



UltraLite® .NET-Programmierung

Version 16.0

Februar 2013

Version 16.0
Februar 2013

© 2013 SAP AG oder ein SAP-Konzernunternehmen. Alle Rechte vorbehalten.

Sie können diese Dokumentation (ganz oder teilweise) unter folgenden Bedingungen benutzen, reproduzieren und verteilen: 1) Sie müssen diese und alle anderen Urheberrechtsvermerke auf allen Kopien oder Auszügen der Dokumentation wiedergeben. 2) Sie dürfen die Dokumentation nicht verändern. 3) Sie dürfen nichts tun, aus dem abgeleitet werden könnte, dass Sie oder jemand anderer als SAP Verfasser oder Quelle der Dokumentation ist. Die hier enthaltenen Informationen können jederzeit ohne vorherigen Hinweis geändert werden.

Einige Softwareprodukte, die von der SAP AG oder einem ihrer Vertriebspartner vermarktet werden, enthalten Softwarekomponenten anderer Softwareanbieter. Die nationalen Produktspezifikationen können unterschiedlich sein.

Diese Dokumentationen werden von der SAP AG und ihren Tochtergesellschaften ("SAP Group") lediglich zu Informationszwecken bereitgestellt, ohne dass eine Gewährleistung oder eine Garantie irgendeiner Art gegeben wird. Die SAP Group übernimmt keine Verantwortung im Hinblick auf Fehler oder Auslassungen in den Dokumentationen. Die einzigen Garantien für Produkte und Dienstleistungen der SAP Group sind diejenigen, die in den mit den Produkten und Dienstleistungen eventuell gelieferten ausdrücklichen Garantieerklärungen enthalten sind. Keine der hier enthaltenen Informationen kann als Gewährung einer weitergehenden Garantie betrachtet werden.

SAP und weitere erwähnte SAP-Produkte und -Dienstleistungen sowie die entsprechenden Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und anderen Ländern. Weitere Hinweise finden Sie unter <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark>.

Inhalt

Über diese Dokumentation	vii
Erforderliche Systemausstattung und unterstützte Plattformen	1
UltraLite.NET-Anwendungsentwicklung	3
SQL Anywhere-Tools in Visual Studio	3
Verbindungseinrichtung für eine UltraLite-Datenbank	3
Datenerstellung und -änderung mit SQL-Anweisungen	5
Datenerstellung und -änderung mit der ULTable-Klasse	10
Transaktionsverwaltung	16
Zugriff auf Schemainformationen	17
Fehlerbehandlung	18
MobiLink-Datensynchronisation	18
UltraLite.NET-Anwendungen erstellen und das Deployment durchführen ..	20
 Praktische Einführung: Erstellen einer Windows Mobile- Anwendung unter Verwendung von UltraLite.NET	 25
Lektion 1: Erstellen eines Visual Studio-Projekts	26
Lektion 2: Erstellen einer UltraLite-Datenbank	29
Lektion 3: Datenbankverbindungs-Steurelemente zur Anwendung hinzufügen	30
Lektion 4: Einfügen, Aktualisieren und Löschen von Daten	33
Lektion 5: Erstellung und Deployment der Anwendung	37
Code der praktischen Einführung für C#	38
Programmcode der praktischen Einführung für Visual Basic	41
 UltraLite.NET-API-Referenz	 43
ULActiveSyncListener-Schnittstelle	44
ULBulkCopy-Klasse	46
ULBulkCopyColumnMapping-Klasse	57
ULBulkCopyColumnMappingCollection-Klasse	63

ULCommand-Klasse	72
ULCommandBuilder-Klasse	109
ULConnection-Klasse	121
ULConnectionParms-Klasse	169
ULConnectionStringBuilder-Klasse	179
ULCreateParms-Klasse	195
ULCursorSchema-Klasse	204
ULDataAdapter-Klasse	212
ULDatabaseManager-Klasse	222
ULDatabaseSchema-Klasse	229
ULDataReader-Klasse	237
ULException-Klasse	275
ULFactory-Klasse	277
ULFileTransfer-Klasse	282
ULFileTransferProgressData-Klasse	297
ULFileTransferProgressListener-Schnittstelle	300
ULIndexSchema-Klasse	301
ULInfoMessageEventArgs-Klasse	308
ULMetaDataCollectionNames-Klasse	310
ULParameter-Klasse	318
ULParameterCollection-Klasse	333
ULResultSet-Klasse	351
ULResultSetSchema-Klasse	375
ULRowsCopiedEventArgs-Klasse	376
ULRowUpdatedEventArgs-Klasse	378
ULRowUpdatingEventArgs-Klasse	381
ULServerSyncListener-Schnittstelle	384
ULSqlProgressData-Klasse	386
ULSyncParms-Klasse	387
ULSyncProgressData-Klasse	399
ULSyncProgressListener-Schnittstelle	409
ULSyncResult-Klasse	410
ULTable-Klasse	414
ULTableSchema-Klasse	437
ULTransaction-Klasse	450

ULInfoMessageEventHandler-Delegat	453
ULRowUpdatedEventHandler-Delegat	454
ULRowUpdatingEventHandler-Delegat	454
ULRowsCopiedEventHandler-Delegat	455
ULSyncProgressedDlg-Delegat	455
ULAuthStatusCode-Enumeration	456
ULBulkCopyOptions-Enumeration	457
ULDBValid-Enumeration	458
ULDateOrder-Enumeration	458
ULDbType-Enumeration	459
ULRuntimeType-Enumeration	463
ULSqlProgressState-Enumeration	464
ULStreamType-Enumeration	464
ULSyncProgressState-Enumeration	465
 Index	 469

Über diese Dokumentation

In diesem Handbuch wird die UltraLite.NET-Programmierschnittstelle beschrieben. Mit UltraLite.NET können Sie Datenbank Anwendungen für Windows-PCs, Handhelds, mobile und eingebettete Geräte entwickeln und das Deployment durchführen.

Erforderliche Systemausstattung und unterstützte Plattformen

Entwicklungsplattformen

Wenn Sie Anwendungen mit UltraLite.NET entwickeln möchten, benötigen Sie Folgendes:

- Eine unterstützte Desktopversion von Microsoft Windows.
- Microsoft Visual Studio 2005 oder Visual Studio 2008.
- Bei Windows Mobile-Geräten: .NET Compact Framework Version 2 oder später.

Zielpattformen

UltraLite.NET unterstützt die folgenden Zielpattformen:

- Microsoft .NET Compact Framework Version 2.0 oder später und .NET Framework 2.0 oder später unter Windows.
- Bei Windows Mobile-Geräten: Microsoft .NET Compact Framework Version 2 oder später.

Weitere Hinweise zu den von UltraLite unterstützten Plattformen finden Sie unter <http://www.sybase.com/detail?id=1002288>.

