Lessons Learned Importing non-CDB Databases into PDBs

DOUG SEWELL, DAVID KENT CONSULTING

WEDNESDAY OCTOBER 14, 13:30-14:30

SESSION ID 3.4

Introduction

Container databases are the new normal for the Oracle world.

Oracle has deprecated non-container (non-CDB) databases. Database version 20c desupports non-CDB databases.

If you haven’t converted your databases already, you will. This session will help you when you’re ready.

Agenda

1 CDBs, PDBs, and why you want them

2 Lessons learned using three methods to create PDBs from non-CDB databases

3 Tips and gotchas

David Kent Consulting

DK

ORACLE and Banner IT Services

ORACLE and Banner Experts

Real-world Practitioners. Solutions You Trust. Always..

My Background

32 years in higher ed system administration and database administration at Youngstown State University – originally for mainframe environment, then moved to Unix/Linux.

20 years with Unix/Linux, 17 years with Oracle, 15 years with Banner.

With David Kent Consulting since 2009, full time since my retirement from YSU at the end of 2015 – primarily database, jobsub, Banner 9, ODS, and system administration.

Part 1: CDBs, PDBs, and why you want them

Multitenant Architecture

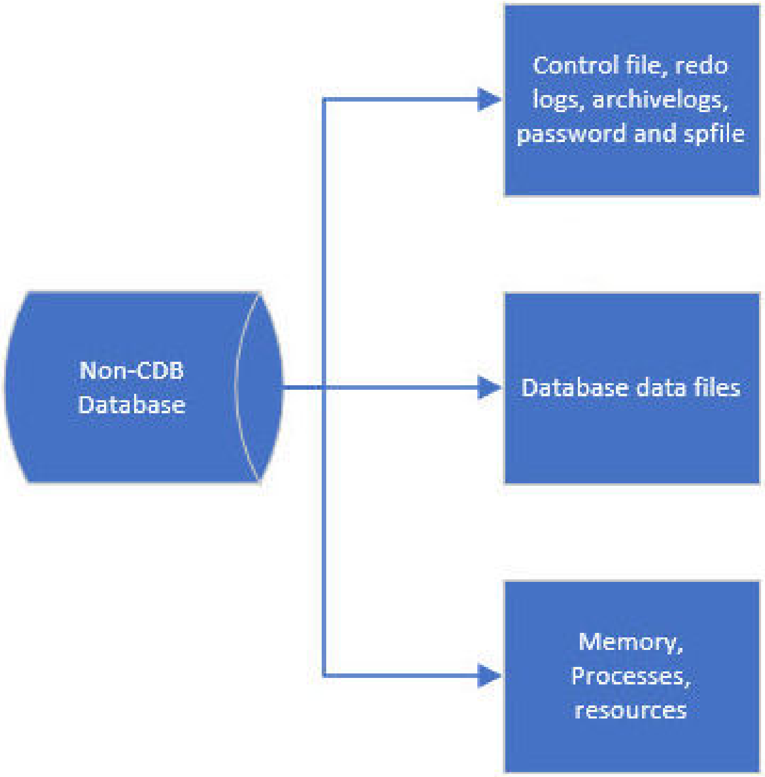
In Database version 12cR1, Oracle released Multitenant Architecture.

“The multitenant architecture enables an Oracle database to function as a multitenant container database (CDB).

