Upgrade Methods

Objectives

- After completing this lesson, you should be able to:
  - Upgrade CDBs from 12c to 19c
  - Upgrade regular PDBs from 12c to 19c
  - Plug in a remote PDB into a target CDB by using RMAN

Upgrading CDB and PDBs to 12c: Methods

- Data Pump Export / Import
  - Can provide better performance depending on data volume, metadata volume
  - Ensures support for new data types
- Database Upgrade Assistant (DBUA)
  - Interactively steps you through the upgrade process
  - Automatically fixes some configuration settings
  - Provides a list of items to fix manually
  - Upgrades the CDB, including all PDBs or a defined list of PDBs
- Manual upgrade
  - Provides finer control over the upgrade process
  - Provides a list of items to fix manually
  - Upgrades the CDB, including all PDBs or a defined list of PDBs

Upgrading a CDB Including PDBs from 12c to 19c

1. Install the 19c Oracle Database software.
2. Execute the Pre-Upgrade Information Tool in the 12c CDB.

```
$ cd /u01/app/oracle/product/18.1.0/dbhome_1/rdbms/admin
$ $ORACLE_HOME/jdk/bin/java -jar preupgrade.jar
```

3. Back up the CDB.
4. Execute the `preupgrade_fixups.sql` script on the 12c CDB.

```
$ORACLE_HOME/perl/bin/perl -I$ORACLE_HOME/perl/lib -I$ORACLE_HOME/rdbms/admin
$ORACLE_HOME/rdbms/admin/catcon.pl -l /u01/app/oracle/cfgtoollogs/cdb12/preupgrade/ -b
preup_cdb12 /u01/app/oracle/cfgtoollogs/cdb12/preupgrade/preupgrade_fixups.sql
```

6. Copy the 12c instance spfile to the 19c `$ORACLE_HOME/dbs` directory.
7. Adjust the parameter file with the Oracle Database 19c parameters.

Upgrading CDB Including PDBs from 12c to 19c

8. Start the CDB and all PDBs in UPGRADE mode in the 19c environment.

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

9. Execute the upgrade script on the CDB root and all PDBs.

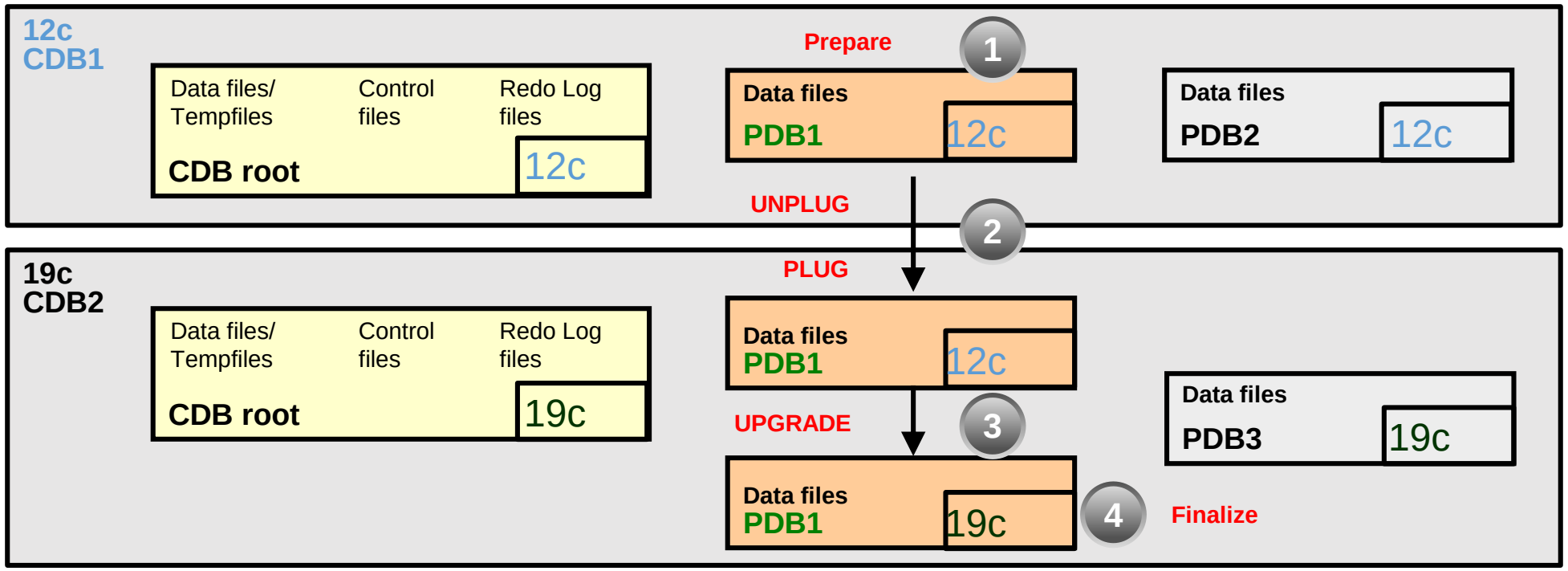
```
$ cd $ORACLE_HOME/rdbms/admin
$ ./catctl.pl [ -C 'PDB1 PDB2' ] [ -l /tmp] catupgrd.sql
```

