionary View Changes](#)
- [Dynamic Performance View Changes](#)

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.

Initialization Parameter Changes

The following sections list changes to initialization parameters across different releases of Oracle Database:

- [Deprecated Initialization Parameters](#)
- [Obsolete Initialization Parameters](#)

See Also: The "What's New in Oracle Database Reference" section of *Oracle Database Reference* for a list of new initialization parameters in Oracle Database 11g Release 1 (11.1)

Deprecated Initialization Parameters

The following sections list initialization parameters that have been deprecated. 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:

- [Initialization Parameters Deprecated in Oracle Database 11g Release 1 \(11.1\)](#)
- [Initialization Parameters Deprecated in Oracle Database 10g Release 2 \(10.2\)](#)
- [Initialization Parameters Deprecated in Oracle Database 10g Release 1 \(10.1\)](#)
- [Initialization Parameters Deprecated in Oracle9i Release 2 \(9.2\)](#)

Initialization Parameters Deprecated in Oracle Database 11g Release 1 (11.1)

The following initialization parameters were deprecated in Oracle Database 11g Release 1 (11.1):

BACKGROUND_DUMP_DEST (replaced by DIAGNOSTIC_DEST)
COMMIT_WRITE
CORE_DUMP_DEST (replaced by DIAGNOSTIC_DEST)
INSTANCE_GROUPS
LOG_ARCHIVE_LOCAL_FIRST
PLSQL_DEBUG (replaced by PLSQL_OPTIMIZE_LEVEL)
PLSQL_V2_COMPATIBILITY
REMOTE_OS_AUTHENT
STANDBY_ARCHIVE_DEST
TRANSACTION_LAG attribute (of the CQ_NOTIFICATION\$_REG_INFO object)
USER_DUMP_DEST (replaced by DIAGNOSTIC_DEST)

Initialization Parameters Deprecated in Oracle Database 10g Release 2 (10.2)

The following initialization parameters were deprecated in Oracle Database 10g Release 2 (10.2):

LOGMNR_MAX_PERSISTENT_SESSIONS
MAX_COMMIT_PROPAGATION_DELAY
REMOTE_ARCHIVE_ENABLE
SERIAL_REUSE
SQL_TRACE

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):

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 Deprecated in Oracle9i Release 2 (9.2)

The following initialization parameters were deprecated in Oracle Database9i Release 2 (9.2):

DRS_START (replaced by DG_BROKER_START)

Obsolete Initialization Parameters

The following sections list initialization parameters that have been made obsolete:

- [Initialization Parameters Obsolete in Oracle Database 11g Release 1 \(11.1\)](#)
- [Initialization Parameters Obsolete in Oracle Database 10g Release 2 \(10.2\)](#)
- [Initialization Parameters Obsolete in Oracle Database 10g Release 1 \(10.1\)](#)
- [Initialization Parameters Obsolete in Oracle9i Release 2 \(9.2\)](#)

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.

Initialization Parameters Obsolete in Oracle Database 11g Release 1 (11.1)

DDL_WAIT_FOR_LOCKS
LOGMNR_MAX_PERSISTENT_SESSIONS
PLSQL_COMPILER_FLAGS

Initialization Parameters Obsolete in Oracle Database 10g Release 2 (10.2)

ENQUEUE_RESOURCES

Initialization Parameters Obsolete in Oracle Database 10g Release 1 (10.1)

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

Initialization Parameters Obsolete in Oracle9i Release 2 (9.2)

DISTRIBUTED_TRANSACTIONS
MAX_TRANSACTION_BRANCHES
PARALLEL_BROADCAST_ENABLED
STANDBY_PRESERVES_NAMES

Compatibility Issues with Initialization Parameters

The lists of deprecated and obsolete initialization parameters earlier in this appendix show changes to initialization parameters across different releases of Oracle Database. However, certain initialization parameter changes require special attention because they might raise compatibility issues for your database. These parameter changes are described in the following sections.

- [Compatibility Issues Introduced in Oracle Database 11g Release 1 \(11.1\)](#)
- [Compatibility Issues Introduced in Oracle Database 10g](#)

Compatibility Issues Introduced in Oracle Database 11g Release 1 (11.1)

The following section describes a compatibility issue introduced in Oracle Database 11g Release 1 (11.1).

New Default Value for UNDO_MANAGEMENT

Starting with Oracle Database 11g 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.

