

Upgrading and Migrating to Oracle Database 12c Release 2 (12.2.0.1)

ORACLE WHITE PAPER | DECEMBER 2016





Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.




ORACLE®



Table of Contents

Disclaimer	ii
Introduction	1
Comparing Upgrade and Migration	2
Database Upgrade	2
Database Migration	2
Multitenant Architecture	3
Selecting a Database Upgrade or Migration Method	3
Direct Upgrade to Oracle Database 12c Release 2	3
Detailed Upgrade and Migration Method Descriptions	5
Method 1: DBUA or Command-Line Upgrades	5
Database Upgrade Assistant	5
Command-line Upgrade	7
Method 2: Full Transportable Export/Import or Transportable Tablespaces	9
Full Transportable Export/Import	9
Transportable Tablespaces	12
Method 3: Oracle Data Pump Export/Import	12
Data Pump Export/Import with Dump Files	13
Oracle Data Pump Network Mode	13
Method 4: Original Export/Import	14
Minimal Downtime Database Upgrades	15
Rolling Upgrade with Oracle Data Guard	15



RMAN Cross-Platform Incremental Backups	16
Oracle GoldenGate	16
Conclusion	16
Appendix A: Upgrade & Migration New Features in Oracle Database 12c (12.1)	18
Appendix B: Upgrade & Migration New Features in Oracle Database 12c Release 2 (12.2)	19



Introduction

Oracle Database 12c Release 2 (12.2.0.1) includes new features and enhancements that make it an attractive upgrade target for existing Oracle databases. Moving to Oracle Database 12c Release 2 may be part of an effort that includes moving to newly purchased server hardware, migrating to different storage architectures such as Oracle Automatic Storage Management, changing the database character set, migrating to a completely different operating system, or migrating to the Oracle Cloud.

Because upgrade and migration scenarios can differ in many ways, Oracle provides multiple methods for you to upgrade and migrate your databases to Oracle Database 12c Release 2. This white paper outlines these upgrade and migration methods. You will learn about different use cases and key factors to consider when choosing the method that best fits your requirements.

"AT DVAG WE DO THOROUGH TESTING IN A 3-STEP APPROACH FOR OUR CRITICAL ENVIRONMENTS. THE DECISION TO UPGRADE DIRECTLY TO ORACLE DATABASE 12C SKIPPING THE TERMINAL PATCH SET OF THE PREVIOUS DATABASE RELEASE WAS TAKEN IN ORDER TO SAVE US ONE COMPLETE TESTING CYCLE, AND FURTHERMORE THE ACCOMPANYING COSTS AS WELL"

MICHAEL KUHN, DEUTSCHE VERMÖGENSBERATUNG AG

Comparing Upgrade and Migration

Although the terms are often used as synonyms in other contexts, there is a difference between database *upgrade* and database *migration*. Understanding this difference is the first step in choosing the best upgrade or migration method for your project.

Note: The term migration can also be used when discussing the move of data from a non-Oracle database into Oracle. This white paper will cover migrations only when both the source and destination are Oracle databases.

Database Upgrade

The act of upgrading an Oracle database involves modifying the data dictionary to be compatible with a newer version of Oracle database software. Typical actions that may be part of a database upgrade include:

- » Adding, dropping, or modifying columns in system tables and views
- » Creating new system packages or procedures
- » Modifying existing system packages or procedures
- » Creating, modifying, or dropping database users, roles, and privileges
- » Modifying seed data that is used by Oracle database components

All of these actions affect the data dictionary of your database. They do not affect the data stored in your user or application tablespaces. Therefore, the sheer volume of data stored in your Oracle database has little or no bearing on a database upgrade.

Database Migration

The term “migration” applies to several different types of changes that can be applied to an Oracle database. In addition to database version, these can include a change to any or all of the following:

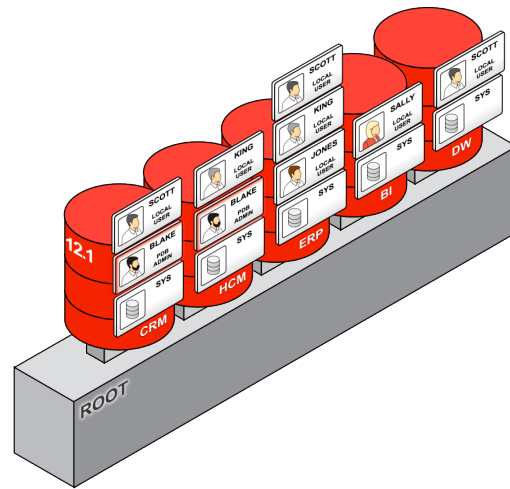
- » Computer server
- » Storage architecture
- » Character set
- » Operating system
- » Schema topology (changing the partitioning scheme)
- » Encryption
- » Compression
- » Database architecture (moving from non-CDB to PDB)

Database migration differs from database upgrade in two important ways. First, database migration generally involves moving or modifying the user and application data in the database. This means that the size of your database has a dramatic impact on your database migration project. Second, any of the above migrations can be performed on a database without upgrading it to a new version. This white paper will discuss techniques that can be used to perform both an upgrade to Oracle Database 12c and a migration simultaneously.

Multitenant Architecture

Oracle Database 12c introduced a new multitenant architecture. It enables an Oracle database to contain a portable collection of schemas, schema objects, and nonschema objects that appears to an Oracle Net client as a separate database. This self-contained collection is called a pluggable database (PDB). A multitenant container database (CDB) is an Oracle database that includes zero, one, or many user-created PDBs. This new architecture enables customers to easily consolidate multiple databases and introduces another very important type of database migration for customers to consider: migration to a PDB.