UltraLite.NET-Anwendungsentwicklung

Die UltraLite.NET API stellt den iAnywhere.Data.UltraLite-Namespace bereit. Dieser Namespace stellt UltraLite eine ADO.NET-Schnittstelle zur Verfügung. Sie hat den Vorzug, dass sie auf einem Industriestandardmodell aufbaut und einen Migrationspfad zu der sehr ähnlichen SQL Anywhere ADO.NET-Schnittstelle verfügbar macht.

Das .NET Compact Framework ist die Microsoft .NET-Laufzeitkomponente für Windows Mobile. Sie unterstützt mehrere Programmiersprachen. Sie können entweder Visual Basic.NET oder C# verwenden, um Anwendungen unter Verwendung von UltraLite.NET zu erstellen.

Hinweis

Sie können die UltraLite C/C++-API als Alternative zur UltraLite.NET API verwenden, um Anwendungen für Windows Mobile und Desktop-Computer zu erstellen. Weitere Hinweise finden Sie unter „[UltraLite C++-Anwendungsentwicklung](#)“ [[UltraLite - C- und C++-Programmierung](#)].

SQL Anywhere-Tools in Visual Studio

Die Visual Studio-Integration wird für UltraLite.NET unterstützt. Sie können auf die SQL Anywhere-Integrationstools über den Visual Studio Server-Explorer in Visual Studio 2005 oder später zugreifen.

Verbindungseinrichtung für eine UltraLite-Datenbank

UltraLite-Anwendungen müssen sich mit einer Datenbank verbinden, bevor sie Vorgänge mit den Daten in dieser Datenbank durchführen können. Dieser Abschnitt beschreibt, wie Sie sich mit einer UltraLite-Datenbank verbinden.

Hinweis

Die Codebeispiele in diesem Kapitel sind in Microsoft C# geschrieben. Wenn Sie eines der anderen unterstützten Entwicklungstools verwenden, müssen Sie die Anweisungen entsprechend ändern.

Verwendung des ULConnection-Objekts

Die meisten Anwendungen verwenden nur eine Verbindung zur UltraLite-Datenbank und halten diese Verbindung geöffnet. Mehrfache Verbindungen sind nur für den Datenzugriff über mehrere Threads erforderlich. Aus diesem Grund ist es häufig die beste Lösung, das ULConnection-Objekt global für die Anwendung zu deklarieren.

Die folgenden Eigenschaften des ULConnection-Objekts steuern das globale Anwendungsverhalten.

- **Commit-Verhalten** Standardmäßig befinden sich UltraLite.NET-Anwendungen im Autocommit-Modus. Jede Insert-, Update- bzw. Delete-Anweisung wird sofort in der Datenbank festgeschrieben. Sie können mit ULConnection.BeginTransaction den Start einer Transaktion in der Anwendung festlegen.

- **Benutzerauthentifizierung** Sie können die Benutzer-ID und das Kennwort für die Anwendung (Standardwerte DBA bzw. sql) ändern, indem Sie Methoden verwenden. Jede UltraLite-Datenbank kann maximal vier Benutzer-IDs festlegen.
- **Synchronisation** Eine Gruppe von Objekten, die die Synchronisation steuern und auf die vom Connection-Objekt aus zugegriffen wird.
- **Tabellen** Auf UltraLite-Tabellen wird mithilfe von Methoden des Connection-Objekts zugegriffen.
- **Befehle** Eine Gruppe von Objekten wird bereitgestellt, um die Ausführung von Dynamic SQL-Anweisungen und die Navigation in Ergebnismengen zu ermöglichen.

Anwendungen mit mehreren Threads

Jedes ULConnection-Objekt und alle von ihm erstellten Objekte sollten auf einem einzigen Thread verwendet werden. Wenn Ihre Anwendung mehrere Threads benötigt, die auf die UltraLite-Datenbank zugreifen, muss jeder Thread seine eigene Verbindung haben. Wenn Sie z.B. eine Anwendung erstellen, die die Synchronisation auf einem separaten Thread ausführt, müssen Sie eine separate Verbindung für die Synchronisation verwenden und die Verbindung auf diesem Thread öffnen.

Siehe auch

- „Transaktionsverwaltung“ auf Seite 16
- „MobiLink-Datensynchronisation“ auf Seite 18
- „Datenerstellung und -änderung mit der ULTable-Klasse“ auf Seite 10
- „Datenerstellung und -änderung mit SQL-Anweisungen“ auf Seite 5
- ULConnection-Klasse [UltraLite.NET] auf Seite 121

Verbindung mit einer UltraLite-Datenbank herstellen

Verwenden Sie das ULConnectionParms-Objekt, um eine Verbindung zu einer UltraLite-Datenbank namens *mydata.udb* herzustellen.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Deklarieren Sie ein ULConnection-Objekt.

```
ULConnection conn;
```

2. Öffnen Sie eine neue Verbindung zu einer vorhandenen Datenbank.

Sie können Verbindungsparameter als Verbindungszeichenfolge oder mithilfe des ULConnectionParms-Objekts angeben.

```
ULConnectionParms parms = new ULConnectionParms();  
parms.DatabaseOnDesktop = "mydata.udb";  
conn = new ULConnection( parms.ToString() );  
conn.Open();
```

Ergebnisse

Eine Verbindung zur *mydata.udb* Datenbank wird hergestellt.

Nächste Schritte

Verwenden Sie die Datenbankverbindung für SQL-Vorgänge zum Erstellen, Ändern oder Löschen von Daten. Sie können vorhandene Datenbankoptionen für eine geöffnete Verbindung ändern. Schließen Sie das `ULConnection`-Objekt, wenn Sie fertig sind.

Datenerstellung und -änderung mit SQL-Anweisungen

UltraLite-Anwendungen können auf Tabellendaten über SQL-Anweisungen oder die Tabellen-API zugreifen. In diesem Abschnitt wird der Datenzugriff mit SQL-Anweisungen beschrieben.

In diesem Abschnitt wird erläutert, wie die folgenden Aufgaben mit SQL erledigt werden:

- Zeilen einfügen, löschen und aktualisieren
- Abfragen ausführen und Zeilen in eine Ergebnismenge auslesen
- Durch die Zeilen einer Ergebnismenge blättern

In diesem Abschnitt wird die SQL-Sprache selbst nicht behandelt.

Siehe auch

- „Datenerstellung und -änderung mit der `ULTable`-Klasse“ auf Seite 10
- „SQL-Anweisungen“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]

Datenänderung mit INSERT, UPDATE und DELETE

In UltraLite können Sie Vorgänge mit der SQL-Datenmanipulationssprache vornehmen. Diese Vorgänge werden mithilfe der `ULCommand.ExecuteNonQuery`-Methode ausgeführt.

