

Oracle® Database

Upgrade Guide

12c Release 1 (12.1)

E41397-11

March 2015

Oracle Database Upgrade Guide, 12c Release 1 (12.1)

E41397-11

Copyright © 2002, 2015, Oracle and/or its affiliates. All rights reserved.

Primary Authors: Cathy Shea, Douglas Williams

Contributor: The Oracle Database 12c documentation is dedicated to Mark Townsend, who was an inspiration to all who worked on this release.

Contributors: Drew Adams, Sanjay Bharadwaj, Ashmita Bose, Vickie Carbonneau, XiaoBin Chen, Jonathan Creighton, Ian Dai, Rhonda Day, Mike Dietrich, Joseph Errede, Winston Huang, Patricia Huey, Prakash Jashnani, Mark Jefferys, Sumit Jeloka, Maria Kaval, Alex Keh, Yash Khandelwal, Reema Khosla, Peter Knaggs, Paul Lane, Cindy Lim, Brian McCarthy, David McDermid, Matthew McKerley, Valarie Moore, Louise Morin, Sandeep Nimmakuri, Carol Palmer, Satish Panchumarthy, Robert Pang, Gurumurthy Ramamurthy, Bert Rich, Mark Richwine, Craig Santelman, Samit Sathe, Neeraj Shodhan, Khethavath Singh, Carol Tagliaferri, Srividya Tata, Andrew Wesley

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

| | |
|---|------|
| Preface | xv |
| Audience | xv |
| Documentation Accessibility | xvi |
| Related Documentation | xvi |
| Conventions | xvi |
| | |
| Changes in This Release for Oracle Database Upgrade Guide | xvii |
| New Features | xvii |
| Deprecated Features | xix |
| Desupported Features | xix |
| | |
| 1 Introduction to Upgrading Oracle Database | |
| 1.1 Overview of Oracle Database Upgrade Tools and Processes..... | 1-1 |
| 1.1.1 Upgrade and Data Migration Methods and Processes | 1-2 |
| 1.1.2 Scripts for Upgrading Oracle Database..... | 1-2 |
| 1.1.3 About Log File Location and DIAGNOSTIC_DEST..... | 1-4 |
| 1.2 Where to Find the Latest Information About Upgrading Oracle Database | 1-5 |
| 1.3 Oracle Database Releases That Support Direct Upgrade..... | 1-5 |
| 1.4 Major Steps in the Upgrade Process for Oracle Database..... | 1-6 |
| 1.5 About Oracle Database Release Numbers | 1-9 |
| 1.5.1 Convention for Referring to Release Numbers in <i>Oracle Database Upgrade Guide</i> | 1-9 |
| 1.6 About Running Multiple Oracle Releases..... | 1-9 |
| 1.6.1 Databases in Multiple Oracle Homes on the Same Computer..... | 1-10 |
| 1.6.2 Databases in Multiple Oracle Homes on Separate Computers..... | 1-10 |
| 1.7 Compatibility and Interoperability between Oracle Database Releases..... | 1-10 |
| 1.7.1 What Is Oracle Database Compatibility? | 1-10 |
| 1.7.1.1 The COMPATIBLE Initialization Parameter in Oracle Database..... | 1-11 |
| 1.7.1.2 Values for the COMPATIBLE Initialization Parameter in Oracle Database..... | 1-11 |
| 1.7.1.3 Compatibility and Oracle Database File System (DBFS) | 1-12 |
| 1.7.1.4 About Downgrading and Compatibility for Upgrading Oracle Database | 1-12 |
| 1.7.1.5 How the COMPATIBLE Initialization Parameter Operates in Oracle Database | 1-12 |
| 1-12 | |
| 1.7.1.6 Checking the Compatibility Level of Oracle Database | 1-13 |
| 1.7.1.7 When to Set the COMPATIBLE Initialization Parameter in Oracle Database.. | 1-13 |
| 1.7.2 What Is Interoperability for Oracle Database Upgrades?..... | 1-13 |

| | | |
|--------|--|------|
| 1.8 | Optimal Flexible Architecture (OFA) and Oracle Database Installations | 1-13 |
| 1.9 | Converting 32-bit Oracle Database to 64-bit Oracle Database | 1-14 |
| 1.10 | About Rolling Upgrades for Oracle ASM and Oracle RAC Databases | 1-14 |
| 1.10.1 | Summary of Methods for Performing Rolling Upgrades for Oracle Database | 1-14 |
| 1.11 | Moving From Standard Edition to Enterprise Edition of Oracle Database..... | 1-15 |
| 1.12 | Moving from Enterprise Edition to Standard Edition of Oracle Database | 1-17 |
| 1.13 | Moving from Oracle Database Express Edition (Oracle Database XE) to Oracle Database | 1-17 |

2 Preparing to Upgrade Oracle Database

| | | |
|---------|---|------|
| 2.1 | Tasks for Preparing to Upgrade Oracle Database..... | 2-1 |
| 2.1.1 | Become Familiar with New Oracle Database Features | 2-2 |
| 2.1.2 | Determine the Upgrade Path for Oracle Database | 2-2 |
| 2.1.3 | Choose an Upgrade Method for Oracle Database | 2-3 |
| 2.1.3.1 | The Automated (DBUA) Method for Upgrading Oracle Database | 2-3 |
| 2.1.3.2 | The Manual, Command-line Method for Upgrading Oracle Database..... | 2-4 |
| 2.1.3.3 | The Export/Import Method for Migrating Data When Upgrading Oracle Database | 2-4 |
| 2.1.4 | Choose a New Location for Oracle Home when Upgrading | 2-6 |
| 2.1.5 | Develop a Test Plan for Upgrading Oracle Database..... | 2-6 |
| 2.1.5.1 | Upgrade Testing | 2-6 |
| 2.1.5.2 | Minimal Testing..... | 2-7 |
| 2.1.5.3 | Functional Testing | 2-7 |
| 2.1.5.4 | High Availability Testing | 2-7 |
| 2.1.5.5 | Integration Testing | 2-7 |
| 2.1.5.6 | Performance Testing | 2-7 |
| 2.1.5.7 | Volume and Load Stress Testing for Oracle Database Upgrades | 2-10 |
| 2.1.6 | Prepare a Backup Strategy before Upgrading Oracle Database | 2-11 |
| 2.2 | Requirements for Upgrading Oracle Database | 2-11 |
| 2.2.1 | Relocating Existing Data Files to the New Oracle Database Environment..... | 2-12 |
| 2.2.2 | About Upgrading PL/SQL Packages That Are Not Installed by Default..... | 2-12 |
| 2.2.3 | Saving Oracle Enterprise Manager Database Control Configuration and Data | 2-12 |
| 2.2.4 | Manually Removing DB Control with emremove.sql..... | 2-13 |
| 2.2.5 | About Upgrading an Oracle ASM Instance | 2-14 |
| 2.2.5.1 | Determining If Oracle ASM is Included in the Current Database Configuration | 2-14 |
| 2.2.6 | About Upgrading Oracle Grid Infrastructure and Oracle Clusterware | 2-15 |
| 2.2.6.1 | Oracle Clusterware Must Be Running on Local Node When Upgrading..... | 2-16 |
| 2.2.6.2 | About Upgrading an Oracle Real Application Clusters (Oracle RAC) Database | 2-16 |
| | with DBUA | 2-16 |
| 2.2.6.3 | About Upgrading and Inaccessible Nodes | 2-16 |
| 2.2.6.4 | About Requirements for Time Synchronization on Oracle RAC | 2-17 |
| 2.2.6.5 | Recommendations for Upgrading Oracle RAC and Oracle Databases That Use | 2-17 |
| | ASM | 2-17 |
| 2.2.6.6 | About Upgrading System Authentication for Oracle ASM Instances..... | 2-17 |
| 2.2.7 | About Upgrading with Read-Only and Offline Tablespaces..... | 2-18 |
| 2.2.8 | About Upgrading Using Standby Databases..... | 2-18 |
| 2.2.9 | About Upgrading Your Operating System..... | 2-19 |

| | | |
|----------|---|------|
| 2.2.10 | Transporting Data to a Different Operating System | 2-19 |
| 2.2.11 | About Oracle GoldenGate and Online Database Upgrade | 2-20 |
| 2.2.11.1 | Overview of Steps for Upgrading Oracle Database Using Oracle GoldenGate | 2-20 |
| 2.2.12 | About Upgrading Oracle OLAP Data Security Policies | 2-20 |
| 2.2.13 | Requirements for Upgrading Databases That Use Oracle Label Security and Oracle Database Vault | 2-21 |
| 2.2.13.1 | Requirement for Upgrading Oracle Databases That Use Oracle Database Vault..... | 2-21 |
| 2.2.13.2 | Running olspreupgrade.sql on Oracle Database Release 11.1.0.7 | 2-22 |
| 2.2.13.3 | Running olspreupgrade.sql on Oracle Database Release 10.2.0.5 or 11.2 | 2-22 |
| 2.2.14 | Requirement for Upgrading Databases that Use Oracle Warehouse Builder (OWB) | 2-23 |
| 2.2.14.1 | Add Oracle Database 12c Access to Existing standalone OWB 11.2.0.3 Installation.. | 2-23 |
| 2.2.14.2 | Keep Existing OWB 11.2.0.3 Installation In-place with Oracle Database Release 11.2.0.3 | 2-24 |
| 2.2.14.3 | Use the Standalone OWB 11.2.0.3 Installation Where Available..... | 2-24 |
| 2.2.15 | Removing the Unified Auditing Schema and Roles | 2-25 |
| 2.3 | Installing the New Oracle Database Software | 2-25 |
| 2.4 | Patch Set Updates and Requirements for Upgrading Oracle Database..... | 2-27 |
| 2.5 | About the Pre-Upgrade Information Tool for Oracle Database..... | 2-27 |
| 2.5.1 | Using the Pre-Upgrade Information Tool (preupgrd.sql) | 2-27 |
| 2.5.2 | Pre-Upgrade Information Tool Warnings and Recommendations for Oracle Database .. | 2-29 |
| 2.5.2.1 | Update Access Control Lists and Network Utility Packages..... | 2-30 |
| 2.5.2.2 | Evaluate Dependencies and Add ACLs for Network Utility Packages | 2-30 |
| 2.5.2.3 | About Database Links with Passwords from Earlier Oracle Database Releases | 2-31 |
| 2.5.2.4 | About Oracle Database Warnings for TIMESTAMP WITH TIME ZONE Data Type | 2-31 |
| 2.5.2.5 | Decrease Oracle Database Downtime by Gathering Optimizer Statistics | 2-32 |
| 2.5.2.6 | Identify Invalid Objects in Oracle Database with the utluiobj Script | 2-32 |
| 2.5.2.7 | Verify That Materialized View Refreshes Have Completed Before Upgrading | 2-33 |
| 2.5.2.8 | Ensure That No Files Need Media Recovery Before Upgrading | 2-33 |
| 2.5.2.9 | Ensure That No Files Are in Backup Mode Before Upgrading | 2-34 |
| 2.5.2.10 | Resolve Outstanding Distributed Transactions Before Upgrading | 2-34 |
| 2.5.2.11 | Purge the Database Recycle Bin Before Upgrading..... | 2-34 |
| 2.5.2.12 | Synchronize the Standby Database with the Primary Database When Upgrading ... | 2-35 |
| 2.5.2.13 | About Upgrading Oracle Application Express Databases | 2-35 |
| 2.5.3 | Sample Output of the Pre-Upgrade Information Tool..... | 2-35 |
| 2.6 | Backing Up Oracle Database for Upgrading | 2-38 |
| 2.7 | Preparing the New Oracle Home for Upgrading..... | 2-39 |
| 2.7.1 | Prerequisites for Preparing Oracle Home on Windows | 2-41 |
| 2.8 | Recommendations for Oracle Net Services When Upgrading Oracle Database | 2-41 |
| 2.9 | Testing the Upgrade Process for Oracle Database | 2-42 |
| 2.9.1 | Upgrade Oracle Call Interface (OCI) and Precompiler Applications | 2-42 |

| | | |
|------|---|------|
| 2.10 | Testing an Upgraded Test Copy of Oracle Database..... | 2-43 |
|------|---|------|

3 Upgrading Oracle Database

| | | |
|---------|--|------|
| 3.1 | Upgrade Strategies for Oracle Database..... | 3-1 |
| 3.1.1 | Upgrading a Non-CDB Oracle Database | 3-2 |
| 3.1.2 | Upgrading an Earlier Release and Plugging a PDB into a CDB..... | 3-2 |
| 3.1.3 | Upgrading a Multitenant (CDB) Oracle Database | 3-3 |
| 3.1.4 | Upgrading a Pluggable Database (PDB) | 3-4 |
| 3.2 | Upgrading with Oracle Database Upgrade Assistant (DBUA)..... | 3-6 |
| 3.2.1 | Recommendations for Using DBUA | 3-7 |
| 3.2.1.1 | About Stopping DBUA When Upgrading..... | 3-7 |
| 3.2.2 | How DBUA Processes the Upgrade for Oracle Database..... | 3-7 |
| 3.2.2.1 | Upgrade Scripts Invoked by DBUA | 3-8 |
| 3.2.3 | Using DBUA to Upgrade the Database on Linux, UNIX, and Windows Systems | 3-8 |
| 3.2.4 | About Windows Authentication When Upgrading | 3-21 |
| 3.2.5 | Moving a Database from an Existing 12c Oracle Home..... | 3-22 |
| 3.2.6 | Using DBUA in Silent Mode to Upgrade Oracle Database | 3-23 |
| 3.2.6.1 | Oracle DBUA Command Line Options for Silent Mode | 3-24 |
| 3.2.6.2 | DBUA Command Line Syntax for Silent Mode | 3-26 |
| 3.3 | About the Parallel Upgrade Utility for Oracle Database | 3-26 |
| 3.3.1 | General Steps for Running the New Parallel Upgrade Utility (catctl.pl) | 3-26 |
| 3.3.2 | Parallel Upgrade Utility (catctl.pl) Parameters | 3-28 |
| 3.3.3 | Example for Using the Parallel Upgrade Utility | 3-29 |
| 3.4 | Manually Upgrading a Non-CDB Oracle Database..... | 3-29 |
| 3.4.1 | Variables for using ORADIM When Upgrading Oracle Database on Windows | 3-34 |
| 3.5 | Manually Upgrading a Multitenant Container Oracle Database (CDB) | 3-35 |
| 3.6 | How the Data Dictionary is Upgraded and Status is Displayed | 3-40 |
| 3.6.1 | Parallel Upgrading of Data Dictionary..... | 3-40 |
| 3.6.2 | Change to Upgrade Status Setting | 3-40 |
| 3.6.3 | Change to Status After Running the Re-compilation utlrp.sql Script..... | 3-40 |
| 3.7 | About the catupgrd.sql Script in Earlier Releases of Oracle Database | 3-41 |
| 3.8 | About the catuppst.sql Script..... | 3-41 |
| 3.9 | About Transporting and Upgrading a Database (Full Transportable Export/Import) | 3-42 |
| 3.10 | Troubleshooting the Upgrade for Oracle Database | 3-42 |
| 3.10.1 | About Starting Oracle Database in Upgrade Mode..... | 3-43 |
| 3.10.2 | Running DBUA with Different ORACLE_HOME Owner | 3-44 |
| 3.10.3 | Oracle Database Upgrade Script Termination Errors | 3-44 |
| 3.10.4 | Resource Limits for Upgrading Oracle Database | 3-44 |
| 3.10.5 | SQL*Plus Edition Session Startup Error for Oracle Database..... | 3-45 |
| 3.10.6 | Manual Workaround for ORA-01408 on Oracle Database Index..... | 3-45 |
| 3.10.7 | Error ORA-00018 for DBCA Processes with Release 11.1.0.7 Databases..... | 3-46 |
| 3.10.8 | Error ORA-00020 Maximum Number of Processes Exceeded When Running utlrp.sql.. | 3-46 |
| 3.10.9 | Running the DBMS_DST Package After Upgrade Can Result in ORA-01822 | 3-46 |
| 3.10.10 | Continuing Upgrades After Server Restarts (ADVM/ACFS Driver Error)..... | 3-46 |
| 3.10.11 | Understanding Component Status of the Post-Upgrade Status Tool | 3-47 |

| | | |
|-----------|---|------|
| 3.10.11.1 | Sample Upgrade Summary Report for the Post-Upgrade Status Tool (utlu121s.sql) | 3-48 |
| 3.10.12 | Standard Edition Starter Database and Components with Status OPTION OFF ... | 3-53 |
| 3.10.13 | Adjusting Oracle ASM Password File Location After Upgrade..... | 3-53 |
| 3.10.14 | Error ORA-27248: sys.dra_reevaluate_open_failures is running | 3-53 |
| 3.10.15 | Oracle Database Enterprise User Security, OLS-OID, and Provisioning Profile Error | 3-53 |
| 3.10.16 | 32K Migration Error with utl32k.sql and MAX_STRING_SIZE | 3-54 |
| 3.10.17 | ASM Loses Rolling Migration State If CRS Shutdown On All Nodes | 3-54 |
| 3.10.18 | Data Type Versioning Could Cause Cross-Version Replication (ORA-26656) | 3-55 |
| 3.10.19 | Oracle XML DB Errors ORA-64123 and ORA-14097 | 3-55 |
| 3.11 | Rerunning the Upgrade for Oracle Database | 3-55 |
| 3.11.1 | Rerunning the Upgrade with the Upgrade (catctl.pl) Script..... | 3-55 |
| 3.11.2 | Options for Rerunning the Upgrade for a Multitenant Database (CDB) | 3-57 |
| 3.11.2.1 | Rerun the Entire Upgrade for the CDB | 3-58 |
| 3.11.2.2 | Rerun the Upgrade Only on Specified CDBs | 3-58 |
| 3.11.2.3 | Rerun the Upgrade While Other PDBs Are Online Using an Exclusion List ... | 3-59 |
| 3.11.2.4 | Rerun the Upgrade Using an Inclusion List to Specify a CDB or PDBs..... | 3-60 |
| 3.11.2.5 | Restarting the Upgrade from a Specific Phase that Failed Using -p | 3-60 |

4 Post-Upgrade Tasks for Oracle Database

| | | |
|--------|--|-----|
| 4.1 | How to Show the Current State of the Oracle Data Dictionary | 4-1 |
| 4.2 | About OPatch Commands After Upgrading Oracle Database | 4-2 |
| 4.3 | Required Tasks to Complete After Upgrading Oracle Database..... | 4-2 |
| 4.3.1 | Set Environment Variables on Linux and UNIX Systems After Manual Upgrades .. | 4-2 |
| 4.3.2 | Set oratab and Scripts to Point to the New Oracle Location After Upgrading Oracle Database | 4-3 |
| 4.3.3 | Enable the New Extended Data Type Capability | 4-3 |
| 4.3.4 | Adjust Minimum and Maximum for Parallel Execution Servers | 4-3 |
| 4.3.5 | Upgrade the Recovery Catalog After Upgrading Oracle Database | 4-4 |
| 4.3.6 | Upgrade the Time Zone File Version After Upgrading Oracle Database | 4-4 |
| 4.3.7 | Upgrade Statistics Tables Created by the DBMS_STATS Package After Upgrading Oracle Database | 4-4 |
| 4.3.8 | Upgrade Externally Authenticated SSL Users After Upgrading Oracle Database | 4-4 |
| 4.3.9 | Configure the FTP and HTTP Ports and HTTP Authentication for Oracle XML DB | 4-5 |
| 4.3.10 | Install Oracle Text Supplied Knowledge Bases After Upgrading Oracle Database .. | 4-5 |
| 4.3.11 | Update Your Oracle Application Express Configuration After Upgrading Oracle Database | 4-6 |
| 4.3.12 | Configure Access Control Lists (ACLs) to External Network Services | 4-6 |
| 4.3.13 | Enable Oracle Database Vault After Upgrading Oracle Database | 4-6 |
| 4.3.14 | Check for the SQLNET.ALLOWED_LOGON_VERSION Parameter Behavior | 4-7 |
| 4.4 | Required Task After Oracle Grid Infrastructure Upgrades..... | 4-7 |
| 4.4.1 | Using Environment Variables for Grid Infrastructure Installations | 4-8 |
| 4.5 | Requirement for Role-Allocated Software Owners and Database Upgrade After Oracle ASM Upgrade | 4-8 |
| 4.5.1 | Keeping the Existing User as the Oracle ASM Operating System User | 4-9 |
| 4.5.2 | Changing the Operating System User for Single-Instance Oracle ASM..... | 4-9 |

| | | |
|----------|--|------|
| 4.5.3 | Changing the Operating System User for an Oracle RAC Database | 4-9 |
| 4.6 | Recommended and Best Practices to Complete After Upgrading Oracle Database..... | 4-9 |
| 4.6.1 | Back Up the Database | 4-10 |
| 4.6.2 | Run the postupgrade_fixups.sql Script | 4-10 |
| 4.6.3 | Gather Fixed Objects Statistics with DBMS_STATS | 4-10 |
| 4.6.4 | Reset Passwords to Enforce Case-Sensitivity | 4-11 |
| 4.6.5 | Understand Changes with Oracle Grid Infrastructure | 4-11 |
| 4.6.6 | Understand Oracle ASM and Oracle Grid Infrastructure Installation and Upgrade..... | 4-12 |
| 4.6.7 | Add New Features as Appropriate..... | 4-12 |
| 4.6.8 | Develop New Administrative Procedures as Needed | 4-12 |
| 4.6.9 | Set Threshold Values for Tablespace Alerts | 4-13 |
| 4.6.10 | Migrate From Rollback Segments to Automatic Undo Mode..... | 4-13 |
| 4.6.11 | Configure Oracle Data Guard Broker | 4-14 |
| 4.6.12 | Migrate Tables from the LONG Data Type to the LOB Data Type..... | 4-14 |
| 4.6.13 | Migrate Your Upgraded Oracle Databases to Use Unified Auditing | 4-14 |
| 4.6.13.1 | About the Unified Auditing Migration Process for Oracle Database..... | 4-14 |
| 4.6.13.2 | Migrating to Unified Auditing for Oracle Database | 4-15 |
| 4.6.13.3 | Managing Earlier Audit Records After You Migrate to Unified Auditing..... | 4-17 |
| 4.6.13.4 | Removing the Unified Auditing Functionality | 4-17 |
| 4.6.13.5 | Documentation References if You Choose Not to Use Unified Auditing..... | 4-18 |
| 4.6.14 | Test the Upgraded Production Oracle Database..... | 4-18 |
| 4.7 | Recommended Tasks After Upgrading an Oracle RAC Database | 4-19 |
| 4.8 | Recommended Tasks After Upgrading Oracle ASM | 4-19 |
| 4.8.1 | Create A Shared Password File in the ASM Diskgroup | 4-19 |
| 4.8.2 | Reset Oracle ASM Passwords to Enforce Case-Sensitivity..... | 4-19 |
| 4.8.3 | Advance the Oracle ASM and Oracle Database Disk Group Compatibility | 4-20 |
| 4.8.4 | Set Up Oracle ASM Preferred Read Failure Groups | 4-20 |
| 4.9 | Recommended Tasks After Upgrading Oracle Database Express Edition | 4-21 |
| 4.10 | Optionally Update Oracle Application Express Packaged Applications | 4-21 |
| 4.11 | Tasks to Complete Only After Manually Upgrading Oracle Database | 4-22 |
| 4.11.1 | Change Passwords for Oracle Supplied Accounts | 4-22 |
| 4.11.2 | Create or Migrate Your Password File with ORAPWD..... | 4-23 |
| 4.11.3 | Migrate Your Initialization Parameter File to a Server Parameter File..... | 4-23 |
| 4.11.4 | Upgrade Oracle Text | 4-23 |
| 4.11.5 | Upgrade the Oracle Clusterware Configuration..... | 4-24 |
| 4.11.6 | Adjust the Initialization Parameter File for the New Release | 4-24 |
| 4.11.6.1 | Setting the COMPATIBLE Initialization Parameter | 4-24 |
| 4.11.6.2 | Configuring tnsnames.ora and Listener Parameters..... | 4-25 |
| 4.11.7 | Set CLUSTER_DATABASE Initialization Parameter For Oracle RAC | 4-26 |

5 Upgrading Applications After Upgrading Oracle Database

| | | |
|-------|--|-----|
| 5.1 | Overview of Upgrading Applications on a New Oracle Database Release | 5-1 |
| 5.2 | Compatibility Issues for Applications on Different Releases of Oracle Database | 5-1 |
| 5.3 | Upgrading Precompiler and OCI Applications in Oracle Database | 5-2 |
| 5.4 | Software Upgrades and Client and Server Configurations for Oracle Database | 5-2 |
| 5.4.1 | Types of Software Upgrades for Oracle Database Client and Server Software | 5-2 |

| | | |
|---------|--|-----|
| 5.4.1.1 | Oracle Database Major Release Upgrade..... | 5-3 |
| 5.4.1.2 | Oracle Database Maintenance Release Upgrade | 5-3 |
| 5.4.2 | Possible Client and Server Configurations for Oracle Database | 5-3 |
| 5.4.2.1 | Oracle Database Client and Server on Different Computers | 5-3 |
| 5.4.2.2 | Oracle Database Client and Server in Different Oracle Locations on the Same Computer 5-3 | |
| 5.4.2.3 | Oracle Database Client and Server in the Same Oracle Location..... | 5-3 |
| 5.5 | Compatibility Rules for Applications When Upgrading Oracle Database Client or Server Software 5-3 | |
| 5.5.1 | Rules for Upgrading Oracle Database Server Software..... | 5-4 |
| 5.5.1.1 | If You Do Not Change the Client Environment, Then You Are Not Required to Relink. 5-4 | |
| 5.5.1.2 | Applications Can Run Against Newer or Older Oracle Database Server Releases.... 5-4 | |
| 5.5.2 | Upgrading the Oracle Database Client Software..... | 5-4 |
| 5.5.2.1 | Applications Can Be Linked with Newer Libraries | 5-5 |
| 5.5.2.2 | Statically Linked Applications Must Always Be Relinked..... | 5-5 |
| 5.5.2.3 | Relinking Dynamically Linked Applications..... | 5-5 |
| 5.6 | Upgrading Options for Oracle Precompiler and OCI Applications..... | 5-5 |
| 5.6.1 | Option 1: Leave the Application Unchanged | 5-6 |
| 5.6.2 | Option 2: Precompile or Compile the Application Using the New Software..... | 5-6 |
| 5.6.3 | Option 3: Change the Application Code to Use New Oracle Database Features..... | 5-6 |
| 5.6.3.1 | Changing Precompiler Applications | 5-7 |
| 5.6.3.2 | Changing OCI Applications..... | 5-7 |
| 5.7 | Upgrading SQL*Plus Scripts and PL/SQL after Upgrading Oracle Database..... | 5-7 |
| 5.7.1 | Evaluation of Numeric Literals..... | 5-7 |
| 5.8 | About Upgrading Oracle Forms or Oracle Developer Applications | 5-8 |

6 Downgrading Oracle Database to an Earlier Release

| | | |
|---------|---|------|
| 6.1 | Supported Releases for Downgrading Oracle Database | 6-1 |
| 6.2 | Checking for Incompatibilities When Downgrading Oracle Database | 6-4 |
| 6.3 | Performing a Full Backup Before Downgrading Oracle Database..... | 6-4 |
| 6.4 | Performing Required Pre-Downgrade Steps for Oracle Database | 6-5 |
| 6.5 | Downgrading a CDB or Non-CDB Oracle Database | 6-7 |
| 6.6 | Downgrading a Single Pluggable Oracle Database (PDB)..... | 6-13 |
| 6.7 | Post-Downgrade Tasks for Oracle Database Downgrades..... | 6-14 |
| 6.7.1 | Oracle XML DB Authentication Recommendations for an Oracle Database Downgrade 6-14 | |
| 6.7.2 | Re-creating the Network Listener When Downgrading to Release 11.1.0.7 | 6-14 |
| 6.7.3 | Re-enabling Oracle Database Vault after Downgrading Oracle Database..... | 6-15 |
| 6.7.4 | Restoration of the Configuration for Oracle Clusterware | 6-15 |
| 6.7.5 | Restoring Oracle Enterprise Manager after Downgrading Oracle Database..... | 6-15 |
| 6.7.5.1 | Running the emdwgrd utility to restore Enterprise Manager Database Control | 6-17 |
| 6.7.6 | Restoring Oracle Application Express to the Earlier Version | 6-18 |
| 6.8 | Troubleshooting the Downgrade of Oracle Database | 6-19 |
| 6.8.1 | Oracle Multimedia Downgrade and imrelod.sql Script Error | 6-19 |

| | | |
|---------|--|------|
| 6.8.2 | Oracle Database Vault and dvreloc.sql Script Error | 6-19 |
| 6.8.3 | Downgrading Grid Infrastructure Standalone (Oracle Restart) After Successful or Failed Upgrade | 6-19 |
| 6.8.4 | Issue with Downgrading Oracle ACFS and Oracle Grid Infrastructure | 6-20 |
| 6.8.5 | Issue with Password for Database Links After Downgrading | 6-20 |
| 6.8.5.1 | Using Oracle Data Pump Export to Create a Dump File Containing All Existing Database Links | 6-20 |

7 Migrating Data Using Oracle Data Pump

| | | |
|-------|--|-----|
| 7.1 | Overview of Data Pump and Export/Import For Migrating Data | 7-1 |
| 7.2 | Migrating Data With Data Pump When Upgrading Oracle Database | 7-2 |
| 7.2.1 | Importing a Full Oracle Database Using a Network Link | 7-3 |
| 7.3 | Data Pump Requirements When Downgrading Oracle Database | 7-4 |

8 Deprecated and Desupported Features for Oracle Database 12c

| | | |
|----------|---|-----|
| 8.1 | Oracle Database Changes | 8-2 |
| 8.1.1 | Deprecation of Non-CDB Architecture | 8-2 |
| 8.1.2 | Deprecation of catupgrd.sql Script and Introduction of New catctl.pl Utility | 8-2 |
| 8.1.3 | Desupport of Oracle Enterprise Manager Database Control | 8-3 |
| 8.1.4 | Changes for Deinstallation and Cleanup of Oracle Base | 8-3 |
| 8.1.4.1 | Change for Standalone Deinstallation Tool..... | 8-3 |
| 8.1.4.2 | Desupport of CLEANUP_ORACLE_BASE Property..... | 8-4 |
| 8.1.4.3 | Deprecation of -cleanupOBase | 8-4 |
| 8.1.5 | Deprecated and Desupported Parameters..... | 8-4 |
| 8.1.5.1 | LOG_ARCHIVE_LOCAL_FIRST | 8-5 |
| 8.1.5.2 | SEC_CASE_SENSITIVE_LOGON..... | 8-5 |
| 8.1.5.3 | FILE_MAPPING | 8-5 |
| 8.1.5.4 | DELETE_CATALOG_ROLE of the O7_DICTIONARY_ACCESSIBILITY Parameter | 8-5 |
| 8.1.5.5 | RDBMS_SERVER_DN | 8-5 |
| 8.1.6 | Deprecated Views..... | 8-5 |
| 8.1.7 | Deprecation of Oracle Streams | 8-5 |
| 8.1.8 | Deprecation of Advanced Replication..... | 8-6 |
| 8.1.9 | Deprecation of Single-Character SRVCTL CLI Options | 8-6 |
| 8.1.10 | Desupported Features on Microsoft Windows Platforms | 8-6 |
| 8.1.10.1 | Desupport of Oracle COM Automation on Windows | 8-6 |
| 8.1.10.2 | Desupport of Oracle Objects for OLE..... | 8-6 |
| 8.1.10.3 | Desupport of Oracle Counters for Windows Performance Monitor | 8-7 |
| 8.1.10.4 | Changes for Oracle Data Provider for .NET..... | 8-7 |
| 8.1.11 | Desupport of Oracle Cluster File System (OCFS) on Windows | 8-7 |
| 8.1.11.1 | About Upgrading Oracle Database Release 10.2 or 11.1 and OCFS and RAW Devices | 8-7 |
| 8.1.11.2 | Desupport of cluvfy comp cfs for OCFS | 8-8 |
| 8.1.12 | Deprecation of Stored List of Administrative Users for Cluster Administration | 8-8 |
| 8.1.13 | Desupport for Raw Storage Devices | 8-8 |
| 8.1.14 | Deprecation of -checkpasswd for QOSCTL Quality of Service (QoS) Command | 8-8 |
| 8.1.15 | Change to VARCHAR2, NVARCHAR2, and RAW Datatypes | 8-8 |

| | | |
|----------|--|------|
| 8.1.16 | Deprecation of JPublisher | 8-9 |
| 8.1.17 | Changes for Oracle JDBC and SQLJ | 8-9 |
| 8.1.17.1 | Deprecation of SQLJ Inside the Server | 8-9 |
| 8.1.17.2 | Deprecated Oracle Update Batching | 8-9 |
| 8.1.17.3 | Deprecated EndToEndMetrics related APIs | 8-10 |
| 8.1.17.4 | Deprecated Stored Outlines | 8-10 |
| 8.1.17.5 | Deprecated Concrete Classes in oracle.sql Package | 8-10 |
| 8.1.17.6 | Deprecated defineColumnType Method | 8-10 |
| 8.1.17.7 | Deprecated CONNECTION_PROPERTY_STREAM_CHUNK_SIZE Property | 8-10 |
| 8.1.17.8 | Desupported Implicit Connection Caching | 8-10 |
| 8.1.18 | Changes for Oracle Call Interface | 8-11 |
| 8.1.19 | Deprecation of Oracle Restart | 8-11 |
| 8.1.20 | Changed Default for resource_limit Parameter | 8-11 |
| 8.2 | Oracle Business Intelligence and Data Warehousing Changes | 8-11 |
| 8.2.1 | Oracle Warehouse Builder (OWB) Not Installed with Oracle Database | 8-11 |
| 8.2.2 | Desupport of Change Data Capture | 8-12 |
| 8.2.3 | Desupported Features in Oracle Data Mining | 8-12 |
| 8.2.4 | Changes to Upgrading Oracle Data Mining Models | 8-12 |
| 8.2.5 | Desupport of OLAP Catalog (AMD) | 8-12 |
| 8.3 | Oracle Database Security Changes | 8-13 |
| 8.3.1 | Changes to Security Auditing Features | 8-13 |
| 8.3.2 | Deprecations for Oracle Label Security | 8-13 |
| 8.3.3 | Deprecated DBMS_NETWORK_ACL_ADMIN PL/SQL package Procedures | 8-14 |
| 8.3.4 | Deprecation of IGNORECASE and SEC_CASE_SENSITIVE_LOGON | 8-14 |
| 8.3.5 | Deprecation of SQLNET.ALLOWED_LOGON_VERSION Parameter | 8-15 |
| 8.3.5.1 | Upgrading a System that Did Not Have SQLNET.ALLOWED_LOGON_VERSION Parameter Setting | 8-15 |
| 8.3.6 | AUDIT_ADMIN and AUDIT_VIEWER Roles Note | 8-15 |
| 8.3.7 | Deprecation of Windows NTS Authentication Using the NTLM Protocol | 8-16 |
| 8.3.8 | Deprecation of Public Key Infrastructure for Transparent Data Encryption | 8-16 |
| 8.3.9 | Desupported Cipher Suites for Secure Sockets Layer (SSL) | 8-16 |
| 8.3.10 | Desupport of Database Rules Manager (RUL) and Expression Filter (EXF) | 8-16 |
| 8.3.11 | Changes for Oracle Data Guard Broker | 8-17 |
| 8.3.12 | Changes for Oracle Data Pump Export Utility | 8-17 |
| 8.3.13 | Changes for Oracle Database Vault | 8-17 |
| 8.3.13.1 | Deprecated Default Rule Sets | 8-17 |
| 8.3.13.2 | Deprecated Default Realms | 8-17 |
| 8.3.13.3 | Deprecated API | 8-18 |
| 8.3.14 | Changes for Oracle Database Semantic Technologies | 8-18 |
| 8.3.14.1 | VPD Support in Oracle Database Semantic Technologies | 8-18 |
| 8.3.14.2 | Version-Enabled Models Support In Oracle Database Semantic Technologies | 8-18 |
| 8.4 | Oracle Globalization Support Changes | 8-18 |
| 8.4.1 | Desupport of CSSCAN and CSALTER | 8-18 |
| 8.5 | Oracle Multimedia Changes | 8-19 |
| 8.6 | Oracle Net Services Changes | 8-19 |
| 8.6.1 | Desupport of Oracle Net Connection Pooling | 8-19 |

| | | |
|-------|---|------|
| 8.6.2 | Desupport of Oracle Names..... | 8-19 |
| 8.6.3 | Desupport of Oracle Net Listener Password | 8-19 |
| 8.6.4 | Desupport of SQLNET.KERBEROS5_CONF_MIT Parameter for Oracle Net Services | 8-19 |
| 8.6.5 | Desupport of Oracle Names Control Utility for Oracle Net Services | 8-20 |
| 8.6.6 | Deprecated NT LAN Manager (NTLM) Protocol for Oracle Net Services | 8-20 |
| 8.7 | Oracle XML Database Changes | 8-20 |
| 8.7.1 | Oracle XML DB is Mandatory and Cannot Be Uninstalled..... | 8-20 |
| 8.7.2 | Deprecated Features for Oracle XML Database | 8-20 |
| 8.8 | Oracle Text Changes..... | 8-23 |
| 8.8.1 | Desupport of CTXXPATH in Oracle Text and Oracle XML DB | 8-23 |
| 8.8.2 | Desupport of ALTER INDEX OPTIMIZE for Text Indexes..... | 8-23 |
| 8.8.3 | Desupport of SYNC [MEMORY memsize] for Text Indexes | 8-23 |

A Changes for Earlier Releases of Oracle Database

| | | |
|----------|---|------|
| A.1 | Behavior Changes in Oracle Database 11g Release 2 (11.2) | A-1 |
| A.1.1 | Deprecation of SNMP Support in Oracle Net Listener | A-2 |
| A.1.2 | Changes to PL/SQL Procedures..... | A-2 |
| A.1.3 | JOB_QUEUE_PROCESSES Parameter and Scheduling Jobs..... | A-2 |
| A.1.4 | Deprecated XML DB Constructs | A-3 |
| A.1.5 | Cursor_sharing=similar Deprecated in Oracle Database | A-3 |
| A.1.6 | Planned Desupport of Change Data Capture..... | A-3 |
| A.1.7 | Deprecation of the Data Mining Java API..... | A-3 |
| A.1.8 | Initialization Parameters Deprecated in Oracle Database 11g Release 2 (11.2) | A-3 |
| A.1.9 | Initialization Parameters Desupported in Oracle Database 11g Release 2 (11.2) | A-4 |
| A.1.10 | Static Data Dictionary Views Deprecated in Oracle Database 11g Release 2 (11.2).. | A-4 |
| A.1.11 | Dynamic Performance Views Deprecated in Oracle Database 11g Release 2 (11.2). | A-4 |
| A.1.12 | Deprecated Features in Oracle Database 11g Release 2 (11.2)..... | A-4 |
| A.1.13 | Changes to LOG_ARCHIVE_DEST_n Parameters..... | A-4 |
| A.2 | Behavior Changes in Oracle Database 11g Release 1 (11.1) | A-5 |
| A.2.1 | Initialization Parameters Deprecated in Oracle Database 11g Release 1 (11.1) | A-5 |
| A.2.2 | Initialization Parameters Desupported in Oracle Database 11g Release 1 (11.1) | A-6 |
| A.2.3 | Static Data Dictionary Views with Dropped Columns in Oracle Database 11g Release 1 (11.1) | A-6 |
| A.2.4 | Deprecated Features in Oracle Database 11g Release 1 (11.1)..... | A-6 |
| A.2.5 | Automatic Maintenance Tasks Management | A-7 |
| A.2.6 | New SYSASM Privilege and OSASM Group for Oracle ASM Administration | A-7 |
| A.2.7 | ASM Disk Group Compatibility..... | A-8 |
| A.2.8 | COMPUTE STATISTICS and ESTIMATE STATISTICS Clauses | A-8 |
| A.2.9 | Oracle Data Mining Models and the DMSYS Schema Objects | A-8 |
| A.2.10 | Oracle Data Mining Scoring Engine..... | A-9 |
| A.2.11 | SQL Plan Management and Control of SQL Plan Baselines..... | A-9 |
| A.2.12 | Binary XML Support for Oracle XML Database | A-10 |
| A.2.12.1 | When Upgrading to Oracle Database 11g Release 1 (11.1)..... | A-10 |
| A.2.13 | PL/SQL Native Compilation and Access Control for Network Utility Packages .. | A-10 |
| A.2.13.1 | PL/SQL Native Compilation..... | A-10 |
| A.2.13.2 | Access Control for Network Utility Packages..... | A-10 |

| | | |
|--------|---|------|
| A.2.14 | PL/SQL Control Parameters..... | A-11 |
| A.2.15 | Change in WebDAV ACL Evaluation Rules in Oracle XML DB..... | A-11 |
| A.2.16 | Summary Management and SQL Access Advisor..... | A-12 |
| A.2.17 | SQL Access Advisor Tasks | A-12 |
| A.2.18 | Standard Edition Starter Database | A-12 |
| A.2.19 | Core Dump Location..... | A-12 |
| A.2.20 | New Default Value for UNDO_MANAGEMENT | A-12 |
| A.2.21 | LOG_ARCHIVE_DEST_# Parameters..... | A-13 |
| A.2.22 | SHARED_POOL_SIZE Parameter..... | A-13 |
| A.2.23 | JOB_QUEUE_PROCESSES Parameter New Default Setting | A-13 |
| A.2.24 | Automatic Diagnostic Repository | A-14 |
| A.3 | Behavior Changes in Oracle Database 10g Release 2 (10.2) | A-14 |
| A.3.1 | Initialization Parameters Deprecated in Oracle Database 10g Release 2 (10.2) | A-14 |
| A.3.2 | Initialization Parameters Desupported in Oracle Database 10g Release 2 (10.2) | A-15 |
| A.3.3 | Static Data Dictionary Views with Dropped Columns in Oracle Database 10g Release 2 (10.2) | A-15 |
| A.3.4 | SQL..... | A-15 |
| A.3.5 | CONNECT Role | A-15 |
| A.3.6 | Time Zone Files | A-15 |
| A.3.7 | New Limit for FAILED_LOGIN_ATTEMPTS..... | A-15 |

Index

Preface

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

Oracle Database Upgrade Guide contains information that describes the features and functions of Oracle Database (also known as the standard edition) and Oracle Database Enterprise Edition products. Oracle Database and Oracle Database Enterprise Edition have the same basic features. However, several advanced features are available only with Oracle Database 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 performs Oracle Database upgrades.

To use this document, you must be familiar with:

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

Documentation Accessibility

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

Access to Oracle Support

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

Related Documentation

For more information, see these Oracle resources:

- "Oracle Upgrade Companion" on My Oracle Support at <http://support.oracle.com>
- *Oracle Database Concepts* for a comprehensive introduction to the concepts and terminology used in this manual
- *Oracle Database Administrator's Guide* for information about administering Oracle Database
- *Oracle Database SQL Language Reference* for information on Oracle Database SQL commands and functions
- *Oracle Database Utilities* for information about utilities bundled with Oracle Database
- *Oracle Database Net Services Administrator's Guide* for information about Oracle Net Services

Many of the examples in this guide use the sample schemas, 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.

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. |

Changes in This Release for Oracle Database Upgrade Guide

The new release of Oracle Database provides improvements to upgrade performance, automation, and reporting.

This preface includes these topics:

- [New Features](#)
- [Deprecated Features](#)
- [Desupported Features](#)

New Features

Oracle Database 12c includes many new features, some of which directly affect upgrading. The following features are new for the upgrade process.

- [Pre-Upgrade Information Tool](#)
- [Parallel Processing for Database Upgrade](#)
- [Enhanced Pre-Upgrade and Post-Upgrade Summary Report](#)
- [Oracle Grid Infrastructure Upgrade Enhancements](#)
- [Pluggable Databases](#)
- [Oracle XML Database is Installed When You Upgrade](#)

See Also: *Oracle Database New Features Guide* for a complete description of the new features in Oracle Database 12c

Pre-Upgrade Information Tool

A new Pre-Upgrade Information Tool has been created for Oracle Database 12c. This new utility provides fixup scripts to address issues that might be present both before and after the upgrade. Database Upgrade Assistant (DBUA) and the Pre-Upgrade Information Tool perform detailed checks of the database to be upgraded, and the special scripts are generated specifically to fix any problems that are found.

During the pre-upgrade phase, you can interactively run fixup scripts to resolve issues that are found. Post-upgrade fixup scripts are generated depending on the results of the upgrade process. DBUA prompts you to run the appropriate scripts as critical issues are found.

See Also: "[Using the Pre-Upgrade Information Tool \(preupgrd.sql\)](#)" on page 2-27

Parallel Processing for Database Upgrade

In Oracle Database 12c, the `catupgrd.sql` Upgrade Utility has been replaced with the `catctl.pl` Parallel Upgrade Utility. You can run upgrade scripts and processes in parallel, taking full advantage of CPU capacity and shortening the upgrade time. You can choose to upgrade your database in parallel or serial mode. In earlier releases, only serial mode was available, which meant that the upgrade scripts and processes ran one after the other, regardless of how much CPU capacity was available.

Database Upgrade Assistant (DBUA) takes full advantage of the Parallel Upgrade Utility, and by default DBUA runs in parallel upgrade mode. Parallel processing decreases the time for the database upgrade by optimally utilizing the CPU power of the system in which the database resides.

See Also: ["About the Parallel Upgrade Utility for Oracle Database"](#)
on page 3-26

DBUA Can Be Restarted During the Upgrade Process

In this release, there are more options for recovering from failures during upgrade and for restarting the upgrade from the point where the failure occurred. For example, Oracle Database 12c introduces Oracle Recovery Manager (RMAN) support and Guaranteed Restore Point features. You can fix a problem that DBUA encounters and then return to the upgrade process.

- With achievement logging that tracks the upgrade progress, you now have the option to use Guaranteed Restore Point to restore your database if the upgrade fails. DBUA automatically creates a Restore Point.
 - If an error occurs while DBUA during pre-upgrade steps, including backup and restore and listener migration, then you can restart DBUA.
 - If post-upgrade steps fail due to encountering an error, then DBUA displays the manual steps for you to finish the upgrade. If the post-upgrade steps stop, then you can defer the saved post upgrade steps to finish the upgrade manually. Post-upgrade steps include moving data, post migration processing, upgrading the time zone file, or configuring Oracle Enterprise Manager.
- The option to use Oracle RMAN to restore your database in case the upgrade fails has been improved in this release. In certain cases, DBUA now enables you to restore your database from your own backup.
- With RMAN backup, DBUA supports backing up of databases that use Oracle Automatic Storage Management (Oracle ASM).

Enhanced Pre-Upgrade and Post-Upgrade Summary Report

DBUA generates HTML reports before and after the upgrade. These reports include component status, and also provide the ability to drill down to database objects, including invalid objects where an action to fix a problem might be available.

See Also: ["Upgrading with Oracle Database Upgrade Assistant \(DBUA\)"](#) on page 3-6

Oracle Grid Infrastructure Upgrade Enhancements

Upgrading Oracle Grid Infrastructure for Oracle Database 12c includes root script automation procedures, improved ability to perform full release and patch upgrades, and enhancements to status checks and upgrade checkpoints to provide improved stability and ease of use.

See Also: *Oracle Grid Infrastructure Installation Guide* for your operating system for information about upgrading Oracle Grid Infrastructure

Pluggable Databases

The pluggable database (PDB) feature is new in Oracle Database 12c. Oracle Database uses multitenant architecture and you can configure a multitenant container database (CDB) and have one PDB. If you purchase the Oracle Multitenant option, then you can have multiple PDBs inside a CDB. One of the key benefits of the multitenant architecture is that you can upgrade or apply a patch to a CDB and the upgrade or patch is implemented on all the PDBs within that CDB. This greatly simplifies administration for upgrading.

See Also: *Oracle Database Concepts* for an overview and documentation roadmap for working with PDBs

Oracle XML Database is Installed When You Upgrade

Oracle XML Database (Oracle XML DB) is now a mandatory component of Oracle Database. You cannot uninstall it, and there is no option not to include it when you create a new database. If during the upgrade, Oracle XML DB is not found, then Oracle XML DB is automatically loaded into the SYSAUX tablespace. The upgraded Oracle Database includes Oracle XML DB.

See Also: *Oracle XML DB Developer's Guide* for information about new features in Oracle XML DB

Deprecated Features

In addition to deprecating one or more features that directly affect the upgrade process, this release includes deprecations for a range of Oracle Database features and respective replacements for those features.

- The `catupgrd.sql` upgrade script is deprecated in Oracle Database 12c. The Parallel Upgrade Utility, `catctl.pl` replaces `catupgrd.sql`. See "[Deprecation of catupgrd.sql Script and Introduction of New catctl.pl Utility](#)" on page 8-2 for more details.
- Changes and deprecations that affect the behavior of Oracle Database after upgrading to the new Oracle Database 12c release are described in [Chapter 8, "Deprecated and Desupported Features for Oracle Database 12c."](#)
- Changes and deprecations that affect the behavior of earlier releases are described in [Appendix A, "Changes for Earlier Releases of Oracle Database."](#)

Desupported Features

The comprehensive list of features that are desupported in Oracle Database 12c is provided in [Chapter 8, "Deprecated and Desupported Features for Oracle Database 12c."](#) Oracle recommends that you check the entire chapter for changes that affect your database environment.

Introduction to Upgrading Oracle Database

Oracle provides upgrade options and strategies that are designed for your database environment, and an array of tools that automate the Oracle Database upgrade process.

This chapter contains the following topics:

- [Overview of Oracle Database Upgrade Tools and Processes](#)
- [Where to Find the Latest Information About Upgrading Oracle Database](#)
- [Oracle Database Releases That Support Direct Upgrade](#)
- [Major Steps in the Upgrade Process for Oracle Database](#)
- [About Oracle Database Release Numbers](#)
- [About Running Multiple Oracle Releases](#)
- [Compatibility and Interoperability between Oracle Database Releases](#)
- [Optimal Flexible Architecture \(OFA\) and Oracle Database Installations](#)
- [Converting 32-bit Oracle Database to 64-bit Oracle Database](#)
- [About Rolling Upgrades for Oracle ASM and Oracle RAC Databases](#)
- [Moving From Standard Edition to Enterprise Edition of Oracle Database](#)
- [Moving from Enterprise Edition to Standard Edition of Oracle Database](#)
- [Moving from Oracle Database Express Edition \(Oracle Database XE\) to Oracle Database](#)

1.1 Overview of Oracle Database Upgrade Tools and Processes

Upgrading transforms an existing Oracle Database environment (including installed components and associated applications) into an Oracle Database 12c environment. The data dictionary for the database is upgraded to the new release. Upgrading does not directly affect user data; no data is touched, changed, or moved during an upgrade.

Migrating data refers to moving data from one Oracle Database into another database previously created for migrating or moving the data. You migrate data when you need to move your database environment to a new hardware or operating system platform, or to a new character set. Migrating does not include upgrading to the latest release. The upgrade process is handled separately after you migrate the data.

The upgrade steps in *Oracle Database Upgrade Guide* apply to all operating systems, except where operating system-specific details might be needed. In this case you are instructed to refer to the *Oracle Database Installation Guide* for your operating system.

See Also: ["Migrating Data With Data Pump When Upgrading Oracle Database"](#) on page 7-2 for information about migrating your data when upgrading

This section contains these topics:

- [Upgrade and Data Migration Methods and Processes](#)
- [Scripts for Upgrading Oracle Database](#)
- [About Log File Location and DIAGNOSTIC_DEST](#)

1.1.1 Upgrade and Data Migration Methods and Processes

Oracle provides features and products to automate the upgrade process and ensure the most efficient use of time. Oracle Database supports the following methods for upgrading a database to the new release:

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

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

- Manual upgrade using SQL scripts and utilities
Enables upgrades to be performed at the command line with SQL scripts and utilities.
- Oracle Data Pump
Provides export and import utilities. Oracle Data Pump can perform a full or partial export from your database, followed by a full or partial import into the new release of Oracle Database. Export/Import in Oracle Data Pump 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 12c database. Data copying can copy a subset of the data, leaving the database unchanged.

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

1.1.2 Scripts for Upgrading Oracle Database

For the upgrade process, Oracle provides a set of tools and scripts that you run before, during, and after upgrading. [Table 1-1, "Upgrade, Post-Upgrade, and Downgrade Scripts"](#) lists the various scripts and tools with a description for each. Some of these scripts cannot be run in UPGRADE mode.

See Also: ["About Log File Location and DIAGNOSTIC_DEST"](#) on page 1-4 for information about where log files are saved

Table 1–1 Upgrade, Post-Upgrade, and Downgrade Scripts

| Script | Description |
|--------------|---|
| catcon.pl | Must be run in UPGRADE mode. This script is used when upgrading a CDB. |
| catctl.pl | Must be run in UPGRADE mode. In Oracle Database 12c, the <code>catupgrd.sql</code> Upgrade Utility has been replaced with the <code>catctl.pl</code> Parallel Upgrade Utility. Upgrade scripts and processes can be run in parallel, taking full advantage of CPU capacity and shortening the upgrade time. DBUA uses this tool. See "About the Parallel Upgrade Utility for Oracle Database" on page 3-26. |
| catdwgrd.sql | This is the downgrade script, which is used in the procedure to downgrade to the earlier release from which you upgraded. See "Downgrading a CDB or Non-CDB Oracle Database" on page 6-7. |
| catnoamd.sql | Do not run in UPGRADE mode. Run this script only after upgrading. The <code>catnoamd.sql</code> script drops the OLAP catalog component, also referred to as AMD. See "Desupport of OLAP Catalog (AMD)" on page 8-12. |
| catnoexf.sql | The upgrade process runs this script for you; however, you can run the <code>catnoexf.sql</code> script before the upgrade. Do not run this script in UPGRADE mode. The <code>catnoexf.sql</code> script removes the Expression Filter (EXF) and Database Rules Manager (RUL) components, which are desupported in this release. See "Desupport of Database Rules Manager (RUL) and Expression Filter (EXF)" on page 8-16. |
| catuppst.sql | Do not run in UPGRADE mode. Run <code>catuppst.sql</code> , located in the <code>ORACLE_HOME/rdbms/admin</code> directory, to perform remaining upgrade actions that do not require the database to be in UPGRADE mode. If an Oracle patch set update (PSU) has been installed in the Oracle Home, then this script will automatically apply that PSU to the database. DBUA automatically runs <code>catuppst.sql</code> , and therefore, you only must run this script separately for manual upgrades as described in "Manually Upgrading a Multitenant Container Oracle Database (CDB)" on page 3-35. See "About the catuppst.sql Script" on page 3-41. IMPORTANT: Running this script is MANDATORY . Note: If upgrading manually, failure to run the <code>catuppst.sql</code> procedure causes performance degradation in your database over time. |
| emdwrdr | Do not run this script in UPGRADE mode. The <code>emdwrdr</code> utility saves Oracle Enterprise Manager Database Control (DB Control) configuration files and data for your release 11g database. Run <code>emdwrdr</code> before you upgrade in order to be able to restore DB Control if you need to downgrade Oracle Database 12c. See Also: Oracle Support document ID 1484775.1 at http://support.oracle.com/oip/faces/secure/km/DocumentDisplay.jspx?id=1484775.1 . See Also: "Saving Oracle Enterprise Manager Database Control Configuration and Data" on page 2-12. |

Table 1–1 (Cont.) Upgrade, Post-Upgrade, and Downgrade Scripts

| Script | Description |
|-------------------------------------|--|
| <code>emremove.sql</code> | <p>The <code>emremove.sql</code> script drops the Oracle Enterprise Manager-related schemas and objects. Use this script to manually remove DB Control. Running <code>emremove.sql</code> before starting the upgrade process minimizes downtime. This is an optional pre-upgrade step because the Parallel Upgrade Utility and DBUA automatically run this script. See "Manually Removing DB Control with emremove.sql" on page 2-13.</p> <p>IMPORTANT: If you need to preserve the DB Control configuration and data in order to have the option of downgrading and restoring DB Control, then you must first follow the procedure for using <code>emdwrdr</code> in "Saving Oracle Enterprise Manager Database Control Configuration and Data" on page 2-12.</p> |
| <code>olspreupgrade.sql</code> | <p>Do not run this script in UPGRADE mode. The <code>olspreupgrade.sql</code> script is a preprocessing script required for upgrading a database earlier than Oracle Database release 12.1 that uses Oracle Label Security (OLS) and Oracle Database Vault. See "Requirements for Upgrading Databases That Use Oracle Label Security and Oracle Database Vault" on page 2-21.</p> |
| <code>postupgrade_fixups.sql</code> | <p>The <code>postupgrade_fixups.sql</code> script is supplied with Oracle Database. Run <code>postupgrade_fixups.sql</code> after upgrading. DBUA runs this script automatically; however, you can run it any time after upgrading. See "Run the postupgrade_fixups.sql Script" on page 4-10.</p> |
| <code>preupgrd.sql</code> | <p>The <code>preupgrd.sql</code> Pre-Upgrade Information Tool is supplied with Oracle Database. Run <code>preupgrd.sql</code> any time before upgrading to analyze your database. The Pre-Upgrade Information Tool provides a preview of the items that DBUA checks and generates "fix-up" scripts that you can run to resolve issues that are flagged in the source database. (This script replaces the <code>utlu121i.sql</code> script.) See "About the Pre-Upgrade Information Tool for Oracle Database" on page 2-27.</p> |
| <code>utlrp.sql</code> | <p>Use <code>utlrp.sql</code> to recompile stored PL/SQL and Java code. DBUA runs this script automatically. When you upgrade manually, you must run this script as described in "Manually Upgrading a Multitenant Container Oracle Database (CDB)" on page 3-35.</p> |
| <code>utlu121s.sql</code> | <p>The <code>utlu121s.sql</code> Post-Upgrade Status Tool is supplied with Oracle Database and displays the version and elapsed upgrade time for each component in <code>DBA_REGISTRY</code>. The Post-Upgrade Status Tool can be run any time after upgrading the database. See "After the Upgrade" on page 2-4.</p> |
| <code>utluiobj.sql</code> | <p>Use <code>utluiobj.sql</code> to display only those objects which are newly invalid because of the upgrade process. See "Identify Invalid Objects in Oracle Database with the utluiobj Script" on page 2-32.</p> |

1.1.3 About Log File Location and DIAGNOSTIC_DEST

The location of diagnostic log files created by the upgrade scripts can vary depending on your environment variables and parameter settings. The `DIAGNOSTIC_DEST` initialization parameter sets the location of the Automatic Diagnostic Repository (ADR), which is a directory structure that is stored outside of the database.

If the `DIAGNOSTIC_DEST` initialization parameter is omitted or left null, then the database uses the following default settings:

- If environment variable `ORACLE_BASE` is set, then `DIAGNOSTIC_DEST` defaults to the directory designated by `ORACLE_BASE`. Therefore, the generated scripts and log files are created under `Oracle_Base/cfgtoollogs/tool_name/SID/upgraden`, where `n` represents the consecutive number for the upgrade that you performed on this system.

For example, during the upgrade, logs created by DBUA on a system with `ORACLE_BASE` defined appear under:

```
ORACLE_BASE/cfgtoollogs/dbua/SID/upgraden
```

Before running the actual upgrade, DBUA logs are saved under `Oracle_Base/cfgtoollogs/dbua/logs`.

- If `ORACLE_BASE` is not defined, then the generated scripts and log files are created under `ORACLE_HOME/cfgtoollogs/tool_name/SID/upgraden`, where `n` represents the consecutive number for the upgrade that you performed on this system.
- Beginning with Oracle Database 11g, because all diagnostic data, including the alert log, are stored in the ADR, the initialization parameters `BACKGROUND_DUMP_DEST` and `USER_DUMP_DEST` are deprecated. They are replaced by the initialization parameter `DIAGNOSTIC_DEST`, which identifies the location of the ADR.

See Also: *Oracle Database Administrator's Guide* for more information about using the `DIAGNOSTIC_DEST` initialization parameter and the ADR

1.2 Where to Find the Latest Information About Upgrading Oracle Database

Through its support website, Oracle provides late-breaking updates, discussions, and best practices about pre-upgrade requirements, upgrade processes, post-upgrade, compatibility, and interoperability.

To find the latest information about Oracle Database releases, go to My Oracle Support at <http://support.oracle.com>. You can search on note IDs or key words like "Database Upgrade." Help for using My Oracle Support is provided at http://docs.oracle.com/cd/E25290_01/index.htm.

Table 1–2 Information about Upgrading Oracle Database on My Oracle Support

| For information about... | Go to or see... |
|--|--|
| The complete knowledge base and information about upgrading to Oracle Database 12c | http://support.oracle.com and enter your search criteria. http://docs.oracle.com/cd/E25290_01/index.htm for information about how to use My Oracle Support (MOS) and support notes. |
| Performing an interim upgrade to Oracle Database 11g Release 2 (11.2) | <i>Oracle Upgrade Companion</i> ID 1462240.1 at My Oracle Support at http://support.oracle.com . Other upgrade companions are also available for upgrading to earlier releases. |
| Downloading and running the Pre-Upgrade Information Tool | <i>Oracle Database Pre-Upgrade Utility</i> ID 884522.1 at My Oracle Support at http://support.oracle.com . |

1.3 Oracle Database Releases That Support Direct Upgrade

You may perform a direct upgrade to Oracle Database 12c if your current Oracle Database is release 10.2.0.5, 11.1.0.7, or release 11.2.0.2 or later. If your current Oracle

Database is earlier than release 10.2.0.5, or is release 11.2.0.1, then directly upgrading to Oracle Database 12c is not supported. In this case, you are required to upgrade to an intermediate release before upgrading to the new Oracle Database 12c release, or you must choose an alternative upgrade method such as export/import.

See Also: ["Determine the Upgrade Path for Oracle Database"](#) on page 2-2 for information on direct and intermediate upgrade paths

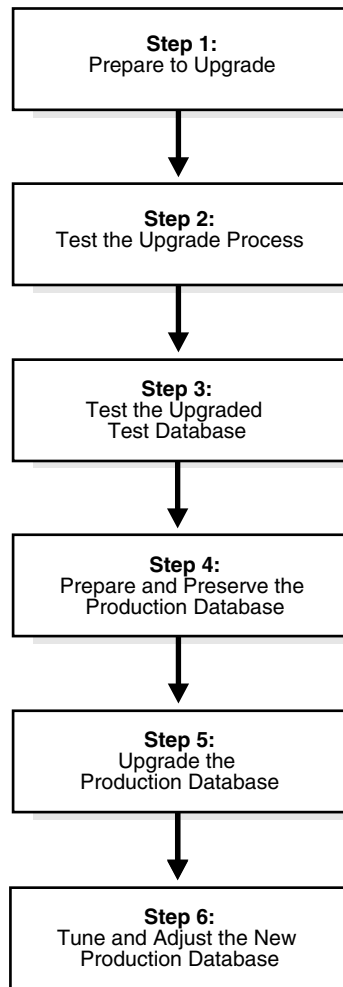
1.4 Major Steps in the Upgrade Process for Oracle Database

Detailed instructions for upgrading an existing Oracle database to the new Oracle Database 12c release are provided in the chapters in *Oracle Database Upgrade Guide*.

