

# Oracle Plattform Migration

## “cross platform transportable tablespaces (XTTS)”

### Methode und Erfahrungsbericht

**Josef Lippert**  
**Freiberuflicher IT Consultant**  
**München**

#### **Schlüsselworte:**

Migration, Transportable Tablespace, Transportable Database, RMAN

#### **Einleitung**

Oracle Plattform Migration ermöglicht das Verschieben der Datenbank zwischen verschiedenen Betriebs- und Hardware-Systemen.

Der „herkömmliche“ Weg über Export/Import/DataPump der Daten birgt ein erhebliches zeitliches Risiko. Mit dem Oracle Data Guard steht eine weitere Migrationsmethode zur Verfügung, die aber nur bei unterstützten Plattformen möglich ist und das gleiche „endian“-Format aufweisen. Siehe hierzu:

„Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration [ID 413484.1]”

Transportable Tablespaces sind seit Oracle 8i verfügbar und wurden seitdem stetig funktional erweitert. Mit Oracle 10g R2 wurden „transportable database (TDB)“ und „cross platform transportable tablespaces (XTTS)“ eingeführt. Letztere erlaubt neben der Datenmigration auch eine Datenkonvertierung unterschiedlicher „endian“-Typen. Die Datenkonvertierung ist z. B. zwischen HP-UX/Solaris/AIX und Linux/Windows erforderlich.

Während der Migration steht die Source-Datenbank „Read Only“ zur Verfügung.

#### **Migrationsmethode - XTTS**

Folgende Idee liegt der Migrationsmethode zu Grunde (siehe Abb. 1):

- Neue, leere Datenbank auf dem Zielsystem anlegen
- Meta-Daten extrahieren
- Nutz-Daten für Zielplattform konvertieren (siehe Abb. 2)
- Nutz-Daten (User-Tablespaces) transportieren
- Meta-Daten importieren

