

Oracle® Database

Unplugging, Plugging, and Upgrading a PDB to a New CDB



19c
F10901-02
May 2019

ORACLE®

Oracle Database Unplugging, Plugging, and Upgrading a PDB to a New CDB, 19c

F10901-02

Copyright © 2018, 2019, Oracle and/or its affiliates. All rights reserved.

Primary Authors: Sunil Surabhi, Nirmal Kumar

Contributing Authors: Lance Ashdown, Padmaja Potineni, Rajesh Bhatiya, Prakash Jashnani, Douglas Williams, Mark Bauer

Contributors: Roy Swonger, Byron Motta, Hector Vieyra Farfan, Carol Tagliaferri, Mike Dietrich, Marcus Doeringer, Umesh Aswathnarayana Rao, Rae Burns, Subrahmanyam Kodavaluru, Cindy Lim, Amar Mbaye, Akash Pathak, Thomas Zhang, Zhihai Zhang

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		
<hr/>		
	Use Case Scenario for this Document	iv
	Documentation Accessibility	iv
1	Upgrading Multitenant Architecture Sequentially Using Unplug-Plug	
<hr/>		
	About Upgrading Pluggable Databases (PDBs) Sequentially	1-1
2	Upgrading Pluggable Databases Sequentially	
<hr/>		
	Unplugging the Earlier Release PDB from the Earlier Release CDB	2-1
	Plugging in the Earlier Release PDB to the Later Release CDB	2-2
	Upgrading the Earlier Release PDB to the Later Release	2-3
	Use Inclusion or Exclusion Lists for PDB Upgrades	2-4

Preface

This guide provides a compilation of topics from the Oracle Database user assistance documentation that are collected to help you complete a specific use case scenario.

- [Use Case Scenario for this Document](#)
- [Documentation Accessibility](#)

Use Case Scenario for this Document

Use this scenario document to assist you to unplug earlier release PDBs, plug them into a later release CDB, and then upgrade the PDBs.

Prerequisites for this Scenario

- You have installed the same release Oracle Database software on the destination server, and it is updated to the same release update and release update revision.
- You have completed preparation of the new Oracle home.
- You have run the Pre-Upgrade Information Tool on the PDB.
- The endian format of the source CDB and the target CDB are identical.
- The same set of options are installed on the source CDB and the target CDB.
- The source CDB and the target CDB have compatible character sets and national character sets.

The source CDB and the target CDB can be on the same or different server hardware.

Outline for this Scenario

- 1. Upgrading Multitenant Architecture Sequentially Using Unplug-Plug.** Learn about the unplug-plug method for carrying out PDB upgrades.
- 2. Upgrading Pluggable Databases Sequentially.** Use these procedures to complete your PDB upgrade:
 - a.** Unplug earlier release PDBs from the earlier release CDB.
 - b.** Plug in the earlier release PDB to the later release CDB.
 - c.** Upgrade the earlier release PDB to the later release CDB.
 - d.** Use inclusion or exclusion lists for PDB upgrades.

These steps correspond to the chapters in this document.

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.

1

Upgrading Multitenant Architecture Sequentially Using Unplug-Plug

To upgrade pluggable databases (PDBs) that are in an earlier release multitenant container databases (CDBs), Oracle Database Release 12c (12.1.0.1) and later, you can unplug the PDBs from the earlier release CDB, and plug the PDBs into the later release CDB.

- [About Upgrading Pluggable Databases \(PDBs\) Sequentially](#)
You can upgrade PDBs by unplugging a PDB from an earlier release CDB, plugging it into a later release CDB, and then upgrading that PDB to the later release.

About Upgrading Pluggable Databases (PDBs) Sequentially

You can upgrade PDBs by unplugging a PDB from an earlier release CDB, plugging it into a later release CDB, and then upgrading that PDB to the later release.

CDBs can contain zero, one, or more pluggable databases (PDBs). After you install a new Oracle Database release, or after you upgrade the CDB (`CDB$ROOT`), you can upgrade one or more PDB without upgrading all of the PDBs on the CDB.

You can choose the upgrade plan that meets the needs for your service delivery. For example, you can use Oracle Database Upgrade Assistant (DBUA) to upgrade a set of PDBs, or you can use a manual upgrade to upgrade PDBs individually, or with inclusion or exclusion lists. You can upgrade the CDB and all PDBs (an In Parallel manual upgrade), or you can upgrade the CDB, and then upgrade PDBs sequentially, either individually, or in sets using inclusion or exclusion lists.

The following is a high-level list of the steps required for sequential PDB upgrades:

1. Unplug the earlier release PDB from the earlier release CDB.
2. Drop the PDB from the CDB.
3. Plug the earlier release PDB into the later release CDB.
4. Upgrade the earlier release PDB to a later release.

