

Datenbank Upgrades Duplicate noopen als Basis für Upgrades

Carsten Voigt

Senior Consultant

OPITZ CONSULTING Deutschland GmbH
Standort Berlin
Tempelhofer Weg 64 - 12347 Berlin

Telefon: +49 30 629 8889- 1616
Telefax: +49 30 629 8889-4600
Mobil: +49

carsten.voigt@opitz-consulting.com
www.opitz-consulting.com



Christian Floreck

System Engineer

OPITZ CONSULTING Deutschland GmbH
Standort Berlin
Tempelhofer Weg 64 - 12347 Berlin

Telefon: +49 30 629 8889-1639
Telefax: +49 30 629 8889-4600
Mobil: +49 173 727 94 50

christian.floreck@opitz-consulting.com
www.opitz-consulting.com





Datenbank Upgrades

von 11.2.0.3 auf 18c

Duplicate noopen als Basis für Upgrades

Datenbank Upgrades

1 Grundsätzliche Überlegungen

2 Die Lösungsmöglichkeiten

3 Der Ablauf im Detail



Datenbank Upgrades

eine ungeliebte Notwendigkeit ?

Eine Notwendigkeit wegen

- auslaufendem Support
 - Doc ID 742060.1
- Wechsel der System Umgebung / OS Upgrade
- Anforderung der Applikation

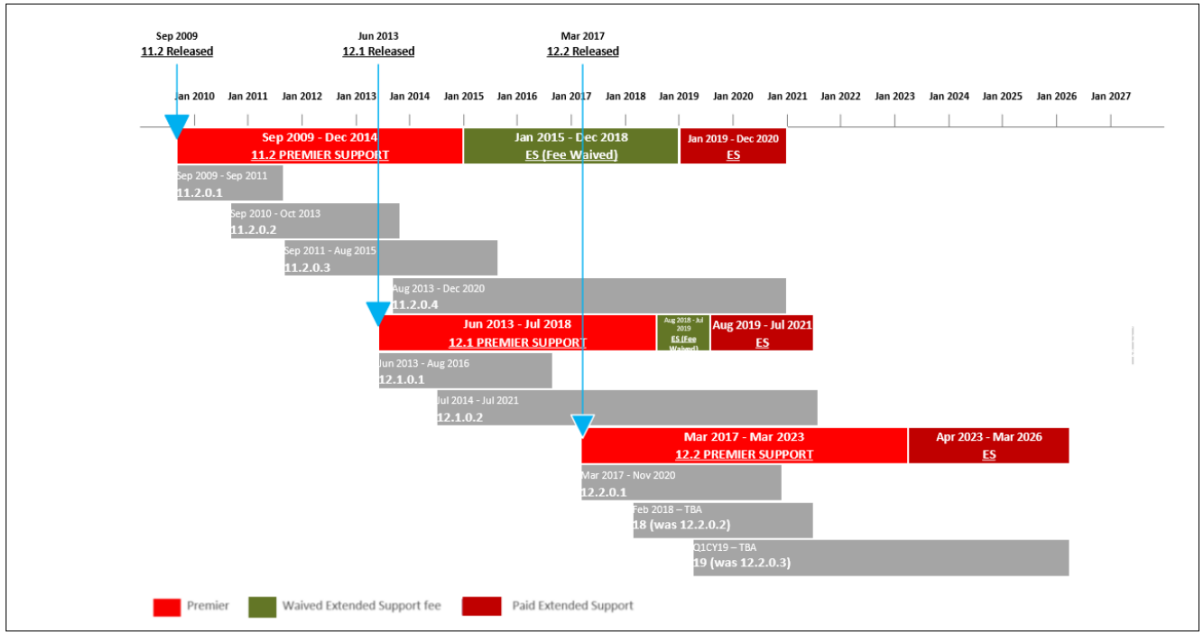
Ungeliebt wegen

- Oft hohe Downtime
- Notwendigkeit komplexer Planungen
- Risiken durch Upgrade
 - Fehler beim Upgrade Prozess
 - Probleme mit der Applikation

- **12c, 18, 19 Extended Support Dates (See Figure 1, Tables 2 & 3)**
- **Added Estimated release dates for 19c (See Figure 1, Tables 2 & 3)**
- **Reorganized document content but retained major sections**

Oracle Database Release Roadmap

To help you plan ahead, below is a roadmap that captures Oracle Database Releases (incl. annual and patch sets) from 11.2 and later, showing their lifetime support stages and duration in relationship to the overall life of the release. For a table with this data, see *Table 3 - Patching End Dates* below.