In some cases, migrating from traditional database architecture (e.g. Oracle Database 11g) to a PDB can be combined with other database upgrade or migration tasks into a single operation. In other cases, migration into a PDB may involve an additional step. The steps for migrating an existing or non-CDB database to a PDB will be described under each upgrade or migration technique in this white paper.



Selecting a Database Upgrade or Migration Method

With several upgrade and migrations available, choosing the best upgrade or migration method for a particular project requires an analysis of several important project characteristics. Each of these characteristics can influence the suitability of a given method when you upgrade or migrate to Oracle Database 12c:

- » The version from which you are upgrading or migrating, down to the patch set level
- » The source and destination operating system and version
- » The source and destination hardware platforms and their endian characteristics
- » Any plans to change the actual data layout or format, such as changing the character set, partitioning, encryption, or compression
- » Availability requirements – the amount of downtime allowed for the upgrade or migration project
- » The size of the database to be migrated
- » Whether the destination of the upgrade is an Oracle Database 12c Release 2 PDB

There is no single upgrade or migration method that is the best option for all possible upgrade and migration scenarios. However, there is a method that is best for any given scenario based on the above factors.

Direct Upgrade to Oracle Database 12c Release 2

A direct upgrade is one where either the Database Upgrade Assistant (DBUA) or command-line upgrade script is used to upgrade your database to Oracle Database 12c. Direct upgrade is supported when the source database is running one of the releases shown in the following table.

TABLE 1. DIRECT UPGRADE PATHS FOR ORACLE DATABASE 12C RELEASE 2

SOURCE RELEASE	SOURCE PATCH SET	Direct Upgrade Supported?
Oracle Database 12c	12.1.0.1, 12.1.0.2	Yes
Oracle Database 11g Release 2	11.2.0.3, 11.2.0.4	Yes
	11.2.0.1, 11.2.0.2	No. Use another method
Oracle Database 11g Release 1 and earlier	All	No. Use another method

For the cases in which direct upgrade is not supported, other methods of moving to Oracle Database 12c Release 2 will apply. These methods are described in the following section.

Upgrade and Migration Methods

There are four different upgrade and migration methods described in this white paper, and three of these methods have variants that can be used in some situations.

The methods are:

1. Database Upgrade, using either DBUA or the command-line upgrade scripts
2. Transportable tablespaces (TTS) export and import, using the Oracle Database 12c feature full transportable export/import, or the traditional TTS mode
3. Oracle Data Pump Export/Import, using either dump files or network mode
4. The Original Export/Import utilities

The following table summarizes the applicability of these upgrade and migration methods for your scenario, based on the project characteristics listed in the previous section.

TABLE 2. DATABASE UPGRADE AND MIGRATION METHODS

Method	Complexity	Speed	Minimum Source Version	Move to New Server	Change O/S	Change Data Layout, character set, encryption, compression
Unplug, Plug and Upgrade	Med	Fastest	12.1.0.1	Yes	No	No
Database Upgrade Assistant	Low	Fastest	11.2.0.3	No	No	No
Command-line Upgrade	Med	Fastest	11.2.0.3	Yes	No	No

All four of the upgrade and migration methods described in this white paper can be used to migrate into a PDB.

Full Transportable Export/Import	Med	Faster	11.2.0.3	Yes	Yes	No
Transportable Tablespaces	High	Faster	8.1.5	Yes	Yes, starting with 10.1	No
Data Pump expdp/impdp	Med	Fast	10.1	Yes	Yes	Yes
Original export/import	Med	Slow	5	Yes	Yes	Yes

Detailed Upgrade and Migration Method Descriptions

For each upgrade or migration method, this section describes how the method works, the benefits of using it, and any considerations that may affect your decision to choose that particular method.

Method 1: DBUA or Command-Line Upgrades

The easiest way to move up to Oracle Database 12c Release 2 is to upgrade your existing database using either DBUA or the command-line upgrade scripts. These are two variants of the same method, differing primarily in their user interfaces. Because these methods act on your database without creating a copy or a new instance, this is sometimes referred to as upgrading “in place.”

Database Upgrade Assistant

Database Upgrade Assistant is a graphical user interface (GUI) that guides you through the database upgrade process and presents a series of screens that allow you to specify options for your database upgrade. During the upgrade process, DBUA invokes the same scripts used for the command-line upgrade. It also performs pre-upgrade validation steps, and can automate post-upgrade tasks. Using DBUA can significantly reduce the amount of manual effort required for a database upgrade, but offers less flexibility than the command-line upgrade. DBUA is fully described in [Oracle Database Upgrade Guide](#).

Example of Using Database Upgrade Assistant

As a GUI-based utility, DBUA is very easy to use because it allows you to select options for your database upgrade using dialog boxes such as this example from the DBUA workflow:

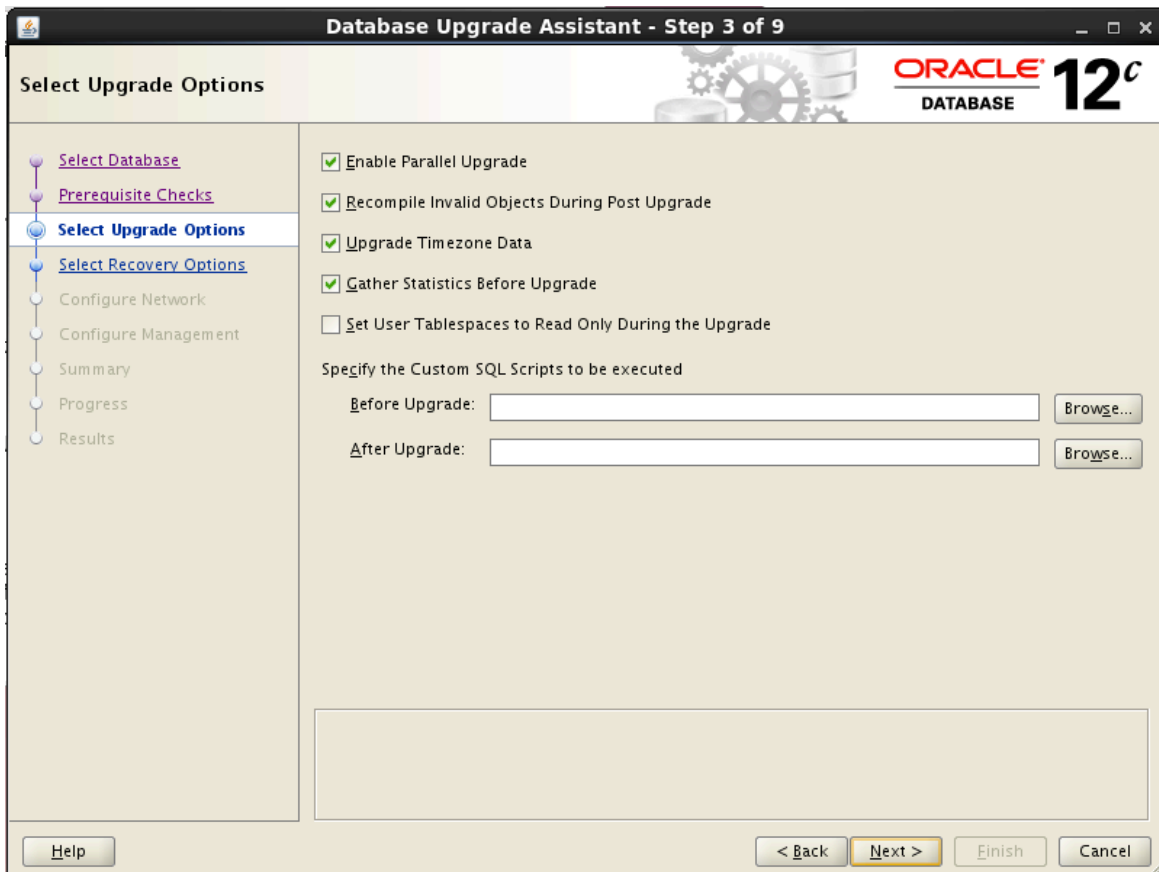


Figure 1. Screenshot of DBUA Upgrade Options Page

In this screenshot, the steps in the workflow are listed in the left-hand pane of the window, while the options available for the current (highlighted) step are in the right-hand pane. The options are shown using a tabbed interface, with the current tab being the Upgrade Options for this operation. These options include

- » Enable Parallel Upgrade, which will use the parallel upgrade introduced in Oracle Database 12c to speed up your upgrade resulting in less downtime.
- » Recompile invalid objects after the upgrade: if you do not select this option, then you should run `utlrip.sql` to recompile objects after the upgrade completes
- » Upgrade Timezone Data: if your source database does not have the latest version of time zone definitions installed, then it is a good idea to upgrade your time zone version during the upgrade. Note that the upgrade of time zone data is distinct from the database upgrade itself, and is performed as a post-upgrade task
- » Gather Statistics Before Upgrade: gathering dictionary statistics before you upgrade your database can improve the performance of the upgrade
- » Set User Tablespaces Read Only During Upgrade: this can be helpful as part of a fallback strategy in case the upgrade is interrupted
- » Specify custom SQL scripts to run before or after the upgrade: this may be helpful if you have management or monitoring scripts that are run throughout your environment during the upgrade process.

All of these options or operations can be specified at different points during a command-line upgrade, but DBUA provides a very convenient interface and automates the upgrade process. This can lower the possibility of a mistake such as a typographical error causing problems during the upgrade.

"THE NEW PARALLEL UPGRADE SCRIPT PROMISES TO DRASTICALLY REDUCE DOWNTIME DUE TO PLANNED MAINTENANCE. WE SAW A 37% IMPROVEMENT OVER THE PREVIOUS UPGRADE PROCESS IN OUR ENVIRONMENT."

HARALD STEFAN, LEITER DATENBANKEN, PAYBACK GMBH

Considerations for Using Database Upgrade Assistant

In general, DBUA can only be used when upgrading your Oracle database on its current hardware system. This is because DBUA performs the pre-upgrade validation steps using the source Oracle home, and then switches to the destination Oracle home to perform the upgrade and post-upgrade steps.

DBUA does not give the user any control over the degree of parallelism used to upgrade the database.

While DBUA does automate many of the pre-upgrade tasks identified in the database pre-upgrade scripts, there are some actions which may still require manual intervention.

Command-line Upgrade

Oracle Database 12c introduced a new command-line upgrade utility (`catctl.pl`). This utility replaces the `catupgrd.sql` script used to upgrade to previous releases of Oracle Database. The new command-line upgrade utility enables parallel processing during the database upgrade, resulting in better upgrade performance and reduced database downtime.

Command-line upgrades follow the same steps and take the same amount of time as upgrading with DBUA. They are most commonly used by database administrators (DBAs) who desire more direct control, or in situations where a database is being moved to a new hardware server in conjunction with their database upgrade.

Starting with Oracle Database 12c, the Pre-Upgrade Information Tool (`preupgrd.sql`) automatically generates fixup scripts to address common issues that may prevent an upgrade from being successful. The post-upgrade phase has also been enhanced to automate the running of many post-upgrade steps.

