



# Data Reloc

Anhang A:

**Vorstudie**



## Inhaltsverzeichnis

<b>I. ANFORDERUNGEN UND FUNKTIONEN.....</b>	<b>3</b>
<b>II. KLASSEN.....</b>	<b>4</b>
1. TDBSESSION.....	4
2. TRELOC.....	4
3. TREPFILE.....	5
<b>III. PROGRAMMABLAUFPLÄNE.....</b>	<b>6</b>
1. TEST IMPORT/EXPORT VERFAHREN.....	6
2. IMPORT/EXPORT VERFAHREN.....	7

## I. Anforderungen und Funktionen

- Ausführbar auf den Betriebssystemen Windows 2000 und Windows XP
- Direkter Import/Export von Daten zwischen Datenbanksystem
- Unterstützung aller in Betrieb eingesetzte Datenbanksysteme
- Möglichkeit Datenbankverbindungen und Import/Export Verfahren zu speichern
- Jedes Import/Export Verfahren muss mindestens eine Quelltable und eine Zieltabelle haben
- Quelltable muss durch eine SQL Abfrage selektiert werden
- Zieltabelle muss direkt mit einer Tabelle im Zieldatenbanksystem verknüpft sein
- Zuordnung der Spalten in der Quelltable zu den Spalten in der Zieltabelle per „Drag and Drop“
- Möglichkeit der Zuordnung eines zusätzlichen festen Wertes einer Spalte in der Zieltabelle
- Möglichkeit die Import/Export Verfahren vor der Ausführung auf Datentypkompatibilität zu prüfen
- Einfache Datentypkonvertierungen (z.B. Dezimalzahl zu Ganzzahl) sollen von der Anwendung automatisch durchgeführt werden
- Operationen beim Import/Export Verfahren sollen automatisch protokolliert werden. Es soll möglich sein das Protokoll in verschiedenen Formaten abzuspeichern
- Im Protokoll sollen folgende Ergebnisse der Import/Export Verfahren enthalten sein:
  1. Anzahl der zu importierenden Datensätze
  2. Anzahl der importierten Datensätze
  3. Anzahl der nicht importierten Datensätze
- Datensätze die nicht importiert wurden sollen zwischengespeichert und am Ende jedes Import/Export Verfahren ausgegeben werden. Fehlerhafte Datensätze sollen abgespeichert werden können

## II. Klassen

### 1. TDBSession

Datenbanksitzung

Stellt die Verbindung zwischen Anwendung und Datenbank her.

TDBSession = class (TObject)	
ATTRIBUTE:	
+ DatabaseName	String
+ DatabaseHost	String
+ DatabaseType:	TDatabaseType
+ Username:	String
+ Password:	String
EREIGNISSE:	
+ OnConnect	TNotifyEvent
+ OnDisconnect	TNotifyEvent
METHODEN:	
+ procedure Disconnect	
+ function Connect: Boolean	

### 2. TReloc

Import/Export Verfahren (*Reloc Operation*).

Testet Import/Export Verfahren und führt sie durch. Erstellt und speichert Protokollnachrichten.

TReloc = class(TObject)	
ATTRIBUTE:	
+ SourceDBConnection:	TDBSession
+ SourceSQL:	String
+ TargetDBConnection:	TDBSession
+ TargetTableName:	String
+ SourceFieldsTargetFieldsLinks:	Array of Array of String
+ Log:	TStrings
EREIGNISSE:	
+ OnInsertError:	TNotifyEvent
METHODEN:	
+ procedure SaveLog(aFileName: TFileName)	
+ procedure PerformReloc	
+ function TestReloc:	Boolean

### 3. TRepFile

Arbeitsumgebung (*Repository*)