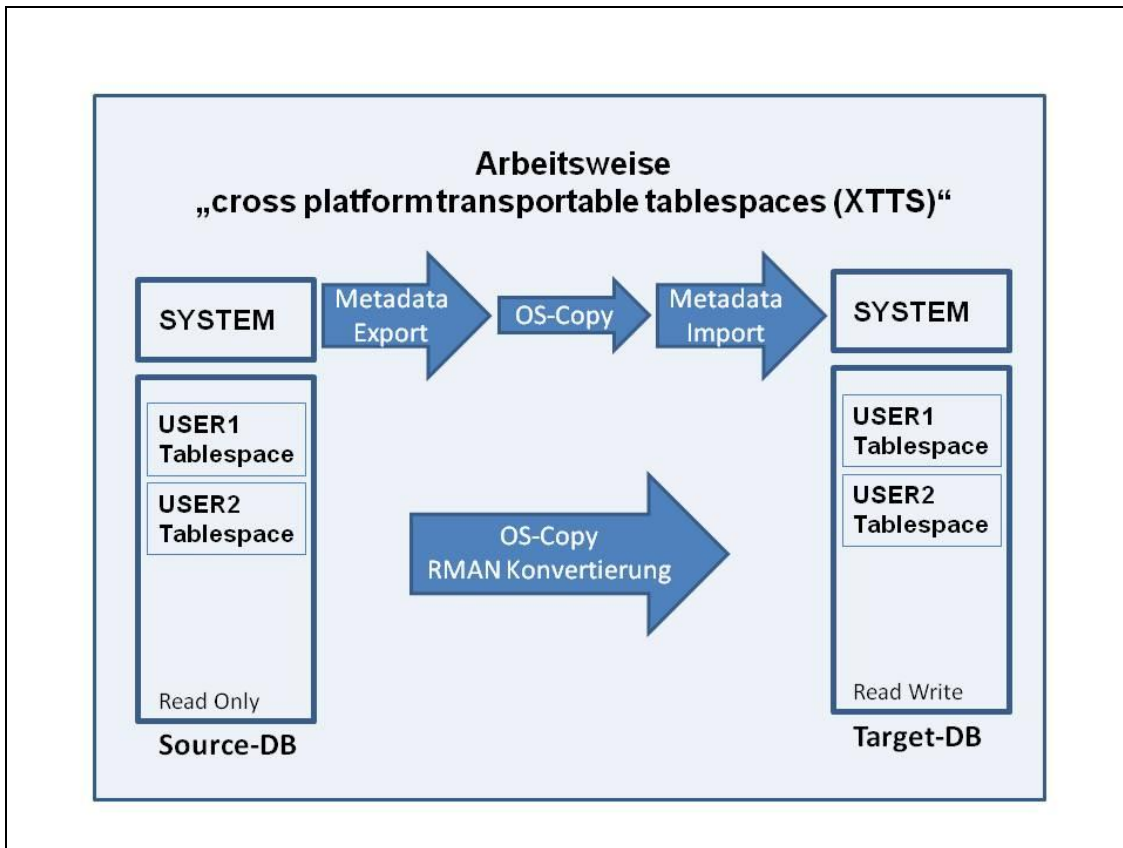


Abb. 1: Arbeitsweise „cross platform transportable tablespaces (XTTS)“

```

2 FROM V$TRANSPORTABLE_PLATFORM order by PLATFORM_NAME;

```

PLATFORM_NAME	ENDIAN_FORMAT
AIX-Based Systems (64-bit)	Big
Apple Mac OS	Big
Apple Mac OS (x86-64)	Little
HP IA Open VMS	Little
HP Open VMS	Little
HP Tru64 UNIX	Little
HP-UX (64-bit)	Big
HP-UX IA (64-bit)	Big
IBM Power Based Linux	Big
IBM zSeries Based Linux	Big
Linux IA (32-bit)	Little
Linux IA (64-bit)	Little
Linux x86 64-bit	Little
Microsoft Windows IA (32-bit)	Little
Microsoft Windows IA (64-bit)	Little
Microsoft Windows x86 64-bit	Little
Solaris Operating System (x86)	Little
Solaris Operating System (x86-64)	Little
Solaris[tm] OE (32-bit)	Big
Solaris[tm] OE (64-bit)	Big

20 rows selected.

Abb. 2: Endian Typen

```

CONVERT DATAFILE
'/oracle/SRC/tools01.dbf',
'/oracle/SRC/user01.dbf',
FROM PLATFORM 'Linux x86 64-bit'
PARALLELISM 2
DB_FILE_NAME_CONVERT '/oracle/SRC' , '/oracle/TGT'

```

Abb. 3: Beispiel RMAN Konvertierung

## Migrations-Arbeitsschritte im Einzelnen

Ziel-Datenbank mit allen genutzten Optionen und entsprechenden Character-Set/National-Character-Set aufbauen

- InSource-Datenbank:
  - Analyse, ob Tablespace im „Set“ transportiert werden kann (siehe Abb. 4)
  - „invalide“ Objekte kompilieren
  - „Recyclebin“ leeren
  - In Source-Datenbank Tablespaces „Read Only“ setzen
  - Evtl. Sequence Informationen ermitteln
  - Externe File: dbms\_tdb.check\_external, dbms\_lob.filegetname
  - Meta-Daten exportieren
  - Non-Segment User Daten exportieren
  - Weitere Daten (z. B. AWR) exportieren (siehe Abb. 5)
  - Create-Skripte für Rollen und User erzeugen
  - Evtl. create-Skripte für Advanced Queueing Settings
  
- In Ziel-Datenbank:
  - Filesysteme via NFS mounten
  - Datenfiles via RMAN konvertieren
  - User und Rollen anlegen
  - Meta-Daten importieren
  - Non-Segment User Daten importieren
  - Evtl. weitere Daten importieren (AWR, ...)
  - Evtl. Advanced Queueing Settings, etc.
  - Evtl. Sequence Information anpassen
  - Parameter Dateien, externe Tabellen, „bfiles“, etc. anlegen/transferieren
  - „startup upgrade“ mit utlirp.sql und utlrp.sql
  - Daten/Inhaltsvergleich zwischen Source- und Ziel-DB durchführen
  - tnsnames/LDAP anpassen

```

SQL> exec dbms_tts.transport_set_check('TOOLS, USERS',TRUE,TRUE);
SQL> select * from transport_set_violations;

VIOLATIONS
-----
ORA-39908: Index DEFUSER.TEST_PK in tablespace IDXTSP enforces
primary constraints of table DEFUSER.ORA_TEST_TEMPLATE in tablespace
TOOLS.