Example of Using Command-line Upgrade

The process of upgrading your Oracle database using command-line upgrade can be broken into three phases, each of which has a small number of steps to follow for a successful upgrade. Starting with Oracle Database 12c, you can then plug your database into a CDB:

- » Pre-upgrade Phase
 1. Run the new Pre-Upgrade Information Tool (`preupgrade.jar`), which validates the readiness of your database to be upgraded
 2. Run the `preupgrade_fixups.sql` script to automatically address issues found by the Pre-Upgrade Information Tool
 3. Perform any manual fixup steps identified by the Pre-Upgrade Information Tool
- » Upgrade Phase
 1. Run the Parallel Upgrade Utility (`catctl.pl`)
- » Post-upgrade Phase
 1. Run the `postupgrade_fixups.sql` script to automatically fix any issues identified by the Pre-Upgrade Information Tool, which need to be addressed after the upgrade.

2. Review the log files generated by the Parallel Upgrade Utility
 3. Recompile invalid objects by running `utlrp.sql`
- » Plug into a CDB
1. Use the `DBMS_PDB.DESCRIBE()` procedure to generate an XML descriptor file for your database
 2. Connect to the CDB and issue the `CREATE PLUGGABLE DATABASE` command to migrate your non-CDB database into a PDB
 3. Run the `noncdb_to_pdb.sql` script to convert the non-CDB into a fully usable PDB

If your source database contains either encrypted tablespaces or tables with encrypted columns, then the keys associated with those tablespaces or tables must be moved from the source database into the new PDB using the export and import functions of the new Oracle Database 12c `ADMINISTER KEY MANAGEMENT SQL` statement.

These steps are an outline of the command-line upgrade process. Full details of command-line upgrade are explained in [Oracle Database Upgrade Guide](#).

Considerations for Using Command-line Upgrade

Many DBAs prefer the level of control that they get from typing commands at the command line, compared to using a GUI such as DBUA. Beyond personal preference, a common use case for command-line upgrade is a situation in which you are migrating to new hardware, but staying on the same operating system architecture.

The command-line upgrade allows the user to specify the number of parallel processes used for the database upgrade, thereby giving you the ability to balance upgrade speed and system resource utilization.

Starting with Oracle Database 12c Release 2, the command-line upgrade can automatically detect the point at which it was interrupted, and restart the upgrade, using the '-R' parameter. If command-line upgrade is interrupted for any reason, you can rerun or restart the Parallel Upgrade Utility as described in [Oracle Database Upgrade Guide](#).

Plugging a non-CDB Database into a CDB

After upgrading to Oracle Database 12c, you can plug the database into a CDB. Note that the source database and destination CDB must meet the following requirements:

- » The hardware and operating system platforms must have the same endianness.
- » They must have compatible database options installed.
- » They must have compatible character sets and national character sets. This means that either:
 - » The character set in the CDB is AL32UTF8, or
 - » The character set in the non-CDB is the same as the character set in the CDB, or
 - » The character set in the non-CDB is a binary subset of the character set in the CDB.

The procedure for creating a PDB from a non-CDB is described in [Oracle Database Administrator's Guide](#). Oracle recommends using the AL32UTF8 character set for CDBs.

Upgrading a CDB with Multiple PDBs

If you are upgrading an Oracle Database 12c container database, you can take advantage of the flexibility of the CDB architecture to approach the upgrade in one of two ways:

- » Upgrade the CDB and all PDBs with a single command
- » Upgrade one PDB, or a subset of the PDBs, at a time
 1. Create a new CDB using the new version of Oracle Database software

2. Unplug one or more PDBs from the source CDB and plug them into the new CDB
3. Upgrade the older PDB(s) using either DBUA or the command-line upgrade

There are tradeoffs with each technique. Some considerations for choosing an upgrade method for a CDB are as follows.

“All At Once” Upgrade

- » Less effort: upgrade up to 252 PDBs with a single command
- » Less flexible: requires common downtime for all application PDBs
- » Upgrade automation reduces overall downtime for the upgrade process
- » Downtime for individual PDBs will be longer
- » Retain original database SID for the CDB

Unplug/Plug/Upgrade

- » More effort: must unplug, plug and upgrade individual PDB or set of PDBs
- » More flexible: allows you to plan migration windows based on user needs
- » Easier migration to a new server
- » Downtime for individual PDB will be shorter
- » Uses a second CDB, which may require resource allocation if on the same server

The choice of an upgrade method for a CDB thus depends on your use case and upgrade or migration scenario, and methods can be combined. For example, a single PDB could be unplugged, plugged into a new CDB and upgraded while the rest of the PDBs are kept at the source version. Then the original CDB could be upgraded all at once, and the first (upgraded) PDB moved back to the original CDB using unplug and plug.

Method 2: Full Transportable Export/Import or Transportable Tablespaces

Transportable tablespaces allows you to copy a set of tablespaces from one database to another. This can be much faster than exporting and importing data from those tablespaces because the tablespaces are copied as physical files without the need to interpret the logical entities, such as rows or indexes, contained within those files. In addition to copying the tablespaces, metadata describing the objects within the source database must be moved to the new database by Data Pump export/import.

Transported tablespaces can be copied to another database that may be on a different operating system platform, or running a different version of Oracle Database software. This makes transportable tablespaces a relatively fast way to migrate and upgrade a database in a single operation. The tradeoff for this speed is that transportable tablespaces can be fairly complicated, because the user is responsible for moving metadata such as procedures, packages, constraints, and so on. Starting with Oracle Database 12c, the new full transportable export/import feature combines the speed of transportable tablespaces with a much easier process for transporting metadata and data from installed database options.

Full Transportable Export/Import

Full transportable export/import is a new feature of Oracle Database 12c that makes it easy to move an entire database using the transportable tablespaces feature. It automates the process of moving metadata, and can move data that resides in non-transportable tablespaces such as SYSAUX and SYSTEM. In addition, full transportable export/import can transport encrypted tablespaces.

