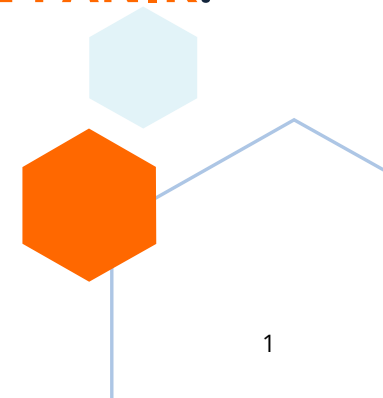


"In vielen der etwas lässigeren Zivilisationen am äußersten Rand der Galaxis hat der Reiseführer Per Anhalter durch die Galaxis die große Encyclopaedia Galactica als Standardwerk für alle Kenntnisse und Weisheiten inzwischen abgelöst. Denn obwohl er viele Lücken hat und viele Dinge enthält, die sehr zweifelhaft oder zumindest wahnsinnig ungenau sind, ist er dem älteren und viel langatmigeren Werk in zweierlei Hinsicht überlegen.

Erstens ist er ein bisschen billiger, und zweitens stehen auf seinem Umschlag in großen freundlichen Buchstaben die Worte **KEINE PANIK."**

(Douglas Adams, Per Anhalter durch die Galaxis)



Multitenant & Data Guard – was kann da schon schiefgehen?

Markus Flechtner

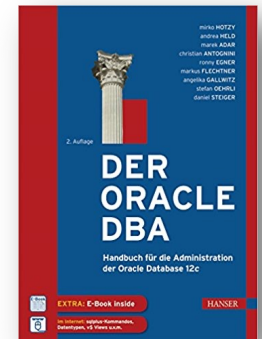
November 2025 - Nürnberg



Markus Flechtner

- Diplom-Mathematiker
- Oracle seit 1990
- Bisher: Entwickler, Oracle-Field-Support, DBA, Consultant
- Aktuell: Principal Consultant & Abteilungsleiter bei der ORDIX AG
- Schwerpunkte
 - Hochverfügbarkeit
 - Migrationen und Upgrades
 - Multitenant
-  @markusdba.bsky.social
-  @markusdba@mastodon.social
-  markusdba.de | .net  mfl@ordix.de

ORDIX®





ora2know

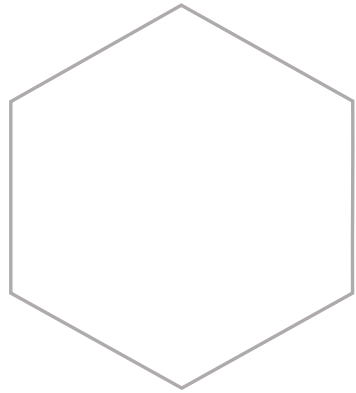
The German Oracle Database Community

noon2noon – KI für den DBA

Hotel Mercure Mitte
Hannover

10. – 11. Februar 2026



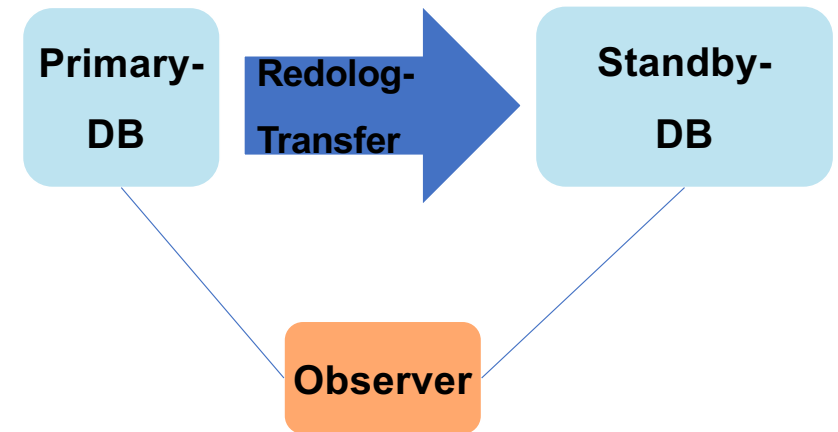


Agenda



Was ist Data Guard?