Upgrade Database

Grundsätzliche Überlegungen

- **Größe der Datenbank Systeme**
 - Catalog Upgrade Zeit ist nicht linear abhängig von DB Größe
 - Laufzeit für „Bereitstellung“ der Datenbank
 - duplicate
 - expdp / impdp
 - Laufzeit von Schema-Statistiken
 - Laufzeit für eventuellen Restore (für Fallback)
- **Installierte Datenbank Optionen**
 - Müssen eventuell entfernt werden
 - Haben Einfluss auf die Upgrade Laufzeit
- **Notwendigkeit von Tests**
 - Test des Datenbank Upgrades
 - Applikations-Test mit neuem DB Release u. aktuellen Daten



Upgrade Database

Grundsätzliche Überlegungen

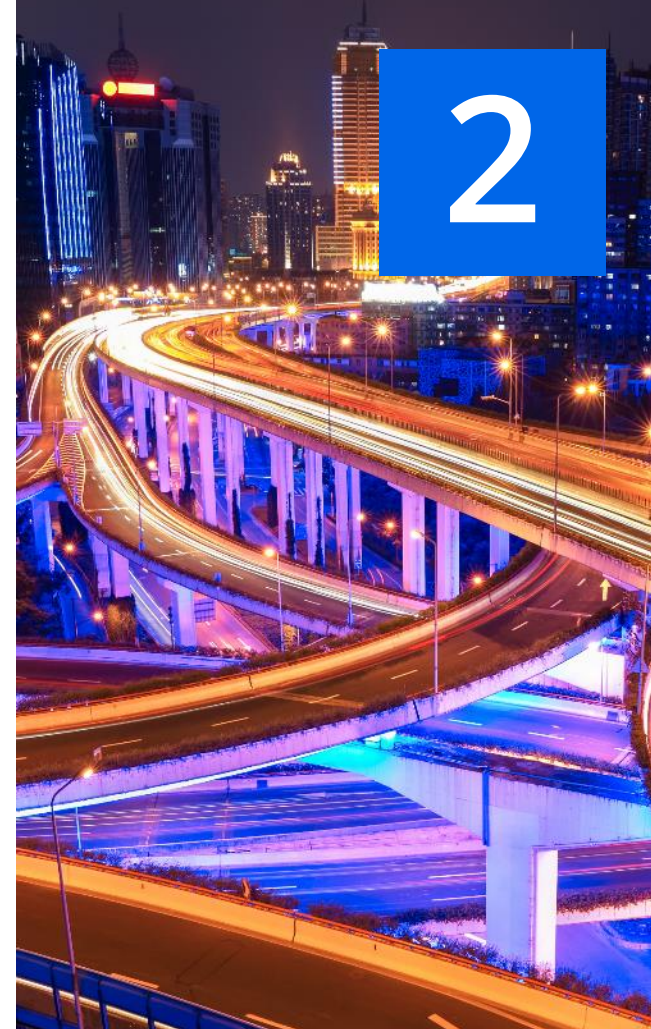
- **Fallback Aspekte**
 - Dauer des Fallbacks
 - Verfügbare Fallback Techniken
 - Zuverlässigkeit des Fallback-Konzepts
- **Verfügbarer Storage**
 - Geschwindigkeit
 - Größe hat Einfluss auf mögliche Vorgehensweise
- **Vertretbare Downtime**
 - Verfügbare Zeit für den Upgrade
 - Zeitpunkt für Abbruch des Upgrades (Fallback Zeit)
 - Prüfung von Lösungen wie Golden Gate