Siehe auch

- `ULCommand.ExecuteNonQuery`-Methode [*UltraLite.NET*] auf Seite 93

Eine Zeile in eine Tabelle einfügen

Platzhalter für Parameter in SQL-Anweisungen werden mit dem `?`-Zeichen (Fragezeichen) angegeben. Für INSERT, UPDATE oder DELETE wird jedes `?` nach seiner Stellung in der Parametersammlung des Befehls referenziert. Das erste `?` wird als 0 referenziert, das zweite als 1.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Deklarieren Sie ein ULCommand-Objekt.

```
ULCommand cmd;
```

2. Weisen Sie dem ULCommand-Objekt eine SQL-Anweisung zu.

```
cmd = conn.CreateCommand();  
cmd.CommandText = "INSERT INTO MyTable(MyColumn) values (?)";
```

3. Weisen Sie der Anweisung Eingabeparameterwerte zu.

Der nachstehende Code zeigt einen Zeichenfolgeparameter.

```
String newValue;  
  
cmd.Parameters.Clear();  
  
// assign value  
cmd.Parameters.Add("", newValue);
```

4. Führen Sie die Anweisung aus.

Der Rückgabewert zeigt die Anzahl von Zeilen, die von der Anweisung betroffen sind.

```
int rowsInserted = cmd.ExecuteNonQuery();
```

5. Wenn Sie explizite Transaktionen verwenden, schreiben Sie die Änderung fest.

```
myTransaction.Commit();
```

Ergebnisse

Eine neue Zeile wird zu MyTable hinzugefügt, wobei der MyColumn-Wert eine leere Zeichenfolge ist.

Eine Zeile in einer Tabelle aktualisieren

Platzhalter für Parameter in SQL-Anweisungen werden mit dem ?-Zeichen (Fragezeichen) angegeben. Für INSERT, UPDATE oder DELETE wird jedes ? nach seiner Stellung in der Parametersammlung des Befehls referenziert. Das erste ? wird als 0 referenziert, das zweite als 1.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Deklarieren Sie ein ULCommand-Objekt.

```
ULCommand cmd;
```

2. Weisen Sie dem ULCommand-Objekt eine Anweisung zu.

```
cmd = conn.CreateCommand();  
cmd.CommandText = "UPDATE MyTable SET MyColumn1 = ? WHERE MyColumn2 = ?";
```

3. Weisen Sie der Anweisung Eingabeparameterwerte zu.

```
String newValue;  
String oldValue;  
  
cmd.Parameters.Clear();  
  
// assign values  
cmd.Parameters.Add("", newValue);  
cmd.Parameters.Add("", oldValue);
```

4. Führen Sie die Anweisung aus.

```
int rowsUpdated = cmd.ExecuteNonQuery();
```

5. Wenn Sie explizite Transaktionen verwenden, schreiben Sie die Änderung fest.

```
myTransaction.Commit();
```

Ergebnisse

Zeileneinträge aus MyTable werden aktualisiert, wobei der MyColumn1-Wert eine leere Zeichenfolge ist. In diesem Szenario wird auch der MyColumn2-Wert auf eine leere Zeichenfolge gesetzt.

Eine Zeile in einer Tabelle löschen

Platzhalter für Parameter in SQL-Anweisungen werden mit dem ?-Zeichen (Fragezeichen) angegeben. Für INSERT, UPDATE oder DELETE wird jedes ? nach seiner Stellung in der Parametersammlung des Befehls referenziert. Das erste ? wird als 0 referenziert, das zweite als 1.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Deklarieren Sie ein ULCommand-Objekt.

```
ULCommand cmd;
```

2. Weisen Sie dem ULCommand-Objekt eine Anweisung zu.

```
cmd = conn.CreateCommand();  
cmd.CommandText = "DELETE FROM MyTable WHERE MyColumn = ?";
```

3. Weisen Sie der Anweisung Eingabeparameterwerte zu.

```
String deleteValue;
```

```
cmd.Parameters.Clear();

// assign value
cmd.Parameters.Add("", deleteValue);
```

4. Führen Sie die Anweisung aus.

```
int rowsDeleted = cmd.ExecuteNonQuery();
```

5. Wenn Sie explizite Transaktionen verwenden, schreiben Sie die Änderung fest.

```
myTransaction.Commit();
```

Ergebnisse

Zeileneinträge aus MyTable werden gelöscht, wobei der MyColumn-Wert in der Tabelle eine leere Zeichenfolge ist.

Daten mit SELECT abrufen

Führen Sie eine SELECT-Anweisung aus, um Informationen aus einer UltraLite-Datenbank abzurufen und die zurückgegebene Ergebnismenge zu bearbeiten.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Deklarieren Sie ein ULCommand-Objekt, das die Abfrage enthält.

```
ULCommand cmd;
```

2. Weisen Sie dem Objekt eine Anweisung zu.

```
cmd = conn.CreateCommand();
cmd.CommandText = "SELECT MyColumn FROM MyTable";
```

3. Führen Sie die Anweisung aus.

Abfrageergebnisse können als einer von mehreren Objekttypen zurückgegeben werden. In diesem Beispiel wird ein ULDataReader-Objekt verwendet.

```
ULDataReader customerNames = cmd.ExecuteReader();
int fc = customerNames.GetFieldCount();
while( customerNames.MoveNext() ) {
    for ( int i = 0; i < fc; i++ ) {
        System.Console.Write(customerNames.GetString( i ) + " " );
    }
    System.Console.WriteLine();
}
```

Ergebnisse

Das Ergebnis der SELECT-Anweisung enthält eine Zeichenfolge, die dann an die Eingabeaufforderung ausgegeben wird.

Beschreibung des Ergebnismengenschemas

Mit der Methode `ULDataReader.GetSchemaTable` und der Eigenschaft `ULDataReader.Schema` können Sie Informationen über eine Ergebnismenge abrufen, wie z.B. Spaltennamen, Gesamtanzahl der Spalten, Gesamtstellenzahl der Spalten, Dezimalstellen der Spalten, Spaltengröße sowie SQL-Datentypen der Spalten.

Beispiel

Das folgende Beispiel zeigt die Verwendung der Eigenschaften `ULDataReader.Schema` und `ResultSet.Schema` zur Anzeige von Schemainformationen in einer Eingabeaufforderung.

```
for ( int i = 0; i < MyResultSet.Schema.GetColumnCount(); i++ ) {  
    System.Console.WriteLine( MyResultSet.Schema.GetColumnName(i)  
                               + " "  
                               + MyResultSet.Schema.GetColumnSQLType(i) );  
}
```

Navigation in SQL-Ergebnismengen

Sie können durch eine Ergebnismenge navigieren, indem Sie die mit dem `ULDataReader`-Objekt verbundenen Methoden verwenden.