- Hochverfügbarkeits- und Disaster-Recovery-Lösung
- Änderungen aus einer (Primär-)Datenbank werden in eine oder mehrere Standby-Datenbanken synchronisiert
- Physische, logische und Snapshot-Standby-Datenbanken
- Automatischer Switchover und Failover
- Flexible Modi (Maximum Performance, Maximum Availability, Maximum Protection)
- Active Data Guard: (z.B.) Öffnen der Standby read-only zur Lastverteilung

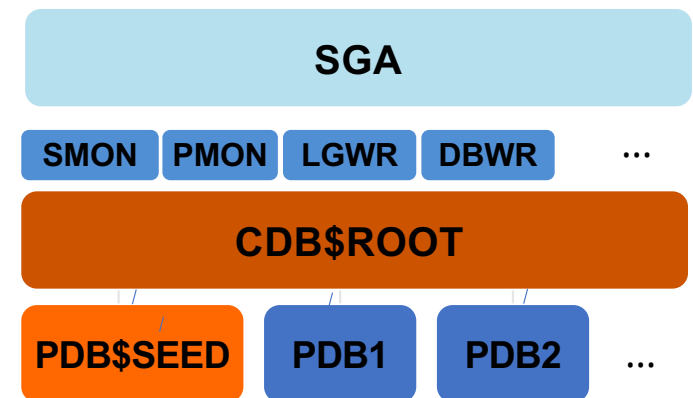


Historie: Data Guard

- verfügbar seit Oracle Database Version 8 (1997)
- Physische oder logische Standby-DB, Snapshot-Standby-DB
- Entwicklung der Features
 - 11g (2007) Active Data Guard, Fast Start Failover
 - 12c (2013) Unterstützung der Multitenant-Architektur, Far-Sync
 - 18c (2018) DML-Redirect
 - 19c (2019) Automatic Block Repair
 - 21c/23ai/26ai PDB-Level-Data Guard

Was ist Oracle Multitenant?

- "neue" Datenbank-Architektur seit Oracle 12c
- In einer Datenbank "Container-Datenbank" (CDB) gibt es eine oder mehrere Pluggable Datenbanken (PDBs)
 - Anwendungsdaten liegen in den PDBs
 - Max. 3 PDBs ohne Lizenz der "Multitenant Option"
- Redo- und Control-Dateien werden gemeinsam genutzt
- Vorteile
 - Konsolidierung
 - Einfache Bereitstellung
 - Bessere Ressourcen-Nutzung
 - "Manage many as one"

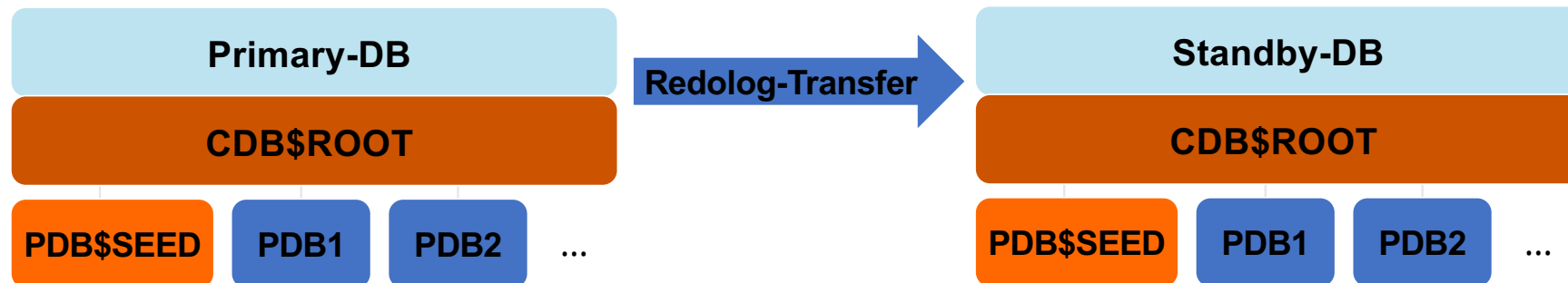


Historie: Multitenant

- verfügbar seit Oracle Database 12c (2013)
- Wesentliche Verbesserungen mit Oracle Database 12c Release 2 (12.2) (2017)
- Oracle Database 19c ist die letzte Version, die die Non-CDB Architektur unterstützt
 - Premier-Support bis 2029
- Oracle Database 21c/23ai/26ai unterstützen nur noch die Multitenant Architektur