```

Abb. 4: Tablespace Transport-Anhängigkeit prüfen

```

SQL> SELECT      occupant_name, schema_name, move_procedure, space_usage_kbytes USAGE_KB
2 FROM          v$sysaux_occupants
3 ORDER BY     occupant_name;

```

OCCUPANT_NAME	SCHEMA_NAME	MOVE_PROCEDURE	USAGE_KB
AO	SYS	DBMS_AW.MOVE_AWMETA	1408
AUDIT_TABLES	SYS	DBMS_AUDIT_MGMT.move_dbaudit_tables	0
AUTO_TASK	SYS		320
EM	SYSMAN	emd_maintenance.move_em_tblspc	0
EM_MONITORING_USER	DESNMP		2432
EXPRESSION_FILTER	EXFSYS		0
JOB_SCHEDULER	SYS		384
LOGMNR	SYSTEM	SYS.DBMS_LOGMNR_D.SET_TABLESPACE	8064
LOGSTDBY	SYSTEM	SYS.DBMS_LOGSTDBY.SET_TABLESPACE	1408
ORDIM	ORDSYS	ordsys.ord_admin.move_ordim_tblspc	0
ORDIM/ORDDATA	ORDDATA	ordsys.ord_admin.move_ordim_tblspc	0
ORDIM/ORDPLUGINS	ORDPLUGINS	ordsys.ord_admin.move_ordim_tblspc	0
ORDIM/SI_INFORMTN_SCHEMA	SI_INFORMTN_SCHEMA	ordsys.ord_admin.move_ordim_tblspc	0
PL/SCOPE	SYS		1600
SDO	MDSYS	MDSYS.MOVE_SDO	0
SM/ADVISOR	SYS		24192
SM/AWR	SYS		85760
SM/OPTSTAT	SYS		59776
SM/OTHER	SYS		5952
SMON_SCN_TIME	SYS		3264
SQL_MANAGEMENT_BASE	SYS		1728
STATSPACK	PERFSTAT		0
STREAMS	SYS		1024
TEXT	CTXSYS	DRI_MOVE_CTXSYS	0
TSM	TSM SYS		0
ULTRASEARCH	WKSYS	MOVE_WK	0
ULTRASEARCH_DEMO_USER	WK_TEST	MOVE_WK	0
WM	WMSYS	DBMS_WM.move_proc	7360
XDB	XDB	XDB.DBMS_XDB.MOVEXDB_TABLESPACE	57728
XGAMD	OLAPSYS	DBMS_AMD.Move_OLAP_Catalog	0
XSOQHIST	SYS	DBMS_XSOQ.OlapiMoveProc	1408

31 rows selected.

Abb. 5: V\$SYSAUX\_OCCUPANT\_NAME – Move procedures

## Limitierungen der Migrationsmethode

- In Source- und Ziel-Datenbank müssen der Character-Set, National-Character-Set, „compatible“-Parameter sowie die Oracle Optionen identisch sein
- Encryption-Columns werden nicht unterstützt
- Migrations-Tablespaces müssen „self-contained“ sein
- Advanced Queue Einstellungen werden nur teilweise migriert
- Binary\_Float und Binary\_Double nur mit Datapump
- Tabellen auf den Tablespaces SYSTEM, UNDO, SYSAUX und TEMP können nicht transportiert werden

- Externe Objekte (parameter files, external tables, bfiles, etc.) müssen manuell bewegt werden
- SYS Objekte (packages, views, java classes, sequences, etc.) werden nicht transportiert

## Erfahrungsbericht

Die XTTS-Methode wurde bislang auf ca. 100 Datenbanken unterschiedlicher Größen von 5 GB bis 2 TB erfolgreich angewandt. Für alle aufgetretenen „Umstände“ konnte eine Lösung/Workaround gefunden werden.