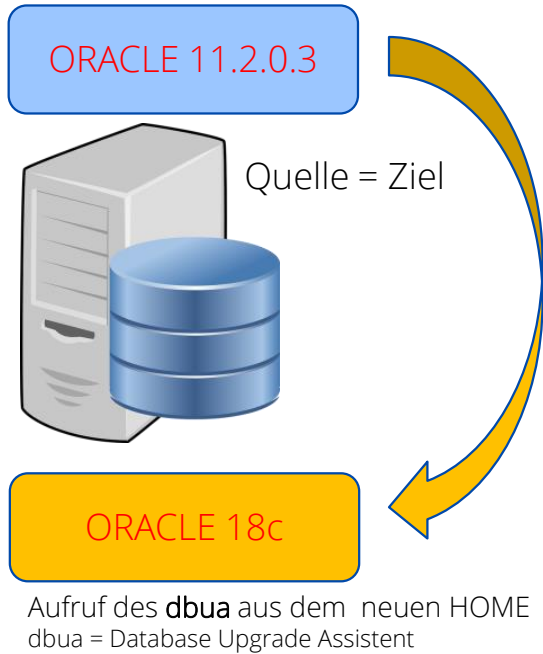
Upgrade Database

Die Lösungsmöglichkeiten

2



Upgrade Database via dbua



Pro

- ✓ Komfortable Vorgehensweise durch GUI
- ✓ Kein zusätzlicher Storage Bedarf

Contra

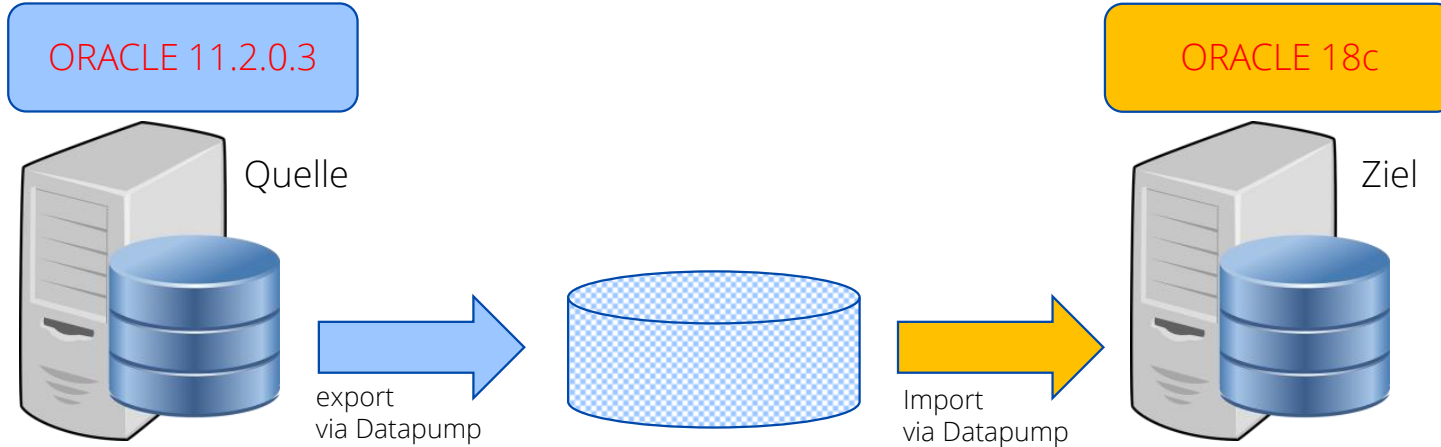
- Fallback u.U. zeitaufwendig
- Kein Test möglich

Sicherer Fallback nur durch

- Restore
- Flashback, benötigt Enterprise Edition

Evtl. Snapshots auf virtuellen Maschinen

Upgrade Database via datapump



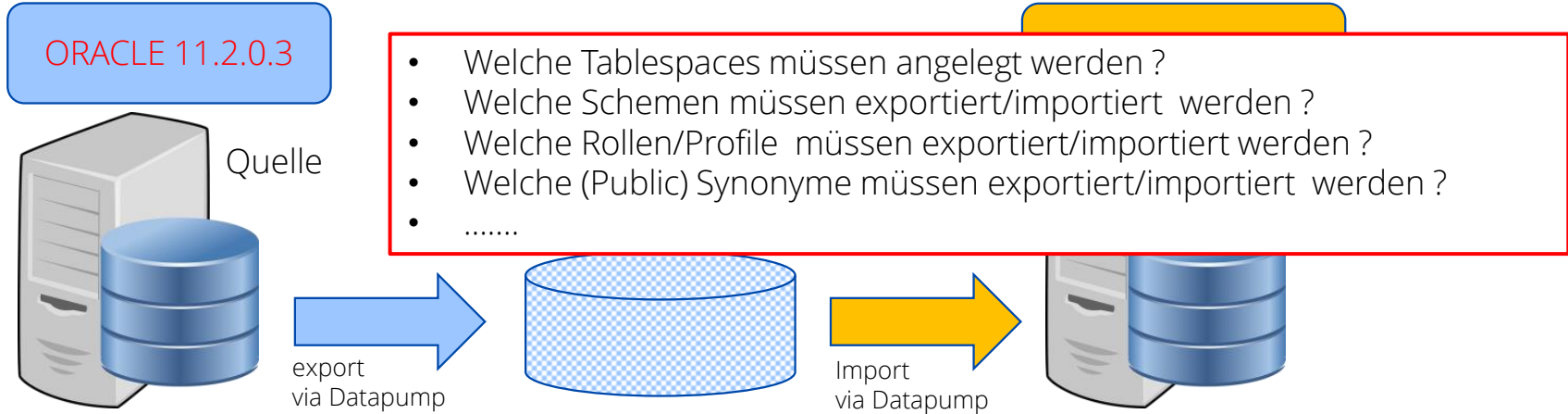
Pro

- ✓ Ursprüngliche Datenbank bleibt erhalten
- ✓ Tests mit aktuellen Daten sind möglich
- ✓ Schneller und sicherer Fallback

Contra

- Höherer Storage Bedarf
- Eventuelle sehr lange Laufzeiten / Downtime
- Komplexe Konfiguration