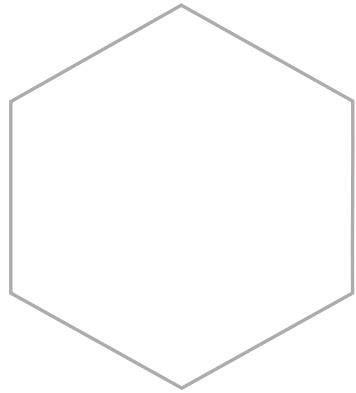
Data Guard & Multitenant (1)

- Grundsätzlich: Data Guard auf CDB-Ebene
- Einzelne PDBs können vom Data Guard ausgeschlossen werden
 - "CREATE PLUGGABLE DATABASE .. STANDBYS=NONE"
 - Parameter ENABLED_PDBS_ON_STANDBY



Data Guard & Multitenant (2)

- Switchover & Failover betreffen die komplette CDB (➔ alle PDBs)
 - Ausnahme "PDB-Level Failover" (DGMGRL> "MIGRATE PLUGGABLE DATABASE ...")
- Neuerungen in Oracle 21c bzw. 23c/23ai/26ai:
 - PDB Level Data Guard
 - Vortrag DOAG K+A 2023 "Oracle Database 23c PDB-Level Data Guard - Lohnt sich das? "
 - "PDB Side Recovery"



Agenda



Was läuft problemlos (in Oracle 19c)? 😊

- Switchover
- Failover
- Anlegen einer PDB als Kopie einer lokalen R/O-PDB (PDB\$SEED etc.)

Anlegen einer PDB als Kopie einer R/O-PDB (PDB\$SEED etc.)

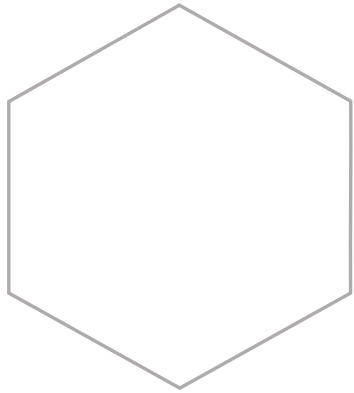
- Primär-Datenbank (SQL)

```
SQL> create pluggable database PDBDEMO2 admin user pdbadmin identified by manager
      2 roles=(DBA) file_name_convert=('pdbseed','PDBDEMO2');
Pluggable database created.
```

- Standby-Datenbank (alert.log)

```
Recovery created pluggable database PDBDEMO2
Recovery copied files for tablespace SYSTEM
Recovery successfully copied file /u01/oradata/DGDEMO_S2/PDBDEMO2/system01.dbf from
/u01/oradata/DGDEMO_S2/pdbseed/system01.dbf
...
DGDEMO_S2/pdbseed/undotbs01.dbf
PDBDEMO2(3):Recovery created file /u01/oradata/DGDEMO_S2/PDBDEMO2/undotbs01.dbf
PDBDEMO2(3):Successfully added datafile 17 to media recovery
PDBDEMO2(3):Datafile #17: '/u01/oradata/DGDEMO_S2/PDBDEMO2/undotbs01.dbf'
```

- Auch ohne Active-Data Guard-Lizenz darf CDB\$ROOT darf auf der Standby-Seite read-only geöffnet sein
 - ADG-Lizenz braucht man erst bei einer R/O-geöffneten PDB ("The container root (CDB\$Root) of the standby database may be opened read-only without requiring an Oracle Active Data Guard license. An Oracle Active Data Guard license is only required if any of the pluggable databases in the standby is opened for read." – Oracle Database 19c Licensing Information)
 - Öffnen der PDBs ➔ via Datenbank-Trigger oder Grid Infrastructure (role-based services) um Lizenzverstöße zu verhindern



Agenda



Welche Aktionen funktionieren nicht ganz "out-of-the-box"?

