



ORACLE

Oracle Database Upgrade: Quick Start Guide

A quick reference to a successful Oracle Database Upgrade

May, 2024, Version 2.0
Copyright © 2024, Oracle and/or its affiliates
Public

Introduction

Oracle is investing in tools, techniques, and procedures that simplify and fully automate the Oracle Database upgrade process, whether it's on-premises or in the cloud. The AutoUpgrade utility makes Oracle Database upgrades easier and doable with just two commands. Although the database upgrade itself is simple, it is often a piece of a more extensive process that includes other tasks and involves many parts of an organization. This quick start guide discusses the four recommended steps for a successful upgrade.

Step 1: Verify your database and application certification

Familiarize yourself with the new release by reading the [Database Upgrade Guide](#) and pay special attention to the chapter documenting [behavior changes, deprecated and desupported features](#). You can also find the hardware and software requirements for the new release in the platform-specific [installation guides](#). For up-to-date information on software certifications and requirements, visit [My Oracle Support](#) and use the “Certifications” tab to search for the new database release.

ORACLE MY ORACLE SUPPORT PowerView is Off

Dashboard Knowledge Service Requests Patches & Updates Community Certifications Managed Cloud CRM On Demand Systems Advanced Customer Serv

Certifications >

Search Results: Oracle Database 19.0.0.0.0

Certification Search

Certification Results

Displaying Oracle Database 19.0.0.0.0 Certifications.

View Share Link

Certified With	Number of Releases / Versions
Operating Systems (7 Items)	
HP-UX Itanium	1 Version (11.31)
IBM AIX on POWER Systems (64-bit)	2 Versions (7.2, 7.1)
Linux on IBM Z	2 Versions (SLES 12, Red Hat Enterprise Linux 7)
Linux x86-64	4 Versions (SLES 15, SLES 12, Red Hat Enterprise Linux 7, Oracle Linux 7)
Microsoft Windows x64 (64-bit)	5 Versions (8.1, 2019, 2016, 2012 R2, 10)
Oracle Solaris on SPARC (64-bit)	1 Version (11)
Oracle Solaris on x86-64 (64-bit)	2 Versions (11.4, 11)
Agents (1 Item)	
Application Servers (1 Item)	
Databases (4 Items)	
Desktop Applications, Browsers and Clients (1 Item)	

Screenshot from My Oracle Support showing the current certification for Oracle Database 19c

In addition, you should verify the certifications for any third-party application connecting to the database. Ensure that the new database release is supported and pay attention to any database release-specific information.

Step 2: Install Oracle Database with the latest patches

Follow the instructions in the [platform-specific installation guides](#) to install the newest database release. Install the software in a new location to allow for an out-of-place upgrade. Although possible, Oracle does not recommend in-place upgrades because it increases downtime and complicates fallback operations.

In addition, Oracle recommends the following actions for your new Oracle Home:

- Apply the latest Release Update
- Apply the latest Monthly Recommended Patches (MRP)
- Review the list of important one-off patches and install the ones appropriate for your database

You should apply the patches before you perform the database upgrade. You can find the latest Release Update for your database version by using the My Oracle Support note, "[Assistant: Download Reference for Oracle Database/GI Update, Revision, PSU, SPU\(CPU\), Bundle Patches, Patchsets and Base Releases \(Doc ID 2118136.2\)](#)". You can find MRPs and the list of important one-off fixes in My Oracle Support note, "[Oracle Database 19c Important Recommended One-off Patches \(Doc ID 555.1\)](#)"

Oracle recommends upgrading to the latest so-called "Long-Term Support" release to ensure that patches, including security-related bug fixes, are available. At the time of writing, this applies to Oracle Database releases 23 and 19, and this will give you a much longer period of support compared to the innovation releases (in this case, release 21)

If you need to upgrade to an innovation release, you must plan the next upgrade in due time to avoid ending up in a situation where the database release is no longer under bug fixing support. Consult My Oracle Support note, "[Release Schedule of Current Database Releases \(Doc ID 742060.1\)](#)" for further information.