Das Ergebnismengen-Objekt bietet Ihnen die folgenden Methoden zur Navigation durch eine Ergebnismenge:

- **MoveAfterLast** Dies bewegt den Cursor an eine Stelle hinter der letzten Zeile.
- **MoveBeforeFirst** Dies bewegt den Cursor an eine Stelle vor der ersten Zeile.
- **MoveFirst** Dies bewegt den Cursor in die erste Zeile.
- **MoveLast** Dies bewegt den Cursor in die letzte Zeile.
- **MoveNext** Dies bewegt den Cursor in die nächste Zeile.
- **MovePrevious** Dies bewegt den Cursor in die vorherige Zeile.
- **MoveRelative(Offset)** Dies bewegt den Cursor eine bestimmte Anzahl von Zeilen in Bezug auf die aktive Zeile, wie im Offset angegeben. Positive Offsetwerte bewirken eine Vorwärtsbewegung in der Ergebnismenge relativ zur augenblicklichen Cursorposition in der Ergebnismenge und negative Offsetwerte bewirken eine Rückwärtsbewegung in der Ergebnismenge. Ein Offsetwert Null bewegt den Cursor nicht, ermöglicht Ihnen aber das Einlesen von Daten in den Zeilenpuffer.

Datenerstellung und -änderung mit der ULTable-Klasse

UltraLite-Anwendungen können über SQL-Anweisungen oder mithilfe der ULTable-Klasse auf Tabellendaten zugreifen. In diesem Abschnitt wird der Datenzugriff mit der ULTable-Klasse beschrieben.

In diesem Abschnitt wird erläutert, wie die folgenden Aufgaben mit der Tabellen-API erledigt werden:

- Durch die Zeilen einer Tabelle blättern
- Auf die Werte der aktuellen Zeile zugreifen
- Zeilen in einer Tabelle mit den Find- und Lookup-Methoden ausfindig machen
- Daten einfügen, aktualisieren und löschen

Siehe auch

- [„Datenerstellung und -änderung mit SQL-Anweisungen“ auf Seite 5](#)

Zeilennavigation

UltraLite.NET bietet mehrere Methoden zur Navigation durch eine Tabelle, um eine Vielzahl von Navigationsvorgängen durchzuführen.

Das table-Objekt bietet Ihnen die folgenden Methoden zur Navigation durch eine Tabelle.

- **MoveAfterLast** Dies bewegt den Cursor an eine Stelle hinter der letzten Zeile.
- **MoveBeforeFirst** Dies bewegt den Cursor an eine Stelle vor der ersten Zeile.
- **MoveFirst** Dies bewegt den Cursor in die erste Zeile.
- **MoveLast** Dies bewegt den Cursor in die letzte Zeile.
- **MoveNext** Dies bewegt den Cursor in die nächste Zeile.
- **MovePrevious** Dies bewegt den Cursor in die vorherige Zeile.
- **MoveRelative(Offset)** Dies bewegt den Cursor eine bestimmte Anzahl von Zeilen in Bezug auf die aktive Zeile, wie im Offset angegeben. Positive Offsetwerte bewirken eine Vorwärtsbewegung in der Tabelle relativ zur augenblicklichen Cursorposition in der Ergebnismenge und negative Offsetwerte bewirken eine Rückwärtsbewegung in der Tabelle. Ein Offsetwert Null bewegt den Cursor nicht, ermöglicht Ihnen aber das Einlesen von Daten in den Zeilenpuffer.

Siehe auch

- [ULTable-Klasse \[UltraLite.NET\] auf Seite 414](#)
- [ULTableSchema-Klasse \[UltraLite.NET\] auf Seite 437](#)

Beispiel

Der folgende Code öffnet die MyTable-Tabelle und zeigt den Wert für die MyColumn-Spalte bei jeder Zeile.

```
ULTable t = conn.ExecuteTable( "MyTable" );
int colID = t.GetOrdinal( "MyColumn" );
while ( t.MoveNext() ){
    System.Console.WriteLine( t.GetString( colID ) );
}
```

Die Zeilen der Tabelle werden der Anwendung vorgelegt, wenn Sie das Tabellenobjekt öffnen. Standardmäßig werden die Zeilen in der Reihenfolge des Primärschlüsselwertes sortiert, Sie können jedoch einen Index angeben und eine Tabelle öffnen, um in einer bestimmten Reihenfolge auf die Zeilen zuzugreifen.

Beispiel

Mit dem folgenden Code wird zur ersten Zeile der Tabelle MyTable gewechselt, wie es der Index ix_col vorsieht.

```
ULTable t = conn.ExecuteTable( "MyTable", "ix_col" );
t.MoveFirst();
```

UltraLite-Modi

Ein UltraLite-Modus legt den Zweck fest, für den die Werte im Puffer benutzt werden. Die vier nachstehenden Betriebsmodi stehen in UltraLite zusätzlich zum Standardmodus zur Verfügung.

- **Einfügungsmodus (insert)** Die Daten im Puffer werden der Tabelle als neue Zeile hinzugefügt, wenn die insert-Methode aufgerufen wird.
- **Aktualisierungsmodus (update)** Die Daten im Puffer ersetzen die aktuelle Zeile, wenn die update-Methode aufgerufen wird.
- **Suchmodus (find)** Er wird verwendet, um eine Zeile ausfindig zu machen, deren Wert genau zu den Daten im Puffer passt, wenn eine der find-Methoden aufgerufen wird.
- **Nachschlagemodus (lookup)** Er wird verwendet, um eine Zeile ausfindig zu machen, deren Wert genau zu den Daten im Puffer passt oder größer ist, wenn einer der Nachschlagemodi aufgerufen wird.

Einfügen von Zeilen

Die Schritte zum Einfügen einer Zeile sind den Schritten zum Aktualisieren von Zeilen sehr ähnlich, außer dass es nicht erforderlich ist, eine Zeile in der Tabelle ausfindig zu machen, bevor der Einfügevorgang vorgenommen werden kann. Die Reihenfolge der Einfügung von Zeilen in die Tabelle hat keine Bedeutung.

Beispiel

Der folgende Code fügt eine neue Zeile ein.

```
t.InsertBegin();  
t.SetInt( id, 3 );  
t.SetString( lname, "Carlo" );  
t.Insert();
```

Wenn Sie keinen Wert für eine der Spalten festlegen und diese Spalte einen Standardwert enthält, wird der Standardwert verwendet. Wenn die Spalte keinen Standardwert enthält, wird einer der folgenden Einträge benutzt:

- Bei nullwertfähigen Spalten: NULL
- Bei numerischen Spalten, die NULL nicht zulassen: Null (0).
- Bei Zeichenspalten, die NULL nicht zulassen: eine leere Zeichenfolge
- Um einen Wert explizit auf NULL zu setzen, verwenden Sie die SetDBNull-Methode.

Bei Aktualisierungsvorgängen wird eine Einfügung in der Datenbank permanent gespeichert, wenn eine Festschreibung durchgeführt wird. Im Autocommit-Modus wird die Commit-Anweisung als Teil der Insert-Methode ausgeführt.