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

- [Step 1: Prepare to Upgrade Oracle Database](#)
- [Step 2: Test the Upgrade Process for Oracle Database](#)
- [Step 3: Test the Upgraded Test Oracle Database](#)
- [Step 4: Prepare and Preserve the Production Oracle Database](#)
- [Step 5: Upgrade the Production Oracle Database](#)
- [Step 6: Tune and Adjust the New Production Oracle Database](#)

[Figure 1–1](#) illustrates the major steps in the upgrade process.

Figure 1–1 Upgrade Steps Workflow for Oracle Database

Step 1: Prepare to Upgrade Oracle Database

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

Note: During the upgrade, consider running multiple releases of the database software so you can use the existing release as your production environment while you test the new release. See "[About Running Multiple Oracle Releases](#)" on page 9.

Step 2: Test the Upgrade Process for Oracle Database

- 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 Oracle Database

- Perform the tests that you planned in Step 1 on the test database that you upgraded to the new release of Oracle Database.
- Review the results, noting anomalies in the tests.
- Investigate ways to correct any anomalies that 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 successful and works with any required applications.

[Chapter 2, "Preparing to Upgrade Oracle Database"](#) provides more information about Steps 1 through 3.

Step 4: Prepare and Preserve the Production Oracle Database

- Prepare the current production database as appropriate to ensure that the upgrade to the new release of Oracle Database 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.

[Chapter 2, "Preparing to Upgrade Oracle Database"](#) provides more information about Step 4.

Step 5: Upgrade the Production Oracle Database

- Upgrade the production database to the new release of Oracle Database.
[Chapter 3, "Upgrading Oracle Database"](#) describes Step 5 when using DBUA or when performing a manual upgrade.
- After the upgrade, perform a full backup of the production database and perform other post-upgrade tasks.

[Chapter 4, "Post-Upgrade Tasks for Oracle Database"](#) describes the backup procedure for Step 5 after the upgrade, and other post-upgrade tasks for Step 6.

Step 6: Tune and Adjust the New Production Oracle Database

- Tune the new production database for Oracle Database 12c. The new production database should perform to the same standards, or better, than the database before the upgrade.
[Chapter 4, "Post-Upgrade Tasks for Oracle Database"](#) describes these adjustments.
- Determine which features of Oracle Database 12c to use, and update your applications accordingly.
- Develop new database administration procedures as needed.
- Do not upgrade production users to the new release until all applications have been tested and operate properly.

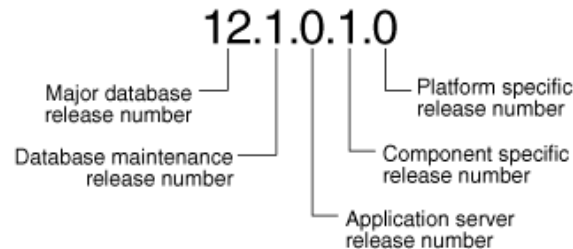
[Chapter 5, "Upgrading Applications After Upgrading Oracle Database"](#) provides information for updating applications.

1.5 About Oracle Database Release Numbers

Oracle Database Upgrade Guide describes moving between different releases of Oracle Database. The first segment in the release number indicates a major release number. Maintenance releases of Oracle Database are denoted by a change to the second segment of a release number. Patch releases are denoted by a change to the fourth segment.

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

Figure 1–2 Example of an Oracle Release Number



See Also:

- ["Convention for Referring to Release Numbers in Oracle Database Upgrade Guide"](#) on page 9
- *Oracle Database Administrator's Guide* for more information about Oracle release numbers

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

When a statement is made in *Oracle Database Upgrade Guide* about a major database release number, the statement applies to all releases within that major database release.

Similarly, when a statement is made in *Oracle Database Upgrade Guide* about a maintenance release, the statement applies to all component-specific and platform-specific releases within that maintenance release. A statement about Oracle Database 12c applies to all component-specific and platform-specific releases within this release. A statement about Oracle Database 11g Release 2 (11.2) applies to release 11.2.0.2, release 11.2.0.3, and all other platform-specific releases within Oracle Database 11g Release 2 (11.2).

1.6 About Running Multiple Oracle Releases

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

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

There cannot be multiple releases for each Oracle home. Oracle recommends that you adopt an Optimal Flexible Architecture (OFA) when creating multiple Oracle homes. See ["Optimal Flexible Architecture \(OFA\) and Oracle Database Installations"](#) on page 13.

- In order to run in production, the database server must be consistent with the major, maintenance, and patch release number. Being consistent with the major release number is not enough.

- A later release of Oracle Database server can access an earlier database, but only for upgrade purposes. For example, Oracle Database 12c database server can access an Oracle Database 10g Release 2 (10.2.0.5) database, provided that it is started up in upgrade mode.

See Also: *Oracle Database Installation Guide* for your operating system for more information about running multiple releases of Oracle Database on your operating system and any restrictions that may apply

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

- [Databases in Multiple Oracle Homes on the Same Computer](#)
- [Databases in Multiple Oracle Homes on Separate Computers](#)

1.6.1 Databases in Multiple Oracle Homes on the Same Computer

You may not be able to install earlier releases of Oracle Database on the same computer that is running Oracle Database 12c and have clients connect to the databases of the earlier releases. For example, you cannot have Oracle Database 10g, Oracle Database 11g, and Oracle Database 12c databases in multiple (or separate) Oracle homes on the same computer and have Oracle Database 10g, Oracle Database 11g, and Oracle Database 12c clients connecting to any or all of the databases on this computer. You may be able to have a combination of some releases on one system. However, be sure to obtain the latest information on compatibility and supported configurations from My Oracle Support at <http://support.oracle.com>.

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

1.6.2 Databases in Multiple Oracle Homes on Separate Computers

You can install Oracle Database 12c and Oracle Database 11g databases in Oracle homes on separate computers and Oracle Database 12c with Oracle Database 11g clients connecting to any or all of the databases. However, you must install the latest release first and install earlier releases subsequently in descending chronological order. This ensures that each installation recognizes the Oracle inventory.

1.7 Compatibility and Interoperability between Oracle Database Releases

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

The following topics discuss compatibility and interoperability:

- [What Is Oracle Database Compatibility?](#)
- [What Is Interoperability for Oracle Database Upgrades?](#)

1.7.1 What Is Oracle Database 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 earlier release.

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

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

The following topics contain more information about compatibility:

- [The COMPATIBLE Initialization Parameter in Oracle Database](#)
- [Values for the COMPATIBLE Initialization Parameter in Oracle Database](#)
- [Compatibility and Oracle Database File System \(DBFS\)](#)
- [About Downgrading and Compatibility for Upgrading Oracle Database](#)
- [How the COMPATIBLE Initialization Parameter Operates in Oracle Database](#)
- [Checking the Compatibility Level of Oracle Database](#)
- [When to Set the COMPATIBLE Initialization Parameter in Oracle Database](#)

1.7.1.1 The COMPATIBLE Initialization Parameter in Oracle Database

Oracle Database enables you to control the compatibility of your database with the COMPATIBLE initialization parameter. By default, when the COMPATIBLE initialization parameter is not set in your parameter file, it defaults to 12.0.0 for Oracle Database 12c. You cannot use the new Oracle Database 12c features, which would make your upgraded database incompatible, unless the COMPATIBLE initialization parameter is set to this value.

Note:

- Before upgrading to Oracle Database 12c, the COMPATIBLE initialization parameter must be set to at least 11.0.0, which is the minimum setting for Oracle Database 12c.
 - Oracle recommends increasing the COMPATIBLE parameter **only after you have thoroughly tested the upgraded database**.
 - After you increase the COMPATIBLE parameter, you cannot downgrade the database.
-
-

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

1.7.1.2 Values for the COMPATIBLE Initialization Parameter in Oracle Database

[Table 1–3](#) lists the default, minimum, and maximum values of the COMPATIBLE initialization parameter in Oracle Database 12c and in each release supported for upgrading to Oracle Database 12c.

Table 1–3 The COMPATIBLE Initialization Parameter

| Oracle Database Release | Default Value | Minimum Value | Maximum Value |
|--------------------------------------|---------------|---------------|---------------|
| Oracle Database 12c | 12.0.0 | 11.0.0 | 12.1.0 |
| Oracle Database 11g Release 2 (11.2) | 11.2.0 | 10.0.0 | 11.2.0 |
| Oracle Database 11g Release 1 (11.1) | 11.0.0 | 10.0.0 | 11.1.0 |
| Oracle Database 10g Release 2 (10.2) | 10.2.0 | 9.2.0. | 10.2.0 |

1.7.1.3 Compatibility and Oracle Database File System (DBFS)

Starting with Oracle Database 12c, you can access Oracle Database File System (DBFS) files and folders from Oracle XML DB Repository, under the repository path `/dbfs`. This feature provides you with FTP and HTTP(S)/WebDAV access to DBFS files and folders. DBFS file systems can be edited and managed collaboratively over the web, extending file system-like access to DBFS file systems on Windows platforms. By setting the `COMPATIBLE` initialization parameter to `12.0.0`, access to the latest enhancements to DBFS are made available.

See Also: *Oracle Database SecureFiles and Large Objects Developer's Guide* for information about using DBFS

1.7.1.4 About Downgrading and Compatibility for Upgrading Oracle Database

Before upgrading to Oracle Database 12c, the `COMPATIBLE` initialization parameter must be set to at least `11.0.0`. Only a subset of Oracle Database 12c features are available while the `COMPATIBLE` initialization parameter is set to this value.

After upgrading to Oracle Database 12c, you can set the `COMPATIBLE` initialization parameter to match the release number of the new release. Doing so enables you to use all features of the new release, but prevents you from downgrading to your earlier release.

Important: After you increase the `COMPATIBLE` parameter, the database cannot be downgraded.

See Also: [Chapter 6, "Downgrading Oracle Database to an Earlier Release"](#) for more information about downgrading

1.7.1.5 How the COMPATIBLE Initialization Parameter Operates in Oracle Database

The `COMPATIBLE` initialization parameter controls the behavior of your Oracle database. The `COMPATIBLE` initialization parameter operates in the following way:

- If you run an Oracle Database 12c database with the `COMPATIBLE` initialization parameter set to `11.0.0`, then it generates database structures on disk that are compatible with Oracle Database 11g. 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 occurs. However, any new features that do not make incompatible changes on disk are enabled.
- If the database becomes incompatible with the `COMPATIBLE` initialization parameter 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

1.7.1.6 Checking the Compatibility Level of Oracle Database

The compatibility level of your database corresponds to the 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 FROM v$parameter
        WHERE name = 'compatible';
```

1.7.1.7 When to Set the `COMPATIBLE` Initialization Parameter in Oracle Database

Oracle recommends increasing the `COMPATIBLE` parameter only after complete testing of the upgraded database has been performed. After the upgrade is complete, you can increase the setting of the `COMPATIBLE` initialization parameter to the maximum level for Oracle Database 12c. However, after you increase the `COMPATIBLE` parameter, you cannot subsequently downgrade the database.

1.7.2 What Is Interoperability for Oracle Database Upgrades?

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

Interoperability across disparate operating systems and operating system versions might be a problem (especially during rolling upgrades) because the minimum requirements for the new Oracle Database 12c might require you to upgrade the operating systems on some or all of your hosts. Therefore, you must check for compatibilities between drivers, network, and storage for all the interim states of the system during the rolling upgrade.

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

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

1.8 Optimal Flexible Architecture (OFA) and Oracle Database Installations

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

OFA provides the following benefits:

- Organizes large amounts of complicated software and data on disk to avoid device bottlenecks and poor performance
- Facilitates routine administrative tasks, such as software and data backup functions, which are often vulnerable to data corruption
- Alleviates switching among multiple Oracle databases
- Adequately manages and administers database growth
- Helps 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 managing datafiles and tempfiles

1.9 Converting 32-bit Oracle Database to 64-bit Oracle Database

If you are installing 64-bit Oracle Database software but were previously using a 32-bit Oracle Database installation, the database is automatically converted to 64-bit during the upgrade to Oracle Database 12c.

1.10 About Rolling Upgrades for Oracle ASM and Oracle RAC Databases

Rolling upgrades work for Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM) (collectively known as Oracle Grid Infrastructure), but not for the Oracle Database software. You cannot upgrade to a new major release, maintenance release, or patch release of Oracle Database using the rolling upgrade process in an Oracle Real Application Clusters (Oracle RAC) environment. You can, however, apply most database patches (sometimes referred to as one-off patches) and all database patch set updates and critical patch updates (PSUs and CPUs) using a rolling patch install in an Oracle RAC environment.

See Also: ["Summary of Methods for Performing Rolling Upgrades for Oracle Database"](#) on page 14

1.10.1 Summary of Methods for Performing Rolling Upgrades for Oracle Database

[Table 1–4](#) summarizes the various methods for performing rolling upgrades and provides references to the appropriate documentation.

Table 1–4 Methods for Performing Rolling Upgrades

| Method | Description | Reference |
|---|---|--|
| Oracle Enterprise Manager Cloud Control | Starting with Oracle Database 12c, Oracle provides Cloud Control support for performing a database rolling upgrade. | Online help in Oracle Enterprise Manager Cloud Control |
| Oracle Data Guard SQL Apply | Use SQL Apply and logical standby databases to upgrade Oracle Database software and patchsets. | "Requirement for Upgrading Oracle Databases That Use Oracle Database Vault" on page 2-21 |
| Oracle Data Guard Physical Standby Database | Use an existing physical standby database to perform a database rolling upgrade by temporarily converting it to a logical standby database. | "About Upgrading Using Standby Databases" on page 2-18 |
| Oracle Clusterware and Oracle Universal Installer (OUI) | Use OUI and Oracle Clusterware to perform a rolling upgrade to apply patchset releases of Oracle Clusterware. This method enables some instances of Oracle RAC to remain available during the patchset upgrades. Only the node that is currently being patched must be stopped; the other instances remain available. OUI enables you to apply the patchset successively to the different instances of the cluster. | <i>Oracle Grid Infrastructure Installation Guide</i> for your operating system and your operating system-specific Oracle Clusterware installation guide |
| Oracle Automatic Storage Management (Oracle ASM) | Use Oracle ASM to independently upgrade or patch clustered Oracle ASM instances. This method enables all of the features of a clustered Oracle ASM environment to continue to function even while one or more Oracle ASM instances run different software releases. | <i>Oracle Grid Infrastructure Installation Guide</i> for your operating system "Using Oracle ASM Rolling Upgrade" in <i>Oracle Automatic Storage Management Administrator's Guide</i> |

1.11 Moving From Standard Edition to Enterprise Edition of Oracle Database

If you have a Standard Edition database at a release earlier than the new release of Oracle Database, you must first upgrade as described in [Chapter 3, "Upgrading Oracle Database."](#) Then you can move to an Oracle Enterprise Edition database by installing Oracle Enterprise Edition software. To move from one edition to another, you must use the same release of Oracle Database for both Enterprise Edition and Standard Edition. For example, both releases must be release 12.1 (or another release level), but must be the same for both editions in order to perform the change.

Caution: Performing this procedure deinstalls the Oracle Standard Edition software and results in deleting database files that exist under the Oracle home. Therefore, back up database files under the current Oracle home.

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

1. Ensure that the release number of your Oracle Standard Edition server software is the same release as your Oracle Enterprise Edition server software.
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. See *Oracle Database Platform Guide for Microsoft Windows* for information about stopping Oracle Database services.

4. Back up all database files under the current Oracle home that you must keep. See *Oracle Database Backup and Recovery User's Guide* for information about backing up the database.
5. Deinstall the Standard Edition server software. (**This step deletes all existing database files that reside under the Oracle home.**)

Run the deinstallation tool from the Oracle home. The deinstallation tool is available as a separate command (`deinstall`) under the Oracle home directory after installation. It is located under `ORACLE_HOME\deinstall`.

To deinstall an Oracle home on Windows, use the following syntax:

```
setup.exe -deinstall -home path_of_oracle_home_to_be_deinstalled
```

To deinstall an Oracle home on Linux and UNIX, use the following syntax:

```
$ ./runInstaller -deinstall -home path_of_oracle_home_to_be_deinstalled
```

Note: Starting with Oracle Database 12c, the deinstallation tool is integrated with the database installation media. You can run the deinstallation tool using `runInstaller` on Linux and UNIX or `setup.exe` on Windows with the `-deinstall` and `-home` options from the base directory of the Oracle Database, Oracle Database Client, or Oracle Grid Infrastructure installation media. See *Oracle Database Installation Guide* for your operating system.

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

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

7. If you have an existing database, then point your `ORACLE_SID` to this pre-existing database.

If your existing database is on Windows, then you must re-create the DB service using ORADIM.

See Also: *Oracle Database Platform Guide for Microsoft Windows* for information about Oracle services on Windows

8. Start up your database.
9. Run the `catalog.sql` and `catproc.sql` scripts to ensure that all Enterprise Edition objects are created.

Your database is now converted to Oracle Database Enterprise Edition. You are ready to upgrade the database to Oracle Database 12c. Proceed to [Chapter 2, "Preparing to Upgrade Oracle Database,"](#) to continue with the upgrade process.

See Also: Note ID 117048.1, "How to Convert Oracle Database from Standard to Enterprise Edition" on My Oracle Support at <http://support.oracle.com>

1.12 Moving from Enterprise Edition to Standard Edition of Oracle Database

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

Enterprise Edition contains data dictionary objects that are not available in Standard Edition. The Export/Import operation does not introduce data dictionary objects specific to the Enterprise Edition, because the *SYS* schema objects are not exported. After the Import in the Standard Edition database, you are only required to drop user schemas related to Enterprise Edition features.

See Also: [Chapter 7, "Migrating Data Using Oracle Data Pump"](#)

1.13 Moving from Oracle Database Express Edition (Oracle Database XE) to Oracle Database

Oracle Database Express Edition (Oracle Database XE) is an entry-level edition of Oracle Database that is quick to download, simple to install and manage. Oracle provides Oracle Database XE free, with certain licensing restrictions, for you to develop, deploy, and distribute. Oracle Database XE makes it easy to upgrade to the other editions of Oracle without costly and complex migrations. Oracle Database XE can be installed on any size computer with any number of CPUs, with one database per computer. Support is provided through a free Oracle online forum monitored by Oracle employees and community experts.

To upgrade Oracle Database 11g Express Edition (Oracle Database XE) to Oracle Database 12c, you must install the new Oracle Database software on the same system as Oracle Database XE and use DBUA to perform the upgrade.

1. Install release 11202 Oracle Database XE with root user.

```
/u01/app/oracle/product/11.2.0/xe
```

2. Start SQL Plus and perform the following steps to avoid prerequisite checks warnings and to ensure successful database upgrade during DBUA interactive phase.

```
Set ORACLE_HOME
Set ORACLE_SID
SQL> alter system set job_queue_processes=64;
SQL> alter database datafile '/u01/app/oracle/oradata/XE/system.dbf'
autoextend on next 640k maxsize unlimited;
```

3. As root user, in the release 11.2.0.2 Oracle Database XE home, give 777 permissions to the following directories, subdirectories, and files before upgrade.

```
cd /u01/app/oracle
$ls
admin diag fast_recovery_area oradata product
```

For the admin directory give 777 permissions to the following directories and files.

```
chmod 777 admin
cd admin
chmod 777 XE
```

```

chmod 777 *
$ls
adump dba dpdump pfile

cd dpdump
chmod 777 dp.log

cd ..
cd adump
chmod 777 * ( Change permissions of *.aud)

```

For the diag directory give 777 permissions as follows. Run this from the Oracle home; then change to diag.

```

cd diag
chmod -R 777 *

```

For fast_recovery_area directory give 777 permissions to the following directories and files. The fast_recovery_area folder is under Oracle home. Change back to the Oracle home first, then run the following commands.

```

chmod 777 fast_recovery_area
cd fast_recovery_area
chmod 777 XE
cd XE
chmod 777 onlineolog

```

For oradata give 777 permissions to the following directory. The oradata folder is under Oracle home. Change back to the Oracle home first, then run the following command.

```

chmod -R 777 oradata

```

For the Oracle product directory give 777 permissions to the following directories and files.

```

chmod 777 11.2.0
cd 11.2.0
chmod 777 xe
cd xe
chmod 777 dba
cd dba
chmod 777 * ( changes permissions of these files hc_XE.dat init.ora lkXE
@ orapwXE spfileXE.ora)

```

4. Under the directory where you installed Oracle Database 12c, create the log directory (for example: /scratch/logdir , and give 777 permissions to logdir) where the log files will be written during database upgrade.

Also create the backup folder (for example: /scratch/backup, and give 777 permissions to the backup folder) where backup files will be stored.

5. Start DBUA with following parameter. See "[Upgrading with Oracle Database Upgrade Assistant \(DBUA\)](#)" on page 3-6 for more information about running DBUA.

```

./dbua -logDir /scratch/logdir -backupLocation /scratch/backup

```

When prompted, create the new listener with a new name and port because the upgrade cannot use the existing listener since it was created by root user.

6. Provide the required inputs during the DBUA interactive phase and click **Finish** when complete.

See Also:

- *Oracle Database Installation Guide* for your operating system
- Oracle online forum at: <http://forums.oracle.com> under "Oracle Database Express Edition (XE)"
- [Chapter 3, "Upgrading Oracle Database"](#)

Preparing to Upgrade Oracle Database

Before you upgrade Oracle Database, you must understand the requirements and recommendations. This chapter provides information and procedures for the pre-upgrade tasks, including installing the new Oracle software for the upgrade.

This chapter contains the following topics:

- [Tasks for Preparing to Upgrade Oracle Database](#)
- [Requirements for Upgrading Oracle Database](#)
- [Installing the New Oracle Database Software](#)
- [Patch Set Updates and Requirements for Upgrading Oracle Database](#)
- [About the Pre-Upgrade Information Tool for Oracle Database](#)
- [Backing Up Oracle Database for Upgrading](#)
- [Preparing the New Oracle Home for Upgrading](#)
- [Recommendations for Oracle Net Services When Upgrading Oracle Database](#)
- [Testing the Upgrade Process for Oracle Database](#)
- [Testing an Upgraded Test Copy of Oracle Database](#)

See Also: "Overview of Oracle Database Upgrade Tools and Processes" on page 1-1

2.1 Tasks for Preparing to Upgrade Oracle Database

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

Complete the following tasks to prepare to upgrade your database:

- [Become Familiar with New Oracle Database Features](#)
- [Determine the Upgrade Path for Oracle Database](#)
- [Choose an Upgrade Method for Oracle Database](#)
- [Choose a New Location for Oracle Home when Upgrading](#)
- [Develop a Test Plan for Upgrading Oracle Database](#)
- [Prepare a Backup Strategy before Upgrading Oracle Database](#)

2.1.1 Become Familiar with New Oracle Database Features

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

Note: Oracle Database training classes are an excellent way to learn how to take full advantage of the features and functions available with Oracle Database. More information can be found at <http://education.oracle.com/>

See Also: *Oracle Database New Features Guide* and "New Features" on page 3-xvii

2.1.2 Determine the Upgrade Path for Oracle Database

The path that you must take to upgrade to the latest release of Oracle Database 12c depends on the release number of your current database. You may perform a direct upgrade to the new release if your current Oracle Database is release 10.2.0.5, 11.1.0.7, or release 11.2.0.2 or later. You may directly upgrade from release 12.1.0.1 to 12.1.0.2.

If your current Oracle Database is a release earlier than 10.2.0.5, or is release 11.2.0.1, then directly upgrading your current release of Oracle Database to the latest release is not supported. In this case, you are required to upgrade to an intermediate release before upgrading to Oracle Database 12c.

For example, if the database from which you are upgrading is running Oracle Database 9i, then follow these steps:

1. Upgrade release 9.0.1.4 to release 10.2.0.5 using the instructions in *Oracle Database Upgrade Guide 10g Release 2 (10.2)*.
2. Upgrade Oracle Database 10g Release 2 (10.2.0.5) directly to the new release of Oracle Database 12c using the instructions in *Oracle Database Upgrade Guide*, which is this book.

[Table 2–1](#) contains the required upgrade path for each release of Oracle Database. Use the upgrade path and the specified documentation to perform an intermediate upgrade of your database before fully upgrading to Oracle Database 12c.

Table 2–1 Examples of Upgrade Paths for Oracle Database 12c

| If your Current Release is: | Then the Direct or Intermediate Upgrade Path is: |
|--|--|
| 12.1.0.1, 11.2.0.2 or later 11.1.0.7 10.2.0.5 | Direct upgrade is supported. Perform the upgrade using the current <i>Oracle Database Upgrade Guide</i> , which is this guide. |
| 11.2.0.1 11.1.0.6 10.2.0.2, 10.2.0.3, and 10.2.0.4 10.1.0.5 9.2.0.8 or earlier | <p>Direct upgrade to Oracle Database 12c is not supported.</p> <p>Solution: Upgrade to an intermediate Oracle Database release (that can be directly upgraded to release 12.1) before you can upgrade to the new Oracle Database 12c. This does not include upgrading using Oracle Data Pump export/import. When upgrading to an intermediate Oracle Database release, follow the instructions in the intermediate release's documentation. Then, upgrade the intermediate release database to the new Oracle Database 12c release using the instructions in Chapter 3, "Upgrading Oracle Database".</p> <p>For example:</p> <ul style="list-style-type: none"> ■ If you are upgrading from 11.2.0.1 or 11.1.0.6, then you must first upgrade to Oracle Database 11g Release 2 (11.2.0.2). ■ If you are upgrading from 10.2.0.2, 10.2.0.3, 10.2.0.4, or 10.1.0.5, then you must first upgrade to 10.2.0.5 or later. ■ For release 9.2.0.8, you must first upgrade to an intermediate Oracle Database release, as follows: 9.2.0.8 -> 11.2.0.2 or 11.2.0.3 -> 12.1 |

See Also: ["Supported Releases for Downgrading Oracle Database"](#) on page 6-1 for information related to downgrading your database

2.1.3 Choose an Upgrade Method for Oracle Database

Oracle recommends that you use Database Upgrade Assistant (DBUA) as the best method to upgrade your database. However, Oracle offers several methods to fully upgrade your database that support the complexities of your enterprise. See ["Overview of Oracle Database Upgrade Tools and Processes"](#) on page 1-1 for the definition and description of the differences between *upgrading* and *migrating* data.

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

- [The Automated \(DBUA\) Method for Upgrading Oracle Database](#)
- [The Manual, Command-line Method for Upgrading Oracle Database](#)
- [The Export/Import Method for Migrating Data When Upgrading Oracle Database](#)

2.1.3.1 The Automated (DBUA) Method for Upgrading Oracle Database

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

In an Oracle Real Application Clusters (Oracle RAC) environment, DBUA does not upgrade all the databases per se; DBUA upgrades all the database files and configuration files on all nodes in the cluster.

2.1.3.2 The Manual, Command-line Method for Upgrading Oracle Database

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

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

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

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

The Pre-Upgrade Information Tool displays warnings about possible upgrade issues with the database, and generates fixup scripts for you to use to address any issues. It also displays information about required initialization parameters for the new release of Oracle Database.

See Also: ["Pre-Upgrade Information Tool Warnings and Recommendations for Oracle Database"](#) on page 29

- Prepare the new Oracle home.

See Also: ["Choose a New Location for Oracle Home when Upgrading"](#) on page 6 for more information

- Perform a backup of the database.

Depending on the release of the database being upgraded, you might be required to perform additional pre-upgrade steps (adjust the parameter file for the upgrade, remove desupported initialization parameters and adjust initialization parameters that might cause upgrade problems).

2.1.3.2.2 After the Upgrade Review the upgrade spool log file and use the Post-Upgrade Status Tool, `utlu121s.sql`. The Post-Upgrade Status Tool is a SQL script that is included with Oracle Database. You run the Post-Upgrade Status Tool in the environment of the new release. You can run the Post-Upgrade Status Tool any time after upgrading the database.

See Also: ["Manually Upgrading a Multitenant Container Oracle Database \(CDB\)"](#) on page 3-35

2.1.3.3 The Export/Import Method for Migrating Data When Upgrading Oracle Database

Unlike DBUA or a manual command-line upgrade, the Oracle Data Pump Export and Import utilities physically migrate a copy of data from your current database to a new database in the new release. Data Pump Export and Import are recommended for higher performance and to ensure support for new datatypes.

The new database must exist, or you must have created the database in the new Oracle home, before the contents of the export dump file(s) can be loaded.

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

Note:

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

2.1.3.3.1 The Effects of Export/Import on Upgraded Oracle Databases The Export/Import data migration 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 newly upgraded Oracle Database 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 newly upgraded Oracle database.

Most importantly, the Export/Import operation results in a completely new database. Although the current target database ultimately contains a copy of the specified data that you migrated, the upgraded database might perform differently from the original source 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 issues.

2.1.3.3.2 Export/Import Benefits for Migrating Data for Oracle Database Migrating data when upgrading Oracle Database using Export/Import has benefits, including compressing and restructuring the database for increased performance.

Using Export/Import to migrate data:

- 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 releases of Oracle Database because an entirely new database is created.
- Enables the copying of specified database objects or users. Importing only the objects, users, and other items you need is useful for establishing a test environment for the new software on only a subset of the production data. Data Pump Export/Import provides flexible data-subsetting capabilities.
- Serves as a backup archive. You can use a full database export as an archive of the current database.
- Enables the upgraded database to be established on an operating system or hardware platform that is different from that which is supporting the database being upgraded.

- Network-based Data Pump Import allows the new Oracle database to be directly loaded across the network from the old database being upgraded. Thus, no intervening dump files are required.

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

See Also: [Chapter 7, "Migrating Data Using Oracle Data Pump"](#)

2.1.4 Choose a New Location for Oracle Home when Upgrading

You must choose a location for Oracle home for the new release of Oracle Database that is separate from the Oracle home of your current release. You cannot install the new software into the same location for Oracle home as your current release.

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

2.1.5 Develop a Test Plan for Upgrading Oracle Database

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

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

Your test plan must include these types of tests:

- [Upgrade Testing](#)
- [Minimal Testing](#)
- [Functional Testing](#)
- [High Availability Testing](#)
- [Integration Testing](#)
- [Performance Testing](#)
- [Volume and Load Stress Testing for Oracle Database Upgrades](#)

2.1.5.1 Upgrade Testing

Upgrade testing for Oracle Database entails planning and testing the upgrade path from your current software to Oracle Database 12c, 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.

2.1.5.2 Minimal Testing

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

2.1.5.3 Functional Testing

Functional testing for Oracle Database 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.

2.1.5.4 High Availability Testing

High Availability testing for Oracle Database ensures that the upgraded database system must meet: Recovery Time Objective (RTO) and Recovery Point Objective (RPO) business requirements.

- In an Oracle RAC environment, injecting node or instance failures during stress testing helps to evaluate if the Oracle RAC recovery capability has changed.
- To minimize downtime, Oracle recommends that you test your fallback plans and procedures.
- To ensure that the upgrade process runs within the time that you have allocated, Oracle recommends that you confirm database performance and stability, and resolve performance problems.

See Also: The Upgrade Companion, Note 1462240.1 on My Oracle Support at <http://support.oracle.com/>

2.1.5.5 Integration Testing

Integration testing for Oracle Database examines the interactions among components of the system.

Consider the following factors when you plan your integration testing:

- Pro*C/C++ applications running against Oracle Database 12c instances should be tested to ensure that there are no problems with the new software. See *Pro*C/C++ Programmer's Guide* for more information about Pro*C/C++ applications.
- Graphical user interfaces should be tested with other components.
- Subtle changes in Oracle Database 12c, 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 12c instance.
- If the connection between two components involves Oracle Net or Oracle Net Services, then those connections should also be tested and stress tested. See *Oracle Database Net Services Reference* for more information about upgrade recommendations for Oracle Net Services.

2.1.5.6 Performance Testing

Performance testing for Oracle Database 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 that the application makes to the database server.

This section describes the following types of performance testing:

- [Database Replay](#)
- [SQL Performance Analyzer](#)
- [SQL Plan Management](#)

2.1.5.6.1 Database Replay You can use the 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. Database Replay also provides analysis and reporting to highlight potential problems; for example, errors encountered, divergence in performance, and so forth. In addition, all the regular Enterprise Manager performance monitoring and reporting tools such as Automatic Database Diagnostic Monitor, Automatic Workload Repository (AWR), and Active Session History are available to address any problems.

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

See Also:

- *Oracle Database Testing Guide* for information about how to capture and replay workloads
- *Oracle Database Performance Tuning Guide* for more information on the Automatic Workload Repository

2.1.5.6.2 SQL Performance Analyzer You can use the SQL Performance Analyzer to forecast the impact of system changes on a SQL workload. SQL Performance Analyzer enables evaluating the impact of a change such as database upgrade by identifying the SQL statements impacted by the upgrade and measuring their performance divergence. The analysis enables you to assess the overall effect of the upgrade on SQL performance and makes it possible to avoid any negative outcome before users can be impacted.

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

2.1.5.6.3 SQL Plan Management A database upgrade that installs a new optimizer version usually results in plan changes for a small percentage of SQL statements, with most of the plan changes resulting in either no performance change or improvement. However, certain plan changes may cause performance regressions. SQL plan management prevents performance regressions resulting from sudden changes to the execution plan of a SQL statement by providing components for capturing, selecting,

and evolving SQL plan information. 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 the performance of corresponding SQL statements, regardless of changes occurring in the system.

With SQL plan management, the optimizer automatically manages execution plans and ensures that only known or verified plans are used. When a new plan is found for a SQL statement, the plan is not used until it has been verified by the database to have comparable or better performance than the current plan. Therefore, if you seed SQL plan management with your current execution plan, which is to become the SQL plan baseline for each statement, then the optimizer uses these plans after the upgrade. If the Oracle Database 12c optimizer determines that a different plan is necessary, then the new plan is queued for verification and is not used until it has been confirmed to have comparable or better performance than the current plan.

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

- [Bulk Load a SQL Management Base from the Cursor Cache](#)
- [Bulk Load a SQL Management Base with a SQL Tuning Set \(STS\)](#)
- [Unpack Existing SQL Plan Baselines from a Staging Table](#)

Bulk Load a SQL Management Base from the Cursor Cache

Bulk loading of execution plans or SQL plan baselines from the cursor cache is useful when upgrading a database from Oracle Database 11g to the latest release of Oracle Database. The cursor cache is a shared SQL area, and SQL plans that are bulk loaded are automatically accepted and added to existing or new plan histories as SQL plan baselines.

1. In the source release of Oracle Database, use the `DBMS_SPM.LOAD_PLAN_FROM_CURSOR_CACHE` procedure or Oracle Enterprise Manager to load all of the execution plans in the cursor cache into the SQL Management Base.
2. Upgrade the database.

See Also: *Oracle Database SQL Tuning Guide* for more information on how to load plans from the shared SQL area using PL/SQL or Oracle Enterprise Manager

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

Bulk loading of execution plans or SQL plan baselines may be done with a SQL Tuning Set. This is useful when upgrading from Oracle Database 10g, where no SQL Management Base (SMB) exists to directly load from the cursor cache, or to load historic plans from the Automatic Workload Repository.

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

See Also: *Oracle Database SQL Tuning Guide* for the complete procedure for bulk loading execution plans or SQL plan baselines

Unpack Existing SQL Plan Baselines from a Staging Table

You can test and tune all of your critical SQL queries on an Oracle Database test environment and then move those SQL execution plans to your Oracle Database production environment. Alternatively, you can take plans for SQL queries from your pre-upgrade Oracle Database production environment and move them to your post-upgrade production environment.

1. On the Oracle Database 12c test system, after completing all testing and tuning, use the `DBMS_SPM.LOAD_PLAN_FROM_CURSOR_CACHE` procedure or Enterprise Manager to load all of the execution plans in the cursor cache into the SQL Management Base.
2. Create a staging table using the `DBMS_SPM.CREATE_STGTAB_BASELINE` procedure.
3. Pack the SQL plan baselines you created in step 1 into the staging table using the `DBMS_SPM.PACK_STGTAB_BASELINE` function.
4. Export the staging table into a flat file using Data Pump.
5. Transfer this flat file to the target system.
6. Import the staging table from the flat file using Data Pump.
7. Unpack the SQL plan baselines from the staging table into the SQL Management Base on the target system using the `DBMS_SPM.UNPACK_STGTAB_BASELINE` function.

See Also:

- *Oracle Database SQL Tuning Guide* for more information about loading plans from a staging table
- *Oracle Database Utilities* for information about using Data Pump

2.1.5.7 Volume and Load Stress Testing for Oracle Database Upgrades

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

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

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

See Also: *Oracle Database Testing Guide* for information about using Database Replay for stress testing

2.1.6 Prepare a Backup Strategy before Upgrading Oracle Database

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

To develop a backup strategy, consider the following questions:

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

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

Tip: *Oracle Database Backup and Recovery User's Guide* for information about implementing backup strategies using RMAN

2.2 Requirements for Upgrading Oracle Database

Depending on the operating system and Oracle components in your database environment, there is specific information and requirements to be aware of.

The following topics discuss system recommendations and requirements for performing an upgrade:

- [Relocating Existing Data Files to the New Oracle Database Environment](#)
- [About Upgrading PL/SQL Packages That Are Not Installed by Default](#)
- [Saving Oracle Enterprise Manager Database Control Configuration and Data](#)
- [Manually Removing DB Control with emremove.sql](#)
- [About Upgrading an Oracle ASM Instance](#)
- [About Upgrading Oracle Grid Infrastructure and Oracle Clusterware](#)
- [About Upgrading with Read-Only and Offline Tablespaces](#)
- [About Upgrading Using Standby Databases](#)
- [About Upgrading Your Operating System](#)
- [Transporting Data to a Different Operating System](#)
- [About Oracle GoldenGate and Online Database Upgrade](#)
- [Requirement for Upgrading Oracle Databases That Use Oracle Database Vault](#)
- [About Upgrading Oracle OLAP Data Security Policies](#)
- [Requirements for Upgrading Databases That Use Oracle Label Security and Oracle Database Vault](#)
- [Requirement for Upgrading Databases that Use Oracle Warehouse Builder \(OWB\)](#)
- [Removing the Unified Auditing Schema and Roles](#)

See Also:

- *Oracle Database Net Services Reference* for information about upgrade recommendations for Oracle Net Services
- *Oracle Database Installation Guide* for your operating system-specific information about preparing to upgrade
- ["About Rolling Upgrades for Oracle ASM and Oracle RAC Databases"](#) on page 1-14 for information about rolling upgrades

2.2.1 Relocating Existing Data Files to the New Oracle Database Environment

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

To relocate data files to the new Oracle Database environment:

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

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

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

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

See Also: ["Update Access Control Lists and Network Utility Packages"](#) on page 2-30

2.2.3 Saving Oracle Enterprise Manager Database Control Configuration and Data

If you need to be able to downgrade and then restore Oracle Enterprise Manager Database Control (DB Control) after upgrading to the new release, then you must save your configuration files and data *before* upgrading your database. Starting with Oracle Database 12c, DB Control is removed as part of the upgrade process. Oracle provides the `emdwgrd` utility to use before upgrading your database to keep a copy of your DB Control configuration and data. This is necessary if you want to be able to downgrade and restore the DB Control configuration from your earlier database.

The `emdwgrd` utility resides in the `ORACLE_HOME/bin` directory in the new Oracle Database 12c release. The `emdwgrd` utility consists of `emdwgrd` and `emdwgrd.pl` for Linux and UNIX, and `emdwgrd.bat` and `emdwgrd.pl` for Windows. Before running the utility, you must install the software for Oracle Database 12c, and then invoke the script from the new Oracle home. The `emdwgrd` utility requires that you set `ORACLE_HOME` to the Oracle home of the release you are upgrading.

To save DB Control files and data using `emdwgrd`:

1. Install the software for the new Oracle Database 12c release.

See Also: ["Installing the New Oracle Database Software"](#) on page 2-25

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 of the database being upgraded.
5. Go to the Oracle home of the new Oracle Database 12c release.
6. Execute `emdwgrd` as follows:
 - For a single-instance database, run the following command, where `old_SID` is the SID of the database being upgraded and `save_directory` is the path to the storage place you have chosen for your database control files and data:

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

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

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

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

Then, execute `emdwgrd` as follows:

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

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

7. Enter the `SYS` password for the database to be upgraded.

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

2.2.4 Manually Removing DB Control with `emremove.sql`

To minimize downtime during the upgrade process, you can optionally run the `emremove.sql` script as part of your pre-upgrade preparation. The `emremove.sql` script drops the Oracle Enterprise Manager-related schemas and objects. This script might take a few minutes to complete as it has six phases in which to complete the process. The script may take longer if you have `SYSMAN` and related sessions active from `SQL*Plus` or Oracle Enterprise Manager or other clients.

Important: You must back up your DB Control configuration and data if you want to downgrade and restore DB Control as described in ["Saving Oracle Enterprise Manager Database Control Configuration and Data"](#) on page 2-12.

To run `emremove.sql` to manually remove DB Control:

1. Shut down the DB Control application immediately.

```
$ emctl stop dbconsole
```
2. Start SQL*Plus and connect to the database using the SYS account AS SYSDBA.
3. Optional step. Do not set this if you want to run the script in silent mode. To view the progress of `emremove.sql` while it is running, set the following variables:

```
SET ECHO ON;
SET SERVEROUTPUT ON;
```
4. Execute `emremove.sql`. The script is located under `ORACLE_HOME/rdbms/admin/` of the new Oracle Database 12c Oracle home.

```
$ @emremove.sql
```
5. After `emremove.sql` completes, you must manually remove `ORACLE_HOME/HOSTNAME_SID` and `ORACLE_HOME/oc4j/j2ee/OC4J_DBConsole_HOSTNAME_SID` directories from your file system.

Note: If the DB Control had been upgraded from release 10.2.0.3 to 10.2.0.4, then the following directories also need to be removed from the file system:

```
ORACLE_HOME/HOSTNAME_SID.upgrade
```

```
ORACLE_HOME/oc4j/j2ee/OC4J_DBConsole_HOSTNAME_SID.upgrade
```

6. On Windows platforms, also delete the DB Console service, generally with the name `OracleDBConsoleSID`.

2.2.5 About Upgrading an Oracle ASM Instance

The recommended practice is to upgrade an Oracle ASM instance with Oracle Grid Infrastructure Oracle Universal Installer (OUI). OUI automatically defaults to upgrade mode when it detects an Oracle ASM instance at an earlier release level. You can also perform a rolling upgrade to clustered Oracle ASM instances in environments running Oracle Database 11g or later. See your *Oracle Grid Infrastructure Installation Guide* for step-by-step instructions about performing a rolling upgrade of Oracle ASM.

See Also: *Oracle Grid Infrastructure Installation Guide* for information on rolling upgrades for Oracle Clusterware or Oracle Automatic Storage Management

2.2.5.1 Determining If Oracle ASM is Included in the Current Database Configuration

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

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

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

2.2.6 About Upgrading Oracle Grid Infrastructure and Oracle Clusterware

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

See Also: *Oracle Grid Infrastructure Installation Guide* for your operating system and go to the appendix for complete information on how to perform Oracle Clusterware and Oracle Automatic Storage Management upgrades

The following recommendations pertain to upgrading Oracle Clusterware, Oracle Grid Infrastructure, and Oracle ASM instances:

- To upgrade Oracle Database, you must install the Oracle Grid Infrastructure software into a new Grid home, and install Oracle Database 12c software into the new Oracle home. This also applies to Oracle Grid Infrastructure for a standalone server (Oracle Restart). You can upgrade from supported releases only. See ["Determine the Upgrade Path for Oracle Database"](#) on page 2-2.
- On the Windows x64 platform, to upgrade Oracle Clusterware from releases 10.2.0.5 and 11.1.0.7 to release 12.1, you must perform an interim upgrade to 11.2.0.3 for Oracle Clusterware. After upgrading Oracle Clusterware to release 11.2.0.3, you must move the Oracle Cluster Registry (OCR) and voting files (VDSK) to Oracle ASM. You can then upgrade Oracle Clusterware release 11.2.0.3 to release 12.1.
- If you are upgrading an Oracle Database release 10.2.0.5 or release 11.1.0.7 environment that stores Oracle Clusterware files on OCFS on Windows or RAW devices, then you cannot directly upgrade to Oracle Database 12c. See ["About Upgrading Oracle Database Release 10.2 or 11.1 and OCFS and RAW Devices"](#) on page 8-7.
- For Oracle Database release 10.2.0.5, either all Oracle software installations were owned by the Oracle user, typically `oracle`, or Oracle Database software was owned by the user `oracle`, and Oracle Clusterware software was owned by a separate user, typically `crsuser`.
- Starting with Oracle Database release 11.1, the user account that is designated as *owner* of the release 10.2.0.5 Oracle Clusterware software must perform the Oracle Clusterware release 11.1.0.7 upgrade. The user account that is performing this upgrade must also be the user that owns the ASM home of the earlier release. If the earlier ASM home has a different owner, then the *owner* account must be changed before performing the upgrade.
- For Oracle Database releases 11.1.0.7 and 10.2.0.5, if your configuration does not include Oracle ASM, then you must shut down the cluster synchronization services (CSS) daemon and delete the CSS service from the system by running the `localconfig` command with the `delete` option. For example:


```
ORACLE_HOME/bin/localconfig delete
```
- When upgrading Oracle Clusterware, OUI automatically calls Oracle ASM Configuration Assistant (ASMCA) to perform the upgrade into the Oracle Grid Infrastructure home. The Grid Infrastructure home can be local to each node.

- For a single-instance configuration, Oracle ASM and Oracle Restart run from the Oracle Grid Infrastructure home. Thus, Oracle ASM and Oracle Restart are upgraded to Oracle Grid Infrastructure 12c at the same time.
- You must ensure that the database compatibility attribute for Oracle ASM disk groups matches the compatibility parameter that is set in `init.ora`.

See Also: *Oracle Grid Infrastructure Installation Guide* for your operating system

2.2.6.1 Oracle Clusterware Must Be Running on Local Node When Upgrading

Starting with Oracle Database 12c, non-rolling upgrades for Oracle Real Application Clusters require Oracle Clusterware on the local node to be up and running. The Oracle Clusterware stack on the local node must be running.

With earlier releases of Oracle Database, *non-rolling* for the cluster meant that OUI set `ISROLLING` to `false` for all stack shutdown before the upgrade process started. If any of the cluster nodes were up, then `ISROLLING` was set to `true`. However, in Oracle Database 12c, *rolling* means that the local stack is up before upgrade and the other stacks must be shut down. Only in this circumstance OUI sets `ISROLLING` to `false`.

To start Oracle Clusterware on the local node:

1. Run the OUI installer on the node where the Oracle Clusterware stack is running or open a terminal.
2. Log in as root or Administrator.
3. Change directory (`cd`) to the Oracle Clusterware home (Grid home), and start Oracle Clusterware by entering the following command:

```
# crsctl start crs
```

2.2.6.2 About Upgrading an Oracle Real Application Clusters (Oracle RAC) Database with DBUA

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

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

See Also: *Oracle Grid Infrastructure Installation Guide* for your operating system

2.2.6.3 About Upgrading and Inaccessible Nodes

If you are using the force cluster upgrade command as documented in the upgrade appendix of *Oracle Grid Infrastructure Installation Guide*, then the information in this section applies.

Starting with Oracle Database 12c, there is a an option to join the inaccessible nodes as an alternative to deleting the nodes, which was required in earlier releases. The new Oracle Database 12c software must already be installed on the nodes.

The following command can be run on the inaccessible or unreachable nodes to upgrade and join them into the cluster.


```
rootcrs.pl -join -existingNode upgraded_node
```

In this command, the `upgraded_node` variable refers to one node in the cluster that has already been upgraded. The `-existingNode` argument must specify the upgraded node.

2.2.6.4 About Requirements for Time Synchronization on Oracle RAC

Oracle Clusterware for Oracle Database 12c requires time synchronization across all nodes within a cluster when Oracle RAC is deployed.

There are two options for time synchronization:

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

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

2.2.6.5 Recommendations for Upgrading Oracle RAC and Oracle Databases That Use ASM

Oracle ASM is installed when you install the Oracle Grid Infrastructure components. Oracle ASM shares an Oracle home with Oracle Clusterware when installed in a cluster with Oracle RAC or with Oracle Restart on a standalone server. Before the new Oracle Database software can be installed on the system, the root script for upgrading Oracle Grid Infrastructure invokes Oracle ASM Configuration Assistant (ASMCA) to upgrade Oracle ASM to the new release.

If you have an existing Oracle ASM instance, then you can either upgrade it when you install Oracle Grid Infrastructure, or you can upgrade it after the installation, using ASMCA. When you install the new software for Oracle Database 12c, Oracle Universal Installer (OUI) runs the root script `root.sh` when it detects a cluster. OUI provides an option to automate the root script execution tasks on the nodes of the cluster. You are prompted to provide information and passwords to enable OUI to run the `root.sh` script automatically. You now have an option to automate the `root.sh` script execution tasks on the nodes of the cluster.

2.2.6.6 About Upgrading System Authentication for Oracle ASM Instances

For Oracle Automatic Storage Management (Oracle ASM) environments, you can create shared ASM password files. The password file is created by the `ORAPWD` utility. See *Oracle Automatic Storage Management Administrator's Guide* for more information about managing a shared password file in a disk group. Oracle recommends that you use the `SYSASM` privilege to separate database management and storage management responsibilities. Also, you have the option to create separate operating system credentials for Oracle ASM and each database. This separation allows for an even greater division of database management and storage management responsibilities. For instance, if there are n databases using Oracle ASM on a given node, then you can configure $n + 1$ sets of operating system credentials groups whose members have `SYS` privileges: one `OSDBA` group for each database with `SYSDBA` privileges, and one `OSASM` group for the Oracle ASM instance with `SYSASM` privileges.

See Also: *Oracle Automatic Storage Management Administrator's Guide* for more information about system authentication for Oracle ASM

2.2.7 About Upgrading with Read-Only and Offline Tablespaces

Oracle Database can read file headers created in earlier releases, so you are not required to do anything to them during the upgrade. During upgrade you may have tablespaces set to read-only or offline mode as long as these tablespaces are not schema-based tablespaces (such as *SYSAUX*, *SYSTEM*, *XDB*, *HTMLDB*, *CTXSYS*, etc.); otherwise your upgrade will fail. 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 when they are changed to read/write.

In the rare case where queue tables reside in a tablespace that has been set to read-only for the upgrade, then that tablespace should be set back to read/write. Then the recreation of these queue objects can be tried again after the upgrade.

See Also: *Oracle Database Administrator's Guide* for information about transporting tablespaces between databases

2.2.8 About Upgrading Using Standby Databases

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

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

- SQL Apply and logical standby databases
You can use Oracle Data Guard SQL Apply on a logical standby database to perform a rolling upgrade to the new Oracle Database release. A logical standby accepts archived logs while the upgrade is in progress. Data protection is maintained during the Data Guard database rolling upgrade process by enabling the standby database that is the target of the upgrade to continue receiving primary database redo while the standby database is open in upgrade mode. Rolling upgrade with SQL Apply provides additional disaster protection.
- Use of physical standby database for rolling upgrades
A physical standby database can take advantage of the rolling upgrade feature provided by a logical standby. Through the use of the `KEEP IDENTITY` clause option to the `SQL ALTER DATABASE RECOVER TO LOGICAL STANDBY` statement, a physical standby database can be temporarily converted into a logical standby database for the rolling upgrade, and then reverted to the original configuration of a primary database and a physical standby database when the upgrade is done.
- Rolling Upgrade Using Active Data Guard in Oracle Database 12c
Oracle Database 12c provides the `DBMS_ROLLING` PL/SQL package, which enables you to upgrade the database software in a Data Guard configuration in a rolling fashion. Rolling Upgrade Using Active Data Guard uses a Data Guard physical standby database and the SQL Apply process. This is supported for rolling upgrades from Oracle Database release 12.1 onward. See *Oracle Data Guard Concepts and Administration* for information about using `DBMS_ROLLING` to perform a rolling upgrade.

Note: Starting with Oracle Database 12c, Oracle Enterprise Manager Cloud Control provides options to perform a rolling upgrade of databases in a Data Guard configuration. The procedures are described in online help within Cloud Control.

See Also:

- *Oracle Data Guard Broker* for information about upgrading and downgrading in an Oracle Data Guard Broker configuration
- *Oracle Data Guard Concepts and Administration* for information about upgrading using standby databases

2.2.9 About Upgrading Your Operating System

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

See Also:

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

2.2.10 Transporting Data to a Different Operating System

When using DBUA or when performing a manual upgrade for Oracle Database, you *cannot* directly migrate or transport data in a database on one operating system to a database on another operating system. For example, you cannot migrate data in an Oracle database on Solaris to an Oracle 12c database on Windows using DBUA. You must follow procedures specific to your operating system platforms. For a discussion of transporting data across platforms, see *Oracle Database Administrator's Guide*.

To see the platforms that support cross-platform data transport, run the following query using SQL*Plus:

```
SELECT * FROM V$TRANSPORTABLE_PLATFORM ORDER BY PLATFORM_NAME;
```

Note: If the source platform and the target platform are of *different* endianness, then you cannot use the `RMAN CONVERT DATABASE` command. This process requires both the source and target platform to be the same endian value. Your available options are Data Pump replication, Data Pump export/import, or Transportable Tablespace, with an `RMAN CONVERT TABLESPACE`. If the platforms are of the same endianness, then no conversion is necessary and data can be transported as if on the same platform.

See Also:

- *Oracle Database Administrator's Guide* for information on transporting data across platforms
- *Oracle Database Backup and Recovery User's Guide* for information on using the `RMAN CONVERT DATABASE` and `RMAN CONVERT TABLESPACE` commands
- [Chapter 7, "Migrating Data Using Oracle Data Pump"](#)

2.2.11 About Oracle GoldenGate and Online Database Upgrade

In an Oracle GoldenGate replication environment, you can perform an online database upgrade to the current release of Oracle Database. Using an Oracle GoldenGate replication environment minimizes database downtime during upgrading. Oracle GoldenGate is an excellent method to minimize downtime during planned maintenance, including application and database upgrades, in addition to platform migrations. Oracle GoldenGate is an Oracle product sold independently of Oracle Database for Oracle and third-party database management systems.

See Also: *Oracle GoldenGate* documentation for more information

2.2.11.1 Overview of Steps for Upgrading Oracle Database Using Oracle GoldenGate

Upgrading to Oracle Database 12c using Oracle GoldenGate consists of the following high-level steps. Unless specified, refer to the *Oracle GoldenGate* documentation library for procedures.

1. Set up a standby database running the earlier database software release using an existing database backup.
2. Upgrade the standby database to Oracle Database 12c. See ["Upgrading with Oracle Database Upgrade Assistant \(DBUA\)"](#) on page 3-6.
3. Synchronize the standby database with the production database.
4. Test your environment in active/live mode.
5. Switch over the application to the standby database.
6. Upgrade the primary database to Oracle Database 12c after comprehensive application testing at standby. See *Oracle Database Testing Guide* for information about testing a database upgrade. After testing, see [Chapter 3, "Upgrading Oracle Database."](#)

2.2.12 About Upgrading Oracle OLAP Data Security Policies

In Oracle Database 12c, Oracle OLAP uses Oracle Real Application Security (ORAS) to store OLAP data security policies instead of Extensible Data Security (XDS), which it used in Oracle Database 11g releases. When you upgrade Oracle Database from release 11g to release 12c, any XDS data security policies are automatically converted to ORAS.

Note: Data security roles defined in a release 11g Oracle Database instance are not automatically converted to ORAS. Before you upgrade an 11g database to Oracle Database 12c, you must delete any data security roles that are defined in the 11g database. After the upgrade, you may use Analytic Workspace Manager 12c to define the data security roles again.

If you upgrade an 11g database to Oracle Database 12c without deleting the 11g data security roles, then any data security policies that include a data security role are invalid in the Oracle Database 12c database.

See Also: *Oracle OLAP User's Guide*

2.2.13 Requirements for Upgrading Databases That Use Oracle Label Security and Oracle Database Vault

If you are upgrading from a database earlier than Oracle Database release 12.1 that uses Oracle Label Security (OLS) and Oracle Database Vault, then you must first run the OLS preprocess script, `olspreupgrade.sql`, to process the `aud$` table contents. The OLS upgrade moves the `aud$` table from the `SYSTEM` schema to the `SYS` schema. The `olspreupgrade.sql` script is a preprocessing script required for this move.

Caution: Running the `olspreupgrade.sql` script before upgrading is mandatory for upgrading databases earlier than Oracle Database release 12.1 that use Oracle Label Security and Oracle Database Vault. Once you have upgraded to Oracle Database release 12.1, you do not have to perform the OLS preprocessing procedure going forward to patch or upgrade the database.

The `olspreupgrade.sql` script creates a temporary table `PREUPG_AUD$` in the `SYS` schema and moves the `SYSTEM.aud$` records to `SYS.PREUPG_AUD$`. As a safety measure, Oracle recommends that you archive your audit trail as described in *Oracle Database Security Guide* before running the `olspreupgrade.sql` script. If Oracle Label Security is installed on your database, and you are upgrading from an earlier release, then you must run the OLS preprocess script before upgrading.

This section contains the following topics:

- [Requirement for Upgrading Oracle Databases That Use Oracle Database Vault](#)
- [Running `olspreupgrade.sql` on Oracle Database Release 11.1.0.7](#)
- [Running `olspreupgrade.sql` on Oracle Database Release 10.2.0.5 or 11.2](#)

See Also: *Oracle Label Security Administrator's Guide* for information about the OLS preprocess script

2.2.13.1 Requirement for Upgrading Oracle Databases That Use Oracle Database Vault

Before you upgrade databases that use Oracle Database Vault, you must be aware of certain requirements, especially for upgrading a CDB.

- When upgrading Oracle Database release 11.2 or earlier to Oracle Database 12c, if you have enabled Oracle Database Vault in your current Oracle home, then by default Oracle Database Vault is disabled in the new target Oracle home.

- If you are upgrading Oracle Database release 12.1.0.1 to release 12.1.0.2, then you must manually disable Oracle Database Vault in the release 12.1.0.1 Oracle home before you begin the upgrade process.
- If you are upgrading a CDB, then you must run `dvsys.dbms_macadm.disable_dv` as DV Owner as described in the following note.

Important: To disable Oracle Database Vault on CDBroot and PDBs, run `dvsys.dbms_macadm.disable_dv` as DV Owner as follows:

```
SQL> execute dvsys.dbms_macadm.disable_dv();
```

Stop the CDB and restart it, and open the PDBs in Read-Write state.

Once the upgrade is complete, enable Oracle Database Vault again in the new Oracle home.

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

2.2.13.2 Running `olspreupgrade.sql` on Oracle Database Release 11.1.0.7

If Oracle Label Security is installed in the earlier release that you are upgrading, then you must run the OLS preprocess `olspreupgrade.sql` script. If Oracle Database Vault is not installed with your release 11.1.0.7 database, then you can skip steps 2, 3, 6, and 7 in this section.

To run the OLS preprocess script on a release 11.1.0.7 database before upgrading:

1. Copy the `ORACLE_HOME/rdbms/admin/olspreupgrade.sql` script from the newly installed Oracle home to the Oracle home of the database to be upgraded.
2. Start SQL*Plus and connect to the database to be upgraded as DVOWNER.
3. Execute the following statement:

```
SQL> EXEC dbms_macadm.add_auth_to_realm('Oracle Database Vault','SYS',NULL, 0);
```

4. At the system prompt, enter:

```
CONNECT SYS AS SYSDBA
```

5. Run the OLS preprocess script:

```
ORACLE_HOME/rdbms/admin/olspreupgrade.sql
```

You can continue running your applications on the database while the OLS preprocess script is running.

6. After the `olspreupgrade.sql` has been successfully run, start SQL*Plus and connect to the database as DVOWNER.
7. Execute the following statement:

```
SQL> EXEC dbms_macadm.delete_auth_from_realm('Oracle Database Vault','SYS');
```

2.2.13.3 Running `olspreupgrade.sql` on Oracle Database Release 10.2.0.5 or 11.2

If Oracle Label Security is installed in the earlier release that you are upgrading, then you must run the OLS preprocess `olspreupgrade.sql` script. If Oracle Database Vault is not installed with your release 10.2.0.5 or 11.2 database, then you can skip steps 2, 3,

6, and 7 in this section.

To run the OLS preprocess script on a release 10.2.0.5 or 11.2 database before upgrading:

1. Copy the `ORACLE_HOME/rdbms/admin/olspreupgrade.sql` script from the newly installed Oracle home to the Oracle home of the database to be upgraded.

2. Start SQL*Plus and connect to the database to be upgraded as `DVOWNER`.

3. Execute the following statement:

```
SQL> GRANT DV_PATCH_ADMIN to SYS;
```

4. At the system prompt, enter:

```
CONNECT SYS AS SYSDBA
```

5. Run the OLS preprocess script:

```
ORACLE_HOME/rdbms/admin/olspreupgrade.sql
```

You may continue running your applications on the database while the OLS preprocess script is running.

6. After the `olspreupgrade.sql` has been successfully run, start SQL*Plus and connect to the database as `DVOWNER`.

7. Execute the following statement:

```
SQL> REVOKE DV_PATCH_ADMIN from SYS;
```

2.2.14 Requirement for Upgrading Databases that Use Oracle Warehouse Builder (OWB)

OWB is not installed as part of the software for Oracle Database 12c, and OWB components that may exist in earlier releases are not upgraded as part of the Oracle Database upgrade process. However, you can use OWB release 11.2.0.3 with Oracle Database 12c. OWB releases earlier than release 11.2.0.3 do not work with Oracle Database 12c.

You can use OWB release 11.2.0.3 with Oracle Database 12c in the following ways:

- [Add Oracle Database 12c Access to Existing standalone OWB 11.2.0.3 Installation](#)
- [Keep Existing OWB 11.2.0.3 Installation In-place with Oracle Database Release 11.2.0.3](#)
- [Use the Standalone OWB 11.2.0.3 Installation Where Available](#)

2.2.14.1 Add Oracle Database 12c Access to Existing standalone OWB 11.2.0.3 Installation

Oracle provides a patch update that enables OWB 11.2.0.3 to be used with Oracle Database 12c. If you have an existing standalone OWB 11.2.0.3 installation, then you can enable Oracle Database 12c to access OWB.

To add Oracle Database 12c access to an existing standalone OWB 11.2.0.3 installation:

1. Stop all OWB applications from running: the OWB Runtime service, Browser, and Name Address server.
2. Install Oracle Database 12c software and follow the database upgrade instructions as described in [Chapter 3, "Upgrading Oracle Database."](#)

3. Apply OWB Cumulative patch 16568042 to the OWB release 11.2.0.3 installation. (You can patch OWB either before or after installing Oracle Database 12c.)
To obtain the patch, go to My Oracle Support at <http://support.oracle.com>, click **Patches**, and search for patch request number 16568042.
4. Follow in-place migration steps in the patch README.txt file.
5. Continue to run OWB release 11.2.0.3 from the standalone Oracle home.

2.2.14.2 Keep Existing OWB 11.2.0.3 Installation In-place with Oracle Database Release 11.2.0.3

If OWB release 11.2.0.3 is running on a platform where a standalone installation is not available (for example, Solaris, HP, AIX, and so forth), then you must keep the Oracle Database release 11.2.0.3 software *in-place* after the migration to Oracle Database 12c.

To run OWB integrated with Oracle Database release 11.2.0.3:

1. Stop all OWB applications from running: the OWB Runtime service, Browser, and Name Address server.
2. Install Oracle Database 12c software and follow the database upgrade instructions as described in [Chapter 3, "Upgrading Oracle Database."](#)
3. After you have installed Oracle Database 12c, do *NOT* remove Oracle Database release 11.2.0.3. You run OWB from this Oracle home.
4. Apply patch 16568042 to the Oracle Database release 11.2.0.3 Oracle home. This patches the OWB installation. (You can patch OWB either before or after installing Oracle Database 12c.)

To obtain the patch, go to My Oracle Support at <http://support.oracle.com>, click **Patches**, and search for patch request number 16568042.

5. Follow in-place migration steps in the patch README.txt file.
6. Run OWB from the integrated Oracle Database release 11.2.0.3 Oracle home.

2.2.14.3 Use the Standalone OWB 11.2.0.3 Installation Where Available

If OWB 11.2.0.3 is running on a platform where a standalone installation is available (on Linux and Windows), you can install the standalone software and then remove the Oracle Database release 11.2.0.3 software.

To convert OWB from an integrated installation to a standalone installation:

1. Stop all OWB applications from running: the OWB Runtime service, Browser, and Name Address server.
2. Install the OWB standalone client into its own OWB home directory.
3. Copy the entire `owb/bin/admin` directory from the old OWB installation to the new OWB installation's `owb/bin/admin` directory. This step ensures all files and subdirectories are copied to the new OWB location.
4. Run the following SQL statement as the OWBSYS user to reset the Control Center home value. Enter from the directory location for the new `OWB_HOME` when prompted:

```
sqlplus OWBSYS/OWBSYS_PASSWORD
% New_OW_B_HOME/owb/UnifiedRepos/reset_owbcc_home.sql
```

5. Install Oracle Database 12c software and follow the database upgrade instructions as described in [Chapter 3, "Upgrading Oracle Database."](#)

6. Apply OWB Cumulative patch 16568042 to the OWB standalone release 11.2.0.3 installation. (You can patch OWB either before or after installing Oracle Database 12c.)

To obtain the patch, go to My Oracle Support at <http://support.oracle.com>, click **Patches**, and search for patch request number 16568042.

7. Follow in-place migration steps in the patch README.txt file.
8. Run OWB release 11.2.0.3 from the standalone Oracle home location.

2.2.15 Removing the Unified Auditing Schema and Roles

Remove the AUDSYS schema and the AUDIT_ADMIN and AUDIT_VIEWER roles. At this stage, there should be no AUDSYS schema.

1. Log into SQL*Plus as user SYS with the SYSDBA system privilege.