See Also: UNDO_MANAGEMENT in *Oracle Database Reference*

Compatibility Issues Introduced in Oracle Database 10g

The following sections describe compatibility issues introduced in Oracle Database 10g.

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 10g 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 Oracle Database9i 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 10g, the `OPTIMIZER_MAX_PERMUTATIONS` initialization parameter has been made obsolete. If you are upgrading from Oracle9i 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 11g 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 10g 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 10g 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 10g 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 10g 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 10g 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 10g Release 1 (10.1), these shared server parameters are dynamically modifiable.

New Default Value for DISPATCHERS Starting with Oracle Database 10g 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 10g 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 10g 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 10g 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 10g 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 10g 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 10g 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.

Static Data Dictionary View Changes

The following sections list changes to static data dictionary views across different releases of Oracle Database:

- [Deprecated Static Data Dictionary Views](#)
- [Obsolete Static Data Dictionary Views](#)
- [Static Data Dictionary Views with Dropped Columns](#)

See Also: The "What's New in Oracle Database Reference" section of *Oracle Database Reference* for a list of new static data dictionary views in Oracle Database 11g Release 1 (11.1)

Deprecated Static Data Dictionary Views

The following sections list static data dictionary views that have been deprecated:

- [Static Data Dictionary Views Deprecated in Oracle Database 11g Release 1 \(11.1\)](#)
- [Static Data Dictionary Views Deprecated in Oracle Database 10g Release 1 \(10.1\)](#)
- [Static Data Dictionary Views Deprecated in Oracle9i Release 2 \(9.2\)](#)

Static Data Dictionary Views Deprecated in Oracle Database 11g Release 1 (11.1)

No static data dictionary views were deprecated in Oracle Database 11g Release 1 (11.1).

Static Data Dictionary Views Deprecated in Oracle Database 10g Release 1 (10.1)

The following static data dictionary views were deprecated in Oracle Database 10g 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 Deprecated in Oracle9i Release 2 (9.2)

The following static data dictionary views were deprecated in Oracle Database9i Release 2 (9.2):

`ALL_RULESETS` (replaced by `ALL_RULE_SETS`)
`DBA_RULESETS` (replaced by `DBA_RULE_SETS`)
`USER_RULESETS` (replaced by `USER_RULE_SETS`)

Obsolete Static Data Dictionary Views

The following sections list static data dictionary views that have been made obsolete:

- [Static Data Dictionary Views Obsolete in Oracle Database 10g Release 1 \(10.1\)](#)

Static Data Dictionary Views Obsolete in Oracle Database 10g Release 1 (10.1)

The following static data dictionary views were made obsolete in Oracle Database 10g Release 1 (10.1):

ALL_ Views	DBA_ Views	USER_ Views
ALL_SOURCE_TAB_COLUMNS	DBA_SOURCE_TAB_COLUMNS	USER_SOURCE_TAB_COLUMNS

Static Data Dictionary Views with Dropped Columns

The following sections list static data dictionary views with dropped columns:

- [Static Data Dictionary Views with Dropped Columns in Oracle Database 11g Release 1 \(11.1\)](#)
- [Static Data Dictionary Views with Dropped Columns in Oracle Database 10g Release 2 \(10.2\)](#)

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

The following static data dictionary view columns were dropped in Oracle Database 11g Release 1 (11.1):

Static Data Dictionary View	Dropped Columns
V\$DATAFILE	PLUGGED_IN

Static Data Dictionary Views with Dropped Columns in Oracle Database 10g Release 2 (10.2)

The following static data dictionary view columns were dropped in Oracle Database 10g Release 2 (10.2):

Static Data Dictionary View	Dropped Columns
DBA_HIST_SQLBIND	CHILD_NUMBER

Dynamic Performance View Changes

The following sections list changes to dynamic performance views (V\$ views) across different releases of Oracle Database:

- [Deprecated Dynamic Performance Views](#)
- [Obsolete Dynamic Performance Views](#)
- [Dynamic Performance Views with Renamed Columns](#)
- [Dynamic Performance Views with Dropped Columns](#)

See Also: The "What's New in Oracle Database Reference" section of *Oracle Database Reference* for a list of new dynamic performance views in Oracle Database 11g Release 1 (11.1)

Deprecated Dynamic Performance Views

The following sections list dynamic performance views that have been deprecated:

- [Dynamic Performance Views Deprecated in Oracle Database 10g Release 1 \(10.1\)](#)
- [Dynamic Performance Views Deprecated in Release 9.2](#)