With Oracle Database 12c Release 2 (12.2) and later releases, you can provide lists to the Parallel Upgrade Utility to upgrade PDBs:

- Priority lists, to set the order in which PDBs are upgraded
- Inclusion lists, which enable you to designate a set of PDBs to upgrade after the PDBs listed in the priority list are upgraded
- Exclusion lists, which enable you to designate a set of PDBs that are not upgraded

 **Note:**

A PDB cannot be recovered unless it is backed up. After upgrading using the method of creating a CDB and plugging in a PDB, be sure to back up the PDB.

Related Topics

- *Oracle Database Backup and Recovery User's Guide*
- *Oracle Database Administrator's Guide*

2

Upgrading Pluggable Databases Sequentially

To upgrade PDBs after upgrading the Container Database (CDB), use the unplug-plug technique.

- [Unplugging the Earlier Release PDB from the Earlier Release CDB](#)
To prepare for upgrading the PDB, use this procedure to unplug the PDB from the earlier release CDB.
- [Plugging in the Earlier Release PDB to the Later Release CDB](#)
To Plug the PDB from the earlier release CDB to the later release CDB, use the `CREATE PLUGGABLE DATABASE` command.
- [Upgrading the Earlier Release PDB to the Later Release](#)
Open PDBs in `UPGRADE` mode use the Parallel Upgrade Utility to carry out the upgrade of the earlier-release PDB to the release level of the CDB.
- [Use Inclusion or Exclusion Lists for PDB Upgrades](#)
If you want to upgrade a subset of earlier release PDBs, then use inclusion or exclusion lists to avoid reupgrading the CDB or PDBs that are at the new release level.

Unplugging the Earlier Release PDB from the Earlier Release CDB

To prepare for upgrading the PDB, use this procedure to unplug the PDB from the earlier release CDB.

1. Run the Pre-Upgrade Information Tool on the PDB.

For example, where the PDB named `salespdb` is running in the CDB in `$ORACLE_HOME_12.2`:

```
$ORACLE_HOME_12.2/jdk/bin/java -jar  
$ORACLE_HOME_19/rdbms/admin/preupgrade.jar dir /tmp -c salespdb
```

2. Run `preupgrade_fixups.sql` on your source database.

For example:

```
CONNECT / AS SYSDBA  
SQL> ALTER SESSION SET CONTAINER=salespdb;  
  
SQL> @/tmp/preupgrade_fixups_salespdb.sql
```

3. Follow all recommendations listed in `preupgrade.log`.
4. Close the PDB you want to unplug.

For example, use the following command to close the PDB `salespdb`:

```
SQL> ALTER PLUGGABLE DATABASE salespdb CLOSE;
```

5. Log back in to `CDB$ROOT`:

```
CONNECT / AS SYSDBA  
SQL> ALTER SESSION SET CONTAINER=CDB$ROOT;
```

6. Unplug the earlier release PDB using the following SQL command syntax, where `pdb` is the name of the PDB, and `path` is the location of the PDB XML file:

```
ALTER PLUGGABLE DATABASE pdb UNPLUG INTO 'path/pdb.xml';
```

For example, where the `pdb` name is `salespdb` and `path` is `/home/oracle/salespdb.xml`:

```
SQL> ALTER PLUGGABLE DATABASE salespdb UNPLUG INTO '/home/oracle/salespdb.xml';
```

The following response displays when the command is completed:

```
Pluggable database altered
```

7. Drop the pluggable database `salespdb`, but keep data files.

Oracle recommends that you drop `salespdb` after this procedure to clean up leftover information in the CDB views, and to help to avoid future issues. As a best practice guideline, back up your PDB in the destination CDB first, and then issue the `DROP` command on the source.

 **Caution:**

After you drop the PDB from its original CDB, you cannot revert to it using previously taken backup, because the `DROP` command removes backup files.

To drop the pluggable database, enter the following command:

```
SQL> DROP PLUGGABLE DATABASE salespdb KEEP DATAFILES;
```

8. Exit.

Plugging in the Earlier Release PDB to the Later Release CDB

To Plug the PDB from the earlier release CDB to the later release CDB, use the `CREATE PLUGGABLE DATABASE` command.

This procedure example shows how to plug in a PDB when you are using Oracle-Managed Files. Refer to *Oracle Database Administrator's Guide* for additional information about plugging in PDBs.

1. Connect to the later release CDB.
2. Plug in the earlier release PDB using the following SQL command, where *pdb* is the name of the PDB, and *path* is the path where the PDB XML file is located:

```
CREATE PLUGGABLE DATABASE pdb USING 'path/pdb.xml';
```

For example:

```
SQL> CREATE PLUGGABLE DATABASE salespdb USING '/home/oracle/salespdb.xml';
```

The following response displays when the command is completed:

```
Pluggable database created.
```

 **Note:**

When you plug in an earlier release PDB, the PDB is in restricted mode. You can only open the PDB for upgrade.