```
sqlplus sys as sysdba
Enter password: password
```

2. Drop the AUDSYS schema (if it does exist) and the AUDIT_ADMIN and AUDIT_VIEWER roles.

```
DROP USER AUDSYS CASCADE;
DROP ROLE AUDIT_ADMIN;
DROP ROLE AUDIT_VIEWER;
```

2.3 Installing the New Oracle Database Software

The following steps describe how to install the software for the new Oracle Database release.

Important: You cannot upgrade a database using Database Upgrade Assistant (DBUA) when the source and target Oracle homes are owned by different users. Attempting to do so returns error PRKH-1014. Either ensure that the source and target databases have the same owner, or perform the manual steps described in "[Manually Upgrading a Multitenant Container Oracle Database \(CDB\)](#)" on page 3-35.

To install the new Oracle Database software for this release:

1. If you are upgrading an Oracle RAC database, then you must perform the following steps in the order shown:
 - a. Upgrade Oracle Clusterware first as described in *Oracle Grid Infrastructure Installation Guide* for your operating system.

Note: When upgrading a non-Oracle RAC database, you must run Oracle Net Configuration Assistant (NETCA) before running DBUA. See "[Recommendations for Oracle Net Services When Upgrading Oracle Database](#)" on page 41. When upgrading an Oracle RAC database, as part of the Oracle Clusterware upgrade, OUI automatically runs NETCA to upgrade the network listener. Therefore, do not separately run NETCA.

- b. Mount the Oracle Grid Infrastructure installation media.
- c. Perform operating system prerequisite checks on each of the nodes that you intend to upgrade, to ensure that they meet the system prerequisites for Oracle Grid Infrastructure (Oracle Clusterware and Oracle ASM).
- d. If necessary, perform patch upgrades of the earlier release of Oracle Clusterware software to the most recent patch version.
- e. Ensure that you are logged in as the user that owns the Oracle Grid Infrastructure installation, and run the Oracle Grid Infrastructure installation. Provide information as prompted by the installer.
- f. When prompted, open a separate terminal session, log in as `root`, and run `root.sh`.

See Also:

- *Oracle Grid Infrastructure Installation Guide* for your operating system
- *Oracle Real Application Clusters Installation Guide* for your operating system

2. After upgrading Oracle Clusterware, follow the instructions in your Oracle operating system-specific documentation to prepare for installation of Oracle Database software and start the Oracle Universal Installer.
 - Oracle recommends that you run the Pre-Upgrade Information Tool before you upgrade using DBUA. You can preview the types of items DBUA checks, and see any issues ahead of time that might be present in the database. The Pre-Upgrade Information Tool helps you fix prerequisite issues that it finds. (See ["About the Pre-Upgrade Information Tool for Oracle Database"](#) on page 27.) You can then run DBUA independently after the installation is complete.
 - If you use Oracle Label Security, Oracle Database Vault, or both, then select **Enterprise Edition** on the Select Database Edition page, click **Select Options**, and enable one or both components from the components list. See ["Requirements for Upgrading Databases That Use Oracle Label Security and Oracle Database Vault"](#) on page 2-21.

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

See Also:

- ["Recommendations for Oracle Net Services When Upgrading Oracle Database"](#) on page 41
- ["Upgrading with Oracle Database Upgrade Assistant \(DBUA\)"](#) on page 3-6

2.4 Patch Set Updates and Requirements for Upgrading Oracle Database

The software for Oracle Database 12c contains a full release that includes all the latest patches and updates for Oracle Database. After upgrading, Oracle recommends that you check for patches and patch set updates as a part of database administration.

My Oracle Support provides detailed notes on how to obtain the latest patches, plus tools for lifecycle management and automated patching. For information about getting started with My Oracle Support, go to <http://support.oracle.com>.

See Also:

- My Oracle Support Note ID 854428.1, "Patch Set Updates for Oracle Products" at <http://support.oracle.com>
- "Oracle Database Upgrade Path Reference List" (Note ID 730365.1) on My Oracle Support at <http://support.oracle.com>, which contains an upgrade reference list for most available Oracle Database releases, including download information, patch numbers, and links to other notes

2.5 About the Pre-Upgrade Information Tool for Oracle Database

After you have installed the software for Oracle Database 12c, you must analyze your database before upgrading it to the new release. This is done by running the `preupgrd.sql` Pre-Upgrade Information Tool from the environment of the database you are to upgrade. Running the Pre-Upgrade Information Tool provides a preview of the items that DBUA checks and information about anything to be fixed. The Pre-Upgrade Information Tool generates fixup scripts that you can run to resolve issues that are flagged in the source database.

This section contains these topics:

- [Using the Pre-Upgrade Information Tool \(preupgrd.sql\)](#)
- [Pre-Upgrade Information Tool Warnings and Recommendations for Oracle Database](#)
- [Sample Output of the Pre-Upgrade Information Tool](#)

Note: If you are upgrading manually and have not run the Pre-Upgrade Information Tool first, then the `catctl.pl` script could terminate with errors. Running the tool is mandatory because it warns you of issues to be fixed before upgrading.

2.5.1 Using the Pre-Upgrade Information Tool (preupgrd.sql)

Oracle Database 12c introduces the `preupgrd.sql` Pre-Upgrade Information Tool. The new `preupgrd.sql` script replaces earlier versions of the Pre-Upgrade Information Tool. With the new Pre-Upgrade Information Tool, the default behavior of the prerequisite upgrade checks has been enhanced in the following ways:

- A log file, `preupgrade.log`, is created containing the output of the Pre-Upgrade Information Tool.
- The `preupgrade_fixups.sql` script is created to list and describe issues that can be fixed using SQL*Plus in the source database. It also attempts to resolve trivial issues when you execute it.
- The `postupgrade_fixups.sql` script is created to address issues that can be fixed after the database has been upgraded. See [Chapter 4, "Post-Upgrade Tasks for Oracle Database."](#)

If `Oracle_Base` is defined, then the generated scripts and log files are created in `Oracle_Base/cfgtoollogs/`. If `Oracle_Base` is not defined, then the generated scripts and log files are created in `ORACLE_HOME/cfgtoollogs/`.

To run the Pre-Upgrade Information Tool on the source database:

1. Copy `preupgrd.sql` and `utluppkg.sql` from the `rdbms/admin` directory of the new Oracle home where you installed Oracle Database 12c to a directory that is accessible when you connect to your source database, which is the database to be upgraded. Preferably, this should be a temp directory.
2. Log in to the system as the owner of the environment of the database being upgraded.

Important: The Pre-Upgrade Information Tool, which consists of `preupgrd.sql` and `utluppkg.sql`, must be copied to and must be run from the environment of the database being upgraded.

3. Start SQL*Plus and connect to the database to be upgraded using an account with DBA privileges:

```
SQL> CONNECT / AS SYSDBA
```

4. (Recommended.) Run the Pre-Upgrade Information Tool (`preupgrd.sql`) for the following scenarios:

- a. Before you upgrade a non-CDB in the source Oracle Home.

```
SQL> @$ORACLE_HOME/rdbms/temp/preupgrd.sql
```

- b. Before you upgrade a CDB in a source Oracle Home.

When running `preupgrd.sql` in a CDB, make sure all the PDBs are opened. To open all the PDBs:

```
SQL> alter pluggable database all open;
```

Run `catcon.pl` and `preupgrd.sql` as follows:

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl -n 1 -d
$ORACLE_HOME/rdbms/admin -l <output directory for catcon logs> -b preupgrd
preupgrd.sql
```

The log file `preupgrd0.log` in `<output directory for catcon logs>` contains the screen output of `preupgrd.sql`. Refer to `preupgrd0.log` for locations to the preupgrade results and fix-up scripts.

- c. Before a PDB is unplugged in the source Oracle Home. (See ["Upgrading a Pluggable Database \(PDB\)"](#) on page 3-4 for additional information.)

If running `preupgrd.sql` in a single PDB, connect to the CDB and switch to the PDB as follows:

```
alter session set container = PDB1;
```

The fixup scripts and log file are generated in the `preupgrade` directory of the source database. To view the PDB-only output, see `cfgtoollogs/SID/preupgrade/pdbfiles`. The PDB files are first written to `preupgrade/pdbfiles` and then concatenated to `preupgrade.log`, `preupgrade_fixups.sql`, and `postupgrade_fixups.sql` in the `cfgtoollogs/SID/preupgrade` directory.

5. View and read through the resulting generated fixup scripts and log file, which are located in `$ORACLE_BASE/cfgtoollogs/db_unique_name/preupgrade` if `ORACLE_BASE` is defined, or `$ORACLE_HOME/cfgtoollogs/db_unique_name/preupgrade` if `ORACLE_BASE` is not defined.
6. After you have reviewed the scripts, Oracle recommends that you execute `preupgrade_fixups.sql` on the source database. The `preupgrade_fixups.sql` script will attempt to resolve issues reported by the preupgrade process.

Issues that cannot be resolved automatically by a fixup script are flagged with ****USER ACTION REQUIRED****.
7. Fix the flagged issues that require you to complete manual procedures. See "Pre-Upgrade Information Tool Warnings and Recommendations for Oracle Database" on page 29 for more details about actions to take.
8. Run the Pre-Upgrade Information Tool as many times as needed for you to work through and resolve warnings.

2.5.2 Pre-Upgrade Information Tool Warnings and Recommendations for Oracle Database

Before upgrading to the new release of Oracle Database, Oracle recommends that you analyze the information and warnings displayed by the Pre-Upgrade Information Tool.

See Also: The following support notes on My Oracle Support at <http://support.oracle.com>:

- Note 472937.1 "Information On Installed Database Components"
- Note 753041.1 "How to Diagnose Components with NON VALID Status"

The following topics describe warnings and the appropriate actions to take:

- [Update Access Control Lists and Network Utility Packages](#)
- [Evaluate Dependencies and Add ACLs for Network Utility Packages](#)
- [About Database Links with Passwords from Earlier Oracle Database Releases](#)
- [About Oracle Database Warnings for TIMESTAMP WITH TIME ZONE Data Type](#)
- [Decrease Oracle Database Downtime by Gathering Optimizer Statistics](#)
- [Identify Invalid Objects in Oracle Database with the utluiobj Script](#)
- [Verify That Materialized View Refreshes Have Completed Before Upgrading](#)
- [Ensure That No Files Need Media Recovery Before Upgrading](#)
- [Ensure That No Files Are in Backup Mode Before Upgrading](#)
- [Resolve Outstanding Distributed Transactions Before Upgrading](#)
- [Purge the Database Recycle Bin Before Upgrading](#)
- [Synchronize the Standby Database with the Primary Database When Upgrading](#)
- [About Upgrading Oracle Application Express Databases](#)

2.5.2.1 Update Access Control Lists and Network Utility Packages

Starting with Oracle Database 12c, the access control of the UTL packages (UTL_TCP, UTL_SMTP, UTL_MAIL, UTL_HTTP, and UTL_INADDR) is implemented using Oracle Database Real Application Security and does not require Oracle XML DB.

To update ACLs and Network Utility packages:

1. Ensure that the logged-in user has the `connect` privilege for the host and port specified by `DBMS_LDAP.init`. New behavior for the `DBMS_LDAP` PL/SQL package and the `HttpUriType` type requires the creation or update of access control lists (ACLs) after performing the upgrade to the new release of Oracle Database.

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

2. If one or all of the `UTL_TCP`, `UTL_SMTP`, `UTL_MAIL`, `UTL_HTTP`, and `UTL_INADDR` packages are installed, then you might be required to re-install these packages after performing the upgrade to ensure that you have the latest version of these packages for the new Oracle Database 12c release.

See Also: *Oracle Database Real Application Security Administrator's and Developer's Guide* for information on configuring access control lists

2.5.2.2 Evaluate Dependencies and Add ACLs for Network Utility Packages

You might be required to evaluate the dependencies of network utility packages and provide access by adding the appropriate access control lists (ACLs).

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

1. Run the Pre-Upgrade Information Tool as described in "[Using the Pre-Upgrade Information Tool \(preupgrd.sql\)](#)" on page 27.
2. Check the output from the Pre-Upgrade Information Tool (`preupgrade.log`) for messages such as:

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

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

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

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

To configure network access control lists (ACLs) in the database so that these packages can work as they did in prior releases, see the example script provided in "[Configure Access Control Lists \(ACLs\) to External Network Services](#)" on page 4-6.

This script shows how to use the `DBMS_NETWORK_ACL_ADMIN` package to create, assign, and add privileges to the access control list.

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

See Also: *Oracle Database Real Application Security Administrator's and Developer's Guide* for information on configuring access control lists

2.5.2.3 About Database Links with Passwords from Earlier Oracle Database Releases

This information is important only for downgrading to your original database release after performing the upgrade. During the upgrade to Oracle Database 12c any passwords in database links are encrypted.

- To downgrade to the release from which you upgraded, all of the database links with encrypted passwords must be dropped before the downgrade. Consequently, the database links are nonexistent in the downgraded database.
- If you anticipate a requirement to be able to downgrade to your original release, then save the information about affected database links from the `SYS.LINK$` table, so that you can re-create the database links after the downgrade.
- For information about earlier releases, refer to the original documentation for the Oracle Database release from which you upgraded. You may also refer to your platform-specific *Oracle Installation Guide* for the earlier release.

See Also: *Oracle Database Administrator's Guide* for information about authentication and database links

2.5.2.4 About Oracle Database Warnings for `TIMESTAMP WITH TIME ZONE` Data Type

The time zone files that are supplied with Oracle Database 12c have been updated to reflect changes in transition rules for some time zone regions. The changes might affect existing data of the `TIMESTAMP WITH TIME ZONE` data type.

Oracle recommends that you ensure that you have the latest time zone files before you upgrade the database. If the time zone file version of the database you are upgrading is not the most recent version of the time zone file available for the new release of Oracle Database, then the Pre-Upgrade Information Tool displays a warning and describes how to proceed. [Table 2-2, "Choices for Fixing the Time Zone File Version"](#) describes the warnings and summarizes how to resolve a mismatch in time zone file versions.

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

Table 2–2 Choices for Fixing the Time Zone File Version

| IF the time zone version on the database being upgraded is... | THEN fix the time zone files... |
|---|--|
| Earlier than the most current version included in the new database release and the Pre-Upgrade Information Tool displays "Database is using a time zone file older than version <i>n</i> ." | <p><i>After</i> completing the database upgrade.</p> <p>Use the DBMS_DST PL/SQL package and follow the instructions in "Steps to Upgrade Time Zone File and Timestamp with Time Zone Data" in <i>Oracle Database Globalization Support Guide</i>.</p> <p>See Also: <i>Oracle Database PL/SQL Packages and Types Reference</i> for information on the DBMS_DST PL/SQL package and support note ID 1509653.1 "Updating the RDBMS DST version in 12c Release 1 (12.1.0.1 and up) using DBMS_DST" on My Oracle Support at http://support.oracle.com</p> |
| Later than the version included in the new database release and the Pre-Upgrade Information Tool displays "Database is using a time zone file greater than version <i>n</i> ." | <p><i>Before</i> beginning the database upgrade.</p> <p>You must patch the Oracle home by using an RDBMS_DST patch with the appropriate patch for the time zone file version in use. Apply the patch for each database to be upgraded. Otherwise, the upgrade script terminates without upgrading the database. The RDBMS_DST patches are available from My Oracle Support. Refer to note ID 412160.1 at http://support.oracle.com.</p> |

See Also:

- The support note "Updated DST Transitions and New Time Zones in Oracle Time Zone File Patches" (ID 412160.1) from My Oracle Support at <http://support.oracle.com>
- *Oracle Database Globalization Support Guide* for detailed information about upgrading the time zone file and timestamp with time zone data

2.5.2.5 Decrease Oracle Database Downtime by Gathering Optimizer Statistics

Statistics gathering occurs for those tables that lack statistics or are significantly changed during the upgrade of Oracle Database. Collect statistics before performing the actual database upgrade. Oracle strongly recommends that if your database contains thousands of dictionary tables, then collect statistics the night before starting the upgrade.

To decrease the amount of downtime:

- Oracle recommends that you use the DBMS_STATS.GATHER_DICTIONARY_STATS procedure to gather these statistics. For example, enter the following SQL statement:

```
SQL> EXEC DBMS_STATS.GATHER_DICTIONARY_STATS;
```

See Also: *Oracle Database PL/SQL Packages and Types Reference* for the syntax and complete information for the GATHER_DICTIONARY_STATS procedure

2.5.2.6 Identify Invalid Objects in Oracle Database with the utluiobj Script

Any invalid SYS or SYSTEM objects found before upgrading the database are stored in the table named registry\$sys_inv_objs after running the Pre-Upgrade Information Tool, preupgrd.sql. Additionally, any invalid non-SYS or non-SYSTEM objects found before upgrading the database are stored in registry\$nonsys_inv_objs.

To identify invalid objects before upgrading and invalid objects that result from the upgrade:

- Run `utluiobj.sql` from `$ORACLE_HOME/rdbms/admin/`
 - Running the `utluiobj.sql` script before and after upgrading enables you to compare the invalid objects after the upgrade to the objects that were invalid before the upgrade. This is like a health check for the database. Having more invalid objects after the upgrade indicates that something went wrong.
- You can fix the invalid objects, or contact My Oracle Support for assistance.

Note: Ensure that you first run the Pre-Upgrade Information Tool as part of the upgrade process. DBUA runs this tool in the Prerequisite Checks phase. You can manually run this tool before upgrading. If the Pre-Upgrade Information Tool has not previously been run, you will see an error stating that the table `registry$sys_inv_objs` does not exist. The Pre-Upgrade Information Tool creates and populates `registry$sys_inv_objs`. See "[Using the Pre-Upgrade Information Tool \(preupgrd.sql\)](#)" on page 2-27.

See Also: *Oracle Database Administrator's Guide* for information on manually recompiling invalid objects with PL/SQL package procedures

2.5.2.7 Verify That Materialized View Refreshes Have Completed Before Upgrading

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

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

- Run the following SQL query:

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

See Also:

- *Oracle Database PL/SQL Packages and Types Reference* for information on using the `DBMS_MVIEW` package to manage materialized views
- *Oracle Database SQL Language Reference* for information about the `DROP MATERIALIZED VIEW` statement to permanently remove an existing materialized view from the database

2.5.2.8 Ensure That No Files Need Media Recovery Before Upgrading

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

To get a list of files that require media recovery:

- Run the following statement:

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

See Also: *Oracle Database Backup and Recovery User's Guide* for information about performing block media recovery

2.5.2.9 Ensure That No Files Are in Backup Mode Before Upgrading

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

To get a list of files in backup mode:

- Run the following statement:

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

See Also: *Oracle Database Backup and Recovery User's Guide* for information about backing up and archiving data

2.5.2.10 Resolve Outstanding Distributed Transactions Before Upgrading

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

To resolve outstanding distributed transactions:

1. Run the following statement:

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

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

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

Tip: *Oracle Database Administrator's Guide* for information on managing distributed transactions

2.5.2.11 Purge the Database Recycle Bin Before Upgrading

All the user recycle bins in the database must be empty before you begin the upgrade process for Oracle Database. You use the `PURGE` statement to remove items and their associated objects and to release their storage space. If you have the `SYSDBA` privilege, then you can purge all the recycle bins in the entire database by specifying `DBA_RECYCLEBIN`, instead of `RECYCLEBIN`. Starting with Oracle Database 12c, you can use the new `PURGE DBA_RECYCLEBIN` system privilege to perform the same action without the requirement of granting or being granted the `SYSDBA` privilege.

The `PURGE DBA_RECYCLEBIN` statement is a special `PURGE` command that removes all the objects from the systemwide recycle bin and is equivalent to purging the recycle bin of every user. In earlier releases, this statement required the `SYSDBA` administrative privilege, which is highly undesirable in terms of separation of duty and least privilege. To provide compliance with separation of duty, Oracle Database 12c introduces a new system privilege, `PURGE DBA_RECYCLEBIN`, which enables you to run `PURGE DBA_RECYCLEBIN` without having the `SYSDBA` administrative privilege.

To empty the database recycle bin, run the following command:

```
SQL> PURGE DBA_RECYCLEBIN
```

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

See Also:

- *Oracle Database Administrator's Guide* for information about purging objects in the recycle bin
- *Oracle Database SQL Language Reference* for complete information about the `PURGE` statement
- *Oracle Database Security Guide* for information about separation of duty for database administration

2.5.2.12 Synchronize the Standby Database with the Primary Database When Upgrading

If a standby database exists, then you must synchronize it with the primary database before upgrading Oracle Database.

To check if a standby database exists and to synchronize the standby database:

1. Run the following query:

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

2. If the query in the previous step returns a row, then synchronize the standby database with the primary database.
 - Make sure all the logs are transported to the standby server after a final log switch in the primary.
 - Start the recovery of the standby database with the `NODELAY` option.

See Also: *Oracle Data Guard Concepts and Administration* for information on synchronizing a physical standby database with the primary database

2.5.2.13 About Upgrading Oracle Application Express Databases

If you are upgrading a database that includes Oracle Application Express:

- You can upgrade Oracle Application Express separately to reduce upgrade time for the database.
- A minimum value of 150 for `open_cursors` is required. Oracle recommends `open_cursors = 150` when Oracle Application Express will be upgraded. Otherwise, you may see the following error.

```
ORA-01000: maximum open cursors exceeded
ORA-06512: at "APEX_040200.WWV_FLOW_API", line 1594
```

2.5.3 Sample Output of the Pre-Upgrade Information Tool

The Pre-Upgrade Information Tool displays recommendations, but does not execute the recommendations automatically to ensure that you have control over how and when the fixup scripts are run.

The following sample shows the output that is generated and written to `preupgrad.log` by running the Oracle Database 12c Pre-Upgrade Information Tool (`preupgrd.sql`) on a release 11.2.0.3 database to be upgraded:

```

Oracle Database Pre-Upgrade Information Tool 03-06-2014 12:18:01
Script Version: 12.1.0.2.0 Build: 006
*****
Database Name: MyDB
ContainerName: Not Applicable in Pre-12.1 database
ContainerID: Not Applicable in Pre-12.1 database
Version: 11.2.0.3.0
Compatible: 11.1.0
Blocksize: 8192
Platform: Linux x86 64-bit
Timezone file: V14
*****
[Update parameters]
[Update Oracle Database 11.2.0.3.0 init.ora or spfile]

--> If Target Oracle is 32-bit, refer here for Update Parameters:
WARNING: --> "shared_pool_size" needs to be increased to at least 247463936
WARNING: --> "java_pool_size" needs to be increased to at least 67108864
WARNING: --> "db_cache_size" needs to be increased to at least 50331648
WARNING: --> "processes" needs to be increased to at least 300

--> If Target Oracle is 64-bit, refer here for Update Parameters:
WARNING: --> "shared_pool_size" needs to be increased to at least 494927872
WARNING: --> "java_pool_size" needs to be increased to at least 134217728
WARNING: --> "db_cache_size" needs to be increased to at least 50331648
WARNING: --> "processes" needs to be increased to at least 300
*****
[Renamed Parameters]
[No Renamed Parameters in use]
*****
[Obsolete/Deprecated Parameters]
[No Obsolete or Desupported Parameters in use]
*****
[Component List]
*****
--> Oracle Catalog Views [upgrade] VALID
--> Oracle Packages and Types [upgrade] VALID
--> JServer JAVA Virtual Machine [upgrade] VALID
--> Oracle XDK for Java [upgrade] VALID
--> Real Application Clusters [upgrade] INVALID
--> Oracle Workspace Manager [upgrade] VALID
--> Oracle Text [upgrade] VALID
--> Oracle XML Database [upgrade] VALID
--> Oracle Java Packages [upgrade] VALID
--> Oracle Multimedia [upgrade] VALID
--> Expression Filter [upgrade] VALID
--> Rule Manager [upgrade] VALID
*****
[Tablespaces]
*****
--> SYSTEM tablespace is adequate for the upgrade.
minimum required size: 909 MB
--> SYSAUX tablespace is adequate for the upgrade.
minimum required size: 500 MB

```

```
--> TEMP tablespace is adequate for the upgrade.
    minimum required size: 60 MB
--> UDI tablespace is adequate for the upgrade.
    minimum required size: 400 MB
```

[No adjustments recommended]

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

[Pre-Upgrade Checks]

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

WARNING: --> Process Count may be too low

Database has a maximum process count of 60 which is lower than the default value of 300 for this release.

You should update your processes value prior to the upgrade to a value of at least 300.

For example:

```
ALTER SYSTEM SET PROCESSES=300 SCOPE=SPFILE
or update your init.ora file.
```

INFORMATION: --> Older Timezone in use

Database is using a time zone file older than version 21.

After the upgrade, it is recommended that DBMS_DST package be used to upgrade the 11.2.0.3.0 database time zone version to the latest version which comes with the new release.

Please refer to My Oracle Support note number 977512.1 for details.

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

[Pre-Upgrade Recommendations]

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

```
*****
***** Dictionary Statistics *****
*****
```

Please gather dictionary statistics 24 hours prior to upgrading the database.

To gather dictionary statistics execute the following command while connected as SYSDBA:

```
EXECUTE dbms_stats.gather_dictionary_stats;
```

^^^ MANUAL ACTION SUGGESTED ^^^

```
*****
***** Hidden Parameters *****
*****
```

Please review and remove any unnecessary hidden/underscore parameters prior to upgrading. It is strongly recommended that these be removed before upgrade unless your application vendors and/or Oracle Support state differently. Changes will need to be made in the init.ora or spfile.

^^^ MANUAL ACTION SUGGESTED ^^^

To view existing hidden parameters execute the following command while connected AS SYSDBA:

```
SELECT name, value from SYS.V$PARAMETER WHERE name
      LIKE '\_%' ESCAPE '\' order by name;
```

Or run the Pre-Upgrade Fixup Script to display the Hidden Parameters currently set.

```
*****
                        [Post-Upgrade Recommendations]
*****
```

```
*****
***** Fixed Object Statistics *****
*****
```

Please create stats on fixed objects two weeks after the upgrade using the command:

```
EXECUTE DBMS_STATS.GATHER_FIXED_OBJECTS_STATS;
```

^^^ MANUAL ACTION SUGGESTED ^^^

```
*****
***** Summary *****
```

```
0 ERRORS exist in your database.
1 WARNING that Oracle suggests are addressed to improve database performance.
1 INFORMATIONAL message that should be reviewed prior to your upgrade.
```

After your database is upgraded and open in normal mode you must run rdbms/admin/catuppst.sql which executes several required tasks and completes the upgrade process.

You should follow that with the execution of rdbms/admin/utlrp.sql, and a comparison of invalid objects before and after the upgrade using rdbms/admin/utluiobj.sql

If needed you may want to upgrade your timezone data using the process described in My Oracle Support note 977512.1

```
*****
```

2.6 Backing Up Oracle Database for Upgrading

After running the Pre-Upgrade Information Tool and cleanly shutting down the database, Oracle recommends that you back up your Oracle database. To minimize downtime, you may perform an online backup or create a guaranteed restore point. Database Upgrade Assistant (DBUA) enables you to specify your backup and restore point.

Important: Before you make any changes to the Oracle software, Oracle recommends that you create a backup of the Oracle software and databases. For Oracle software running on Windows operating systems, you must also take a backup of the Windows registry. Without a registry backup, you cannot restore the Oracle software to a working state if the upgrade to Oracle Database 12c fails and you want to revert to the previous software installation.

See Also:

- *Oracle Database Backup and Recovery User's Guide* for information on online backups and backup mode
- *Oracle Database Backup and Recovery User's Guide* for information on Flashback Database and restore points

To perform a backup of the database to be upgraded:

1. Sign on to Oracle RMAN:

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

2. Run the following RMAN commands:

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

See Also: *Oracle Database Backup and Recovery User's Guide* for more information about performing RMAN backups

2.7 Preparing the New Oracle Home for Upgrading

After backing up the database to be upgraded, prepare the new Oracle home in a new location. Install the software for Oracle Database 12c into the new location.

See Also:

- ["Backing Up Oracle Database for Upgrading"](#) on page 38
- ["Prerequisites for Preparing Oracle Home on Windows"](#) on page 41

To prepare the new Oracle home for upgrading:

1. Copy configuration files from the Oracle home of the database being upgraded to the new Oracle home for Oracle Database 12c. If you are using DBUA, the configuration files are copied for you automatically and you can ignore this step.

If you need to manually copy your configuration files to the new 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, but it must not reside in the Oracle home of the old environment after you upgrade to Oracle Database 12c.

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. See *Oracle Database Administrator's Guide* for information about managing initialization parameters. See *Oracle Real Application Clusters Administration and Deployment Guide* for information about initialization parameter files in Oracle RAC environments.

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

```
CREATE pfile FROM spfile;
```

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

- c. If your parameter file is a text-based initialization parameter file with either an IFILE (include file) or a SPFILE (server parameter file) entry, and the file specified in the IFILE or SPFILE entry resides within the old environment's Oracle home, then copy the file specified by the IFILE or SPFILE entry to the new Oracle home. The file specified in the IFILE or SPFILE entry contains additional initialization parameters.
- d. If you have a password file that resides within the old environment's Oracle home, then move or copy the password file to the new Oracle home in Oracle Database 12c.

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.

2. Adjust your parameter file in Oracle Database 12c by completing the following steps:

- a. Remove desupported initialization parameters and adjust deprecated initialization parameters. Certain parameters are desupported in Oracle Database 12c, while other parameters have become deprecated. Remove all desupported parameters from any parameter file that starts an Oracle Database 12c instance. Desupported parameters might cause errors in Oracle Database 12c. Also, alter any parameter whose syntax has changed in the new release.

The Pre-Upgrade Information Tool displays any deprecated parameters and desupported parameters it finds in the **Deprecated Parameters** and **Desupported Parameters** sections, respectively.

See Also: [Chapter 8, "Deprecated and Desupported Features for Oracle Database 12c"](#) for the initialization parameters that have been deprecated or have been desupported

- b. Adjust the values of the initialization parameters to at least the minimum values indicated by the Pre-Upgrade Information Tool.
- c. Make sure all path names in the parameter file are fully specified. You should not have relative path names in the parameter file.
- d. If the parameter file contains an IFILE entry, then change the IFILE entry in the parameter file to point to the new location of the include file that you specified in step 1. 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.
- e. If you are upgrading a cluster database, then you may need to modify the SPFILE or `initORACLE_SID.ora` files.

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

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`.

See Also:

- *Oracle Database Reference* for information about `CLUSTER_DATABASE`
- *Oracle Real Application Clusters Administration and Deployment Guide* for information about initialization parameter files in Oracle RAC

2.7.1 Prerequisites for Preparing Oracle Home on Windows

For security reasons, different Windows user accounts configured as Oracle home users for different Oracle homes are not allowed to share the same Oracle Base.

The following recommendations apply before you can upgrade Oracle Database on Windows platforms:

- Database upgrade is supported when the same Windows user account is used as the Oracle home user in both the *source* and *destination* Oracle homes.
- Database upgrade is supported when the Oracle home from which the database is being upgraded uses the Windows Built-in Account. Releases earlier than Oracle Database 12c (release 11.2 and earlier) only supported the built-in account option for the Oracle home user on Windows.
- The Oracle home user may not have access to files outside its own Oracle Base and Oracle home. Therefore, if you choose a different Oracle Base during upgrade, Oracle Database services may not have access to the files in the older Oracle Base. Using DBUA for database upgrade ensures that the Oracle home user has access to files outside its own Oracle Base and Oracle home.

Before upgrading manually or using the custom files (such as, wallets and configuration files) from the older Oracle Base, you must grant access to the Oracle home user for these outside files or copy these files to the new Oracle Base.

See Also: *Oracle Database Platform Guide for Microsoft Windows* for information about database administration on Windows

2.8 Recommendations for Oracle Net Services When Upgrading Oracle Database

In Oracle Database 12c, new, underlying net services parameters enable data compression, which reduces the size of the session data unit to be transmitted over a SQL TCP connection.

The following new parameters for the `sqlnet.ora` file specify compression, and the preferred compression scheme:

- `SQLNET.COMPRESSION`
- `SQLNET.COMPRESSION_LEVELS`
- `SQLNET.COMPRESSION_THRESHOLD`

These new parameters are not supported in earlier releases and are only available in Oracle Database 12c.

See Also: *Oracle Database Net Services Reference* for information about the new `sqlnet.ora` compression parameters

If a listener was not configured in the database from which you are upgrading, then you must run Oracle Net Configuration Assistant (Net CA) to configure the listening protocol address and service information for the new release of Oracle Database, including a `listener.ora` file, *before* running DBUA. A new version of the listener is required for releases of Oracle Database earlier than release 11.2. The new listener is backward compatible with earlier Oracle databases.

You can upgrade an Oracle RAC database with DBUA, which automatically migrates the listener from your old Oracle home to the new Oracle Grid Infrastructure home. You must administer the listener by using the `lsnrctl` command in the Oracle Grid Infrastructure home. Do not attempt to use the `lsnrctl` commands from Oracle home locations for earlier releases.

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

2.9 Testing the Upgrade Process for Oracle Database

Oracle recommends that you create a full working copy of your database environment in which to test all the pre-upgrade, upgrade, and post-upgrade processes. You can create a test environment that does not interfere with the current production Oracle database. Oracle Data Guard, for example, enables you to create physical and snapshot standby databases.

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 of the current production database.
- If you plan to use Data Pump Export/Import, then export and import in stages, using subsets of the current production database.

Practice upgrading the database using the test environment. The best practice is to perform testing of the upgrade process on an exact copy of the database to be upgraded, rather than on a downsized copy or test data. If an exact copy is impractical, then carefully chose a representative subset of your data to move over to your test environment and test the upgrade on that data.

See Also:

- *Oracle Database Testing Guide* for information about testing a database upgrade
- *Oracle Database Utilities* for information on Data Pump Export and Import utilities
- *Oracle Data Guard Concepts and Administration* for information on physical and snapshot standby databases
- Note ID 1462240.1 "Oracle Upgrade Companion" on My Oracle Support at <http://support.oracle.com> for detailed information and pointers about testing before upgrading

2.9.1 Upgrade Oracle Call Interface (OCI) and Precompiler Applications

Upgrade any Oracle Call Interface (OCI) and precompiler applications that you plan to use with the new release of Oracle Database. Oracle recommends that you test these

applications on a sample database before upgrading your current production database.

See Also: ["Upgrading Precompiler and OCI Applications in Oracle Database"](#) on page 5-2 for more information

2.10 Testing an Upgraded Test Copy of Oracle Database

Perform the planned tests on the current database and on the test database that you upgraded to the new release of Oracle Database.

- Compare the test results, noting anomalies.
- Repeat the test upgrade as many times as necessary until issues are resolved.

Test the newly upgraded test database with existing applications to verify that they operate properly with a new Oracle database.

- Test enhanced functions and new capabilities by adding available Oracle Database features.
- Ensure that the applications operate in the same manner as they did in the current database.

See Also:

- [Chapter 5, "Upgrading Applications After Upgrading Oracle Database"](#) for more information on using applications with Oracle Database
- *Oracle Database Testing Guide* for information on testing a database upgrade

Upgrading Oracle Database

Oracle provides a comprehensive set of tools for upgrading Oracle Database with minimal downtime and for migrating your applications to the new release.

This chapter contains the following topics:

- [Upgrade Strategies for Oracle Database](#)
- [Upgrading with Oracle Database Upgrade Assistant \(DBUA\)](#)
- [About the Parallel Upgrade Utility for Oracle Database](#)
- [Manually Upgrading a Non-CDB Oracle Database](#)
- [Manually Upgrading a Multitenant Container Oracle Database \(CDB\)](#)
- [How the Data Dictionary is Upgraded and Status is Displayed](#)
- [About the catupgrd.sql Script in Earlier Releases of Oracle Database](#)
- [About the catuppst.sql Script](#)
- [About Transporting and Upgrading a Database \(Full Transportable Export/Import\)](#)
- [Troubleshooting the Upgrade for Oracle Database](#)
- [Rerunning the Upgrade for Oracle Database](#)

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

3.1 Upgrade Strategies for Oracle Database

Starting with Oracle Database 12c, multitenant architecture enables an Oracle database to function as a multitenant container database (CDB) with pluggable databases. All Oracle Database releases earlier than Oracle Database 12c were non-CDB. See *Oracle Database Concepts* for an overview of multitenant architecture.

This release provides more strategies for upgrading. After upgrading an earlier release to Oracle Database 12c, you can plug the upgraded database into a CDB as described in "[Upgrading an Earlier Release and Plugging a PDB into a CDB](#)" on page 2.

Important: You cannot downgrade a database once you have set the compatible initialization parameter to 12.1.0.2. Only if the compatibility is set to 12.1.0.1 will a downgrade be possible for a pluggable database (PDB), and there may still be restrictions on downgrading. See [Chapter 6, "Downgrading Oracle Database to an Earlier Release."](#)

This section contains the following topics:

- [Upgrading a Non-CDB Oracle Database](#)
- [Upgrading an Earlier Release and Plugging a PDB into a CDB](#)
- [Upgrading a Multitenant \(CDB\) Oracle Database](#)
- [Upgrading a Pluggable Database \(PDB\)](#)

See Also: *Oracle Database Administrator's Guide* for complete information about creating and configuring a CDB and *Oracle Database Concepts* for an overview of multitenant architecture

3.1.1 Upgrading a Non-CDB Oracle Database

A non-CDB Oracle Database does not use the multitenant architecture and does not contain pluggable databases (PDBs). You can upgrade the database with Oracle Database Upgrade Assistant (DBUA) or manually. Follow the procedures in "[Upgrading with Oracle Database Upgrade Assistant \(DBUA\)](#)" on page 6, or "[Manually Upgrading a Non-CDB Oracle Database](#)" on page 29.

3.1.2 Upgrading an Earlier Release and Plugging a PDB into a CDB

You can upgrade earlier releases of Oracle Database using either DBUA or the Parallel Upgrade Utility and then plug the upgraded database into a multitenant container database (CDB). The upgraded release 12.1 database can be plugged into an existing CDB. Be sure to follow any necessary pre-upgrade procedures described in [Chapter 2, "Preparing to Upgrade Oracle Database."](#) The following procedure assumes that both databases are on the same system.

See Also: *Oracle Database Administrator's Guide* for information about creating and removing PDBs with SQL*Plus, and *Oracle Database Administrator's Guide* for examples of various scenarios with different factors.

To upgrade a database and plug it into a CDB:

1. Install the new Oracle Database 12c software. See *Oracle Database Installation Guide* for your operating system platform.
2. Upgrade the database as described in this guide. See [Chapter 3, "Upgrading Oracle Database."](#)
3. Set the COMPATIBLE parameter to 12.0.0, if you have not already done so as part of the upgrade process. See "[The COMPATIBLE Initialization Parameter in Oracle Database](#)" on page 1-11.
4. Make sure the database is in read-only mode.

```
SQL> startup mount
SQL> alter database open read only;
```

5. Ensure that the prerequisites for plugging an unplugged PDB are met. See *Oracle Database Administrator's Guide* for information about using the `DBMS_PDB.CHECK_PLUG_COMPATIBILITY` function to determine whether these requirements are met.
6. Create the XML file for the PDB. The root name for the XML file matches the name of the PDB. The *path* to the location can be anything you choose. This location is where the XML file will be saved.

```
SQL> exec DBMS_PDB.DESCRIBE('path/PDB1.xml');
```

7. Shut down the database.

```
SQL> SHUTDOWN IMMEDIATE
```

8. Plug the database into the CDB.

```
SQL> CREATE PLUGGABLE DATABASE PDB1 USING 'path/PDB1.xml' NOCOPY TEMPFILE REUSE;
```

In this example, the PDB is named `PDB1`. You can use any name, but the name must be unique for this CDB.

This example also assumes that the data files remain in the same location as they were when the database was standalone. If the data files have been copied to a different location (for example, stored with Oracle ASM), then the parameter `SOURCE_FILE_NAME_CONVERT` must be specified. See *Oracle Database Administrator's Guide* for an example of using the `SOURCE_FILE_NAME_CONVERT` clause.

`TEMPFILE REUSE` specifies that the existing `TEMP` tablespaces can be reused.

You should see "Pluggable database created." The upgraded database is now a PDB ready for the CDB.

9. Connect to the PDB using a previously configured listener. The name of the PDB in this example is `PDB1`.

```
SQL> sys/oracle@PDB1 as sysdba
```

You can alternatively connect to the PDB as follows:

```
SQL> ALTER SESSION set container=PDB1;
```

10. Convert the dictionary to the PDB type. From the `admin` directory, run the `noncdb_to_pdb.sql` script. This script must be run before the PDB can be opened for the first time.

```
@$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql
```

11. Startup and open the new PDB in read/write mode. You must open the new PDB in read/write mode for Oracle Database to complete the integration of the new PDB into the CDB. See *Oracle Database Administrator's Guide* for more information about modifying the open mode of PDBs with the `STARTUP SQL*Plus` command.

```
SQL> STARTUP PLUGGABLE DATABASE PDB1 OPEN;
```

12. Back up the PDB. See *Oracle Database Backup and Recovery User's Guide* for information about backing up a PDB.

3.1.3 Upgrading a Multitenant (CDB) Oracle Database

Starting with Oracle Database 12c, Oracle provides multitenant architecture, which enables the creation and administration of pluggable databases (PDBs). You can

upgrade the CDB with Oracle Database Upgrade Assistant (DBUA) or manually, as you would a non-CDB database. Follow the procedures in ["Upgrading with Oracle Database Upgrade Assistant \(DBUA\)"](#) on page 6, or ["Manually Upgrading a Multitenant Container Oracle Database \(CDB\)"](#) on page 35.

3.1.4 Upgrading a Pluggable Database (PDB)

A CDB can contain zero, one, or more pluggable databases (PDBs). You can upgrade one PDB without upgrading the whole CDB. For example, you can unplug a PDB from a release 12.1.0.1 CDB, plug it into a release 12.1.0.2 CDB, and then upgrade that PDB to release 12.1.0.2. You cannot use DBUA to upgrade a PDB; therefore you can manually upgrade the PDB. Follow the procedure in this section, and for more information see ["Manually Upgrading a Multitenant Container Oracle Database \(CDB\)"](#) on page 35.

Note: A PDB cannot be recovered unless it is backed up. After upgrading using the method of creating a CDB and plugging in a database, be sure to back up the PDB. See *Oracle Database Backup and Recovery User's Guide* for information about backing up a PDB.

Step 1 Prepare the release 12.1.0.1 PDB

In this step you prepare the PDB you want to upgrade to release 12.1.0.2.

1. In the release 12.1.0.1 environment, switch to the PDB you want to upgrade. For example, PDB1:

```
SQL> alter session set container=PDB1;
```

2. Run the **preupgrade** script from the release 12.1.0.2 environment to generate fixup scripts:

```
@$ORACLE_HOME_12102/rdbms/admin/preupgrd.sql
```

The output of the `preupgrade.log` includes the location of the fixup scripts. The log files and fixup scripts will be created under `ORACLE_BASE/cfgtoollogs`, if `ORACLE_BASE` is defined. If `ORACLE_BASE` is not defined, then the files will be created under `ORACLE_HOME/cfgtoollogs` instead of `ORACLE_BASE/cfgtoollogs`.

To view the PDB-only output, see `cfgtoollogs/SID/preupgrade/pdbfiles`. The PDB files are first written to `preupgrade/pdbfiles` and then concatenated to `preupgrade.log`, `preupgrade_fixups.sql`, and `postupgrade_fixups.sql` in the `cfgtoollogs/SID/preupgrade` directory.

3. Run the fixup script as follows:

```
@$ORACLE_BASE/cfgtoollogs/CDB1/preupgrade/preupgrade_fixups.sql
```

Note: You must be connected to the PDB before you run `preupgrade_fixups.sql`.

4. To decrease the amount of downtime during upgrade, gather optimizer statistics as follows:

```
SQL> exec dbms_stats.gather_dictionary_stats;
```

At this time, also apply any additional recommendations from the `preupgrade.log`.

5. Switch to the CDB root container:

```
SQL> alter session set container=CDB$ROOT;
```

6. Close the PDB, in this example PDB1:

```
SQL> alter pluggable database PDB1 close;
```

7. Unplug PDB1 and stage it so it can be plugged into the 12.1.0.2 CDB. Then exit this CDB when finished. For example:

```
SQL> alter pluggable database PDB1 unplug into '/stage/PDB1.xml';
SQL> exit
```

You are now ready to plug the standalone PDB in to the release 12.1.0.2 CDB, as described in ["Plug the PDB into the 12.1.0.2 CDB"](#) on page 5.

Step 2 Plug the PDB into the 12.1.0.2 CDB

In this step you plug the PDB that you prepared in Step 1 in to the release 12.1.0.2 CDB. The example uses a PDB named PDB1. Note that when you ran DBCA for release 12.1.0.2, the compatible parameter was set to 12.1.0.2 by default. If you changed the compatible parameter to an earlier release, then you must set it to 12.1.0.2 in order to perform the upgrade.

1. Connect to the release 12.1.0.2 CDB:

```
SQL> alter session set container=CDB$ROOT;
```

2. Create the 12.1.0.2 PDB using the staged 12.1.0.1 PDB:

```
SQL> create pluggable database PDB1 using '/stage/PDB1.xml'
file_name_convert=('/oradata/CDB1/PDB1', '/oradata/CDB2/PDB1');
```

See Also: *Oracle Database Administrator's Guide* for information about creating PDBs with SQL*Plus

3. Open the PDB in upgrade mode:

```
SQL> alter pluggable database PDB1 open upgrade;
```

4. Exit SQL*Plus in order to run the Parallel Upgrade Utility, which is a Perl script, in Step 3.

```
SQL> Exit
```

Step 3 Upgrade the PDB

In this step you run the Parallel Upgrade Utility and the fixup scripts to complete the upgrade.

1. From the command prompt, change to the admin directory of the 12.1.0.2 Oracle home:

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

2. Run the Parallel Upgrade Utility, `catctl.pl`. Use the syntax you normally use with `catctl.pl`, except that you must use the `-c PDBname` option to specify which PDB you are upgrading. Capitalize the name of your PDB as shown in the example using PDB1.

```
$ORACLE_HOME/perl/bin/perl catctl.pl -c "PDB1" -l /home/oracle/upgrade
catupgrd.sql
```

Refer to ["Parallel Upgrade Utility \(catctl.pl\) Parameters"](#) on page 28 for information about the syntax for `catctl.pl`.

3. Log in to SQL*Plus and start the PDB to run required scripts and the **post-upgrade** fixup scripts.

```
SQL> alter session set container=PDB1;
SQL> startup
```

4. Run `utlrp.sql` to compile objects.

```
@?/rdbms/admin/utlrp.sql
```

5. Run the post-upgrade fixup scripts to complete the upgrade. (Make sure you are connected to the PDB.)

```
$_ORACLE_BASE/cfgtoollogs/CDB1/preupgrade/postupgrade_fixups.sql
```

If `ORACLE_BASE` is not set, the files will be created under `$_ORACLE_HOME/cfgtoollogs` instead of `$_ORACLE_BASE/cfgtoollogs`.

Note:

- If you move either a PDB or any other stand-alone database from server A to server B, you must copy the `postupgrade_fixups.sql` script to the new location to execute it post-upgrade in the new environment.
 - The post-upgrade fixup script for the specified PDB is located in `preupgrade/pdbfiles/postupgrade_fixups.PDB_NAME.sql` and its contents are concatenated to `preupgrade/postupgrade_fixups.sql`.
-
-

Repeat the steps to upgrade multiple PDBs. When you run `catctl.pl` you can specify all the PDBs at once as follows:

```
$_ORACLE_HOME/perl/bin/perl catctl.pl -c "PDB1, PDB2" -l /home/oracle/upgrade
catupgrd.sql
```

3.2 Upgrading with Oracle Database Upgrade Assistant (DBUA)

Database Upgrade Assistant (DBUA) provides a graphical user interface to guide you through the upgrade of Oracle Database. DBUA works for CDB and non-CDB database systems.

The following topics describe recommendations and procedures for upgrading a database using DBUA:

- [Recommendations for Using DBUA](#)
- [How DBUA Processes the Upgrade for Oracle Database](#)
- [Using DBUA to Upgrade the Database on Linux, UNIX, and Windows Systems](#)
- [About Windows Authentication When Upgrading](#)
- [Moving a Database from an Existing 12c Oracle Home](#)
- [Using DBUA in Silent Mode to Upgrade Oracle Database](#)

Note: You can invoke DBUA in silent mode, which does not present a user interface, as described in ["Using DBUA in Silent Mode to Upgrade Oracle Database"](#) on page 23. This is useful for large roll-outs and scripts.

3.2.1 Recommendations for Using DBUA

The procedures for using DBUA are the same for a multitenant container database (CDB), pluggable databases (PDBs), and for a non-CDB. See *Oracle Database Concepts* for an overview and documentation roadmap for working with a CDB and PDBs. However, there are considerations and recommendations for running DBUA. This section summarizes the recommendations. Also see ["About Stopping DBUA When Upgrading"](#) on page 3-7.

Recommendations for using DBUA are as follows:

- Oracle strongly recommends that you run the Pre-Upgrade Information Tool before starting the upgrade with DBUA. Although DBUA runs the Pre-Upgrade Information Tool as part of the pre-requisite checks, it is good practice to run the tool ahead of time in order to analyze the database and take actions that can decrease downtime for upgrading. See ["About the Pre-Upgrade Information Tool for Oracle Database"](#) on page 2-27 for information and steps.
- Set Archive Log ON during upgrade. Oracle recommends that you set Archive Log ON in order for DBUA to create and update the log file for the upgrade process.
- If the database instance is not running, then DBUA tries to start the instance. If the instance is up and running, then DBUA connects to it.
- For Oracle RAC, if you upgrade a cluster database using DBUA, then you must leave the CLUSTER_DATABASE initialization parameter set to TRUE.
- If you restore your database manually (not using DBUA), then remove the `Welcome_SID.txt` file, which is located in the `ORACLE_HOME/cfgtoollogs/dbua/logs/` directory, before starting DBUA. The presence of this file indicates to DBUA that this is a re-run operation.
- If Oracle Database Vault is enabled, then see ["Requirement for Upgrading Oracle Databases That Use Oracle Database Vault"](#) on page 2-21.

3.2.1.1 About Stopping DBUA When Upgrading

If you stop the upgrade, but do not restore the database, then you cannot restart DBUA until you start the existing database in UPGRADE mode using the Oracle Database 12c server. You cannot go back to the original Oracle Database server unless you restore your database. Instead, you must continue with a manual (command line) upgrade as described in ["Manually Upgrading a Multitenant Container Oracle Database \(CDB\)"](#) on page 35.

3.2.2 How DBUA Processes the Upgrade for Oracle Database

If you installed the software for Oracle Database 12c and specified that you are upgrading an existing Oracle database, then DBUA starts automatically. You can also start DBUA independently after the installation is complete as described in ["Rerunning the Upgrade for Oracle Database"](#) on page 55.

While the upgrade is in process, DBUA shows the upgrade progress for each component. DBUA writes detailed trace and log files and produces a complete HTML report for later reference. To enhance security, DBUA automatically locks new user

accounts in the upgraded database. DBUA then proceeds to create new configuration files (parameter and listener files) in the new Oracle home.

DBUA does not begin the upgrade process until all of the pre-upgrade steps are completed. Refer to ["Tasks for Preparing to Upgrade Oracle Database"](#) on page 2-1, which provides a general overview and pointers to further information and procedures.

See Also: ["Upgrade Scripts Invoked by DBUA"](#) on page 3-8

3.2.2.1 Upgrade Scripts Invoked by DBUA

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

During the prerequisite phase, DBUA runs the Pre-Upgrade Information Tool, `preupgrd.sql`, and uses the following logic to modify or create new required tablespaces:

- If the datafiles are auto-extensible and have enough disk space to grow, then DBUA continues with the upgrade.
- If the datafiles are not auto-extensible, then DBUA prompts you and makes the files auto-extensible.
- If the tablespaces are auto-extensible and the `MAXSIZE` initialization parameter needs adjustment, then DBUA prompts for the same and adjusts the `MAXSIZE` parameter.
- If there is not enough disk space to grow, then DBUA prompts you to create space (by adding more datafiles). DBUA does not automatically add new datafiles because DBUA cannot determine where to create the files.

DBUA addresses many issues found during the prerequisite phase. For example, DBUA can ensure that the correct time zone file is used and make ACL adjustments for network access control lists.

During the upgrade phase, DBUA runs `catctl.pl`, which runs the upgrade processes in parallel instead of serially, optimally taking advantage of CPU capacity to decrease downtime as much as possible.

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

The steps to upgrade a database using the DBUA graphical user interface are performed from within the new Oracle home where the Oracle Database 12c software has been installed. Only an Administrator should invoke DBUA for Windows systems (see ["About Windows Authentication When Upgrading"](#) on page 21).

See Also:

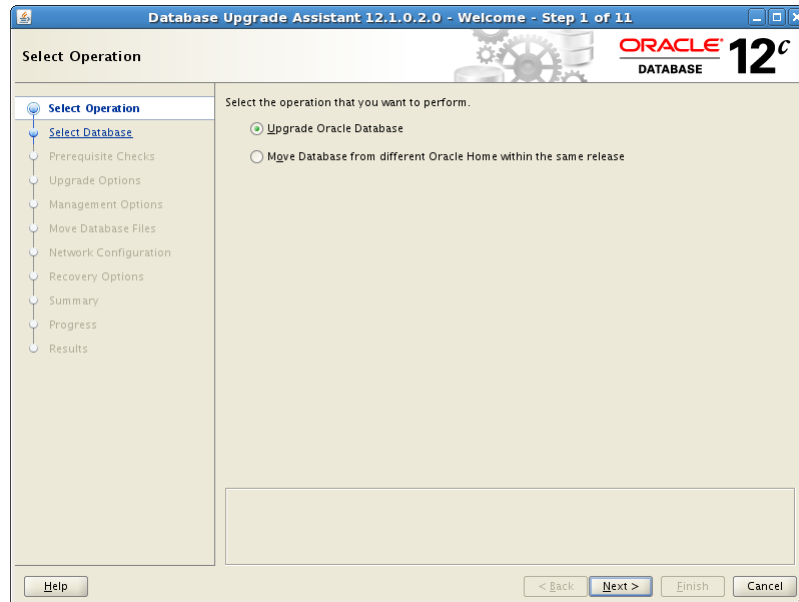
- ["Installing the New Oracle Database Software"](#) on page 2-25 for more information about the software location
- ["Pre-Upgrade Information Tool Warnings and Recommendations for Oracle Database"](#) on page 2-29 for information about preparing for the upgrade to minimize down time
- ["Moving a Database from an Existing 12c Oracle Home"](#) on page 3-22 for the procedure to move an existing database

To upgrade a database using DBUA on Linux, UNIX, or Windows:

1. The dbua executable is located in the `ORACLE_HOME/bin` directory. Start DBUA from the Oracle home where the new database software has been installed:
 - On Linux or UNIX platforms, enter the following command at a system prompt in the new home for Oracle Database 12c:


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

The DBUA Select Operation page displays.



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

Click **Next**.

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

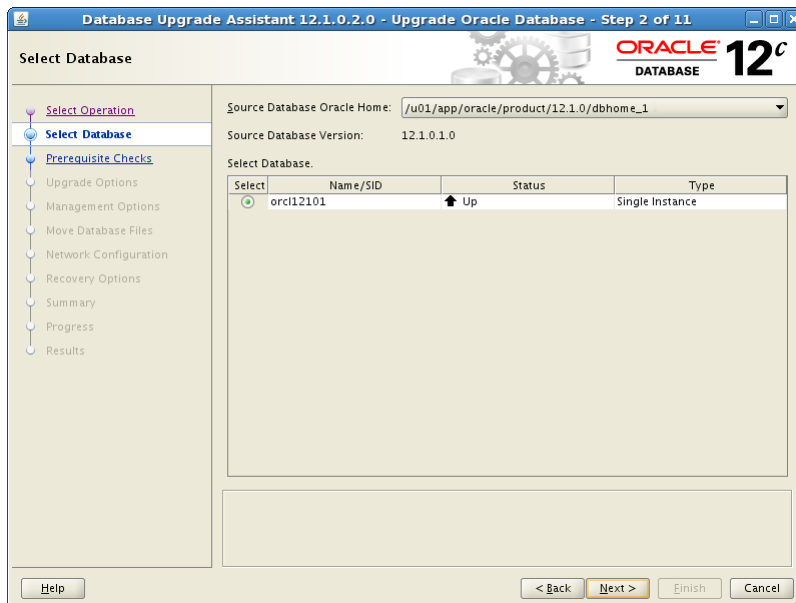
If the source database does not have operating system authentication, then DBUA prompts you for SYSDBA user name and password.

See Also: ["Moving a Database from an Existing 12c Oracle Home"](#) on page 22 for the procedure to move a database from an existing 12.1 Oracle home to a new 12.1 Oracle home

Note:

- This may affect your authentication on Windows systems: Starting with Oracle Database 12c, Windows NTS authentication using the NTLM protocol is no longer supported for security reasons. Kerberos authentication is the only supported authentication. In this release, NTS does not work in Windows NT domains nor in domains with Windows NT controllers.
- This may affect user accounts on Windows Systems: Starting with Oracle Database 12c, Oracle uses the standard Windows User Account instead of Local System Account to run Oracle database services. This separation of roles provides better security on Windows.

See *Oracle Database Platform Guide for Microsoft Windows* for information about managing user accounts on Windows systems.



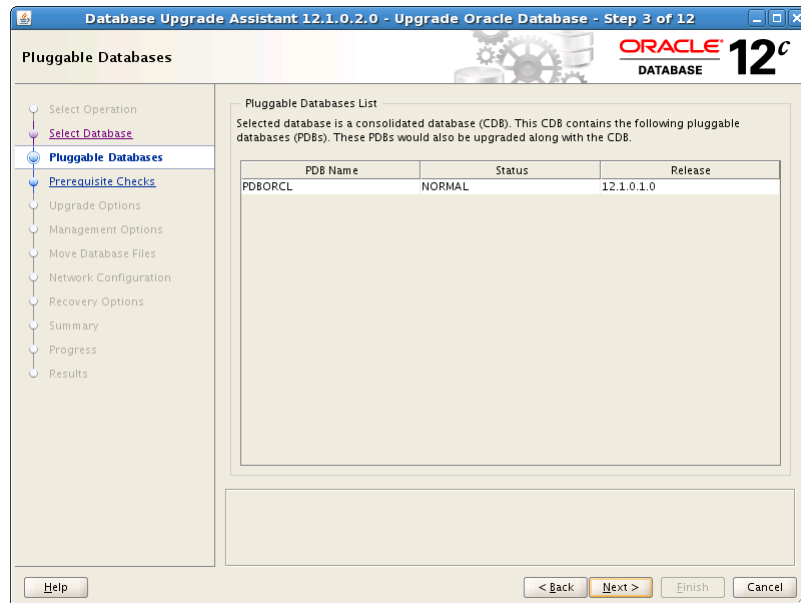
3. From the **Source Database Oracle Home** drop-down list, choose the database you want to upgrade to Oracle Database 12c. DBUA fills in the information for the release and selects the Oracle database associated with the selected Oracle home.

You can select only one database at a time. If the database does not appear in the list, then make sure an entry with the database name exists in the `oratab` file in the `etc` directory.

If you run 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**.

If the selected database is a multitenant container database (CDB), then DBUA displays the Pluggable Databases page. The Pluggable Databases page lists the pluggable databases contained in the CDB, which will be upgraded along with the selected CDB.

