



# Data Reloc

Anhang C:

## Klassendokumentation



## Inhaltsverzeichnis

<b>1</b>	<b>KLASSEN</b> .....	<b>3</b>
1.1	<b>TDBSESSION</b> .....	<b>3</b>
1.1.1	Attribute.....	4
1.1.2	Ereignisse .....	5
1.1.3	Methoden .....	5
1.2	<b>TDBSESSIONS</b> .....	<b>6</b>
1.3	<b>TRELOC</b> .....	<b>6</b>
1.3.1	Attribute.....	7
1.3.2	Ereignisse .....	9
1.3.3	Methoden .....	10
1.4	<b>TRELOCS</b> .....	<b>10</b>
1.5	<b>TREPFILE</b> .....	<b>11</b>
1.5.1	Attribute.....	11
1.5.2	Ereignisse .....	13
1.5.3	Methoden .....	13
<b>2</b>	<b>STRUCTS, RECORDS, ENUMS</b> .....	<b>16</b>
2.1	<b>TDBSESSIONTYPE</b> .....	<b>16</b>
2.2	<b>TDATASOURCEMETHOD</b> .....	<b>16</b>
2.3	<b>TRELOCOPERATIONTYPE</b> .....	<b>17</b>
2.4	<b>TLOGMESSAGE</b> TYPE.....	<b>17</b>
<b>3</b>	<b>METHODENTYPEN</b> .....	<b>18</b>
3.1	<b>TCONVERTFUNCTION</b> .....	<b>18</b>
3.2	<b>TRELOCLOGNOTIFYEVENT</b> .....	<b>18</b>
3.3	<b>TRELOCONDBSESSIONNAMECHANGE</b> .....	<b>18</b>
3.4	<b>TRELOCONINSERTERROR</b> .....	<b>19</b>

## 1 Klassen

### 1.1 TDBSession

<b>TDBSession = class(TObject)</b>	
<b>ATTRIBUTE:</b>	
# DBConnection:	TADOCnncetion
+ DBSessionType:	TDBSessionType
+ DataSourceMethod:	TDataSourceMethod
+ ConnectString:	String
+ SessionName:	String
<b>EREIGNISSE:</b>	
+ OnConnect	TNotifyEvent
+ OnDisconnect	TNotifyEvent
<b>METHODEN:</b>	
+ procedure Disconnect	
+ function Connect: Boolean	
+ function BuildConnectionString: Boolean	

Die Klasse *TDBSession* stammt von der Klasse *TObject* und enthält alle erforderliche Informationen und Funktionen zur Herstellung einer Verbindung zu einem Datenbanksystem (Datenbanksitzung). *TDBSession* stellt die Verbindung zwischen der Anwendung und einer beliebigen Datenquelle durch einen entsprechenden OLE DB Treiber von Microsoft.

Sofern ein passender OLE DB Treiber verfügbar ist, ist es möglich zu jeder Datenquelle zuzugreifen, sei es ein komplexes Datenbanksystem wie ORACLE oder eine einfache CSV-Datei. Hierdurch ist es möglich Daten auszulesen und/oder einzufügen.

### 1.1.1 Attribute

#### TDBSession.DBConnection

```
property DBConnection: TDBConnection read vDBConnection;
```

Objekt vom Typ *TADOConnection*. Stellt die Verbindung zur Datenquelle her, gemäß *TDBSession.ConnectString*. Dieses Nur „Lese-Attribut“ ist geschützt und wird von *TReloc* (siehe 1.3) benutzt um Daten auszulesen und/oder einzufügen.

#### TDBSession.DBSessionType

```
property DBSessionType: TDBSessionType read vDBSessionType write vDBSessionType;
```

Öffentliches Attribut vom Typ *TDBSessionType* (siehe 2.1). Gibt an, ob die Daten von ein ORACLE Datenbanksystem kommen oder durch eine ODBC-Verbindung bzw. durch eine OLE DB Verknüpfung bezogen werden.