Dynamic Performance Views Deprecated in Oracle Database 10g Release 1 (10.1)

The following dynamic performance views were deprecated in Oracle Database 10g 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 Deprecated in Release 9.2

The following dynamic performance views were deprecated in Oracle Database 9i Release 2 (9.2):

GV\$SORT_USAGE (replaced by GV\$TEMPSEG_USAGE)
 V\$SORT_USAGE (replaced by V\$TEMPSEG_USAGE)

Obsolete Dynamic Performance Views

The following sections list dynamic performance views that have been made obsolete:

- [Dynamic Performance Views Obsolete in Oracle Database 10g Release 1 \(10.1\)](#)
- [Dynamic Performance Views Obsolete in Release 9.2](#)

Dynamic Performance Views Obsolete in Oracle Database 10g Release 1 (10.1)

The following dynamic performance views were made obsolete in Oracle Database 10g Release 1 (10.1):

GV\$ Views	V\$ Views
GV\$COMPATIBILITY	V\$COMPATIBILITY

GV\$ Views	V\$ Views
GV\$COMPATSEG	V\$COMPATSEG
GV\$MLS_PARAMETERS	V\$MLS_PARAMETERS
GV\$MTS	V\$MTS

Dynamic Performance Views Obsolete in Release 9.2

The following dynamic performance views were made obsolete in Oracle Database9i Release 2 (9.2):

GV\$ Views	V\$ Views
GV\$LOADCSTAT	V\$LOADCSTAT
GV\$LOADTSTAT	V\$LOADTSTAT

Dynamic Performance Views with Renamed Columns

The following sections list dynamic performance views with renamed columns:

- [Dynamic Performance Views with Renamed Columns in Release 9.2](#)

Dynamic Performance Views with Renamed Columns in Release 9.2

The dynamic performance view columns listed in [Table A-1](#) were renamed in Oracle Database9i Release 2 (9.2):

Table A-1 Dynamic Performance Views with Renamed Columns in Release 9.2

Dynamic Performance View	Pre-Release 9.2 Column Name	Release 9.2 and Higher Column Name
GV\$ARCHIVE_DEST and V\$ARCHIVE_DEST	MANIFEST	REGISTER
GV\$ARCHIVE_DEST and V\$ARCHIVE_DEST	REGISTER	REMOTE_TEMPLATE
GV\$DATABASE and V\$DATABASE	STANDBY_MODE	PROTECTION_MODE
GV\$LOGMNR_CALLBACK and V\$LOGMNR_CALLBACK	CALLBACK_STATE	STATE
GV\$LOGMNR_CALLBACK and V\$LOGMNR_CALLBACK	CALLBACK_TYPE	TYPE
GV\$LOGMNR_CALLBACK and V\$LOGMNR_CALLBACK	CALLBACK_CAPABILITY	CAPABILITY
GV\$LOGMNR_REGION and V\$LOGMNR_REGION	ID	MEMSTATE
GV\$LOGMNR_REGION and V\$LOGMNR_REGION	CURRENT_STATE	STATE

Dynamic Performance Views with Dropped Columns

The following sections list dynamic performance views with dropped columns. If an application requires one or more of these columns, then modify the application accordingly:

- [Dynamic Performance Views with Dropped Columns in Release 9.2](#)

Dynamic Performance Views with Dropped Columns in Release 9.2

The following dynamic performance view columns were dropped in Oracle Database9i Release 2 (9.2):

Dynamic Performance View	Dropped Columns
GV\$DATABASE and V\$DATABASE	STANDBY_MODE
GV\$LOGMNR_CALLBACK and V\$LOGMNR_CALLBACK	FUNC_NAME CALLBACK_ID CALLBACK_RESULT_SIZE CALLBACK_STATE CALLBACK_TYPE CALLBACK_CAPABILITY NUMBER_INVOKED
GV\$LOGMNR_REGION and V\$LOGMNR_REGION	ID CURRENT_STATE