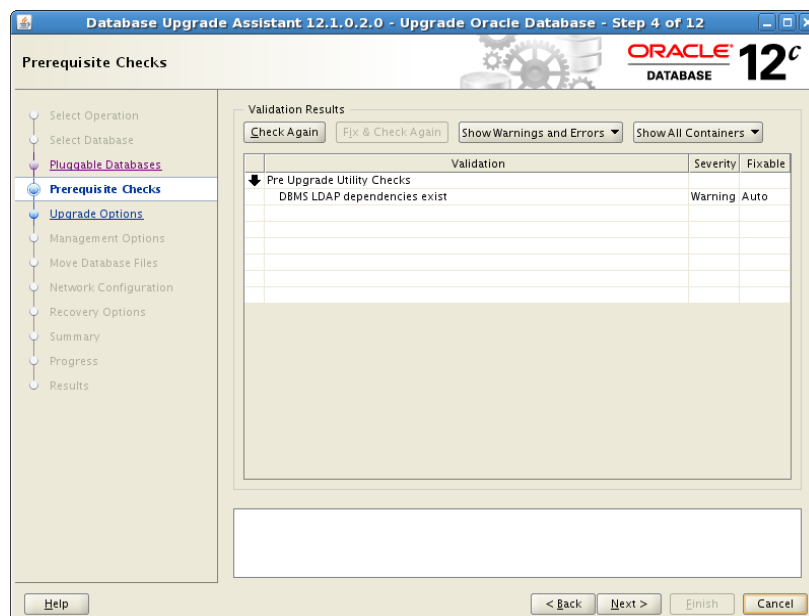


Click **Next** again.

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

- Empty database recycle bin.
- Invalid objects.
- Deprecated and desupported initialization parameters.
- Time zone data file version.

When DBUA finishes its analysis, the Prerequisite Checks page displays. The analysis takes several minutes to complete.



4. The Prerequisite Checks page shows the validation and results, the severity, whether the result is fixable if there is a problem, and the action you can take.

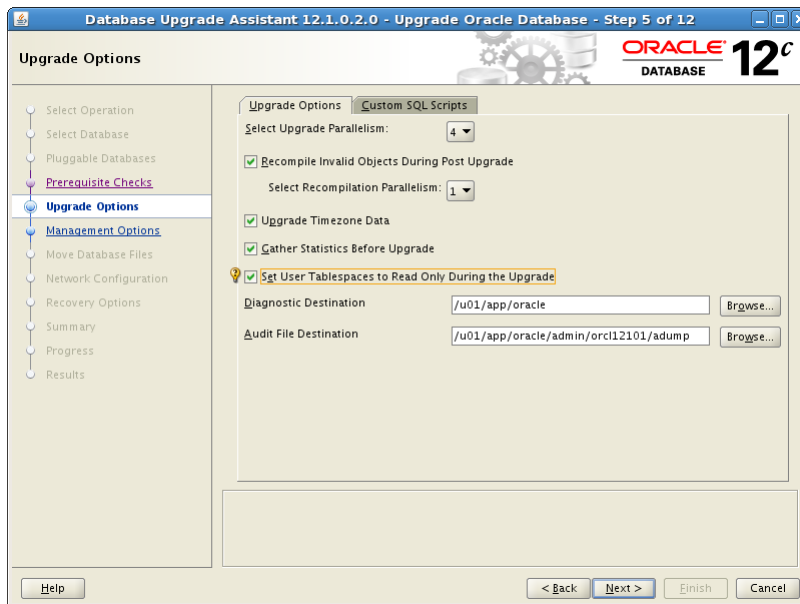
When you select the validation result, the Fixable column displays whether the result can be fixed or not. The Action drop-down list shows actions you can take. For example, select **Fix** for DBUA to run a script or command to fix the problem.

When you select a result in the Validation column, DBUA displays information about the result in the bottom area of the page.

- Click the link for **more details** in the text information area. The Validation Details box appears with more information.
- If there are validation errors or warnings and these are fixable, you can select an Action to take.

Click **Next**.

The Upgrade Options Configuration page appears with the Upgrade Options tab selected.



5. The Upgrade Options page with the Upgrade Options tab selected provides the following options:

Select Upgrade Parallelism

The Upgrade Parallelism section enables the degree of parallelism for the upgrade process. This option reduces the time needed to perform the upgrade, based on the number of CPUs available to handle the running of scripts and processes simultaneously.

By default, DBUA sets Upgrade Parallelism to the number of CPUs or 2 if the number of CPUs is less than 4. You can adjust this default value by selecting a new value from the Degree of Parallelism drop-down list.

Recompile Invalid Objects During Post Upgrade

Recompile all invalid PL/SQL modules after the upgrade is complete. Specify the parallelism for the recompilation of invalid objects during post upgrade. DBUA provides a recommended degree of Recompilation Parallelism, which it sets to one less than the number of CPU you have available. Taking advantage of parallelism can significantly reduce the upgrade time. If you do not have DBUA recompile invalid objects in its post-upgrade phase, then you must manually recompile invalid objects after the database is upgraded.

Upgrade Time Zone Data

Update the time zone data file for this release. If you do not select this option, then you must update the time zone configuration file manually after the upgrade.

Gather Statistics Before Upgrade

Reduce the overall time for the upgrade process by gathering statistics before upgrading.

Set User Tablespaces to Read Only During the Upgrade

Uncheck if you are upgrading a database in which you must transport tablespaces. Transportable tablespaces must have writable file headers.

Diagnostic Destination

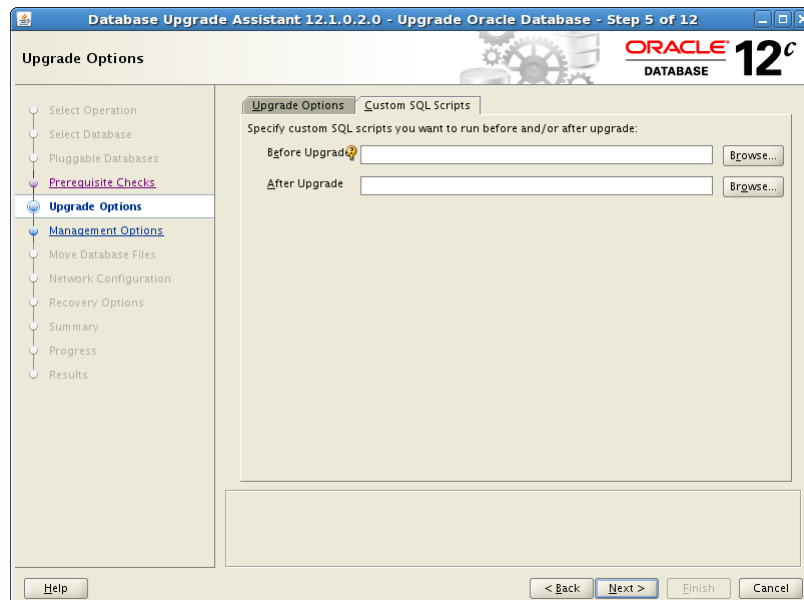
The location for output that DBUA creates for diagnostics. You can accept the default, enter a full path into the field, or click **Browse** to navigate to a location.

Audit File Destination

The location for DBUA to save audit files. Accept the default, or click **Browse** to navigate to a different location.

(Optional) Click the **Custom SQL Scripts** tab to specify custom SQL scripts that you would like to run before and after the upgrade.

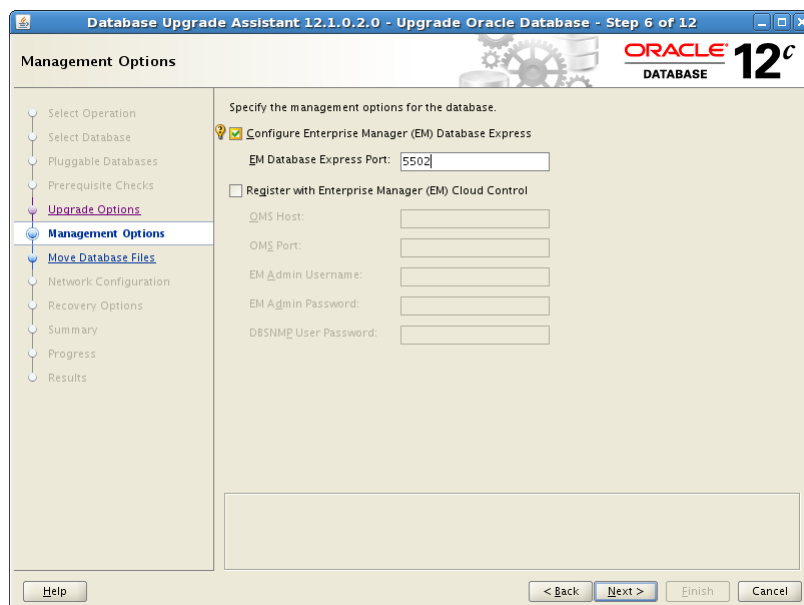
The Custom Scripts page appears. You may click **Next** without using the Custom SQL Scripts option.



- Optional: In the **Before Upgrade** and **After Upgrade** fields, browse to the location of any custom SQL scripts that you would like to run. You can specify either one or both, or leave the fields blank to skip this option.

Click **Next**.

The Management Options page appears.



7. In the Management Options page, select the management options:

- **Configure Enterprise Manager (EM) Database Express**

Oracle Enterprise Manager Database Express is a web-based database management application that is built into Oracle Database 12c. EM Express replaces the DB Control component that was available in releases 10g and 11g. Enter the EM Database Express Port number, for example 5502.

- **Register with Enterprise Manager (EM) Cloud Control**

Registering with Oracle Enterprise Manager Cloud Control adds the database and its related entities such as Listener, Oracle ASM disk groups, and Oracle Clusterware as managed targets.

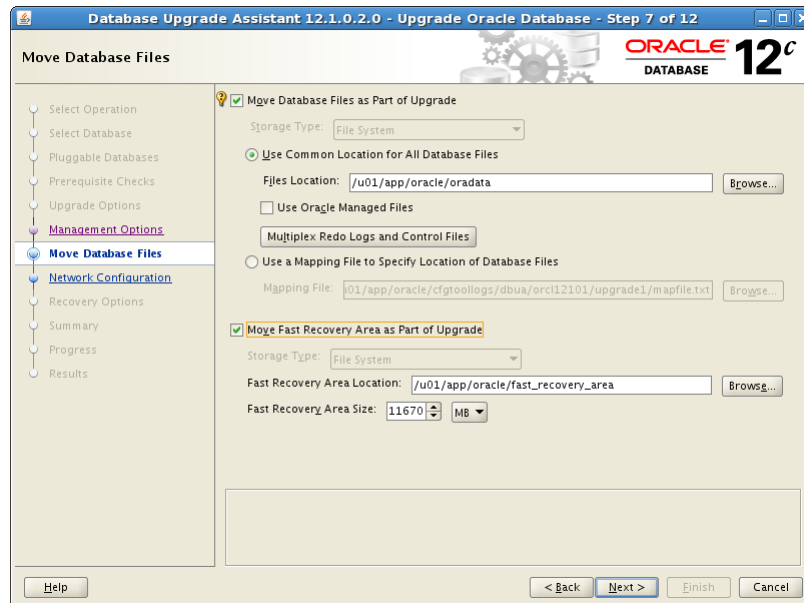
If you select this option, then you must provide information in the following fields:

- OMS Host
- OMS Port
- EM Admin Username
- EM Admin Password
- DBSNMP User Password

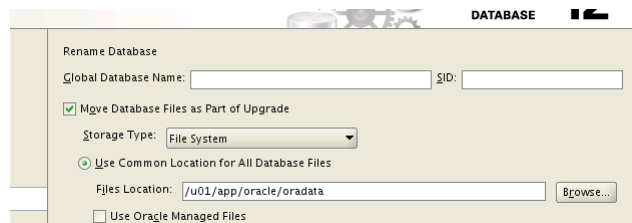
Click **Next**.

If you are upgrading a single-instance database or Oracle Express Edition (XE), then the Move Database Files page appears.

Note: If you are upgrading an Oracle Real Application Clusters database, then the Move Database Files page does not display. To continue with your Oracle RAC upgrade, you can skip the steps pertaining to moving database files and configuring the network.



If you are upgrading an Oracle Express database (Oracle XE), then the Move Database Files screen additionally includes Rename Database with the Global Database Name and SID fields as follows:



8. In the Move Database Files page, select an option:
 - **Move Database Files as Part of Upgrade**
 - **Move Fast Recovery Area as Part of Upgrade**

If you are upgrading an Oracle XE database, then you must provide the **Global Database Name** and **SID**. The rest of the Move Database Files options are the same as for Oracle Database.

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

9. If you choose **Move Database Files as Part of Upgrade**, then you must also configure **Storage Type** for the database files.

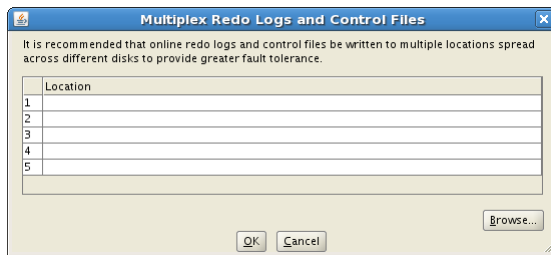
In the **Storage Type** drop-down list, select **File System** or **Oracle ASM**.

- If you select **File System**, then your database files are moved to the host file system.

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

See Also:

- *Oracle Grid Infrastructure Installation Guide* for information about installing and configuring Oracle ASM
- *Oracle Automatic Storage Management Administrator's Guide* for information about managing Oracle ASM instances with ASMCA
- You can choose either **Use Common Location for All Database Files** or **Use a Mapping File to Specify Location of Database Files**. Specify the location in the file location field or browse to the location.
- If you choose **Use Oracle Managed Files**, then click **Multiplex Redo Logs and Control Files**. The Multiplex Redo Logs and Control Files dialog box appears with location fields. Enter locations for online redo logs and control files to be written. Use multiple locations spread across different disks to provide greater fault tolerance.



10. If you choose **Move Fast Recovery Area as Part of Upgrade**, then you must also configure the storage type and location for the fast recovery area and specify the size to be allocated as described in this step.
 - The Move Database Files and Move Fast Recovery Area options are independent of each other. For example, you can choose to move database files to Oracle ASM and leave the fast recovery area on the file system.
 - When you choose to move the fast recovery area, DBUA does not physically move existing archived redo logs to a new location. Instead, DBUA sets the `DB_RECOVERY_FILE_DEST` and `DB_RECOVERY_FILE_DEST_SIZE` initialization parameters to the new location and new size when the database is started from the new Oracle home.
 - If an Oracle Express Edition database is being upgraded to Oracle Enterprise Edition, then you must configure a fast recovery area. If a fast recovery area is currently configured, then the current settings are retained but the page displays to enable you to override these values.

Storage Type

In the Storage Type list, select **File System** or **Oracle ASM**.

If you select File System, then your fast recovery area is on the host file system.

If you select Oracle Automatic Storage Management (Oracle ASM), then your fast recovery area is on Oracle ASM storage, which must currently exist on your system. If you do not have an Oracle ASM instance, then you can create one using

Automatic Storage Management Configuration Assistant (ASMCA) from the Grid Infrastructure home and then restart DBUA.

Fast Recovery Area Location

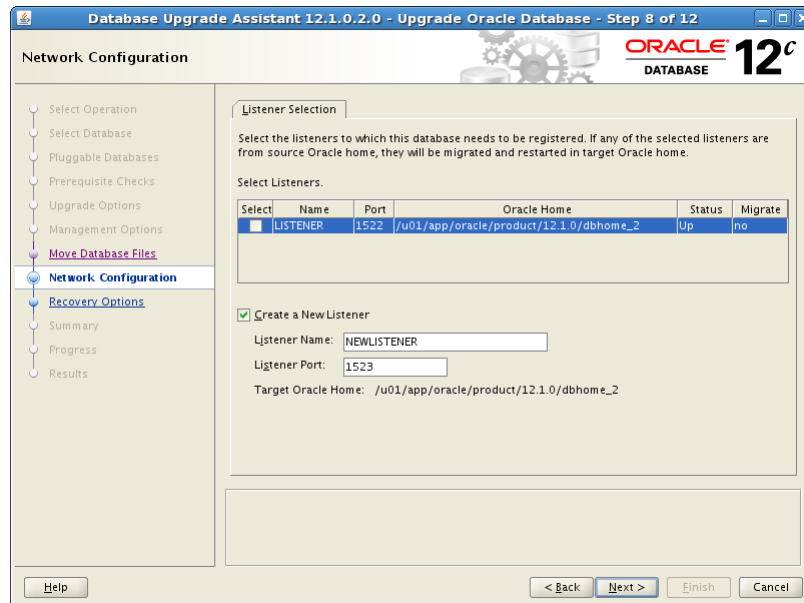
Browse to the location on the host file system, or on Oracle ASM storage.

Fast Recovery Area Size

Specify the size to allocate for the fast recovery area. The default is 1024 MB.

Click **Next**.

The Network Configuration page appears for listener selection.



11. The Listener Selection area of the Network Configuration page shows a table with: Name, Port, Oracle home, Status, and Migrate columns. To the left of the listener name is a box for selecting the listener.

Select one or more listeners from the source Oracle home to be migrated to the new upgraded Oracle home.

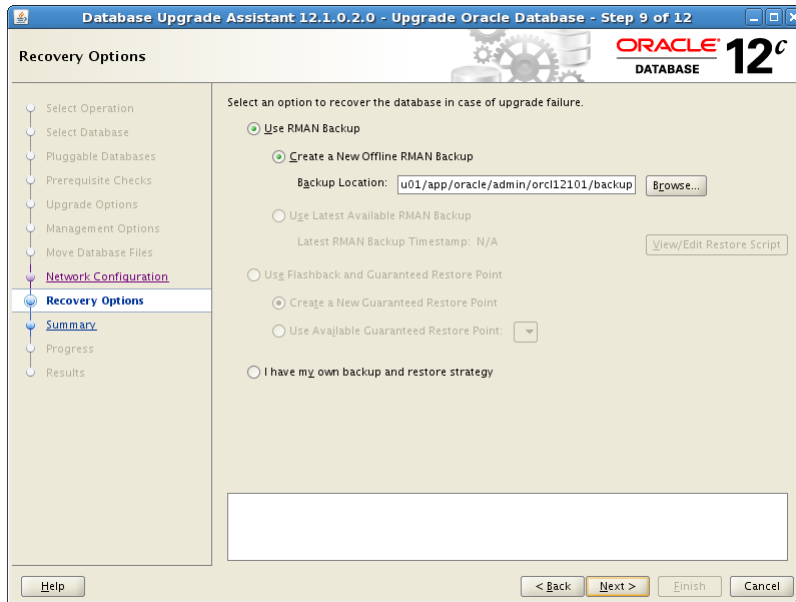
- DBUA adds the selected listener to the `listener.ora` file of the target Oracle home and starts it.
- DBUA removes the entry of the upgraded database from the old (source) `listener.ora` file.
- DBUA reloads the `listener.ora` file in both the source and target Oracle Database environments.

Note: If there are other databases registered with the same listener, then their new client connection requests may be affected during listener migration.

Select **Create a New Listener** to create a new listener. Provide the name and port number.

Click **Next**.

DBUA displays the Recovery Options page. DBUA performs the listener migration during the pre-upgrade steps.



12. In the Recovery Options page, select the recovery method to use in case the upgrade process encounters a problem.

The following recovery options and configurations are available:

- **Use RMAN Backup**
- **Use Flashback and Guaranteed Restore Point**
- **I have my own backup and restore strategy**

Use RMAN Backup

If you select **Create a New Offline RMAN Backup**, then enter the full path for a location for the backup in the Backup Location field.

If you select **Use Latest Available RMAN Backup** but do not choose to create an RMAN backup before upgrading, then DBUA displays the time stamp for the latest RMAN backup that exists. You can click **Restore Script** next to the time stamp to select an existing script to run for restoring this backup.

Use Flashback and Guaranteed Restore Point

Select **Create a New Guaranteed Restore Point** to have DBUA create a restore point before DBUA enters the upgrade process.

If you previously enabled Flashback Database and configured a fast recovery area with a flashback retention target, then you can select **Use Available Guaranteed Restore Point** and pick the named SCN from the drop-down list. Your current settings for the restore point are retained. DBUA displays a page to permit you to override these values if needed.

I have my own backup and restore strategy

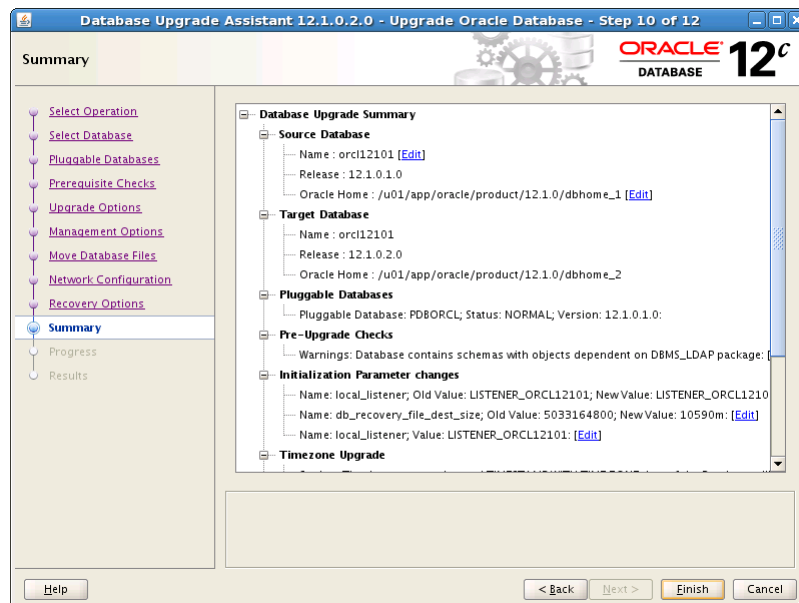
Select this option only if you used your own backup procedure to back up the database. In this case, **Restore** restores only the original database settings. To restore the database itself, you must restore the backup you created with your own backup utilities.

Note: The database you are upgrading must be release 11.1.0.7 or later in order to take advantage of Flashback and Guaranteed Restore Point, and this must be enabled in the source database.

See Also: *Oracle Database Backup and Recovery User's Guide* for more information on using Flashback Database and guaranteed restore points

Click **Next**.

The Database Upgrade Summary page appears.



13. The Summary page shows information about the upgrade before it starts. Scroll down the list to see all the details. For example:

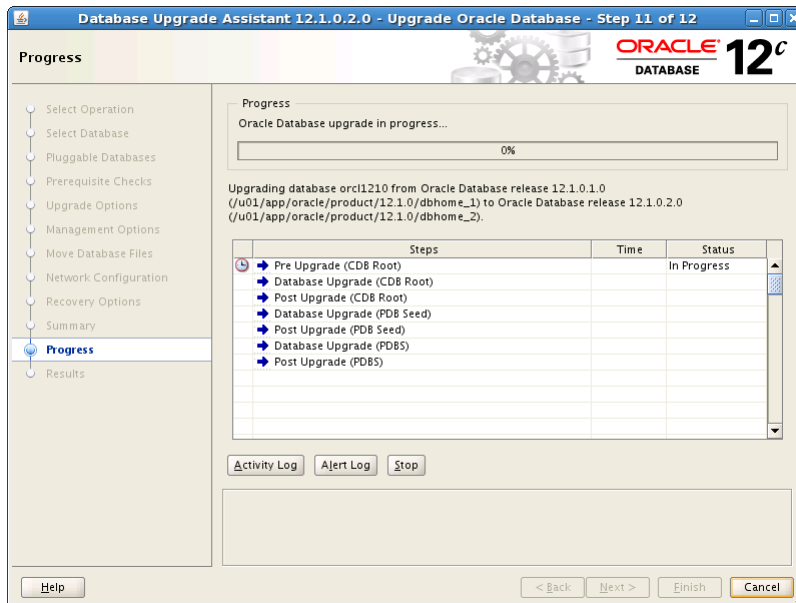
- Source Database
- Target Database
- Pluggable Databases
- Pre-Upgrade Checks
- Initialization Parameters changes
- Timezone Upgrade

See Also: "[Setting the COMPATIBLE Initialization Parameter](#)" on page 4-24 for information about setting the COMPATIBLE initialization parameter after the upgrade

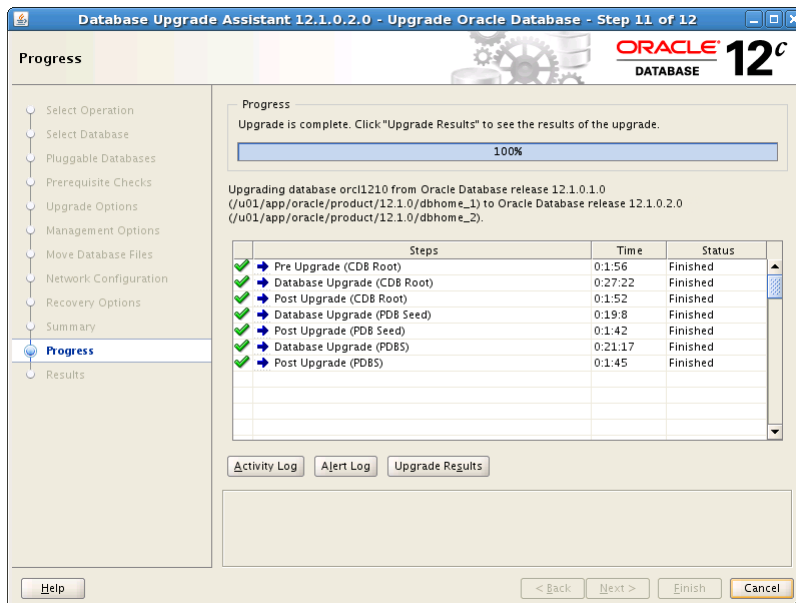
Check all of the details. Then click **Back** or **Finish** as follows:

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

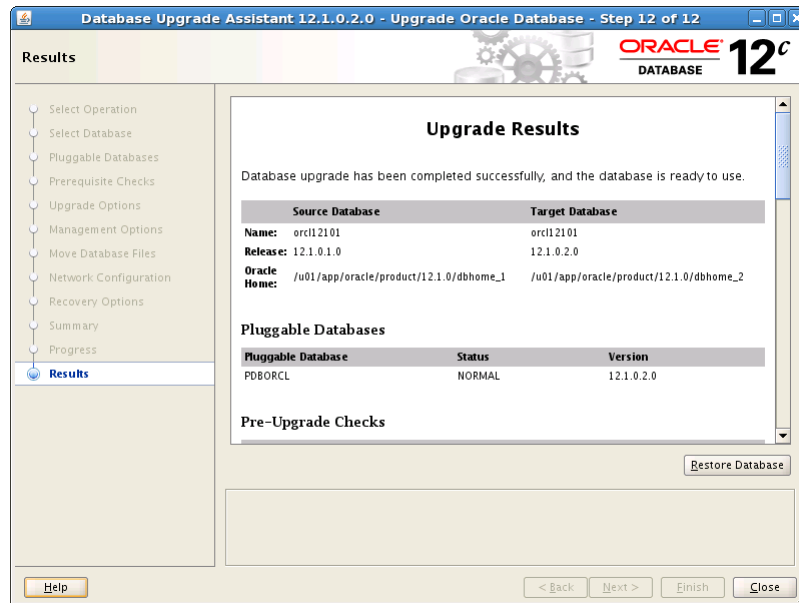
The Progress page displays with the progress bar and DBUA begins the upgrade. The Progress page also shows a table with the steps being performed, the time duration, and the status as the upgrade proceeds. DBUA provides a Stop button in case you must cancel the upgrade at this point.



When the upgrade has progressed through finishing the upgrade of the CDB root and each PDB seed, the Progress page marks the status Finished. You can click Activity Log, Alert Log, and Upgrade Results to view more information.



- After the upgrade has completed, the Upgrade Results page displays a description of the original database and the upgraded database and shows the changes made to the initialization parameters. The page also shows the directory where various log files are stored after the upgrade, and pluggable databases. Scroll down to see more details about pre-upgrade checks.



15. Optional: Examine the log files to obtain more details about the upgrade process. The DBUA log files are located under `/ORACLE_BASE/cfgtoollogs/dbua/logs`.

Note: An HTML version of the Upgrade Results is also saved in the log files directory. You can click the links in this HTML page to view the log pages in your browser.

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

16. Complete the procedures described in [Chapter 4, "Post-Upgrade Tasks for Oracle Database"](#).

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 security settings for Oracle Database 12c are in place, then passwords must be at least eight characters, and passwords such as `welcome` and `oracle` are not allowed. See *Oracle Database Security Guide* for information about configuring authentication.

3.2.4 About Windows Authentication When Upgrading

On Windows systems, users who connect with the SYSDBA privilege can take advantage of Windows native authentication. If these users work with Oracle Database using their domain accounts, then you must explicitly grant them local administrative privileges and ORA_DBA membership.

Unlike earlier releases, it is not mandatory that the Oracle Database services run using the privileges of a Local System account. You can run Oracle Database services using the privileges of a Windows user account. To authenticate to the database, you can use the Oracle Database Windows-specific operating system groups.

See Also:

- *Oracle Database Security Guide* for information on authenticating database administrators by using the operating system
- *Oracle Database Platform Guide for Microsoft Windows* for information on authenticating database users with Windows

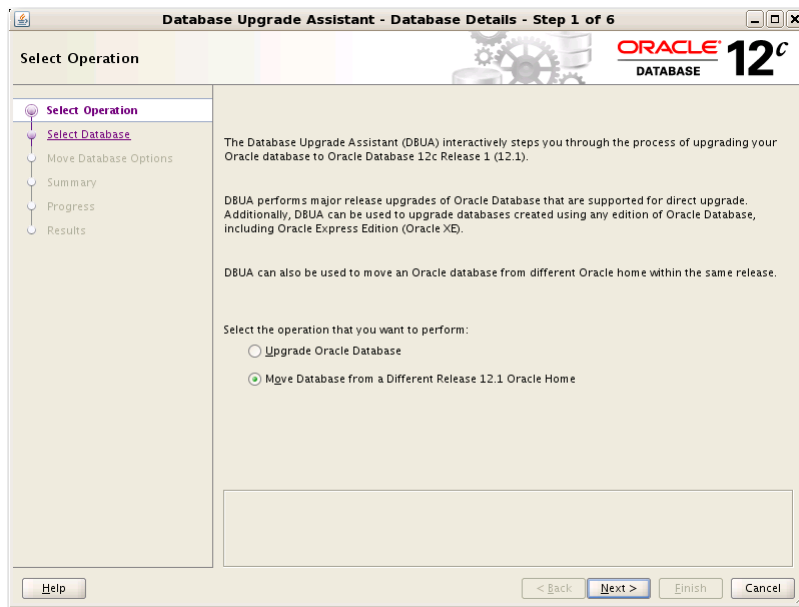
3.2.5 Moving a Database from an Existing 12c Oracle Home

With DBUA you can migrate an Oracle Database 12c database from an existing 12c Oracle home to another 12c Oracle home.

To move a database from an existing 12c home to a new 12c Oracle home:

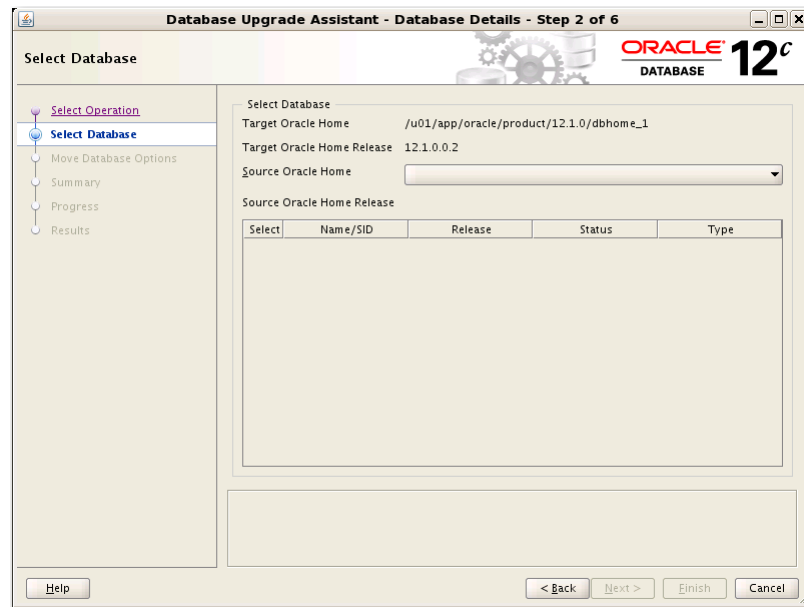
1. Start DBUA as described in ["Using DBUA to Upgrade the Database on Linux, UNIX, and Windows Systems"](#) on page 8.

DBUA starts with the The Select Operation page.



2. In the Select Operation page, choose **Move Database from a Different Release 12.1 Oracle Home** and click **Next**.

The Select Database page appears.



3. In the Select Database page, select the database you want to move to the new Oracle home. Only databases under a release 12c Oracle home appear. If you do not have an Oracle Database 12c database, then no database is available for the move operation.

Click **Next**. The Move Database Options page appears.

4. In the Move Database Options page, you can specify a different location for the diagnostic log files that DBUA creates, and you can optionally specify your custom SQL scripts, if any, to run after moving the database.

Click **Next**. The Database Move Summary page appears.

5. Review the summary for the move operation and click **Next**. The Progress page appears showing the processes for DBUA moving the database.

6. Click **Finish** when the move operation completes.

3.2.6 Using DBUA in Silent Mode to Upgrade Oracle Database

When invoked with the `-silent` command line option, DBUA operates in silent mode. In silent mode, DBUA does not present a user interface. DBUA writes messages (including information, errors, and warnings) to a log file in `ORACLE_HOME/cfgtoollogs/dbua/SID/upgraden`, where *n* represents the consecutive number for the upgrade that DBUA has run. Oracle strongly recommends that you read the resulting DBUA log files to ensure a successful upgrade.

To upgrade a database with DBUA in silent mode

- Issue the following command

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

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

3.2.6.1 Oracle DBUA Command Line Options for Silent Mode

Database Upgrade Assistant (DBUA) supports command line options when run in silent mode. [Table 3–1](#) describes the various options and the corresponding parameters that are supported by DBUA.

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

Table 3–1 DBUA Command Line Options

| Option | Description |
|---|--|
| <code>-silent</code> | Specifies that DBUA should operate in silent mode. |
| <code>-sid SID</code> | Specifies the system identifier (SID) of the database to upgrade. |
| <code>-oracleHome home_name</code> | Specifies the Oracle Database home directory of the database to upgrade. |
| <code>-oracleBase base_name</code> | Specifies the Oracle Database base directory of the database to upgrade. |
| <code>-listeners</code> | To create a listener in the Oracle home of the new release, specify <code>listenerName:lsnrPort</code> . |
| <code>-createListeners</code> | To register the database with existing listeners, pass listeners by comma separated <code>listenerName:Oracle Home</code> . Listeners from the earlier release home are migrated to the new release home. By specifying <code>-listeners lsnrName1, lsnrName2</code> , DBUA searches the specified listeners from the Grid Infrastructure home (if Grid Infrastructure is configured), the target home, and the source home. |
| <code>-diagnosticDest diagnostic_destination</code> | Specifies the default location to store Oracle trace and diagnostic files. It replaces the initialization parameter settings for background dump destination and user dump destination from earlier releases. |
| <code>-sysDBAUserName SYSDBA_user</code> | Specifies a user with SYSDBA privileges. |
| <code>-sysDBAPassword SYSDBA_pwd</code> | Specifies the password for <code>SYSDBA_user</code> . |
| <code>-autoextendFiles</code> | Autoextends database files during the upgrade. Data files revert to their original autoextend settings after the upgrade. |
| <code>-newGlobalDbName db_name</code> | Specifies a new global database name. This option applies only if you are moving data files or upgrading an Oracle XE database. |
| <code>-newSid new_SID</code> | 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. |
| <code>-upgradeTimezone</code> | Upgrades the time zone file version for the database. |

Table 3–1 (Cont.) DBUA Command Line Options

| Option | Description |
|---|---|
| -generateMapFile | Applies only if you are moving data files or upgrading an Oracle XE database. If you specify this option, then DBUA only generates a database map file in the log location and then exits. |
| -useASM | Applies only if you are moving data files or upgrading an Oracle XE database. If the database to upgrade has an Oracle ASM instance, then this option tells DBUA to use it for the upgrade. |
| -commonFileLocation <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. |
| -omfLocation <i>omf_area</i> | Specifies a database area for Oracle Managed Files. This option applies only if you are moving data files or upgrading an Oracle XE database. |
| -databaseMapFile <i>map_file_name</i> | Specifies the full name of the map file to map database files. This option applies only if you are moving data files or upgrading an Oracle XE database. |
| -newRecoveryArea <i>recover_area</i> | Specifies the recovery area for a database that is moved during upgrade. This option applies only if you are moving data files or upgrading an Oracle XE database. |
| -newRecoveryAreaSize <i>recover_size</i> | Specifies the recovery area size (MB) for a database that is moved during upgrade. This option applies only if you are moving data files or upgrading an Oracle XE database. |
| -apexAdminPassword <i>apex_pwd</i> | Specifies the password for the Application Express Administrator. |
| -disableUpgradeScriptLogging | Disables the detailed log generation for running SQL scripts during the upgrade process. This is enabled by default. To enable log generation, do not specify this option. |
| -backupLocation <i>directory</i> | Specifies a directory to back up your database before the upgrade starts. |
| -postUpgradeScripts <i>script</i> [, <i>script</i>] ... | Specifies a comma-delimited list of SQL scripts. Specify complete path names. The scripts are executed after the upgrade. |
| -initParam <i>parameter=value</i> [, <i>parameter=value</i>] ... | Specifies a comma-delimited list of initialization parameter values of the form <i>name=value</i> . |
| -disableArchiveLogMode | Turns off archiving and flashback logging for the duration of the upgrade. |
| -recompile_invalid_objects <i>true false</i> | When you specify TRUE for this option, DBUA recompiles all invalid PL/SQL modules immediately after the upgrade is performed. |
| -degree_of_parallelism <i>number</i> | Specifies the level to be used for parallel recompilation. |
| -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 3–1 (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. |

3.2.6.2 DBUA Command Line Syntax for Silent Mode

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

```
dbua [ -silent ] [ -sid SID ] [-oracleHome home_name] [-oracleBase base_name]
[-diagnosticDest diagnostic_destination]
[-sysDBAUserName SYSDBA_user] [-sysDBAPassword SYSDBA_pwd]
[-upgradeASM] [-autoextendFiles] [-newGlobalDbName db_name] [-newSid new_SID]
[-generateMapFile] [-useASM] [-commonFileLocation common_files]
[-omfLocation omf_area] [-databaseMapFile map_file_name]
[-newRecoveryArea recover_area] [-newRecoveryAreaSize recover_size]
[-apexAdminPassword apex_pwd] [-disableUpgradeScriptLogging ]
[-backupLocation directory]
[-sysauxTablespace -datafileName name -datafileSize size
-datafileSizeNext size -datafileSizeMax size]
[-postUpgradeScripts script [, script ] ... ]
[-initParam parameter=value [, parameter=value ] ... ]
[-disableArchiveLogMode] [-recompile_invalid_objects true | false]
[-degree_of_parallelism number]
[-recoveryAreaDestination directory] [-h|-help]
```

3.3 About the Parallel Upgrade Utility for Oracle Database

Oracle Database 12c introduces the Parallel Upgrade Utility (`catctl.pl`), which enables components that do not have to be upgraded in a specific order to be upgraded at the same time, taking full advantage of CPU capacity. Oracle continues to make improvements to the upgrade process to simplify both manual upgrades and upgrades performed with the Database Upgrade Assistant (DBUA). DBUA and the manual upgrade procedures take advantage of the new Parallel Upgrade Utility. The Parallel Upgrade Utility reduces the total amount of time it takes to perform an upgrade by loading the database dictionary in parallel using multiple SQL processes to upgrade the database.

Related topics about the new Parallel Upgrade Utility (`catctl.pl`) are as follows:

- [General Steps for Running the New Parallel Upgrade Utility \(catctl.pl\)](#)
- [Parallel Upgrade Utility \(catctl.pl\) Parameters](#)
- [Example for Using the Parallel Upgrade Utility](#)

3.3.1 General Steps for Running the New Parallel Upgrade Utility (catctl.pl)

The Parallel Upgrade Utility loads the data dictionary and components in parallel, thus reducing the overall upgrade time. Before invoking the Parallel Upgrade Utility, `catctl.pl`, follow the procedures for backing up your database that you would normally do before upgrading. Also, as a prerequisite, you must run the Pre-Upgrade Information Tool (`preupgrd.sql`) to identify any problems that a database administrator must address before the upgrade proceeds.

The detailed procedure for manually running `catctl.pl` is described in ["Manually Upgrading a Multitenant Container Oracle Database \(CDB\)"](#) on page 35. The general steps for upgrading your database with the Parallel Upgrade Utility are as follows:

1. Back up your current database as described in ["Backing Up Oracle Database for Upgrading"](#) on page 2-38.
2. Install the Oracle Database 12c software for the new release as described in ["Installing the New Oracle Database Software"](#) on page 2-25.
3. Ensure the Pre-Upgrade Information Tool (`preupgrd.sql`) has been run on the source database as described in ["About the Pre-Upgrade Information Tool for Oracle Database"](#) on page 2-27.

4. From the `ORACLE_HOME/rdbms/admin` directory, start SQL*Plus.

5. Connect to the database to be upgraded using an account with DBA privileges:

```
CONNECT / AS SYSDBA
```

6. Start the database in upgrade mode.

For a multitenant container database (CDB):

```
SQL> alter pluggable database all open upgrade;
```

For a non-CDB:

```
SQL> startup upgrade
```

Note: The `UPGRADE` keyword performs operations that prepare the environment for the upgrade.

You might be required to use the `PFILE` option in your startup command to specify the location of your initialization parameter file. See *Oracle Database Administrator's Guide* for information about specifying initialization parameters at startup and the initialization parameter file.

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

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

7. Exit SQL*Plus.
8. Run `catctl.pl` from the new Oracle home. See ["Parallel Upgrade Utility \(catctl.pl\) Parameters"](#) on page 28 for the parameters available for running `catctl.pl`.

To run `catctl.pl` on Linux:

```
cd $ORACLE_HOME/rdbms/admin
$ORACLE_HOME/perl/bin/perl catctl.pl catupgrd.sql -l
```

To run `catctl.pl` on Windows:

```
cd %ORACLE_HOME%\rdbms\admin
%ORACLE_HOME%\perl\bin\perl catctl.pl catupgrd.sql -l
```

3.3.2 Parallel Upgrade Utility (catctl.pl) Parameters

The parameters listed in [Table 3–2](#) are available when invoking `catctl.pl`. See ["Example for Using the Parallel Upgrade Utility"](#) on page 29 for more information on how `catctl.pl` processes the upgrade phases and for an example.

Table 3–2 *catctl.pl* Parameters

| Parameter | Description |
|-----------|---|
| -c | <p>Specifies the PDBs that you want to upgrade.</p> <p>This parameter provides an inclusion list. For example, include the PDBs named here (skip all PDBs not named here):</p> <p>For Linux and UNIX (use single quotes):</p> <pre>-c 'PDB1 PDB2'</pre> <p>For Windows (use double quotes):</p> <pre>-c "PDB1 PDB2"</pre> |
| -C | <p>Specifies the PDBs that you do not want to upgrade.</p> <p>This parameter provides an exclusion list. For example, skip all PDBs named here (all other PDBs are upgraded):</p> <p>For Linux and UNIX (use single quotes):</p> <pre>-C 'PDB1 PDB2'</pre> <p>For Windows (use double quotes):</p> <pre>-C "PDB1 PDB2"</pre> <p>Note: <code>-c</code> and <code>-C</code> are mutually exclusive.</p> |
| -d | Specifies the location of the directory containing the files to be processed. |
| -e | Sets echo OFF while running the scripts. The default is echo ON. |
| -i | Specifies an identifier to use when creating spool log files. |
| -l | <p>Specifies the location for the directory to use for spool log files. The default is the current working directory.</p> <p>Oracle strongly recommends that you do not write log files to the <code>\admin</code> directory.</p> |
| -M | <p>Keeps CDB\$ROOT in UPGRADE mode while the PDBs are upgraded.</p> <p>For non-CDBs, this parameter is ignored.</p> <p>During CDB upgrades, using this parameter setting places the CDB and all of its PDBs in upgrade mode, which may reduce total upgrade time. However, you cannot bring up any of the PDBs until the CDB and all of its PDBs are upgraded.</p> <p>By default, if you do not use the <code>-M</code> parameter setting, then CDB\$ROOT is upgraded and restarted in normal mode, and the normal background processes are started. As each PDB is upgraded, you can bring the PDB online while other PDBs are still being upgraded.</p> |

Table 3–2 (Cont.) catctl.pl Parameters

| Parameter | Description |
|-----------|--|
| -n | Specifies the number of processes to use for parallel operations. For non-CDBs, it specifies the number of SQL processors to use when upgrading the database. The maximum value is 8. The minimum value is 1. The default value is 4. For CDBs, multiple PDB upgrades are processed together. The number of PDBs upgraded concurrently is controlled by the value of the -n parameter. The default value is the number of CPUs divided by 2 (cpu_count/2). A cpu_count equal to 24 equates to 12 PDB upgrades running concurrently. The maximum value is 64. The minimum value is 4. The maximum PDB upgrades running concurrently is 32 (64/2), and the minimum value is 2 (4/2). |
| -N | Specifies the number of SQL processors to use when upgrading PDBs. For non-CDBs, this parameter is ignored. For CDBs, the maximum value is 8. The minimum value is 1. The default value is 2. |
| -P | Restarts the specified phase (skip successful phases on a rerun). |
| -P | Stop the specified phase. |
| -s | Names the SQL script to initialize sessions. |
| -S | Specify <i>serial</i> upgrade instead of parallel. |
| -u | Specifies username and prompts for password. |
| -y | Displays phases only. |
| -z | Turns on debug information for catcon.pm. |
| -Z | Turns on debug tracing information for catctl.pl. For example, to set the number to 1, enter -Z 1. |

3.3.3 Example for Using the Parallel Upgrade Utility

The `catctl.pl` Parallel Upgrade Utility is integrated with DBUA. However, for manual, command line upgrades, you can run `catctl.pl` with various parameters. For example, to run `catctl.pl` in serial mode instead of using parallel operations, specify the `-S` parameter. See [Table 3–2, "catctl.pl Parameters"](#) for the available parameters.

Example 3–1 Running catctl.pl with select parameters

If `-n 4` is used when invoking `catctl.pl`, then `catupgrd0.log`, `catupgrd1.log`, `catupgrd2.log`, and `catupgrd3.log` are created during the upgrade process. When evaluating if an upgrade succeeded or failed, all of the `catupgrd#.log` files must be checked. If `catctl.pl` is rerun, then the previous log files are overwritten unless you specify a different log directory.

```
cd $ORACLE_HOME/rdbms/admin
$ORACLE_HOME/perl/bin/perl catctl.pl -n 4 -l $ORACLE_HOME/diagnostics catupgrd.sql
```

3.4 Manually Upgrading a Non-CDB Oracle Database

After installing the software for Oracle Database 12c and preparing the new Oracle home, you can proceed with a manual, command-line upgrade rather than use DBUA.

The procedure in this section provides steps for upgrading an Oracle Database that is **not** a multitenant container database (CDB). This procedure assumes that you installed the software for Oracle Database 12c, that you prepared the new Oracle home, and that you have run the Pre-Upgrade Information Tool. Refer to ["Installing the New Oracle Database Software"](#) on page 2-25 and ["Preparing the New Oracle Home for Upgrading"](#) on page 2-39 for additional procedures.

See Also: *Oracle Database Concepts* for an overview of multitenant architecture

To manually upgrade a non-CDB database:

1. If you have not done so, run the Pre-Upgrade Information Tool as described in ["About the Pre-Upgrade Information Tool for Oracle Database"](#) on page 2-27. Review the Pre-Upgrade Information tool output and correct all issues noted in the output before proceeding. Refer to ["Pre-Upgrade Information Tool Warnings and Recommendations for Oracle Database"](#) on page 2-29 for information about fixing any issues.
2. Back up the source database as described in ["Backing Up Oracle Database for Upgrading"](#) on page 2-38.
3. If you have not done so, prepare the new Oracle home as described in ["Preparing the New Oracle Home for Upgrading"](#) on page 2-39.
4. Shut down the database.

```
SQL> SHUTDOWN IMMEDIATE
```

5. 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. Refer to ["Variables for using ORADIM When Upgrading Oracle Database on Windows"](#) on page 34 for more information. Refer to your platform guide for a complete list of the ORADIM syntax and commands.

If your SID is ORCL, then enter the following command. Substitute the actual name for your SID.

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

- c. Create the service for Oracle Database 12c at a command prompt using the ORADIM command of the new Oracle Database release as shown in the following example:

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

Most Oracle Database services log on to the system using the privileges of the Oracle Home User. The service runs with the privileges of this user. The ORADIM command prompts you for the password to this user account. You can specify other options using ORADIM.

In this example, if your SID is ORCL, your password (SYSPWD) is TWxy5791, the maximum number of users (MAXUSERS) is 10, and the ORACLE_HOME

directory is C:\ORACLE\PRODUCT\11.2.0\DB, then enter the following command:

```
C:\> ORADIM -NEW -SID ORCL -SYSPWD Twxy5791 -MAXUSERS 10
-STARTMODE AUTO -PFILE C:\ORACLE\PRODUCT\11.2.0\DB\DATABASE\INITORCL.ORA
```

ORADIM writes a log file to the ORACLE_HOME\database directory. The log file contains the name of the PDB in the multitenant database.

6. If your operating system is Linux or UNIX, then perform the following checks:
 - a. Your ORACLE_SID is set correctly
 - b. The oratab file points to the Oracle home for Oracle Database 12c
 - c. The following environment variables point to the Oracle Database 12c directories:
 - ORACLE_HOME
 - PATH
 - d. Any scripts that clients use to set the ORACLE_HOME value must point to the new Oracle home.

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

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

7. Log in to the system as the owner of the Oracle home under the new Oracle Database 12c.
8. From the ORACLE_HOME/rdbms/admin directory, start SQL*Plus.
9. Connect to the database to be upgraded using an account with DBA privileges:


```
SQL> CONNECT / AS SYSDBA
```
10. Start the instance by issuing the following command in SQL*Plus:


```
SQL> startup upgrade pfile=pfile_name
```

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.

Specify the location of your initialization parameter file PFILE. See *Oracle Database Administrator's Guide* for information about specifying initialization parameters at startup and the initialization parameter file.

If errors appear listing desupported initialization parameters, then make a note of the desupported initialization parameters and continue with the upgrade. Remove

the desupported initialization parameters the next time you shut down the database.

11. Exit SQL*Plus before proceeding to the next step.
12. Run the `catctl.pl` script from the new Oracle home as described in this step. The Parallel Upgrade Utility, `catctl.pl`, provides parallel upgrade options that reduce downtime.

To run `catctl.pl` on Linux:

```
cd $ORACLE_HOME/rdbms/admin
$ORACLE_HOME/perl/bin/perl catctl.pl catupgrd.sql
```

To run `catctl.pl` on Windows:

```
cd %ORACLE_HOME%\rdbms\admin
%ORACLE_HOME%\perl\bin\perl catctl.pl catupgrd.sql
```

Note:

- You can use the `-d` option to specify the directory that contains the files to be processed, and `-l` to specify the directory to use for spool log files. You can use `catctl.pl -S` to run the upgrade processes serially. Refer to [Table 3–2, "catctl.pl Parameters"](#) for all the options.
- If you choose to run the `catupgrd.sql` script on a non-CDB, then doing so now requires an additional input parameter as follows:

```
SQL> @catupgrd.sql PARALLEL=NO
```

See Also: ["General Steps for Running the New Parallel Upgrade Utility \(catctl.pl\)"](#) on page 26 for information about `catctl.pl` parameters and options

13. The upgraded database is shut down after running `catctl.pl`. Restart the instance to reinitialize the system parameters for normal operation.

```
SQL> STARTUP
```

This restart, following the database shutdown performed as part of the `catctl.pl` 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.

Note: If you encountered a message listing desupported initialization parameters when you started the database in Step 11, then remove the desupported initialization parameters from the parameter file before restarting. If necessary, convert the `SPFILE` to a `PFILE` so you can edit the file to delete parameters. See *Oracle Database Administrator's Guide* for information about specifying initialization parameters at startup and the initialization parameter file.

- 14. Important:** The `catuppst.sql` script is run as part of the upgrade unless the upgrade returns errors during the process. Check the log file for "Rem BEGIN `catuppst.sql`" to verify that `catuppst.sql` ran during the upgrade process.

If `catuppst.sql` has not run, then proceed to run `catuppst.sql` as shown in this step. Warning messages are also displayed when running `catctl.pl` indicating that `catuppst.sql` was not run during the upgrade.

Run `catuppst.sql` as follows:

```
$ORACLE_HOME/perl/bin/perl catcon.pl -n 1 -e -b catuppst -d ''.'''
catuppst.sql
```

The log file `catuppst0.log` will be generated with the results of the post upgrade.

- 15.** Run `catcon.pl` to invoke `utlrbp.sql` to recompile any remaining stored PL/SQL and Java code. Use the following syntax:

```
$ORACLE_HOME/perl/bin/perl catcon.pl -n 1 -e -b utlrbp -d ''.''' utlrbp.sql
```

The log file `utlrbp0.log` is generated with the results of the recompilations.

- 16.** Run `postupgrade_fixups.sql`. See "[Run the postupgrade_fixups.sql Script](#)" on page 4-10.
- 17.** Run `utlu121s.sql` to verify that all issues have been fixed.

```
SQL> @rdbms/admin/utlu121s.sql
```

The log file `utlu121s0.log` will be generated with the upgrade results.

To see information about the state of the database, you can run `utlu121s.sql` as many times as you want, any time *after* completing the upgrade. If the `utlu121s.sql` script returns errors or shows components that are not `VALID` or not the most recent release, then see "[Troubleshooting the Upgrade for Oracle Database](#)" on page 42 for more information.

- 18.** Verify that all expected packages and classes are valid. (This step assumes that you previously ran `preupgrd.sql` before starting the upgrade.)

```
$ORACLE_HOME/perl/bin/perl catcon.pl -n 1 -e -b utluiobj -d ''.'''
utluiobj.sql
```

The log file `utluiobj0.log` will be generated with the results of the verification.

Note: If the Pre-upgrade Information Tool detected `INVALID` objects and populated the `registry$sys_inv_objs` and `registry$nonsys_inv_objs` tables, then execute `ORACLE_HOME/rdbms/admin/utluiobj.sql` to display only those objects which are newly invalid because of the upgrade process. The `utluiobj.sql` script only displays objects that are now `INVALID` but which were `VALID` before the upgrade. See "[Using the Pre-Upgrade Information Tool \(preupgrd.sql\)](#)" on page 2-27..

- 19.** Exit SQL*Plus if you are still in it.
- 20.** If you are upgrading a cluster database from releases 10.2, 11.1, or 11.2, then upgrade the database configuration in Oracle Clusterware using the following command:

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

where *db-unique-name* is the database name assigned to it (not the instance name), and *oraclehome* is the Oracle home location in which the database is being upgraded. The *SRVCTL* utility supports long GNU-style options in addition to short CLI options used in earlier releases.

Your database is now upgraded to Oracle Database 12c. You are ready to complete the procedures described in [Chapter 4, "Post-Upgrade Tasks for Oracle Database"](#).

Note: Oracle Warehouse Builder (OWB) components are not upgraded as part of the Oracle Database upgrade process. Oracle Warehouse Builder is not installed as part of Oracle Database 12c. See ["Requirement for Upgrading Databases that Use Oracle Warehouse Builder \(OWB\)"](#) on page 2-23 for more information.

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

3.4.1 Variables for using ORADIM When Upgrading Oracle Database on Windows

On Windows platforms, *ORADIM* provides a command-line interface to manually perform administrative tasks for Windows databases and services. Database Configuration Assistant (DBCA) provides a graphical user interface to perform the same tasks. The variables for *ORADIM* that you must know about when upgrading Oracle Database include the SID of the database you are upgrading, the new Oracle home location, and the password for the new database instance. Also, *ORADIM* writes a log file to the *ORACLE_HOME\database* directory.

The following table describes the variables for using *ORADIM* when upgrading:

| ORADIM 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 12c database instance. This is the password for the user connected with <i>SYSDBA</i> privileges. The <i>-SYSPWD</i> option is required. The default Oracle Database 12c security settings require that passwords must be at least eight 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 location for Oracle Database 12c. Ensure that you specify the full path name with the <i>-PFILE</i> option, including the drive letter of the Oracle home location. See <i>Oracle Database Administrator's Guide</i> for information about specifying initialization parameters at startup and the initialization parameter file. |

See Also: *Oracle Database Platform Guide for Microsoft Windows* for complete information about using ORADIM to administer a database instance

3.5 Manually Upgrading a Multitenant Container Oracle Database (CDB)

Oracle Database 12c introduces multitenant architecture, which enables an Oracle database to function as a multitenant container database (CDB) with pluggable databases. You can upgrade the CDB, thereby upgrading all the pluggable databases in the CDB at the same time. After installing the software for Oracle Database 12c and preparing the new Oracle home, you can proceed with a manual, command-line upgrade rather than use DBUA.

The procedure in this section provides steps for upgrading a CDB. This procedure assumes that you installed the software for Oracle Database 12c, that you prepared the new Oracle home, and that you have run the Pre-Upgrade Information Tool. Refer to ["Installing the New Oracle Database Software"](#) on page 2-25 and ["Preparing the New Oracle Home for Upgrading"](#) on page 2-39 for additional procedures.

To manually upgrade a CDB database:

1. If you have not done so, run the Pre-Upgrade Information Tool as described in ["About the Pre-Upgrade Information Tool for Oracle Database"](#) on page 2-27. Review the Pre-Upgrade Information tool output and correct all issues noted in the output before proceeding. Refer to ["Pre-Upgrade Information Tool Warnings and Recommendations for Oracle Database"](#) on page 2-29 for information about fixing any issues.
2. Back up the source database as described in ["Backing Up Oracle Database for Upgrading"](#) on page 2-38.
3. If you have not done so, prepare the new Oracle home as described in ["Preparing the New Oracle Home for Upgrading"](#) on page 2-39.
4. Shut down the database. (The syntax is the same for a non-CDB and a CDB.)

```
SQL> SHUTDOWN IMMEDIATE
```

Note: To close a PDB, you can specify it from the CDB root: `alter pluggable database PDBname close.`

5. 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. Refer to ["Variables for using ORADIM When Upgrading Oracle Database on Windows"](#) on page 34 for more information. Also refer to your platform guide for a list the ORADIM syntax and commands.

If your `SID` is `ORCL`, then enter the following command. Substitute the actual name for your `SID`.

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

- c. Create the service for Oracle Database 12c at a command prompt using the ORADIM command of the new Oracle Database release as shown in the following example:

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

Most Oracle Database services log on to the system using the privileges of the Oracle Home User. The service runs with the privileges of this user. The ORADIM command prompts you for the password to this user account. You can specify other options using ORADIM.

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

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

ORADIM writes a log file to the ORACLE_HOME\database directory. The log file contains the name of the PDB in the multitenant database.

6. If your operating system is Linux or UNIX, then perform the following checks:
 - a. Your ORACLE_SID is set correctly
 - b. The oratab file points to the Oracle home for Oracle Database 12c
 - c. The following environment variables point to the Oracle Database 12c directories:
 - ORACLE_HOME
 - PATH
 - d. Any scripts that clients use to set the ORACLE_HOME value must point to the new Oracle home.

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

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

7. Log in to the system as the owner of the Oracle home under the new Oracle Database 12c.
8. From the ORACLE_HOME/rdbms/admin directory, start SQL*Plus.
9. Connect to the database to be upgraded using an account with DBA privileges:


```
SQL> CONNECT / AS SYSDBA
```
10. Start the CDB in upgrade mode:


```
SQL> startup upgrade
```
11. Start the instance by issuing the following command in SQL*Plus:


```
SQL> alter pluggable database all open 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.

Specify the location of your initialization parameter file PFILE. See *Oracle Database Administrator's Guide* for information about specifying initialization parameters at startup and the initialization parameter file.

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

12. Exit SQL*Plus before proceeding to the next step.
13. To upgrade an entire CDB, Run the `catctl.pl` script from the new Oracle home as described in this step. The Parallel Upgrade Utility, `catctl.pl`, provides parallel upgrade options that reduce downtime.

To run `catctl.pl` on Linux:

```
cd $ORACLE_HOME/rdbms/admin
$ORACLE_HOME/perl/bin/perl catctl.pl catupgrd.sql
```

To run `catctl.pl` on Windows:

```
cd %ORACLE_HOME%\rdbms\admin
%ORACLE_HOME%\perl\bin\perl catctl.pl catupgrd.sql
```

Note:

- You can use the `-d` option to specify the directory that contains the files to be processed, and `-l` to specify the directory to use for spool log files. You can use `catctl.pl -S` to run the upgrade processes serially. Refer to [Table 3–2, "catctl.pl Parameters"](#) for all the options.
- If you choose to run the `catupgrd.sql` script on a non-CDB, doing so now requires an additional input parameter as follows:

```
SQL> @catupgrd.sql PARALLEL=NO
```

- When upgrading an entire CDB, upgrade aborts if there are errors in the CDB root or PDB seed.
- To run `catupgrd.sql` on a CDB, you must run it in the CDB\$ROOT and PDB\$SEED for all the PDBs. For example:

```
SQL> ALTER SESSION SET CONTAINER = "CDB$ROOT"
SQL> catupgrd.sql PARALLEL=NO
SQL> ALTER SESSION SET CONTAINER = "PDB$SEED"
SQL> catupgrd.sql PARALLEL=NO
SQL> ALTER SESSION SET CONTAINER = "PDB1"
SQL> catupgrd.sql PARALLEL=NO
```

See Also: ["General Steps for Running the New Parallel Upgrade Utility \(catctl.pl\)"](#) on page 26 for information about `catctl.pl` parameters and options

14. To upgrade a subset of PDBs within a CDB, you can specify either an inclusion list or an exclusion list as shown in the examples in this step.

- a. Inclusion list to upgrade PDB1 only:

```
cd $ORACLE_HOME/rdbms/admin
$ORACLE_HOME/perl/bin/perl catctl.pl -c 'PDB1' catupgrd.sql
```

- b. Exclusion list to upgrade everything in the CDB except PDB1:

```
cd $ORACLE_HOME/rdbms/admin
$ORACLE_HOME/perl/bin/perl catctl.pl -C 'PDB1' catupgrd.sql
```

Note: You can also upgrade a specific PDB by itself by unplugging it and then plugging it back into the CDB. This is described in ["Upgrading a Pluggable Database \(PDB\)"](#) on page 4.

For Windows, the `-C` option must be specified with quotes around the CDB root name and PDB seed name, for example:

```
... -C "CDB$ROOT PDB$SEED"
```

15. **Important:** The `catuppst.sql` script is run as part of the upgrade process unless the upgrade returns errors during the process. Check the log file for "Started: `catuppst.sql`" to verify that `catuppst.sql` ran during the upgrade process. If `catuppst.sql` has not run, then proceed to run `catuppst.sql` as shown in this step. Warning messages are also displayed when running `catctl.pl` indicating that `catuppst.sql` was not run during the upgrade.

Run `catuppst.sql` as follows:

```
$ORACLE_HOME/perl/bin/perl catcon.pl -n 1 -e -b catuppst -d ''.'''
catuppst.sql
```

The log file `catuppst0.log` will be generated with the results of the post upgrade.

16. Run `catcon.pl` to invoke `utlrbp.sql` to recompile any remaining stored PL/SQL and Java code. Use the following syntax:

```
$ORACLE_HOME/perl/bin/perl catcon.pl -n 1 -e -b utlrbp -d ''.''' utlrbp.sql
```

The log file `utlrbp0.log` is generated with the results of the recompilations.

17. Run `utlu121s.sql` to verify that all issues have been fixed.

In a CDB:

```
$ORACLE_HOME/perl/bin/perl catcon.pl -n 1 -e -b utlu121s -d ''.'''
utlu121s.sql
```

In a non-CDB:

```
SQL> @rdbms/admin/utlu121s.sql
```

The log file `utlu121s0.log` will be generated with the upgrade results.

To see information about the state of the database, you can run `utlu121s.sql` as many times as you want, any time *after* completing the upgrade. If the `utlu121s.sql` script returns errors or shows components that are not `VALID` or not the most recent release, then see ["Troubleshooting the Upgrade for Oracle Database"](#) on page 42 for more information.

18. Verify that all expected packages and classes are valid. (This step assumes that you previously ran `preupgrd.sql` before starting the upgrade.)