Zeilenaktualisierungen

Vorsicht

Sie können den Primärschlüsselwert einer Zeile nicht aktualisieren. Löschen Sie stattdessen die Zeile und fügen Sie eine neue Zeile hinzu.

Standardmäßig operiert UltraLite.NET im Autocommit-Modus, in dem die Aktualisierung sofort an der Zeile in der permanenten Speicherung angewendet wird. Wenn Sie den Autocommit-Modus deaktiviert haben, wird das Update erst übernommen, sobald Sie den Commit-Vorgang ausführen.

Siehe auch

- „[Transaktionsverwaltung](#)“ auf Seite 16

Eine Zeile in einer Tabelle aktualisieren

Verwenden Sie die Update-Methode, um eine Zeile in einer Tabelle zu aktualisieren.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Wechseln Sie zu der zu aktualisierenden Zeile.

Sie können dafür entweder die Tabelle abrollen oder die find- oder lookup-Methode verwenden.

2. Aktivieren Sie den Update-Modus.

Die folgende Anweisung startet den Aktualisierungsmodus für eine Tabelle.

```
t.UpdateBegin();
```

3. Legen Sie die neuen Werte für die zu aktualisierende Zeile fest.

Die nachstehende Anweisung setzt die id-Spalte im Puffer auf 3.

```
t.SetInt( id , 3);
```

4. Führen Sie die Aktualisierung aus.

```
t.Update();
```

Ergebnisse

Die aktuelle Zeile wird aktualisiert. Wenn Sie den Wert einer Spalte in jenem Index ändern, der beim Öffnen des table-Objekts angegeben wurde, wirkt sich das auf die Positionierung aus.

Zeilendurchsuchungen

UltraLite hat beim Arbeiten mit Daten mehrere Betriebsmodi. Zwei dieser Modi, die Modi zum Suchen und Nachschlagen, werden zum Suchen verwendet. Das Table-Objekt verfügt über zwei Methoden, die diesen Modi zum Ausfindigmachen bestimmter Zeilen in einer Tabelle entsprechen:

Hinweis

Die Spalten, die mit den Find- und Lookup-Methoden gesucht werden, müssen sich in dem Index befinden, der benutzt wird, um die Tabelle zu öffnen.

- **Find-Methoden** Diese Methoden platzieren sich auf die erste Zeile, die den angegebenen Suchwerten genau entspricht, und zwar anhand der Sortierreihenfolge, die beim Öffnen des Table-Objekts angegeben wurde. Sollte die Suche nicht erfolgreich sein, wird die Anwendung vor der ersten oder nach der letzten Zeile positioniert.
- **Lookup-Methoden** Sie wechseln gemäß der Sortierfolge, die beim Öffnen des Table-Objekts angegeben wurde, zur ersten Zeile, die größer gleich dem angegebenen Suchwert ist.

Siehe auch

- [ULTable-Klasse \[UltraLite.NET\] auf Seite 414](#)

Suchen nach einer Zeile mit den Find- und Lookup-Methoden

Verwenden Sie die Find- und Lookup-Methoden, um in einem ULTable-Objekt nach einer Zeile zu suchen.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Aktivieren Sie den Find- oder Lookup-Modus.

Der Modus wird aktiviert, indem Sie eine Methode für das Table-Objekt aufrufen. Der folgende Code leitet beispielsweise den Find-Modus ein.

```
t.FindBegin();
```

2. Legen Sie Suchwerte fest.

Dies geschieht durch Festlegen von Werten in der aktuellen Zeile. Die Festlegung dieser Werte betrifft nur den Puffer, der die aktuelle Zeile enthält, nicht die Datenbank. Der folgende Code setzt den Wert im Puffer beispielsweise auf Kaminski.

```
int lname = t.GetOrdinal( "lname" );  
t.SetString( lname, "Kaminski" );
```

3. Suchen Sie nach der Zeile.

Verwenden Sie die geeignete Methode zum Ausführen der Suche. Mit der folgenden Anweisung wird z.B. nach der ersten Zeile gesucht, die genau mit dem angegebenen Wert im aktuellen Index übereinstimmt:

Für Indizes mit mehreren Spalten wird immer ein Wert für die erste Spalte benutzt, aber Sie können die anderen Spalten auslassen.

```
t.FindFirst();
```

4. Suchen Sie nach der nächsten Instanz der Zeile.

Verwenden Sie die geeignete Methode zum Ausführen der Suche. Für einen Suchvorgang ermittelt FindNext die nächste Instanz der Parameter im Index. Beim Nachschlagen ermittelt MoveNext die nächste Instanz.

Ergebnisse

Der Cursor zeigt auf die gewünschte Zeile.

Nächste Schritte

Führen Sie Vorgänge auf der Zeile durch, wie etwa Daten für die Zeile löschen oder ändern.

Zeilenabruf

Ein Tabellenobjekt befindet sich immer an einer der folgenden Positionen:

- Vor der ersten Zeile der Tabelle.
- In einer Zeile der Tabelle.
- Nach der letzten Zeile der Tabelle.

Wenn sich das Tabellenobjekt in einer Zeile befindet, können Sie je nach Datentyp eine von mehreren Methoden verwenden, um die Werte der einzelnen Spalten abzurufen oder zu ändern.

Spaltenwerte abrufen

Das Table-Objekt bietet eine Reihe von Methoden für den Abruf von Spaltenwerten. Diese Methoden nehmen die Spalten-ID als Argument.

Beispiele

Der folgende Code ruft den Wert der lname-Spalte ab, der eine Zeichenfolge ist.

```
int lname = t.GetOrdinal( "lname" );
string lastname = t.GetString( lname );
```

Der folgende Code ruft den Wert der cust_id-Spalte ab, der eine Ganzzahl ist.

```
int cust_id = t.GetOrdinal( "cust_id" );
int id = t.GetInt( cust_id );
```

Spaltenwerte ändern

Zusätzlich zu den Methoden zum Abrufen von Werten gibt es Methoden, die Werte festlegen. Diese Methoden nehmen die Spalten-ID und den Wert als Argument.

Beispiel

Der folgende Code definiert z.B. den Wert der lname-Spalte auf Kaminski.

```
t.SetString( lname, "Kaminski" );
```

Wenn diesen Eigenschaften Werte zugeordnet werden, ändert sich nicht der Wert der Daten in der Datenbank. Sie können den Eigenschaften sogar dann Werte zuordnen, wenn Sie sich vor der ersten Zeile oder hinter der letzten Zeile der Tabelle befinden, aber es ist ein Fehler, einen Datenzugriff zu versuchen, wenn sich die aktuelle Zeile an einer dieser Positionen befindet, indem z.B. die Eigenschaft einer Variablen zugeordnet wird.

```
// This code is incorrect
t.MoveBeforeFirst();
id = t.GetInt( cust_id );
```

Werte umwandeln

Die von Ihnen gewählte Methode muss dem Datentyp entsprechen, den Sie zuordnen möchten. UltraLite wandelt automatisch Datentypen der Datenbank um, wenn sie kompatibel sind, und daher können Sie die getString-Methode verwenden, um einen Ganzzahlwert in eine Zeichenfolge-Variable zu übergeben usw.