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 Oracle Database9i 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
```


Numerics

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

A

access control to network utility packages, 5-9
access-control lists (ACLs)
 changed behavior in 11g release 1 (11.1), 5-10
applications
 compatibility, 6-1
 linking with newer libraries, 6-4
 running against older server, 6-4
 upgrading, 6-1
 client/server configurations, 6-2
 compatibility rules, 6-3
 options, 6-5
 relinking rules, 6-3

ASM

 cluster ASM upgrade, 4-16
 database upgrade after, 4-17
 disk group compatibility, 5-6
 optional tasks after upgrade, 4-16
ASM_PREFERRED_READ_FAILURE_
 GROUPS, 4-15
automatic maintenance tasks management
 AutoTask, 5-5

Automatic Storage Management (ASM)

 disk group compatibility, 4-15
 file access control, 2
 post-upgrade topics, 4-7
 preferred read failure groups, 4-15
 rolling upgrade, 1-8
 upgrading, 3-2

automatic undo management

 migrating to, 4-8
 UNDO_MANAGEMENT, A-4
AutoTask, 5-5

B

backups
 after upgrading, 4-6
 before downgrading, 7-2
 preparing a strategy, 2-8
binary XML storage, 5-8

C

capturing and replaying database workload, 2-7
case sensitivity
 for passwords, 4-6
catdwgrd.sql script, 7-4
CATRELOD.SQL script, 7-6
CATUPGRD.SQL script, 3-42
change passwords
 for oracle-supplied accounts, 4-11
client software
 upgrading, 6-4
client-server configurations, 1-5
cluster ASM, 4-16
collecting optimizer statistics, B-1
command line options
 for Database Upgrade Assistant, 3-33
compatibility
 applications, 6-1
 checking for incompatibilities, 7-2
 compatibility level, 5-2
 COMPATIBLE initialization parameter, 5-1
 downgrading, 5-2
 initialization parameters, A-4
 original Export utility, 8-3
COMPATIBLE initialization parameter, 5-1
 checking, 5-3
 database structures, 5-2
 setting, 5-3
 when to set, 5-3

D

data copying
 using Export/Import, 8-1
data mining models, 5-6
Data Pump
 advantages of using, 2-4
Data Pump Export/Import
 recommendations, 2-4, 8-1
 versus Original Export/Import, 8-1
 when to use, 8-1
data type
 NCHAR, 5-18
Database Replay
 database workloads before upgrading, 2-7

- Database Upgrade Assistant (DBUA)
 - advantages, 2-3
 - command line options, 3-33
 - Oracle RAC enhancements, xvi
 - registering the database in the listener.ora file, 3-15
 - running, 3-16
 - silent mode, 3-33
 - single-instance database enhancements, xvi
 - single-instance database enhancements 11g, xvi
 - starting, 3-17
- database upgrade process
 - overview, 1-1
- Database Vault Administrator
 - deploy manually, 4-4
- databases
 - downgrading, 7-3
 - upgrading, 1-1
 - upgrading the client software, 6-4
- DB_BLOCK_SIZE
 - new default value, A-4
- DB_BLOCK_SIZE initialization parameter
 - compatibility, A-4
- DBMS_STATS package
 - upgrading statistics tables, 4-2
- DBMS_STATS procedure
 - use when creating a statistics table, B-2
- DBUA. *See* Database Upgrade Assistant
- deprecated dynamic performance views, A-9
- deprecated features
 - initialization parameters, A-1
 - static data dictionary views, A-7
- Developer/2000 Applications
 - upgrading, 6-7
- DGConnectIdentifier property, 4-8
- direct upgrades, 2-2
- disk group compatibility, 4-15
- disks
 - specifying preferred read failure groups, 4-15
- DMSYS schema objects, 5-6
- downgrading
 - backing up your database, 7-2
 - binary XML storage, 5-8
 - CATRELOD.SQL, 7-6
 - checking for incompatibilities, 7-2
 - ORADIM, 7-5
 - patchset releases, 7-1
 - procedure for, 7-3
 - scripts, 7-4
 - rerunning, 7-4
- downstream capture
 - upgrading, 3-5
- dump files
 - generated by export utilities, 8-2
- dynamic performance views
 - changes, A-8
 - deprecated, A-9
 - obsolete, A-9
 - with dropped columns, A-10