#### TDBSession.DataSourceMethod

```
property DataSourceMethod: TDataSourceMethod read vDataSourceMethod  
write vDataSourceMethod;
```

Öffentliches Attribut vom Typ *TDataSourceMethod* (siehe 2.2). Gibt an, ob die Daten durch eine SQL-Abfrage gelesen werden oder ob eine direkte Verbindung zu einer bestimmten Tabelle existiert. Daten können nur eingefügt werden, wenn eine direkte Verbindung zu einer Tabelle verfügbar ist. Hiermit wird es eindeutig gemacht, dass Daten von einer oder mehreren Tabellen stammen.

#### TDBSession.ConnectionString

```
property ConnectString: String read vConnectionString write vConnectionString;
```

Öffentliches Attribut vom Typ *String*. Enthält die Verbindungsparameter wie Username und Passwort um die Verbindung zur Datenquelle herzustellen.

#### TDBSession.SessionName

```
property SessionName: String read vSessionName write vSessionName;
```

Öffentliches Attribut vom Typ *String*. Enthält einen eindeutigen Sitzungsname. Wird von den Klassen *TRepFile* (siehe 1.5) und *TReloc* (siehe 1.3) benutzt, um Datenbanksitzungen eindeutig zu erkennen und auf diese zuzugreifen.

## 1.1.2 Ereignisse

### TDBSession.OnConnect

```
property OnConnect: TNotifyEvent read vOnConnect write vOnConnect;
```

Ereignis von Typ *TNotifyEvent*. Wird ausgelöst nach erfolgreicher Verbindung zur Datenquelle.

### TDBSession.OnDisconnect

```
property OnDisconnect: TNotifyEvent read vOnDisconnect write vOnDisconnect;
```

Ereignis von Typ *TNotifyEvent*. Wird ausgelöst bevor die Verbindung zur Datenquelle getrennt wird.

## 1.1.3 Methoden

### TDBSession.Disconnect

```
procedure Disconnect;
```

Prozedur

Trennt die Verbindung zwischen Anwendung und Datenquelle. Löst *TDBSession.OnDisconnect* (siehe 1.1.2) aus.

### TDBSession.Connect

```
function Connect: Boolean;
```

Funktion

Liefert einen Wert des Typ *Boolean*. Stellt Verbindung zur Datenquelle her. Bei erfolgreicher Verbindung gibt *TDBSession.Connect* den Wert *True* zurück und löst *TDBSession.OnConnect* (siehe 1.1.2) aus. Ansonsten gibt *TDBSession.Connect* den Wert *False* zurück.

## TDBSession.BuildConnectionString

```
function BuildConnectionString: Boolean;
```

### Funktion

Liefert einen Wert des Typ *Boolean*. Zeigt Formular zur Erstellung eines Connect-String (siehe 1.1.1). Bei Abbruch durch den Benutzer gibt *TDBSession.BuildConnectionString* den Wert *False* zurück, ansonsten gibt *TDBSession.BuildConnectionString* den Wert *True* zurück und setzt das Attribut *TDBSession.ConnectString* (siehe 1.1.1) entsprechend der Angaben des Benutzers.

## 1.2 TDBSessions

Die Klasse *TDBSessions* ist ein Array von *TDBSession* (siehe 1.1)

## 1.3 TReloc

TReloc = class(TObject)	
ATTRIBUTE:	
+ FieldCount:	Integer
+ RelocName:	String
+ RelocOperationType:	TRelocOperationType
+ SourceDBSession:	TDBSession
+ SourceFieldNames:	Array of String
+ SourceSQL:	String;
+ TargetDBSession:	TDBSession
+ TargetFieldNames:	Array of String
+ TargetTableName:	String
EREIGNISSE:	
+ OnInsertError:	TRelocOnInsertError
+ OnLogMessage:	TRelocLogNotifyEvent
# OnSourceDBSessionNameChange:	TRelocOnDBSessionNameChange
# OnTargetDBSessionNameChange:	TRelocOnDBSessionNameChange
METHODEN:	
+ procedure PerformReloc	
+ function TestReloc:	Boolean

Die Klasse *TReloc* stammt von der Klasse *TObject* und enthält alle erforderlichen Informationen und Funktionen zum Import/Export von Daten zwischen Datenquellen. Import/Export Operationen (Reloc Operationen) werden von *TReloc* geprüft und durchgeführt.