Siehe auch

- „CAST-Funktion [Datentypkonvertierung]“ [[UltraLite - Datenbankverwaltung](#)]
- „CONVERT-Funktion [Datentypkonvertierung]“ [[UltraLite - Datenbankverwaltung](#)]

Zeilenlöschungen

Die Schritte zum Löschen einer Zeile sind einfacher als zum Einfügen oder Aktualisieren von Zeilen. Es gibt keinen Löschmodus, der dem Einfügens- oder Aktualisierungsmodus entsprechen würde.

Sie löschen eine Zeile, indem Sie den Cursor auf die Zeile bewegen, die Sie löschen wollen, und dann die Table.Delete-Methode ausführen.

Beispiel

Der folgende Code veranschaulicht, wie die erste Zeile in der Tabelle gelöscht wird:

```
t.MoveFirst();  
t.Delete();
```

Transaktionsverwaltung

UltraLite ermöglicht die Transaktionsverarbeitung, damit die Integrität der Daten in Ihrer Datenbank gewährleistet ist. Eine Transaktion ist eine logische Arbeitseinheit. Entweder wird eine ganze Transaktion oder keine der Anweisungen in der Transaktion ausgeführt.

Standardmäßig arbeitet UltraLite.NET im Autocommit-Modus, sodass jede Einfügung, Aktualisierung oder Löschung als eigenständige Transaktion ausgeführt wird. Nachdem ein Vorgang abgeschlossen wurde, wird die Änderung in der Datenbank definitiv eingetragen.

Um Transaktionen mit Multi-Thread-Anweisungen zu verwenden, müssen Sie ein ULTransaction-Klassenobjekt erstellen, indem Sie ULConnection.BeginTransaction aufrufen. Wenn die Anwendung z.B. Geld zwischen zwei Konten überträgt, muss sowohl die Subtraktion vom Quellkonto als auch die Addition zum Zielkonto als eigenständiger Vorgang ausgeführt werden. Andernfalls darf keine der beiden Anweisungen abgeschlossen werden.

Wenn die Verbindung eine gültige Transaktion ausgeführt hat, müssen Sie die ULTransaction.Commit-Anweisung ausführen, um die Transaktion abzuschließen und die Änderungen in der Datenbank festzuschreiben. Wenn die Gruppe der Aktualisierungen abgebrochen werden soll, führen Sie die ULTransaction.Rollback-Anweisung aus, um den Vorgang abubrechen und alle Anweisungen der Transaktion zurückzusetzen. Wenn eine Transaktion festgeschrieben oder zurückgesetzt wurde, wechselt die Verbindung bis zum nächsten Aufruf von ULConnection.BeginTransaction zurück in den Autocommit-Modus.

Das folgende Codefragment zeigt z.B., wie Sie eine Transaktion einrichten, die mehrere Vorgänge umfasst (und das standardmäßige Autocommit-Verhalten vermeidet):

```
// Assuming an already open connection named conn  
ULTransaction txn = conn.BeginTransaction(IsolationLevel.ReadUncommitted);  
// Perform transaction operations here  
txn.Commit();
```

Hinweis

UltraLite unterstützt nur die ReadCommitted- und ReadUncommitted-Mitglieder der IsolationLevel-Enumeration.

Einige SQL-Anweisungen - v.a. Anweisungen, die die Datenbankstruktur verändern - bewirken, dass alle offenen Transaktionen festgeschrieben werden. Beispiele für SQL-Anweisungen, die laufende Transaktionen automatisch festschreiben, sind CREATE TABLE und ALTER TABLE.

Siehe auch

- [ULConnection-Klasse \[UltraLite.NET\] auf Seite 121](#)
- [ULTransaction-Klasse \[UltraLite.NET\] auf Seite 450](#)
- [ULConnection.BeginTransaction-Methode \[UltraLite.NET\] auf Seite 129](#)

Zugriff auf Schemainformationen

Die Objekte in der Tabellen-API repräsentieren Tabellen, Spalten, Indizes und Synchronisationspublikationen. Jedes Objekt hat eine Schema-Eigenschaft, die Zugriff auf Angaben zur Struktur dieses Objekts bietet.

Das Schema kann nicht über die API geändert werden. Sie können lediglich Angaben zum Schema abrufen.

Sie können auf die folgenden Schemaobjekte und Informationen zugreifen:

- **ULDatabaseSchema** Gibt die Anzahl und Namen der Tabellen in der Datenbank sowie die globalen Eigenschaften wie Datums- und Zeitformat an.

Rufen Sie ULConnection.Schema auf, um ein ULDatabaseSchema-Objekt zu erhalten.

- **ULTableSchema** Die Anzahl und Namen der Spalten und Indizes in dieser Tabelle

Rufen Sie ULTable.Schema auf, um ein ULTableSchema-Objekt zu erhalten.

- **ULIndexSchema** Informationen über die Spalten im Index. Da ein Index keine direkt mit ihm verbundenen Daten hat, gibt es keine eigene Index-Klasse, sondern nur eine ULIndexSchema-Klasse.

Rufen Sie die Methoden ULTableSchema.GetIndex, ULTableSchema.GetOptimalIndex oder ULTableSchema.GetPrimaryKey auf, um ein ULIndexSchema-Objekt zu erhalten. Siehe und .

Siehe auch

- [ULTable.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 436](#)
- [ULDatabaseSchema-Klasse \[UltraLite.NET\] auf Seite 229](#)
- [ULTableSchema-Klasse \[UltraLite.NET\] auf Seite 437](#)
- [ULIndexSchema-Klasse \[UltraLite.NET\] auf Seite 301](#)

Fehlerbehandlung

Zur Fehlerbehandlung können Sie die standardmäßigen .NET-Fehlerbehandlungsmethoden benutzen. Die meisten UltraLite-Methoden geben `ULException`-Fehler aus. Sie können `ULException.NativeError` verwenden, um den diesem Fehler zugewiesenen `ULSQLCode`-Wert zu erhalten. `ULException` hat eine `Message`-Eigenschaft, die Sie benutzen können, um einen beschreibenden Text des Fehlers zu erhalten. `ULSQLCode`-Fehler sind negative Zahlen, die den Fehlertyp angeben.