 - with renamed columns, A-10

E

- enforcing case-sensitivity for passwords, 4-6
- environment variables
 - required for upgrading, 3-40, 3-56
- evaluation of numeric literals, 5-18, 6-7
- export and import
 - recommendations, 2-4, 8-1
- Export utility, 8-1
 - data copying, 8-1
 - requirements, 8-2
- Export/Import
 - advantages and disadvantages, 2-4
 - benefits, 2-4
 - effects on upgraded databases, 2-4
 - incompatible data, 8-4
 - time requirements, 2-5
 - upgrading, 8-4
- extended distance cluster configurations
 - preferred read disks, 4-15
- extents
 - reading from secondary, 4-15

F

- FAILED_LOGIN_ATTEMPTS initialization parameter
 - DEFAULT limit, 5-13
- file access control
 - with the SYSASM role, 2
- Flash Recovery Area, 3-46
- Forms
 - upgrading Oracle Forms applications, 6-7

H

- Hardware Assisted Resilient Data (HARD)
 - upgrading systems, 5-3

I

- Import utility, 8-1
 - data copying, 8-1
 - requirements, 8-2
- importing
 - dump files for downgrades, 8-2
- incompatibilities
 - checking for, 7-2
- incompatible data
 - Export/Import, 8-4
- initialization parameters
 - adjusting for Oracle Database 10g, 3-38, 3-54, 4-13
 - ASM_PREFERRED_READ_FAILURE_GROUPS, 4-15
 - changes, A-1
 - compatibility, A-4
 - DB_BLOCK_SIZE, A-4
 - SESSION_CACHED_CURSORS, A-4
 - COMPATIBLE, 5-1
 - when to set, 5-3
 - deprecated, A-1
 - obsolete, A-3

INIT.ORA parameters. *See* initialization parameters
installation
 Oracle Database 11g release 1 (11.1), 3-5
instances
 starting after a downgrade, 7-6
INTEGER arithmetic
 evaluation of numeric literals, 5-18, 6-7
intermediate releases
 upgrading, 2-2
interoperability, 5-4

L

listener.ora file
 modifying, 3-15
listeners
 modifying with Oracle Net Configuration Assistant, 3-15
load testing, 2-8
LocalListenerAddress property
 Oracle Data Guard, 4-8
logical standby databases
 rolling upgrades, 1-7
login
 new DEFAULT limit, 5-13

M

maintenance tasks
 scheduling with AutoTask, 5-5
manual upgrade
 advantages, 2-3
 backup the database, 3-37
 OCR configuration, 4-13
migrating data
 to a different operating system, 3-4
moving data with export/import, 8-1
multiversioning, 1-5

N

NCHAR data type
 limitations, 5-18
network utility packages
 access control to, 5-9
new features
 adding after upgrade, 4-7
NUMBER arithmetic
 evaluation of numeric literals, 5-18, 6-7
numeric computation
 evaluation of numeric literals, 5-18, 6-7
numeric literals
 evaluating, 5-18, 6-7

O

obsolete features
 dynamic performance views, A-9
 initialization parameters, A-3
 static data dictionary views, A-8
OCI application

 statically-linked, 6-4
OCI applications
 changing, 6-6
 dynamically-linked, 6-4
 upgrading, 6-2
 upgrading options, 6-5
OFA, 1-6
one-off patches, 1-7
OPatch utility
 rolling upgrades, 1-7
operating system
 migrating data to, 3-4
Optimal Flexible Architecture. *See* OFA
optimizer statistics
 collecting for dictionary objects, B-1
 creating a table to collect, B-2
Oracle Application Express
 update, 4-3
Oracle Application Express configuration, 4-3
Oracle Cluster Registry (OCR)
 upgrading manually, 4-13
Oracle Clusterware
 upgrading, 3-2
Oracle Clusterware configuration
 downgrade, 7-7
Oracle Data Guard
 configuring broker properties, 4-8
 rolling upgrades, 1-7
Oracle Database Express Edition
 upgrading to Oracle Database, 1-9
Oracle Database XE
 upgrading to Oracle Database, 1-9
Oracle Express Edition
 recommended tasks after upgrade, 4-11
Oracle home
 multiple, 1-5
Oracle Net Configuration Assistant, 3-15
Oracle Net Services
 relinking, 6-2
Oracle Real Application Clusters, xvi
 11gR1 enhancements, xvi
 rolling upgrades with OPatch, 1-7
 upgrading, 3-2
Oracle release numbers, 1-4
Oracle Streams
 downstream capture
 upgrading, 3-5
 rolling upgrades, 1-7
Oracle Universal Installer, 1-1