10. Open the CDB and upgraded PDBs in normal mode.
11. Execute the `postupgrade_fixups.sql` script.

```
$ cd /u01/app/oracle/product/18.1.0/dbhome_1/rdbms/admin
$ $ORACLE_HOME/perl/bin/perl catcon.pl -c PDB1 -b postupgrade
    $ORACLE_BASE/cfgtoollogs/cdb12/preupgrade/postupgrade_fixups.sql
```

12. Shut the instance down to update the `/etc/oratab` file and create the password file.

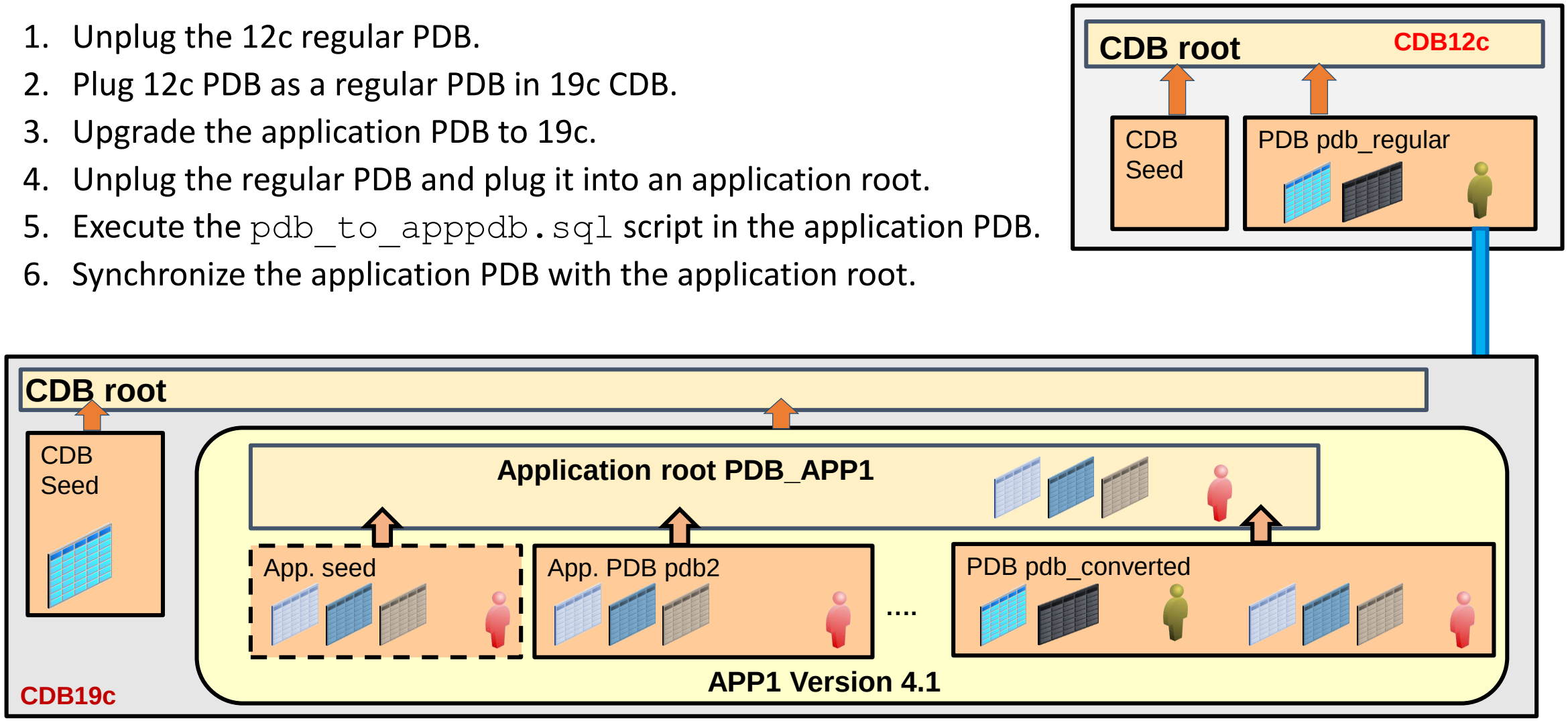
Upgrading a Single Regular PDB from 12c to 19c