Nach der Synchronisation können Sie die `SyncResult`-Eigenschaft der Verbindung verwenden, um detailliertere Fehlerinformationen zu erhalten. Das folgende Beispiel zeigt ein mögliches Verfahren, um während der Synchronisation auftretende Fehler zu melden:

```
public void Sync() {
    try {
        _conn.Synchronize( this );
        _inSync = false;
    }
    catch( ULException uEx ) {
        if( uEx.NativeError ==
            ULSQLCode.SQLE_MOBILINK_COMMUNICATIONS_ERROR ) {
            MessageBox.Show(
                "StreamErrorCode = " +
                _conn.SyncResult.StreamErrorCode.ToString() + "\r\n"
                + "StreamErrorParameters = " +
                _conn.SyncResult.StreamErrorParameters + "\r\n"
                + "StreamErrorSystem = " +
                _conn.SyncResult.StreamErrorSystem + "\r\n"
            );
        }
        else {
            MessageBox.Show(uEx.Message);
        }
    }
    catch(System.Exception ex) {
        MessageBox.Show(ex.Message);
    }
}
```

Siehe auch

- [Fehlermeldungen](#)
- [ULSyncProgressListener-Schnittstelle \[UltraLite.NET\] auf Seite 409](#)
- [ULSyncResult-Klasse \[UltraLite.NET\] auf Seite 410](#)

MobiLink-Datensynchronisation

Eine UltraLite-Datenbank synchronisiert mit einer zentralen konsolidierten Datenbank. Die Synchronisation erfordert die MobiLink-Synchronisationssoftware, die in SQL Anywhere enthalten ist.

In diesem Abschnitt finden Sie eine kurze Einführung in die Synchronisation und die Beschreibung einiger Funktionen von besonderem Interesse für Benutzer von UltraLite.NET.

Sie finden überdies ein funktionierendes Synchronisationsbeispiel in der CustDB-Beispielanwendung. Weitere Hinweise finden Sie im Unterverzeichnis *Samples\UltraLite.NET\CustDB* Ihrer SQL Anywhere 16-Installation.

UltraLite.NET unterstützt die Synchronisation über TCP/IP, HTTP, HTTPS und TLS (Transportschichtssicherheit). Die Synchronisation wird durch die UltraLite-Anwendung eingeleitet. Verwenden Sie immer die Eigenschaften des SyncParms-Objektes zur Steuerung der Synchronisation.

Hinweis

Erforderliche getrennt lizenzierbare Komponenten.

FIPS-zertifizierte Verschlüsselung erfordert eine separate Lizenz. Alle Technologien für starke Verschlüsselungen unterliegen Exportbestimmungen.

Siehe „Getrennt lizenzierbare Komponenten“ [[SQL Anywhere 16 - Einführung](#)].

Siehe auch

- „UltraLite-Clients“ [[UltraLite - Datenbankverwaltung](#)]

Starten der Synchronisation in einer C#-Anwendung

Der folgende Code veranschaulicht, wie Sie die Synchronisation in einer in C# geschriebenen Anwendung initiieren.

```
private void Sync( ULConnection conn )
{
    // Sync
    try
    {
        // setup to synchronize a publication named "high_priority"
        conn.SyncParms.Publications = "high_priority";

        // Set the synchronization parameters
        conn.SyncParms.Version      = "Version1";
        conn.SyncParms.StreamParms = "";
        conn.SyncParms.Stream       = ULStreamType.TCPIP;
        conn.SyncParms.UserName     = "51";
        conn.Synchronize();
    }
    catch (System.Exception t)
    {
        MessageBox.Show("Exception: " + t.Message);
    }
}
```

ActiveSync-Synchronisation einrichten

In diesem Abschnitt wird beschrieben, wie Ihrer UltraLite.NET-Anwendung die ActiveSync-Synchronisation hinzugefügt und Ihre Anwendung für die Verwendung mit ActiveSync auf den Computern Ihrer Endbenutzer registriert wird.

Die ActiveSync-Synchronisation kann nur durch ActiveSync initiiert werden. ActiveSync initiiert eine Synchronisation, wenn das Gerät in die Dockingstation gestellt oder im ActiveSync-Fenster **Synchronisieren** gewählt wird.

Wenn ActiveSync die Synchronisation initiiert, startet der MobiLink-Provider für ActiveSync die UltraLite-Anwendung, falls sie noch nicht läuft, und sendet eine Meldung an sie. Ihre Anwendung muss ein `ULActiveSyncListener`-Objekt implementieren, um Meldungen vom MobiLink-Provider empfangen und verarbeiten zu können. Ihre Anwendung muss das Listener-Objekt unter Verwendung der Methode `SetActiveSyncListener` angeben, wobei **MyAppClassName** ein eindeutiger Windows-Klassenname für die Anwendung ist.

```
dbMgr.SetActiveSyncListener( "MyAppClassName", listener );
```

Wenn UltraLite eine ActiveSync-Meldung erhält, wird die `ActiveSyncInvoked`-Methode des angegebenen Listeners auf einem anderen Thread aufgerufen. Um Multi-Thread-Probleme zu vermeiden, sollte Ihre `ActiveSyncInvoked`-Methode ein Ereignis an die Benutzerschnittstelle absenden.

Wenn Ihre Anwendung eine Multi-Thread-Anwendung ist, verwenden Sie eine separate Verbindung sowie das **lock**-Schlüsselwort in C# oder das **SyncLock**-Schlüsselwort in Visual Basic .NET, um auf jene Objekte zuzugreifen, die vom Rest der Anwendung gemeinsam genutzt werden. Die `ActiveSyncInvoked`-Methode sollte ein `ULStreamType.ACTIVE_SYNC` für den `SyncParms.Stream`-Datenstrom ihrer Verbindung angeben und anschließend `ULConnection.Synchronize` aufrufen.

Wenn Sie Ihre Anwendung registrieren, legen Sie den folgenden Parameter fest:

- **Klassenname** Derselbe Klassenname, den die Anwendung mit der `Connection.SetActiveSyncListener`-Methode verwendet hat.

Siehe auch

- [ULActiveSyncListener-Schnittstelle \[UltraLite.NET\] auf Seite 44](#)

UltraLite.NET-Anwendungen erstellen und das Deployment durchführen

UltraLite.NET-Anwendungen können unter Windows Mobile und Windows verwendet werden. Wenn Sie ein Deployment unter Windows Mobile vornehmen, erfordert UltraLite.NET das .NET Compact Framework. Wenn Sie ein Deployment unter Windows vornehmen, ist .NET Framework erforderlich. UltraLite.NET unterstützt auch die ActiveSync-Synchronisation.

Siehe auch

- „Kompilierungs- und Deploymentspezifikationen für UltraLite-Anwendungen“ [[UltraLite - Datenbankverwaltung](#)]
- „Deployment des ActiveSync-Providers für UltraLite“ [[UltraLite - Datenbankverwaltung](#)]

Deployment einer UltraLite.NET-Anwendung für Windows Mobile

Geben Sie die entsprechenden Erstellungsparameter, Verbindungsparameter, Synchronisationsparameter, Protokolloptionen, Referenzen und Deployment-Dateien an, um sicherzustellen, dass Ihre UltraLite.NET-Anwendung auf Windows- und Windows Mobile-Geräten erfolgreich läuft.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Geben Sie die folgenden Parameter an:

- Wenn Sie Verschleierung verwenden, setzen Sie beim Erstellen der Datenbank den **obfuscate=1**-Erstellungsparameter.
- Wenn Sie AES- oder FIPS 140-2 AES-Verschlüsselung verwenden, setzen Sie beim Erstellen der Datenbank oder beim Herstellen einer Verbindung mit der Datenbank den Verbindungsparameter **DBKEY=Chiffrierschlüssel**.