- Kopieren einer R/W-PDB (lokal und remote)
 - Refreshable PDB
 - PDB Relocate
- Plug-In einer PDB
- Remote-Cloning einer R/O-PDB
- Tempfiles auf der Standby-Seite (nach Switchover/Failover)

Was ist das Problem (Teil 1) – Kopieren einer R/W-PDB

- Ergebnis auf der Standby-Seite (12.2)
 - Nach dem Kopieren erfolgt auf der Primary-Seite ein Recovery-Versuch der neuen PDB
 - Fehler "ORA-1153 an incompatible media recovery is active"
 - MRP-Prozess bricht ab ☹

Was ist das Problem (Teil 2) – Kopieren einer R/W-PDB

- Ergebnis auf der Standby-Seite (18c und 19c)
 - PDB wird nicht kopiert (alert.log "PDBDEMO3(5):**Tablespace-SYSTEM during PDB create skipped since source is in r/w mode or this is a refresh clone**")
 - PDB wird nicht auf die Standby-Seite übertragen ☹
 - PDB ist "anscheinend "auf der Standby-CDB angelegt

```
CON_ID NAM
-----
5 /u00/app/oracle/homes/OraDB19Home1/dbs/UNNAMED00018
5 /u00/app/oracle/homes/OraDB19Home1/dbs/UNNAMED00019
5 /u00/app/oracle/homes/OraDB19Home1/dbs/UNNAMED00020
SQL> !ls /u00/app/oracle/homes/OraDB19Home1/dbs/

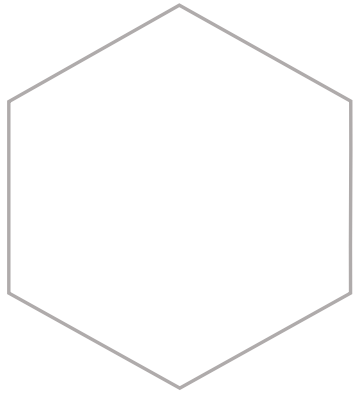
SQL>
```

- MRP läuft weiter ☺

Was ist das Problem (Teil 3) – Einhängen einer PDB

- Problem: Wie kommt die Standby-CDB an die Dateien?
- Fehlermeldung auf der Standby-Seite:

```
PDB2(4):Errors in file
/u00/app/oracle/diag/rdbms/dgcdb_s2/DGCDB/trace/DGCDB_pr00_10195.trc:
ORA-01565: error in identifying file '/u01/oradata/DGCDB/PDB2/system01.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 7
PDB2(4):Recovery was unable to create the file as:
PDB2(4) : '/u01/oradata/DGCDB/PDB2/system01.dbf'
[...]
Background Media Recovery process shutdown (DGCDB)
```