Jede Reloc Operation wird vor Ausführung geprüft und bei eventuellen Fehlern abgebrochen. Sämtliche Operationen werden automatisch von *TReloc* protokolliert.

### 1.3.1 Attribute

#### TReloc.RelocOperationType

```
property RelocOperationType: TRelocOperationType read vRelocOperationType  
write vRelocOperationType;
```

Öffentliches Attribut vom Typ *TRelocOperationType* (siehe 2.3). Enthält die Art der Reloc Operation die durchzuführen ist.

#### TReloc.RelocName

```
property RelocName: String read vRelocName write vRelocName;
```

Öffentliches Attribut vom Typ *String*. Enthält einen eindeutigen Reloc Name. Wird von den Klassen *TRepFile* (siehe 1.5) und *TReloc* (siehe 1.3) benutzt um Reloc Operationen explizit zu erkennen und darauf zuzugreifen.

#### TReloc.FieldCount

```
property FieldCount: Integer read vFieldCount write SetFieldCount;
```

Öffentliches Attribut vom Typ *Integer*. Nur „Lese-Attribut“, der die Anzahl der zugeordneten Spalten zurück gibt.

#### TReloc.SourceDBSessionName

```
property SourceDBSessionName: String read vSourceDBSessionName  
write SetSourceDBSessionName;
```

Öffentliches Attribut vom Typ *String*. Name einer Datenbanksitzung [*TDBSession* (siehe 1.1)]. Dient als Verknüpfung zur Quelle.

#### TReloc.SourceDBSession

```
property SourceDBSession: TDBSession read vSourceDBSession;
```

Öffentliches Attribut vom Typ *TDBSession* (siehe 1.1). Nur „Lese-Attribut“ der die Verbindung zur Quelle herstellt. Dieses Attribut wird von *TRepFile* (siehe 1.5) erstellt gemäß *TReloc.SourceDBSessionName* (siehe 1.3.1).

### **TReloc.SourceFieldNames**

```
property SourceFieldNames: TArray<String> read vSourceFieldNames  
write vSourceFieldNames;
```

Öffentliches Attribut vom Typ *Array of String*. Enthält die Namen der Spalten der Quelle. Index *n* von *TReloc.SourceFieldNames* ist verknüpft mit Index *n* von *TReloc.TargetDBSessionName* (siehe 1.3.1).

### **TReloc.SourceSQL**

```
property SourceSQL: String read vSourceSQL write vSourceSQL;
```

Öffentliches Attribut vom Typ *String*. Enthält die SQL Abfrage, die die Quelldaten selektiert.

### **TReloc.TargetDBSessionName**

```
property TargetDBSessionName: String read vTargetDBSessionName  
write SetTargetDBSessionName;
```

Öffentliches Attribut vom Typ *String*. Name einer Datenbanksitzung [*TDBSession* (siehe 1.1)]. Dient als Verknüpfung zum Ziel.

### **TReloc.TargetDBSession**

```
property TargetDBSession: TDBSession read vTargetDBSession;
```

Öffentliches Attribut vom Typ *TDBSession* (siehe 1.1). Nur „Lese-Attribut“ der die Verbindung zum Ziel herstellt. Dieses Attribut wird von *TRepFile* (siehe 1.5) erstellt gemäß *TReloc.TargetDBSessionName* (siehe 1.3.1).