Step 3: Upgrade using AutoUpgrade

Oracle Database release 23 supports the multitenant architecture only. If your database is a non-CDB database, you must convert it to a pluggable database as part of the upgrade. You plug the PDB into an existing container database already running Oracle Database release 23.

Before starting the upgrade, you must ensure a viable fallback option, such as a backup or a restore point. Familiarize yourself with these options and ensure adequate experience in using them.

Oracle strongly recommends using the AutoUpgrade utility to perform the actual database upgrade and conversion. This tool offers the best balance between configurability, control, and ease of use. Plus, it automatically employs the latest best practices and recommendations, does extensive logging, and can perform multiple upgrades simultaneously. Furthermore, it utilizes all available nodes in a Real Application Clusters environment, deals with encrypted databases, automates migrations from non-CDB to PDB, and much more.

The database Oracle Home includes a version of AutoUpgrade, however, we strongly encourage you to always download the latest version from My Oracle Support: "[AutoUpgrade Tool \(Doc ID 2485457.1\)](#)". Newer versions of AutoUpgrade are fully backward compatible. For instance, AutoUpgrade version 24 can also upgrade databases of earlier releases (in this case Oracle Database releases 23, 21, 19, 18, and 12.2.0.1).

To use AutoUpgrade, you must create a simple configuration file that specifies the database or databases you want to upgrade:

```
upg1.source_home=/u01/app/oracle/product/19
upg1.target_home=/u01/app/oracle/product/23
upg1.sid=MYDB
```

If your database is a non-CDB, you must also configure the PDB conversion by specifying the SID of the container database where you want to plug in.

```
upg1.source_home=/u01/app/oracle/product/19
upg1.target_home=/u01/app/oracle/product/23
upg1.sid=MYDB
upg1.target_cdb=CDB23
```

Next, you analyze the database to identify any potential showstoppers and get information on issues you should consider resolving. Be sure that the parameter `-config` points to the name of your configuration file:

```
java -jar $ORACLE_HOME/rdbms/admin/autoupgrade.jar -config config.cfg -mode analyze
```

Finally, the deploy phase will conduct the actual upgrade:

```
java -jar $ORACLE_HOME/rdbms/admin/autoupgrade.jar -config config.cfg -mode deploy
```

Following these simple steps, your database is now upgraded to the new release and ready to use. In the event of an error, the default configuration of AutoUpgrade will automatically revert the database using Flashback Database to its pre-upgrade state, and you can use it as if the upgrade did not happen. Note, this applies to Enterprise Edition only. For Standard Edition 2 databases, you must have your own fallback option in place.

Please refer to the [documentation](#) for complete information on AutoUpgrade, and also, please visit the [Upgrade your database – Now!](#) blog for valuable information, tips, and recommendations. In addition, we have multiple [on-demand webinars](#) with additional information on upgrades.

Step 4: Test using the right features, options, and packs

When testing a database before the actual production upgrade, it is essential to have a comparable test system to ensure that your tests are as realistic as possible. This applies not only to the underlying hardware but also to the amount of data being used and the workload being generated.

The [Diagnostics and Tuning packs](#) are very helpful in gathering performance baselines from your production system prior to making any major change, including database upgrades. Oracle recommends retaining at least 31 days of Automatic Workload Repository (AWR) snapshots to characterize and compare system performance before and after the upgrade.

[Oracle Real Application Testing](#) helps you assess the effect of the upgrade by running realistic workloads on the test system using Database Replay. Even more important, SQL Performance Analyzer can help you identify regressed SQLs.

In addition, you should use [SQL Plan Management](#) to ensure plan stability by identifying key SQL statements and fixing their plans. Later, a potentially better plan can be verified by the database and put into use in a controlled manner.

Speaking of testing, please ensure you have tested your rollback and fallback options in a test system. It is important to verify, for example, that a backup can – in fact – be restored in the required service window or that you can properly downgrade the database after go-live, and that you have the necessary experience and training to do so.

Connect with us

Call **+1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 2024, Oracle and/or its affiliates. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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