Agenda

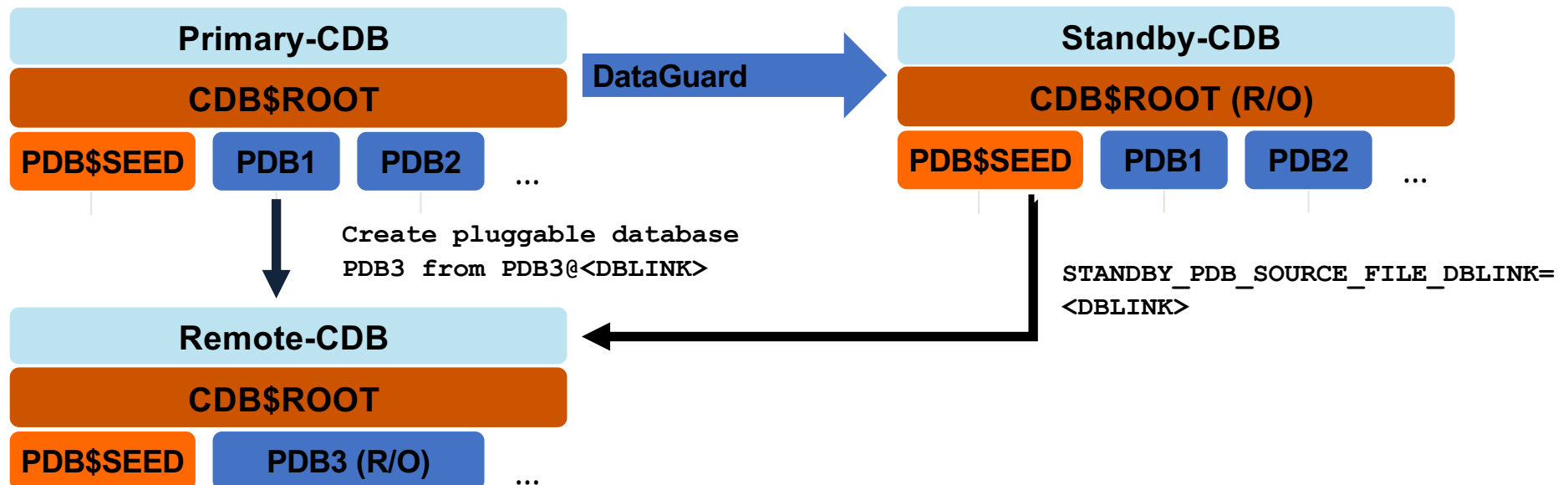


"Neue" Parameter seit 18c

- STANDBY_PDB_SOURCE_FILE_DBLINK
 - Name des Datenbank-Links, der beim Remote-Cloning auf die Quell-PDB (R/O) zeigt
- STANDBY_PDB_SOURCE_FILE_DIRECTORY
 - Verzeichnis, aus dem die Standby-PDB die Daten einer einzuhängenden PDB ("Plug-In")
 - Oft greifen Primary DB und Standby-DB auf den gleichen NFS-Mount zu

STANDBY_PDB_SOURCE_FILE_DBLINK (1) - Übersicht

- Name des Datenbank-Links, der beim Remote-Cloning auf die Quell-PDB (R/O) zeigt
- Standby-CDB (CDB\$ROOT) muss Read-Only geöffnet sein



STANDBY_PDB_SOURCE_FILE_DBLINK (2) - Voraussetzungen

- Remote-PDB Read-Only geöffnet
- Standby-CDB Read-Only geöffnet
- Und natürlich
 - DB-Link von Primary-CDB zur Remote-CDB
 - Zugriff zur Remote-CDB von der Standby-CDB via DB-Link testen ☺
 - Remote-User mit den Rechten
 - CREATE PLUGGABLE DATABASE
 - SYSOPER, CREATE SESSION (CONTAINER=ALL)

STANDBY_PDB_SOURCE_FILE_DBLINK (3) – Ablauf

- (Primary-CDB): Datenbank-Link anlegen

```
SQL> create database link remcdb connect to C##ORA2KNOW identified by <geheim>  
2    using 'server3.ora2know.internal:1521/REMCDB';  
Database link created.
```

- (Standby-CDB): Parameter STANDBY_PDB_SOURCE_FILE_DBLINK setzen

```
SQL> alter system set STANDBY_PDB_SOURCE_FILE_DBLINK=remcdb scope=memory;  
System altered.
```

- (Primary-CDB): Remote-PDB kopieren

```
SQL> create pluggable database REMTEST from REMTEST@REMCDB  
2    file_name_convert=('/u01/oradata/REMCDB/REMTEST', '/u01/oradata/DGDEMO/REMTEST');  
Pluggable database created.
```