### **TReloc.TargetFieldNames**

```
property TargetFieldNames: TArray<String> read vTargetFieldNames  
write vTargetFieldNames;
```

Öffentliches Attribut vom Typ *Array of String*. Enthält die Namen der Spalten der Zieltabelle. Index *n* von *TReloc.TargetFieldNames* ist verknüpft mit Index *n* von *TReloc.SourceDBSessionName* (siehe 1.3.1).

## TReloc.TargetTableName

```
property TargetTableName: String read vTargetTableName write vTargetTableName;
```

Öffentliches Attribut vom Typ *String*. Enthält der Name der Zieltabelle.

## 1.3.2 Ereignisse

### TReloc.OnLogMessage

```
property OnLogMessage: TRelocLogNotifyEvent read vOnLogMessage write vOnLogMessage;
```

Ereignis von Typ *TRelocLogNotifyEvent* (siehe 3.2). Wird ausgelöst sobald eine Protokollnachricht von *TReloc* (siehe 1.3) ausgegeben wird.

### TReloc.OnSourceDBSessionNameChange

```
property OnSourceDBSessionNameChange: TRelocOnDBSessionNameChange  
read vOnSourceDBSessionNameChange  
write vOnSourceDBSessionNameChange;
```

Ereignis von Typ *TReloc.OnDBSessionNameChange* (siehe 3.3). Wird ausgelöst, wenn *TReloc.SourceDBSessionName* (siehe 1.3.1) geändert wurde. Wird von *TRepFile* (siehe 1.5) benutzt um *TReloc.SourceDBSession* (siehe 1.3.1) gemäß *TReloc.SourceDBSessionName* (siehe 1.3.1) zu erstellen.

### TReloc.OnTargetDBSessionNameChange

```
property OnTargetDBSessionNameChange: TRelocOnDBSessionNameChange  
read vOnTargetDBSessionNameChange  
write vOnTargetDBSessionNameChange;
```

Ereignis von Typ *TRelocOnDBSessionNameChange* (siehe 3.3). Wird ausgelöst, wenn *TReloc.TargetDBSessionName* (siehe 1.3.1) geändert wurde. Wird von *TRepFile* (siehe 1.5) benutzt um *TReloc.TargetDBSession* (siehe 1.3.1) gemäß *TReloc.TargetDBSessionName* (siehe 1.3.1) zu erstellen.

### TReloc.OnInsertError

```
property OnInsertError: TRelocOnInsertError read vOnInsertError  
write vOnInsertError;
```

Ereignis von Typ *TRelocOnInsertError* (siehe 3.4). Wird ausgelöst, sobald ein Fehler beim Einfügen eines Datensatzes in die Zieltabelle auftritt.

### 1.3.3 Methoden

#### TReloc.TestReloc

```
function TestReloc: Boolean;
```

##### Funktion

Liefert einen Wert des Typ *Boolean*. Prüft die zu ausführende Reloc Operation auf erfolgreiche Verbindung zur Quelle und Ziel, Verfügbarkeit der Spalten und Datentypkompatibilität. Erstellt ein Testergebnis-Protokoll.

Bei Fehler gibt *TReloc.TestReloc* den Wert *False* zurück, ansonsten gibt *TReloc.TestReloc* den Wert *True* zurück. Bei Warnungen wird ebenfalls *True* zurückgegeben.

#### TReloc.PerformReloc

```
procedure PerformReloc;
```

##### Prozedur

Nach erfolgreicher Prüfung der Reloc Operationen durch *TReloc.TestReloc*, führt *TReloc.PerformReloc* sämtliche Operationen aus. Ein Protokoll der ausgeführten Operationen wird erstellt. Wenn Fehler beim Import entstehen wird *TReloc.OnInsertError* (siehe 1.3.2) ausgelöst, dabei werden die nicht importierte Datensätze protokolliert und durch *TReloc.OnInsertError* (siehe 1.3.2) als Parameter übergeben.