Wichtig hier ist, im Vorfeld die Source-Datenbank stets hinsichtlich bekannter Probleme zu untersuchen. Hierzu zählen „typische“ bekannte Oracle Bugs oder Erkenntnisse, die sich während der Migration selbst ergeben.

Folgende Fehler/Probleme/Bugs traten auf und verlangten „händische“-Eingriffe:  
(Anmerkung: Die Aufzählung erhebt keinen Anspruch auf Vollständigkeit und ist vom Oracle Release-Stand abhängig.)

- Tabellen mit dem Datentyp „ROWDEPENDENCIES“
  - Patch 8229276 muss eingebracht sein
- Tabellen mit dem Datentyp „BINARY\_FLOAT“ und „BINARY\_DOUBLE“
  - „nur“ mit Datapump zu bewegen
- Oracle Text-Objekte auf dem Tablespace „SYSAUX“
  - Siehe V\$SYSAUX\_OCCUPANT\_NAME
- Oracle Text User Index Preferences
  - u. U. manuelle Nacharbeiten
- XML Data
  - In 10g „nur“ mit export/import zu bewegen
- Indexe auf Temp-Tables
  - In 10g nicht mit transportiert
- Komprimierte Tabellen
  - Korruptionen traten auf
- Advanced Queueing Settings
  - u. U. manuelle Nacharbeiten
- RMAN: „CONFIGURE DEVICE TYPE DISK PARALLELISM ...“
  - Device Type muss „Disk“ sein

Für die Migration selbst wurde auf der Ziel-Datenbank „db\_block\_checking“ und „db\_block\_checksum“ eingeschaltet um potentielle Datenkorruptionen rechtzeitig zu entdecken.

Aus Performancegründen ist im Nachgang der Parameter „db\_block\_checking“ zu deaktivieren.

Die Migrationsdauer wird von mehreren Größen beeinflusst:

- Größe des Data-Dictionary
  - z. B. Partitioning Option
- CPU-Performance für die „endian“-Typ-Konvertierung mit dem RMAN
- Parallelität der Konvertierung
- Performance I/O Subsystem
  - z. B. FILESYSTEMIO\_OPTIONS = { none | setall | directIO | asynch }
- Netz-Performance für den Filetransfer
- Einsatz von „huge pages“ prüfen
  - [http://en.wikipedia.org/wiki/Huge\\_pages#Huge\\_pages](http://en.wikipedia.org/wiki/Huge_pages#Huge_pages)

Nach der Migration empfiehlt sich u. a. das DB Cache Sizing zu betrachten, da dies u. U. auf den verschiedenen Plattformen unterschiedlich behandelt wird.

Ein möglicher Ansatzpunkt ist der Cache Advisor:

```
alter system set db_cache_advice = on scope=memory;
SELECT size_for_estimate, buffers_for_estimate,
       estd_physical_read_factor, estd_physical_reads/1000
FROM v$db_cache_advice
WHERE advice_status = 'ON';
```

Cache Size (MB)	Estd Phys		
	Buffers	Read Factor	ESTD_PHYSICAL_READS/1000
3,584	443,296	1.01	7070.031
4,032	498,708	1.00	7049.6
4,480	554,120	1.00	7022.63
4,928	609,532	.98	6890.872
5,376	664,944	.95	6646.521
5,824	720,356	.87	6140.577
6,272	775,768	.81	5722.296
6,720	831,180	.79	5564.413
7,168	886,592	.78	5501.977
7,616	942,004	.78	5494.624
8,064	997,416	.78	5480.898
8,512	1,052,828	.78	5476.055
8,960	1,108,240	.78	5448.579