- (Standby-CDB): Parameter STANDBY_PDB_SOURCE_FILE_DBLINK zurücksetzen

STANDBY_PDB_SOURCE_FILE_DBLINK (4) – Ergebnis

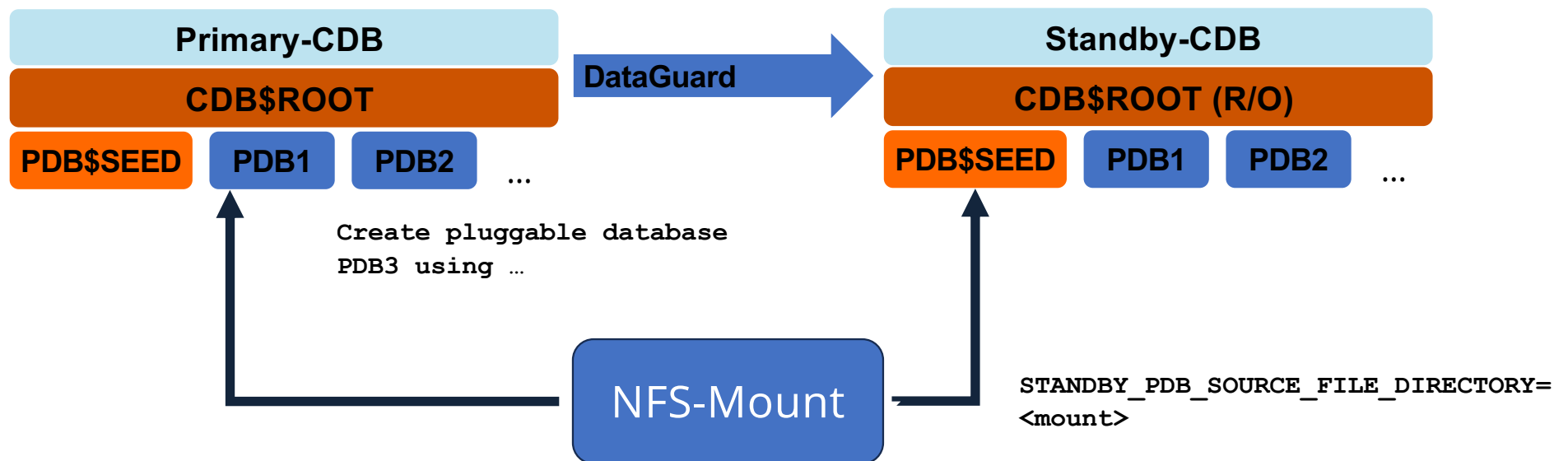
- Ergebnis auf der Standby-Seite (alert.log):

```
2025-11-16T09:41:03.153161+01:00
ALTER SYSTEM SET standby_pdb_source_file_dblink='REMCDB' SCOPE=BOTH;
2025-11-16T09:44:59.374446+01:00
Recovery created pluggable database REMTEST
Recovery attempting to copy datafiles for pdb-REMTEST from source pdb-REMTEST at dblink-
REMCDB
2025-11-16T09:45:02.076192+01:00
Recovery copied files for tablespace SYSTEM
Recovery successfully copied file /u01/oradata/DGDEMO/REMTEST/system01.dbf from
/u01/oradata/REMCDB/REMTEST/system01.dbf
[...]
REMTEST(6):Successfully added datafile 24 to media recovery
REMTEST(6):Datafile #24: '/u01/oradata/DGDEMO/REMTEST/users01.dbf'
```

- Data Guard Status: OK, MRP läuft weiter

STANDBY_PDB_SOURCE_FILE_DIRECTORY (1)

- Verzeichnis, aus dem die Standby-PDB die Daten einer einzuhängenden PDB ("Plug-In") liest



STANDBY_PDB_SOURCE_FILE_DIRECTORY (2) - Voraussetzungen

- Dateien der ausgehängten PDB liegen im Share

```
SQL> alter pluggable database PLUGTEST unplug into '/nfs/ora2know-share/plugtest.xml';  
Pluggable database altered.
```

```
SQL> drop pluggable database PLUGTEST keep datafiles;  
Pluggable database dropped.
```

```
SQL> !cp /u01/oradata/REMCDB/PLUGTEST/* /nfs/ora2know-share/.
```

STANDBY_PDB_SOURCE_FILE_DIRECTORY (3) - Ablauf

- (Standby-CDB): Parameter setzen

```
SQL> alter system set STANDBY_PDB_SOURCE_FILE_DIRECTORY='/nfs/ora2know-share/'  
2   scope=memory;  
System altered.
```

- (Primary-CDB): PDB einhängen

```
SQL> create pluggable database PLUGTEST using '/nfs/ora2know-share/plugtest.xml'  
2   source_file_name_convert=('/u01/oradata/REMCDB/PLUGTEST/', '/nfs/ora2know-share/')  
3   copy  
4   file_name_convert=('/nfs/ora2know-share/', '/u01/oradata/DGDEMO/PLUGTEST/');  
Pluggable database created.
```

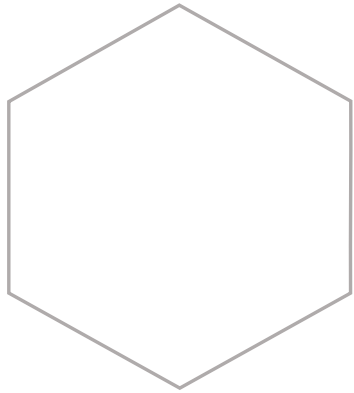
- (Standby-CDB): Parameter zurücksetzen

STANDBY_PDB_SOURCE_FILE_DIRECTORY (4) - Ergebnis

- Ergebnis auf der Standby-Seite:

```
2025-11-16T10:15:35.327043+01:00
Recovery created pluggable database PLUGTEST
Recovery attempting to copy datafiles for pdb-PLUGTEST from          source dir-
/nfs/ora2know-share/
Recovery copied files for tablespace SYSTEM
Recovery successfully copied file /u01/oradata/DGDEMO/PLUGTEST/system01.dbf from
/nfs/ora2know-share/system01.dbf
[...]
PLUGTEST(7):Successfully added datafile 28 to media recovery
PLUGTEST(7):Datafile #28: '/u01/oradata/DGDEMO/PLUGTEST/users01.dbf'
```