Full transportable export/import moves metadata using either dump files or over a database link. This combination of speed and simplicity makes full transportable export/import a good choice for many migration scenarios. The detailed steps for using full transportable export/import are described in the Oracle white paper, [“Oracle Database 12c: Full Transportable Export/Import.”](#)

"WITH ORACLE DATABASE 12C, WE CAN NOW COPY AN ENTIRE DATABASE FROM ONE INSTANCE TO ANOTHER USING FULL TRANSPORTABLE EXPORT/IMPORT. THIS MINIMIZES THE DOWNTIME WHEN MIGRATING CLIENTS' DATABASES."

JULIAN DONTCHEFF, HEAD OF DATABASE MANAGEMENT, ACCENTURE

Migrating to a Pluggable Database Using Full Transportable Export/Import

Because full transportable export/import allows you to migrate an entire database to both a new operating system platform and a new release of Oracle Database in a single operation, it is a very useful method for migrating to a PDB. The procedure for using full transportable export/import to migrate to a PDB is the same as for migrating into a non-CDB.

Examples of Using Full Transportable Export/Import

The complete implementation of the full transportable export/import feature is included in Oracle Database 12c. In addition, export-side support for this feature is available starting in Oracle Database 11g Release 2 (11.2.0.3).

In these examples, assume that you start with an 11.2.0.3 source database containing the following user tablespaces:

TABLE 3. SOURCE DATABASE TABLESPACES

TABLESPACE NAME	ENCRYPTED?	Datafile Name
HR	Yes	/data3/oracle/dbs/hr_1.f
ENGTABLES	No	/data4/oracle/dbs/eng_1.f
ENGINEXES	No	/data4/oracle/dbs/eng_2.f

All three of these tablespaces, as well as all of the metadata such as procedures, packages, constraints, triggers, and so on, will be moved from the source to the destination in a single operation.

The first example is a dumpfile-based full transportable export/import operation. In this case the metadata from the source database is exported to a dump file, and both the dump file and the tablespace data files are transferred to a new system. The steps would be as follows:

1. Set user tablespaces in the source database to READ ONLY.
2. From the Oracle Database 11g Release 2 (11.2.0.3) environment, export the metadata and any data residing in administrative tablespaces from the source database using the FULL=Y and TRANSPORTABLE=ALWAYS parameters. Note that the VERSION=12 parameter is required only when exporting from an Oracle Database 11g Release 2 database:

```
expdp src112admin/<passwd>@src112 DIRECTORY=src112_dir
DUMPFILE=src112fulltts.dmp VERSION=12 FULL=Y
TRANSPORTABLE=ALWAYS
EXCLUDE=TABLE_STATISTICS, INDEX_STATISTICS
```

One advantage of full transportable export/import is that it can transport encrypted tablespaces or tables with encrypted columns to a new database.

```
ENCRYPTION_PASSWORD=<enc_passwd> METRICS=Y
LOGFILE=src112fullttsexp.log
```

3. Copy the tablespace data files from the source system to the destination system. Note that the log file from the export operation will list the data files required to be moved.
4. Create a CDB on the destination system, including a PDB into which you will import the source database. Administration of PDBs is described in [Oracle Database Administrator's Guide](#).
5. In the Oracle Database 12c environment, connect to the pre-created PDB and import the dump file. The act of importing the dump file will plug the tablespace data files into the destination PDB:

```
impdp pdbadmin/<passwd>@tgtpdb DIRECTORY=src112_dir
      DUMPFILE=src112fulltts.dmp
      ENCRYPTION_PASSWORD=<enc_passwd>
      METRICS=Y LOGFILE=src112fullttsimp.log
      TRANSPORT_DATAFILES='/recovery1/data/hr_1.f '
      TRANSPORT_DATAFILES='/recovery1/data/eng_1.f '
      TRANSPORT_DATAFILES='/recovery1/data/eng_2.f '
```

The second example of using full transportable export/import employs the network mode of Data Pump to eliminate the need for a dumpfile. In this case, we will assume that the tablespace data files are in a location, such as a Storage Area Network (SAN) device, which is accessible to both the source and destination systems. This enables you to migrate from a non-CDB into a PDB with one Data Pump command:

1. Create a CDB on the destination system, including a PDB into which you will import the source database.
2. Create a database link in the destination PDB, pointing to the source database.
3. Set user tablespaces in the source database to READ ONLY.
4. In the Oracle Database 12c environment, import directly from the source database into the destination PDB using full transportable export/import in network mode:

```
impdp pdbadmin/<passwd>@tgtpdb NETWORK_LINK=src112 VERSION=12
      FULL=Y TRANSPORTABLE=ALWAYS
      EXCLUDE=TABLE_STATISTICS,INDEX_STATISTICS
      ENCRYPTION_PASSWORD=<enc_passwd>
      METRICS=Y LOGFILE=tgtpdb_dir:src112fullimp.log
      TRANSPORT_DATAFILES='/recovery1/data/hr_1.f '
      TRANSPORT_DATAFILES='/recovery1/data/eng_1.f '
      TRANSPORT_DATAFILES='/recovery1/data/eng_2.f '
```

Whether you use conventional dump files or network mode, full transportable export/import is a convenient way to upgrade a database to a new version, migrate to a different operating system or hardware platform, migrate into a PDB – or even to perform all three of these upgrades and migrations in a single operation!


Considerations for Using Full Transportable Export/Import

Full transportable export/import is subject to the limitations detailed in [Oracle Database Administrator's Guide](#).

Full transportable export/import can be used to migrate source databases starting with Oracle Database 11g Release 2 (11.2.0.3).

Full transportable export/import jobs cannot be restarted. If the operation is interrupted, then the entire job must start over from the beginning.

For complete documentation of full transportable export/import, including details about network mode migration into a PDB, see [Oracle Database Utilities Guide](#).



