



## Distribution von Updates mittels Transbase® CD

### 1. Transbase® CD

Transbase® CD ist eine Variante von Transbase®, die direkt auf Readonly-Speichern wie DVD oder CD arbeiten kann und trotzdem volle SQL Funktionalität ermöglicht (einschließlich aller Updates).

Dies wird erreicht durch einen ins System integrierten Verschattungsmechanismus, der ROM-Seiten bei ihrer (ersten) Veränderung auf die Festplattenpartition verlagert und damit deren ROM-Version dauerhaft verschattet. Dieser Mechanismus ist persistent und transaktionsgesichert.

Die Festplattenpartition kann auch genutzt werden, um häufig benötigte Seiten dort zu cachen, deswegen wird diese Partition auch als Festplattencache bezeichnet.

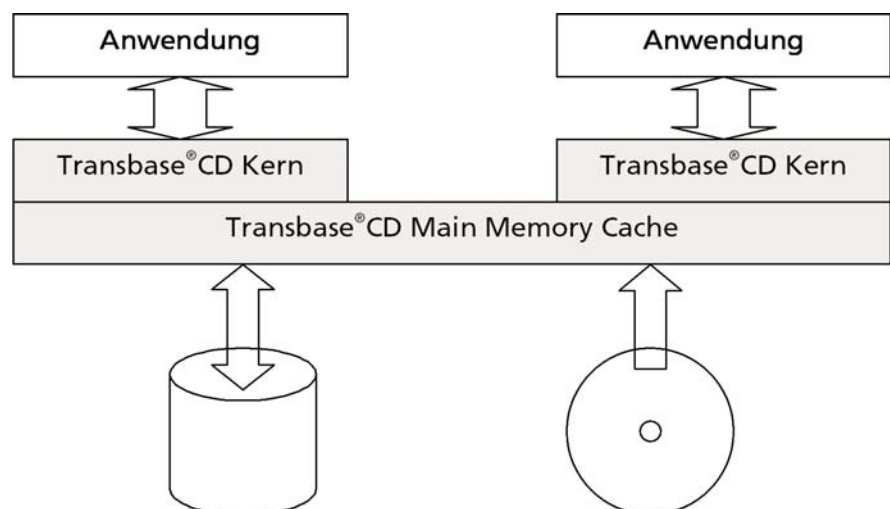


Abbildung 1:

Transbase® CD



Initial ist der Festplattencache leer, die organisatorischen Seiten betragen weniger als 1MB. Mit jeder Änderungstransaktion werden geänderte Seiten von ROM nach DISK verlagert, so dass mit der Zeit (und der Menge der Änderungen) der Platzbedarf der Festplattenpartition wächst.

Die Einheit der Verschattung ist eine Datenbankseite; um einen kleinen Datensatz zu ändern, muss also eine ganze Seite in die Festplattenpartition wandern. Treffen Änderungen mehrfach die gleiche Seite (sind also "lokal"), so ist kein weiterer Festplattenplatz erforderlich.

Die Möglichkeit von Änderungen in Transbase® CD Datenbanken kann sowohl für Änderungen, die den globalen Datenbestand betreffen, als auch für lokale Änderungen verwendet werden, z.B. für individuelle Leselisten, Anmerkungen etc. Es sollte allerdings darauf geachtet werden, dass diese lokalen Änderungen in separaten Tabellen oder sogar in separaten Datenbanken stattfinden.

## 2. Ausgangssituation

---

Im folgenden gehen wir von folgender Ausgangssituation für inkrementelle Updates aus:

- Eine (Transbase®) Datenbank PUB wird zentral gefüllt mit dem zu publizierenden Datenbestand.
- Die Datenbank PUB in ihrer Version PUB<sub>0</sub> wird per Distribution von verschlüsselten und komprimierten ROM-Files auf Datenträger physisch verteilt.
- Updates, die nach Distribution von PUB<sub>0</sub> entstehen, werden in PUB gesammelt, bis ein Stand PUB<sub>1</sub> entsteht, der auf die publizierten Datenbanken propagiert werden soll.
- PUB<sub>1</sub> wird als Delta zu PUB<sub>0</sub> distribuiert (online oder über kleine Datenträger).
- Updates, die nach Distribution von PUB<sub>1</sub> entstehen, werden in PUB gesammelt, bis ein Stand PUB<sub>2</sub> entsteht, der auf die publizierten Datenbanken propagiert werden soll.
- PUB<sub>2</sub> könnte als "Inkrement" zu PUB<sub>1</sub> distribuiert werden oder als "Delta" zu PUB<sub>0</sub>; nach unserer Erfahrung ist ein Delta zu PUB<sub>0</sub> wesentlich robuster, weil dieses auf alle verteilten Datenbanken angewendet werden kann, ohne dass deren Stand bekannt sein müsste. Ein Inkrement zu PUB<sub>1</sub> erfordert, dass der empfangende Client bereits alle vorherigen Inkremente (hier zu PUB<sub>0</sub>) installiert hat - was in großen Organisationen wohl nicht erzwungen werden kann.