1. Execute `preupgrade.jar` and then the generated `preupgrade_fixups.sql` script in the 12c PDB.
2. Unplug the PDB from the 12c CDB and plug the PDB into the 19c CDB.
3. Open the PDB in `UPGRADE` mode and upgrade the PDB.
4. Finalize by executing the `postupgrade_fixups.sql` script.

Converting and Upgrading Regular PDBs to Application PDBs

1. Unplug the 12c regular PDB.
2. Plug 12c PDB as a regular PDB in 19c CDB.
3. Upgrade the application PDB to 19c.
4. Unplug the regular PDB and plug it into an application root.
5. Execute the `pdb_to_apppdb.sql` script in the application PDB.
6. Synchronize the application PDB with the application root.



Practice 13: Overview

- 13-1: Upgrading and converting a 12c regular PDB to a 19c application PDB

Note: *While the PDB upgrade (practice 13-1) is taking place:*

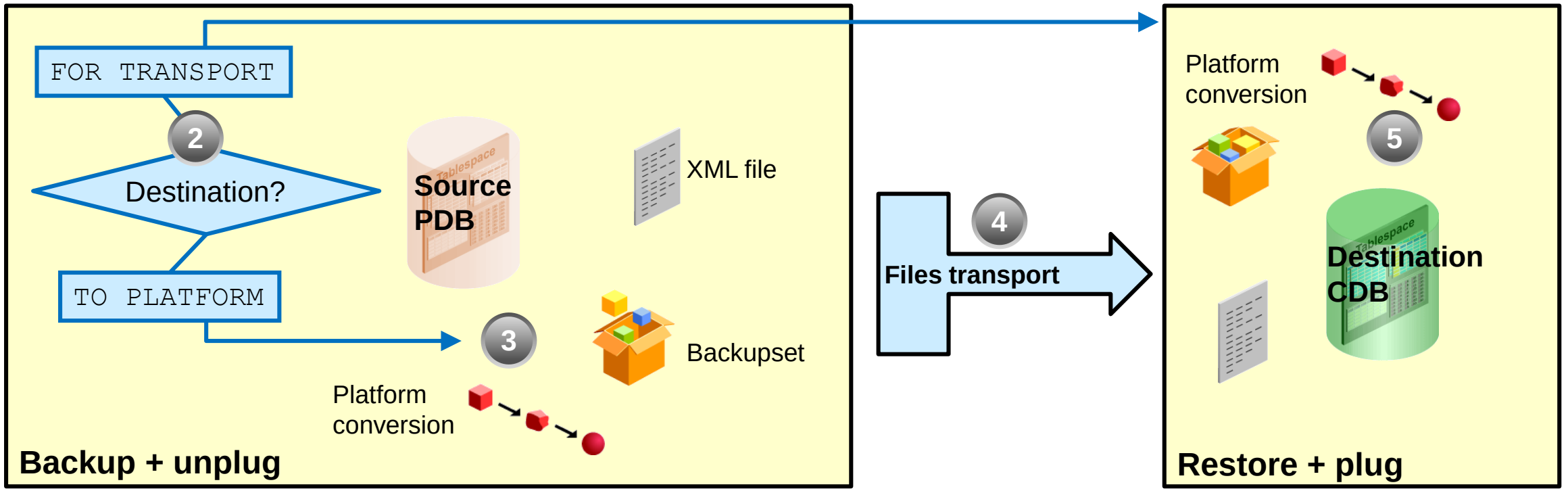
1. *The end of this lesson and also lesson 14 can be taught.*
2. *Then practice 13-2 will be completed and practice 13-3 will be started.*