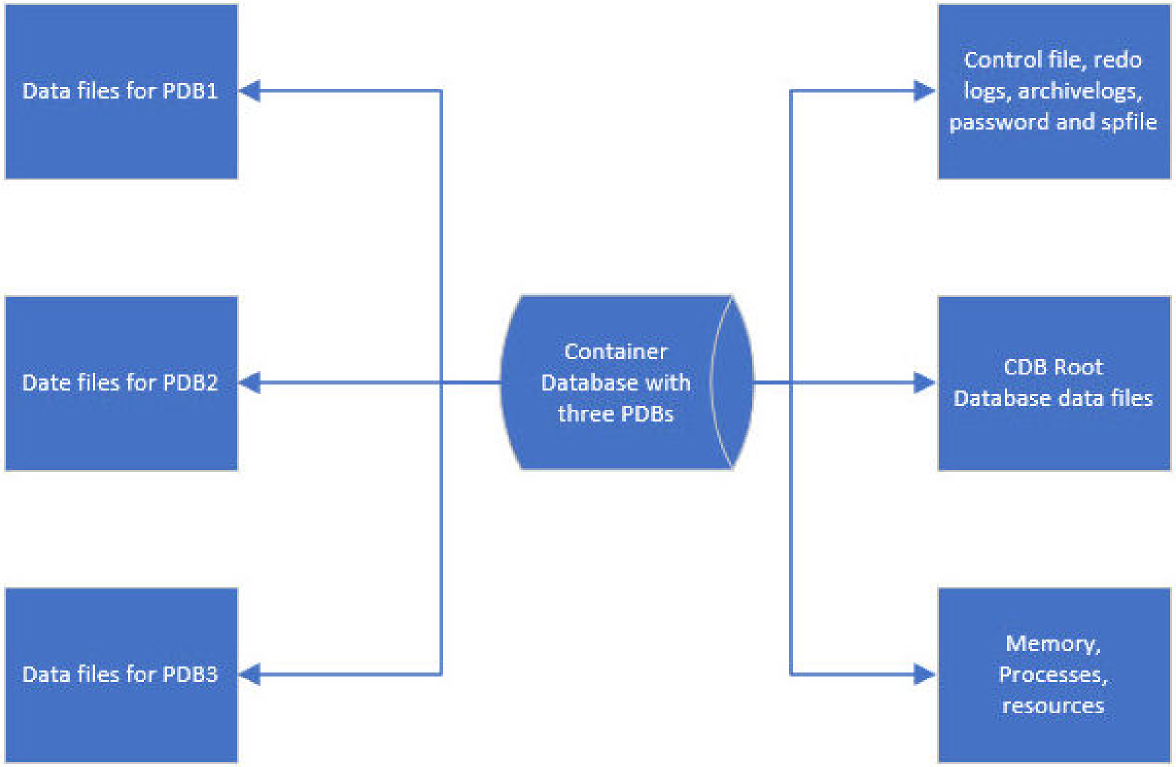
A CDB includes zero, one, or many customer-created pluggable databases (PDBs).” - Oracle 12cR1 Database Concepts guide

(note: 12cR2 introduced Application Containers, but I will ignore them in this presentation)

Non-container Database


Non-CDB Database
Control file, redo logs, archivelogs, password and spfile
Database data files
Memory, Processes, resources

Container Database



Data files for PDB1

Date files for PDB2

Data files for PDB3

Container Database with three PDBs

Control file, redo logs, archivelogs, password and spfile

CDB Root Database data files

Memory, Processes, resources

Benefits

- Required by Oracle starting with 20c
- Supported by Ellucian (in single-tenant configuration)
- Largely transparent to applications
- Simplified cloning
- Potential overhead savings for non-Banner databases

Challenges

- Learning new technology
- There may be changes required to set up connection
- Scripts and operational practices will change
- TDE (Transparent Data Encryption) is anything but transparent during setup

CoHEsion

Part 2:
Lessons learned using three methods to create PDBs from non-CDB databases

Three Paths to Same Result

Solutions I've used to migrate Banner and DegreeWorks databases

- Datapump export and import
- DBMS_PDB and copying datafiles
- Cloning a non-CDB database into a PDB via database link

Datapump export and import

Datapump Export and Import

- Necessary if switching byte sex (endian-ness)
- You can perform an upgrade at the same time without DBUA
- You can restructure your tablespaces at the same time
- Usable when moving from on-site to cloud environment

Datapump Export and Import

- There’s preparation before import and cleanup afterward
- Ellucian article 000040979 – although dated – has helpful information and script
- This will encrypt your data in a TDE environment
- The biggest downside I found to this path is …

Oracle Application Express (APEX)

Why is APEX a problem?

Important Application Express information will not transport via expdp and impdp – the schemas are protected.

- You must use APEX Administration Services or APEXExport to export the APEX pieces
- You must install the appropriate APEX version in the target database before importing database data
- After running the database import and post-import cleanup, you must run the scripts generated by APEXExport or use Administration Services

Do you have an APEX problem?

Check if APEX is installed:

```
select comp_name, version, status from dba_registry
where comp_id = 'APEX’;
```

Note the version, as you will need to install the same or higher version in the target database (19c requires minimum APEX version 18.2).

Do you have an APEX problem?

Check if you have user workspaces:

```
select workspace, workspace_display_name from apex_workspaces;
```

- Oracle provides INTERNAL, COM.ORACLE.CUST.REPOSITORY, and COM.ORACLE.APEX.REPOSITORY
- Additional workspaces indicate APEX development happened at some point.