```
$ORACLE_HOME/perl/bin/perl catcon.pl -n 1 -e -b utluiobj -d ''.'''
utluiobj.sql
```

The log file `utluiobj0.log` will be generated with the results of the verification.

Note: If the Pre-upgrade Information Tool detected `INVALID` objects and populated the `registry$sys_inv_objs` and `registry$nonsys_inv_objs` tables, then execute `ORACLE_HOME/rdbms/admin/utluiobj.sql` to display only those objects which are newly invalid because of the upgrade process. The `utluiobj.sql` script only displays objects that are now `INVALID` but which were `VALID` before the upgrade. See ["Using the Pre-Upgrade Information Tool \(preupgrd.sql\)"](#) on page 2-27..

19. At this time you may ensure that the time zone data files are current. See ["Upgrade the Time Zone File Version After Upgrading Oracle Database"](#) on page 4-4. You can also make this adjustment after the upgrade.
20. Exit `SQL*Plus` if you are still in it, for example, in a non-CDB.
21. If you are upgrading a cluster database from releases 10.2, 11.1, or 11.2, then upgrade the database configuration in Oracle Clusterware using the following command:

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

where `db-unique-name` is the database name assigned to it (not the instance name), and `oraclehome` is the Oracle home location in which the database is being upgraded. The `SRVCTL` utility supports long GNU-style options in addition to short CLI options used in earlier releases.

Your database is now upgraded to Oracle Database 12c. You are ready to complete the procedures described in [Chapter 4, "Post-Upgrade Tasks for Oracle Database"](#).

Note: Oracle Warehouse Builder (OWB) components are not upgraded as part of the Oracle Database upgrade process. Oracle Warehouse Builder is not installed as part of Oracle Database 12c. See ["Requirement for Upgrading Databases that Use Oracle Warehouse Builder \(OWB\)"](#) on page 2-23 for more information.

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

3.6 How the Data Dictionary is Upgraded and Status is Displayed

In Oracle Database 12c, improvements have been made to the upgrade process and how upgrade status appears for the upgraded database.

The following topics discuss changes that affect performance and status for data dictionary objects:

- [Parallel Upgrading of Data Dictionary](#)
- [Change to Upgrade Status Setting](#)
- [Change to Status After Running the Re-compilation utlrlp.sql Script](#)

3.6.1 Parallel Upgrading of Data Dictionary

Upgrading the data dictionary in Oracle Database is now done in parallel with the `catctl.pl` utility, which speeds up the upgrade process. Instead of one SQL process loading the data dictionary, there can now be multiple processes, depending on CPU capacity. The `catctl.pl` utility loads data into the dictionary as fast as possible and defers any work that can be done in *normal* mode instead of *upgrade* mode. Upgrading the database in this fashion reduces the overall downtime.

3.6.2 Change to Upgrade Status Setting

When the upgrade is complete and if there are no errors, then the status of the database is set to `UPGRADED`. This behavior differs from earlier releases where the status of the database was set to `VALID` after upgrading. The `UPGRADED` status indicates that the data dictionary has been loaded without any errors. If any errors are logged to `registry$error`, then the status of the upgrade is set to `INVALID`.

3.6.3 Change to Status After Running the Re-compilation utlrlp.sql Script

In earlier releases of Oracle Database after upgrading a database, `VALID` meant that all objects in the data dictionary have been recompiled and are ready for use. In Oracle Database 12c, obtaining a `VALID` status differs. Running the `utlrlp.sql` script in *normal* mode, recompiles data dictionary objects and moves the data dictionary from an `UPGRADED` status to a `VALID` status.

Oracle cannot guarantee that the database upgrade is *valid* unless `utlrlp.sql` has been run after upgrading and before starting the upgraded database for the first time. Not running `utlrlp.sql` after performing an upgrade forces data dictionary objects to be compiled during their first access. The first user accessing the database suffers the performance hit for each of the invalid objects accessed at this point. Once recompilation of those invalid objects is completed, then normal processing returns. Running `utlrlp.sql` ensures that the database is ready for use after upgrading. Any errors found are reported immediately and can be fixed before users of the upgraded database encounter problems.

3.7 About the catupgrd.sql Script in Earlier Releases of Oracle Database

Earlier releases of Oracle Database used the `catupgrd.sql` Upgrade Utility to process the upgrade. Oracle strongly recommends that starting with Oracle Database 12c, you use the `catctl.pl` Parallel Upgrade Utility, which replaces `catupgrd.sql` and provides both parallel processing mode and serial mode. The information about `catupgrd.sql` is provided here for backward compatibility with any custom scripts or processes that you use for upgrading. You must execute the `catupgrd.sql` script from the Oracle Database 12c environment.

Note: If you run `catupgrd.sql` by itself, then `catuppst.sql` does not run as part of the upgrade process. The `catuppst.sql` script must be run after `catupgrd.sql` completes.

See Also: ["Deprecation of catupgrd.sql Script and Introduction of New catctl.pl Utility"](#) on page 8-2, for information about running the earlier Upgrade Utility

The `catupgrd.sql` script creates and alters certain data dictionary tables. It also upgrades or installs the following database components in the new Oracle Database 12c 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
- Oracle Text
- Oracle Spatial and Graph
- Oracle Data Mining
- Oracle Label Security
- Messaging Gateway
- Oracle Database Vault
- Oracle Application Express

3.8 About the catuppst.sql Script

The `catuppst.sql` script generates informational messages and log files during the upgrade. DBUA and `catctl.pl` automatically run `catuppst.sql`. Run this script separately if any errors occurred during the manual upgrade.

The `catuppst.sql` script, which is located in the `ORACLE_HOME/rdbms/admin` directory, performs remaining upgrade actions that do not require the database to be in UPGRADE mode. This script also automatically applies the latest Oracle patch set update (PSU).

An example of the `catuppst.sql` output is as follows:

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

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

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

3.9 About Transporting and Upgrading a Database (Full Transportable Export/Import)

The Full Transportable Export/Import feature of Oracle Data Pump provides two options, which are described in detail in *Oracle Database Administrator's Guide*. The two options are:

- Using a file-based Oracle Data Pump export/import
- Using a non-file based network mode Oracle Data Pump import

See Also:

- Information about transporting a database using an export dump file in *Oracle Database Administrator's Guide*
- The full procedure in the *Oracle Database Administrator's Guide* for transporting a database over the network

3.10 Troubleshooting the Upgrade for Oracle Database

Oracle provides troubleshooting tips for errors or issues you may encounter while upgrading your database. To learn about important changes that affect this release, be sure to read [Chapter 8, "Deprecated and Desupported Features for Oracle Database 12c."](#)

This section contains the following topics:

- [About Starting Oracle Database in Upgrade Mode](#)
- [Running DBUA with Different ORACLE_HOME Owner](#)
- [Oracle Database Upgrade Script Termination Errors](#)

- [Resource Limits for Upgrading Oracle Database](#)
- [SQL*Plus Edition Session Startup Error for Oracle Database](#)
- [Manual Workaround for ORA-01408 on Oracle Database Index](#)
- [Error ORA-00018 for DBCA Processes with Release 11.1.0.7 Databases](#)
- [Error ORA-00020 Maximum Number of Processes Exceeded When Running utlrp.sql](#)
- [Running the DBMS_DST Package After Upgrade Can Result in ORA-01822](#)
- [Continuing Upgrades After Server Restarts \(ADVM/ACFS Driver Error\)](#)
- [Understanding Component Status of the Post-Upgrade Status Tool](#)
- [Adjusting Oracle ASM Password File Location After Upgrade](#)
- [Error ORA-27248: sys.dra_reevaluate_open_failures is running](#)
- [Oracle Database Enterprise User Security, OLS-OID, and Provisioning Profile Error](#)
- [32K Migration Error with utl32k.sql and MAX_STRING_SIZE](#)
- [ASM Loses Rolling Migration State If CRS Shutdown On All Nodes](#)
- [Data Type Versioning Could Cause Cross-Version Replication \(ORA-26656\)](#)
- [Oracle XML DB Errors ORA-64123 and ORA-14097](#)

See Also: ["Rerunning the Upgrade for Oracle Database"](#) on page 55

3.10.1 About Starting Oracle Database in Upgrade Mode

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

Errors described in this section might occur when attempting to start the new Oracle Database 12c database. Some of these errors write 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 11.0.0.

- 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-00704: bootstrap failure

See Also: [Ora-00704, Ora-39700: Database Must Be Opened With Upgrade Option \(Doc ID 1349722.1\)](#) on My Oracle Support at <http://support.oracle.com>

- 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 desupported initialization parameters, then make a note of the desupported initialization parameters and continue with the upgrade. Remove the desupported initialization parameters the next time you shut down the database.

3.10.2 Running DBUA with Different ORACLE_HOME Owner

When upgrading your database, DBUA expects that both the source (pre-12c) Oracle home and the destination (new 12c) Oracle home are owned by the same user. If this is not the case in your environment, then you need to make some changes to database file permissions and pass additional parameters to DBUA. Otherwise on the DBUA Prerequisite Checks page you will see the "upgrade.xml not found" error, and you will not be able to proceed with the upgrade.

- First, ensure all database files (data files, the redo file, control files, archive log destination, recovery area, spfile, and password file) are readable and writable by both the new 12c and pre-12c binary owners. This can be achieved by having both users in the same group, such as "oinstall" and by giving the group read/write access to all the database files.
- Run DBUA by specifying the `-logdir` command line switch, and provide a directory that both 12c and pre-12c binary owners can write to, for example, `/tmp`. DBUA will use this directory to store the output from the Pre-upgrade Information Tool (which you execute on the pre-12c instance running under the pre-12c binary owner) and any DBUA logs generated by the 12c binary owner during upgrade.
 - For example, run DBUA as follows at the command line:

```
dbua -logdir /tmp
```

3.10.3 Oracle Database Upgrade Script Termination Errors

If you did not run the Pre-Upgrade Information Tool before starting the upgrade, then the `catctl.pl` and `catupgrd.sql` scripts terminate with errors as follows:

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

If you receive any of these errors, issue the `SHUTDOWN ABORT` statement. Then revert to the original Oracle home directory, and run the Pre-Upgrade Information Tool (`ultu121i.sql`) as described in ["About the Pre-Upgrade Information Tool for Oracle Database"](#) on page 2-27.

3.10.4 Resource Limits for Upgrading Oracle Database

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.

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

- `SYSTEM` and `SYSAUX` tablespaces

Typically if your `SYSTEM` tablespace size is insufficient, you receive an error message during the upgrade as follows:

```
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 Administrator's Guide* for information about using manual shared memory management

- Rollback segments/undo tablespace

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

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

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

- Fast Recovery Area

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

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

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

3.10.5 SQL*Plus Edition Session Startup Error for Oracle Database

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

3.10.6 Manual Workaround for ORA-01408 on Oracle Database Index

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

```
SQL> create index system.repcat$_audit_column_f2_idx on
```

```

2  system.repcat$_audit_column(base_sname,base_ename,base_conflict_type_id,
3  base_reference_name)
4  /
system.repcat$_audit_column(base_sname,base_ename,base_conflict_type_id,
                                *
ERROR at line 2:
ORA-01408: such column list indexed

```

The workaround is to drop the REPCAT\$_AUDIT_COLUMN_IDX1 index and rerun the upgrade as described in ["Rerunning the Upgrade for Oracle Database"](#) on page 3-55.

3.10.7 Error ORA-00018 for DBCA Processes with Release 11.1.0.7 Databases

If you are using Oracle Grid Infrastructure and you want to create an Oracle RAC release 11.1.0.7 database, then you may need to increase the DBCA default for session processes. For Oracle Database 12c, DBCA sets the default value for processes to 300. In earlier releases, DBCA set the default value to 150.

If you see the error message "ORA-00018:maximum number of session exceeded," then change the default value for session processes in DBCA to 300. DBCA then successfully creates the release 11.1.0.7 database to use with Oracle Grid Infrastructure release 12.1.

3.10.8 Error ORA-00020 Maximum Number of Processes Exceeded When Running utlrp.sql

This error may indicate that your Oracle configuration does not have sufficient number of processes available for the recompile. Refer to documentation for more details about setting the PROCESSES parameter.

See Also: *Oracle Database Reference*

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

Running the DBMS_DST package after upgrading to Oracle Database 12c can result in the ORA-01882: time zone region not found error.

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

```

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

```

To fix this problem, update the time zone version as described in ["About Oracle Database Warnings for TIMESTAMP WITH TIME ZONE Data Type"](#) on page 2-31 and rerun the upgrade as described in ["Rerunning the Upgrade for Oracle Database"](#) on page 3-55.

3.10.10 Continuing Upgrades After Server Restarts (ADVM/ACFS Driver Error)

On Windows platforms, an error may occur related to ADVM or ACFS drivers if a server restarts during an upgrade. You may see the following error message.

ACFS-9427: Failed to unload ADVM/ACFS drivers. A system reboot is recommended, or ACFS-9428 Failed to load ADVM/ACFS drivers. A system reboot is recommended.

Cause: If you see these errors during an upgrade, then the cause is that the ADVM and ACFS drivers are still in use, and you must restart the system to start the new drivers.

Action: Complete the steps as described in the following procedures.

For nodes other than the first node (the node on which the upgrade is started):

1. Restart the node where the error occurs.
2. Run the root script on that node again.

For first nodes (the node on which the upgrade is started):

1. Complete the upgrade of all other nodes in the cluster.
2. Restart the first node.
3. Run the root script on the first node again.
4. To complete the upgrade, log in as root, and run the script `configToolAllCommands`, located in the path `Grid_home/cfgtoollogs/configToolAllCommands`.

See Also: *Oracle Grid Infrastructure Installation Guide* for your operating system for more information about troubleshooting upgrade issues for clusters

3.10.11 Understanding Component Status of the Post-Upgrade Status Tool

The `utlu121s.sql` Post-Upgrade Status Tool reports the status for database components after the upgrade. This tool can be run anytime after upgrade, post-upgrade, or after recompiling invalid objects with `utlrp.sql`. An example of the summary report is provided in "[Sample Upgrade Summary Report for the Post-Upgrade Status Tool \(utlu121s.sql\)](#)" on page 48.

The following list briefly describes the status values that you might see:

- **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`. See "[Scripts for Upgrading Oracle Database](#)" on page 1-2.
- **VALID**
The component is valid with no errors.
- **LOADING**
The component is loading
- **LOADED**
The component has successfully finished loading.
- **UPGRADING**
The component is in process being upgraded.
- **UPGRADED**

The component has completed upgrading with no errors.

- DOWNGRADING

The component is in process being downgraded.

- DOWNGRADED

The component has completed downgrading with no errors.

- REMOVING

The component is in process being removed.

- REMOVED

The component was not upgraded because it was removed from the database.

- OPTION OFF

The server option required for the component was not installed or was not linked with the server. Check the `V$OPTION` view and the install logs. Install the component or relink the server with the required option, and then rerun `catctl.pl`. See ["Scripts for Upgrading Oracle Database"](#) on page 1-2.

- NO SCRIPT

The component upgrade script was not found in `ORACLE_HOME`. Check the install logs, install the component software, and then rerun `catctl.pl`. See ["Scripts for Upgrading Oracle Database"](#) on page 1-2.

See Also: ["Change to Upgrade Status Setting"](#) on page 40.

3.10.11.1 Sample Upgrade Summary Report for the Post-Upgrade Status Tool (utlu121s.sql)

The Post-Upgrade Status Tool creates the upgrade summary report. The upgrade summary report contains the status of all components within the database as shown in [Example 3-2, "Upgrade Summary Report for the Post-Upgrade Status Tool"](#). The example shows the report for a multitenant container database (CDB) and the pluggable databases (PDBs).

After the upgrade completes, a copy of the report is placed in one of the following locations:

1. `$ORACLE_BASE/cfgtoollogs/<db-unique-name>/upgrade/upg_summary.rpt`
2. `$ORACLE_HOME/cfgtoollogs/<db-unique-name>/upgrade/upg_summary.rpt`
3. `$ORACLE_HOME/rdbms/logs/upg_summary.rpt`

The locations are shown in the order that `utlu121s.sql` uses to create the report. If it cannot create the report under the first directory scheme, then it tries the second and then the third location before it does not try further to write the report.

Example 3-2 Upgrade Summary Report for the Post-Upgrade Status Tool

```
Oracle Database 12.1 Post-Upgrade Status Tool          06-12-2014 20:37:07
.
.                                CDB$ROOT:1
Component                               Current      Version     Elapsed Time
Name                                     Status      Number      HH:MM:SS
.
Oracle Server
.
.                                UPGRADED    12.1.0.2.0  00:14:26
JServer JAVA Virtual Machine
```

```

.
Oracle Real Application Clusters          VALID      12.1.0.2.0  00:05:50
.
Oracle Workspace Manager                 OPTION OFF  12.1.0.2.0  00:00:01
.
Oracle Label Security                    VALID      12.1.0.2.0  00:01:19
.
Oracle XDK                               VALID      12.1.0.2.0  00:00:11
.
Oracle Text                             VALID      12.1.0.2.0  00:00:59
.
Oracle XML Database                     VALID      12.1.0.2.0  00:00:32
.
Oracle Database Java Packages           VALID      12.1.0.2.0  00:01:40
.
Oracle Multimedia                       VALID      12.1.0.2.0  00:00:21
.
Oracle Database Vault                   VALID      12.1.0.2.0  00:03:43
.
Post Upgrade                            VALID      12.1.0.2.0  00:00:13
.
Final Actions                            00:01:25
.
Total Upgrade Time: 00:31:47

```

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.11

```

Oracle Database 12.1 Post-Upgrade Status Tool      06-12-2014 21:00:02
.
PDB$SEED:2
Component          Current      Version  Elapsed Time
Name              Status      Number   HH:MM:SS
.
Oracle Server      UPGRADED   12.1.0.2.0  00:13:19
.
JServer JAVA Virtual Machine          VALID      12.1.0.2.0  00:02:24
.
Oracle Real Application Clusters      OPTION OFF  12.1.0.2.0  00:00:01
.
Oracle Workspace Manager              VALID      12.1.0.2.0  00:00:51
.
Oracle Label Security                 VALID      12.1.0.2.0  00:00:06
.
Oracle XDK                            VALID      12.1.0.2.0  00:00:40
.
Oracle Text                           VALID      12.1.0.2.0  00:00:11
.
Oracle XML Database                   VALID      12.1.0.2.0  00:00:48
.
Oracle Database Java Packages         VALID      12.1.0.2.0  00:00:11
.
Oracle Multimedia                     VALID      12.1.0.2.0  00:01:41
.
Oracle Database Vault                 VALID      12.1.0.2.0  00:00:12
.
Post Upgrade                            00:01:19
.
Final Actions                            00:00:01
.

```

Total Upgrade Time: 00:22:24

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.08

```

.
Oracle Database 12.1 Post-Upgrade Status Tool           06-12-2014 21:24:16
.
                                CDB1_PDB1:3
Component      Current      Version  Elapsed Time
Name           Status      Number   HH:MM:SS
.
Oracle Server          UPGRADED   12.1.0.2.0  00:13:05
.
JServer JAVA Virtual Machine      VALID      12.1.0.2.0  00:03:01
.
Oracle Real Application Clusters   OPTION OFF 12.1.0.2.0  00:00:02
.
Oracle Workspace Manager          VALID      12.1.0.2.0  00:01:01
.
Oracle Label Security             VALID      12.1.0.2.0  00:00:06
.
Oracle XDK                       VALID      12.1.0.2.0  00:00:45
.
Oracle Text                      VALID      12.1.0.2.0  00:00:15
.
Oracle XML Database              VALID      12.1.0.2.0  00:00:55
.
Oracle Database Java Packages     VALID      12.1.0.2.0  00:00:17
.
Oracle Multimedia                VALID      12.1.0.2.0  00:01:59
.
Oracle Database Vault            VALID      12.1.0.2.0  00:00:10
.
Post Upgrade                      00:01:18
.
Final Actions                      00:00:01
.
Total Upgrade Time: 00:23:33

```

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.12

```

.
Oracle Database 12.1 Post-Upgrade Status Tool           06-12-2014 21:27:03
.
                                CDB1_PDB2:4
Component      Current      Version  Elapsed Time
Name           Status      Number   HH:MM:SS
.
Oracle Server          UPGRADED   12.1.0.2.0  00:14:01
.
JServer JAVA Virtual Machine      VALID      12.1.0.2.0  00:03:11
.
Oracle Real Application Clusters   OPTION OFF 12.1.0.2.0  00:00:02
.
Oracle Workspace Manager          VALID      12.1.0.2.0  00:01:05
.
Oracle Label Security             VALID      12.1.0.2.0  00:00:07
.
Oracle XDK                       VALID      12.1.0.2.0  00:00:49
.

```

```

Oracle Text
.
VALID 12.1.0.2.0 00:00:12
Oracle XML Database
.
VALID 12.1.0.2.0 00:00:54
Oracle Database Java Packages
.
VALID 12.1.0.2.0 00:00:15
Oracle Multimedia
.
VALID 12.1.0.2.0 00:02:03
Oracle Database Vault
.
VALID 12.1.0.2.0 00:00:13
Post Upgrade
.
00:01:18
Final Actions
.
00:00:01
Total Upgrade Time: 00:24:51
    
```

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.11

```

.
Oracle Database 12.1 Post-Upgrade Status Tool          06-12-2014 21:49:18
.
CDB1_PDB3:5
Component          Current          Version  Elapsed Time
Name              Status          Number    HH:MM:SS
.
Oracle Server
.
UPGRADED 12.1.0.2.0 00:13:57
JServer JAVA Virtual Machine
.
VALID 12.1.0.2.0 00:03:01
Oracle Real Application Clusters
.
OPTION OFF 12.1.0.2.0 00:00:02
Oracle Workspace Manager
.
VALID 12.1.0.2.0 00:01:07
Oracle Label Security
.
VALID 12.1.0.2.0 00:00:07
Oracle XDK
.
VALID 12.1.0.2.0 00:00:46
Oracle Text
.
VALID 12.1.0.2.0 00:00:12
Oracle XML Database
.
VALID 12.1.0.2.0 00:00:58
Oracle Database Java Packages
.
VALID 12.1.0.2.0 00:00:13
Oracle Multimedia
.
VALID 12.1.0.2.0 00:02:10
Oracle Database Vault
.
VALID 12.1.0.2.0 00:00:13
Post Upgrade
.
00:01:17
Final Actions
.
00:00:01
Total Upgrade Time: 00:24:42
    
```

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.10

```

.
Oracle Database 12.1 Post-Upgrade Status Tool          06-12-2014 21:51:15
.
CDB1_PDB4:6
Component          Current          Version  Elapsed Time
    
```

| Name | Status | Number | HH:MM:SS |
|----------------------------------|------------|------------|----------|
| . | | | |
| Oracle Server | UPGRADED | 12.1.0.2.0 | 00:13:41 |
| . | | | |
| JServer JAVA Virtual Machine | VALID | 12.1.0.2.0 | 00:02:48 |
| . | | | |
| Oracle Real Application Clusters | OPTION OFF | 12.1.0.2.0 | 00:00:04 |
| . | | | |
| Oracle Workspace Manager | VALID | 12.1.0.2.0 | 00:00:58 |
| . | | | |
| Oracle Label Security | VALID | 12.1.0.2.0 | 00:00:07 |
| . | | | |
| Oracle XDK | VALID | 12.1.0.2.0 | 00:00:40 |
| . | | | |
| Oracle Text | VALID | 12.1.0.2.0 | 00:00:15 |
| . | | | |
| Oracle XML Database | VALID | 12.1.0.2.0 | 00:00:50 |
| . | | | |
| Oracle Database Java Packages | VALID | 12.1.0.2.0 | 00:00:16 |
| . | | | |
| Oracle Multimedia | VALID | 12.1.0.2.0 | 00:02:04 |
| . | | | |
| Oracle Database Vault | VALID | 12.1.0.2.0 | 00:00:15 |
| . | | | |
| Post Upgrade | | | 00:01:16 |
| . | | | |
| Final Actions | | | 00:00:01 |
| . | | | |
| Total Upgrade Time: | | | 00:23:54 |

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.08

| Component Name | Current Status | Version Number | Elapsed Time HH:MM:SS |
|---|----------------|----------------|-----------------------|
| Oracle Database 12.1 Post-Upgrade Status Tool | | | |
| CDB1_PDB5:7 | | | |
| Oracle Server | UPGRADED | 12.1.0.2.0 | 00:09:42 |
| . | | | |
| JServer JAVA Virtual Machine | VALID | 12.1.0.2.0 | 00:02:31 |
| . | | | |
| Oracle Real Application Clusters | OPTION OFF | 12.1.0.2.0 | 00:00:01 |
| . | | | |
| Oracle Workspace Manager | VALID | 12.1.0.2.0 | 00:00:52 |
| . | | | |
| Oracle Label Security | VALID | 12.1.0.2.0 | 00:00:05 |
| . | | | |
| Oracle XDK | VALID | 12.1.0.2.0 | 00:00:42 |
| . | | | |
| Oracle Text | VALID | 12.1.0.2.0 | 00:00:13 |
| . | | | |
| Oracle XML Database | VALID | 12.1.0.2.0 | 00:00:44 |
| . | | | |
| Oracle Database Java Packages | VALID | 12.1.0.2.0 | 00:00:12 |
| . | | | |
| Oracle Multimedia | VALID | 12.1.0.2.0 | 00:01:44 |
| . | | | |
| Oracle Database Vault | | | |


```

VALID          12.1.0.2.0  00:00:12
Post Upgrade
.
Final Actions
.
Total Upgrade Time: 00:18:44
00:01:08
00:00:01

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.08
.

```

3.10.12 Standard Edition Starter Database and Components with Status OPTION OFF

When the Standard Edition (SE) starter database is upgraded, components that are not included with the starter database cannot be upgraded by the SE server because they require options that are not installed in the Standard Edition.

After the upgrade, components may have a STATUS value of OPTION OFF in the DBA_REGISTRY view, and there are some invalid objects in the associated component schemas. The Database Upgrade Assistant (DBUA) shows unsuccessful upgrades for these components.

3.10.13 Adjusting Oracle ASM Password File Location After Upgrade

The Oracle ASM password file location is not shown in the command output when you invoke `srvctl config asm` after a Grid Infrastructure upgrade. The location of the password file is not automatically passed to the new Oracle ASM disk group. To enable SRVCTL to have the password file location after upgrade, you must advance the diskgroup compatibility setting and create a PWFIL in the diskgroup. Then SRVCTL will report the configured location of the shared PWFIL.

Follow the procedure in *Oracle Automatic Storage Management Administrator's Guide* for managing a shared password file in a disk group.

3.10.14 Error ORA-27248: sys.dra_reevaluate_open_failures is running

If DBUA failed with error "ORA-27248: sys.dra_reevaluate_open_failures is running" when upgrading the database, then the job DRA_REEVALUATE_OPEN_FAILURES is running and causes upgrade failure. You need to ensure that the job is stopped.

In a job definition, if ALLOW_RUNS_IN_RESTRICTED_MODE is set to TRUE, the job is permitted to run when the database is in restricted or upgrade mode, provided that the job owner is permitted to log in during this mode. The default setting is FALSE.

You can use the following query to see the state of any jobs that may be running:

```
SQL> select OBJECT_NAME, Owner, OBJECT_TYPE from dba_objects where
object_name like '%DRA_REEVA%';
```

3.10.15 Oracle Database Enterprise User Security, OLS-OLD, and Provisioning Profile Error

After upgrading databases that use OLS and a standby database, you might see ORA-16000 (database open for read-only access). After switchover, the Provisioning profile should be updated with the connect information of the new primary. Otherwise, the policies will continue to get propagated to the new standby (old primary) and will fail with ORA-16000. See *Oracle Database Enterprise User Security*

Administrator's Guide for information about the additional steps required for upgrading Oracle Database from release 10g (10.1) and higher to Oracle Database 12c.

3.10.16 32K Migration Error with utl32k.sql and MAX_STRING_SIZE

An error is generated if the initialization parameter `MAX_STRING_SIZE` is set to `EXTENDED` but the 32K migration had not completed running the `utl32k.sql` script. The database upgrade will terminate with the following error:

```
SELECT TO_NUMBER('32K_MIGRATION_NOT_COMPLETED')
      *
      ERROR at line 1:
      ORA-01722: invalid number
```

The database upgrade does not automatically run the `utl32k.sql` script and does not perform the 32K migration.

To complete the upgrade and the 32K migration after upgrading:

1. Reset the initialization parameter `MAX_STRING_SIZE` to `STANDARD`.
2. Restart the database in `UPGRADE` mode.
3. Rerun the upgrade, either with `DBUA` or the manual procedure.
4. After the database has been upgraded, set the initialization parameter `MAX_STRING_SIZE` to `EXTENDED`.
5. Restart the database in `UPGRADE` mode.
6. Run `./rdbms/admin/utl32k.sql`.

The upgraded database will complete the 32K migration and will support the `EXTENDED` parameter.

3.10.17 ASM Loses Rolling Migration State If CRS Shutdown On All Nodes

Oracle Automatic Storage Management (Oracle ASM) loses the rolling migration state if Cluster Ready Services (CRS) shuts down on all nodes. Consider the following scenario of four nodes (node1, node2, node3, and node4) that are at Oracle Database release 11.2.0.2 and being upgraded to release 12.1.0.2.

- Node1 and node2 are upgraded to 12.1.0.2 and running.
- Node3 and node 4 are still at 11.2.0.2 and running.
- Now consider that there is an outage where all CRS stacks are down, which leaves the cluster in a heterogeneous state (that is, two nodes at 11.2.0.2 and two nodes at 12.1.0.2).

To proceed with the upgrade, only nodes at release 11.2.0.2 (i.e., node3 or node4 or both) should be started and the following command needs to be executed: `ALTER SYSTEM START ROLLING MIGRATION TO '12.1.0.2'` command on the Oracle ASM instance on node3 or node4 before starting any 12.1.0.2 node.

Note that before executing above step to bring the Oracle ASM cluster back into rolling migration, you cannot start two nodes of different versions in the cluster. If you do so, one of the Oracle ASM versions will fail with either the `ORA-15153` or `ORA-15163` error message.

Continue the upgrade procedure as already documented from this point forward.

3.10.18 Data Type Versioning Could Cause Cross-Version Replication (ORA-26656)

As part of release 12.1.0.2, versioning of data types that could be attributes of Oracle object types have been introduced (reference Bug 18897657). Because of this feature, cross-version replication between release 12.1.0.1 and release 12.1.0.2 databases might be affected resulting in the ORA-26656 error.

If any user-defined object types contain attributes of DATE, TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH LOCAL TIME ZONE, BINARY_FLOAT, BINARY_DOUBLE, NCHAR, NVARCHAR2, NCLOB, ANYDATA, etc., you must apply the mandatory Patch Set Update 18038108 to all release 12.1.0.1 instances.

3.10.19 Oracle XML DB Errors ORA-64123 and ORA-14097

The issue described in this section is for upgrading release 11.2.0.2 Oracle Database only.

The following scenario will cause the ALTER TABLE EXCHANGE PARTITION statement to fail with errors:

On an 11.2.0.2 database, you create an XMLType table or a table with an XMLType column that is partitioned and then create a structured XMLIndex on the table. Next, you upgrade the 11.2.0.2 database to Oracle Database 12c or above. If you execute the ALTER TABLE EXCHANGE PARTITION statement on one of the base table partitions with another partition or table that is created in the Oracle Database 12c database with the same structure and similar XMLIndex, then it will fail with the following error:

```
ORA-64123: XMLIndex DDL: failure of a recursive DDL
ORA-14097: column type or size mismatch in ALTER TABLE EXCHANGE PARTITION
```

This is not an issue if your index was created on a non-11.2.0.2 database. 11.2.0.2 is the only database release with this issue. Other database releases, like 11.2.0.1 and 11.2.0.3, do not have this issue.

The **workaround** is as follows: after upgrading to Oracle Database 12c or later, drop and re-create the structured XMLIndex on the table where the index was upgraded from release 11.2.0.2.

3.11 Rerunning the Upgrade for Oracle Database

When upgrading Oracle Database, the upgrade process can be rerun or restarted with DBUA or with the `catctl.pl` script.

You can use the following methods to rerun the upgrade:

- [Rerunning the Upgrade with the Upgrade \(catctl.pl\) Script](#)
- [Options for Rerunning the Upgrade for a Multitenant Database \(CDB\)](#)

3.11.1 Rerunning the Upgrade with the Upgrade (catctl.pl) Script

You can rerun the upgrade with the `catctl.pl` script. Also see "[Options for Rerunning the Upgrade for a Multitenant Database \(CDB\)](#)" on page 57.

1. Shut down the database. For a non-CDB and a CDB, the syntax is the same.

```
SQL> SHUTDOWN IMMEDIATE
```

2. Restart the database in UPGRADE mode.

For a non-CDB:

```
SQL> STARTUP UPGRADE
```

For a CDB:

```
SQL> STARTUP UPGRADE alter pluggable database all open upgrade;
```

3. Rerun catctl.pl.

Note: You can rerun the `catctl.pl` script as many times as necessary. For a CDB you can rerun `catctl.pl` for a specific PDB. For example, for a PDB named `PDB1`:

```
$ORACLE_HOME/perl/bin/perl catctl.pl -n 4 -l $ORACLE_
HOME/diagnostics catupgrd.sql -c 'PDB1'
```

4. Run `utlu121s.sql`, the Post-Upgrade Status Tool, which provides a summary of the status of the upgrade in the spool log. You can run `utlu121s.sql` any time *before* or *after* completing the upgrade.

In a non-CDB:

```
SQL> @rdbms/admin/utlu121s.sql
```

In a CDB:

```
$ORACLE_HOME/perl/bin/perl catcon.pl -n 1 -e -b utlu121s -d ''.'''
utlu121s.sql
```

If the `utlu121s.sql` script returns errors or shows components that are not `VALID` or not the most recent release, then see "[Troubleshooting the Upgrade for Oracle Database](#)" on page 42 for more information.

5. **Important:** The `catuppst.sql` script is run as part of the upgrade process unless the upgrade returns errors during the process. Check the log file for "Started: `catuppst.sql`" to verify that `catuppst.sql` ran during the upgrade process. If `catuppst.sql` has not run, then proceed to run `catuppst.sql` as shown in this step. Warning messages are also displayed when running `catctl.pl` indicating that `catuppst.sql` was not run during the upgrade.

Run `catuppst.sql`, which is located in the `ORACLE_HOME/rdbms/admin` directory, as follows:

```
$ORACLE_HOME/perl/bin/perl catcon.pl -n 1 -e -b catuppst -d ''.'''
catuppst.sql
```

The log file `catuppst0.log` will be generated with the results of the post upgrade.

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

```
$ORACLE_HOME/perl/bin/perl catcon.pl -n 1 -e -b utlrlp -d ''.''' utlrlp.sql
```

The log file `utlrlp0.log` will be generated with the results of the re-compilations.

Verify that all expected packages and classes are valid.

In a single PDB (PDB1 in this example), open the PDB in normal mode as follows:

```
alter pluggable database PDB1 open;
```

Run `catcon.pl` to invoke `utlrlp.sql` in the PDB to recompile any remaining stored PL/SQL and Java code. Use the following syntax:

```
$ORACLE_HOME/perl/bin/perl catcon.pl -n 1 -e -b utlrlp -d ''.''' -c 'PDB1'
```

```
utlrp.sql
```

In a non-CDB:

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

In an entire CDB:

```
SQL> ALTER SESSION SET CONTAINER = "CDB$ROOT"
SQL> SELECT count(*) FROM dba_invalid_objects;
SQL> SELECT distinct object_name FROM dba_invalid_objects;
SQL> ALTER SESSION SET CONTAINER = "PDB$SEED"
SQL> SELECT count(*) FROM dba_invalid_objects;
SQL> SELECT distinct object_name FROM dba_invalid_objects;
SQL> ALTER SESSION SET CONTAINER = "PDB1"
SQL> SELECT count(*) FROM dba_invalid_objects;
SQL> SELECT distinct object_name FROM dba_invalid_objects;
```

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

7. Run `utlu121s.sql` again to verify that all issues have been fixed.

In a non-CDB:

```
SQL> @rdbms/admin/utlu121s.sql
```

In a CDB:

```
$ORACLE_HOME/perl/bin/perl catcon.pl -n 1 -e -b utlu121s -d ''.'''
utlu121s.sql
```

8. Exit SQL*Plus.
9. If you are upgrading a cluster database from releases 10.2, 11.1, or 11.2, then upgrade the database configuration in Oracle Clusterware using the following command:

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

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

Your database is now upgraded to Oracle Database 12c. You are ready to complete the procedures described in [Chapter 4, "Post-Upgrade Tasks for Oracle Database"](#).

3.11.2 Options for Rerunning the Upgrade for a Multitenant Database (CDB)

Depending on the configuration of your multitenant database (CDB), there are several scenarios for rerunning the upgrade. You may opt to do the following:

- [Rerun the Entire Upgrade for the CDB](#)

- [Rerun the Upgrade Only on Specified CDBs](#)
- [Rerun the Upgrade While Other PDBs Are Online Using an Exclusion List](#)
- [Rerun the Upgrade Using an Inclusion List to Specify a CDB or PDBs](#)
- [Restarting the Upgrade from a Specific Phase that Failed Using -p](#)

3.11.2.1 Rerun the Entire Upgrade for the CDB

You may need to rerun the entire upgrade in the case where several different issues occur during the first attempt. This example demonstrates reupgrading the CDB\$ROOT, PDB\$SEED and containers when a problem such as running out of shared pool occurs.

1. Start the upgrade again.

```
SQL> startup upgrade;
      alter pluggable database all open upgrade;
```

2. Run catctl.pl as follows:

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catctl.pl -d $ORACLE_
HOME/rdbms/admin -l $ORACLE_HOME/rdbms/log catctl.pl catupgrd.sql
```

The upgrade will be run again on all the containers included with CDB\$ROOT and PDB\$SEED.

3.11.2.2 Rerun the Upgrade Only on Specified CDBs

You may rerun the upgrade only on one or more specified multitenant containers (CDBs). You can do this with an exclusion list. An exclusion list contains containers that you do not want to upgrade. An exclusion list uses the -C parameter option in catctl.pl. (You must use uppercase C for this parameter; it is case sensitive.) This method can come in handy when there are many PDBs to reupgrade.

In this example, the multitenant container database contains five PDBs. All the upgrades were successful except that CDB1_PDB1 and CDB1_PDB2 failed with an upgrade error. The goal is to run the upgrade in CDB1_PDB1 and CDB1_PDB2 containers only. You can achieve this by shutting down the entire multitenant database and restarting only the PDBs you want to upgrade.

1. Shutdown and then startup in upgrade mode.

```
SQL> shutdown immediate;
      startup upgrade;
      alter pluggable database CDB1_PDB1 open upgrade;
      alter pluggable database CDB1_PDB2 open upgrade;
```

2. Show the status of the CDB and PDBs.

```
SQL> show pdbs
```

| CON_ID | CON_NAME | OPEN MODE | RESTRICTED |
|--------|-----------|-----------|------------|
| 2 | PDB\$SEED | MIGRATE | YES |
| 3 | CDB1_PDB1 | MIGRATE | YES |
| 4 | CDB1_PDB2 | MIGRATE | YES |
| 5 | CDB1_PDB3 | MOUNTED | |
| 6 | CDB1_PDB4 | MOUNTED | |
| 7 | CDB1_PDB5 | MOUNTED | |

3. Rerun the upgrade excluding CDB\$ROOT and PDB\$SEED from the upgrade.

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catctl.pl -d $ORACLE_
```

```
HOME/rdbms/admin -l $ORACLE_HOME/rdbms/log catctl.pl -C 'CDB$ROOT PDB$SEED'
catupgrd.sql
```

The upgrade reruns and completes on CDB1_PDB1 and CDB1_PDB2.

Note: For Windows, the `-C` option must be specified with quotes around the CDB root name and PDB seed name, for example:

```
... -C "CDB$ROOT PDB$SEED"
```

3.11.2.3 Rerun the Upgrade While Other PDBs Are Online Using an Exclusion List

In this example, the CDB\$ROOT will be started up in normal mode and the containers: CDB1_PDB3, CDB1_PDB4, and CDB1_PDB5 are online. In this case the upgrade failed in containers CDB1_PDB1 and CDB1_PDB2, but succeeded in containers: CDB1_PDB3, CDB1_PDB4, and CDB1_PDB5. The DBA reviews the upgrade logs for CDB1_PDB3, CDB1_PDB4, and CDB1_PDB5 and brings these specific containers online.

To rerun upgrade on containers: CDB1_PDB1 and CDB1_PDB2 do the following:

1. Bring up CDB\$ROOT in normal mode and open CDB1_PDB1 and CDB1_PDB2 in upgrade mode. CDB1_PDB3, CDB1_PDB4, CDB1_PDB5 are in normal mode. The example is as follows:

```
SQL> startup;
      alter pluggable database CDB1_PDB1 open upgrade;
      alter pluggable database CDB1_PDB2 open upgrade;
      alter pluggable database cdb1_pdb3 open;
      alter pluggable database cdb1_pdb4 open;
      alter pluggable database cdb1_pdb5 open;
```

2. Show the PDBs status.

```
SQL> show pdbs
```

| CON_ID | CON_NAME | OPEN MODE | RESTRICTED |
|--------|-----------|------------|------------|
| 2 | PDB\$SEED | READ ONLY | NO |
| 3 | CDB1_PDB1 | MIGRATE | YES |
| 4 | CDB1_PDB2 | MIGRATE | YES |
| 5 | CDB1_PDB3 | READ WRITE | NO |
| 6 | CDB1_PDB4 | READ WRITE | NO |
| 7 | CDB1_PDB5 | READ WRITE | NO |

3. Now rerun the upgrade, excluding CDB\$ROOT from the upgrade.

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catctl.pl -d $ORACLE_
HOME/rdbms/admin -l $ORACLE_HOME/rdbms/log catctl.pl -C 'CDB$ROOT'
catupgrd.sql
```

For Windows, the `-C` option must be specified with quotes around the CDB root name and PDB seed name, for example:

```
... -C "CDB$ROOT PDB$SEED"
```

The upgrade reruns and completes on CDB1_PDB1 and CDB1_PDB2.

3.11.2.4 Rerun the Upgrade Using an Inclusion List to Specify a CDB or PDBs

Instead of using an exclusion list to rerun the upgrade, you can alternatively use an inclusion list to specify the CDB and/or PDBs you want to rerun the upgrading on. An inclusion list uses the `-c` parameter in `catctl.pl` as shown in this example:

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catctl.pl -d $ORACLE_
HOME/rdbms/admin -l $ORACLE_HOME/rdbms/log catctl.pl -c 'CDB1_PDB1 CDB1_PDB2'
catupgrd.sql
```

The upgrade reruns and completes on CDB1_PDB1 and CDB1_PDB2.

3.11.2.5 Restarting the Upgrade from a Specific Phase that Failed Using `-p`

You can use `catctl.pl` to rerun an upgrade and skip upgrade phases that already have run successfully. To determine the phase number to restart, examine the upgrade log to identify where the first error occurred, and in what phase.

3.11.2.5.1 Reviewing Non-CDB Log Files for Failed Phases Log files can span from `catupgrd0...catupgrd7.log`.

3.11.2.5.2 Reviewing CDB Log Files for Failed Phases Log files for CDB\$ROOT (CDBs) can span from `catupgrd0...catupgrd7.log`. Log files for Containers (PDBs) span from `catupgrdpdbname0...catupgradepdbname7.log`.

3.11.2.5.3 Procedure for Finding and Restarting Multitenant Upgrades from a Failed Phase To restart a multitenant upgrade from a failed phase, first identify which PDB created the error and then search its appropriate log file.

To identify the PDB that caused a multitenant upgrade failure, look at the upgrade summary report, or review `catupgrd0.log`; this log contains the upgrade summary report at the end of the file.

Use this procedure to check each log file looking for errors:

1. Locate log files with errors. For example:

Linux and UNIX:

```
$ grep -i 'error at line' catupgrd*.log
```

Windows:

```
C:\> find /I "error at line" catupgrd*.log
```

The `grep` or `find` command displays the filenames of log files in which an error is found.

2. Check each log file that has an error and identify where the first error occurred. Use the text editor of your choice to review each log file. Search for the first occurrence of the phrase 'error at line'. When you find the phrase, then search backwards from the error (reverse search) for `PHASE_TIME__START`. For example:

```
PHASE_TIME__START 15 15-01-16 08:49:41
```

The number after `PHASE_TIME__START` is the phase number where the error has occurred. In this example, the phase number is 15.

Each log file can have an error in it. Repeat this phrase check, and identify the phase number with errors for each log file that contains an error, and identify the log file that contains the lowest phase number.

The log file that contains the lowest phase number is the restart phase number. This is the phase number from which you restart the upgrade.

For example:

```
catupgrd0.log error occurred in phase 15:
```

```
PHASE_TIME__START 15 15-01-16 08:49:41
```

```
catupgrd1.log error occurred in phase 19:
```

```
PHASE_TIME__START 19 15-01-16 08:50:01
```

In this example, the restart phase number is 15. Ensure that you identify the first error seen in all the log files, so that you can restart the upgrade from that phase.

3. Restart the upgrade from the failed phase by changing directory to the running the `catctl.pl` command using the `-p` flag, and providing the restart phase number. In multitenant databases, also use the `-c` flag using the syntax `-c 'PDBname'`, where `PDBname` is the name of the PDB where the failure occurred.

For example:

Non-CDB Oracle Database:

```
cd $ORACLE_HOME/rdbms/admin
$ORACLE_HOME/perl/bin/perl catctl.pl -p 15 catupgrd.sql
```

PDB in a multitenant Oracle Database (CDB):

```
cd $ORACLE_HOME/rdbms/admin
$ORACLE_HOME/perl/bin/perl catctl.pl -p 15 -c 'PDB1' catupgrd.sql
```

In both examples, the upgrade is restarted from phase 15, identified with the `-p` flag. In the multitenant example, the PDB with the error is identified with the `-c` flag.

In these examples, the upgrade starts from phase 15 and runs to the end of the upgrade.

4. (Optional) You can also run the phase that contained an error by specifying a stop phase, using the `-P` flag. Using a stop phase allows the upgrade to just rerun that phase in which the error occurred. You can determine quickly if the error is fixed by running it on the phase with the error, without running the entire upgrade.

In the following `catctl.pl` command example, the upgrade is restarted at phase 15, and it is stopped at the end of phase 15 using the `-P` flag to specify the stop phase:

```
cd $ORACLE_HOME/rdbms/admin
$ORACLE_HOME/perl/bin/perl catctl.pl -p 15 -P 15 -c 'PDB1' catupgrd.sql
```

After you confirm that the error is fixed in the phase with the error, you can then resume the upgrade after that phase.

For example, if you have confirmed that the error in phase 15 of your multitenant database upgrade of PDB1 is fixed, then you can use the following command to continue the upgrade at phase 16:

```
cd $ORACLE_HOME/rdbms/admin
$ORACLE_HOME/perl/bin/perl catctl.pl -p 16 -c 'PDB1' catupgrd.sql
```

Post-Upgrade Tasks for Oracle Database

After performing the procedures for upgrading Oracle Database, you must complete required tasks and consider recommendations for the new release.

This chapter contains the following topics:

- [How to Show the Current State of the Oracle Data Dictionary](#)
- [About OPatch Commands After Upgrading Oracle Database](#)
- [Required Tasks to Complete After Upgrading Oracle Database](#)
- [Required Task After Oracle Grid Infrastructure Upgrades](#)
- [Requirement for Role-Allocated Software Owners and Database Upgrade After Oracle ASM Upgrade](#)
- [Recommended and Best Practices to Complete After Upgrading Oracle Database](#)
- [Recommended Tasks After Upgrading an Oracle RAC Database](#)
- [Recommended Tasks After Upgrading Oracle ASM](#)
- [Recommended Tasks After Upgrading Oracle Database Express Edition](#)
- [Optionally Update Oracle Application Express Packaged Applications](#)
- [Tasks to Complete Only After Manually Upgrading Oracle Database](#)

4.1 How to Show the Current State of the Oracle Data Dictionary

You can collect upgrade and migration diagnostic information about the current state of the data dictionary by running the `dbupgdiag.sql` script. The script can be run in SQL*Plus both before the upgrade on the source database and after the upgrade on the upgraded database as SYS user.

See Also: Note 556610.1 Script to Collect DB Upgrade/Migrate Diagnostic Information (`dbupgdiag.sql`) on My Oracle Support at <http://support.oracle.com>

To show the current state of the dictionary, execute a SQL query similar to the following example:

```
SQL> spool /tmp/regInvalid.out
SQL> set echo on
-- query registry
SQL> set lines 80 pages 100
SQL> select substr(comp_id,1,15) comp_id,substr(comp_name,1,30)
       comp_name,substr(version,1,10) version,status
```

```
from dba_registry order by modified;
```

To query invalid objects, execute a SQL query similar to:

```
SQL> select substr(owner,1,12) owner,substr(object_name,1,30)
object,substr(object_type,1,30) type, status
from dba_objects where status <> 'VALID' order by owner, type;
SQL> spool off
SQL> set echo off
```

4.2 About OPatch Commands After Upgrading Oracle Database

After you upgrade Oracle Database, you must run OPatch commands from the new Oracle home. For example, run the `lsinventory` command from the new Oracle home in order to list an accurate and complete inventory of what is currently installed on the system.

See Also: "Appendix A" in *Oracle OPatch User's Guide for Windows and UNIX* for OPatch syntax and commands

4.3 Required Tasks to Complete After Upgrading Oracle Database

After you upgrade Oracle Database, regardless of whether you perform the upgrade manually, or upgrade by using Database Upgrade Assistant (DBUA), you must complete any required tasks that are specified for your environment.

- [Set Environment Variables on Linux and UNIX Systems After Manual Upgrades](#)
- [Upgrade the Recovery Catalog After Upgrading Oracle Database](#)
- [Upgrade the Time Zone File Version After Upgrading Oracle Database](#)
- [Upgrade Statistics Tables Created by the DBMS_STATS Package After Upgrading Oracle Database](#)
- [Upgrade Externally Authenticated SSL Users After Upgrading Oracle Database](#)
- [Configure the FTP and HTTP Ports and HTTP Authentication for Oracle XML DB](#)
- [Install Oracle Text Supplied Knowledge Bases After Upgrading Oracle Database](#)
- [Update Your Oracle Application Express Configuration After Upgrading Oracle Database](#)
- [Configure Access Control Lists \(ACLs\) to External Network Services](#)
- [Enable Oracle Database Vault After Upgrading Oracle Database](#)
- [Check for the SQLNET.ALLOWED_LOGON_VERSION Parameter Behavior](#)

4.3.1 Set Environment Variables on Linux and UNIX Systems After Manual Upgrades

If your operating system is Linux or UNIX, and if you performed a manual upgrade of Oracle Database, then you must ensure that certain environment variables point to the directories of the new Oracle Database release. Additionally, if you are upgrading a cluster database, then perform these checks on all nodes on which the cluster database has instances configured.

Confirm that the following environment variables point to the directories of the new Oracle home:

- ORACLE_HOME

- PATH

Note: DBUA automatically makes necessary changes to Oracle environment variables.

See Also:

- *Oracle Database Administrator's Guide* for information about setting environment variables for the database
- *Oracle Database Installation Guide* for your operating system for information about setting other important environment variables

4.3.2 Set oratab and Scripts to Point to the New Oracle Location After Upgrading Oracle Database

After you upgrade Oracle Database to the new release, you must ensure that your `oratab` file and any client scripts that set the value of `ORACLE_HOME` point to the new Oracle home that is created for the new Oracle Database 12c release. Although DBUA automatically points `oratab` to the new Oracle home, client scripts must be checked no matter which method you use to upgrade.

See Also: *Oracle Database Administrator's Guide* for information about setting operating system environment variables

4.3.3 Enable the New Extended Data Type Capability

Oracle Database 12c introduces `MAX_STRING_SIZE` to control the maximum size of `VARCHAR2`, `NVARCHAR2`, and `RAW` data types in SQL. Setting `MAX_STRING_SIZE = EXTENDED` enables the 32767 byte limit introduced in Oracle Database 12c. The `COMPATIBLE` initialization parameter must be set to `12.0.0.0` or higher in order to be able to set `MAX_STRING_SIZE = EXTENDED`.

Enabling a system to take advantage of the new extended data types requires specific upgrade actions, as documented in the *Oracle Database Reference*. You can change the value of `MAX_STRING_SIZE` from `STANDARD` to `EXTENDED`. However, you cannot change the value of `MAX_STRING_SIZE` from `EXTENDED` to `STANDARD`. By setting `MAX_STRING_SIZE = EXTENDED`, you are taking an explicit action that could introduce application incompatibility in your database.

See Also: *Oracle Database Reference* for complete information about `MAX_STRING_SIZE`, including recommendations and procedures

4.3.4 Adjust Minimum and Maximum for Parallel Execution Servers

In Oracle Database 12c the default for `PARALLEL_MIN_SERVERS` has been changed from 0 to a value depending on your hardware platform to accommodate sufficient minimal support for parallel execution. If the new default setting is too high for your environment, then adjust the setting for your requirements.

The default for `PARALLEL_MAX_SERVERS` has not changed, and, therefore, if you have not changed the default in your old environment, no actions are needed.

See Also: *Oracle Database Reference* for information about `PARALLEL_MIN_SERVERS`

4.3.5 Upgrade the Recovery Catalog After Upgrading Oracle Database

If you use a version of the recovery catalog schema that is older than that required by the RMAN client, then you must upgrade it. For complete information about upgrading the recovery catalog and the `UPGRADE CATALOG` command, see *Oracle Database Backup and Recovery User's Guide*.

See Also: *Oracle Database Backup and Recovery User's Guide* for information on managing an RMAN recovery catalog

4.3.6 Upgrade the Time Zone File Version After Upgrading Oracle Database

If the Pre-Upgrade Information Tool instructed you to upgrade the time zone files after completing the database upgrade, then use the `DBMS_DST` PL/SQL package to upgrade the time zone file.

Oracle Database supplies multiple versions of time zone files, and there are two types of file associated with each one: a large file, which contains all the time zones defined in the database, and a small file, which contains only the most commonly used time zones. The large versions are designated as `timezlrg_version_number.dat`, while the small versions are designated as `timezone_version_number.dat`. The files are located in the `oracore/zoneinfo` subdirectory under the Oracle Database home directory.

See Also:

- *Oracle Database Globalization Support Guide* and follow the procedure in "Steps to Upgrade Time Zone File and Timestamp with Time Zone Data"
- Note ID 1509653.1 "Updating the RDBMS DST version in 12c Release 1 (12.1.0.1 and up) using `DBMS_DST`" on My Oracle Support at <http://support.oracle.com>
- "[About Oracle Database Warnings for TIMESTAMP WITH TIME ZONE Data Type](#)" on page 2-31

4.3.7 Upgrade Statistics Tables Created by the `DBMS_STATS` Package After Upgrading Oracle Database

If you created statistics tables using the `DBMS_STATS.CREATE_STAT_TABLE` procedure, then upgrade these tables by running the following procedure:

```
EXECUTE DBMS_STATS.UPGRADE_STAT_TABLE('green', 'stat_table');
```

In the example, `green` is the owner of the statistics table and `STAT_TABLE` is the name of the statistics table. Perform this procedure for each statistics table.

See Also: *Oracle Database PL/SQL Packages and Types Reference* for information about the `DBMS_STATS` package

4.3.8 Upgrade Externally Authenticated SSL Users After Upgrading Oracle Database

If you are upgrading from Oracle9i 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 SSL external users conversion (`extusrupgrade`) script to upgrade those users. The script has the following syntax:

```
ORACLE_HOME/rdbms/bin/extusrupgrade --dbconnectstring  
host_name:port_no:sid --dbuser <db admin> --dbuserpassword  
password -a
```

Note: If you are upgrading from Oracle Database 10g Release 2 (10.2) or later, then you are not required to run the `extusrupgrade` script.

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

4.3.9 Configure the FTP and HTTP Ports and HTTP Authentication for Oracle XML DB

For Oracle Database 12c, Database Creation Assistant (DBCA) does not configure ports for Oracle XML DB. You should also configure the authentication for HTTP for accessing Oracle XML DB Repository to take advantage of improved security features.

Starting with Oracle Database 12c, Oracle has enhanced database security by providing support for digest authentication. Digest authentication is an industry standard protocol commonly used with the HTTP protocol, and is supported by most HTTP clients. Digest authentication ensures that passwords are always transmitted in a secure manner, even when an encrypted (HTTPS) connection is not in use. Support for digest authentication enables organizations to deploy applications that leverage the Oracle XML DB HTTP without having to worry about passwords being compromised. Digest authentication support in Oracle XML DB also ensures that the Oracle XML DB HTTP server remains compatible with Microsoft Web Folders WebDAV clients.

See Also: *Oracle XML DB Developer's Guide* for information on configuring and managing authentication mechanisms for HTTP

After installing or upgrading for the new release, you must manually configure the FTP and HTTP ports for Oracle XML DB as follows:

1. Use `DBMS_XDB_CONFIG.setHTTPPort(HTTP port number)` to set the HTTP port for Oracle XML DB.

```
SQL> exec DBMS_XDB_CONFIG.setHTTPPort(port_number);
```

2. Use `DBMS_XDB_CONFIG.setFTPPort(FTP port number)` to set the FTP port for Oracle XML DB.

```
SQL> exec DBMS_XDB_CONFIG.setFTPPort(port_number);
```

Note: You can query the port numbers to use for FTP and HTTP in the procedure by using `DBMS_XDB_CONFIG.getFTPPort` and `DBMS_XDB_CONFIG.getHTTPPort` respectively.

3. To see all the used port numbers, you can use `DBMS_XDB_CONFIG.usedport`.

See Also: *Oracle XML DB Developer's Guide* for complete information about accessing the Oracle XML DB Repository data using FTP and HTTP(S)/WebDAV protocols

4.3.10 Install Oracle Text Supplied Knowledge Bases After Upgrading Oracle Database

The Oracle Text-supplied knowledge bases are part of the companion products for Oracle Database 12c and are not immediately available after an upgrade to Oracle Database 12c. Any Oracle Text features dependent on the supplied knowledge bases which were available before the upgrade do not function after the upgrade. To

re-enable such features, you must install the Oracle Text supplied knowledge bases from the installation media.

After an upgrade, all user extensions to the Oracle Text supplied knowledge bases must be regenerated. These changes affect all databases installed in the given Oracle home.

See Also:

- *Oracle Text Application Developer's Guide* for information about Oracle Text-supplied knowledge bases
- The postinstallation tasks section of your platform-specific *Oracle Database Installation Guide* for companion products

4.3.11 Update Your Oracle Application Express Configuration After Upgrading Oracle Database

If your database originally included Oracle Application Express release 3.2 or later, then there is no additional configuration necessary after upgrading to Oracle Database 12c. However, if Oracle Application Express is in the registry and Oracle Application Express will be upgraded, then you should set the `open_cursors` parameter to a minimum of 200.

If your database was not an Oracle Express Edition database, but contained an earlier release of Oracle Application Express, then the latest release is automatically installed during the upgrade. You must complete a series of postinstallation steps to configure Application Express for use with the new Oracle Database 12c.

See Also: *Oracle Application Express Installation Guide* for postinstallation tasks for Oracle Application Express

If your database is an Oracle Express Edition database, then it contains an earlier release of Oracle Application Express, which is tailored for the Oracle Express Edition environment. Review the Oracle document describing the differences between Oracle Express Edition and Oracle Application Express at the following URL:

<http://www.oracle.com/technetwork/developer-tools/apex/overview/index.html>

4.3.12 Configure Access Control Lists (ACLs) to External Network Services

Oracle Database 12c includes fine-grained access control to the `UTL_TCP`, `UTL_SMTP`, `UTL_MAIL`, `UTL_HTTP`, or `UTL_INADDR` packages. If you have applications that use these packages, then after upgrading Oracle Database you must configure network *access control lists* (ACLs) in the database before these packages can work as they did in earlier releases. Without the ACLs, your applications may fail with the error "ORA-24247: network access denied by access control list (ACL)."

See Also: *Oracle Database Security Guide* for more complicated situations, such as connecting some users to host A and other users to host B

4.3.13 Enable Oracle Database Vault After Upgrading Oracle Database

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

To start Oracle DV enforcement in the upgraded database, enable DV using the procedure `dvsys.dbms_macadm.enable_dv()`. A user with the `DV_OWNER` or `DV_ADMIN`

role is the only one who can execute this procedure. For the procedure to take effect, the database instance needs to be restarted.