Upgrade Database via datapump



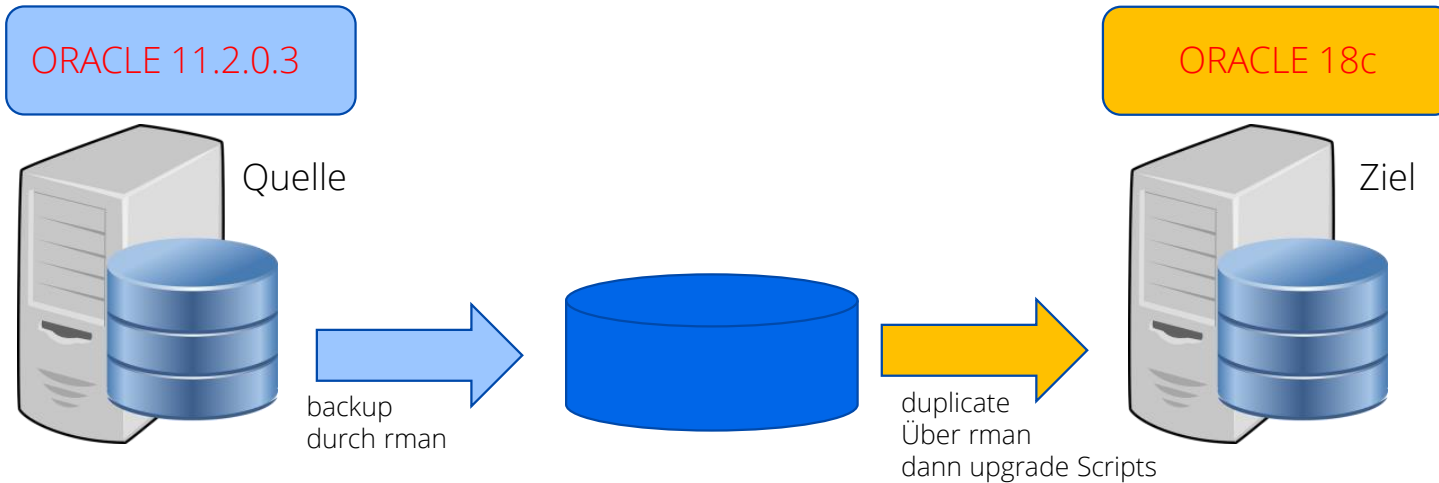
Pro

- ✓ Ursprüngliche Datenbank bleibt erhalten
- ✓ Tests mit aktuellen Daten sind möglich
- ✓ Schneller und sicherer Fallback

Contra

- Höherer Storage Bedarf
- Eventuelle sehr lange Laufzeiten / Downtime
- Komplexe Konfiguration

Upgrade Database via rman



Pro

- ✓ Ursprüngliche Datenbank bleibt erhalten
- ✓ Tests mit aktuellen Daten sind möglich
- ✓ Schneller und sicherer Fallback

Contra

- Höherer Storage Bedarf
- Längere Laufzeiten / Downtime
aber geringer als export / import

rman NOOPEN-Option :

Vor 12c : die duplizierte Datenbank wurde im letzten Schritt von RMAN automatisch geöffnet

Ab 12c NOOPEN-Option: die duplizierte Datenbank nach dem Duplizieren nicht öffnen
→ geklonte DB verbleibt im MOUNT-Status und muss manuell geöffnet werden

Jetzt kann eine Aktualisierung der duplizierten Datenbank erfolgen:

- Öffnen der Datenbank im UPGRADE-Modus
- Ausführung des entsprechenden Upgrade-Skripts

Upgrade Database via rman

Siehe Doc ID 2022820.1

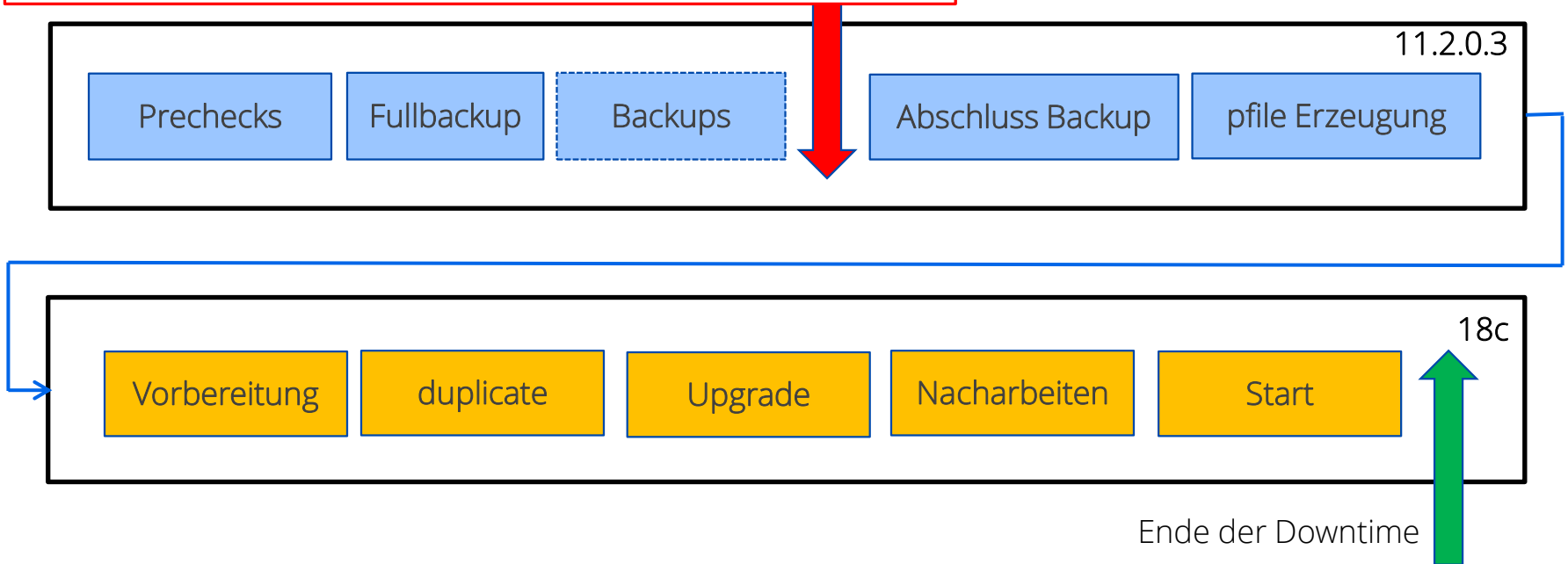
Quell System >= 11.2.0.3
Ziel System >= 18c