- Data Guard Status: OK, MRP läuft weiter



Agenda



Was sagt Oracle? – MOS-Note 2049127.1

- When using 12.2 hot clone, PDB refresh and PDB relocate functionality, the new PDBs should always be created with the STANDBYS=NONE clause.

[...]

In 18c and 19c the database will automatically create the PDBs as if STANDBYS=NONE has been specified. You will not need to use the clause for hot clones, PDB refresh PDB and PDB relocate in this version and later."

- **→ Beim Kopieren einer R/W-PDB, einer Refreshable PDB und PDB Relocate arbeitet Oracle (18c und höher) automatisch (implizit) mit "STANDBYS=NONE" um ORA-1153 zu vermeiden**

Wann muss "STANDBYS=NONE" angewendet werden

- Kopieren einer lokalen R/W-PDB
 - Kopieren einer Remote R/W-PDB
 - "Refreshable Clone"
 - "Relocate einer PDB"
-
- Es kann aber bei allen PDB-Operationen (Kopieren einer Remote R/O-PDB, Einhängen einer PDB) verwendet werden.

Ablauf (1)

- Beispiel: Kopieren einer lokalen R/W-PDB

Nur zur Verdeutlichung,
seit 18c nicht mehr
erforderlich

- **Schritt 1: (Primary-CDB): Kopieren der PDB**

```
SQL> create pluggable database RWDEMO from PDBDEMO
2   file_name_convert=('PDBDEMO','RWDEMO') standbys=none;
Pluggable database created.
```

- (Standby-CDB): alert.log

```
2025-11-16T10:24:15.155916+01:00
Recovery created pluggable database RWDEMO
RWDEMO(8):Tablespace-SYSTEM during PDB create skipped since source is in r/w
mode or this is a refresh clone
RWDEMO(8):File #29 added to control file as 'UNNAMED00029'. Originally created as:
RWDEMO(8):'/u01/oradata/DGDEMO_S1/RWDEMO/system01.dbf'
RWDEMO(8):because the pluggable database was created with nostandby
RWDEMO(8):or the tablespace belonging to the pluggable database is
RWDEMO(8):offline.
```

Ablauf (2)

- **Schritt 2: (Standby-CDB): Kopieren der Daten der PDB von der Primary-PDB mit rman**

```
RMAN> connect target /  
RMAN run{  
set newname for datafile 29 to '/u01/oradata/DGDEMO/RWDEMO/system01.dbf';  
set newname for datafile 30 to '/u01/oradata/DGDEMO/RWDEMO/sysaux01.dbf';  
set newname for datafile 31 to '/u01/oradata/DGDEMO/RWDEMO/undotbs01.dbf';  
[...]  
restore pluggable database RWDEMO from service 'DGDEMO_S1';  
}
```

- OMF macht das DBA-Leben einfacher ("set newname for pluggable database <PDB> to new")

- **Schritt 3: (Standby-CDB): Recovery mittels dgmgrl temporär deaktivieren**

- **Ab jetzt sind die beiden Datenbanken nicht mehr synchron!**

```
DGMGRL> edit database DGDEMO_s2 set state='apply-off';
```

Ablauf (3)

- **Schritt 4: (Standby-CDB): rman switch to copy**

```
RMAN> connect target /  
RMAN> switch pluggable database RWDEMO to copy;
```

- **Schritt 5: (Standby-CDB) Starten in den Mount-Status**

```
SQL> startup nomount;  
SQL> alter database mount standby database;
```

- **Schritt 6: (Standby-CDB) PDB-Recovery aktivieren**

```
SQL> alter session set container=RWDEMO;  
SQL> SQL> alter pluggable database enable recovery;
```

Ablauf (4)

