

# Wie Oracle Single/Multitenant den DBA-Alltag verändern wird

**Mike Dietrich**  
**Oracle Corporation**  
**München, Deutschland**

## Schlüsselworte

Oracle Database 12c ;12.1.0.2; Single Tenant; Multitenant; Deprecation of non-CDB architecture;

## Einleitung

Zu Beginn des Jahres 2015 hat Oracle eine "Deprecation" Nachricht zur non-CDB Datenbank-Architektur veröffentlicht, die etwas unpräzise vermeldete, dass irgendwann in einem zukünftigen Release der Datenbank die non-CDB (also Standard-) Architektur noch mehr unterstützt werde und möglicherweise sogar nicht mehr zur Verfügung stünde. Später wurde diese Nachricht dann präzisiert und erweitert, denn dieser Wechsel wird nicht mit Oracle 12.1 oder Oracle 12.2 passieren. Was bedeutet das aber nun für den Alltag eines DBAs?

## Deprecation of the non-CDB architecture

Aktuell findet sich folgender Hinweis im Oracle Database 12.1.0.2 Upgrade Guide:

### 8.1.1 Deprecation of Non-CDB Architecture

The non-CDB architecture is deprecated in Oracle Database 12c, and may be desupported and unavailable in a release after Oracle Database 12c Release 2. Oracle recommends use of the CDB architecture.

#### Note:

There remain a small number of features that do not work with the CDB architecture (see README, section 2.2.1 "Features Restricted or Not Available for a Multitenant Container Database"). If you need these features, then continue to use the non-CDB architecture until your required feature works with the CDB architecture.

Abb. 1: <https://docs.oracle.com/database/121/UPGRD/deprecated.htm#UPGRD60000>

## Keine Panik!

Zuerst ist eine *Deprecation Note* nur ein Hinweis, um ein bestimmtes Feature zu gegebener Zeit in den Desupport Status überführen zu können. *Deprecated* bedeutet erst einmal "wird nicht mehr weiter entwickelt, aber noch zu 100% unterstützt", wohingegen *desupported* meint, dass es für ein bestimmtes Feature oder Produkt keine neuen Fehlerbehebungen (volgu: Bugfixes bzw Patches) mehr gibt. Diese mögliche Veränderung wird weder mit Oracle Database 12.1 noch mit Oracle Database 12.2 eingeführt werden.

Nichtsdestotrotz sollte jeder aktuell zumindest ein paar kleinere Installationen als Single Tenant betreiben. Selbst dann, wenn keine Bestrebungen bestehen, Oracle Multitenant zur Konsolidierung in naher Zukunft einzusetzen.

## Warum muss ich Oracle Single Tetant schon heute nutzen?

Diese Frage ist leicht und schnell beantwortet: Bei der Nutzung von Oracle Single/Multitenant ergeben sich viele Veränderungen im täglichen DBA-Geschäft. Und wer heute wartet und sich die

Thematik nicht näher ansieht – unter Realbedingungen – wird morgen möglicherweise mit tiefgreifenden Veränderungen konfrontiert werden, die den Arbeitsalltag vor grosse Herausforderungen stellen – neben den normalen Aufgaben.

### Was ist Single/Multitenant überhaupt – und wozu brauche ich das?

Oracle Single Tenant ist eine Container-Database Architektur, die aber nur eine aktive Pluggable Database besitzt. Die immer existente, und im Read-Only-Modus befindliche PDB\$SEED zählt nicht als Pluggable Database. Für Oracle Single Tenant ist keine extra Lizenz oder Option erforderlich. Diese Möglichkeit ist in allen verfügbaren Oracle Editionen seit Oracle Database 12.1.0.1 mit inbegriffen, also auch in Standard Edition 2 (SE2).

Oracle Multitenant ist eine kostenpflichtige Option, die zusätzlich zur Enterprise Edition erworben werden kann. Sie erlaubt aktuell maximal 252 aktive Pluggable Databases gleichzeitig in einer Container Datenbank und ist deshalb ein ideales Mittel zur Konsolidierung und zur besseren Nutzung von Rechnerressourcen ohne sich damit die Nachteile von Virtualisierungssoftware und der damit verbundenen Pflege einzuhandeln.

Nach wie vor gibt es aber auch die Möglichkeit, eine Datenbank ganz normal (ich nenne das "stand alone") zu installieren und zu betreiben.