### 3. Datenbankdifferenzen

---

Um ein Delta zwischen zwei Ständen  $PUB_n$  und  $PUB_0$  (oder das Inkrement zwischen Ständen  $PUB_n$  und  $PUB_{n-1}$ ) zu ermitteln, stellt Transbase® ein Tool TBDIFF zur Verfügung. TBDIFF vergleicht zwei Datenbanken tupelweise und liefert ein SQL-Skript, welches die eine Datenbank in die andere überführt. Dieses Skript enthält DELETE Statements für alle gelöschten und geänderten Tupel sowie SPOOL Statements für alle neuen und geänderten Tupel und sogar CREATE oder DROP TABLE Statements. Auch geänderte Tupel mit BLOBs werden jeweils komplett gelöscht und durch neue ersetzt.

Die Größe des SQL-Skripts kann – je nach Art der Änderung – groß werden. Zu ersetzende BLOBs können wesentlich in die Größe des Updates eingehen.

Zu beachten ist, dass natürlich nur die Änderungen der Primärtupel beschrieben zu werden brauchen, da die Sekundärindexwartung implizit beim Anwenden der Statements erfolgt.

Diese SQL-Skript kann nun entweder:

- zu jedem Client verschickt werden (ggf. in komprimierter Form) und dort auf seinen Datenbankstand angewendet werden (siehe 4.) oder
- auf einem Referenzclient angewendet werden und dessen Diskpartition in ein sogenanntes Delta-ROM-File überführt werden. Dieses Delta-ROM-File verschattet dann das ursprüngliche ROM-File (siehe 5.)

### 4. SQL Skript

---

Wird das SQL Skript zur Propagierung von Updates verwendet, ist folgendes zu beachten:

- Das Skript muss lokal auf der Client-DB ausgeführt werden.
- Handelt es sich um ein Delta zu  $PUB_0$ , so muss sichergestellt sein, dass die Client-DB sich im Zustand  $PUB_0$  befindet; dies kann durch ein einfaches Reattach-Kommando erreicht werden. Handelt es sich um ein Inkrement zu  $PUB_{n-1}$ , so muss sichergestellt sein, dass die Client-DB sich im Zustand  $PUB_{n-1}$  befindet; dies muss durch die Anwendung gewährleistet werden.
- Das SQL-Skript kann entweder durch die Anwendung ausgeführt werden (einfache Syntax) oder durch einen Transbase® Query Prozessor (z.B. TBI). Die Laufzeit des Skripts hängt wiederum von Anzahl und Art der Änderungen ab. Dabei entstehen temporäre Dateien (zu Zwecken der Recovery) in u.U. erheblichem Ausmaß; die Festplattenpartition wächst ebenfalls, da jede geänderte Seite dort gespeichert wird. Insbesondere ist zu berücksichtigen, dass Seiten in der Festplattenpartition nicht mehr komprimiert sind.



## 5. Page Delta

### 5.1. SQL Prozess auf Publikationsserver

Bei Verwendung von Page-Deltas läuft der Update-Prozess auf einem Referenz-Client-System. Dort ergibt sich nach Ausführung des SQL-Skripts ein Diskfile, welches entweder als Diskfile zu jedem Client transportiert werden kann (wenig gebräuchlich) oder mittels des Transbase® Tools TBMKROM in ein Delta-ROM-File verwandelt werden kann, welches wieder komprimiert ist.

Diese Delta-ROM-File verschattet wiederum das originale ROM-File.

Mit Delta-ROM-Files ist nur ein Delta von PUB<sub>0</sub> nach PUB<sub>n</sub> möglich.

### 5.2. Delta-ROM Files im Client

Der Client installiert Delta-ROM-Files, indem er einen Reattach der ursprünglichen Datenbank durchführt und dabei das Delta-ROM-File einspielt.

Beim Reattach gehen sämtliche vorherigen Änderungen verloren (insbesondere die lokalen Änderungen). Sollen diese Änderungen über den Update hinaus verfügbar bleiben, so sind solche Daten vorher zu retten.

Zu beachten ist, dass client-seitig kaum Rechenzeit und – außer dem Retten der lokalen Daten – kein zusätzlicher Plattenplatz benötigt wird. Ferner ist hervorzuheben, dass die Daten im Delta-ROM-File komprimiert sind.

## 6. Erfahrungen

Die Erfahrungen unserer Kunden zeigen, dass Delta-Verfahren grundsätzlich robuster als Inkrement-Verfahren sind. In der großen Mehrzahl der Fälle werden Delta-ROM-Files eingesetzt, weil client-seitig einfachere Prozesse entstehen, weniger Rechen- und Plattenressourcen benötigt werden, kurzum der Prozess robuster ist.

Als Referenzbeispiel können genannt werden:

Kunde	Dienstleister	Zyklus	Updateverfahren
Bosch	Empolis	Vierteljährlich	Delta ROM
Daimler Chrysler	HP	Monatlich	Delta ROM
Daimler Chrysler	ProQuest		Delta ROM

### Kontakt

Transaction Software GmbH  
 Willy-Brandt-Allee 2  
 81829 München

Tel.: +49 89 / 627 09 - 0  
 Fax: +49 89 / 627 09 - 11

info@transaction.de  
 www.transaction.de  
 www.transbase.de