To Release	From Release	Minimum require version
12.1c	10g	10.2.0.5.0
12.1c	11gR1	11.1.0.7, 11.2.0.2.0
12.2c	11gR2	11.2.0.3, 11.2.0.4
12.2c	12cR1	12.1.0.1, 12.1.0.2
18c	11gR2	11.2.0.3, 11.2.0.4
18c	12cR1	12.1.0.1, 12.1.0.2
18c	12cR2	12.2.0.1

Upgrade Database Gesamt Ablauf

Nach den Fullbackup können weitere Backups gefahren werden (z.B. ArchiveLogs). Sie müssen für den duplicate Prozess bereit gestellt werden

Downtime



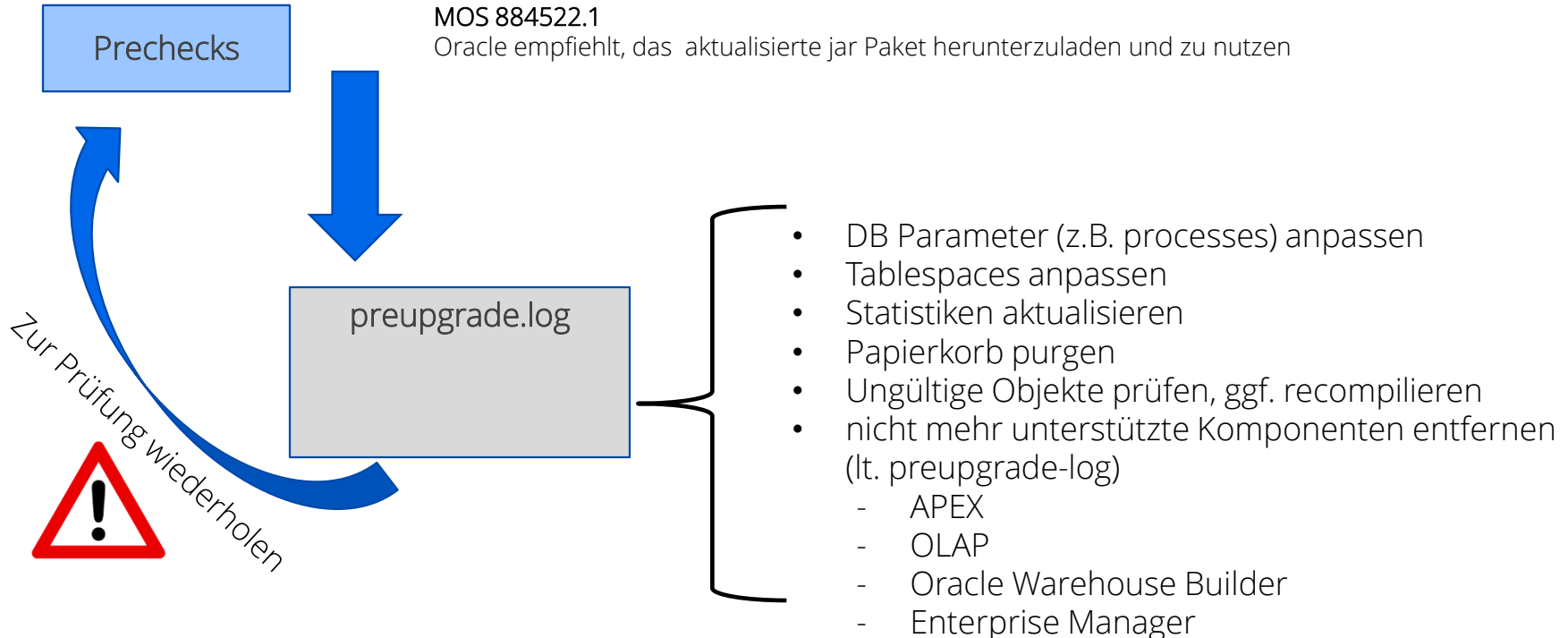
Upgrade Database Vorarbeiten

Oracle >= 12.2:

```
$ORACLE_HOME/jdk/bin/java -jar /u01/app/oracle/product/18c/dbhome/rdbms/admin/preupgrade.jar
```