Oracle XML Database
 binary XML storage, 5-8
ORACLE_BASE environment variable
 , xv
oracle-supplied accounts
 change passwords, 4-11
ORADIM
 downgrading, 7-5
 upgrading, 3-40, 3-55
original Export utility
 downward compatibility, 8-3

Original Export/Import
versus Data Pump Export/Import, 8-1
when to use, 8-1

P

passwords
case sensitive, 4-6
patchset releases
downgrading, 7-1
PL/SQL Native Compilation, 5-8
post-upgrade status tool, 3-44
precompilers
applications
changing, 6-6
upgrading options, 6-5
upgrading applications, 6-2
preferred read failure groups
setting up, 4-15
preparing to upgrade
collecting optimizer statistics, B-1

R

recovery catalog
upgrading, 4-2
releases
definition, 1-4
multiple, 1-5
upgrade paths, 2-2
relinking with Oracle Net Services, 6-2
rollback segments
migrating to automatic undo management, 4-8
rolling upgrades, 1-7
Oracle Real Application Clusters and OPatch, 1-7
with logical standby databases and SQL
Apply, 1-7
with Streams, 1-7
running multiple Oracle releases on the same
computer, 1-5

S

scheduling
AutoTask, 5-5
schemas
collecting system component statistics, B-1
scripts
downgrading, 7-4
rerunning, 7-4
upgrading, 3-7, 3-42, 3-44
security
case-sensitive passwords, 4-6
server parameter file
migrating to, 4-11
SESSION_CACHED_CURSORS
change in behavior, A-4
SESSION_CACHED_CURSORS initialization
parameter
compatibility, A-4
single-instance ASM upgrade

ASM
single-instance upgrade, 4-16
SPFILE
upgrading systems with HARD-compliant
storage, 5-3
SPNC_COMMANDS file
release 10.1, 5-18
SQL Access Advisor, 5-10
SQL Performance Analyzer, 2-7
SQL plan baseline, 5-7
SQL plan management, 2-7, 5-7
SQL*Plus
scripts
upgrading, 6-7
Standard Edition
moving to Enterprise Edition, 1-8
starter database, 5-10
STARTUP UPGRADE command, 7-6
static data dictionary views
changes, A-7
deprecated, A-7
obsolete, A-8
with dropped columns, A-8
statistics
collecting for dictionary objects, B-1
collecting for system component schemas, B-1
creating a table for, B-2
importing with DBMS_STATS PL/SQL
procedure, B-2
statistics tables
upgrading, 4-2
status tools
for upgrades and post-upgrade, 3-44
SYSASM Privilege, 5-6
SYSASM role
ASM file access control, 2
system component schemas
collecting statistics for, B-1

T

testing
applications for upgrade, 2-9
developing a plan, 2-5
functional for upgrade, 2-6
integration for upgrading, 2-6
minimal for upgrade, 2-6
performance for upgrade, 2-6
the upgrade process, 2-8
the upgraded test database, 2-9
using Database Replay, 2-7
volume/load stress for upgrade, 2-8
TIMESTAMP WITH TIMEZONE data type, 3-12
troubleshooting
upgrades, 3-45

U

UNDO_MANAGEMENT initialization
parameter, 4-8

- default, A-4
- upgrade methods
 - choosing, 2-3
 - Database Upgrade Assistant, 1-1, 2-3
 - Export/Import, 2-4, 8-4
 - manual, 2-3, 3-37
 - silent mode, 3-33
- upgrade paths, 2-2
- upgrade status tool, 3-44
- upgrading
 - abandoning, 3-47
 - applications, 6-1
 - compatibility rules, 6-3
 - options, 6-5
 - relinking, 6-3
 - ASM, 1-7, 3-2
 - backup strategy, 2-8
 - binary XML storage, 5-8
 - initialization parameters, 3-38, 3-54
 - new administrative procedures, 4-7
 - Oracle Application Express, 4-3
 - Oracle Clusterware, 3-2
 - Oracle Forms applications, 6-7
 - Oracle Real Application Clusters, 3-2
 - ORADIM, 3-40, 3-55
 - post upgrade actions, 4-1
 - preparation, 2-1
 - recovery catalog, 4-2
 - rolling upgrades, 1-7
 - running the CATUPGRD.SQL script, 3-42
 - scripts, 3-7, 3-42, 3-44
 - SQL*Plus scripts, 6-7
 - statistics tables, 4-2
 - testing, 2-5
 - troubleshooting, 3-45
 - using the Database Upgrade Assistant, 3-16
 - when to set the COMPATIBLE initialization parameter, 5-3
- utlu111s.sql
 - example, 3-44

W

- word size
 - 64-bit software, 1-6
- workloads
 - capturing and replaying, 2-7