If hardware and operating system platforms of your source and destination databases have different endian characteristics, you will need to use the RMAN CONVERT command to convert each tablespace to the new platform. See *Oracle Database Backup and Recovery Reference* for a description of the RMAN CONVERT command.

Transportable Tablespaces

The transportable tablespaces feature can be used to copy a single tablespace or a set of tablespaces to a new database. While this method of migrating data is both fast and reliable, it requires a number of manual steps that can be more complicated than some DBAs would prefer.

Migrating to a Pluggable Database Using Transportable Tablespaces

Like full transportable export/import, transportable tablespaces can be used to migrate to a PDB. The procedure for using transportable tablespaces to migrate a database to a PDB is the same as a transportable tablespaces migration to a non-CDB.

Example of Using Transportable Tablespaces

Oracle recommends using full transportable export/import to migrate databases starting with Oracle Database 11g Release 2 (11.2.0.3). However, you can still use the transportable tablespaces feature to migrate databases from earlier releases. The use of the transportable tablespaces feature is described in detail by the following white papers:

- » [“Database Upgrade Using Transportable Tablespaces”](#)
- » [“Platform Migration Using Transportable Tablespaces”](#)

The steps for using this feature remain unchanged for Oracle Database 12c.

Considerations for Using Transportable Tablespaces

Transporting tablespaces between databases is subject to the limitations detailed in [Oracle Database Administrator's Guide](#).

Transportable tablespace export/import jobs cannot be restarted. If the operation is interrupted, then the entire job must start over from the beginning.

Transportable tablespaces can be used on databases starting with Oracle8i Database. This feature gained the ability to migrate cross-platform starting with Oracle Database 10g (10.1.0.3).

If hardware and operating system platforms of your source and destination databases have different endian characteristics, you will need to use the RMAN CONVERT command to convert each tablespace to the new platform. See [Oracle Database Backup and Recovery Reference](#) for a description of the RMAN CONVERT command.

Method 3: Oracle Data Pump Export/Import

Oracle Data Pump provides high-speed movement of data and metadata within and between Oracle databases. Because they are extremely flexible and easy to use, the Oracle Data Pump export (`expdp`) and import (`impdp`) utilities are commonly used to migrate tables, schemas, and databases to new hardware servers, to different operating system platforms, and to new releases of Oracle Database software.

Oracle Data Pump can write data to dump files on disk, or it can transfer data from the source database to the destination directly over the network. When the data is imported, it can be transformed to match the characteristics of the destination database. Some interesting ways in which a database can be transformed upon

import include migrating to a new character set, implementing encryption or compression, changing BasicFiles LOBs to SecureFiles LOBs, or changing the partitioning of tables in the database.

Data Pump Export/Import with Dump Files

Exporting data to a dump file has been the most common way to move data between databases for many years. Data Pump export chooses the best available method to extract data from the source database into flat files, and Data Pump import makes a similar decision to read from dump files and insert data into the destination database.

The biggest advantage of exporting to dump files is that you retain a persistent copy of the data on disk. This allows you to reuse the dump files for multiple imports, which can be particularly helpful when you need to test or tune your import or your source database. Another case in which exporting to dump files can be helpful is when the network between the source and destinations is relatively slow. In such cases it may be faster to physically disconnect a disk from the source system and connect it to the destination, compared to copying data over the network.

Oracle Data Pump Network Mode

Instead of exporting the source database to dump files and then importing from those same dump files into the destination database, you may choose to migrate your database using Data Pump import over a network link. This eliminates the need to store, manage, and transfer dump files. Instead, data is extracted from the source database and inserted directly into the destination database over a database link.

In addition to the reduced need for file storage and management, network mode simplifies your migration by letting you migrate cross-platform and to a new release of Oracle in a single step. Data Pump network mode jobs are fully restartable.

Migrating to a Pluggable Database Using Oracle Data Pump

Oracle Data Pump export/import is an easy and flexible way to migrate a database into a PDB. This is a particularly useful method to use when your source database resides on a different operating system, uses a different character set, or otherwise needs to be modified or reconfigured during the migration process.

Example of Using Oracle Data Pump


The steps for using Oracle Data Pump to upgrade or migrate a database have not changed from previous releases. If you would like to migrate with Oracle Data Pump using conventional dump files, the steps would be as follows:

1. Export the source database to a dump file:

```
expdp srcadmin/<passwd> DIRECTORY=src_dir
      DUMPFILE=srcfull.dmp LOGFILE=srcfullexp.log
      FULL=Y METRICS=Y PARALLEL=4
      EXCLUDE=TABLE_STATISTICS,INDEX_STATISTICS
```
2. Make the dump file available on the destination system, either by copying it to the destination system or placing it on a network-mounted disk.
3. Create a CDB on the destination system, including a PDB into which you will import the source database.
4. In the destination PDB, create a directory object for the dump file to be imported.
5. Import the Data Pump dump file into the destination PDB:

```
impdp pdbadmin/<passwd>@pdb DIRECTORY=src_dir
      DUMPFILE=srcfull.dmp LOGFILE=srcfullimp.log
      PARALLEL=4 METRICS=Y
```

Oracle Data Pump is also the command-line interface for the full transportable export/import Feature. full transportable export/import can move data faster than Data Pump export, but Data Pump export has fewer restrictions and is more flexible in handling data layout and formats.



In this example, we take advantage of Oracle Data Pump features such as the `PARALLEL` parameter to improve performance. In other cases, you might use network mode to eliminate the need for a dump file. Oracle Data Pump is fully documented in [Oracle Database Utilities Guide](#).

In Oracle Database 12c Release 2, Data Pump adds the ability to export and import metadata in parallel. This greatly speeds up migration of metadata-heavy databases.

Considerations for Using Oracle Data Pump

Oracle Data Pump is available starting with Oracle Database 10g.