MOS 884522.1

Oracle empfiehlt, das aktualisierte jar Paket herunterzuladen und zu nutzen



Upgrade Database Vorarbeiten

Oracle >= 12.2:

```
$ORACLE_HOME/jdk/bin/java -jar /u01/app/oracle/product/18c/dbhome/rdbms/admin/preupgrade.jar
```

MOS 884522.1

Oracle empfiehlt, das aktualisierte jar Paket herunterzuladen und zu nutzen

Prechecks

preupgrade.log

- DB Parameter (z.B. processes) anpassen
- Tablespaces anpassen
- Statistiken aktualisieren
- Papierkorb purgen
- Ungültige Objekte prüfen, ggf. recompilieren
- nicht mehr unterstützte Komponenten entfernen (lt. preupgrade-log)
 - APEX

Oracle 12.1:

Scripte **preupgrd.sql** & **utluppkg.sql** aus <ORACLE_12_HOME>/rdbms/admin für 11er bereitstellen
preupgrd.sql aus 11er mit sysdba Rechten starten.

Zur Prüfung wiederholen



Upgrade Database Vorarbeiten

Oracle >= 12.2:

```
$ORACLE_HOME/bin/lsnrctl stop && cd $ORACLE_HOME/bin && java -jar dbupgrade.jar
```

M
O

Achtung:

preupgrade.log

=> Unbedingt prüfen!

preupgrade_fixups.sql

=> Unbedingt ausführen!

[MOS Note: 2009405.1](#)

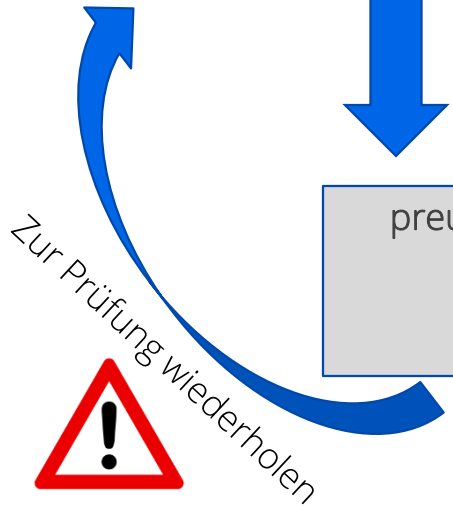


- DB Parameter (z.B. processes) anpassen
- Tablespaces anpassen
- Statistiken aktualisieren
- Papierkorb purgen
- Ungültige Objekte prüfen, ggf. recompilieren
- nicht mehr unterstützte Komponenten entfernen (lt. preupgrade-log)
 - APEX
 - OLAP
 - Oracle Warehouse Builder
 - Enterprise Manager

Prechecks



preupgrade.log



Zur Prüfung wiederholen

Aufgetreten bei Migration 11.2.0.4 auf 12.1

First of all it's important to know that you can't upgrade your database without resolving this error condition before. If you'd ignore it you'll see the magic **universal** ORA-1722: invalid number **error** indicating that **one of the mandatory checks in the upgrade scripts had failed**.



For more information on [type evolution](#) check the [Oracle Database 12c Object-Relational Developer's Guide](#).

What does this warning mean?

The error results from a failed check for table data in columns of **evolved types**. Those **must be upgraded before the database can be upgraded**, otherwise they will be **considered as "invalid" data**. The same thing can happen when you try to plugin a stand-alone (non-CDB) database making it a pluggable database. The sanity script `noncdb_to_pdb.sql` will also check for this condition.

Aufgetreten bei Migration 11.2.0.4 auf 12.1

First of all it's important to know that you can't upgrade your database without resolving this error condition before. If you'd ignore it you'll see the magic **universal** ORA-1722: invalid number **error** indicating that **one of the mandatory checks in the upgrade scripts had failed**.



For more info

Relational De

What does

The error res

must be upg

considered

Inkompatibilität in 11er Datenbank

⇒ Upgrade nach 12 **nicht** möglich

Datenbank mit Release 12 geöffnet

⇒ Control File Header modifiziert

⇒ Datenbank kann nicht mehr in 11er Umgebung gestartet werden.

⇒ Rollback nötig, ggf. Restore

stand-alone (non-CDB) database making it a pluggable database. The sanity script noncdb_to_pdb.sql will also check for this condition.

Upgrade Database

Sonderfall Zweistufiges Upgrade

Ein direkter Upgrade von 10.2.0.4 auf 18c ist **nicht** möglich.



Upgrade Database Sonderfall Multitenant Systemen



Bei Migrationen in multitenant Systemen:

Alle Pluggable Databases müssen vor Start des preupgrade.jar Scripts geöffnet sein!