▪ Schritt 7: (Standby CDB) Dateien der PDB online setzen

```
SQL> alter database datafile 29 online;  
SQL> alter database datafile 30 online;  
SQL> alter database datafile 31 online;
```

- Bei größeren PDBs natürlich per Skript ☺

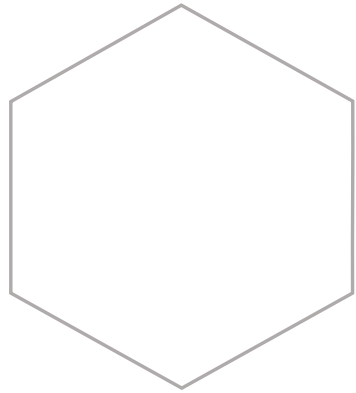
▪ Schritt 8: (Standby CDB): Recovery wieder aktivieren

```
DGMGRL> edit database DGDEMO_s2 set state='apply-on';
```

- MRP spielt die seit Schritt 3 angefallenen Änderungen auf der Standby-Seite ein

Tempfiles

- Tempfiles werden auf der Standby-Seite nicht automatisch angelegt
- Workaround:
 - Nach dem Öffnen der PDB (Switchover, Failover) Tempfiles anlegen
("ALTER DATABASE .. ADD TEMPFILE ...")
- Mit 26ai wird alles besser 😊



Agenda



Zusammenfassung (1)

Szenario	
Kopieren einer lokalen R/O-PDB	out-of-the-box ☺
Kopieren einer remote R/O-PDB	STANDBY_PDB_SOURCE_FILE_DBLINK
Einhängen einer PDB ("Plug-In")	STANDBY_PDB_SOURCE_FILE_DIRECTORY
Kopieren einer lokalen R/W-PDB	"STANDBYS=NONE"-Vorgehen
Kopieren einer remote R/W-PDB	"STANDBYS=NONE"-Vorgehen
Refreshable PDB	"STANDBYS=NONE"-Vorgehen
Relocate einer PDB	"STANDBYS=NONE"-Vorgehen

Zusammenfassung (2)

ORDIX®

- **Frage:** **Multitenant & Data Guard – was kann da schon schiefgehen?**
- **Antwort:** **Wenn man es richtig macht, praktisch nichts 😊**

Und was ist mit Non-CDBs?

- Remote-Cloning einer R/O-Non-CDB → STANDBY_PDB_SOURCE_FILE_DBLINK
- Remote-Cloning einer R/W-Non-CDB → "Standbys=NONE"
 - Analog bei Refreshable PDB
- Einhängen einer Non-CDB → STANDBY_PDB_SOURCE_FILE_DIRECTORY
- .. Und natürlich das Skript "noncdb2pdb.sql" auf der Primär-Seite nicht vergessen

- Using standby_pdb_source_file_dblink and standby_pdb_source_file_directory to Maintain Standby Databases when Performing PDB Remote Clones or Plugins (Doc ID 2274735.1)
- PDB Hot Clone in Physical Standby Database (Doc ID 3012032.1)
- Known Issues of ORA-01274 in a Multitenant Dataguard Environment (Doc ID 2766887.1)
- Data Guard Impact on Oracle Multitenant Environments (Doc ID 2049127.1)

Referenzen & weitere Informationen – MOS-Notes (2)



- Data Guard FAQ in Multitenant Architecture (Doc ID 2911617.1)
- Making Use Deferred PDB Recovery and the STANDBYS=NONE Feature with Oracle Multitenant (Doc ID 1916648.1)
- Datafile on Standby Database is created under incorrect directory (Doc ID 2346623.1)



**Vielen Dank für
Ihre Aufmerksamkeit**

Markus Flechtner



@markusdba.bsky.social



@markusdba@mastodon.social



markusdba.de | .net



mfl@ordix.de

ORDIX[®]



Aktiengesellschaft für Softwareentwicklung,
Schulung, Beratung und Systemintegration

Zentrale Paderborn
Karl-Schurz-Straße 19a
33100 Paderborn
Tel.: 05251 1063-0
Fax: 0180 1 67349 0

Seminarzentrum Wiesbaden
Kreuzberger Ring 13
65205 Wiesbaden
Tel.: 0611 77840-00

info@ordix.de
<https://www.ordix.de/>