Related Topics

- *Oracle Database Administrator's Guide*

Upgrading the Earlier Release PDB to the Later Release

Open PDBs in UPGRADE mode use the Parallel Upgrade Utility to carry out the upgrade of the earlier-release PDB to the release level of the CDB.

1. If needed, switch to the PDB that you want to upgrade. For example, enter the following command to switch to the PDB *salespdb*:

```
SQL> ALTER SESSION SET CONTAINER=salespdb;
```

2. Open the PDB in UPGRADE mode.

```
SQL> ALTER PLUGGABLE DATABASE OPEN UPGRADE;
```

3. Upgrade the PDB using the Parallel Upgrade Utility command (*catctl.pl*, or the shell utility *dbupgrade*).

When you upgrade a PDB, you use the commands you normally use with the Parallel Upgrade Utility. However, you also add the option `-c PDBname` to specify

which PDB you are upgrading. Capitalize the name of your PDB as shown in the following example using the PDB named `salespdb`:

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catctl.pl -d \
$ORACLE_HOME/rdbms/admin -c 'salespdb' -l $ORACLE_BASE catupgrd.sql
```

4. Review results.

The default file path for the logs is in the path `Oracle_base/cfgtoollogs/dbname/upgradedatetime`, where `Oracle_base` is the Oracle base path, `dbname` is the database name, and `upgradedatetime` is the date and time for the upgrade. The date and time strings are in the character string format `YYYYMMDDHHMMSC`, in which `YYYY` designates the year, `MM` designates the month, `DD` designates the day, `HH` designates the hour, `MM` designates the minute, and `SC` designates the second.

For example:

```
$ORACLE_BASE/cfgtoollogs/salespdb/upgrade20181015120001/upg_summary.log
```

5. Log in to SQL*Plus, and open the PDB to execute post-upgrade fixups, and to recompile the `INVALID` objects in the database:

```
SQL> STARTUP;
SQL> ALTER SESSION SET CONTAINER=salespdb;
```

6. Use the utility `catcon.pl` to run the script `postupgrade_fixups.sql`:

```
$ORACLE_HOME/perl/bin/perl catcon.pl -c 'salespdb' -n 1 -e -b
postfixups -d '''.' /tmp/cfgtoollogs/salespdb/preupgrade/
postupgrade_fixups.sql
```

7. Use the utility `catcon.pl` to run `utlrp.sql` from the `$ORACLE_HOME/rdbms/admin` directory:

```
$ORACLE_HOME/perl/bin/perl catcon.pl -c 'salespdb' -n 1 -e -b comp -d
'''.' utlrp.sql
```

The script recompiles `INVALID` objects in the database, and places a log file in the current directory with the name `comp0.log`.

Use Inclusion or Exclusion Lists for PDB Upgrades

If you want to upgrade a subset of earlier release PDBs, then use inclusion or exclusion lists to avoid reupgrading the CDB or PDBs that are at the new release level.

Oracle recommends that you record the containers that you upgrade, and use inclusion or exclusion lists to exclude these containers from successive bulk upgrades. Excluding upgraded containers from successive bulk upgrades ensures that the upgrade only runs on PDBs that require the upgrade. Avoiding reupgrades minimizes the overall upgrade time, and avoids unnecessary unavailability.

For example: If you have installed Oracle Database using a multitenant architecture deployment, then the containers `CDB$ROOT`, `PDB$SEED`, and any other PDBs created

when the CDB was created, are part of the new release multitenant architecture. If you upgraded a CDB, and at the same time upgraded a set of PDBs to the new release, then you do not need to upgrade either the CDB containers or the upgraded PDBs again.

In either case, when you plug in earlier release PDBs and then upgrade them, upgrade the PDBs with either an exclusion list, or an inclusion list:

- Use an inclusion list to specify only the set of PDBs that you want to upgrade.
- Use an exclusion list to exclude the CDB and PDB containers that are already upgraded.

If you do not use an inclusion list or an exclusion list to limit the upgrade scope, then the Parallel Upgrade Utility (`catctl.pl`) attempts to upgrade the entire CDB, not just the PDBs that require the upgrade. During that upgrade process, your system undergoes needless downtime. The inclusion list and the exclusion list options are mutually exclusive.

Index

E

exclusion lists
and PDB upgrades, [2-4](#)

O

Oracle Multitenant upgrades, [1-1](#)

P

PDB upgrades after CDB upgrade, [2-4](#)

PDBs

upgrading individually, [1-1](#)

T

troubleshooting

PDB upgrades, [2-4](#)