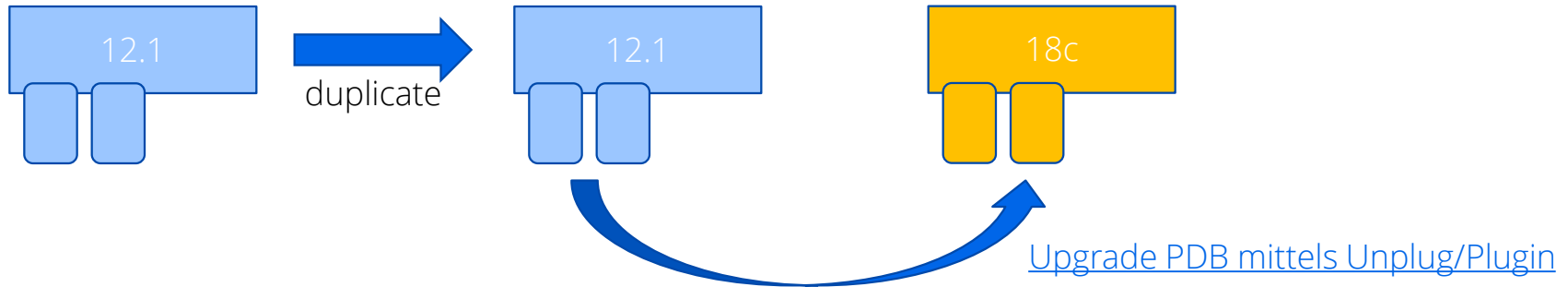
```
alter pluggable database all open;
```

```
preupgrade.jar TERMINAL TEXT -c "pdb1" << Nur Container DB pdb1 prüfen
```

```
preupgrade_fixups_<pdb1>.sql << Nur Container DB pdb1 „fixen“.
```

Achtung, vorher in den Container wechseln:

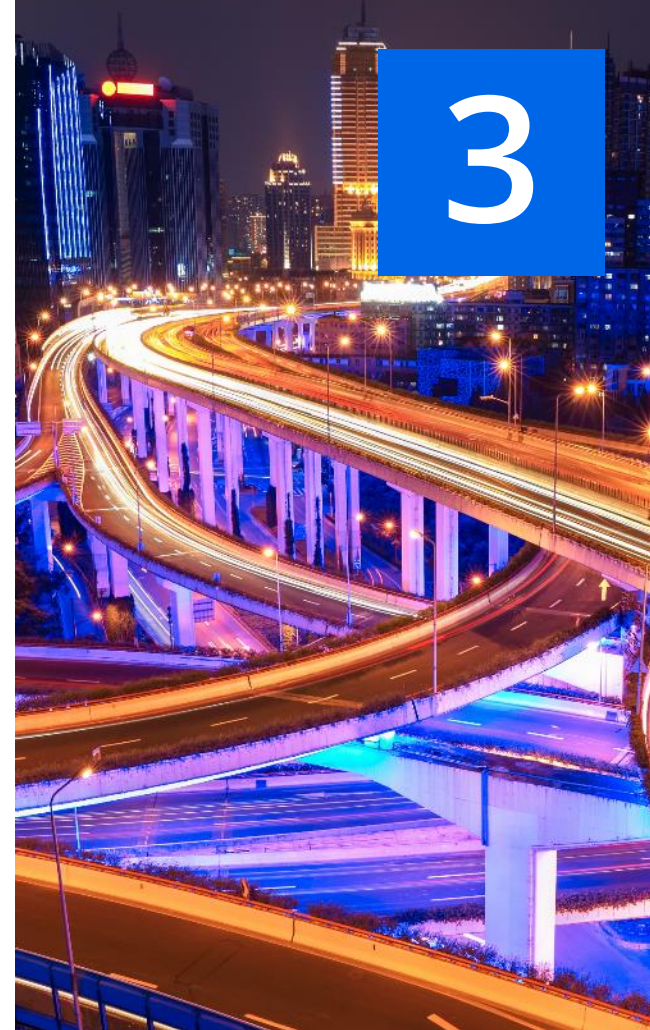
```
alter session set container= pdb1
```



Upgrade Database via RMAN

Der Ablauf

3



Upgrade Database

Vollständiges Backup der Datenbank

Fullbackup

```
rman
connect target /
backup database format '/backups/files/db_%U' plus archivelog format '/backups/files/arch_%U';
```

ggf. Archive

```
rman
connect target /
backup archivelog all format '/backups/files/arch_%U';
```

Die Datenbank ist während des Backups für die Anwender verfügbar.