This slide left blank intentionally.

DBMS_PDB and copying datafiles

DBMS_PDB and copying datafiles

- Usable when moving from on-site to cloud environment
- Requires a database outage or using a cloned or restored database
- Source and target databases should be same version, OS, and RU level
- You will have the same number of data files in the PDB
- If you are copying datafiles into a database with TDE enabled, this won’t automatically encrypt your data

DBMS_PDB procedure

1. Start source database read only
2. Run dbms_pdb.describe and save generated XML file
3. Edit the XML file if necessary
4. Stop the database and copy data files, along with XML file

DBMS_PDB procedure

5. Check target database compatibility with DBMS_PDB
6. Create pluggable database PDBNAME using ‘/path/to/xml’
   copy
   file_name_convert=(‘/source/path/’,’/target/path/’);
7. Run noncdb_to_pdb
8. Turn TDE on (if appropriate) for the new PDB

This slide left blank intentionally.

Cloning into PDB via DB Link

Cloning into PDB via DB Link

- This method does not require a source database outage
- This method works best when the source and target servers are "close" network-wise.
- Source and target databases should be same version, OS, and RU level
- You will have the same number of data files in the PDB
- If you are cloning into a database with TDE enabled, this won't automatically encrypt your data
- The process of cloning PDBs between CDBs is similar

Cloning via DB Link procedure

1. create a database link in target database accessing source database
2. Create pluggable database PDBNAME from NON$CDB@DBLINK;
3. Run noncdb_to_pdb
4. Turn TDE on (if appropriate) for the new PDB

This slide left blank intentionally.

CoHEsion

Part 3: Tips and gotchas

Two important tips

1. Set up a “playground” with a couple of small databases to play and practice with. They don’t have to be full-size Banner databases.
2. Once you’re satisfied with what you’ve learned, practice with a copy of a Banner database, apart from any of your working instances.

Additional tips

- You switch between root (CDB$ROOT) and PDB via “alter session set container = ….” ;
- RMAN backups are similar, but you back up the CDB if it’s a container database.
- Don’t clone a CDB via RMAN. Drop the PDB and create over a DB link.
- Naming conventions are helpful – for example UPGR in UPGRCDB and TEST in TESTCDB. So why would FINAID use FINACDB vs FACDB?

Show con_name

- Database name for non-CDB databases
- CDB$ROOT for CDB root container
- Container name for a PDB

Show con_id

- 0 for non-CDB databases
- 1 for CDB root container
- >1 for a PDB

v$database

select name, created, cdb from v$database ;

- “name” and “created” are the same whether in non-CDB, CDB$ROOT, or a PDB container.
- Name is the database name (non-CDB) or container DB name (CDB).
- Created is the date the database was created (non-CDB or CDB).
- CDB = ‘YES’ for container database, ‘NO’ for non-CDB.

v$pdbs

```
select name, creation_time, con_id from v$pdbs ;
```

- Returns zero rows for a non-CDB database.
- In root container, returns one row per PDB (including PDB$SEED).
- Inside a PDB, returns one row for the current PDB.
- Note: in DB12cR1, v$pdbs does not have CREATION_TIME field.

sys_context()

select sys_context(‘USERENV’,’INSTANCE_NAME’) from dual;

- Returns database name (non-CDB) or container DB name (CDB).
- This may require code changes in your applications!

Select sys_context(‘USERENV’,’CON_NAME’) from dual;

- Returns database name for non-CDB.
- Returns CDB$ROOT for root container of CDB.
- Returns container name for PDB.

You can use a difference between the two values to your advantage!

Miscellaneous tips

- PDBs don’t use SIDs, you must use SERVICE_NAMEs.
  - tnsnames.ora
  - JDBC connection information in tomcat and other config files
  - ORDS configuration
  - PHP, Perl, and other code
- You don’t create additional service names in the spfile
  - Use dbms_service.create_service
  - Ellucian Article 000036987 is helpful here – particularly if ESM is being uncooperative.

Summary

- You want to containerize your Oracle databases – at some point, you won’t have a choice.
- There are several ways to do this, each with tradeoffs.
- Practice and test before you work with “real” databases.
- Review and update code that’s sensitive to database instance name.

Questions?

Thank You!

Doug Sewell

dmsewell@davidkentconsulting.com

SESSION ID 3.4