Cross-Platform Transportable PDB

Prerequisites

- Source and target platforms have same endianess
- `COMPATIBLE=12.2` (or greater)
- Closed PDB for cross-platform PDB backup

1



Cross-Platform PDB Transport: Phase 1

1. Verify the prerequisites:
   - `COMPATIBLE`: Greater or equal to 12.2
   - `OPEN_MODE`: `MOUNTED`
2. Start an RMAN session to connect to the CDB of the PDB.
3. Query the exact name of the destination platform from the `V$TRANSPORTABLE_PLATFORM` view.
4. Back up the source PDB, including the XML file (*metadata*):
   - Conversion on the source host

```
RMAN> BACKUP TO PLATFORM 'Linux x86 64-bit'
      UNPLUG INTO '/tmp/pdb2.xml' PLUGGABLE DATABASE pdb1
      FORMAT '/bkp_dir/transport_%U';
```

   - Conversion at the destination host

```
RMAN> BACKUP FOR TRANSPORT UNPLUG INTO '/tmp/pdb2.xml'
      PLUGGABLE DATABASE pdb1 FORMAT '/bkp_dir/transport_%U';
```

Cross-Platform PDB Transport: Phase 2

Destination CDB

5. Disconnect from the source CDB.
6. Move the backup sets and XML file to destination host.
7. Start an RMAN session to connect to the new target CDB.
8. Restore the full backup set to create the new PDB with the `RESTORE` command by using the XML file.
   - When the conversion occurs on the source host

```
RMAN> RESTORE USING '/tmp/pdb2.xml'  FOREIGN PLUGGABLE DATABASE pdb1 TO NEW
      FROM BACKUPSET '/bkp_dir/transport_0gqoejqv_1_1';
```

   - When the conversion occurs at the destination host

```
RMAN> ALTER SYSTEM SET DB_CREATE_FILE_DEST='/oradata/new_pdb';
RMAN> RESTORE FROM PLATFORM 'Linux x86 64-bit' USING '/tmp/pdb2.xml'
      FOREIGN PLUGGABLE DATABASE pdb1 TO NEW
      FROM BACKUPSET '/bkp_dir/transport_0gqoejqv_1_1';
```

Summary

- In this lesson, you should have learned how to:
  - Upgrade CDBs from 12c to 19c
  - Upgrade regular PDBs from 12c to 19c
  - Plug in a remote PDB into a target CDB by using RMAN

Practice 13: Overview

- 13.1: Upgrading a 12.2 regular PDB to an 19c application PDB
- 13-2: Plugging remote PDBs through XTTS
- 13-3: Upgrading a 12.2 CDB to an 19c CDB