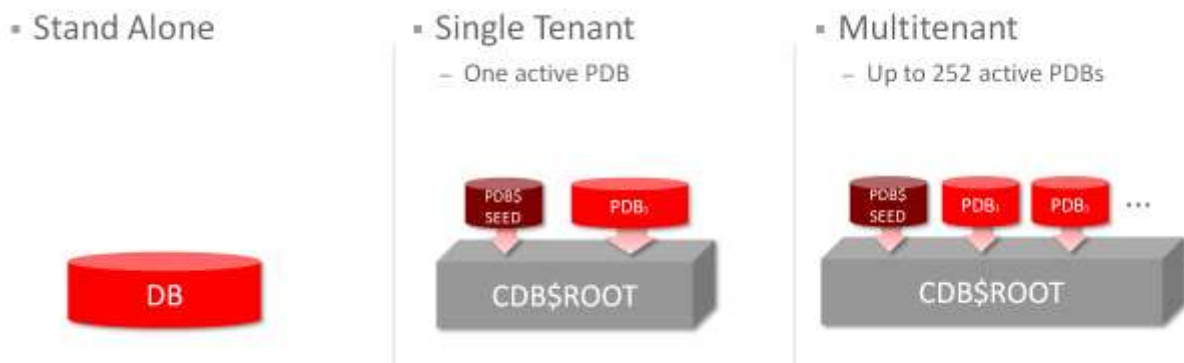


Abb. 2: Non-CDB Architektur – Single Tenant - Multitenant

Und alle genannten Möglichkeiten lassen sich mit den bekannten Oracle Features wie DataGuard, Real Application Clusters u.s.w. kombinieren.

Das Potential von Oracle Multitenant liegt auf der Hand. Aber warum sollte jeder DBA mindest ein paar kleinere Datenbanken als Oracle Single Tenant betreiben?

Einmal, um die notwendige Erfahrung zu sammeln, die sich aus dieser neuen Architektur ergibt. Zum anderen bieten sich aber auch ansonsten einige Vorteile, die sich im ersten Moment nicht erschliessen, in der Praxis aber durchaus einen grossen Nutzen haben, wie bspw. beim Remote Cloning, beim Upgrade oder durch die eingebaute Trennung von Daten- und Administrationslayer.

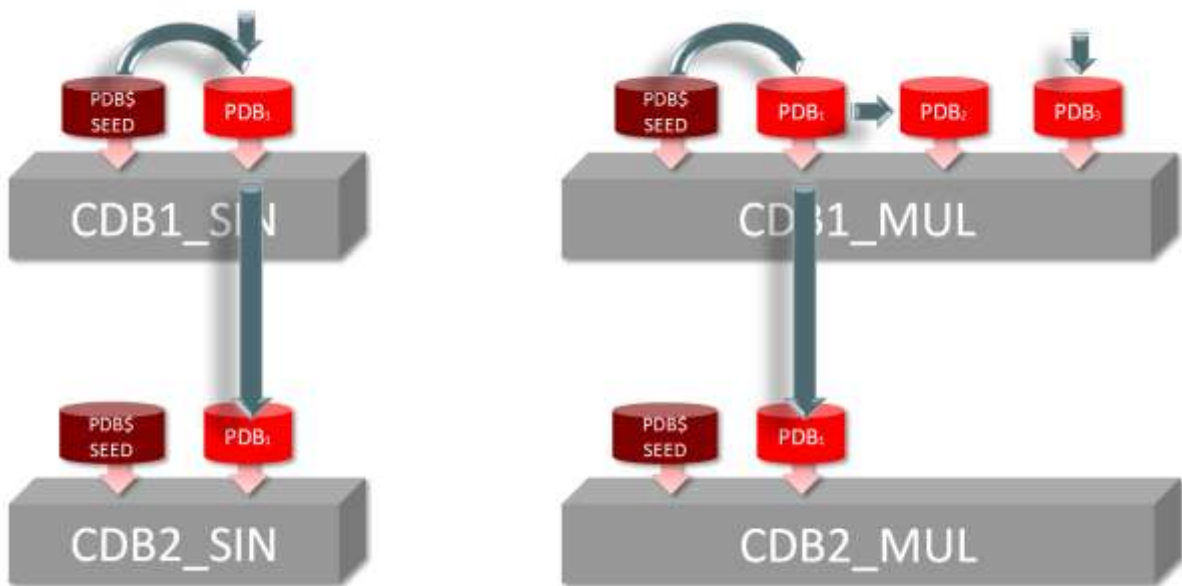


Abb.3: Single Tenant erlaubt schnelles Provisionieren und Remote Cloning, Multitenant erlaubt auch lokales Cloning und Snapshot Cloning – in beiden Fällen muss eine einzuhängende Datenbank bereits auf 12c sein

### Wie beginnen?

Am Anfang steht das Anlegen einer neuen Container Datenbank mit dem DBCA (Database Configuration Assistant). Dabei gibt es zwei Wege: einmal das "All Inclusive" Programm, bei dem alle Optionen in der Container Datenbank präsent sind, oder der Umweg, den DBCA erst nur Skripte anlegen zu lassen, um diese dann zu editieren, damit bestimmte Optionen nicht angelegt werden.