### 1.4 TRelocs

Die Klasse *TRelocs* ist ein Array von *TReloc* (siehe 1.2)

## 1.5 TRepFile

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

Die Klasse *TRepFile* dient als Arbeitsumgebung und Verwaltungsassistent für mehrere Datenbanksitzungen und Reloc Operationen. *TRepFile* kann sämtliche Daten in einer Datei für späteren Zugriff speichern.

*TRepFile* ist verantwortlich, dass alle Quell- und Zieldatenbanksitzungen der *TReloc* (siehe 1.2) Objekte mit die gespeicherte Datenbanksitzungen durch ihre Sitzungsnamen [*TDBSession.SessionName* (siehe 1.1.1)] synchronisiert werden.

### 1.5.1 Attribute

#### TRepFile.RepositoryName

```
property RepositoryName: String read vRepositoryName write SetRepositoryName;
```

Öffentliches Attribut vom Typ *String*. Enthält der Name der Arbeitsumgebung, so ist es möglich mehrere Arbeitumgebungen zu verwalten und eindeutig zu definieren.

### TRepFile.FileName

```
property FileName: TFileName read vFileName;
```

Öffentliches Attribut vom Typ *TFileName*. Nur „Lese-Attribut“, der den aktuell verwendeten Dateinamen enthält.

### TRepFile.DBSessions

```
property DBSessions: TDBSessions read vDBSessions;
```

Öffentliches Attribut vom Typ *TDBSessions* (siehe 1.2). Enthält alle verfügbaren Datenbanksitzungen. Wird eine Datenbanksitzung geändert, so wirken sich die Änderungen auf alle Datenbanksitzungen der Reloc Operationen die den selben Sitzungsnamen besitzen, d.h. die Quelldatenbanksitzung und die Zieldatenbanksitzung von jede Reloc Operation ist durch den Sitzungsnamen der Datenbanksitzung verknüpft.

### TRepFile.Relocs

```
property Relocs: TRelocs read vRelocs;
```

Öffentliches Attribut vom Typ *TRelocs* (siehe 1.4). Enthält alle verfügbare Reloc Operationen.

### TRepFile.SessionCount

```
property SessionCount: Integer read GetSessionCount;
```

Öffentliches Attribut vom Typ *Integer*. Nur „Lese-Attribut“, der die Anzahl der Datenbanksitzungen enthält.

### TRepFile.RelocCount

```
property RelocCount: Integer read GetRelocCount;
```

Öffentliches Attribut vom Typ *Integer*. Nur „Lese-Attribut“, der die Anzahl der Reloc Operationen enthält.

## 1.5.2 Ereignisse

### TRepFile.OnSave

```
property OnSave: TNotifyEvent read vOnSave write vOnSave;
```

Ereignis von Typ *TNotifyEvent*. Wird ausgelöst nach erfolgreicher Speicherung des Arbeitsbereiches.

## 1.5.3 Methoden

### TRepFile.Save

```
procedure Save;
```

Prozedur

Speichert sämtliche Daten der Arbeitsumgebung in die Datei *TRepFile.FileName* (siehe 1.5.1). Löst *TRepFile.OnSave* (siehe 1.5.2) aus.

### TRepFile.GetDBSessionByName

```
function GetDBSessionByName  
(aDBSessionName: String; var aDBSession: TDBSession): Boolean;
```

Parameter:

Parameter Name	Datentyp
aDBSessionName	String
var aDBSession	TDBSession

Funktion

Liefert einen Wert des Typ *Boolean*. Sucht nach einer Datenbanksitzung in *TRepFile.DBSessions* (siehe 1.5.1) mit den Namen *aDBSessionName*. Existiert eine Datenbanksitzung mit diesen Namen, gibt *TRepFile.GetDBSessionByName* den Wert *True* zurück und liefert die entsprechende Datenbanksitzung durch *aDBSession* aus. Ansonsten liefert *TRepFile.GetDBSessionByName* den Wert *False* zurück, dabei bleibt *aDBSession* im Originalzustand. *aDBSession* darf nicht *null* sein.