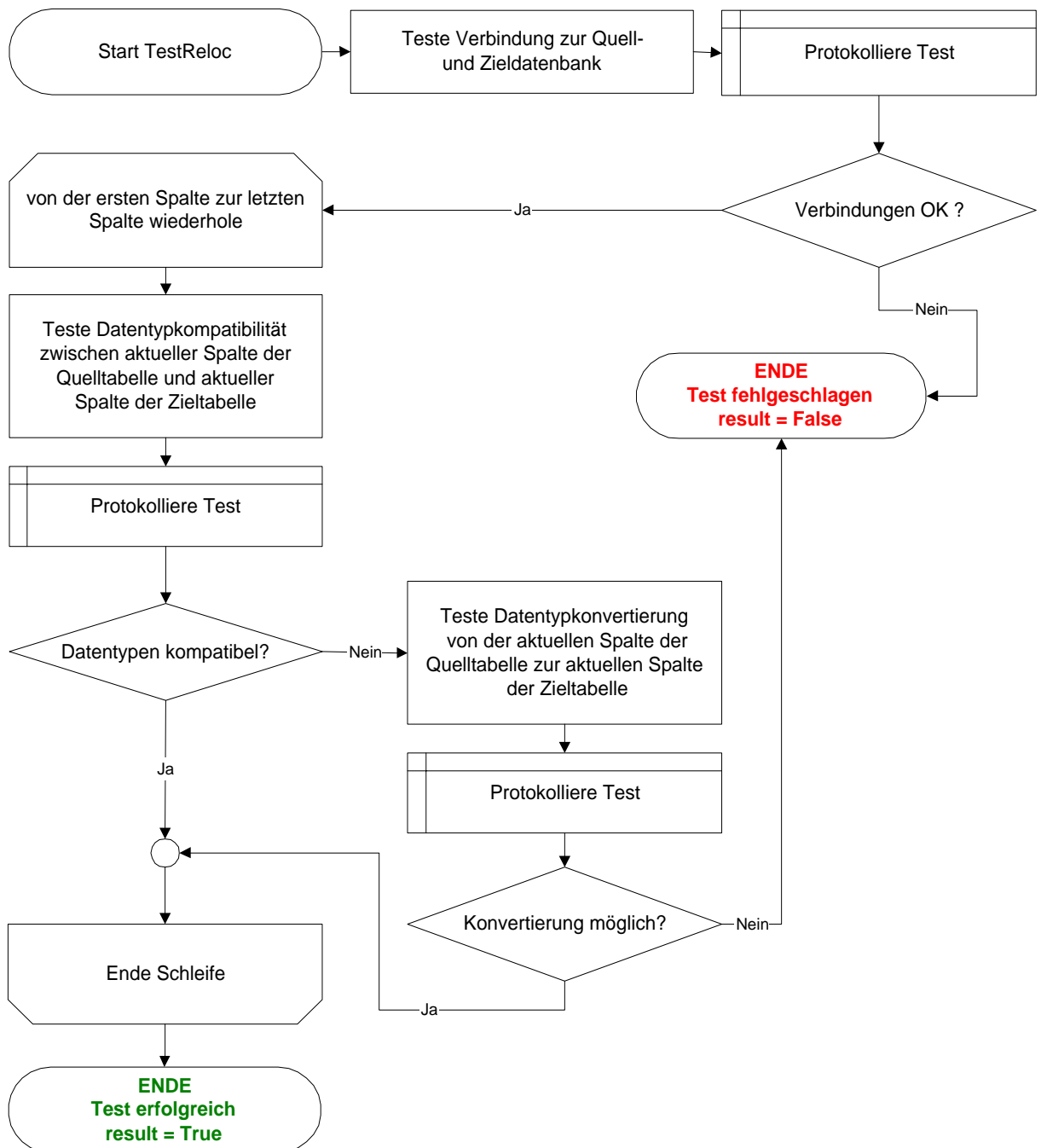
Verwaltet und speichert mehrere Datenbanksitzungen und Import/Export Verfahren.

TRepFile = class(TObject)	
ATTRIBUTE:	
+ RepositoryName:	String
+ FileName:	String
+ DBSessions:	Array of TDBSession
+ DBSessionCount:	Integer
+ Relocs:	Array of TReloc
+ RelocCount:	Integer
EREIGNISSE:	
+ OnSave:	TNotifyEvent
METHODEN:	
+ procedure Save;	
+ function AddReloc(aReloc: TReloc): Boolean	
+ function AddDBSession(aDBSession: TDBSession): Boolean	
+ function DeleteReloc(aRelocName: String): Boolean	
+ function DeleteDBSession(aDBSessionName: String): Boolean	
+ constructor Create(aFileName: TFileName)	

### III. Programmablaufpläne

#### 1. Test Import/Export Verfahren

**function TestReloc: Boolean**



## 2. Import/Export Verfahren

### procedure PerformReloc