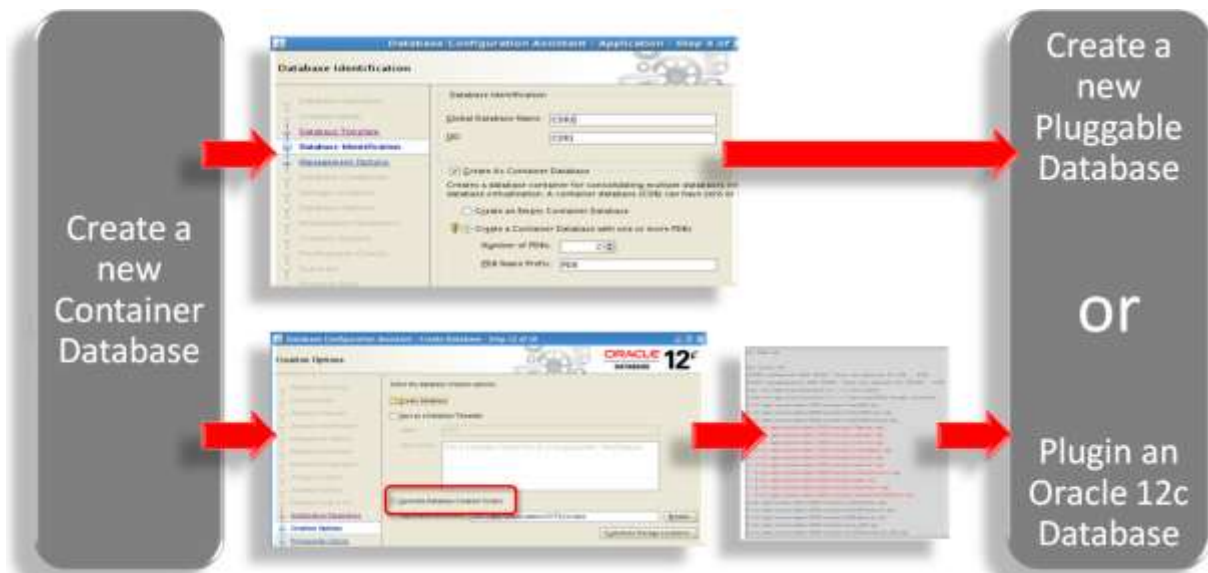


Abb.4: Anlegen einer Container Datenbank mit dem DBCA – entweder "All Inclusive" oder mit selektierten Optionen per Skripte, die editiert werden müssen

## Wo liegen die Unterschiede?

Bei der Nutzung von Single Tenant gegenüber einer normalen Datenbank (non-CDB) Architektur ergeben sich die Unterschiede zuerst einmal im *Footprint* auf der Festplatte, denn bei einer Single Tenant Datenbank existieren mehr Dateien.

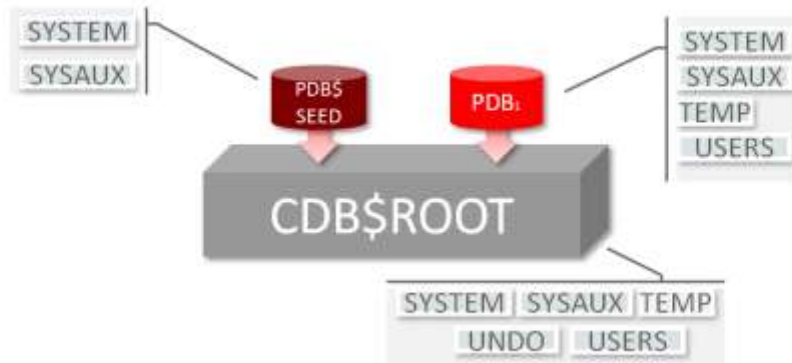


Abb. 5: Single Tenant – Footprint on Disk

Ausserdem müssen viele Skripte nun mit dem PERL Skript `catcon.pl` ausgeführt werden, das einige Parameter erfordert. Warum `catcon.pl`? Die `PDB$SEED`, die kleine PDB, mit deren Hilfe schnell eine neue PDB provisioniert werden kann, ist immer im Read-Only-Modus. `catcon.pl` kann nun bspw ein Patch Skript, das notwendigerweise nicht nur in der PDB, sondern auch in der `PDB$SEED` und im Root Container `CDB$ROOT` ablaufen muss, ausführen.

Auch die vielleicht entscheidendste Einschränkung im täglichen Umgang mit PDBs will berücksichtigt werden: eine PDB erlaubt keine direkte Authentifizierung über das Betriebssystem, der bestens vertraute "`connect / as sysdba`" funktioniert nicht. Das kann gerade bei Jobs, die über das Betriebssystem gestartet werden, einen gewissen Umstellungsaufwand nach sich ziehen.

## Fazit

Single Tenant wird den Alltag von DBAs verändern. Wichtig ist es, sich rechtzeitig mit den Anpassungen vertraut zu machen, um von den Stärken des Konzepts zu profitieren.

Slides zu diesem Vortrag finden Sie unter: <http://blogs.oracle.com/UPGRADE> im **Slides Download Center** unter dem Titel "*Single Tenant for DBAs*".

## Kontaktadresse:

Mike Dietrich  
Oracle Corporation  
D-80992 München

Internet: <http://blogs.oracle.com/UPGRADE>  
Twitter: MikeDietrichDE