Abb. 6: v\$db\_cache\_advice

### Methodischer Ablaufplan

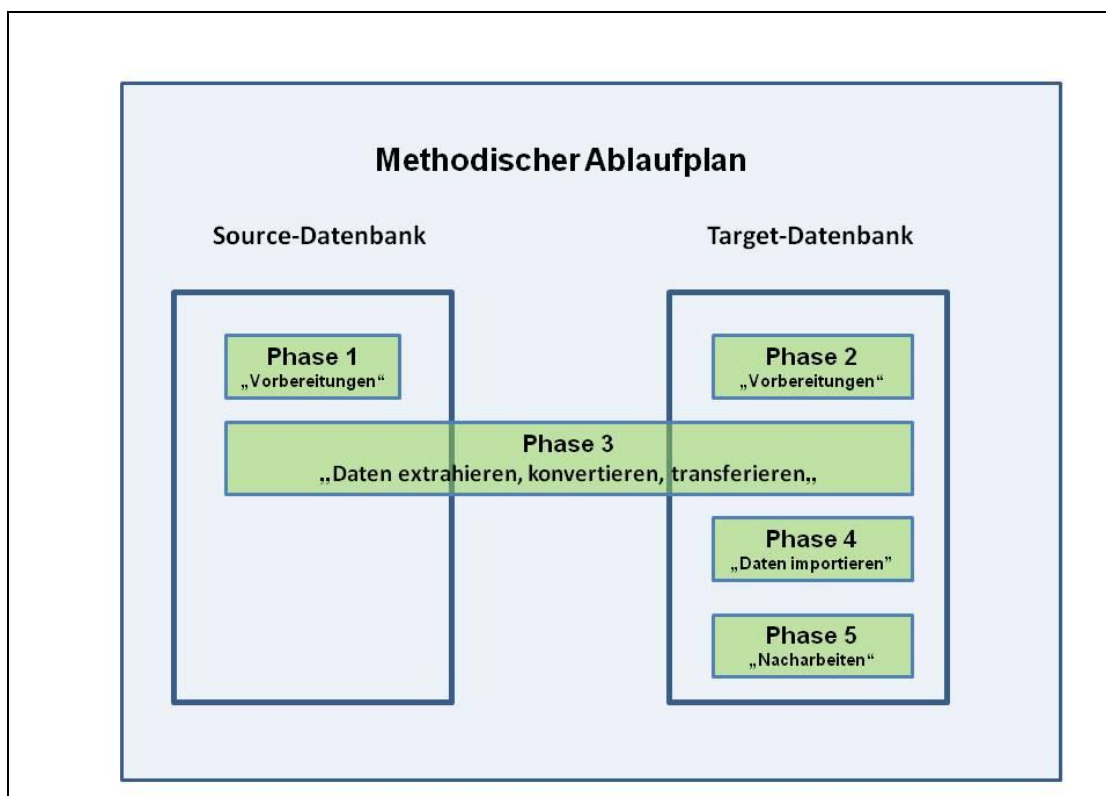


Abb. 7: Ablaufplan

## Fallback Szenarien

### Szenario 1:

Die Source-Datenbank wird inhaltlich nicht modifiziert. Während der Migration werden die User-Tablespace auf „Read Only“ gesetzt und müssen somit beim Fallback nur wieder auf „Read Write“ gebracht werden.

D. h. in relativ kurzer Zeit steht das Ausgangssystem unverändert wieder zur Verfügung.

Zu beachten ist aber, dass nach dem Fallback die im „neuen“ Zielsystem bereits modifizierten Daten verloren sind.

### Szenario 2:

Dieses Szenario bietet sich an, wenn die Daten in der Ziel-Datenbank bereits „erheblich“ modifiziert wurden.

Durch Rück-Konvertierung der Ziel-Datenbank mittels der gleichen Migrationsmethode werden die modifizierten Daten in die Ausgangsdatenbank gebracht.

Hierdurch entsteht aber eine erneute Downtime sowie ein meist hohes Risiko, da dieser „Rückweg“ oft nicht hinreichend getestet beschritten wird.

### Kontaktadresse:

Name  
Josef Lippert  
IT Beratung  
München

Telefon: +49 (0) 171-7964 133  
E-Mail [Josef.Lippert@jal-bs.de](mailto:Josef.Lippert@jal-bs.de)  
Internet: [www.jal-bs.de](http://www.jal-bs.de)