### TRepFile.AddNullReloc

```
function AddNullReloc(aRelocName: String): Boolean;
```

#### Parameter:

Parameter Name	Datentyp
aRelocName	String

#### Funktion

Liefert Wert des Typ *Boolean*. Erstellt eine leere *Reloc* Operation mit den Namen *aRelocName*. Ist der Name *aRelocName* bereits in *TRepFile.Relocs* (siehe 1.5.1) vorhanden so wird keine *Reloc* Operation hinzugefügt und *TRepFile.AddNullReloc* gibt den Wert *False* zurück. Ansonsten gibt *TRepFile.AddNullReloc* den Wert *True* zurück.

### TRepFile.AddReloc

```
function AddReloc(aReloc: TReloc): Boolean;
```

#### Parameter:

Parameter Name	Datentyp
aReloc	TReloc

#### Funktion

Liefert einen Wert des Typ *Boolean*. Fügt *aReloc* zu *TRepFile.Relocs* (siehe 1.5.1) hinzu. Ist der Name von *aReloc* bereits in *TRepFile.Relocs* (siehe 1.5.1) vorhanden so wird keine *Reloc* Operation hinzugefügt und *TRepFile.AddReloc* gibt den Wert *False* zurück. Ansonsten gibt *TRepFile.AddReloc* den Wert *True* zurück.

### TRepFile.AddDBSession

```
function AddDBSession(aDBSession: TDBSession): Boolean;
```

#### Parameter:

Parameter Name	Datentyp
aDBSession	TDBSession (siehe 1.1)

#### Funktion

Liefert einen Wert des Typ *Boolean*. Fügt *aDBSession* zu *TRepFile.DBSessions* (siehe 1.5.1) hinzu. Ist der Name von *aDBSession* bereits in *TRepFile.DBSessions* (siehe 1.5.1) vorhanden so wird keine Datenbanksitzung hinzugefügt und *TRepFile.AddDBSession* gibt den Wert *False* zurück. Ansonsten gibt *TRepFile.AddDBSession* den Wert *True* zurück.

### TRepFile.DeleteReloc

```
function DeleteReloc(aRelocName: String): Boolean;
```

#### Parameter:

Parameter Name	Datentyp
aRelocName	String

#### Funktion

Liefert Wert des Typ *Boolean*. Entfernt eine Reloc Operation von *TRepFile.Relocs* (siehe 1.5.1) die den Namen *aRelocName* besitzt. Ist keine Reloc Operation mit den Namen *aRelocName* in *TRepFile.Relocs* (siehe 1.5.1) vorhanden so gibt *TRepFile.DeleteReloc* den Wert *False* zurück. Ansonsten gibt *TRepFile.DeleteReloc* den Wert *True* zurück.

### TRepFile.DeleteDBSession

```
function DeleteDBSession(aDBSessionName: String): Boolean;
```

#### Parameter:

Parameter Name	Datentyp
aDBSessionName	String

#### Funktion

Liefert Wert des Typ *Boolean*. Entfernt eine Datenbanksitzung von *TRepFile.DBSessions* (siehe 1.5.1) die den Namen *aDBSessionName* besitzt. Ist keine Datenbanksitzung mit den Namen *aDBSessionName* in *TRepFile.DBSessions* (siehe 1.5.1) vorhanden so gibt *DeleteDBSession* den Wert *False* zurück. Ansonsten gibt *TRepFile.DeleteDBSession* den Wert *True* zurück.

### TRepFile.Create

```
constructor Create(aFileName: TFileName);
```

### Parameter:

Parameter Name	Datentyp
aFileName	TFileName

### Constructor

Erstellt eine Instanz der Klasse *TRepFile*. Sämtliche Daten werden von der Datei *aFileName* in *TRepFile* geladen.