See Also:

- ["Requirement for Upgrading Oracle Databases That Use Oracle Database Vault"](#) on page 2-21
- The appendix about "Disabling and Enabling Database Vault" in *Oracle Database Vault Administrator's Guide*
- The appendix about "Post-installation Database Vault Procedures" in *Oracle Database Vault Administrator's Guide*

4.3.14 Check for the SQLNET.ALLOWED_LOGON_VERSION Parameter Behavior

Starting with Oracle Database 12c, the default value for the `SQLNET.ALLOWED_LOGON_VERSION` parameter has changed from 8 to 11. The use of this parameter has been deprecated, and it is now replaced with the `SQLNET.ALLOWED_LOGON_VERSION_SERVER` and `SQLNET.ALLOWED_LOGON_VERSION_CLIENT` parameters. If you have not explicitly set the `SQLNET.ALLOWED_LOGON_VERSION_SERVER` parameter in the upgraded database, then connections from clients earlier than release 10g will fail with the error ORA-28040: No matching authentication protocol. For better security, check the password verifiers of your database users, and then configure the database to use the correct password verifier by setting the `SQLNET.ALLOWED_LOGON_VERSION_SERVER` and `SQLNET.ALLOWED_LOGON_VERSION_CLIENT` parameters.

If you have password-protected roles (secure roles) in your existing database and if you upgrade to Oracle Database 12c with the default `SQLNET.ALLOWED_LOGON_VERSION_SERVER` setting of 11, because those secure roles only have release 10g verifiers, the password for each secure role must be reset by the administrator so that the secure roles will remain usable after the upgrade.

See Also:

- *Oracle Database Security Guide* for information about ensuring against password security threats
- *Oracle Database Security Guide* for information about setting the password versions of users
- ["Deprecation of SQLNET.ALLOWED_LOGON_VERSION Parameter"](#) on page 8-15

4.4 Required Task After Oracle Grid Infrastructure Upgrades

After upgrading, you should confirm that your environment variable settings are correct. See ["Using Environment Variables for Grid Infrastructure Installations"](#) on page 4-8. Oracle ASM is included as part of an Oracle Grid Infrastructure installation. If you upgrade Oracle Clusterware and Oracle ASM for a cluster, then Oracle Clusterware and Oracle ASM are both located in the same home, which is referred to as *Grid home*. You can have one installation owner that owns all Oracle software installations, or you can use role-allocated owners, in which case you use a separate software owner for the Grid Infrastructure installation, and separate software owners for one or more Oracle Database installations.

See Also: *Oracle Grid Infrastructure Installation Guide* for your platform for more information about role-allocated installation owners

4.4.1 Using Environment Variables for Grid Infrastructure Installations

If your operating system is Linux or UNIX, then confirm that your environment variable settings are correct after performing an upgrade.

If you use a single Oracle installation owner for all installations, then be aware that you should change environment variables such as `ORACLE_HOME` either to an Oracle Database home, or to the Grid home, depending on whether you are administering an Oracle Database instance as part of database administration, or administering an Oracle ASM instance as part of storage administration.

If you use role-allocated Oracle installation owners, so that you have a separate owner for the Oracle Grid Infrastructure (Oracle Clusterware and Oracle ASM) software, then set the following environment variables for the Grid Infrastructure installation owner so that they point to the directories of the Oracle ASM home in the Grid home:

- `ORACLE_HOME`
- `PATH`

Also, check that your `oratab` file and any client scripts for Oracle ASM that set the value of `ORACLE_HOME` point to the Oracle ASM home in the Grid home.

Note: If you are upgrading a clustered Oracle ASM installation to an Oracle Grid Infrastructure for a cluster installation, then perform these checks on all cluster member nodes. DBUA automatically points `oratab` to the new Oracle home. Client scripts must be checked no matter how you upgrade.

See Also:

- Your operating system-specific *Oracle Database Installation Guide* for information about setting other important environment variables on your operating system
- *Oracle Grid Infrastructure Installation Guide* or *Oracle Database Installation Guide* for your platform
- *Oracle Automatic Storage Management Administrator's Guide* for information about upgrading an Oracle ASM instance

4.5 Requirement for Role-Allocated Software Owners and Database Upgrade After Oracle ASM Upgrade

If you separate the operating system user ownership of the Oracle Grid Infrastructure binaries and the Oracle Database installation owners of one or more databases, then you must migrate the operating system user of an upgraded Oracle ASM or database home. For example, if you are migrating from one software binary owner (such as `oracle`) to multiple role-allocated software owner user accounts (such as `grid`, `oracle1`, `oracle2`), then change the owner of the existing Oracle ASM installation owner to the installation owner that you plan to use for the Oracle Grid Infrastructure installation.

There are three scenarios to consider as follows:

- [Keeping the Existing User as the Oracle ASM Operating System User](#)
- [Changing the Operating System User for Single-Instance Oracle ASM](#)
- [Changing the Operating System User for an Oracle RAC Database](#)

See Also: *Oracle Automatic Storage Management Administrator's Guide* for information on making an Oracle ASM disk group compatible with new releases, and for additional information about Oracle ASM upgrades

4.5.1 Keeping the Existing User as the Oracle ASM Operating System User

If you are using the same operating system user for your Oracle Grid Infrastructure installation that you used for your existing Oracle ASM installation, then run Oracle Universal Installer (OUI) to perform a Grid Infrastructure installation, and select the upgrade option. OUI automatically upgrades your existing Oracle ASM installation from the earlier release to Oracle Database 12c in the Grid home.

4.5.2 Changing the Operating System User for Single-Instance Oracle ASM

Consider your earlier release Oracle ASM installation is installed in Oracle home 4 (OH4) and currently running `oracle` as the operating system user, and you want to change the Oracle ASM operating system user to `grid`. This is useful if you have two databases using Oracle ASM, and you had installed Oracle ASM with an installation owner that is identical to the owner of the existing databases, and you want to change the operating system installation owner of Oracle ASM to enable separate databases to run as separate operating system users, where neither Oracle Database installation owner has Oracle Grid Infrastructure binary ownership.

4.5.3 Changing the Operating System User for an Oracle RAC Database

There may be scenarios where you must change the operating system user for an Oracle RAC database. For example, if your earlier release database is installed in Oracle home 4 (OH4) and currently running `oracle` as the operating system user, then you should consider changing the Oracle ASM operating system user to `grid`. Changing the operating system user of Oracle ASM enables separate databases to run as separate operating system users, where no Oracle Database installation owner has Grid Infrastructure binary ownership.

See Also: *Oracle Grid Infrastructure Installation Guide* for the procedures to change the operating system user for an Oracle RAC database with Grid Infrastructure and Oracle ASM

4.6 Recommended and Best Practices to Complete After Upgrading Oracle Database

After you have upgraded Oracle Database, there are tasks Oracle recommends that you complete. These tasks represent good practices for updating Oracle Database, and are recommended regardless of whether you performed the upgrade manually or by using DBUA.

- [Back Up the Database](#)
- [Run the `postupgrade_fixups.sql` Script](#)
- [Gather Fixed Objects Statistics with `DBMS_STATS`](#)
- [Reset Passwords to Enforce Case-Sensitivity](#)
- [Understand Changes with Oracle Grid Infrastructure](#)
- [Understand Oracle ASM and Oracle Grid Infrastructure Installation and Upgrade](#)

- [Add New Features as Appropriate](#)
- [Develop New Administrative Procedures as Needed](#)
- [Set Threshold Values for Tablespace Alerts](#)
- [Migrate From Rollback Segments to Automatic Undo Mode](#)
- [Configure Oracle Data Guard Broker](#)
- [Migrate Tables from the LONG Data Type to the LOB Data Type](#)
- [Migrate Your Upgraded Oracle Databases to Use Unified Auditing](#)
- [Test the Upgraded Production Oracle Database](#)

4.6.1 Back Up the Database

Make sure you perform a full backup of the production database. Although this step is not required, Oracle strongly recommends that you back up your production database.

See Also: *Oracle Database Backup and Recovery User's Guide* for details about backing up a database with RMAN

4.6.2 Run the `postupgrade_fixups.sql` Script

Although DBUA runs the `postupgrade_fixups.sql` script as part of completing the upgrade process, you can run it any time after upgrading. The `postupgrade_fixups.sql` script generates three categories of information for your upgraded database: general warnings, errors, and informational recommendations.

Run this script any time after completing an upgrade with DBUA or manually. If `Oracle_Base` is defined, then the generated scripts and log files are created in `Oracle_Base/cfgtoollogs/` of the original database from which you ran the upgrade. If `Oracle_Base` is not defined, then the generated scripts and log files are created in `ORACLE_HOME/cfgtoollogs/` of the database from which you ran the upgrade.

Set the system to spool results to a log file so you can read the output. Do not, however, spool to the `admin` directory. Run the script from the location of the database from which you ran the upgrade (not the new upgraded location):

```
SQL> SPOOL postupgrade.log
```

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

```
SQL> SPOOL OFF
```

Note: If you move either a PDB or any other stand-alone database from server A to server B, you must copy the `postupgrade_fixups.sql` script to the new location to execute it post-upgrade in the new environment.

4.6.3 Gather Fixed Objects Statistics with `DBMS_STATS`

A few days after upgrading Oracle Database, a best practice is to gather fixed objects statistics with the `DBMS_STATS.GATHER_FIXED_OBJECTS_STATS` PL/SQL procedure. This can have a positive impact on overall database performance. `DBMS_STATS.GATHER_FIXED_OBJECTS_STATS` also displays a recommendation to remove all hidden or underscore parameters and events from `init.ora/spfile`.

Because of the transient nature of the x\$ tables, it is important that you gather fixed objects statistics when there is a representative workload on the system. This may not always be feasible on large systems due to additional resources needed to gather the statistics. If you cannot do this during peak load, then you should do it after the system has warmed up and the key types of fixed object tables have been populated.

To gather statistics for fixed objects, run the following PL/SQL procedure:

```
SQL> execute dbms_stats.gather_fixed_objects_stats;
```

See Also: *Oracle Database PL/SQL Packages and Types Reference* for more information about using the GATHER_FIXED_OBJECTS_STATS procedure

4.6.4 Reset Passwords to Enforce Case-Sensitivity

You can enforce case sensitivity for passwords. For example, the password hpp5620qr fails if it is entered as hpp5620QR or hPp5620Qr.

If you must reset the passwords of existing users during the database upgrade procedure, then for existing databases each user password must be reset with an ALTER USER statement. For new databases created after the upgrade, there are no additional tasks or management requirements.

To take advantage of enforced case-sensitive passwords for releases earlier than 11.1.0.7, you must reset the passwords of existing users during the database upgrade procedure. In this case, for upgraded databases, you can run the DBMS_VERIFIER.EXPIRE_ACCOUNTS_WITHOUT_LATEST_VERIFIER procedure, which forces users whose accounts do not yet have the latest verifier to change their passwords the next time they log in. The server can then generate the latest verifier for their account. For new databases, there are no additional tasks or management requirements.

For SYSDBA and SYSOPER users, you can generate a new ORAPWD file using the new command line switch ignorecase.

Note:

- If the default security settings for Oracle Database 12c are in place, then passwords must be at least eight characters, and passwords such as welcome and oracle are not allowed. See *Oracle Database Security Guide* for more information on password strength.
 - The IGNORECASE parameter is deprecated in this release. Oracle recommends not using this parameter.
-
-

See Also: *Oracle Database Security Guide* for more information on enabling password case sensitivity

4.6.5 Understand Changes with Oracle Grid Infrastructure

Oracle Clusterware and Oracle ASM are both part of an Oracle Grid Infrastructure installation.

If Oracle Grid Infrastructure is installed for a single server, then it is deployed as an Oracle Restart installation with Oracle ASM. If Oracle Grid Infrastructure is installed for a cluster, then it is deployed as an Oracle Clusterware installation with Oracle ASM.

Oracle Restart enhances the availability of Oracle Database in a single-instance environment. If you install Oracle Restart, and there is a temporary failure of any part of the Oracle Database software stack, including the database, listener, and Oracle ASM instance, Oracle Restart automatically restarts the failed component. In addition, Oracle Restart starts all these components when the database host computer is restarted. The components are started in the proper order, taking into consideration the dependencies among components.

Oracle Clusterware is portable cluster software that enables clustering of single servers so that they cooperate as a single system. Oracle Clusterware also provides the required infrastructure for Oracle RAC. In addition, Oracle Clusterware enables the protection of any Oracle application or any other application within a cluster. In any case Oracle Clusterware is the intelligence in those systems that ensures required cooperation between the cluster nodes.

See Also: *Oracle Grid Infrastructure Installation Guide* for more information and procedures

4.6.6 Understand Oracle ASM and Oracle Grid Infrastructure Installation and Upgrade

In earlier releases, Oracle ASM was installed as part of the Oracle Database installation. Starting with Oracle Database release 11.2, Oracle ASM is installed when you install the Grid Infrastructure components. Oracle ASM shares an Oracle home with Oracle Clusterware when it is installed in a cluster such as with Oracle RAC or with Oracle Restart on a standalone server.

See Also: *Oracle Grid Infrastructure Installation Guide* for more information and procedures

4.6.7 Add New Features as Appropriate

The *Oracle Database New Features Guide* describes many of the new features available in the new Oracle Database 12c release. Determine which of these new features can benefit the database and applications. You can then develop a plan for using these features.

It is not necessary to make any immediate changes to begin using your new Oracle Database software. You might prefer to introduce these enhancements into your database and corresponding applications gradually.

[Chapter 5, "Upgrading Applications After Upgrading Oracle Database"](#) describes ways to enhance your applications so that you can take advantage of the features of the new Oracle Database 12c release. However, before you implement new features, test your applications and successfully run them with the upgraded database.

4.6.8 Develop New Administrative Procedures as Needed

After familiarizing yourself with the features of the new Oracle Database 12c release, review your database administration scripts and procedures to determine whether any changes are necessary.

Coordinate your changes to the database with the changes that are necessary for each application. For example, by enabling integrity constraints in the database, you might be able to remove some data checking from your applications.

4.6.9 Set Threshold Values for Tablespace Alerts

An upgraded Oracle Database 12c 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 12c database are:

- 85% full warning
- 97% full critical

4.6.10 Migrate From Rollback Segments to Automatic Undo Mode

If your database is earlier than Oracle Database 11g, then you must migrate the database that is being upgraded from using rollback segments (manual undo management) to automatic undo management.

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 10.2 or 11.1 releases to Oracle Database 12c.

- If `UNDO_MANAGEMENT=MANUAL`, then undo space is allocated externally as rollback segments.

To migrate to automatic undo management, perform the following steps:

1. Set `UNDO_MANAGEMENT=MANUAL`.
2. Start the instance again and run through a standard business cycle to obtain a representative workload. Doing this to assess the workload and compute the size of the undo tablespace required for automatic undo management.
3. After the standard business cycle completes, run the following function to collect the undo tablespace size and help with the sizing of the undo tablespace (DBA privileges are required to run this function):

```
DECLARE
    utbsiz_in_MB NUMBER;
BEGIN
    utbsiz_in_MB := DBMS_UNDO_ADV.RBU_MIGRATION;
end;
/
```

This function runs a PL/SQL procedure that provides information on how to size your new undo tablespace based on the configuration and usage of the rollback segments in your system. The function returns the sizing information directly.

4. Create an undo tablespace of the required size and turn on the automatic undo management by setting `UNDO_MANAGEMENT=AUTO` or by removing the parameter.
5. For Oracle RAC configurations, repeat these steps on all instances.

4.6.11 Configure Oracle Data Guard Broker

The value of `DGConnectIdentifier` is used for all Data Guard network traffic, all of the time. If you are upgrading an Oracle Database release 10g configuration, which requires you to first upgrade to Oracle Database 11g, the value that exists for `InitialConnectIdentifier` is retained as the new value for `DGConnectIdentifier` for the database. When upgrading an Oracle RAC database, the database administrator must ensure that the value for the `InitialConnectIdentifier` property reaches all instances.

4.6.12 Migrate Tables from the LONG Data Type to the LOB Data Type

The LOB data types (BFILE, BLOB, CLOB, and NCLOB) can provide many advantages over LONG data types. You can use the `ALTER TABLE` statement to change the data type of a LONG column to CLOB and that of a LONG RAW column to BLOB.

In the following example, the LONG column named `long_col` in table `long_tab` is changed to data type CLOB:

```
SQL> ALTER TABLE Long_tab MODIFY ( long_col CLOB );
```

After using this method to change LONG columns to LOBs, all the existing constraints and triggers on the table are still usable. However, all the indexes, including Domain indexes and Functional indexes, on all columns of the table become unusable and must be rebuilt using an `ALTER INDEX . . . REBUILD` statement. Also, the Domain indexes on the LONG column must be dropped before changing the LONG column to a LOB.

See Also: *Oracle Database SecureFiles and Large Objects Developer's Guide* for information about modifying applications to use LOB data

4.6.13 Migrate Your Upgraded Oracle Databases to Use Unified Auditing

In unified auditing, all Oracle Database audit trails (`SYS.AUD$` for the database audit trail, `SYS.FGA_LOG$` for fine-grained auditing, `DVYS.AUDIT_TRAIL$` for Database Vault, and so on) are combined into one single audit trail, which you can view by querying the `UNIFIED_AUDIT_TRAIL` data dictionary view for single-instance installations and `GV$UNIFIED_AUDIT_TRAIL` for Oracle Real Application Clusters environments. If you want to use the full, pure unified auditing facility, then you must manually migrate to it as described in "[Migrating to Unified Auditing for Oracle Database](#)" on page 15.

See Also: *Oracle Database Security Guide* for information about how the audit features have changed for this release

This section contains the following topics:

- [About the Unified Auditing Migration Process for Oracle Database](#)
- [Migrating to Unified Auditing for Oracle Database](#)
- [Managing Earlier Audit Records After You Migrate to Unified Auditing](#)
- [Removing the Unified Auditing Functionality](#)
- [Documentation References if You Choose Not to Use Unified Auditing](#)

4.6.13.1 About the Unified Auditing Migration Process for Oracle Database

By default, unified auditing is not enabled for upgraded databases. If you have upgraded from an earlier release to Oracle Database 12c, then your database uses the same auditing functionality that was used in the earlier release. For newly created

databases, the mixed-mode method of unified auditing is enabled by default. After you complete the migration to unified auditing, traditional auditing is disabled and the new audit records write to the unified audit trail.

To enable and configure the audit policies and how they are used, choose one method as follows:

- Use the pure unified audit facility.

Follow the procedure described in "[Migrating to Unified Auditing for Oracle Database](#)" on page 15 to use the pure unified auditing facility. Once the procedure for migrating to unified auditing is complete, you can create and enable new audit policies and also use the predefined audit policies. The audit records for these policies write to the unified audit trail. The earlier audit trails and their audit records remain, but no new audit records write to the earlier audit trails.

Note: The audit configuration from the earlier release has no effect in the unified audit system. Only unified audit policies generate audit records inside the unified audit trail.

- Use a mixed-mode audit facility.

The mixed-mode audit facility enables both traditional and unified auditing facilities to run simultaneously and applies to both new and upgraded databases. The mixed-mode unified auditing facility becomes available if you enable at least one of the unified auditing predefined audit policies. Audit records for these policies write to the unified audit trail. The audit configuration in the earlier release of Oracle Database is also available, and the audit records for this configuration write to the earlier audit trails. If you decide that you prefer using the pure unified audit facility, then you can switch to it by following the procedure in "[Migrating to Unified Auditing for Oracle Database](#)" on page 15.

Note: If the database is not writable, then audit records write to new format operating system files in the \$ORACLE_BASE/audit/\$ORACLE_SID directory.

See Also:

- *Oracle Database Security Guide* for information about the predefined audit policies
- *Oracle Database Security Guide* for information about the ora_SecureConfig audit policy

4.6.13.2 Migrating to Unified Auditing for Oracle Database

In a multitenant container database (CDB) environment, perform the following procedure in the `root`. The procedure will migrate both the `root` and any associated PDBs to unified auditing.

To migrate your database to enable unified auditing:

1. Log in to SQL*Plus as user SYS with the SYSDBA privilege.

```
sqlplus sys as sysdba
Enter password: password
```

In a Pluggable Databases environment, this login connects you to the `root`.

2. Run the following query to check if your Oracle database has already been migrated to unified auditing:

```
SQL> SELECT VALUE FROM V$OPTION WHERE PARAMETER = 'Unified Auditing';
```

If the output is for the VALUE column is TRUE, then unified auditing is already enabled in your database. See ["Managing Earlier Audit Records After You Migrate to Unified Auditing"](#) on page 4-17 for what you should do next. If the output is FALSE, then complete the remaining steps in this procedure.

3. Stop the database. For single-instance environments, enter the following commands from SQL*Plus:

```
SQL> SHUTDOWN IMMEDIATE
SQL> EXIT
```

For Windows systems, stop the Oracle service:

```
net stop OracleService%ORACLE_SID%
```

For Oracle RAC installations, shut down each database instance as follows:

```
srvctl stop database -db db_name
```

4. Stop the listener. (Stopping the listener is not necessary for Oracle RAC and Grid Infrastructure listeners.)

```
lsnrctl stop listener_name
```

You can find the name of the listener by running the `lsnrctl status` command. The name is indicated by the `Alias` setting.

5. Go to the `$ORACLE_HOME/rdbms/lib` directory.
6. Enable the unified auditing executable as follows:

- For UNIX, run the following command:

```
make -f ins_rdbms.mk uniaud_on ioracle ORACLE_HOME=$ORACLE_HOME
```

- For Windows, rename the `%ORACLE_HOME%/bin/orauniaud12.dll.dbl` file to `%ORACLE_HOME%/bin/orauniaud12.dll`.

7. Restart the listener.

```
lsnrctl start listener_name
```

8. Restart the database. Log in to SQL*Plus and then enter the `STARTUP` command as follows:

```
sqlplus sys as sysoper
Enter password: password
```

```
SQL> STARTUP
```

For Windows systems, start the Oracle service again.

```
net start OracleService%ORACLE_SID%
```

For Oracle RAC installations, start each database instance as follows:

```
srvctl start database -db db_name
```

4.6.13.3 Managing Earlier Audit Records After You Migrate to Unified Auditing

After you complete the procedure to migrate Oracle Database to use unified auditing, any audit records that your database had before remain in their earlier audit trails. You can archive these audit records and then purge their audit trails. With unified auditing in place, any new audit records write to the unified audit trail.

See Also:

- "Archiving the Audit Trail" in *Oracle Database Security Guide*
- "Purging Audit Trail Records" in *Oracle Database Security Guide*

4.6.13.4 Removing the Unified Auditing Functionality

If after you have enabled your databases to use unified auditing and you decide that you do not want unified auditing, you can remove the unified auditing functionality. In this case, your database uses the mixed-mode audit facility as described in ["Migrating to Unified Auditing for Oracle Database"](#) on page 4-15.

To remove unified auditing:

1. Stop the database.

```
sqlplus sys as sysoper
Enter password: password
```

```
SQL> SHUTDOWN IMMEDIATE
SQL> EXIT
```

For Windows systems, stop the Oracle service:

```
net stop OracleService%ORACLE_SID%
```

For Oracle RAC installations, shut down each database instance as follows:

```
srvctl stop database -db db_name
```

2. Go to the \$ORACLE_HOME/rdbms/lib directory.

3. Disable the unified auditing executable.

- **UNIX:** Run the following command:

```
make -f ins_rdbms.mk uniaud_off ioracle ORACLE_HOME=$ORACLE_HOME
```

- **Windows:** Rename the %ORACLE_HOME%/bin/orauniaud12.dll file to %ORACLE_HOME%/bin/orauniaud12.dll.dbl.

4. Restart the database.

```
sqlplus sys as sysoper
Enter password: password
```

```
SQL> STARTUP
SQL> EXIT
```

For Windows systems, start the Oracle service again.

```
net start OracleService%ORACLE_SID%
```

For Oracle RAC installations, start each database instance as follows:

```
srvctl start database -db db_name
```

4.6.13.5 Documentation References if You Choose Not to Use Unified Auditing

After upgrading to Oracle Database 12c, if you choose not to change to unified auditing, then you can find information about traditional non-unified auditing from Oracle documentation and from Oracle Technology Network.

Refer to information about non-unified auditing at the following locations:

- *Oracle Database Security Guide*: This guide is the main source of information for configuring auditing. You must use the Oracle Database Release 11g version of this manual. To access this guide:
 1. Visit Oracle Technology Network at the following URL:
<http://www.oracle.com/technetwork/index.html>
 2. From the Downloads menu, under **Databases**, select **Database 11g**.
 3. In the Downloads page, select the **Documentation** tab.
 4. From the most recent Oracle Database 11g Release 2 (11.2) Documentation page, select the **View Library** link to display the home page of the Release 11g documentation set.
 5. Under the **Search** field, select the **Master Book List** link.
 6. Search for **Security Guide**.
 7. Select either the **HTML** or the **PDF** link for this guide.
- *Oracle Database SQL Language Reference*: This guide explains how to use the `AUDIT` and `NOAUDIT` statements for both unified auditing and non-unified auditing environments.
- *Oracle Database Reference*: This guide explains how to use the initialization parameters and data dictionary views that are associated with a non-unified auditing environment. For a list of these, see *Oracle Database Security Guide*.
- *Oracle Database Vault Administrator's Guide*: This guide explains how to configure auditing in a non-unified auditing environment for Database Vault.
- *Oracle Label Security Administrator's Guide*: This guide explains how to configure auditing in a non-unified auditing environment for Oracle Label Security.

4.6.14 Test the Upgraded Production Oracle Database

If you upgraded a test database to the new Oracle Database release and then tested it, then you can now repeat those tests on the production database that you upgraded to the new Oracle Database 12c release. Compare the results, noting anomalies. Repeat the test upgrade as many times as necessary.

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

See Also: [Chapter 5, "Upgrading Applications After Upgrading Oracle Database"](#) for more information on using applications with Oracle Database

4.7 Recommended Tasks After Upgrading an Oracle RAC Database

Oracle Real Application Clusters 12c Release 1 (12.1) uses the Single Client Access Name (SCAN). The SCAN is a single name that resolves to three IP addresses in the public network. When a release of an Oracle RAC database earlier than release 11.2 is upgraded, it is registered with SCAN listeners as remote listeners, and also continues to register with all node listeners. You can configure clients to use SCANS, or continue to use the node listeners. If you migrate all of your client connections to use SCANS, you can then remove the node listeners from the `REMOTE_LISTENERS` parameter. However, you cannot remove the listeners themselves, because only node listeners can create dedicated servers for the database.

See Also: *Oracle Clusterware Administration and Deployment Guide* for more information on the Single Client Access Name (SCAN)

4.8 Recommended Tasks After Upgrading Oracle ASM

After you have upgraded Oracle ASM, Oracle recommends that you perform tasks such as resetting the Oracle ASM passwords and configuring disk groups.

The following tasks are recommended after upgrading Oracle ASM:

- [Create A Shared Password File in the ASM Diskgroup](#)
- [Reset Oracle ASM Passwords to Enforce Case-Sensitivity](#)
- [Advance the Oracle ASM and Oracle Database Disk Group Compatibility](#)
- [Set Up Oracle ASM Preferred Read Failure Groups](#)

You should also consider performing the following tasks, discussed earlier in this chapter:

- ["Add New Features as Appropriate"](#) on page 12
- ["Develop New Administrative Procedures as Needed"](#) on page 12

4.8.1 Create A Shared Password File in the ASM Diskgroup

If you advanced the `COMPATIBLE.ASM` disk group attribute to 12.1, then you must create a shared password file in the ASM diskgroup. See *Oracle Automatic Storage Management Administrator's Guide* for complete information about managing a shared password file in a disk group.

4.8.2 Reset Oracle ASM Passwords to Enforce Case-Sensitivity

You can enforce case sensitivity for passwords. For example, the password `hPP5620qr` fails if it is entered as `hpp5620QR` or `hPp5620Qr`.

In releases earlier than Oracle Database 11g Release 1 (11.1), passwords were not case sensitive. To take advantage of enforced case-sensitive passwords, you must reset the passwords of existing users during the database upgrade procedure. For new Oracle ASM instances, there are no additional tasks or management requirements. For upgraded Oracle ASM instances, each user password must be reset with an `ALTER USER` statement.

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

4.8.3 Advance the Oracle ASM and Oracle Database Disk Group Compatibility

You can advance the Oracle Database and the Oracle ASM disk group compatibility settings across software versions.

Caution: If you advance the `COMPATIBLE.RDBMS` attribute, then you *cannot* revert to the previous setting. Therefore, before advancing the `COMPATIBLE.RDBMS` attribute, ensure that the values for the `COMPATIBLE` initialization parameter for all of the databases that use the disk group are set to at least the new setting for `COMPATIBLE.RDBMS` before you advance the attribute value.

Advancing compatibility enables new features only available in the new release. However, doing so makes the disk group incompatible with older releases of the software. Advancing the on disk compatibility is an irreversible operation.

You use the `compatible.rdbms` and `compatible.asm` attributes to specify the minimum software release required by the database instance and the Oracle ASM instance, respectively, to access the disk group. For example, the following `ALTER DISKGROUP` statement advances the Oracle ASM compatibility of the disk group `asmhg2`:

```
ALTER DISKGROUP asmdg2 SET ATTRIBUTE 'compatible.asm' = '11.2'
```

In this case, the disk group can be managed only by Oracle ASM software of release 11.2 or later, while any database client of release 10.2 or later can use the disk group.

See Also: *Oracle Automatic Storage Management 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

4.8.4 Set Up Oracle ASM Preferred Read Failure Groups

Oracle ASM administrators can specify some disks to be preferred over others for read i/o operations. When Oracle ASM preferred read failure groups are defined, Oracle ASM can read from the extent that is closest to it, rather than always reading the primary copy.

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for information about specifying failure groups settings in an extended cluster
- *Oracle Automatic Storage Management Administrator's Guide* for complete information about Oracle ASM preferred read failure groups, and specifying the new `ASM_PREFERRED_READ_FAILURE_GROUPS` initialization parameter to list failure group names that contain the preferred read disks for each node in a cluster
- *Oracle Database Reference* for the `ASM_PREFERRED_READ_FAILURE_GROUPS` initialization parameter

4.9 Recommended Tasks After Upgrading Oracle Database Express Edition

An Oracle Database Express database contains only a subset of the components available in an Oracle Database Standard Edition or Oracle Database Enterprise Edition database. After upgrading to the new Oracle Database release, you can use the Database Configuration Assistant (DBCA) to install additional components into your database.

4.10 Optionally Update Oracle Application Express Packaged Applications

If your database originally included Oracle Application Express version 4.2 or up to version 4.2.5.00.08, then the 4.2.5 patch set was applied. However, the packaged applications that are shipped with Oracle Application Express were not updated to the 4.2.5 versions when the patch set was applied. You will need to run a script to update the packaged applications. If Oracle Application Express is installed in a non-CDB or is installed locally in a PDB, follow the instructions provided here.

To update the packaged applications in a non-CDB:

1. Set your current directory to the top-level "apex" directory in the Oracle home.
2. Start SQL*Plus and connect to the database where Oracle Application Express is installed as SYS specifying the SYSDBA role.

On Windows:

```
C:\ sqlplus /nolog
SQL> CONNECT SYS as SYSDBA
Enter password: SYS_password
```

On Linux:

```
$ sqlplus /nolog
SQL> CONNECT SYS as SYSDBA
Enter password: SYS_password
```

3. Run `apex_pkgapp_ins.sql` as shown in the following example:

```
@apex_pkgapp_ins.sql
```

To update the packaged applications in a CDB:

1. Set your current directory to the top-level "apex" directory in the Oracle home.

2. Start SQL*Plus and connect to the database where Oracle Application Express is installed as SYS specifying the SYSDBA role.

On Windows:

```
C:\ sqlplus /nolog
SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
```

On Linux:

```
$ sqlplus /nolog
SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
```

3. Run apex_pkgapp_con.sql as shown in the following example:

```
@apex_pkgapp_con.sql
```

4.11 Tasks to Complete Only After Manually Upgrading Oracle Database

If you are performing a manual upgrade of Oracle Database rather than using DBUA, then you must perform required tasks after your database is upgraded.

- [Change Passwords for Oracle Supplied Accounts](#)
- [Create or Migrate Your Password File with ORAPWD](#)
- [Migrate Your Initialization Parameter File to a Server Parameter File](#)
- [Upgrade Oracle Text](#)
- [Upgrade the Oracle Clusterware Configuration](#)
- [Adjust the Initialization Parameter File for the New Release](#)
- [Set CLUSTER_DATABASE Initialization Parameter For Oracle RAC](#)

4.11.1 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 12c security settings are in place, then passwords must be at least eight characters, and passwords such as *welcome* and *oracle* are not allowed. See *Oracle Database Security Guide* for more information.

You can view the status of all accounts by issuing the following SQL statement:

```
SQL> SELECT username, account_status
      FROM dba_users
      ORDER BY username;
```

To lock and expire passwords, issue the following SQL statement:

```
SQL> ALTER USER username PASSWORD EXPIRE ACCOUNT LOCK;
```


4.11.2 Create or Migrate Your Password File with ORAPWD

If the `REMOTE_LOGIN_PASSWORDFILE` initialization parameter is set to either `exclusive` or `shared`, create or migrate the password file with `ORAPWD`. Oracle Database 12c provides a new option to `ORAPWD` for migrating the password file from your existing database.

See Also: *Oracle Database Administrator's Guide* for more information about creating or migrating password files

4.11.3 Migrate Your Initialization Parameter File to a Server Parameter File

If you are currently using a traditional initialization parameter file, then perform the following steps to migrate to a server parameter file:

1. If the initialization parameter file is located on a client computer, then transfer the file from the client computer to the server computer.

Note: If you are using Oracle RAC, then you must combine all of your instance-specific initialization parameter files into a single initialization parameter file. Instructions and other actions unique to using a server parameter file for cluster databases, are discussed in:

- *Oracle Real Application Clusters Administration and Deployment Guide*
 - The *Oracle Real Application Clusters Installation Guide* for your operating system
-
-

2. Create a server parameter file using the `CREATE SPFILE` statement. This statement reads the initialization parameter file to create a server parameter file. You are not required to start the database to issue a `CREATE SPFILE` statement.
3. Start up the instance using the newly-created server parameter file.

See Also:

- *Oracle Database Administrator's Guide* for more information about creating server parameter files
- *Oracle Database SQL Language Reference* for information about the `CREATE SPFILE` statement

4.11.4 Upgrade Oracle Text

After an upgrade to the new Oracle Database 12c release, copy the following files from the previous Oracle home to the new Oracle home:

- Stemming user-dictionary files
- User-modified `KOREAN_MORPH_LEXER` dictionary files
- `USER_FILTER` executables

These files affect all databases installed in the given Oracle home.

You can obtain a list of these files as follows:

1. Read the text file at `$ORACLE_HOME/ctx/admin/ctxf102.txt`.
2. Run `$ORACLE_HOME/ctx/admin/ctxf102.sql` as database user `SYS`, `SYSTEM`, or `CTXSYS`.

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

4.11.5 Upgrade the Oracle Clusterware Configuration

If you are using Oracle Clusterware, then you must upgrade the Oracle Clusterware keys for the database.

Run `srvctl` for Oracle Database 12c to upgrade the database. For example:

```
ORACLE_HOME/bin/srvctl upgrade database -db name -o ORACLE_HOME
```

See Also: *Oracle Real Application Clusters Administration and Deployment Guide* for the syntax for `srvctl upgrade database`

4.11.6 Adjust the Initialization Parameter File for the New Release

Each release of Oracle Database introduces new initialization parameters, deprecates some initialization parameters, and desupports some initialization parameters. You must adjust the parameter file to account for these changes and to take advantage of new initialization parameters that might be beneficial to your system. Additionally, when you perform a manual upgrade without using DBUA, the `tnsnames.ora` file is not automatically populated with new configuration information and settings. Therefore, you must manually update `tnsnames.ora` and adjust `local_listener` and `remote_listener` parameter references if these must be resolved.

See Also:

- The "What's New in Oracle Database Reference" section of *Oracle Database Reference* for a list of the new initialization parameters in Oracle Database 12c, and for information about each parameter
- [Chapter 8, "Deprecated and Desupported Features for Oracle Database 12c"](#) for desupported and deprecated initialization parameters in Oracle Database 12c
- [Appendix A, "Changes for Earlier Releases of Oracle Database"](#)

4.11.6.1 Setting the COMPATIBLE Initialization Parameter

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

Complete the following steps to set the `COMPATIBLE` initialization parameter to a higher value:

1. Perform a backup of your database before you raise the `COMPATIBLE` initialization parameter (optional).

Raising the `COMPATIBLE` initialization parameter might cause your database to become incompatible with earlier releases of Oracle Database, and a backup ensures that you can return to the earlier release if necessary.

See Also: *Oracle Database Backup and Recovery User's Guide* for more information about performing a backup

2. If you are using a server parameter file, then complete the following steps:
 - a. Update the server parameter file to set or change the value of the `COMPATIBLE` initialization parameter.

For example, to set the `COMPATIBLE` initialization parameter to `11.0.0`, enter the following statement:

```
SQL> ALTER SYSTEM SET COMPATIBLE = '11.0.0' SCOPE=SPFILE;
```

- b. Shut down and restart the instance.

Note: When upgrading systems with HARD-compliant storage (Hardware Assisted Resilient Data), consider the following:

- If the `COMPATIBLE` parameter is set to a release number earlier than `11.0.0`, then you cannot locate the server parameter file (`SPFILE`) on HARD storage.
- If the `COMPATIBLE` parameter is set to `11.0.0`, then you can optionally locate the server parameter file on HARD storage.

Because the default `SPFILE` location (`ORACLE_HOME/dbs`) might not be on a HARD-compliant storage system, it is likely you must provide a parameter file that specifies the location of the `SPFILE`.

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 for Oracle Database release `12.1`, enter the following in the initialization parameter file:

```
COMPATIBLE = 12.1.0
```

- c. Start the instance using `STARTUP`.

Note: If you are using an ASM disk group, then the disk group's compatibility attribute must match or be lower than that of the database compatibility parameter in `init.ora`.

4.11.6.2 Configuring `tnsnames.ora` and Listener Parameters

After performing a manual upgrade, you must adjust `local_listener` and `remote_listener` parameter references if they must be resolved in `tnsnames.ora`. `DBUA` handles changes to network naming and listeners during automatic upgrades, but during a manual upgrade, `tnsnames.ora` is not changed, nor are the listeners.

See Also:

- Local Naming Parameters (tnsnames.ora) in *Oracle Database Net Services Reference*
- "Configuring the tnsnames.ora File After Installation" in *Oracle Database Net Services Administrator's Guide*
- "Configuring and Administering Oracle Net Listener" in *Oracle Database Net Services Administrator's Guide* for information on registering information with a local listener and a remote listener
- "Net Service Names (tnsnames.ora File)" in *Oracle Real Application Clusters Installation Guide for Microsoft Windows x64 (64-Bit) for Windows*
- "Net Service Names (tnsnames.ora File)" in *Oracle Real Application Clusters Installation Guide for Linux and UNIX*

4.11.7 Set CLUSTER_DATABASE Initialization Parameter For Oracle RAC

For upgrades of Oracle RAC databases, in ["Preparing the New Oracle Home for Upgrading"](#) on page 2-39 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`.

Upgrading Applications After Upgrading Oracle Database

Many new features and enhancements are available after upgrading to a new release of Oracle Database. To take full advantage of the new features, you must upgrade applications running in the new release.

This chapter contains the following topics:

- [Overview of Upgrading Applications on a New Oracle Database Release](#)
- [Upgrading Precompiler and OCI Applications in Oracle Database](#)
- [Upgrading SQL*Plus Scripts and PL/SQL after Upgrading Oracle Database](#)
- [About Upgrading Oracle Forms or Oracle Developer Applications](#)

5.1 Overview of Upgrading Applications on a New Oracle Database Release

Existing applications running in a new release of Oracle Database function the same as they did in earlier releases and achieve the same, or enhanced, performance. You are not required to modify existing applications that do not use features available in the new Oracle Database release.

Many new features and enhancements are available after upgrading to the new Oracle Database release. Some of these features provide added features and functions, while others provide improved performance. Before you upgrade your applications, you should review these new features to decide which ones you want to use.

See Also:

- *Oracle Database New Features Guide* for information about the new features available in Oracle Database 12c
- The Oracle Technology Network (OTN) at <http://www.oracle.com/technetwork/indexes/documentation/> for information about using individual Oracle Database 12c database products or options

5.2 Compatibility Issues for Applications on Different Releases of Oracle Database

There might be compatibility issues between different releases of Oracle Database that could affect your applications. These compatibility issues result from differences in Oracle Database in various releases. Also, in each new release of Oracle Database, new

Oracle reserved words might be added, changes might be made to initialization parameters, and changes might be made to the data dictionary.

When you upgrade your Oracle Database software to a new release, ensure 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, "Changes for Earlier Releases of Oracle Database"](#) for information about initialization parameter changes and data dictionary changes
- *Oracle Database SQL Language Reference* for a complete list of Oracle reserved words
- *Oracle Database Installation Guide* for your operating system-specific Oracle documentation for information about operating system requirements

5.3 Upgrading Precompiler and OCI Applications in Oracle Database

The upgrade path is very similar for precompiler and Oracle Call Interface (OCI) applications.

- Create a test environment before you upgrade your production environment.
- Include your upgraded application and the new Oracle Database software in your test environment.
- Ensure that your test environment provides a realistic test of your application.

See Also:

- ["Testing the Upgrade Process for Oracle Database"](#) on page 2-42
- *Pro*C/C++ Programmer's Guide*
- *Pro*COBOL Programmer's Guide*
- *Oracle Call Interface Programmer's Guide*

5.4 Software Upgrades and Client and Server Configurations for Oracle Database

To understand your options for upgrading precompiler and Oracle Call Interface (OCI) applications, you first must understand the type of software upgrade that you are performing and your client and server configurations.

- [Types of Software Upgrades for Oracle Database Client and Server Software](#)
- [Possible Client and Server Configurations for Oracle Database](#)

5.4.1 Types of Software Upgrades for Oracle Database Client and Server Software

Two types of upgrades are possible for Oracle Database client and server software: a major release of Oracle Database and maintenance release for Oracle Database.

- [Oracle Database Major Release Upgrade](#)

- [Oracle Database Maintenance Release Upgrade](#)

5.4.1.1 Oracle Database Major Release Upgrade

The upgrade changes the first digit of the release number. For example, upgrading from Oracle Database 11g to Oracle Database 12c is a major database release upgrade.

5.4.1.2 Oracle Database Maintenance Release Upgrade

The upgrade changes the second digit of the release number. For example, upgrading from Oracle Database 11g Release 1 (11.1) to Oracle Database 11g Release 2 (11.2) is a database maintenance release upgrade.

5.4.2 Possible Client and Server Configurations for Oracle Database

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 client/server configurations in your environment as follows:

- [Oracle Database Client and Server on Different Computers](#)
- [Oracle Database Client and Server in Different Oracle Locations on the Same Computer](#)
- [Oracle Database Client and Server in the Same Oracle Location](#)

5.4.2.1 Oracle Database Client and Server on 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.

5.4.2.2 Oracle Database Client and Server in Different Oracle Locations 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.

5.4.2.3 Oracle Database Client and Server in the Same Oracle Location

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* for more information about client/server environments

5.5 Compatibility Rules for Applications When Upgrading Oracle Database Client or Server Software

Compatibility rules apply when you upgrade Oracle Database client or server software. The rules are based on the type of software upgrade you are performing and the type of client/server configuration.

The following sections contain compatibility rules depending on the type of upgrade:

- [Rules for Upgrading Oracle Database Server Software](#)
- [Upgrading the Oracle Database Client Software](#)

Note: This section uses the terms introduced in ["Software Upgrades and Client and Server Configurations for Oracle Database"](#) on page 5-2.

5.5.1 Rules for Upgrading Oracle Database Server Software

Different rules apply when you upgrade Oracle Database server software depending on your database environment.

- [If You Do Not Change the Client Environment, Then You Are Not Required to Relink.](#)
- [Applications Can Run Against Newer or Older Oracle Database Server Releases](#)

5.5.1.1 If You Do Not Change the Client Environment, Then You Are Not Required to Relink.

If your client and server are on different computers or are in different Oracle home directories on the same computer, and you upgrade the Oracle Database server software without changing the client software, then you are not required to precompile, compile, or relink your applications. In these cases, the client software is separate from the server software and continues to function against the server.

However, if your applications are using the same Oracle home as the Oracle Database server, then your server upgrade also upgrades your client software, and you must follow the rules in ["Upgrading the Oracle Database Client Software"](#) on page 5-4.

Note: It is possible to upgrade the Oracle Database server software but not install the new precompiler or OCI client software when you are using the same Oracle home for both. In this case, the client software is not upgraded. However, such a configuration is not recommended.

5.5.1.2 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 later than the client software release, but this configuration is not strictly required. For example, if your client software is Oracle9i Release 2 (9.2.0.8), then Oracle recommends that your server software be Oracle9i Release 2 (9.2.0.8) or later to run a precompiler application on the client against the server.

5.5.2 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 12c, then Oracle recommends upgrading the client software to Oracle Database 12c 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.

Depending on how your applications are linked, different rules apply when you upgrade the Oracle Database client software.

- [Applications Can Run Against Newer or Older Oracle Database Server Releases](#)
- [Applications Can Be Linked with Newer Libraries](#)
- [Statically Linked Applications Must Always Be Relinked](#)
- [Relinking Dynamically Linked Applications](#)

5.5.2.1 Applications Can Be Linked with Newer Libraries

The code generated by precompiler applications can be linked with a release of the client library that equals or is later than the server release.

OCI applications can be linked with a release of the OCI run-time library that equals or is later than the release of the OCI library with which the application was developed.

5.5.2.2 Statically Linked Applications Must Always Be Relinked

Statically linked OCI applications must 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 becomes incompatible with the statically-linked code.

5.5.2.3 Relinking Dynamically Linked Applications

Dynamically linked OCI applications from Oracle Database 10g Release 1 (10.1) and later releases are upward compatible with the current release. That is, the Oracle client-side dynamic library is upward compatible with the previous version of the library. Oracle Universal Installer creates a symbolic link for the previous version of the library that resolves to the current version. Therefore, an application that is dynamically linked with the previous version of the Oracle client-side dynamic library does not require relinking to operate with the current version of the Oracle client-side library.

Note: If the application is linked with a run-time library search path (such as `-rpath` on Linux), then the application may still run with the version of the Oracle client-side library with which it is linked. To run with the current version of the Oracle client-side library, it must be relinked.

If the application is linked with the deferred option (for example, statically-linked application), then it must be relinked.

If the application is from a release earlier than Oracle Database 10g Release 1 (10.1), then it must be relinked.

5.6 Upgrading Options for Oracle Precompiler and OCI Applications

Oracle provides several options for upgrading your precompiler and Oracle Call Interface (OCI) applications running on a new release of Oracle Database.

The upgrade 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.

- [Option 1: Leave the Application Unchanged](#)
- [Option 2: Precompile or Compile the Application Using the New Software](#)

- [Option 3: Change the Application Code to Use New Oracle Database Features](#)

5.6.1 Option 1: Leave the Application Unchanged

Leave the application and its environment unchanged. Do not relink, precompile, or compile the application, and do not change the application code. The application continues to work against the new Oracle Database 12c. 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 Oracle Database 12c server. The major advantage to this option is that it is simple and easy. In addition, this option requires the least amount of administration, because you are not required to upgrade any of your client computers. If you have a large number of client computers, then avoiding the administrative costs of upgrading all of them can become very important.

The major disadvantage to this option is that your application cannot use the features that are available in the new release of Oracle Database. In addition, your application cannot leverage all the possible performance benefits of Oracle Database 12c.

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

Precompile or compile and then relink the application using the new release of Oracle Database. Application code must be changed if any APIs are deprecated or changed. When upgrading to the new release of Oracle Database 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.

This option requires that you install the new Oracle Database client software on each client computer. You are required to precompile or compile, and relink your application only one time, regardless of the number of clients you have.

By recompiling, you perform a syntax check of your application code. Some problems in the application code that were not detected by previous releases of the Oracle software might emerge when you precompile or compile with the new Oracle software. Therefore, precompiling and compiling with the new software often helps you detect and correct problems in the application code that might have gone unnoticed before.

Also, recompiling affords maximum stability for your application, because you are sure that it works with the new Oracle software. Further, your environment is ready for new development using the latest tools and features available. In addition, you might benefit from performance improvements that are available with the new Oracle software only after you recompile and relink.

5.6.3 Option 3: Change the Application Code to Use New Oracle Database Features

Change the application code to use new features in Oracle Database 12c. Then, precompile or compile and then relink the code. You can make code changes to your application to take advantage of new Oracle Database features. This option is the most difficult, but it can provide the most potential benefits. You gain all of the advantages described in "[Option 2: Precompile or Compile the Application Using the New Software](#)" on page 6. In addition, you also benefit from changes to your application that might leverage performance and scalability benefits available with the new release of Oracle Database. Further, you can add new features to your application that are available only with the new release.

Become familiar with the features of Oracle Database 12c 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 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](#)

5.6.3.1 Changing Precompiler Applications

Complete the following steps to change your precompiler application to use features of Oracle Database 12c:

1. If you want to take advantage of new features in Oracle Database 12c, then incorporate the code for these new features into the existing application.
2. Precompile the application using the Oracle precompiler.
3. Compile the application.
4. Relink the application with the run-time library of the new Oracle Database 12c, `SQLLIB`, which is included with the precompiler.

5.6.3.2 Changing OCI Applications

Complete the following steps to change your OCI application to use features of Oracle Database 12c:

1. Incorporate OCI calls of the new Oracle Database 12c into the existing application.
2. Compile the application.
3. Relink the application with the run-time library of the new Oracle Database 12c.

5.7 Upgrading SQL*Plus Scripts and PL/SQL after Upgrading Oracle Database

To use features and functions of the new Oracle Database release, change existing SQL scripts to use the syntax of the new Oracle Database release. Existing SQL scripts run unchanged on the new Oracle Database release, and require no modification, if they do not use features and functions of the new Oracle Database release.

Improved error checking in the new Oracle Database release might now identify errors at compile time rather than at run time.

5.7.1 Evaluation of Numeric Literals

Evaluation of numeric literals has changed such that at least one constant in a numeric computation with literals must be a decimal specified to the 10th place. This is because Oracle Database releases later than Oracle Database 10g Release 1 (10.1) use `INTEGER` arithmetic (approximately nine significant digits) for some expressions whereas Oracle9i Release 2 (9.2) used `NUMBER` arithmetic (approximately 38 significant digits). Therefore, if you are dealing with results of greater than nine (9) significant digits, then one literal should be in decimal format to prevent numeric overflow errors. For example, in Oracle Database 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 numeric literal as a decimal (256.0), as follows:

```
DECLARE
  v1 NUMBER(38);
BEGIN
  v1 := 256*256*256*256.0;
  DBMS_OUTPUT.PUT_LINE(v1);
END;
/
```

See Also:

- The "What's New in SQL*Plus" section in the *SQL*Plus User's Guide and Reference* to learn about new features in SQL*Plus
- *Oracle Database SQL Language Reference* for more information about upgrading SQL scripts

5.8 About Upgrading Oracle Forms or Oracle Developer Applications

Forms applications run the same on Oracle9i, Oracle Database 10g, and Oracle Database 11g. In Oracle Database 12c, the title of *Oracle Database Development Guide* has been changed to *Oracle Database Advanced Application Developer's Guide*.

Review the new features and changes described for developing applications. Also, review the new features described in *Oracle Database New Features Guide* to determine whether any of the features of Oracle Database 12c would be beneficial to your applications or might otherwise affect them.

Downgrading Oracle Database to an Earlier Release

For supported releases of Oracle Database, you can downgrade a database to the release from which you last upgraded. For example, if you recently upgraded from release 11.2.0.4 to Oracle Database 12c, and you did not change the compatible initialization parameter to 12.1 or higher, then you can downgrade to release 11.2.0.4. If your Oracle Database 12c is release 12.1.0.2, and you did not change the compatible initialization parameter to 12.1.0.2, then you can downgrade to release 12.1.0.1 and so forth.

This chapter contains the following topics:

- [Supported Releases for Downgrading Oracle Database](#)
- [Checking for Incompatibilities When Downgrading Oracle Database](#)
- [Performing a Full Backup Before Downgrading Oracle Database](#)
- [Performing Required Pre-Downgrade Steps for Oracle Database](#)
- [Downgrading a CDB or Non-CDB Oracle Database](#)
- [Downgrading a Single Pluggable Oracle Database \(PDB\)](#)
- [Post-Downgrade Tasks for Oracle Database Downgrades](#)
- [Troubleshooting the Downgrade of Oracle Database](#)

See Also: *Oracle Database Installation Guide* for your operating system for discussions of downgrading that are operating system-specific

6.1 Supported Releases for Downgrading Oracle Database

You can downgrade both major releases and patchset releases, based on the original Oracle Database release from which the database was upgraded. Major release downgrades are supported back to Oracle Database 11g Release 2 (11.2.0.2) and subsequent 11.2 releases, and Oracle Database 11g Release 1 (11.1.0.7). Patchset downgrades are supported back to Oracle Database release 11.2 patchsets, except release 11.2.0.1.

[Table 6–1](#) provides a summary of releases supported for downgrading. When using this table, also read about compatibility in "[Checking for Incompatibilities When Downgrading Oracle Database](#)" on page 4.

Table 6–1 Supported Releases and Editions for Downgrading

| Oracle Database Release or Edition | Downgradable? | Notes |
|------------------------------------|---------------|---|
| 12.1.0.2 | Y | <p>You cannot downgrade a database once you have set the compatible initialization parameter to 12.1.0.2.</p> <p>Only if the compatibility is set to 12.1.0.1 will a downgrade be possible for a pluggable database (PDB).</p> <p>The patch set update PSU4 is required in order to downgrade a CDB, or unplug and downgrade a PDB. PSUs are available for download on My Oracle Support (MOS) at https://support.oracle.com/. See MOS Note 756671.1 to obtain the latest patch set and any required set of additional fixes. (Search in the general search box on the Patches and Updates page.)</p> <p>You cannot downgrade back to release 10.2.0.5 because the minimum compatibility setting for Oracle Database 12c is 11.0.</p> |
| 12.1.0.1 | Y | <p>If you unplug a 12.1.0.1 PDB from a 12.1.0.1 database and then plug this PDB into a 12.1.0.2 database for upgrade, then you cannot downgrade this PDB if the compatible initialization parameter in the 12.1.0.2 database is higher than '12.1.0.1.0'.</p> <p>You cannot downgrade back to release 10.2.0.5 because the minimum compatibility setting for Oracle Database 12c is 11.0.</p> |
| Oracle Enterprise Manager | N | <p>You must reconfigure Oracle Enterprise Manager controls if you downgrade to an earlier supported release. See "Restoring Oracle Enterprise Manager after Downgrading Oracle Database" on page 15.</p> |
| Oracle Database Express Edition | N | <p>You cannot downgrade a database that was upgraded from Oracle Database Express Edition</p> |

Important: The following information is very important to understand regarding compatibility if you expect to be able to downgrade.

- You cannot downgrade a database once you have set the compatible initialization parameter to 12.1.0.2.
 - Only if the compatibility is set to 12.1.0.1 will a downgrade be possible for a pluggable database (PDB).
 - If you unplug a release 12.1.0.1 PDB from a 12.1.0.1 database and then plug this PDB into a release 12.1.0.2 database, you cannot downgrade this PDB.
 - PSU4 (that is *patch set update*) is required in order to downgrade a CDB or unplug and downgrade a PDB. See MOS Note 756671.1 to obtain the latest patch set, and any required set of additional fixes, at My Oracle Support (MOS) at <https://support.oracle.com/>
-
-

Note: You cannot downgrade back to release 10.2.0.5 because the minimum compatibility setting for Oracle Database 12c is 11.0. Additionally, you cannot downgrade a database that was upgraded from Oracle Database Express Edition.

The following recommendations for earlier supported releases affect downgrading for Oracle Database:

- **This release includes multitenant architecture, which provides features for a multitenant container database (CDB) and pluggable databases (PDBs). Therefore, setting the compatible initialization parameter to the highest level after upgrading to this release prevents the capability to downgrade.**
- This release contains a new object privilege, `READ`, in addition to `SELECT`. After you downgrade:
 - If you have the `SELECT` and `READ` object privileges, the `READ` privilege is removed.
 - If you only had the `READ` object privilege, then this is transformed into the `SELECT` object privilege.

See *Oracle Database Security Guide* for information about the `READ` and `SELECT` object privileges.

- The earliest release that you can downgrade back to is Oracle Database release 11.1.0.7. Because the minimum compatibility for Oracle Database 12c is 11.0, after you upgrade to release 12.1, the compatibility must be set to at least 11.0. Once compatibility is set, this changes the on-disk structure of the database and enables new features that are controlled by the compatible setting. Therefore, downgrade to a release earlier than 11.1.0.7 is no longer an option.
- If Oracle XML DB was not present in the database to be upgraded, then Oracle XML DB is uninstalled during the downgrade. For example, if you did not install Oracle XML DB with Oracle Database 11g Release 2 (11.2), then when upgrading to Oracle Database 12c, Oracle XML DB is installed because it is included with this release. During a downgrade back to Oracle Database 11g Release 2 (11.2), Oracle XML DB is removed.

- During upgrade to Oracle Database 12c, the Database (DB) Control repository is removed. If you downgrade, you must reconfigure the Database (DB) Control to use it again in the earlier, downgraded release. See ["Saving Oracle Enterprise Manager Database Control Configuration and Data"](#) on page 2-12.
- Downgrade is not supported for Oracle Enterprise Manager. You must reconfigure Oracle Enterprise Manager controls if you downgrade to an earlier supported release. See ["Restoring Oracle Enterprise Manager after Downgrading Oracle Database"](#) on page 6-15.
- Oracle Clusterware for Oracle Database 12c does not support downgrading to Oracle Database release 10.2 and release 11.1 on Windows. A fresh install is required for releases earlier than release 11.2.

See Also: ["Oracle Database Releases That Support Direct Upgrade"](#) on page 1-5

6.2 Checking for Incompatibilities When Downgrading Oracle Database

Check the compatibility level of your database to see if the database might have incompatibilities that prevent you from downgrading. If the compatibility level of your Oracle Database 12c database is 12.1.0, then you are not able to downgrade.

See Also: ["Checking the Compatibility Level of Oracle Database"](#) on page 1-13

If you are downgrading to Oracle Database 11g Release 2 (11.2.0.2), Oracle Database 11g Release 2 (11.2.0.3), or Oracle Database 11g Release 2 (11.2.0.4), then the `COMPATIBLE` initialization parameter must be set to 11.2.0 or lower. Downgrading to release 11.2.0.1 is not supported.

If you are downgrading to Oracle Database 11g Release 1 (11.1.0.7), then the `COMPATIBLE` initialization parameter must be set to 11.1.0 or lower. Only downgrading to release 11.1.0.7 is supported for Oracle Database 11g Release 1.

Note: For Oracle ASM disk groups, if you changed disk group compatibility to 12.1.0.0.0 when you upgraded your database, then when you downgrade to the earlier release you will be unable to mount your Oracle ASM disk groups.

You must manually restore compatibility of Oracle ASM disk groups before downgrade. Otherwise, the instance cannot mount the disk groups after downgrade.

See *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM disk group compatibility.

See Also: ["About Downgrading and Compatibility for Upgrading Oracle Database"](#) on page 1-12 and [Appendix A, "Changes for Earlier Releases of Oracle Database"](#)

6.3 Performing a Full Backup Before Downgrading Oracle Database

Oracle strongly recommends that you perform a full backup of your Oracle Database 12c database before you downgrade to a supported earlier releases.

See Also: *Oracle Database Backup and Recovery User's Guide* for information about performing RMAN backups

6.4 Performing Required Pre-Downgrade Steps for Oracle Database

Before downgrading to the earlier release from which you upgraded, you must complete required preparation. For example, you may need to determine if components in the database must be disabled first.

To prepare for downgrading:

1. If you are downgrading a CDB or unplugging and downgrading a PDB in Oracle Database release 12.1.0.2, then you must first apply PSU4 (patch set update 4). You obtain PSU4 and any required set of additional fixes by going to My Oracle Support at <https://support.oracle.com/>. The patch set information and link is contained in MOS Note 756671.1. Search for 756671.1 in the general search box on the Patches and Updates page. This is a requirement before you can downgrade.
2. If you are downgrading to Oracle Database release 11.1, then before starting the downgrade procedures, ensure that you know the password for any newly-created database links. These are database links that you may have created while running releases 11.2 or 12.1. Knowing the database link password is necessary because you will need to reset the database link password after downgrading to Oracle Database release 11.1. See "[Issue with Password for Database Links After Downgrading](#)" on page 20 for more information.
3. If you have enabled Oracle Database Vault on your database, then disable Oracle Database Vault before downgrading the database.

To find if Oracle Database Vault is enabled, query the V\$OPTION dynamic view as follows:

Enter Oracle Database Vault in the case shown in this query. If the output is TRUE, then Oracle Database Vault is enabled, so you must disable it. See *Oracle Database Vault Administrator's Guide* for instructions.

4. If your database uses Oracle Label Security **and** you are downgrading to release 11.2 or earlier, then run the Oracle Label Security (OLS) preprocess downgrade `olspre downgrade.sql` script in the new Oracle Database 12c Oracle home. You do not need to run `olspre downgrade.sql` if you are downgrading release 12.1.0.2 to release 12.1.0.1.

Important: Running the `olspre downgrade.sql` script before downgrading is mandatory for downgrading from Oracle Database 12c to database release 11.2 (and earlier) in databases that use Oracle Label Security and Oracle Database Vault.

- a. To find out if Oracle Label Security is enabled, query the V\$OPTION dynamic view as follows:

```
SQL> SELECT VALUE FROM V$OPTION WHERE PARAMETER = 'Oracle Label Security';
```

- b. Run the `olspre downgrade.sql` script.

```
SQL> @ORACLE_HOME/rdbms/admin/olspre downgrade.sql
```

See *Oracle Label Security Administrator's Guide* for more information.

5. If you have enabled unified auditing, then optionally back up and purge the unified audit trail.

- a. Find if unified audit records exist.
- b. Back up the existing audit data to a table. For example:

```
SQL> CREATE TABLE UA_DATA AS (SELECT * FROM V$UNIFIED_AUDIT_TRAIL);
```

- c. Clean up the audit trail.

```
EXEC DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL(audit_trail_type => DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED, use_last_arch_timestamp => FALSE);
```

6. If you previously installed a recent version of the time zone file and used the DBMS_DST PL/SQL package to upgrade `TIMESTAMP WITH TIME ZONE` data to that version, then install the same version of the time zone file in the release to which you are downgrading. If after the database upgrade, you had used DBMS_DST to upgrade the `TIMESTAMP WITH TIME ZONE` data to the latest version of the time zone file, then install this latest version time zone file in the release to which you are downgrading. This ensures that your `TIMESTAMP WITH TIME ZONE` data is not logically corrupted during retrieval.