Oracle Data Pump does not export objects (including GRANT objects) from the SYS schema. If you have user objects in the SYS schema, such as user GRANTs on SYS-owned objects, these must be reproduced separately in the target database.

Oracle Data Pump network mode is subject to limitations on network links, as described in [Oracle Database Utilities Guide](#).

Starting with Oracle Database 11g Release 2 (11.2.0.3), you can set the parameter `VERSION=12` when exporting to a dump file. This will specify that all data from registered database options and components should be included in the export. A dump file produced with the setting `VERSION=12` can be imported starting with Oracle Database 12c.

Method 4: Original Export/Import

Oracle recommends that you use the Data Pump export and import utilities to move data between Oracle databases. However, the original Export (`exp`) and Import (`imp`) may be useful when upgrading or migrating older databases to Oracle Database 12c. For example, you may want to migrate from Oracle9i Database to Oracle Database 12c on a different operating system platform. Because Oracle Data Pump is available starting with Oracle Database 10g, you would use original Export for this operation.

Migrating to a Pluggable Database Using Original Export/Import

The original Import utility is still fully supported specifically for the purpose of migrating older Export dump files into newer versions of Oracle Database. You can export an older database using the original Export utility, and then import that database into a PDB using the original Import utility. This will allow you to consolidate databases from old and even obsolete platforms into a CDB.

Example of Using Original Export/Import

Oracle9i Database and earlier releases can be migrated to Oracle Database 12c using original Export/Import. The steps for such a migration are:

1. Export the source database to a dump file:

```
exp src92admin/<passwd> full=y file=exp9i.dmp
```
2. Make the dump file available on the destination system, either by copying it to the new system or placing it on a network-mounted disk.
3. Create a CDB on the destination system, including a PDB into which you will import the source database.
4. Import the original Export dump file into the destination PDB:

```
imp pdbadmin/<passwd> full=y file=exp9i.dmp
```

While the steps for using original Export/Import to upgrade and migrate into a CDB are conceptually similar to those for using Data Pump Export/Import, there are several practical differences between the original Export/Import and Data Pump Export/Import utilities:

- » Data Pump Export/Import is generally much faster than original Export/Import, especially on large volumes of data
- » Data Pump Export/Import supports all data types and features in your database. Original Export/Import may not support export of data types or features added in Oracle Database 10g Release 2 and later releases
- » Original Export/Import may not be able to migrate data from installed options in your database

Therefore, Oracle recommends that you use Oracle Data Pump Export/Import rather than original Export/Import unless you are migrating a database from Oracle Database 9i or earlier.

Considerations for Using Original Export/Import

Original Export is desupported for general use as of Oracle Database 11g. This method should only be used for migrating pre-11g databases.

Migrating a database using Original Export/Import can be quite time consuming, especially if you have more than 100GB of data.

Minimal Downtime Database Upgrades

Many customers are interested in maximizing database availability, particularly during planned maintenance such as a database upgrade. In addition to choosing the fastest upgrade or migration method that can be applied to your upgrade scenario, there are techniques that can be combined with these upgrade and migration methods to further minimize downtime and maximize system availability.


Rolling Upgrade with Oracle Data Guard

Oracle Data Guard provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases to enable production Oracle databases to survive disasters and data corruptions. This feature is often used primarily as a disaster recovery solution, but Data Guard can also provide the advantage of rolling upgrade for Oracle Database patch sets and full releases. There are two ways in which Oracle Data Guard can be used to minimize the downtime associated with a database upgrade.

Starting with Oracle Database 10g (10.1.0.3), you can use a logical standby database to perform a rolling upgrade. During a rolling upgrade, you can run different releases of an Oracle database on the primary and logical standby database while you upgrade them one at a time, incurring minimal downtime on the primary database. The process for using SQL Apply to perform a rolling upgrade of your database is described in [Oracle Data Guard Concepts and Administration](#).

Starting with Oracle Database 11g, you can perform an upgrade using a physical standby database as the starting point for a rolling upgrade. The process for using a physical standby to minimize downtime for a database upgrade is called Transient Logical Standby. It is described in the white paper, "[Database Rolling Upgrades Made Easy by Using a Physical Standby Database](#)." Transient Logical Standby is available starting with Oracle Database 11g.

Transient Logical Standby is also very helpful for reducing downtime when migrating your database to a new hardware system. If you are staying with the same operating system platform, or if you are migrating between



supported combinations of platforms (see My Oracle Support note [413484.1](#)), then Transient Logical Standby can maximize your system availability for planned maintenance including database upgrade or migration.

In order to use Data Guard Rolling Upgrade to migrate to Oracle Database 12c Release 2, the source database version must be Oracle Database 11g Release 2 (11.2.0.3) or later.

RMAN Cross-Platform Incremental Backups

Recovery Manager (RMAN) is an Oracle Database client that performs backup and recovery tasks on your databases and automates administration of your backup strategies. It greatly simplifies backing up, restoring, and recovering database files. RMAN can help minimize downtime during some types of database migrations.

When you migrate a very large database to a new hardware system, the time needed to physically move data can often be the most time-consuming aspect of the migration. This is particularly true when you use full transportable export/import or the transportable tablespaces feature. Even though the data is not exported and imported by Data Pump, your tablespace data files must still be moved over the network from the source system to the destination system. Simply copying terabytes of data can require several hours, and your production database must be read only while the copy takes place.

Starting with migrations to Oracle Database 11g Release 2 (11.2.0.4), the RMAN Cross-Platform Incremental Backups feature is available to reduce the downtime associated with a transportable tablespace migration to a supported Linux platform. The process for using Cross-Platform Incremental Backups is described in My Oracle Support note [1389592.1](#). This can be used to reduce downtime for a whole database migration, or for moving a set of one or more individual tablespaces.