## 2 Structs, Records, Enums

### 2.1 TDBSessionType

```
type TDBSessionType = (dbstOracle, dbstODBC);
```

Wird von *TDBSession* (siehe 1.1) benutzt um anzugeben, ob die Daten von ein ORACLE Datenbanksystem kommen oder durch eine ODBC-Verbindung bzw. durch eine OLE DB Verknüpfung bezogen werden.

Möglich sind folgende Werte:

<i>dbstOracle</i>	Daten kommen von ein ORACLE Datenbanksystem
<i>dbstODBC</i>	Daten werden durch eine ODBC-Verbindung bzw. durch eine OLE DB Verknüpfung bezogen

### 2.2 TDataSourceMethod

```
type TDataSourceMethod = (dsmSQL, dsmTableDirect);
```

Wird von *TDBSession* (siehe 1.1) benutzt um anzugeben, ob die Daten durch eine SQL-Abfrage gelesen werden oder ob es eine direkte „Verbindung“ zu einer bestimmten Tabelle existiert.

Möglich sind folgende Werte:

<i>dsmSQL</i>	Daten werden durch eine SQL-Abfrage bezogen. Daten können aus mehrere Tabellen kommen, deswegen ist das Hinzufügen von Datensätze bei <i>dsmSQL</i> nicht möglich.
<i>dsmTableDirect</i>	Daten werden direkt von einer Tabelle bezogen und können bei <i>dsmTableDirect</i> eingefügt werden.

## 2.3 TRelocOperationType

```
type TRelocOperationType = (rotUpdate, rotInsert);
```

Art der Reloc Operation die durchzuführen ist.

Möglich sind folgende Werte:

<i>rotInsert</i>	Daten der Quelle werden direkt ins Ziel eingefügt.
<i>rotUpdate</i>	Daten der Quelle werden durch ein Schlüssel mit den Daten des Ziels verknüpft und aktualisiert (außer Schlüsselattribute). (Wird momentan nicht von <i>TReloc</i> (siehe 1.3) unterstützt)

## 2.4 TLogMessageType

```
type TLogMessageType = (lmError, lmMessage, lmWarning, lmComment);
```

Typ der Protokollnachricht ausgegeben von *TReloc* (siehe 1.3) durch *TReloc.OnLogMessage* (siehe 1.3.2).

Möglich sind folgende Werte:

<i>lmComment</i>	Kommentar
<i>lmMessage</i>	Meldung
<i>lmWarning</i>	Warnung
<i>lmError</i>	Fehler

### 3 Methodentypen

#### 3.1 TConvertFunction

```
type TConvertFunction = function(aSourceValue: Variant): Variant;
```

Funktion

Liefert einen Wert des Typ *Variant*.

Parameter:

Parameter Name	Datentyp
aSourceValue	Variant

#### 3.2 TRelocLogNotifyEvent

```
type TRelocLogNotifyEvent = procedure(Sender: TObject; AMessage: String;  
    AMessageType: TLogMessageType; ShowDateTime: Boolean) of object;
```

Prozedur

Ereignis aus Objekt.

Parameter:

Parameter Name	Datentyp
Sender	TObject
AMessage	String
AMessageType	TLogMessageType
ShowDateTime	Boolean

#### 3.3 TRelocOnDBSessionNameChange

```
type TRelocOnDBSessionNameChange = procedure(Sender: TObject;  
    var Succesfull: Boolean) of object;
```

Prozedur

Ereignis aus Objekt.

Parameter:

Parameter Name	Datentyp
Sender	TObject
Successful	Boolean

### 3.4 TRelocOnInsertError

```
type TRelocOnInsertError = procedure(Sender: TObject; DataRecord: TStringArray;  
                                     FldErr: Integer) of object;
```

Prozedur

Ereignis aus Objekt.

Parameter:

Parameter Name	Datentyp
Sender	TObject
DataRecord	Array of String
FldErr	Integer