To find which time zone file version your database is currently using, query `V$TIMEZONE_FILE` as follows:

```
SQL> select * from V$TIMEZONE_FILE;
```

Copy the time zone files from the current Oracle home to the Oracle home to which you are downgrading. For example, after you see the results of your query, you might choose from the following:

```
%ORACLE_HOME%\oracore\zoneinfo\timezlg_18.dat
%ORACLE_HOME%\oracore\zoneinfo\timezone_18.dat
```

See *Oracle Database Globalization Support Guide* for more information on upgrading time zone files, and *Oracle Database PL/SQL Packages and Types Reference* for information about the DBMS_DST package.

7. If you set the `ORA_TZFILE` environment variable to the full path name of the time zone file when you upgraded to Oracle Database 12c, then you must unset it if you subsequently downgrade your database.

Two time zone files are included in the Oracle home directory:

- The default time zone file, which is for the database server, at
`$ORACLE_HOME/oracore/zoneinfo/timezone1rg.dat`
- A smaller time zone file, which is typically used for clients, at
`$ORACLE_HOME/oracore/zoneinfo/timezone.dat`

If you do not unset the `ORA_TZFILE` variable, then connecting to the database using the smaller time zone file might produce the following errors:

```
SP2-1503: Unable to initialize Oracle call interface
SP2-0152: ORACLE may not be functioning properly
```

See *Oracle Database Globalization Support Guide* for more information about date and time data types and time zone support.

8. Optional: If you are downgrading to Oracle Database 11g Release 1 (11.1.0.7) and you have Oracle Application Express in your database, then you must copy the

`apxrelod.sql` file from the Oracle Database 12c `ORACLE_HOME/apex/` directory to a directory outside of the Oracle home, such as a temporary directory on your system.

Make a note of the new location of this file. You need this later in the procedure.

9. If you have created objects based on fixed objects, then drop these objects to avoid possible `ORA-00600` errors. You can re-create these objects after the downgrade.
10. If you have Oracle Enterprise Manager configured in your database, then drop the Enterprise Manager user:

```
DROP USER sysman CASCADE;
```

Note: After this step, `MGMT*` synonyms may be invalid. You must reconfigure Oracle Enterprise Manager to use any Oracle Enterprise Manager controls in the downgraded database.

6.5 Downgrading a CDB or Non-CDB Oracle Database

Oracle provides a procedure for downgrading your Oracle Database 12c database to a supported major release or a relevant patchset upgrade. If you are downgrading from release 12.1.0.2 or higher to release 12.1.0.1, you can downgrade all databases in a multitenant container database (CDB) or one pluggable database (PDB) within a CDB. Oracle Database releases earlier than Oracle Database 12c did not use multitenant architecture. See "[Downgrading a Single Pluggable Oracle Database \(PDB\)](#)" on page 6-13 for the procedure for downgrading one PDB.

The procedure in this section assumes that you understand "[Supported Releases for Downgrading Oracle Database](#)" on page 1.

To downgrade the database, whether a CDB or not, to an earlier supported release:

1. Log in to the system as the owner of the Oracle Database 12c Oracle home directory.
2. At a system prompt, change to the `ORACLE_HOME/rdbms/admin` directory, where `ORACLE_HOME` is the Oracle home on your system.

Note: If you are downgrading a cluster database, then shut down the database completely and change the `CLUSTER_DATABASE` initialization parameter to `FALSE`. After the downgrade, you must set this parameter back to `TRUE`.

3. Using SQL*Plus, connect to the database instance as a user with `SYSDBA` privileges.

```
sqlplus sys as sysdba
Enter password: password
```

4. Log in to the system as the owner of the Oracle home under the new Oracle Database 12c.
5. From the `ORACLE_HOME/rdbms/admin` directory, start SQL*Plus.
6. Connect to the database to be upgraded using an account with `DBA` privileges:

```
CONNECT / AS SYSDBA
```

7. Start the instance in downgrade mode by issuing the following SQL*Plus command for your Oracle Database instance type. You may be required to use the PFILE option to specify the location of your initialization parameter file.

For Non-CDB instances:

```
SQL> startup downgrade pfile=pfile_name
```

For CDB instances:

```
SQL> startup downgrade pfile=pfile_name
SQL> alter pluggable database all open downgrade;
```

Specify the location of your initialization parameter file PFILE. See *Oracle Database Administrator's Guide* for information about specifying initialization parameters at startup and the initialization parameter file.

8. Optional: If you are downgrading a non-CDB, you can set the system to spool results to a log file so you can track the changes and issues. (A CDB automatically spools output to the catcon_logs; therefore, you can skip this step for a CDB.)

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

9. Run catdwgrd.sql.

For a non-CDB:

```
SQL> @catdwgrd.sql
```

For a CDB:

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl -d
$ORACLE_HOME/rdbms/admin -e -b catdwgrd -l <output directory> -r
catdwgrd.sql
```

In the CDB example catcon.pl is run with catdwgrd. You first start PERL. The -d parameter tells catcon.pl where to find catdwgrd. The -l parameter specifies the output directory for log files (instead of writing to the rdbms/admin directory). You must use the -r parameter to run the two scripts together.

Note:

- You must use the version of the catdwgrd.sql script included with Oracle Database 12c.
 - You must run catdwgrd.sql in the Oracle Database 12c environment.
 - You must specify the -r parameter in order to run the two scripts together at the same time.
 - The catdwgrd.sql script downgrades all Oracle Database components in the database to the supported major release or patch release from which you originally upgraded.
-
-

If you encounter any problems when you run the catdwgrd.sql 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.

Caution: 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.

10. For non-CDB only, 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 8 and the suggested name was `downgrade.log`. Correct any problems you find in this file and rerun the downgrade script if necessary.

Note: If you want to save the results of the first time you ran the downgrade script, then before you rerun it be sure to rename `downgrade.log` to something else.

11. Shut down the instance:

```
SQL> SHUTDOWN IMMEDIATE
```

12. Exit SQL*Plus.

13. 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 *Oracle Database Installation Guide* for your operating system for information about setting other important environment variables on your operating system.

14. 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 12c 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 *Oracle Database Platform Guide for Microsoft Windows* for more information about stopping Oracle services on Windows.

- 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 MANUAL -PFILE ORACLE_HOME\DATABASE\INITSID.ORA
```

See *Oracle Database Platform Guide for Microsoft Windows* for information about administering an Oracle Database instance using ORADIM. The syntax for ORADIM includes the following variables:

| Variable | Description |
|--------------------|---|
| <i>SID</i> | Same SID name as the SID of the database being downgraded. |
| <i>PASSWORD</i> | Password for the database instance. This is the password for the user connected with SYSDBA privileges. The -INTPWD option is not required. If you are prompted for a password, use the password for the standard user account for this Windows platform. |
| <i>USERS</i> | Maximum number of users who can be granted SYSDBA and SYSOPER 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 -PFILE option, including drive letter of the Oracle home directory. See <i>Oracle Database Administrator's Guide</i> for information about specifying initialization parameters at startup and the initialization parameter file. |

For example, 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 MANUAL -PFILE C:\ORANT\DATABASE\INITORCL.ORA
```

Note: The ORADIM command prompts you for the password for the Oracle home user account. You can specify other options using ORADIM.

You are not required to change any Windows Registry settings when downgrading a database. The ORADIM utility makes all necessary changes automatically.

15. 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 on 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`.

See *Oracle Real Application Clusters Administration and Deployment Guide* for information about initialization parameter use in Oracle RAC.

16. At a system prompt, change to the `ORACLE_HOME/rdbms/admin` directory of the earlier release to which you are downgrading.

17. Start SQL*Plus and connect to the database instance as a user with `SYSDBA` privileges.

For a non-CDB:

```
SQL> CONNECT / AS SYSDBA
SQL> STARTUP UPGRADE
```

For a CDB:

```
connect / as sysdba
startup database mount;
alter database open upgrade;
alter pluggable database all open upgrade;
```

18. Optional: For a non-CDB, set the system to spool results to a log file to track changes and issues. This step is not needed for a CDB.

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

19. Run `catrelod.sql`.

For a non-CDB:

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

For a CDB:

```
$ORACLE_HOME/perl/bin/perl catcon.pl -n 1 -e -b catrelod -d ''.'''
catrelod.sql
```

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

20. If you are downgrading to Oracle Database 11g Release 1 (11.1.0.7) and you have Oracle Application Express in your database, then change to the directory to which you had copied the `apxrelod.sql` script. Manually reload Oracle Application Express by running the `apxrelod.sql` script:

```
SQL> @apxrelod.sql
```

Running the `apxrelod.sql` script avoids package `APEX_030200.WWV_FLOW_HELP` being `INVALID` due to the following error:

```
PLS-00201: identifier 'CTX_DDL' must be declared
```

21. If you had turned spooling on for a non-CDB, now 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. Correct any problems you find in this log file and rerun the appropriate script if necessary.

22. 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. See *Oracle Database Administrator's Guide* for information about specifying initialization parameters at startup and the initialization parameter file.

23. Perform this step if the database is configured for Oracle Label Security.

- a. Copy the `olstrig.sql` script from the Oracle home under Oracle Database 12c to the Oracle home of the release to which the database is to be downgraded.
- b. From the Oracle home of the release you are downgrading to, run `olstrig.sql` to re-create DML triggers on tables with Oracle Label Security policies:

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

See *Oracle Label Security Administrator's Guide* for more information about Oracle Label Security.

24. Run the `utlrp.sql` script.

For a non-CDB:

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

For a CDB run `catcon.pl` to invoke `utlrp.sql` to recompile any remaining stored PL/SQL and Java code. Use the following syntax:

```
$ORACLE_HOME/perl/bin/perl catcon.pl -n 1 -e -b utlrp -d ''.''' 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. The log file `utlrp0.log` is generated with the results of the recompilations.

25. Exit `SQL*Plus`.

26. If you are downgrading a cluster database, then you must run the following command to downgrade the Oracle Clusterware database configuration:

```
$ srvctl downgrade database -d db-unique-name -o oraclehome -t to_version
```

[where `db-unique-name` is the database name (not the instance name), `oraclehome` is the location of the old Oracle home for the downgraded database, and `to_version` is the database release to which the database is downgraded. (For example: 11.2.0.3.0.)]

Note: Execute this command from the current Oracle Database 12c Oracle home, not from the Oracle home to which the database is being downgraded.

Your database is now downgraded.

6.6 Downgrading a Single Pluggable Oracle Database (PDB)

If you are downgrading the latest release of Oracle Database 12c, you can downgrade one PDB without downgrading the whole CDB. For example, you can unplug a PDB from a release 12.1.0.2 CDB, downgrade it, and then plug it in to a release 12.1.0.1 CDB or convert the database to a standalone database.

The procedure in this section assumes that you understand "[Supported Releases for Downgrading Oracle Database](#)" on page 1.

Step 1 Downgrade the PDB

In this step you downgrade the PDB to release 12.1.0.1.

1. Start up the release 12.1.0.2 PDB in DOWNGRADE mode. The CDB can be in normal mode when you do this.

```
SQL> alter pluggable database CDB1_PDB1 open downgrade;
```

2. Downgrade the PDB by running `catdwgrd`, which in this example is PDB1.

Run `catdwgrd` as follows:

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl -d
$ORACLE_HOME/rdbms/admin -n 1 -l <output directory> -e -b catdwgrd -c 'PDB1'
catdwgrd.sql
```

In the example `catcon.pl` is run with `catdwgrd`. You first start PERL. The `-d` parameter tells `catcon.pl` where to find `catdwgrd`. The `-l` parameter specifies the output directory for log files (instead of writing to the `rdbms/admin` directory). You must use the `-r` parameter to run the two scripts together at the same time.

3. Close the PDB.

Step 2 Unplug the PDB from the CDB

In this step you unplug the downgraded PDB from the release 12.1.0.2 CDB.

1. Connect to the 12.1.0.2 CDB.
2. Close the PDB to be unplugged.

```
SQL> alter pluggable database PDB1 close;
```

3. Unplug the downgraded 12.1.0.1 PDB.

```
SQL> alter pluggable database PDB1 unplug into 'path/PDB1.xml';
```

You will see Pluggable database altered.

Step 3 Plug in the Downgraded 12.1.0.1 PDB

In this step you plug the downgraded 12.1.0.1 PDB into the 12.1.0.1 CDB. To do this, you must create the PDB in this CDB. In this example, we create PDB1.

1. Connect to the 12.1.0.1 CDB.
2. Plug in the 12.1.0.1 PDB.

```
SQL> create pluggable database PDB1 using 'path/PDB1.xml';
```

You will see Pluggable database created.

3. Open the PDB in upgrade mode in order to run the scripts shown in the subsequent steps in this procedure.

```
SQL> alter pluggable database PDB1 open upgrade;
```

4. Connect to the PDB:

```
SQL> alter session set container=PDB1;
```

5. Run `catrelod` in the PDB:

```
SQL> @$ORACLE_HOME/rdbms/admin/catrelod.sql
```

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

6. Run `utlrlp` in the PDB:

```
SQL> @$ORACLE_HOME/rdbms/admin/utlrlp.sql
```

The `utlrlp.sql` script recompiles all existing PL/SQL modules that were previously in an `INVALID` state, such as packages, procedures, types, and so on.

6.7 Post-Downgrade Tasks for Oracle Database Downgrades

Additional tasks may be required after downgrading an Oracle database due to changes that affect compatibility, components, and supported protocols.

This section contains the following topics:

- [Oracle XML DB Authentication Recommendations for an Oracle Database Downgrade](#)
- [Re-creating the Network Listener When Downgrading to Release 11.1.0.7](#)
- [Re-enabling Oracle Database Vault after Downgrading Oracle Database](#)
- [Restoration of the Configuration for Oracle Clusterware](#)
- [Restoring Oracle Enterprise Manager after Downgrading Oracle Database](#)
- [Restoring Oracle Application Express to the Earlier Version](#)

6.7.1 Oracle XML DB Authentication Recommendations for an Oracle Database Downgrade

If you downgrade to a release that is earlier than Oracle Database 12c, in which digest authentication is not supported, digest authentication is disabled and made unavailable as an authentication choice. This affects HTTP authentication for Oracle XML DB Repository. If you did not take advantage of digest authentication and instead used the default configuration, then no further actions are necessary.

See Also: *Oracle XML DB Developer's Guide* for information about authentication with Oracle XML DB for database installation, upgrade, and downgrade

6.7.2 Re-creating the Network Listener When Downgrading to Release 11.1.0.7

If you are downgrading to Oracle Database 11g Release 1 (11.1.0.7), then you must re-create the listener after performing the downgrade steps. Run Oracle Net Manager to delete the old listener and create a new listener.

- To start Oracle Net Manager from the Oracle Enterprise Manager console, select **Service Management** from the Tools menu, then select **Oracle Net Manager**.

- On Windows, select **Programs** from the Start menu, and then select **Oracle - HOME_NAME**. Next, select **Configuration and Migration Tools**, and then **Net Manager**.

See Also: *Oracle Database Net Services Administrator's Guide*

6.7.3 Re-enabling Oracle Database Vault after Downgrading Oracle Database

If you use Oracle Database Vault, then you may have been instructed to disable it before downgrading your database. To use Oracle Database Vault after downgrading, you must re-enable it.

See Also: *Oracle Database Vault Administrator's Guide* for the complete procedure

6.7.4 Restoration of the Configuration for Oracle Clusterware

You can restore the Oracle Clusterware configuration to the state it was in before the Oracle Clusterware 12c Release 1 (12.1) upgrade. To restore the configuration, you must restore the release from which you were upgrading. Any configuration changes that you have performed during or after the Oracle Database 12c upgrade process are removed and cannot be recovered.

See Also:

- *Oracle Grid Infrastructure Installation Guide for Linux*
- *Oracle Grid Infrastructure Installation Guide for Microsoft Windows x64 (64-Bit)*
- *Oracle Clusterware Administration and Deployment Guide*

6.7.5 Restoring Oracle Enterprise Manager after Downgrading Oracle Database

The *restore* task described in this section is required only if you are downgrading in some form and Oracle Enterprise Manager is configured on the host. To restore Oracle Enterprise Manager to its previous state, you must have saved your Oracle Enterprise Manager configuration files and data before upgrading. You restore your configuration by first running the Oracle Enterprise Manager configuration assistant (EMCA) as described in this section, and then running the `emdwgrd` utility as described in ["Running the emdwgrd utility to restore Enterprise Manager Database Control"](#) on page 6-17.

See Also: ["Saving Oracle Enterprise Manager Database Control Configuration and Data"](#) on page 2-12

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 release to which the database is being downgraded.

See Also: *Oracle Clusterware Administration and Deployment Guide* for information about `srvctl`

Run the `emca -restore` command with the appropriate options to restore Oracle Enterprise Manager Database Control or Grid Control to the old Oracle home.

Important: Use the Oracle Database 12c version of `emca` for this procedure. See *Oracle Database Administrator's Guide* for complete information about `emca`.

The options that you specify for `emca` depend on whether the database being downgraded is a single-instance database, an Oracle RAC database, or an Oracle ASM database as follows:

- [Running emca on a Single-Instance Oracle Database Without Oracle ASM](#)
- [Running emca on an Oracle RAC Database Without Oracle ASM](#)
- [Running emca on a Single-Instance Oracle ASM Instance](#)
- [Running emca on an Oracle ASM on Oracle RAC Instance](#)
- [Running emca on a Single-Instance Oracle Database With Oracle ASM](#)
- [Running emca on an Oracle RAC Database and Oracle ASM Instance](#)

Running emca on a Single-Instance Oracle Database Without Oracle ASM

```
ORACLE_HOME/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

Running emca on an Oracle RAC Database Without Oracle ASM

```
ORACLE_HOME/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

Running emca on a Single-Instance Oracle ASM Instance

```
ORACLE_HOME/bin/emca -restore asm
```

You are prompted to enter the following information:

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

Running emca on an Oracle ASM on Oracle RAC Instance

```
ORACLE_HOME/bin/emca -restore asm -cluster
```

You are prompted to enter the following information:

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

Running emca on a Single-Instance Oracle Database With Oracle ASM

```
ORACLE_HOME/bin/emca -restore db_asm
```

You are prompted to enter the following information:

- Oracle home for the database to be restored
- Database SID
- Listener port number
- Oracle ASM port
- Oracle ASM home
- Oracle ASM SID [+ASM]

Running emca on an Oracle RAC Database and Oracle ASM Instance

```
ORACLE_HOME/bin/emca -restore db_asm -cluster
```

You are prompted to enter the following information:

- Oracle home for the database to be restored
- Database unique name
- Listener port number
- Oracle ASM port
- Oracle ASM Oracle home
- Oracle ASM SID [+ASM]

The output of `emca` varies according to the options that you specify and the values that you enter at the prompts. This step must be repeated on all the nodes in Oracle RAC environments.

You must now run the `emdwrgrd` utility to restore Oracle Enterprise Manager Database Control and data.

See Also: ["Running the emdwgrd utility to restore Enterprise Manager Database Control"](#) on page 6-17

6.7.5.1 Running the emdwgrd utility to restore Enterprise Manager Database Control

After completing the `emca -restore` procedure you are ready to restore the Oracle Enterprise Manager Database Control and data with the `emdwrgrd` utility. You must run the `emdwrgrd` utility from the Oracle Database 12c home. Then `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 Oracle Enterprise Manager Database Control 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 12c Oracle 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_directory -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 the 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. When `emdwrgrd` completes, Oracle Enterprise Manager Database Control is downgraded to the old Oracle home.

6.7.6 Restoring Oracle Application Express to the Earlier Version

If the database that was upgraded had an Oracle Application Express version earlier than version 4.2, then downgrading the database will have caused Oracle Application Express to also be downgraded. You will need to compile the SYS-owned objects `WWV_FLOW_VAL` and `WWV_DBMS_SQL` from the earlier version of the Oracle Application Express source.

1. Change your working directory to the core directory of the earlier version of Application Express source.
2. Connect to the database as SYS / SYSDBA.
3. Run the following scripts in the order shown:
 - a. `@wv_flow_val.plb`
 - b. `@wv_dbms_sql.sql`
 - c. `@wv_dbms_sql.plb`
4. If you were using the embedded PL/SQL gateway of the Oracle XML DB Protocol Server, you will need to restore the images directory to the earlier version as follows.
 - a. Change your working directory to the earlier version of the Application Express source.
 - b. Invoke `apxldimg.sql` connected as SYS as SYSDBA passing the path to the earlier version source as in the following example.

```
@apxldimg.sql /tmp/apex32
```

6.8 Troubleshooting the Downgrade of Oracle Database

Oracle provides troubleshooting information for issues that may occur when downgrading Oracle Database. This section contains known errors that may occur and corresponding workarounds when needed.

This section contains the following topics:

- [Oracle Multimedia Downgrade and imrelod.sql Script Error](#)
- [Oracle Database Vault and dvrelod.sql Script Error](#)
- [Downgrading Grid Infrastructure Standalone \(Oracle Restart\) After Successful or Failed Upgrade](#)
- [Issue with Downgrading Oracle ACFS and Oracle Grid Infrastructure](#)
- [Issue with Password for Database Links After Downgrading](#)
-

6.8.1 Oracle Multimedia Downgrade and imrelod.sql Script Error

When downgrading Oracle Database from release 12.1 to release 11.2.0.2, an error may be raised from the `imrelod.sql` script, which is included with release 11.2.0.2.

ORA-20000: Oracle ORDIM component in registry is status: DOWNGRADED. Oracle ORDIM must be installed and valid prior to Oracle Multimedia upgrade, downgrade, or patch.

Cause: The `imrelod.sql` script raises this error because it does not know the status of ORDIM.

Action: No action. This error can safely be ignored.

6.8.2 Oracle Database Vault and dvrelod.sql Script Error

When downgrading Oracle Database from release 12.1 to release 11.2.0.3, databases that use Oracle Database Vault may encounter an error from the `dvrelod.sql` script, which is included with release 11.2.0.3.

ORA-31011: XML parsing failed; Oracle Database Vault downgrade to release 11.2.0.3

Cause: The `dvrelod.sql` script does not know the status of XML parsing.

Action: No action. This error can safely be ignored.

6.8.3 Downgrading Grid Infrastructure Standalone (Oracle Restart) After Successful or Failed Upgrade

In order to downgrade Oracle Restart, you must deconfigure and then reinstall Oracle Grid Infrastructure. You can then add back the databases and services.

See Also:

- Support note 1364412.1 on My Oracle Support at <http://support.oracle.com>
- *Oracle Grid Infrastructure Installation Guide for Linux* or *Oracle Grid Infrastructure Installation Guide for Microsoft Windows x64 (64-Bit)* for information about downgrading Oracle Clusterware after an upgrade

6.8.4 Issue with Downgrading Oracle ACFS and Oracle Grid Infrastructure

If you use Oracle ASM Cluster File System (Oracle ACFS), and you upgrade to Oracle Grid Infrastructure 12c, either for a cluster or for an Oracle Restart standalone server, and then you choose to downgrade to Oracle Grid Infrastructure 11g Release 2 (11.2), you must run the command `acfsroot install` from the release 11.2 Oracle home before you attempt to start the release 11.2 software stack.

See Also: *Oracle Database Readme* and the release notes for your operating system platform for the latest information about issues and solutions

6.8.5 Issue with Password for Database Links After Downgrading

After downgrading to Oracle Database release 11.1, you are required to reset the passwords for any database links that were created in releases 11.2 or 12.1. See "[Using Oracle Data Pump Export to Create a Dump File Containing All Existing Database Links](#)" on page 20. Unless the database link password is reset, an internal error will be displayed when anyone attempts to make use of the database link. For reference, the internal error which is reported in the oracle trace file when the oracle server fails to retrieve the password of the database link is as follows: "ORA-00600: [kzdlk_zt2 err], [18446744073709551601]".

To reset the password for the database link after downgrading to release 11.1, use the `ALTER DATABASE LINK` command to change the password back to the original password by specifying the original password in the `IDENTIFIED BY` clause. See *Oracle Database SQL Language Reference* for information about `ALTER DATABASE LINK`, and *Oracle Database Administrator's Guide* for information about controlling connections established by database links.

If you want to create new database links while running Oracle Database release 11.2 or 12.1 that do not have this password issue, then you can contact Oracle support for information on how to use the `IDENTIFIED BY VALUES` clause of the `CREATE DATABASE LINK` command.

6.8.5.1 Using Oracle Data Pump Export to Create a Dump File Containing All Existing Database Links

Before performing the downgrade, please use Oracle Data Pump export to create a dump file that contains all the existing database links. This will include any newly-created database links. The procedure uses the `FULL=Y` and `INCLUDE=DB_LINK` parameters with the `expdp` command.

To create the dump file that contains all the existing database links:

1. Log in to SQL*Plus.

```
sqlplus system/manager
```

2. Drop the dump directory in case one exists.

```
SQL> DROP DIRECTORY dpump_dir;
SQL> CREATE DIRECTORY dpump_dir AS '/location_to_write_datapump_dump_file';
```

3. Export the database links as follows:

```
$ expdp system/manager FULL=Y directory=dpump_dir
dumpfile=saved_dblink.dmp INCLUDE=DB_LINK;
```

4. Optional step for troubleshooting. After the downgrade, if any of the downgraded database links are not working properly (for example, exhibiting the internal error

ORA-00600), they can be dropped and imported from the dump file as follows, after which they should work as intended.

```
$ impdp system/manager directory=dpump_dir dumpfile=saved_dblinks.dmp;
```

Migrating Data Using Oracle Data Pump

To facilitate upgrading Oracle Database, you can use the Export and Import utilities in Oracle Data Pump to migrate data from one database to another. Oracle Data Pump provides high performance Export (expdp) and Import (impdp) utilities.

This chapter contains the following topics:

- [Overview of Data Pump and Export/Import For Migrating Data](#)
- [Migrating Data With Data Pump When Upgrading Oracle Database](#)
- [Data Pump Requirements When Downgrading Oracle Database](#)

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

7.1 Overview of Data Pump and Export/Import For Migrating Data

Oracle provides Data Pump Export and Import to *migrate*, or move, data from one Oracle database to another. This is useful after upgrading Oracle Database.

Data Pump offers the following benefits:

- Supports filtering the metadata that is exported and imported based upon objects and object types, using `INCLUDE` and `EXCLUDE` parameters.
- Supports different modes for unloading/loading portions of the database including: full database mode, schema mode, table mode, tablespace mode, and transportable tablespace mode. (See the topics about Data Pump export modes and Data Pump import modes in *Oracle Database Utilities*.)
- Enables you to specify how partitioned tables should be handled during import operations, using the `PARTITION_OPTIONS` parameter.
- Provides support for the full range of data types.

See Also:

- *Oracle Database Utilities* for an overview of Data Pump Export and Import
- ["The Export/Import Method for Migrating Data When Upgrading Oracle Database"](#) on page 2-4 for information on when to use this method

7.2 Migrating Data With Data Pump When Upgrading Oracle Database

You can take advantage of Oracle Data Pump to export data from the source database before you install the new Oracle Database software, and then import the data into the target upgraded database.

To use Oracle Data Pump with the upgrade procedures:

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, the current database must not be available for updates during and after the export. If the current database is available to users for updates after the export, then, before making the current database available, put procedures in place to copy the changes made in the current database to the new database after the import is complete.

2. Install the new Oracle Database software. Installation steps for Oracle Database are covered in your operating system-specific Oracle documentation. Refer to *Oracle Database Installation Guide* for your operating system.
3. If the new database has the same name as the current database, then shut down the current database before creating the new database.
4. Create the new database.

See Also: *Oracle Database Administrator's Guide* for information about creating a database

5. Start SQL*Plus in the new Oracle Database environment.
6. Connect to the database instance as a user with SYSDBA privileges.
7. Start an Oracle Database instance using `STARTUP`.
8. Optionally, you can change the storage parameters from the source database.

You can pre-create tablespaces, users, and tables in the new database to improve space usage by changing storage parameters. When you pre-create tables using SQL*Plus, either run the database in the original database compatibility mode or make allowances for the specific data definition conversions that occur during import.

When items have been pre-created, specify the `TABLE_EXISTS_ACTION=APPEND` option for Data Pump 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 an option 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 a parameter to save the informational messages and error messages from the import session to a file as follows:

- 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 the imports of specific objects that completed successfully. If there were failures, check for information about any objects that failed.

See Also:

- *Oracle Database Utilities*
- *Oracle Database Readme*

11. Use further Import scenarios as described in *Oracle Database Utilities*, or use 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 an unrecoverable error, then the job can be restarted after the condition inducing the failure is corrected. The job continues automatically from the point of failure.

12. If changes are made to the current database after the export, then make sure those changes are propagated to the new database before making it available to users. Refer to step 1 in this procedure.
13. Complete the procedures described in [Chapter 4, "Post-Upgrade Tasks for Oracle Database"](#).

7.2.1 Importing a Full Oracle Database Using a Network Link

You can use the Data Pump Import utility with a database link to perform a full database import from a source database to a destination database without intermediate dump files. This method is an alternative to the procedure in "[Migrating Data With Data Pump When Upgrading Oracle Database](#)" on page 7-2

Note: In order to avoid interoperability errors, ensure that you have applied the appropriate patchset to the database being upgraded. See My Oracle Support at <http://support.oracle.com> to obtain the latest patchsets.

You may also refer to support note ID 4511371.8, which discusses ORA-6544 and ORA-4052 errors.

To use a network link with the Data Pump Import utility, `impdp`:

1. Ensure that the exporting user at the source database has the DATAPUMP_EXP_FULL_DATABASE role.

This user must be specified when you create the database link.

2. Ensure that the importing user at the destination database has the `DATAPUMP_IMP_FULL_DATABASE` role.
3. Create and test a database link between the source and destination databases.
4. Run the following command, where `import_user` is the username for the importing user, and `db_link` is the name of the database link owned by the exporting user:

```
IMPDP import_user NETWORK_LINK=db_link FULL=Y;
```

5. A log file for the import operation writes to the `DATA_PUMP_DIR` directory. You can discover the location of this directory by running the following command:

```
SQL> select * from dba_directories where DIRECTORY_NAME like 'DATA_PUMP_DIR';
```

Note: XML objects are not exported from the source database.

See Also: Note 466181.1 on My Oracle Support at <http://support.oracle.com> for more information on Data Pump Imports using a database link

Note: The import operation re-creates users on the new destination server, and the creation date for `dba_users` shows the actual import date. The expiration date is updated to be `creation_date + password_life_time`. The parameters for `dba_users` on the new server are different than the `dba_users` parameters on the source server.

7.3 Data Pump Requirements When Downgrading Oracle Database

When using Data Pump with the downgrade process, the Oracle Database release to which you downgrade can be one release earlier, at the most.

To obtain a downward compatible dump file with Data Pump Export:

- Use the Data Pump Export utility in the *current* release, and set the `VERSION` parameter to the release of the earlier target to which you are downgrading.

Data Pump Import cannot read dump file sets created by a database release that is later than the current database release, unless those dump file sets were created with the `VERSION` parameter set to the release number of the target database. Therefore, the best way to perform a downgrade is to use Data Pump Export with the `VERSION` parameter set to the release number of the target database to which you are downgrading.

See Also: *Oracle Database Utilities* for more information about using the `VERSION` parameter

Deprecated and Desupported Features for Oracle Database 12c

Oracle Database 12c introduces behavior changes for your database in addition to new features. Changes in behavior include deprecated and desupported initialization parameters, options, syntax, and the deprecation and desupport of features and components. This chapter does not describe new features introduced in Oracle Database 12c, nor does it describe changes in earlier releases of Oracle Database.

This chapter describes behavior changes where features have been deprecated or desupported in this release. By deprecate, we mean that the feature is no longer being enhanced but is still supported for the full life of the 12.1 release. By desupported, we mean that Oracle will no longer fix bugs related to that feature and may remove the code altogether. Where indicated, a deprecated feature may be desupported in a future major release. For more information about support considerations, refer to My Oracle Support at <http://support.oracle.com>.

This chapter contains the following topics:

- [Oracle Database Changes](#)
- [Oracle Business Intelligence and Data Warehousing Changes](#)
- [Oracle Database Security Changes](#)
- [Oracle Globalization Support Changes](#)
- [Oracle Multimedia Changes](#)
- [Oracle Net Services Changes](#)
- [Oracle XML Database Changes](#)
- [Oracle Text Changes](#)

See Also:

- *Oracle Database New Features Guide* for a complete list of all new features introduced in this release
- *Oracle Database Reference* for a list of new initialization parameters, new static data dictionary views, and new dynamic performance views
- *Oracle Database Installation Guide* for your operating system
- [Appendix A, "Changes for Earlier Releases of Oracle Database"](#)

8.1 Oracle Database Changes

Oracle Database 12c introduces changes that affect Oracle Database in general.

This section contains these topics:

- [Deprecation of Non-CDB Architecture](#)
- [Deprecation of catupgrd.sql Script and Introduction of New catctl.pl Utility](#)
- [Desupport of Oracle Enterprise Manager Database Control](#)
- [Changes for Deinstallation and Cleanup of Oracle Base](#)
- [Deprecated and Desupported Parameters](#)
- [Deprecated Views](#)
- [Deprecation of Oracle Streams](#)
- [Deprecation of Advanced Replication](#)
- [Deprecation of Single-Character SRVCTL CLI Options](#)
- [Desupported Features on Microsoft Windows Platforms](#)
- [Deprecation of Stored List of Administrative Users for Cluster Administration](#)
- [Desupport for Raw Storage Devices](#)
- [Deprecation of -checkpasswd for QOSCTL Quality of Service \(QoS\) Command](#)
- [Change to VARCHAR2, NVARCHAR2, and RAW Datatypes](#)
- [Deprecation of JPublisher](#)
- [Changes for Oracle JDBC and SQLJ](#)
- [Changes for Oracle Call Interface](#)
- [Deprecation of Oracle Restart](#)
- [Changed Default for resource_limit Parameter](#)

8.1.1 Deprecation of Non-CDB Architecture

The non-CDB architecture is deprecated in Oracle Database 12c, and may be desupported and unavailable in a release after Oracle Database 12c Release 2. Oracle recommends use of the CDB architecture.

Note: There remain a small number of features that do not work with the CDB architecture (see README, section 2.2.1 "Features Restricted or Not Available for a Multitenant Container Database"). If you need these features, then continue to use the non-CDB architecture until your required feature works with the CDB architecture.

8.1.2 Deprecation of catupgrd.sql Script and Introduction of New catctl.pl Utility

Oracle Database 12c introduces the new Parallel Upgrade Utility, `catctl.pl`. This utility replaces the `catupgrd.sql` script that was used in earlier releases. Although you can still use the `catupgrd.sql` script, it is deprecated starting with Oracle Database 12c and will be removed in future releases. Oracle recommends database upgrades be performed with the new Parallel Upgrade Utility, `catctl.pl`.

If you choose to run the `catupgrd.sql` script instead of running `catctl.pl`, doing so now requires an additional input parameter as follows:

```
SQL> catupgrd.sql PARALLEL=NO
```

If you run `catupgrd.sql` without the parameter, then Oracle displays the following error message:

NOTE

The `catupgrd.sql` script is being deprecated in the 12.1 release of Oracle Database. Customers are encouraged to use `catctl.pl` as the replacement for `catupgrd.sql` when upgrading the database dictionary.

```
cd $ORACLE_HOME/rdbms/admin
$ORACLE_HOME/perl/bin/perl catctl.pl -n 4 catupgrd.sql
```

Refer to the Oracle Database Upgrade Guide for more information.

This database upgrade procedure must be called with the following argument when invoking from the SQL prompt:

```
@catupgrd.sql PARALLEL=NO
```

See Also: ["About the Parallel Upgrade Utility for Oracle Database"](#) on page 3-26 for information about `catctl.pl` and how to run it

8.1.3 Desupport of Oracle Enterprise Manager Database Control

Starting with Oracle Database 12c, Oracle Enterprise Manager Database Control is desupported and is no longer available. Oracle introduces Oracle Enterprise Manager Database Express (Oracle EM Express) as a replacement. Oracle EM Express is installed when you upgrade to Oracle Database 12c.

See Also:

- *Oracle Database Concepts* for an overview of Oracle EM Express
- *Oracle Database 2 Day DBA* for information about using Oracle EM Express
- *Oracle Database Concepts* for an introduction to Oracle Enterprise Manager Cloud Control

8.1.4 Changes for Deinstallation and Cleanup of Oracle Base

The following topics discuss changes for deinstallation and cleanup of Oracle Base:

- [Change for Standalone Deinstallation Tool](#)
- [Desupport of CLEANUP_ORACLE_BASE Property](#)
- [Deprecation of -cleanupOBase](#)

8.1.4.1 Change for Standalone Deinstallation Tool

Starting with Oracle Database 12c, the deinstallation standalone utility is replaced with a `deinstall` option using Oracle Universal Installer (OUI). You can also run the deinstallation tool using the `runInstaller` command on Linux and UNIX (or `setup.exe` on Windows) with the `-deinstall` and `-home` options from the base

directory of the installation media for Oracle Database, Oracle Database Client, or Oracle Grid Infrastructure.

See Also:

- *Oracle Database Installation Guide* for your operating system for information about using OUI and runInstaller
- *Oracle Grid Infrastructure Installation Guide* for your operating system

8.1.4.2 Desupport of CLEANUP_ORACLE_BASE Property

In Oracle Database 12c, the CLEANUP_ORACLE_BASE property is removed and does not support an Oracle base removal during silent or response file mode deinstalls.

8.1.4.3 Deprecation of -cleanupOBase

The -cleanupOBase flag of the deinstallation tool is deprecated in this release.

See Also: *Oracle Database Installation Guide* for your operating system

8.1.5 Deprecated and Desupported Parameters

In Oracle Database 12c, some database parameters are deprecated and desupported, or removed.

To obtain a current list of deprecated parameters, run the following query in SQL*Plus:

```
SQL> SELECT name from v$parameter
        WHERE isdeprecated = 'TRUE' ORDER BY name;
```

The query returns a list of the deprecated parameters as of the date and time that you run it. The following example shows the query results that appear:

```
active_instance_count
background_dump_dest
buffer_pool_keep
buffer_pool_recycle
commit_write
cursor_space_for_time
fast_start_io_target
global_context_pool_size
instance_groups
lock_name_space
log_archive_start
max_enabled_roles
parallel_automatic_tuning
parallel_io_cap_enabled
parallel_server
parallel_server_instances
plsql_v2_compatibility
remote_os_authent
resource_manager_cpu_allocation
sec_case_sensitive_logon
serial_reuse
sql_trace
standby_archive_dest
user_dump_dest
```

8.1.5.1 LOG_ARCHIVE_LOCAL_FIRST

The LOG_ARCHIVE_LOCAL_FIRST initialization parameter is removed and desupported in Oracle Database 12c.

8.1.5.2 SEC_CASE_SENSITIVE_LOGON

The SEC_CASE_SENSITIVE_LOGON initialization parameter is deprecated in this release. See *Oracle Database Reference* for information about the SEC_CASE_SENSITIVE_LOGON initialization parameter.

8.1.5.3 FILE_MAPPING

The FILE_MAPPING initialization parameter is deprecated. It is still supported for backward compatibility. See *Oracle Database Reference* for information about the FILE_MAPPING initialization parameter.

8.1.5.4 DELETE_CATALOG_ROLE of the O7_DICTIONARY_ACCESSIBILITY Parameter

The DELETE_CATALOG_ROLE role is deprecated in Oracle Database 12c. See *Oracle Database Reference* for information about this role and dictionary objects.

8.1.5.5 RDBMS_SERVER_DN

RDBMS_SERVER_DN is deprecated in release 12.1.0.2. LDAP_DIRECTORY_ACCESS should be used instead. See *Oracle Database Reference* for information about this parameter.

8.1.6 Deprecated Views

The following views are deprecated in Oracle Database 12c:

- ALL_SCHEDULER_CREDENTIALS view. See *Oracle Database Reference* for more information.
- DBA_NETWORK_ACL_PRIVILEGES view. See *Oracle Database Reference* for more information.
- DBA_NETWORK_ACLS view. See *Oracle Database Reference* for more information.
- DBA_SCHEDULER_CREDENTIALS view. See *Oracle Database Reference* for more information.
- USER_NETWORK_ACL_PRIVILEGES view. See *Oracle Database Reference* for more information.
- USER_SCHEDULER_CREDENTIALS view. See *Oracle Database Reference* for more information.
- V\$OBJECT_USAGE view. Use the USER_OBJECT_USAGE view instead. See *Oracle Database Reference* for more information.

See Also: *Oracle Database Administrator's Guide* for information about specifying Scheduler job credentials

8.1.7 Deprecation of Oracle Streams

Oracle Streams is deprecated in Oracle Database 12c and may be desupported and unavailable in a later Oracle Database release. Use Oracle GoldenGate to replace all replication features of Oracle Streams.

Note: Oracle Database Advanced Queuing is independent of Oracle Streams and continues to be enhanced.

See Also: *Oracle GoldenGate documentation* for information

8.1.8 Deprecation of Advanced Replication

Oracle Database Advanced Replication is deprecated in Oracle Database 12c. Use Oracle GoldenGate to replace all features of Advanced Replication, including multimaster replication, updatable materialized views, hierarchical materialized views, and deployment templates.

See Also: *Oracle GoldenGate documentation* for information

8.1.9 Deprecation of Single-Character SRVCTL CLI Options

The Server Control Utility (SRVCTL) command line interface (CLI) supports long GNU-style options in addition to short CLI options used in earlier releases. Starting with Oracle Database 12c, single-character options are deprecated and may be desupported in a later release.

See Also: *Oracle Real Application Clusters Administration and Deployment Guide* for information about SRVCTL

8.1.10 Desupported Features on Microsoft Windows Platforms

In Oracle Database 12c, some features have been deprecated or desupported for the Windows platform. The following topics discuss these changes:

- [Desupport of Oracle COM Automation on Windows](#)
- [Desupport of Oracle Objects for OLE](#)
- [Desupport of Oracle Counters for Windows Performance Monitor](#)
- [Desupport of Oracle Cluster File System \(OCFS\) on Windows](#)
- [Changes for Oracle Data Provider for .NET](#)

8.1.10.1 Desupport of Oracle COM Automation on Windows

Oracle Database 12c does not contain Oracle COM Automation. This was deprecated in Oracle Database 11g, which is the last database release that contains the database component Oracle COM Automation. Oracle recommends that you migrate your Oracle COM applications to current technology such as the .NET Framework.

See Also: My Oracle Support Note 1175293.1 "Obsolescence Notice: Oracle COM Automation" on <http://support.oracle.com>

8.1.10.2 Desupport of Oracle Objects for OLE

Oracle Database 12c does not contain Oracle Objects for OLE. This was deprecated in Oracle Database 11g. You can migrate your code to the OLE DB data access standard and ActiveX Data Objects (ADO), or you can migrate your applications to .NET (or Java or another application architecture) and use another driver.

See Also: My Oracle Support Note 1175303.1 "Obsolescence Notice: Oracle Objects for OLE" on <http://support.oracle.com>

8.1.10.3 Desupport of Oracle Counters for Windows Performance Monitor

Oracle Database 12c does not contain Oracle Counters for Windows Performance Monitor. This was deprecated in Oracle Database 11g. The counters were not installed by default in earlier releases, and the counters only work on Windows. For monitoring, Oracle recommends that you use Oracle Enterprise Manager Cloud Control.

See Also: My Oracle Support Note 1175297.1 "Obsolescence Notice: Oracle Counters for Windows Performance Monitor" on <http://support.oracle.com>

8.1.10.4 Changes for Oracle Data Provider for .NET

The following Oracle Data Provider for .NET application programming interfaces for Transaction Guard are deprecated in Oracle Database 12c (12.1.0.2), and may be desupported in a future release.

- `OracleLogicalTransactionStatus` class
- `OracleConnection.LogicalTransactionId` property
- `OracleConnection.GetLogicalTransactionStatus` method

8.1.11 Desupport of Oracle Cluster File System (OCFS) on Windows

Starting with Oracle Database 12c, Oracle Cluster File System (OCFS) is desupported on Windows. Support and distribution of OCFS on Linux (OCFS and OCFS2) remains unaffected by this desupport notice.

Databases currently using OCFS on Windows to host either the Oracle cluster files (Oracle Cluster Registry and voting files) or database files or both need to have these files migrated off OCFS *before* upgrading to Oracle Database 12c.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for information about migrating a database to use Oracle Automatic Storage Management (Oracle ASM)
- My Oracle Support Note 1392280.1 "Desupport of Oracle Cluster File System (OCFS) on Windows with Oracle Database 12" on <http://support.oracle.com>
- "[About Upgrading Oracle Database Release 10.2 or 11.1 and OCFS and RAW Devices](#)" on page 8-7

8.1.11.1 About Upgrading Oracle Database Release 10.2 or 11.1 and OCFS and RAW Devices

If you are upgrading an Oracle Database release 10.2.0.5 or release 11.1.0.7 environment that stores Oracle Clusterware files on OCFS on Windows or RAW devices, then you cannot directly upgrade to Oracle Database 12c. You must first perform an interim upgrade to Oracle Database release 11.2 and migrate the Oracle Clusterware files to Oracle Automatic Storage Management (Oracle ASM). Then you can upgrade from release 11.2 to Oracle Database 12c.

See Also:

- The *Oracle Database Upgrade Guide* for release 11.2 for procedures on upgrading from release 10.2.0.5 or 11.1.0.7; you may need to download this document from Oracle Technology Network at <http://www.oracle.com/technetwork/>
- *Oracle Database Installation Guide* for your operating system for procedures on installing the Oracle Database software
- *Oracle Automatic Storage Management Administrator's Guide* for information about migrating a database to use Oracle Automatic Storage Management (Oracle ASM)

8.1.11.2 Desupport of cluvfy comp cfs for OCFS

The `cluvfy comp cfs` component verification command is removed from Oracle Database as Oracle Cluster File System (OCFS) is no longer supported.

See Also: *Oracle Clusterware Administration and Deployment Guide* for more information about cluster administration

8.1.12 Deprecation of Stored List of Administrative Users for Cluster Administration

Starting with Oracle Database 12c, the method of cluster administration using a stored list of administrative users is being replaced with more comprehensive management of administrative user roles by configuring the access control list of the policy set.

See Also: *Oracle Clusterware Administration and Deployment Guide* for more information about cluster administration

8.1.13 Desupport for Raw Storage Devices

Starting with Oracle Database 12c, Oracle Database and related technologies, such as Oracle Clusterware, no longer support the direct use of raw (block) storage devices. This means that files must be moved to Oracle ASM or a supported shared file system (cluster file system or Network File System (NFS)) prior to upgrading to Oracle Clusterware 12c.

This also affects the OCR and voting files for Oracle Clusterware. You cannot store the OCR or voting files on raw devices. Oracle Clusterware files stored on raw devices must be moved to Oracle ASM, a cluster file system, or Network File System (NFS) before upgrading.

See Also: "[About Upgrading Oracle Database Release 10.2 or 11.1 and OCFS and RAW Devices](#)" on page 8-7

8.1.14 Deprecation of -checkpasswd for QOSCTL Quality of Service (QoS) Command

The syntax `qosctl -checkpasswd username password` has been deprecated. See *Oracle Database Quality of Service Management User's Guide* for information about QOSCTL syntax and commands.

8.1.15 Change to VARCHAR2, NVARCHAR2, and RAW Datatypes

Starting with Oracle Database 12c, the maximum size of the VARCHAR2, NVARCHAR2, and RAW datatypes has been increased to 32767 bytes when the COMPATIBLE initialization parameter is set to 12.0 and the MAX_STRING_SIZE initialization parameter is set to EXTENDED.

See Also: *Oracle Database SQL Language Reference* for more information about `MAX_STRING_SIZE`

8.1.16 Deprecation of JPublisher

Oracle JPublisher is deprecated in Oracle Database 12c Release 1, as of October 2014, and all JPublisher features are desupported and unavailable in Oracle Database 12c Release 2. Oracle recommends that you use the alternatives listed here:

- To continue to use Web service callouts, Oracle recommends that you use the OJVM Web Services Callout utility, which is a replacement for the Web Services Callout utility.
- To create Java client applications for PL/SQL programs and SQL objects, Oracle recommends that developers use other JDK development tools that assist you to create java STRUCT classes, and other prestructured options.

See Also:

My Oracle Support Note 1937939.1 for more information about JPublisher deprecation and desupport:

<https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1937939.1>

JDK Tools and Utilities on Oracle Technology Network:

<http://docs.oracle.com/javase/8/docs/technotes/tools/>

8.1.17 Changes for Oracle JDBC and SQLJ

In Oracle Database 12c, some features are deprecated or desupported that affect JDBC, UCP, SQLJ, JPublisher, and Java in the Database (Oracle JVM). The JDBC Javadoc also contains a page for all the deprecated APIs.

The following features are deprecated or desupported in this release for Oracle JDBC and SQLJ:

- [Deprecation of SQLJ Inside the Server](#)
- [Deprecated Oracle Update Batching](#)
- [Deprecated EndToEndMetrics related APIs](#)
- [Deprecated Stored Outlines](#)
- [Deprecated Concrete Classes in oracle.sql Package](#)
- [Desupported Implicit Connection Caching](#)

8.1.17.1 Deprecation of SQLJ Inside the Server

SQLJ usage inside the database server is deprecated in this release. The capability of translating and running SQLJ applications inside the database will not be available in later releases. SQLJ can only be used as a client tool to translate the applications that can connect to Oracle Database and run as a client. SQLJ cannot be used inside stored procedures, functions, or triggers.

8.1.17.2 Deprecated Oracle Update Batching

Oracle Update Batching is deprecated in Oracle Database 12c. Use Standard Update Batching instead.

The following APIs are deprecated and marked deprecated in the JDBC Javadoc:

- `OraclePreparedStatement.setExecuteBatch()`
- `OraclePreparedStatement.getExecuteBatch()`
- `OracleCallableStatement.setExecuteBatch()`

See Also: *Oracle Database JDBC Developer's Guide* for information on Oracle Update Batching

8.1.17.3 Deprecated EndToEndMetrics related APIs

`EndToEndMetrics` -related APIs are deprecated in Oracle Database 12c. Use Universal Connection Pool instead.

The following APIs are deprecated and marked deprecated in the JDBC Javadoc:

- `getEndToEndMetrics`
- `getEndToEndECIDSequenceNumber`
- `setEndToEndMetrics`
- `setApplicationContext`
- `clearAllApplicationContext`

See Also: *Oracle Database JDBC Developer's Guide* and the JDBC Javadoc for information about Universal Connection Pool

8.1.17.4 Deprecated Stored Outlines

Stored outlines are deprecated in Oracle Database 12c. Use plan baselines instead.

See Also: *Oracle Database SQLJ Developer's Guide* for information about plan baselines

8.1.17.5 Deprecated Concrete Classes in oracle.sql Package

The concrete classes in the `oracle.sql` package are deprecated in Oracle Database 12c. These are replaced with new interfaces in the `oracle.jdbc` package.

See Also: *Oracle Database JDBC Developer's Guide* for information about the new interfaces for `oracle.jdbc`

8.1.17.6 Deprecated defineColumnType Method

The JDBC method `defineColumnType` is deprecated in this release.

8.1.17.7 Deprecated CONNECTION_PROPERTY_STREAM_CHUNK_SIZE Property

The JDBC property `CONNECTION_PROPERTY_STREAM_CHUNK_SIZE` is deprecated in this release.

8.1.17.8 Desupported Implicit Connection Caching

Implicit Connection Caching is desupported in Oracle Database 12c. Use Universal Connection Pool instead.

See Also: *Oracle Universal Connection Pool for JDBC Developer's Guide* for information

8.1.18 Changes for Oracle Call Interface

The following Oracle Call Interface (OCI) features are deprecated in Oracle Database 12c, and may be desupported in a future release:

- OCI deployment parameters in `sqlnet.ora` are deprecated. These include the following parameters:
 - OCI client result cache parameters: `OCI_RESULT_CACHE_MAX_SIZE`, `OCI_RESULT_CACHE_MAX_RSET_SIZE`, and `OCI_RESULT_CACHE_MAX_RSET_ROWS`

8.1.19 Deprecation of Oracle Restart

Oracle Restart is deprecated in Oracle Database 12c. Oracle Restart is currently restricted to manage single-instance Oracle databases and Oracle ASM instances only, and is subject to desupport in future releases. Oracle continues to provide Oracle ASM as part of the Oracle Grid Infrastructure installation for Standalone and Cluster deployments.

See Also:

- *Oracle Database Installation Guide* for your operating system
- *Oracle Database Administrator's Guide*

8.1.20 Changed Default for `resource_limit` Parameter

The Oracle `RESOURCE_LIMIT` parameter determines whether resource limits are enforced in database profiles. In this release, the `RESOURCE_LIMIT` parameter is set to `TRUE` by default. If Oracle resource limits are disabled, any defined profile limits will be ignored. This does not apply to password resources.

To check the `resource_limit` setting from SQL*Plus:

```
SQL> select value from v$parameter where name = 'resource_limit';
```

8.2 Oracle Business Intelligence and Data Warehousing Changes

Oracle Database 12c provides changes to business intelligence and data warehousing applications and features.

This section includes these topics:

- [Oracle Warehouse Builder \(OWB\) Not Installed with Oracle Database](#)
- [Desupport of Change Data Capture](#)
- [Changes to Upgrading Oracle Data Mining Models](#)
- [Desupport of OLAP Catalog \(AMD\)](#)

8.2.1 Oracle Warehouse Builder (OWB) Not Installed with Oracle Database

Starting with Oracle Database 12c, Oracle Warehouse Builder (OWB) is not installed as part of the software for Oracle Database. An installer for Oracle Warehouse Builder is available on Oracle Technology Network. OWB components that may exist from earlier releases are not upgraded as part of the Oracle Database upgrade process.

See Also: ["Requirement for Upgrading Databases that Use Oracle Warehouse Builder \(OWB\)"](#) on page 2-23

8.2.2 Desupport of Change Data Capture

Oracle Change Data Capture is not included in Oracle Database 12c and has been replaced with Oracle GoldenGate.

See Also: *Oracle GoldenGate documentation* for information

8.2.3 Desupported Features in Oracle Data Mining

The following Oracle Data Mining features are desupported in Oracle Database 12c:

- The Oracle Data Mining Java API is no longer available. The programmatic interfaces to Oracle Data Mining 12c consist of two PL/SQL packages, `DBMS_DATA_MINING` and `DBMS_DATA_MINING_TRANSFORM`, and a family of SQL language functions for scoring data.

See Also: *Oracle Data Mining User's Guide* for information about the Data Mining PL/SQL packages

- The Adaptive Bayes Network (ABN) algorithm is no longer available. The Decision Tree algorithm replaces ABN in Oracle Data Mining 12c.

See Also: *Oracle Data Mining Concepts* for information about the Decision Tree algorithm

8.2.4 Changes to Upgrading Oracle Data Mining Models

Starting with Oracle Database 12c, Oracle is desupporting the Data Mining Java API and the Adaptive Bayes Network (ABN) algorithm. Therefore, models created by the Oracle Data Mining Java API cannot be upgraded from Oracle Database release 11g to Oracle Database 12c. All other models and metadata are upgraded automatically during the upgrade from release 11g to Oracle Database 12c. ABN models can be upgraded, but you cannot use them in an Oracle Database 12c database. You must drop ABN models either before the upgrade or afterward. You can replace ABN models by building new classification models in the Oracle Database 12c database.

See Also: *Oracle Data Mining User's Guide* for information about the Data Mining API and data mining models

8.2.5 Desupport of OLAP Catalog (AMD)

Starting with Oracle Database 12c, the Common Warehouse Metamodel (CWM) standard is desupported for the OLAP catalog (AMD). This support was deprecated in Oracle Database 11g Release 2 (11.2). If your existing database has CWM metadata in the OLAP catalog and you upgrade to Oracle Database 12c, then the upgraded database has the AMD component. If the database you upgrade does not have the AMD component, then the upgraded Oracle Database 12c database also does not have the AMD component because new installations for Oracle Database 12c do not include AMD. If your database has the AMD component and you want to remove it, then run the `ORACLE_HOME/olap/admin/catnoamd.sql` script before or after upgrading.

Note: If the OLAP catalog exists in the database you are upgrading, you may see AMD OLAP Catalog OPTION OFF and invalid CWM OLAP objects. You can safely ignore the invalid OLAP objects as they are not needed.

See Also:

- [Oracle OLAP Java API Developer's Guide](#) for information about Java API metadata
- [Oracle OLAP User's Guide](#) for more information on the OLAP option of Oracle Database and online analytic processing

8.3 Oracle Database Security Changes

Oracle Database 12c includes changes to security features in Oracle Database Vault and Oracle Data Guard and other areas. This section contains these topics:

- [Changes to Security Auditing Features](#)
- [Deprecations for Oracle Label Security](#)
- [Deprecation of IGNORECASE and SEC_CASE_SENSITIVE_LOGON](#)
- [Deprecation of SQLNET.ALLOWED_LOGON_VERSION Parameter](#)
- [AUDIT_ADMIN and AUDIT_VIEWER Roles Note](#)
- [Deprecation of Windows NTS Authentication Using the NTLM Protocol](#)
- [Deprecation of Public Key Infrastructure for Transparent Data Encryption](#)
- [Desupported Cipher Suites for Secure Sockets Layer \(SSL\)](#)
- [Desupport of Database Rules Manager \(RUL\) and Expression Filter \(EXF\)](#)
- [Changes for Oracle Data Guard Broker](#)
- [Changes for Oracle Database Vault](#)
- [Changes for Oracle Database Semantic Technologies](#)

8.3.1 Changes to Security Auditing Features

The auditing functionality has been redesigned in Oracle Database 12c. When you create a new database with Oracle Database 12c, the full set of auditing enhancement features are automatically available. If you upgrade from an earlier release, then you are given the option of using some of the new audit features and the audit functionality from the release from which you upgraded. Oracle strongly recommends that you migrate to the full set of the latest audit features.

See Also:

- [Oracle Database Security Guide](#) for information about new auditing features and changes for security
- ["Migrate Your Upgraded Oracle Databases to Use Unified Auditing"](#) on page 4-14

8.3.2 Deprecations for Oracle Label Security

The following Oracle Label Security features are deprecated in Oracle Database 12c, and may be desupported in a future release:

- `LEAST_UBOUND`. Use `OLS_GREATEST_LBOUND` instead. See [Oracle Label Security Administrator's Guide](#) for information about finding greatest lower bound with `GREATEST_LBOUND`.

- LUBD. Use OLS_GLBD instead. See *Oracle Label Security Administrator's Guide* for information about OLS_GLBD.
- DOMINATES. See *Oracle Label Security Administrator's Guide* for information about the OLS_DOMINATES standalone function.
- DOM. See *Oracle Label Security Administrator's Guide* for information about OLS_STRICTLY_DOMINATES standalone function.
- STRICTLY_DOMINATES. See *Oracle Label Security Administrator's Guide* for information about OLS_STRICTLY_DOMINATES standalone function.
- S_DOM. See *Oracle Label Security Administrator's Guide* for information about OLS_STRICTLY_DOMINATES standalone function.
- DOMINATED_BY. See *Oracle Label Security Administrator's Guide* for information about OLS_DOMINATED_BY standalone function.
- DOM_BY. See *Oracle Label Security Administrator's Guide* for information about OLS_DOMINATED_BY standalone function.
- STRICTLY_DOMINATED_BY. See *Oracle Label Security Administrator's Guide* for information about OLS_STRICTLY_DOMINATED_BY standalone function.
- S_DOM_BY. See *Oracle Label Security Administrator's Guide* for information about OLS_STRICTLY_DOMINATED_BY standalone function.

8.3.3 Deprecated DBMS_NETWORK_ACL_ADMIN PL/SQL package Procedures

In Oracle Database 12c, the following procedures from the DBMS_NETWORK_ACL_ADMIN PL/SQL package are deprecated:

- CREATE_ACL
- ADD_PRIVILEGE
- DELETE_PRIVILEGE
- ASSIGN_ACL
- UNASSIGN_ACL
- DROP_ACL
- ASSIGN_WALLET_ACL
- UNASSIGN_WALLET_ACL
- CHECK_PRIVILEGE
- CHECK_PRIVILEGE_ACLID

See Also: *Oracle Database Security Guide* for information about managing fine-grained access in PL/SQL Packages and Types

8.3.4 Deprecation of IGNORECASE and SEC_CASE_SENSITIVE_LOGON

The IGNORECASE argument of ORAPWD and the SEC_CASE_SENSITIVE_LOGON system parameter are deprecated in Oracle Database 12c. By default, passwords in Oracle Database 12c are case sensitive.

See Also: *Oracle Database Security Guide* for more information about configuring authentication and password protection

8.3.5 Deprecation of SQLNET.ALLOWED_LOGON_VERSION Parameter

The `SQLNET.ALLOWED_LOGON_VERSION` parameter is deprecated in Oracle Database 12c. This parameter has been replaced with two new Oracle Net Services parameters:

- `SQLNET.ALLOWED_LOGON_VERSION_SERVER` (See *Oracle Database Net Services Reference* for information)
- `SQLNET.ALLOWED_LOGON_VERSION_CLIENT` (See *Oracle Database Net Services Reference* for information)

See Also:

- *Oracle Database Security Guide* for information about this deprecation
- ["Upgrading a System that Did Not Have SQLNET.ALLOWED_LOGON_VERSION Parameter Setting"](#) on page 8-15
- ["Check for the SQLNET.ALLOWED_LOGON_VERSION Parameter Behavior"](#) on page 4-7

8.3.5.1 Upgrading a System that Did Not Have SQLNET.ALLOWED_LOGON_VERSION Parameter Setting

If you are upgrading a system that did *not* have a `SQLNET.ALLOWED_LOGON_VERSION` parameter setting (that is, it was using the default 8), then you might need to set the value of the `SQLNET.ALLOWED_LOGON_VERSION_SERVER` to 8 in the upgraded Oracle Database 12c server to maintain compatibility with clients on earlier releases. Otherwise, if no setting for `SQLNET.ALLOWED_LOGON_VERSION_SERVER` (or the deprecated `SQLNET.ALLOWED_LOGON_VERSION`) parameter is made in the upgraded Oracle Database 12c server, then the *new* default value becomes 11 in the new Oracle Database 12c.

The effect of the new default value of 11 for `SQLNET.ALLOWED_LOGON_VERSION_SERVER` in Oracle Database 12c is that clients using Oracle Database release 10g and later can connect to the Oracle Database 12c server. Clients using releases earlier than Oracle Database release 11.2.0.3 that have not applied critical patch update CPUOct2012 or later patches must use the 10g password version.

Using a setting of 12 is most secure. However, this setting only permits Oracle Database 12c clients to connect.

See Also: *Oracle Database Readme* for the topic "Protection Against Password-Guessing Attacks" and *Oracle Database Net Services Reference* for information about `SQLNET.ALLOWED_LOGON_VERSION_SERVER`

8.3.6 AUDIT_ADMIN and AUDIT_VIEWER Roles Note

In Oracle Database 12c, two roles may exist in your database that affect upgrading: `AUDIT_ADMIN` and `AUDIT_VIEWER`.

The Pre-Upgrade Information Tool and DBUA perform a pre-upgrade check to make sure such users or roles do not exist in the database. Oracle recommends that you do not use these names in your databases. If these users or roles exist, then you should rename or drop them as appropriate before upgrading to Oracle Database 12c.

See Also: *Oracle Database Security Guide* for information on configuring privilege and role authorization for database security

8.3.7 Deprecation of Windows NTS Authentication Using the NTLM Protocol

Windows users can no longer authenticate using the NTS adaptor on Windows clients and servers that require the NT Lan Manager (NTLM) protocol. Because of security vulnerabilities, NTLM is deprecated as of Oracle Database 12c. Windows users can still use Kerberos. NTLM is still used for local user authentication, and in cases in which the database service runs as a local user.

A new client side `sqlnet.ora` boolean parameter `NO_NTLM` (defaulting to false) allows you to control when NTLM is used in NTS authentication. When you set `NO_NTLM` to true, this parameter value prevents NTLM from being used in Windows NTS authentication.

See Also: *Oracle Database Platform Guide for Microsoft Windows* for information on changes that affect the Windows platform in this release

8.3.8 Deprecation of Public Key Infrastructure for Transparent Data Encryption

Public Key Infrastructure (PKI) is deprecated for Transparent Data Encryption (TDE) in Oracle Database 12c. To configure TDE, use the `ADMINISTER KEY MANAGEMENT SQL` statement. Other implementations of PKI are not affected.

See Also: *Oracle Database Advanced Security Administrator's Guide* for information about configuring TDE

8.3.9 Desupported Cipher Suites for Secure Sockets Layer (SSL)

Oracle Advanced Security has desupported the following cipher suites in Oracle Database 12c:

- `SSL_DH_anon_WITH_DES_CBC_SHA`
- `SSL_RSA_EXPORT_WITH_DES40_CBC_SHA`
- `SSL_RSA_EXPORT_WITH_RC4_40_MD5`
- `SSL_RSA_WITH_DES_CBC_SHA`

See Also: *Oracle Database Security Guide* for information about the supported SSL cipher suites

8.3.10 Desupport of Database Rules Manager (RUL) and Expression Filter (EXF)

Starting with Oracle Database 12c, the Expression Filter (EXF) and Database Rules Manager (RUL) features are desupported. If you are using Rules Manager, Oracle recommends that you consider migrating to Oracle Business Rules, a component of Oracle Fusion Middleware. The Continuous Query Notification feature of Oracle Database replaces Expression Filter.

This script is executed by the upgrade process. If you want to remove these components before upgrading, then you can run the `catnoexf.sql` script before the upgrade. The `catnoexf.sql` script is located under `ORACLE_HOME/rdbms/admin/`.

See Also: Support note 1244535.1 *Rules Manager and Expression Filter Features of Oracle Database* on My Oracle Support at <http://support.oracle.com/>

8.3.11 Changes for Oracle Data Guard Broker

As part of Oracle Data Guard Broker's support for Separation of Duty features, the following broker properties are deprecated in Oracle Database 12c with no replacements:

- `LsbyASkipCfgPr`
- `LsbyASkipErrorCfgPr`
- `LsbyASkipTxnCfgPr`
- `LsbyDSkipCfgPr`
- `LsbyDSkipErrorCfgPr`
- `LsbyDSkipTxnCfgPr`
- `LsbySkipTable`
- `LsbySkipTxnTable`

8.3.12 Changes for Oracle Data Pump Export Utility

The `XML_CLOBS` option of the Oracle Data Pump Export `DATA_OPTIONS` parameter is deprecated.

See Also: *Oracle Database Utilities* for information about the Export `DATA_OPTIONS` parameter

8.3.13 Changes for Oracle Database Vault

Starting with Oracle Database 12c, the Database Vault Configuration Assistant (DVCA) and Database Vault Administrator (DVA) are being deprecated. Use Oracle Enterprise Manager Cloud Control for Oracle Database Vault configuration and administration.

Additional Oracle Database Vault features are being deprecated in this release as follows:

- [Deprecated Default Rule Sets](#)
- [Deprecated Default Realms](#)
- [Deprecated API](#)

8.3.13.1 Deprecated Default Rule Sets

The following rule sets have been deprecated for this release:

- Allow Oracle Data Pump Operation rule set
- Allow Scheduler Job rule set

See Also: *Oracle Database Vault Administrator's Guide* for information on default rule sets

8.3.13.2 Deprecated Default Realms

The Oracle Data Dictionary realm and Oracle Enterprise Manager realm are deprecated in this release. The objects formerly protected by the Oracle Data Dictionary realm have been migrated to new realms.

See Also: *Oracle Database Vault Administrator's Guide* for information about new realms and default realms

8.3.13.3 Deprecated API

The `DVSYS.DBMS_MACADM.SYNC_RULES` procedure has been deprecated because its functionality has been built into the rule creation functionality.

See Also: *Oracle Database Vault Administrator's Guide* for information on rule creation

8.3.14 Changes for Oracle Database Semantic Technologies

The following features are deprecated for Oracle Database Semantic Technologies in Oracle Database 12c:

- [VPD Support in Oracle Database Semantic Technologies](#)
- [Version-Enabled Models Support In Oracle Database Semantic Technologies](#)

8.3.14.1 VPD Support in Oracle Database Semantic Technologies

Transition existing Semantic Technologies applications that depend on Virtual Private Database (VPD) to use Oracle Label Security (OLS) instead.

See Also: *Oracle Database Semantic Technologies Developer's Guide* for information about fine-grained access control for RDF data

8.3.14.2 Version-Enabled Models Support In Oracle Database Semantic Technologies

The specific alternative to using Workspace Manager with semantic data depends on the purpose of the application. For information about use cases and recommended alternatives, see My Oracle Support Note 1468273.1.

See Also: *Oracle Database Semantic Technologies Developer's Guide* for information about Workspace Manager support for RDF data

8.4 Oracle Globalization Support Changes

Oracle Database 12c provides changes to how Oracle supports globalization.

This section contains the following topic:

- [Desupport of CSSCAN and CSALTER](#)

8.4.1 Desupport of CSSCAN and CSALTER

With the introduction of Oracle Database Migration Assistant for Unicode (DMU), Oracle is proceeding with the desupport of the legacy database tools, CSSCAN and CSALTER. DMU provides a complete end-to-end Unicode migration solution for database administrators. Starting with Oracle Database 12c, DMU is included with Oracle Database, and the CSSCAN and CSALTER tools are no longer included nor supported.

See Also:

- Support note "CSSCAN and CSALTER To Be Desupported After DB 11.2" (Doc ID 1418321.1) on My Oracle Support at <http://support.oracle.com/>
- *Oracle Database Migration Assistant for Unicode Guide* for information about the database character set migration process

8.5 Oracle Multimedia Changes

In Oracle Database 12c, ORDImage support for DICOM is deprecated.

See Also: *Oracle Multimedia Reference* for information about ORDImage

8.6 Oracle Net Services Changes

In this release, Oracle Net Services has deprecated or no longer supports some features, parameters, and commands. This section contains the following topics:

- [Desupport of Oracle Net Connection Pooling](#)
- [Desupport of Oracle Names](#)
- [Desupport of Oracle Net Listener Password](#)
- [Desupport of SQLNET.KERBEROS5_CONF_MIT Parameter for Oracle Net Services](#)
- [Desupport of Oracle Names Control Utility for Oracle Net Services](#)
- [Deprecated NT LAN Manager \(NTLM\) Protocol for Oracle Net Services](#)

8.6.1 Desupport of Oracle Net Connection Pooling

In Oracle Database 12c, Oracle Net connection pooling is no longer supported. Oracle Net connection pooling was deprecated in Oracle Database 11g. This includes the DISPATCHERS attributes TICKS, SESSIONS, and CONNECTIONS.

See Also: Support note 1469466.1 about Oracle Net Connection Pooling on My Oracle Support at <http://support.oracle.com/>

8.6.2 Desupport of Oracle Names

Oracle Names has not been supported as a naming method since Oracle Database 11g. You must migrate to directory naming.

See Also: *Oracle Database Net Services Administrator's Guide* for additional information about migrating to directory naming

8.6.3 Desupport of Oracle Net Listener Password

In Oracle Database 12c, the Oracle Net Listener password feature is no longer supported. This does not cause a loss of security because authentication is enforced through local operating system authentication.

See Also: *Oracle Database Net Services Reference* for more information about Oracle Net Listener security

8.6.4 Desupport of SQLNET.KERBEROS5_CONF_MIT Parameter for Oracle Net Services

The SQLNET.KERBEROS5_CONF_MIT networking parameter is no longer needed (nor supported) in sqlnet.ora. This parameter is set to FALSE by default, which has no affect on the Kerberos configuration. Formerly when set to TRUE, this parameter specified that the configuration file should be parsed in a format as specified by MIT

Kerberos 5. Starting with Oracle Database 12c, only this format of Kerberos configuration is supported. Thus, there is no longer a need for this parameter.

See Also: *Oracle Database Net Services Reference* for information on Kerberos parameters for the `sqlnet.ora` file

8.6.5 Desupport of Oracle Names Control Utility for Oracle Net Services

The Oracle Names Control Utility is desupported and has not been available starting with Oracle Database 10g. This includes all the related control utility commands. Oracle Database clients cannot use a Names Server to resolve connect strings. Migrate your applications to Oracle Internet Directory with LDAP directory naming.

See Also: *Oracle Database Net Services Reference* for information about configuring the directory naming method

8.6.6 Deprecated NT LAN Manager (NTLM) Protocol for Oracle Net Services

The NT LAN Manager (NTLM) protocol for domain authentication is deprecated in the Oracle Windows adapter. Only Kerberos authentication is used for the NTS adapter.

Tip: *Oracle Database Net Services Reference* for information on Kerberos parameters for the `sqlnet.ora` file

8.7 Oracle XML Database Changes

In addition to being installed with the Oracle Database software, Oracle XML Database has changed features in Oracle Database 12c.

- [Oracle XML DB is Mandatory and Cannot Be Uninstalled](#)
- [Deprecated Features for Oracle XML Database](#)
- [Desupport of CTXXPATH in Oracle Text and Oracle XML DB](#)

8.7.1 Oracle XML DB is Mandatory and Cannot Be Uninstalled

Starting with Oracle Database 12c, Oracle XML DB is a mandatory component of Oracle Database. You cannot uninstall Oracle XML DB, and there is no option to exclude it when you create an Oracle database. Oracle XML DB is automatically installed or upgraded when you upgrade an existing database to Oracle Database 12c.

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

8.7.2 Deprecated Features for Oracle XML Database

The following features are deprecated in this release, and may be desupported in a future release. See *Oracle XML DB Developer's Guide* for a complete list and more information.

- CLOB storage of `XMLType`, also known as unstructured storage, is deprecated. Use binary XML storage of `XMLType` instead.

If you need to preserve insignificant whitespace then store two copies of your original XML document: one as an `XMLType` instance for database use and XML processing, the other as a CLOB instance to provide document fidelity.

- Creation of an `XMLIndex` index over an XML fragment stored as a `CLOB` instance embedded in object-relational `XMLType` data is deprecated. If you need to index the data in such a fragment then store the document using binary XML storage, not object-relational storage.
- The following PL/SQL subprograms in package `DBMS_XMLSCHEMA` are deprecated:

- `generateSchema`
- `generateSchemas`

There are no replacements for these constructs and there is no workaround for this change.

- PL/SQL package `DBMS_XDB_CONFIG` is new. All Oracle XML DB configuration functions, procedures, and constants have been moved from package `DBMS_XDB` to `DBMS_XDB_CONFIG`. They are *deprecated* for package `DBMS_XDB`. Oracle recommends that you use them in package `DBMS_XDB_CONFIG` instead.

These are the subprograms that are deprecated in package `DBMS_XDB`:

- `ADDHTTPEXPIREMAPPING`
- `ADDMIMEMAPPING`
- `ADDSCHEMALOCMAPPING`
- `ADDSERVLET`
- `ADDSERVLETMAPPING`
- `ADDSERVLETSECROLE`
- `ADDXMLEXTENSION`
- `CFG_GET`
- `CFG_REFRESH`
- `CFG_UPDATE`
- `DELETEHTTPEXPIREMAPPING`
- `DELETMIMEMAPPING`
- `DELETESCHEMALOCMAPPING`
- `DELETESERVLET`
- `DELETESERVLETMAPPING`
- `DELETESERVLETSECROLE`
- `DELETXMLEXTENSION`
- `GETFTPSPORT`
- `GETHTTPSPORT`
- `GETLISTENERENDPOINT`
- `SETFTPSPORT`
- `SETHTTPSPORT`
- `SETLISTENERENDPOINT`
- `SETLISTENERLOCALACCESS`

These are the constants that are deprecated in package `DBMS_XDB`:

- XDB_ENDPOINT_HTTP
- XDB_ENDPOINT_HTTP2
- XDB_PROTOCOL_TCP
- XDB_PROTOCOL_TCPS

See Also: *Oracle Database PL/SQL Packages and Types Reference*, Chapter "DBMS_XDB_CONFIG"

- All Oracle SQL functions for updating XML data are deprecated. Oracle recommends that you use XQuery Update instead. These are the deprecated XML updating functions:

- updateXML
- insertChildXML
- insertChildXMLbefore
- insertChildXMLafter
- insertXMLbefore
- insertXMLafter
- appendChildXML
- deleteXML

See Also: *Oracle XML DB Developer's Guide* for the appendix that describes Oracle SQL functions for updating XML data that are deprecated and recommendations for replacements

- Oracle SQL function `sys_xmlgen` is deprecated. Oracle recommends that you use the SQL/XML generation functions instead.

See Also: *Oracle XML DB Developer's Guide* for information about SQL functions that you can use to construct XML data

- The following Oracle XQuery functions are deprecated. Use the corresponding standard XQuery functions instead, that is, the functions with the same names but with namespace prefix `fn`.

- `ora:matches` – use `fn:matches` instead
- `ora:replace` – use `fn:replace` instead

- The following Oracle constructs that provide support for XML translations are deprecated.

- PL/SQL package `DBMS_XMLTRANSLATIONS`
- Oracle XPath function `ora:translate`
- XML Schema annotations `xdb:maxOccurs`, `xdb:srcLang`, and `xdb:translate`

There are no replacements for these constructs and there is no workaround for this change.

See Also: *Oracle XML DB Developer's Guide* for more information

- The following XML Schema annotations are deprecated:

- `xdb:defaultTableSchema`
- `xdb:maintainOrder`
- `xdb:mapUnboundedStringToLob`
- `xdb:maxOccurs`
- `xdb:SQLCollSchema`
- `xdb:SQLSchema`
- `xdb:srclang`
- `xdb:storeVarrayAsTable`
- `xdb:translate`

There are no replacements for these constructs, and there is no workaround for this change.

- The value `xml_clobs` for export parameter `data_options` is *deprecated* starting with Oracle Database 12c.

See Also: *Oracle XML DB Developer's Guide* for information about exporting and importing XMLType Tables

8.8 Oracle Text Changes

Oracle Database 12c introduces changes for Oracle Text.

- [Desupport of CTXXPATH in Oracle Text and Oracle XML DB](#)
- [Desupport of ALTER INDEX OPTIMIZE for Text Indexes](#)
- [Desupport of SYNC \[MEMORY memsize\] for Text Indexes](#)

8.8.1 Desupport of CTXXPATH in Oracle Text and Oracle XML DB

`CTXSYS.CTXXPATH` index is desupported as of Oracle Database 12c. This does not affect `CTXCAT`. Use `XMLIndex` indexes instead.

See Also: *Oracle XML DB Developer's Guide* for more information about `XMLIndex`

8.8.2 Desupport of ALTER INDEX OPTIMIZE for Text Indexes

The `ALTER INDEX OPTIMIZE [token index_token | fast | full [maxtime (time | unlimited)]]` operation is not supported for Oracle Database 12c. To optimize your index, use `CTX_DDL.OPTIMIZE_INDEX`.

See Also: *Oracle Text Reference* for information about `OPTIMIZE_INDEX`

8.8.3 Desupport of SYNC [MEMORY memsize] for Text Indexes

The `SYNC [MEMORY memsize]` operation is not supported for Oracle Database 12c. To synchronize your index, use `CTX_DDL.SYNC_INDEX`.

See Also: *Oracle Text Reference* for information about `SYNC_INDEX`

Changes for Earlier Releases of Oracle Database

There are important changes in behavior between Oracle Database 10g Release 2 (10.2), Oracle Database 11g Release 1 (11.1), and Oracle Database 11g Release 2 (11.2). Behavior changes typically require a database administrator (DBA) to make informed decisions to minimize the risks that may be introduced by the changes after upgrading Oracle Database.

This appendix contains the following topics:

- [Behavior Changes in Oracle Database 11g Release 2 \(11.2\)](#)
- [Behavior Changes in Oracle Database 11g Release 1 \(11.1\)](#)
- [Behavior Changes in Oracle Database 10g Release 2 \(10.2\)](#)

See Also:

- *Oracle Database New Features Guide* for a complete list of all new features introduced in this release
- The "What's New in Oracle Database Reference" section of *Oracle Database Reference* for a list of new initialization parameters, new static data dictionary views, and new dynamic performance views
- *Oracle Database Installation Guide* for your operating system

Note:

- [Appendix A, "Changes for Earlier Releases of Oracle Database"](#) does not describe all changed behavior and new features in each release of Oracle Database. For complete information about new features, refer to *Oracle Database New Features Guide*.
 - 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.
-
-

A.1 Behavior Changes in Oracle Database 11g Release 2 (11.2)

Oracle Database 11g Release 2 (11.2) introduces new features and changes that affect compatibility and interoperability. The following changes may affect the upgrade process.

- [Deprecation of SNMP Support in Oracle Net Listener](#)
- [Changes to PL/SQL Procedures](#)
- [JOB_QUEUE_PROCESSES Parameter and Scheduling Jobs](#)
- [Deprecated XML DB Constructs](#)
- [Cursor_sharing=similar Deprecated in Oracle Database](#)
- [Planned Desupport of Change Data Capture](#)
- [Deprecation of the Data Mining Java API](#)
- [Initialization Parameters Deprecated in Oracle Database 11g Release 2 \(11.2\)](#)
- [Initialization Parameters Desupported in Oracle Database 11g Release 2 \(11.2\)](#)
- [Static Data Dictionary Views Deprecated in Oracle Database 11g Release 2 \(11.2\)](#)
- [Dynamic Performance Views Deprecated in Oracle Database 11g Release 2 \(11.2\)](#)
- [Deprecated Features in Oracle Database 11g Release 2 \(11.2\)](#)
- [Changes to LOG_ARCHIVE_DEST_n Parameters](#)

A.1.1 Deprecation of SNMP Support in Oracle Net Listener

Oracle has deprecated SNMP support in Oracle Net Listener in release 11.2.0.3 of Oracle Database. Oracle recommends not using SNMP in new implementations.

See Also: My Oracle Support Note 1341834.1 "Planned end of support for SNMP in Oracle Net Listener" at <http://support.oracle.com>

A.1.2 Changes to PL/SQL Procedures

Some PL/SQL procedures have been moved from package DBMS_XDB to package DBMS_XDB_ADMIN in release 11.2.0.3 of Oracle Database.

The procedures moved to DBMS_XDB_ADMIN include:

- `moveXDB_tablespace`
- `rebuildHierarchicalIndex`

See Also: *Oracle Database PL/SQL Packages and Types Reference*

A.1.3 JOB_QUEUE_PROCESSES Parameter and Scheduling Jobs

Starting with Oracle Database 11g Release 2 (11.2), setting `JOB_QUEUE_PROCESSES` to 0 causes both `DBMS_SCHEDULER` and `DBMS_JOB` jobs to not run. Previously, setting `JOB_QUEUE_PROCESSES` to 0 caused `DBMS_JOB` jobs to not run, but `DBMS_SCHEDULER` jobs were unaffected and would still run. The default value is 1000.

Oracle Database overrides the job queue setting to disable scheduler jobs during upgrade mode. Therefore, you can ignore this setting when upgrading Oracle Database. When the database is opened under `UPGRADE` mode, Oracle internally enforces that no jobs can be run (under `dbms_job` or `scheduler`). It does not matter what `JOB_QUEUE_PROCESSES` is set to in `UPGRADE` mode.

See Also: *Oracle Database Reference* for more information on this parameter

A.1.4 Deprecated XML DB Constructs

The following XML DB constructs have been deprecated in release 11.2.0.3:

- PL/SQL procedure `DBMS_XDB_ADMIN.createRepositoryXMLIndex`
- PL/SQL procedure `DBMS_XDB_ADMIN.XMLIndexAddPath`
- PL/SQL procedure `DBMS_XDB_ADMIN.XMLIndexRemovePath`
- PL/SQL procedure `DBMS_XDB_ADMIN.dropRepositoryXMLIndex`
- XML schema annotation (attribute) `csx:encodingType`
- `XMLIndex` index on CLOB portions of hybrid `XMLType` storage (this is CLOB data that is embedded within object-relational storage)

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

A.1.5 `cursor_sharing=similar` Deprecated in Oracle Database

The `cursor_sharing=similar` parameter is deprecated in Oracle Database release 11.2.0.3. Use adaptive cursor sharing instead.

See Also: *Oracle Database SQL Tuning Guide* for information about adaptive cursor sharing

A.1.6 Planned Desupport of Change Data Capture

Oracle Change Data Capture is being replaced with Oracle GoldenGate. Therefore, Oracle strongly recommends that you use Oracle GoldenGate for new applications.

For Oracle Database 11g Release 2 (11.2), Change Data Capture continues to function as in earlier releases. If you are currently using Change Data Capture, then you can continue to do so for the foreseeable future. However, Change Data Capture is not being further enhanced.

A.1.7 Deprecation of the Data Mining Java API

Starting with release 11.2.0.3 of Oracle Database, the Data Mining Java API is deprecated. Oracle Data Mining now supports a new release of Oracle Data Miner. The earlier release, Oracle Data Miner Classic, is still available for download on Oracle Technology Network (OTN), but it is no longer under active development.

A.1.8 Initialization Parameters Deprecated in Oracle Database 11g Release 2 (11.2)

A deprecated parameter behaves the same way as a regular parameter, except that a warning message is displayed at instance startup if a deprecated parameter is specified in the parameter file. In addition, all deprecated parameters are logged to the alert log at instance startup.

To get a list of all initialization parameters that are specified as deprecated for the current database, issue the following SQL statement:

```
SQL> SELECT name FROM v$parameter
        WHERE isdeprecated = 'TRUE';
```

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

```
ACTIVE_INSTANCE_COUNT
PARALLEL_IO_CAP_ENABLED
```

A.1.9 Initialization Parameters Desupported in Oracle Database 11g Release 2 (11.2)

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

Note: An attempt to start a database using one or more desupported initialization parameters succeeds, but a warning is returned and recorded in the alert log.

DRS_START
GC_FILES_TO_LOCKS
MAX_COMMIT_PROPAGATION_DELAY
PLSQL_NATIVE_LIBRARY_DIR
PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT
SQL_VERSION

A.1.10 Static Data Dictionary Views Deprecated in Oracle Database 11g Release 2 (11.2)

These static data dictionary views were deprecated in Oracle Database 11g Release 2 (11.2):

ALL_STREAMS_STMTS (replaced by DBA_STREAMS_STMTS)
ALL_STREAMS_STMT_HANDLERS (replaced by DBA_STREAMS_STMT_HANDLERS)
DBA_COMPARISON_SCAN_SUMMARY (replaced by DBA_COMPARISON_SCAN)
USER_COMPARISON_SCAN_SUMMARY (replaced by USER_COMPARISON_SCAN)

A.1.11 Dynamic Performance Views Deprecated in Oracle Database 11g Release 2 (11.2)

A dynamic performance view was deprecated in Oracle Database 11g Release 2 (11.2).

V\$FLASH_RECOVERY_AREA_USAGE (replaced by V\$RECOVERY_AREA_USAGE)

A.1.12 Deprecated Features in Oracle Database 11g Release 2 (11.2)

Oracle Database features have been deprecated in Oracle Database 11g Release 2 (11.2). The deprecated features are supported in this release for backward compatibility. However, Oracle recommends that you migrate away from these deprecated features.

- [Dictionary-managed Tablespaces](#)
- [MAX_JOB_SLAVE_PROCESSES](#)

Dictionary-managed Tablespaces

Oracle recommends that you create locally managed tablespaces. Locally managed tablespaces are much more efficiently managed than dictionary-managed tablespaces.

MAX_JOB_SLAVE_PROCESSES

MAX_JOB_SLAVE_PROCESSES has been deprecated. Use JOB_QUEUE_PROCESSES instead.

A.1.13 Changes to LOG_ARCHIVE_DEST_n Parameters

Starting with Oracle Database 11g Release 2 (11.2), the number of supported destinations in the LOG_ARCHIVE_DEST_n and the LOG_ARCHIVE_DEST_STATE_n parameters have been increased from 10 to 31. Destinations LOG_ARCHIVE_DEST_11

through LOG_ARCHIVE_DEST_31 do not support the SYNC, ARCH, LOCATION, MANDATORY, ALTERNATE, or DEPENDENCY attributes, and cannot be specified as the target of the ALTERNATE or DEPENDENCY attributes.

LOG_ARCHIVE_DEST_11 through LOG_ARCHIVE_DEST_31 can only be used when the COMPATIBLE initialization parameter is set to 11.2.0 or higher.

A.2 Behavior Changes in Oracle Database 11g Release 1 (11.1)

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

- [Initialization Parameters Deprecated in Oracle Database 11g Release 1 \(11.1\)](#)
- [Initialization Parameters Desupported in Oracle Database 11g Release 1 \(11.1\)](#)
- [Static Data Dictionary Views with Dropped Columns in Oracle Database 11g Release 1 \(11.1\)](#)
- [Deprecated Features in Oracle Database 11g Release 1 \(11.1\)](#)
- [Automatic Maintenance Tasks Management](#)
- [New SYSASM Privilege and OSASM Group for Oracle ASM Administration](#)
- [ASM Disk Group Compatibility](#)
- [COMPUTE STATISTICS and ESTIMATE STATISTICS Clauses](#)
- [Oracle Data Mining Models and the DMSYS Schema Objects](#)
- [Oracle Data Mining Scoring Engine](#)
- [SQL Plan Management and Control of SQL Plan Baselines](#)
- [Binary XML Support for Oracle XML Database](#)
- [PL/SQL Native Compilation and Access Control for Network Utility Packages](#)
- [PL/SQL Control Parameters](#)
- [Change in WebDAV ACL Evaluation Rules in Oracle XML DB](#)
- [Summary Management and SQL Access Advisor](#)
- [Standard Edition Starter Database](#)
- [Core Dump Location](#)
- [New Default Value for UNDO_MANAGEMENT](#)
- [LOG_ARCHIVE_DEST_n Parameters](#)
- [SHARED_POOL_SIZE Parameter](#)
- [JOB_QUEUE_PROCESSES Parameter New Default Setting](#)
- [Automatic Diagnostic Repository](#)

A.2.1 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).

To get a list of all deprecated initialization parameters, issue the following SQL statement:

```
SQL> SELECT name FROM v$parameter
      WHERE isdeprecated = 'TRUE';
```

A deprecated parameter behaves the same way as a regular parameter, except that a warning message is displayed at instance startup if a deprecated parameter is specified in the parameter file. In addition, all deprecated parameters are logged to the alert log at instance startup.

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

A.2.2 Initialization Parameters Desupported in Oracle Database 11g Release 1 (11.1)

These initialization parameters were desupported in Oracle Database 11g Release 1 (11.1).

```
DDL_WAIT_FOR_LOCKS
LOGMNR_MAX_PERSISTENT_SESSIONS
PLSQL_COMPILER_FLAGS
```

Note: An attempt to start a database using one or more desupported initialization parameters succeeds, but a warning is returned and recorded in the alert log.

A.2.3 Static Data Dictionary Views with Dropped Columns in Oracle Database 11g Release 1 (11.1)

These 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 |

A.2.4 Deprecated Features in Oracle Database 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. However, Oracle recommends that you migrate away from these deprecated features.

- Oracle Ultra Search
 - Oracle recommends that you use Oracle Secure Enterprise Search.
- Java Development Kit (JDK) 1.4
 - Oracle recommends that you use JDK 5.0; however, JDK 1.5 is also fully supported.

- CTXXPATH index
Oracle recommends that you use XMLIndex instead.

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

A.2.5 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 on

All other Automatic Maintenance Tasks Management clients are enabled by default.

Although Automatic Maintenance Tasks Management is automatically enabled when upgrading to Oracle Database 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

A.2.6 New SYSASM Privilege and OSASM Group for Oracle ASM Administration

Oracle Database 11g Release 1 (11.1) introduces a new `SYSASM` privilege that is specifically intended for performing Oracle 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 appear in the Oracle 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 group that is used exclusively for Oracle ASM. Members of the `OSASM` group can connect `AS SYSASM` using operating system authentication and have full access to Oracle ASM.

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

A.2.7 ASM Disk Group Compatibility

Starting 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 Automatic Storage Management Administrator's Guide* for more information on ASM disk group compatibility

A.2.8 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 desupported. Oracle Database 11g Release 1 (11.1) automatically collects statistics during index creation and rebuild. These clauses are no longer supported.

A.2.9 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 must 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 for 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 plus 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.

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 are required to drop the new mining models before downgrading.

A.2.10 Oracle Data Mining Scoring Engine

Starting with Oracle Database 11g Release 1 (11.1), the Oracle Data Mining Scoring Engine can no longer be installed.

See Also: *Oracle Data Mining User's Guide*

A.2.11 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 can 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 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. 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 SQL 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

A.2.12 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.

A.2.12.1 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 XML DB 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).

A.2.13 PL/SQL Native Compilation and Access Control for Network Utility Packages

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

- [PL/SQL Native Compilation](#)
- [Access Control for Network Utility Packages](#)

A.2.13.1 PL/SQL Native Compilation

Starting with 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 system 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 must be concerned with PL/SQL Native Compilation.

A.2.13.2 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 run time 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 installed on the system, then it is installed during the upgrade procedure.

See Also: ["Configure Access Control Lists \(ACLs\) to External Network Services"](#) on page 4-6

A.2.14 PL/SQL Control Parameters

The behavior of some Oracle parameters that control the behavior of PL/SQL have changed as of 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 desupported. It has no effect any longer and draws an error message that an illegal option is being set.
- `PLSQL_V2_COMPATIBILITY` is deprecated.

A.2.15 Change in WebDAV ACL Evaluation Rules in Oracle XML DB

Oracle XML DB uses a security mechanism that is based on access-control lists (ACLs) to restrict access to any Oracle XML DB resource. An ACL is a list of access-control entries (ACEs) that determine which users, roles, and groups have access to a given resource.

There have been changes to the treatment of WebDAV ACL entries. Before Oracle Database 11g Release 1 (11.1), a `<deny>` entry always trumped any `<allow>` entry in a given ACL. Starting 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

A.2.16 Summary Management and SQL Access Advisor

The `DBMS_OLAP` package is deprecated and has been replaced by the SQL Access Advisor.

A.2.17 SQL Access Advisor Tasks

Due to internal structural changes to the SQL Access Advisor repository, a database upgrade resets all existing SQL Access Advisor tasks to their initial state. This effectively deletes all recommendation information for tasks that have successfully completed before the upgrade.

After upgrade, the recommendation information can be restored by re-executing the existing SQL Access Advisor tasks.

A.2.18 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 are some invalid objects in the associated component schemas. The Database Upgrade Assistant (DBUA) shows unsuccessful upgrades for these components.

A.2.19 Core Dump Location

On UNIX systems, when an application program crashes due to an unhandled signal, such as a segmentation fault, a core dump file is usually generated. The system default file name for this file is `core`, and it is located in the directory in which the application is currently running.

Starting with Oracle Database 11g Release 1 (11.1), applications using the Oracle Call Interface (OCI) can create a subdirectory named `core_process_id`, where `process_id` is the UNIX ID of the process that crashed. The `core` file is then placed in that subdirectory instead of the location where the application is running.

In `sqlnet.ora`, setting `DIAG_SIGHANDLER_ENABLED = TRUE` also puts the generated `core` file in the directory named `core_process_id`.

A.2.20 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.

The `UNDO_MANAGEMENT` and `ROLLBACK_SEGMENTS` initialization parameters have changed from basic initialization parameters to non-basic initialization parameters. Most databases must have only basic parameters set to run properly and efficiently.

See Also: UNDO_MANAGEMENT in *Oracle Database Reference*

A.2.21 LOG_ARCHIVE_DEST_n Parameters

You can use the LOG_ARCHIVE_DEST_n parameter 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.

A.2.22 SHARED_POOL_SIZE Parameter

Migration utilities for this release recommend new values for SHARED_POOL_SIZE based on the value of internal SGA overheads in the pre-upgrade environment, which you can determine by running the following query before upgrading to Oracle Database 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.

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.

A.2.23 JOB_QUEUE_PROCESSES Parameter New Default Setting

Beginning with Oracle Database 11g Release 1 (11.1), the JOB_QUEUE_PROCESSES parameter is changed from a basic to a non-basic initialization parameter. Most databases must only have basic parameters set to run properly and efficiently. The default value is also changed from 0 to 1000.

Oracle Database overrides the job queue setting to disable scheduler jobs during upgrade mode. Therefore, you can ignore this setting when upgrading Oracle Database. When the database is opened under UPGRADE mode, Oracle internally

enforces that no jobs can be run (under `dbms_job` or scheduler). It does not matter what `JOB_QUEUE_PROCESSES` is set to in UPGRADE mode.

See Also: *Oracle Database Reference* for more information on this parameter

A.2.24 Automatic Diagnostic Repository

The locations of alert logs and trace files are no longer set by the initialization parameters `BACKGROUND_DUMP_DEST` and `USER_DUMP_DEST`. They are now kept in the Automatic Diagnostic Repository (ADR), whose location is set by the initialization parameter `DIAGNOSTIC_DEST`.

See Also: *Oracle Database Administrator's Guide* for more information on the management of diagnostic information

A.3 Behavior Changes in Oracle Database 10g Release 2 (10.2)

Changes that affect compatibility and interoperability, and changes to roles and parameters were introduced in Oracle Database 10g Release 2 (10.2). If you are upgrading to Oracle Database 11g Release 1 (11.1) from 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 changes.

- [Initialization Parameters Deprecated in Oracle Database 10g Release 2 \(10.2\)](#)
- [Initialization Parameters Desupported in Oracle Database 10g Release 2 \(10.2\)](#)
- [Static Data Dictionary Views with Dropped Columns in Oracle Database 10g Release 2 \(10.2\)](#)
- [SQL](#)
- [CONNECT Role](#)
- [Time Zone Files](#)
- [New Limit for FAILED_LOGIN_ATTEMPTS](#)

A.3.1 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). To get a list of all deprecated initialization parameters, issue the following SQL statement:

```
SQL> SELECT name FROM v$parameter
        WHERE isdeprecated = 'TRUE';
```

A deprecated parameter behaves the same way as a regular parameter, except that a warning message is displayed at instance startup if a deprecated parameter is specified in the parameter file. In addition, all deprecated parameters are logged to the alert log at instance startup:

```
LOGMNR_MAX_PERSISTENT_SESSIONS
MAX_COMMIT_PROPAGATION_DELAY
REMOTE_ARCHIVE_ENABLE
SERIAL_REUSE
SQL_TRACE
```

A.3.2 Initialization Parameters Desupported in Oracle Database 10g Release 2 (10.2)

The following initialization parameters were desupported in Oracle Database 10g Release 2 (10.2).

Note: An attempt to start a database using one or more desupported initialization parameters succeeds, but a warning is returned and recorded in the alert log.

ENQUEUE_RESOURCES

A.3.3 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 |

A.3.4 SQL

The behavior of date formats has changed when used with XML functions. The XML Schema standard specifies that dates and timestamps in XML data be in standard formats. Before Oracle Database 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

A.3.5 CONNECT Role

After upgrading from a release earlier than 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.

A.3.6 Time Zone Files

The time zone files that are supplied with Oracle Database 10g Release 2 (10.2) have been updated from version 4 to version 8 to reflect changes in transition rules for some time zone regions. The changes might affect existing data of TIMESTAMP WITH TIME ZONE data type.

A.3.7 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. Before Oracle Database 10g Release 2 (10.2), the default was UNLIMITED.

Numerics

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

A

access control lists (ACLs)
 changed behavior in 11g release 1 (11.1), A-11
 granting access to network utility packages, 2-30
access control to network utility packages, A-10
adding ACLs for network utility packages, 2-30
ALTER DATABASE RECOVER TO LOGICAL
 STANDBY statement, 2-18
application code
 not changing after upgrade, 5-6
applications
 compatibility, 5-1
 linked and upgrading, 5-4
 linking with newer libraries, 5-5
 running against older server, 5-4
 upgrading, 5-1
 client/server configurations, 5-2
 compatibility rules, 5-3
 options, 5-5
 relinking rules, 5-3
apxrelod.sql file
 reloading after downgrade, 6-7
ASM. *See* Oracle Automatic Storage Management
ASM_PREFERRED_READ_FAILURE_
 GROUPS, 4-21
attributes
 xdb:defaultTableSchema (deprecated), 8-23
 xdb:maintainOrder (deprecated), 8-23
 xdb:mapUnboundedStringToLob
 (deprecated), 8-23
 xdb:maxOccurs (deprecated), 8-23
 xdb:SQLCollSchema (deprecated), 8-23
 xdb:SQLSchema (deprecated), 8-23
 xdb:srclang (deprecated), 8-23
 xdb:storeVarrayAsTable (deprecated), 8-23
 xdb:translate (deprecated), 8-23
auditing
 unified auditing migration
 about, 4-14
 audit options, 4-14
 documentation references for non-unified

 auditing, 4-18
 managing earlier audit records after
 migration, 4-17
 procedure, 4-15
 removing unified auditing, 4-17
Automatic Diagnostic Repository (ADR), 1-4
automatic maintenance tasks management
 AutoTask, A-7
Automatic Storage Management (ASM)
 upgrading, 2-14
automatic undo management
 migrating to, 4-13
 UNDO_MANAGEMENT, A-12
AutoTask, A-7

B

backing up the database, 2-38
backup
backup mode, files in, 2-34
backups
 after upgrading, 4-10
 before downgrading, 6-4
 preparing a strategy, 2-11
benchmarks, 2-10
benefits of options for upgrading precompiler and
 Oracle Call Interface (OCI) applications, 5-5
binary XML storage, A-10

C

capturing and replaying database workload, 2-8
case sensitivity
 for passwords, 4-11
catcon.pl, 1-3, 3-33, 3-38, 6-12
catctl.pl, 1-3
catdwgrd.sql, 1-3
catdwgrd.sql script, 6-8
catnoamd.sql, 1-3
catnoexf.sql, 1-3
CATRELOD.SQL script, 6-11, 6-14
CATUPGRD.SQL script
 manual upgrade and. *See* CATCTL.PL script
 manual upgrade and
catuppst.sql, 1-3
CDBs, 3-7

- upgrading and, 3-1
- change passwords
 - for Oracle-supplied accounts, 4-22
- changing scripts to use new features, 5-7
- client and server
 - configurations and upgrading, 5-3
 - types of upgrades, 5-2
- client software
 - upgrading, 5-4
- client-server configurations, 1-9
- client-side dynamic library, 5-5
- CLUSTER_DATABASE initialization
 - parameter, 2-41, 3-43
- command-line upgrade. *See* manual upgrade
- compatibility
 - applications, 5-1
 - between Oracle releases, 1-10
 - checking for incompatibilities, 6-4
 - COMPATIBLE initialization parameter, 1-11
 - downgrading, 1-12
 - overview for Oracle Database, 1-10
- COMPATIBLE initialization parameter, 1-11, 4-20
 - checking the level of, 1-13
 - considerations for downgrading, 1-12
 - default, 1-11
 - Oracle recommendation, 1-13
 - setting, 4-24
 - values, 1-11
- component status, 3-47
- compression
 - sqlnet.ora file parameters and, 2-41
- compression scheme, 2-41
 - SQLNET.COMPRESSION, 2-41
 - SQLNET.COMPRESSION_LEVELS, 2-41
 - SQLNET.COMPRESSION_THRESHOLD, 2-41
- configuration files
 - copying, 2-39
- copying configuration files, 2-39
- CREATE pfile FROM spfile, 2-40
- CREATE TABLE AS, 1-2
- crsuser user, 2-15
- CSS daemon
 - cluster synchronization services daemon, 2-16
- cursor cache, SMB, 2-9

D

- data copying
 - using Export/Import, 7-1
- data dictionary
 - about changes that affect applications, 5-2
- data files
 - relocating to upgraded environment, 2-12
- data mining models, A-8
- Data Pump Export/Import
 - advantages of using, 2-4
 - recommendations, 7-1
 - versus Original Export/Import, 7-1
 - when to use, 7-1
 - with subsets of production database, 2-42

- Database Links, 2-31
- database recycle bin
 - purging, 2-34
- Database Replay
 - database workloads before upgrading, 2-8
- Database Upgrade Assistant (DBUA)
 - advantages, 2-3
 - registering the database in the listener.ora
 - file, 2-41
 - running, 3-6
 - silent mode, 3-23
 - starting, 3-7, 3-8
- Database XE, 1-17
- databases
 - downgrading, 6-7
 - upgrading the client software, 5-4
- DBA_RECYCLEBIN statement, 2-34
- DBA_REGISTRY view, 3-47
- DBMS_DST package, 2-32
- DBMS_DST PL/SQL package
 - ORA-01822 time zone region not found
 - error, 3-46
- DBMS_LDAP package, 2-30
- DBMS_LDAP.init parameter, 2-30
- DBMS_NETWORK_ACL_ADMIN package, 2-31
- DBMS_STATS package
 - upgrading statistics tables, 4-4
- DBMS_STATS.GATHER_DICTIONARY_STATS
 - procedure, 2-32
- DBUA. *See* Database Upgrade Assistant
- defaultTableSchema attribute (deprecated), 8-23
- deprecated parameters and desupported parameters
 - display, 2-40
- developer applications
 - upgrading forms and, 5-8
- diagnostic log files
 - location of, 1-4
- DIAGNOSTIC_DEST, 1-4
- direct upgrade, 2-2
- disk group compatibility, 4-20
- disks
 - specifying preferred read failure groups, 4-20
- distributed transactions
 - resolving outstanding, 2-34
- DMSYS schema objects, A-8
- downgrading
 - backing up your database, 6-4
 - binary XML storage, A-10
 - CATRELOD.SQL, 6-11, 6-14
 - checking for incompatibilities, 6-4
 - Oracle Enterprise Manager and, 6-15
 - ORADIM and, 6-10
 - patchset releases, 6-1
 - procedure for, 6-7
 - scripts, 6-8
 - rerunning, 6-8
- downstream capture
 - Oracle Streams, 2-20

E

emca -restore command, 6-15
emdwrgrd, 1-3
emdwrgrd utility, 1-3, 2-12, 6-15
emptying the database recycle bin, 2-34
emremove.sql, 1-4
enforcing case-sensitivity for passwords, 4-11
environment variables
 required for upgrading, 3-31, 3-36
evaluation of numeric literals, 5-7
Export utility, 7-1
 data copying, 7-1
export/import
 advantages and disadvantages, 2-4
 benefits, 2-5
 effects on upgraded databases, 2-5
 recommendations, 7-1
 time requirements, 2-6
extended distance cluster configurations
 preferred read disks, 4-20
extents
 reading from secondary, 4-20
externally authenticated SSL users, 4-4
extusrupgrade script, 4-4

F

FAILED_LOGIN_ATTEMPTS initialization parameter
 DEFAULT limit, A-15
Fast Recovery Area, 3-45
file access control
 with the SYSASM role, 17
files in backup mode, 2-34
 getting a list of, 2-34
fine-grained access control
 network utility packages, 2-30
fn:matches XQuery function, 22
fn:replace XQuery function, 22
Forms
 upgrading Oracle Forms applications, 5-8
Full Transportable Export/Import, 3-42

G

gathering statistics, 2-32
getting a list of files in backup mode, 2-34

H

Hardware Assisted Resilient Data (HARD)
 upgrading systems, 4-25
HttpUriType type, 2-30

I

identifying invalid objects, 2-33
IFILE (include file)
 editing entry, 2-40
 See also text initialization parameter file
 (PFILE), 2-40

Import utility, 7-1
 data copying, 7-1
incompatibilities
 checking for, 6-4
initialization parameters
 adjusting, 2-40
 adjusting for Oracle Database 10g, 4-24
 ASM_PREFERRED_READ_FAILURE_
 GROUPS, 4-21
 COMPATIBLE, 1-11, 4-20
 See also server parameter file (SPFILE), 2-40
install logs, 3-48
installation
 Oracle Database software, 2-25
installation owners, 2-15
instances
 starting after a downgrade, 6-11
INTEGER arithmetic
 evaluation of numeric literals, 5-7
interim upgrade, 2-2
intermediate releases
 interim upgrading, 2-2
interoperability, 1-13
invalid non-SYS or non-SYSTEM objects, 2-32
invalid objects
 utlrlp.sql script and, 3-33, 3-38, 3-47, 6-12
invalid objects, identifying, 2-33
INVALID status, 3-47
invalid SYS or SYSTEM objects, 2-32

K

KEEP IDENTITY clause, 2-18
knowledge base, 1-5

L

listener.ora file, 2-42
 modifying, 2-41
listeners
 modifying with Oracle Net Configuration
 Assistant, 2-41
load
 level of concurrent demand when
 upgrading, 2-10
load testing, 2-10
localconfig command, 2-15
logical standby databases
 rolling upgrades, 1-15
login
 new DEFAULT limit, A-15
lsinventory command, 4-2
lsnrctl command
 Oracle Grid Infrastructure home and, 2-42

M

maintainOrder attribute (deprecated), 8-23
maintenance release upgrade, 5-3
maintenance tasks
 scheduling with AutoTask, A-7

- major database release upgrade, 5-3
- manual upgrade, 1-2
 - advantages, 2-4
 - backing up the database, 2-38
 - OCR configuration, 4-24
- mapUnboundedStringToLob attribute (deprecated), 8-23
- matches XQuery function, 22
- materialized view refreshes
 - query status, 2-33
- maxOccurs attribute in xdb namespace (deprecated), 8-23
- media recovery, 2-33
- migrating data, 1-1
 - to a different operating system, 2-19
- migrating listener from Oracle home with lsnrctl command, 2-42
- moving data with Data Pump Export/Import, 7-1
- multitenant container database, 3-1, 3-35
- multiversioning, 1-9
- My Oracle Support
 - knowledge base, 1-5
 - link to Upgrade Companion, 1-5, 2-7

N

- network time protocol (NTP), 2-17
- network utility packages
 - access control to, A-10
- networks
 - granting ACL access to network utility packages, 2-30
- new features
 - adding after upgrade, 4-12
 - changing scripts to use and, 5-7
 - for latest release of Oracle Database, xvii
- new features, learning about, 2-2
- NO SCRIPT status, 3-48
- not relinking upgraded application, 5-6
- NTP. *See* network time protocol
- NUMBER arithmetic
 - evaluation of numeric literals, 5-7
- numeric computation
 - evaluation of numeric literals, 5-7
- numeric literals
 - evaluating, 5-7

O

- OCI applications
 - changing, 5-7
 - changing to use new features, 5-7
 - dynamically-linked, 5-5
 - statically-linked, 5-5
 - upgrade and linking, 5-5
 - upgrading, 5-2
 - upgrading options, 5-5
- OFA, 1-13
- olspreupgrade.sql, 1-4
- one-off patches, 1-14

- OPatch lsinventory command, 4-2
- operating system
 - migrating data to, 2-19
- operating system requirements, 2-19
- Optimal Flexible Architecture. *See* OFA
- optimizer statistics, gathering, 2-32
- OPTION OFF status, 3-48
- options for upgrading precompiler and Oracle Call Interface (OCI) applications, 5-5
- ORA_TZFILE
 - unsetting after downgrade, 6-6
- ora:matches Oracle XQuery function (deprecated), 22
- ora:replace Oracle XQuery function (deprecated), 22
- ORA-00336 log file size xxxx blocks error, 3-43
- ORA-00401 value for parameter compatible error, 3-43
- Ora-00704 bootstrap failure error, 3-43
- ORA-00904 "TZ_VERSION" invalid identifier error, 3-44
- ORA-00942
 - table or view does not exist, 3-44
- ORA-00942 table or view does not exist error, 3-44
- ORA-01408 error message, 3-45
- ORA-01562 failed to extend rollback segment number error, 3-45
- ORA-01722 invalid number error, 3-44
- ORA-01822 time zone region not found error, 3-46
- ORA-04031 unable to allocate nnn bytes of shared memory error, 3-45
- ORA-19815 WARNING db_recovery_file_dest_size error, 3-45
- ORA-24247
 - network access denied by access control list (ACL) error, 2-30
- ORA-39700 database must be opened with UPGRADE error, 3-43
- ORA-39701 database must be mounted EXCLUSIVE error, 3-43
- Oracle Application Express
 - apexrelod.sql file, 6-7
 - update, 4-6
- Oracle Application Express configuration, 4-6
- Oracle ASM
 - database upgrade after, 4-8
 - disk group compatibility, A-8
 - optional tasks after upgrade, 4-8
 - RMAN backup and, xviii
- Oracle ASM Cluster Assistant, 2-15
- Oracle ASM Configuration Assistant (ASMCA), 2-17
- Oracle ASM. *See* Oracle Automatic Storage Management
- Oracle Automatic Storage Management, 1-14
 - disk group compatibility, 4-20
 - file access control, 17
 - preferred read failure groups, 4-20
 - rolling upgrades and, 1-14, 1-15
 - upgrading, 2-15
- Oracle Call Interface (OCI)
 - upgrading applications and, 2-42

- Oracle Change Data Capture. *See* Oracle GoldenGate, 8-5, 8-12, A-3
- Oracle Cluster Registry (OCR)
 - upgrading manually, 4-24
- Oracle Cluster Time Synchronization Service, 2-17
- Oracle Clusterware
 - upgrading, 2-15
- Oracle Data Guard
 - rolling upgrades, 1-15
- Oracle Data Pump, 1-2
- Oracle Database Enterprise Edition
 - converting to Standard Edition, 1-17
- Oracle Database Express Edition, 1-17
 - recommended tasks after upgrade, 4-21
 - upgrading to Oracle Database, 1-17
- Oracle Database Standard Edition
 - converting to Enterprise Edition, 1-15
- Oracle Database Vault, 2-21, 2-26
- Oracle Database XE, 1-17
 - upgrading to Oracle Database, 1-17
- Oracle GoldenGate, 8-12, A-3
- Oracle home
 - copying configuration files from, 2-39
 - multiple, 1-10
 - ORACLE_HOME database directory on Windows, 2-39
 - ORACLE_HOME dbs directory on Linux or UNIX, 2-39
- Oracle Label Security, 2-26
- Oracle Names support, 8-19
- Oracle Net Configuration Assistant, 2-41
- Oracle Real Application Clusters
 - rolling upgrades and, 1-14
 - upgrading, 2-15
- Oracle release numbers, 1-9
- Oracle RMAN
 - backing up the database, 2-38
 - running, 2-39
- Oracle software installation owners, 2-15
- Oracle Streams
 - downstream capture, 2-20
- Oracle Text-supplied knowledge bases
 - upgrading and, 4-5
- Oracle Universal Installer, 1-2, 2-26
- Oracle user accounts, 2-15
- Oracle Warehouse Builder (OWB)
 - upgrading and, 3-34, 3-39
- Oracle XML Database
 - binary XML storage, A-10
 - mandatory with installation, xix
- ORACLE_BASE
 - DIAGNOSTIC_DEST and, 1-5
- Oracle-supplied accounts
 - change passwords, 4-22
- ORADIM
 - downgrading and, 6-10
 - upgrading and, 3-30, 3-35
- orapwSID password file, 2-40
- Original Export/Import
 - versus Data Pump Export/Import, 7-1

- OSDBA group, 2-17
- OUI. *See* Oracle Universal Installer

P

- parallel processing, xviii
- Parallel Upgrade Utility
 - catctl.pl, xviii
- parameter file
 - backing up, 2-40
- passwords
 - case sensitive, 4-11
- patch set updates, 2-27
- patches
 - one-off, 1-14
- patchset releases
 - downgrading, 6-1
- PDBs, xix, 3-7
- PFILE. *See* text initialization parameter file (PFILE)
- physical standby database
 - performing rolling upgrades, 2-18
 - rolling upgrades, 1-15
- PL/SQL Native Compilation, A-10
- PL/SQL packages
 - upgrading those not installed by default, 2-12
- Pluggable Databases, xix
 - unified auditing migration and, 4-15
- postupgrade_fixups.sql, 1-4
- precompiler application
 - changing to use new features, 5-7
- precompiler applications
 - upgrading and, 2-42
- precompilers
 - applications
 - changing, 5-7
 - upgrading options, 5-5
 - upgrading applications, 5-2
- preferred read failure groups
 - setting up, 4-20
- preprocessing script
 - OLS and DV, 1-4
- preupgrade directory, 2-29
- Pre-Upgrade Information Tool
 - deprecated parameters and unsupported parameters display, 2-40
 - running, 2-28
 - warnings and actions to take, 2-29
 - working with, 2-29
- preupgrd.sql, 1-4
- PRKH-1014 error, 2-25
- Pro*C/C++ applications, 2-7
- procedure for gathering optimizer statistics, 2-32
- PURGE DBA_RECYCLE_BIN system privilege, 2-34
- PURGE statement, 2-34
- purging the database recycle bin, 2-34

Q

- query for status of materialized view refreshes, 2-33

R

- RDBMS DST patch, 2-32
- read-only and offline tablespaces, 2-18
- recompiling invalid objects, 3-33, 3-38, 6-12
- recovery catalog
 - upgrading, 4-4
- recycle bin. *See* database recycle bin
- refreshes
 - materialized view, 2-33
- registry\$nonsys_inv_objs table, 2-32
- registry\$sys_inv_objs table, 2-32
- release numbers, 1-9
- releases
 - definition, 1-9
 - multiple, 1-9
 - upgrade paths, 2-2
- releasing storage space, 2-34
- REMOVED status, 3-48
- REPCAT\$_AUDIT_COLUMN_IDX1 index, 3-46
- replace XQuery function, 22
- reserved words
 - additions and applications, 5-2
- RMAN. *See* Oracle RMAN
- role-allocated software owner, 4-8
- rollback segments
 - migrating to automatic undo management, 4-13
- Rolling Upgrade Using Active Data Guard, 2-18
- rolling upgrades
 - methods, 1-14
 - Oracle Clusterware and, 1-15
 - physical standby database, 1-15
 - summary of methods, 1-14
 - to clustered ASM instances, 2-14
 - with logical standby databases, 2-18
 - with physical standby database, 2-18
 - with SQL Apply and logical standby databases, 1-15, 2-18
- root.sh script, 2-17
- rootupgrade.sh script, 2-26
- rpath option for linking, 5-5
- running multiple Oracle releases, 1-9
- run-time library search path, 5-5

S

- scheduling
 - AutoTask, A-7
- scripts
 - downgrading, 6-8
 - manual upgrade and, 3-32, 3-37
 - rerunning, 6-8
 - running the Pre-Upgrade Information tool, 2-27
- security
 - case-sensitive passwords, 4-11
- server
 - compatibility rules, 5-4
- server parameter file (SPFILE), 2-39
 - migrating to, 4-23
 - upgrading systems with HARD-compliant storage, 4-25

- SP2-1540 "Oracle Database cannot startup in an Edition session" error, 3-45
- SPFILE. *See* server parameter file (SPFILE)
- spool upgrade results to a log file, 2-29
- SQL Access Advisor, A-12
- SQL Apply
 - performing rolling upgrades, 2-18
 - rolling upgrades, 1-15
- SQL execution plans, 2-10
- SQL Management Base (SMB), 2-9
 - cursor cache, 2-9
- SQL Performance Analyzer, 2-8
- SQL plan baselines, A-9
 - unpacking, 2-10
- SQL plan management, 2-8, A-9
 - SQL Management Base and, 2-9
- SQL queries
 - testing, 2-10
- SQL workload, 2-8
- SQL*Plus
 - scripts
 - upgrading, 5-7
- SQLCollSchema attribute (deprecated), 8-23
- sqlnet.ora file
 - compression and, 2-41
- SQLSchema attribute (deprecated), 8-23
- srclang attribute (deprecated), 8-23
- SSL external users conversion, 4-4
- staging table
 - creating, 2-10
- Standard Edition, 1-17
 - Export utility, 1-17
 - starter database, 3-53, A-12
- standby databases, 2-18
- STARTUP UPGRADE command, 6-11
- statically linked Oracle client-side library code, 5-5
- statistics
 - gathering optimizer, 2-32
- statistics tables
 - upgrading, 4-4
- status
 - INVALID, 3-47
 - materialized view refreshes, 2-33
 - NO SCRIPT, 3-48
 - OPTION OFF, 3-48
 - REMOVED, 3-48
 - UPGRADED, 3-47
- storeVarrayAsTable attribute (deprecated), 8-23
- support note 472937.1, Information On Installed Database Components, 2-29
- support note 730365.1, Oracle Database Upgrade Path Reference List, 2-27
- support note 753041.1, How to Diagnose Components with NON VALID Status, 2-29
- support note 854428.1, Patch Set Updates for Oracle Products, 2-27
- support note ID 412160.1, for RDBMS DST patches, 2-32
- support. *See* My Oracle Support
- supported releases for direct upgrade, 1-5

- symbolic link
 - Oracle Universal Installer and, 5-5
- synchronizing the standby database, 2-35
- syntax check
 - application code, 5-6
- SYSASM role, A-7
 - Oracle ASM file access control, 17
- SYSAUX tablespace
 - Oracle XML Database and, xix
- SYS.LINK\$ table, 2-31
- system considerations and requirements, 2-11

T

- tablespaces
 - read-only and offline, 2-18
- testing
 - applications for upgrade, 2-43, 4-18
 - developing a plan, 2-6
 - functional for upgrade, 2-7
 - high availability for upgrade, 2-7
 - integration for upgrade, 2-7
 - minimal for upgrade, 2-7
 - performance for upgrade, 2-7
 - the upgraded test database, 2-43
 - using Database Replay, 2-8
 - volume/load stress for upgrade, 2-10
- testing the upgrade process, 2-42
- text initialization parameter file (PFILE), 2-39
- time synchronization on Oracle RAC, 2-17
- time zone file
 - how to resolve a mismatch, 2-31
 - unsettling after downgrade, 6-6
 - version mismatch, 3-46
- TIMESTAMP WITH TIME_ZONE data type, 2-31
- TIMESTAMP WITH TIMEZONE data type, 2-31
- training, where to find, 2-2
- transactions
 - distributed
 - resolving outstanding, 2-34
- translate attribute (deprecated), 8-23
- Transportable Export/Import, 3-42
- troubleshooting
 - EDITION session error, 3-45
 - Flash Recovery Area, 3-45
 - ORA-00942 table or view does not exist, 3-44
 - ORA-01408 index error, 3-45
 - rollback segments/undo tablespace, 3-45
 - running out of resources, 3-44
 - shared memory, 3-45
 - starting database in upgrade mode, 3-43
 - SYSTEM and SYSAUX tablespaces, 3-44
 - upgrade termination
 - due to ORA_00904, 3-44
 - due to ORA_01722, 3-44
 - upgrades, 3-42
- troubleshooting the upgrade
 - termination due to ORA_00942, 3-44
- type of software upgrade, 5-2

U

- UNDO_MANAGEMENT initialization
 - parameter, 4-13
 - default, A-12
- unified auditing
 - removal of schema and roles, 2-25
 - See auditing
- Upgrade Companion
 - link to on My Oracle Support, 1-5, 2-7
- upgrade methods
 - choosing, 2-3
 - Data Pump Export/Import, 2-4
 - Database Upgrade Assistant, 1-2
 - Database Upgrade Assistant (DBUA), 2-3
 - manual, 2-4
 - silent mode, 3-23
- upgrade path
 - determining, 2-2
 - table, 2-2
- upgrade procedures
 - summary of major steps, 1-6
 - workflow, 1-6
- upgrade process testing, 2-42
- UPGRADED status, 3-47
- upgraded test database, 2-43
- upgrading
 - applications, 5-1
 - compatibility rules, 5-3
 - options, 5-5
 - relinking, 5-3
 - backup strategy, 2-11
 - binary XML storage, A-10
 - initialization parameters, 2-40
 - new administrative procedures, 4-12
 - Oracle Application Express, 4-6
 - Oracle Automatic Storage Management, 2-15
 - Oracle Clusterware, 2-15
 - Oracle Forms applications, 5-8
 - Oracle Real Application Clusters, 2-15
 - ORADIM and, 3-30, 3-35
 - post upgrade actions, 4-1
 - preparation, 2-1
 - recovery catalog, 4-4
 - rolling upgrades and, 1-14
 - scripts and manual upgrade, 3-32, 3-37
 - SQL*Plus scripts, 5-7
 - statistics tables, 4-4
 - testing, 2-6
 - troubleshooting, 3-42
 - using the Database Upgrade Assistant, 3-6
 - where to find information about, 1-5
- upgrading a cluster database
 - setting the CLUSTER_DATABASE initialization parameter, 2-41
- upgrading and plugging into a CDB, 3-2
- UTL_INADDR package, 2-30
- UTL_MAIL package, 2-30
- UTL_SMTP package, 2-30
- UTL_TCP package, 2-30
- utlrl.sql, 1-4

- utlrp.sql script
 - for recompiling invalid objects, 3-33, 3-38, 6-12
- utlu121s.sql, 1-4
- utluiobj.sql, 1-4
- utluiobj.sql script, 2-32

V

- V\$OPTION view, 3-48
- volume
 - amount of data to be upgraded, 2-10

W

- word size
 - 64-bit software, 1-14
- workloads
 - capturing and replaying, 2-8

X

- xdb:defaultTableSchema attribute (deprecated), 8-23
- xdb:maintainOrder attribute (deprecated), 8-23
- xdb:mapUnboundedStringToLob attribute (deprecated), 8-23
- xdb:maxOccurs attribute (deprecated), 8-23
- xdb:SQLCollSchema attribute (deprecated), 8-23
- xdb:SQLSchema attribute (deprecated), 8-23
- xdb:srclang attribute (deprecated), 8-23
- xdb:storeVarrayAsTable attribute (deprecated), 8-23
- xdb:translate attribute (deprecated), 8-23
- XE, 1-17
- XQuery language
 - functions
 - fn:matches, 22
 - fn:replace, 22
 - ora:,matches (deprecated, Oracle), 22
 - ora:replace (deprecated, Oracle), 22