Oracle GoldenGate


Oracle GoldenGate is a high-performance software application for real-time heterogeneous data replication and transformation. Oracle GoldenGate can be combined with any of the migration methods described in this white paper to deliver a minimal downtime database upgrade and migration. Oracle GoldenGate is extremely flexible, and can be used to minimize downtime when migrating to a new operating system platform, upgrading to a new release of Oracle Database, migrating to a new character set, or any combination of those changes. The use of Oracle GoldenGate in a database upgrade or migration project is described by the white paper, "[Zero-Downtime Database Upgrades Using Oracle GoldenGate](#)."

Conclusion

This white paper provides an overview of the different tools, techniques, and utilities provided by Oracle to help you upgrade and migrate to Oracle Database 12c Release 2, as well as guidance and information to help you choose among the various upgrade and migration methods available. The best method to use for your migration scenario depends on the source database version, the source and destination operating systems, your downtime requirements, and the personal preference of the DBA. Based on these factors, there is a method available that is the best fit for your migration scenario.

Upgrading or migrating to Oracle Database 12c Release 2 can bring many benefits to your organization. You can take advantage of new features to improve performance, enhance security, and expand functionality. You can modernize your IT infrastructure. And, you can improve operational efficiency to reduce costs and raise productivity.

If you would like to achieve the minimum possible downtime for your database upgrade or migration, you can start by choosing the fastest upgrade or migration method based on your source and destination platform and



Oracle Database release. You can then combine your chosen upgrade or migration method with minimal downtime features and products to maximize system availability.

"OUR EXPERIENCE WITH UPGRADE HAS BEEN VERY POSITIVE, UPGRADING TO 12C, AND WE HAVE BEEN VERY IMPRESSED WITH THE ROBUSTNESS OF 12C."

NICK SALEM, **DISTINGUISHED ENGINEER, NEUSTAR, INC.**

Appendix A: Upgrade & Migration New Features in Oracle Database 12c (12.1)

TABLE 4: UPGRADE AND MIGRATION NEW FEATURES IN ORACLE DATABASE 12C

Feature	New Capability
Parallel Processing for Database Upgrade	<p>Since Oracle Database 12c, the <code>catupgrd.sql</code> Upgrade Utility has been replaced with the new <code>catct1.pl</code> 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.</p> <p>Database Upgrade Assistant (DBUA) takes advantage of the new Parallel Upgrade Utility, but does not give the user any control over the degree of parallelism used.</p>
Upgrade a CDB with one command	<p>You can upgrade a CDB with multiple PDBs using a single command-line or DBUA upgrade. This can greatly reduce the effort needed to upgrade multiple databases.</p>
Unplug, Plug and Upgrade	<p>In a CDB environment, you can unplug a PDB from one CDB, plug into a newer version CDB, and upgrade the PDB to the new version. This gives you the flexibility to move and upgrade PDBs between different servers and containers.</p>
Full Transportable Export and Import	<p>Introduced for Oracle Database 12c, this feature combines the ease of use of Data Pump with the speed of Transportable Tablespaces.</p>
Data Pump Supports CDBs	<p>You can use Data Pump to migrate all, or portions of, a database from a non-CDB into a PDB, between PDBs within the same or different CDBs, and from a PDB into a non-CDB.</p>
Data Pump Allows You to Change the COMPRESSION or LOB_STORAGE for All Tables During Import	<p>During import jobs, you can now change the compression type for all tables in the job, including tables that provide storage for materialized views. This is done by specifying a <code>TABLE_COMPRESSION_CLAUSE</code> using the <code>TRANSFORM</code> parameter. Also during import jobs, you can now change the LOB storage (either <code>SECUREFILE</code> or <code>BASICFILE</code>) for all tables in the job, including tables that provide storage for materialized views. This is done by specifying a <code>LOB_STORAGE</code> option using the <code>TRANSFORM</code> parameter.</p>
Oracle XML Database is Installed When You Upgrade	<p>Starting with Oracle Database 12c, Oracle XML Database (Oracle XML DB) is 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 <code>SYSAUX</code> tablespace. The upgraded Oracle database includes Oracle XML DB.</p>

Appendix B: Upgrade & Migration New Features in Oracle Database 12c Release 2 (12.2)

TABLE 5: UPGRADE AND MIGRATION NEW FEATURES IN ORACLE DATABASE 12C RELEASE 2

Feature	New Capability
New Pre-Upgrade Script	A new Pre-Upgrade Information Tool has been created for this release. The <code>preupgrade.jar</code> file replaces the <code>preupgrd.sql</code> script. 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.
Automatically detect restart point	If your upgrade is interrupted, restarting the upgrade with the 'R' parameter will automatically detect the correct point at which to begin processing the upgrade for another run.
Parallel Processing for Upgrade of Multimedia and XDB	Starting with Oracle Database 12c Release 2, parallel processing is used for the Spatial, Multimedia, and XDB components in addition to the parallel processing added in Oracle Database 12c for the core RDBMS Server component.
Upgrade a CDB with one command	You can upgrade a CDB with multiple PDBs using a single command-line or DBUA upgrade. This can greatly reduce the effort needed to upgrade multiple databases.
Data Pump parallel export and import of metadata	Starting with Oracle Database 12c Release 2, Data Pump can import metadata in parallel. This can greatly speed up the metadata-processing portion of your data pump migrations.





Oracle Corporation, World Headquarters

500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries

Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

-  blogs.oracle.com/upgrade
-  oracle.com/goto/upgrade

Integrated Cloud Applications & Platform Services

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. 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 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. 0116

Upgrading and Migrating to Oracle Database 12c Release 2 (12.2.0.1)
November 2016
Author: Roy F. Swonger
Contributing Authors: Carol Tagliaferri, Mike Dietrich, Jim Stenoish