Upgrade Database

Erzeugen pfile und Erstellen Backup

Pfile Erzeugung
Abschluß Backup

```
create pfile='/backups/files/pfile/p.txt' from spfile;  
alter system switch logfile;  
alter system switch logfile;  
alter system switch logfile;
```

```
rman  
connect target /  
backup current controlfile format '/backups/files/cf_%U';  
backup archivelog all format '/backups/files/arch_%U';
```

Der Zugriff auf die Datenbank muss vorher gesperrt werden.
→ Beginn der Downtime

Upgrade Database

Vorbereitungen für Restore

Vorbereitung

Anpassen des Pfiles der 11er Datenbank → initDBR18.ora

- Dynamische Memory Einstellungen löschen
- control_files Eintrag komplett löschen
- db_name
- db_create_file_dest
- db_create_online_log_dest_X
- db_recovery_file_dest
- audit_file_dest

Dieses Verzeichnis manuell anlegen

```
orapwd file=orapwDBR18 format=12
```

```
startup nomount;
```

Diese Arbeiten erfolgen nun im 18c Home

Upgrade Database

Vorbereitungen für Restore

Vorbereitung

Anpassen des Pfiles der 11er Datenba

- Dynamische Memory Einstellungen
- control_files Eintrag komplett lösch
- db_name
- db_create_file_dest
- db_create_online_log
- db_recovery_file_dest
- audit_file_dest

Dieses Verzeichnis

```
orapwd file=orapwDBR18
```

```
startup nomount;
```

Diese Arbeiten erfolge

```
*.audit_file_dest='/u01/app/oracle/admin/DBR18/adump'  
*.db_create_file_dest='/data/database/R18/data'  
*.db_create_online_log_dest_1='/data/database/R18/mirrorA'  
*.db_create_online_log_dest_2='/data/database/R18/mirrorB'  
*.db_name='DBR18'  
*.db_recovery_file_dest='/data/database/R18/fast'
```

```
DBR11._db_cache_size=4009754624  
DBR11._java_pool_size=16777216  
DBR11._large_pool_size=16777216  
DBR11._oracle_base='/u01/app/oracle/#ORACLE_BASE set from environment  
DBR11._pga_aggregate_target=1660944384  
DBR11._sga_target=4966055936  
DBR11._shared_io_pool_size=0  
DBR11._shared_pool_size=872415232  
DBR11._streams_pool_size=16777216  
*.audit_file_dest='/u01/app/oracle/admin/DBR11/adump'  
*.control_files='/data/database/R11/mirrorA/DBR11/controlfile/o1_mf_g0z8rl6z_.ctl',  
*.db_create_file_dest='/data/database/R11/data'  
*.db_create_online_log_dest_1='/data/database/R11/mirrorA'  
*.db_create_online_log_dest_2='/data/database/R11/mirrorB'  
*.db_name='DBR11'  
*.db_recovery_file_dest='/data/database/R11/fast'
```

Upgrade Database Restore in den noopen Modus

Restore

```
rman  
connect auxiliary /  
duplicate database to DBR18 noopen backup location '/backups/files/';
```



.....

.....

Reenabling controlfile options for auxiliary database

Executing: alter database force logging

Leaving database unopened, as requested

Cannot remove created server parameter file

Finished Duplicate Db at 03-FEB

Upgrade Database Upgrade

Upgrade

```
alter database open resetlogs upgrade;
```

```
cd $ORACLE_HOME/rdbms/admin  
$ORACLE_HOME/perl/bin/perl catctl.pl -n 6 -l /tmp/step1 catupgrd.sql
```

```
startup;
```

```
cd $ORACLE_HOME/rdbms/admin  
$ORACLE_HOME/perl/bin/perl catctl.pl -n 6 -l /tmp/step2 catuppst.sql
```

Upgrade Database

Upgrade - Nacharbeiten

Nacharbeiten

```
shutdown immediate;  
startup upgrade;  
EXEC DBMS_DST.BEGIN_UPGRADE (31);  
shutdown immediate;  
startup;  
select * from v$timezone_file;
```

Siehe preupgrade.log:

Upgrade the database time zone file using the DBMS_DST package.
The database is using time zone file version 14 and the target 18.0.0.0.0 release ships with **time zone file version 31**.

```
alter profile default limit password_life_time unlimited;
```

Upgrade Database

Upgrade - Nacharbeiten

Nacharbeiten

```
set linesize 100
set pagesize 0
col comp_id format a10
col comp_name format a30
col version format a10
col status format a20
select substr(comp_id,1,15) comp_id,substr(comp_name,1,30)
comp_name,substr(version,1,10) version,status
from dba_registry;
```

```
@/u01/app/oracle/cfgtoollogs/DBR11/preupgrade/postupgrade_fixups.sql;
```


Upgrade Database

Upgrade - Nacharbeiten

Nacharbeiten

```
alter system set COMPATIBLE = '18.3.0' SCOPE=SPFILE;  
shutdown immediate;  
startup;
```

Upgrade Database

Upgrade - Nacharbeiten

Nacharbeiten

```
EXECUTE DBMS_STATS.GATHER_SCHEMA_STATS
```

```
EXECUTE DBMS_STATS.GATHER_DICTIONARY_STATS;  
Wenn repräsentative Systemlast vorhanden ist:  
EXECUTE DBMS_STATS.GATHER_FIXED_OBJECTS_STATS;
```

Prüfen von:

- /etc/oratab
- listener.ora
- tnsnames.ora
- sqlnet.ora

Upgrade Database

Upgrade der Datenbank abgeschlossen !

ORACLE 18c



Der Zugriff auf die Datenbank ist wieder möglich
→ Ende der Downtime

Fragen ?