2. Legen Sie die entsprechenden Parametereinstellungen fest, wenn Sie die Synchronisation in Ihrer UltraLite-Anwendung verwenden:

Synchronisationstyp	Parametereinstellungen
TCP/IP	Setzen Sie den Synchronisationsparameter Stream auf tcpip .
HTTP	Setzen Sie den Synchronisationsparameter Stream auf http .
RSA_TLS	Setzen Sie den Synchronisationsparameter Stream auf tls .
RSA HTTPS	Setzen Sie den Synchronisationsparameter Stream auf https .
RSA FIPS 140-2 TLS	Setzen Sie den Synchronisationsparameter Stream auf tls . Setzen Sie die Protokolloption fips=yes .
RSA FIPS 140-2 HTTPS	Setzen Sie den Synchronisationsparameter Stream auf https . Setzen Sie die Protokolloption fips=yes .

3. Wenn Sie RSA- oder RSA FIPS 140-2 RSA-Ende-zu-Ende-Verschlüsselung verwenden, setzen Sie die Protokolloption **e2ee_public_key=key-file**.

4. Wenn Sie ZLIB-Kompression verwenden, setzen Sie die Protokolloption **compression=zlib**.

5. Hinzufügen der Referenzen zu:

- **iAnywhere.Data.UltraLite**
- **iAnywhere.Data.UltraLite.resources**

6. Stellen Sie die folgenden Dateien bereit:

- %SQLANY16%\UltraLite\UltraLite.NET\Assembly\V2\iAnywhere.Data.UltraLite.dll.
- %SQLANY16%\UltraLite\UltraLite.NET\Assembly\V2\en\iAnywhere.Data.UltraLite.resources.dll.

- *ulnet16.dll* befindet sich in %SQLANY16%\UltraLite\UltraLite.NET\CE\Arm50 für Windows Mobile. Für Windows befindet es sich in %SQLANY16%\UltraLite\UltraLite.NET\x64 oder %SQLANY16%\UltraLite\UltraLite.NET\win32.

7. Deployment der für Ihre Anwendung erforderlichen Dateien:

- Wenn Sie ZLIB-Komprimierung verwenden, *mlczlib16.dll*.
- Wenn Sie RSA TLS, RSA HTTPS oder RSA E2EE verwenden, *mlcrsa16.dll*.
- Bei Verwendung von FIPS 140-2 AES-Verschlüsselung *ulfips16.dll* und *sbgse2.dll*.
- Wenn Sie RSA FIPS 140-2 TLS, RSA FIPS 140-2 HTTPS oder RSA FIPS 140-2 E2EE verwenden, *sbgse2.dll* und *mlcrsafips16.dll*.

Für Windows Mobile befinden sich die Dateien in %SQLANY16%\UltraLite\CE\Arm.50. Für Windows befinden sich die Dateien in %SQLANY16%\UltraLite\Windows\x64 oder %SQLANY16%\UltraLite\Windows\x86.

Ergebnisse

Die UltraLite.NET-Anwendung wird erfolgreich auf dem Windows Desktop oder dem Windows Mobile Device ausgeführt, auf dem sie bereitgestellt wurde.

Nächste Schritte

Führen Sie das Deployment einer UltraLite-Datenbank auf einem Windows-Desktop-Computer oder einem Windows Mobile-Gerät durch, auf dem die Anwendung bereitgestellt wurde, oder erstellen Sie eine neue Datenbank mit der per Deployment bereitgestellten Anwendung.

Siehe auch

- „Kompilierungs- und Deploymentspezifikationen für UltraLite-Anwendungen“ [[UltraLite - Datenbankverwaltung](#)]
- „Datenbank-Deploymenttechniken für UltraLite und UltraLite Java Edition“ [[UltraLite - Datenbankverwaltung](#)]

Deployment einer UltraLite.NET-Anwendung für Windows Mobile (UltraLite-Engine)

Geben Sie die entsprechenden Erstellungsparameter, Verbindungsparameter, Synchronisationsparameter, Protokolloptionen, Referenzen und Deployment-Dateien an, um sicherzustellen, dass Ihre UltraLite.NET-Anwendung auf Windows- und Windows Mobile-Geräten erfolgreich läuft.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Geben Sie die folgenden Parameter an:

- Wenn Sie Verschleierung verwenden, setzen Sie beim Erstellen der Datenbank den **obfuscate=1**-Erstellungsparameter.
- Wenn Sie AES- oder FIPS 140-2 AES-Verschlüsselung verwenden, setzen Sie beim Erstellen der Datenbank oder beim Herstellen einer Verbindung mit der Datenbank den Verbindungsparameter **DBKEY=encryption-key**.

2. Legen Sie die entsprechenden Parametereinstellungen fest, wenn Sie die Synchronisation in Ihrer UltraLite-Anwendung verwenden:

Synchronisationstyp	Parametereinstellungen
TCP/IP	Setzen Sie den Synchronisationsparameter Stream auf tcpip .
HTTP	Setzen Sie den Synchronisationsparameter Stream auf http .
RSA_TLS	Setzen Sie den Synchronisationsparameter Stream auf tls .
RSA HTTPS	Setzen Sie den Synchronisationsparameter Stream auf https .
RSA FIPS 140-2 TLS	Setzen Sie den Synchronisationsparameter Stream auf tls . Setzen Sie die Protokolloption fips=yes .
RSA FIPS 140-2 HTTPS	Setzen Sie den Synchronisationsparameter Stream auf https . Setzen Sie die Protokolloption fips=yes .

3. Wenn Sie RSA- oder FIPS 140-2 RSA-Ende-zu-Ende-Verschlüsselung verwenden, setzen Sie die Protokolloption **e2ee_public_key=key-file**.

4. Wenn Sie ZLIB-Kompression verwenden, setzen Sie die Protokolloption **compression=zlib**.

5. Hinzufügen der Referenzen zu:

- **iAnywhere.Data.UltraLite**
- **iAnywhere.Data.UltraLite.resources**

6. Stellen Sie die folgenden Dateien bereit:

- **%SQLANY16%\UltraLite\UltraLite.NET\Assembly\V2\iAnywhere.Data.UltraLite.dll**.
- **%SQLANY16%\UltraLite\UltraLite.NET\Assembly\V2\en\iAnywhere.Data.UltraLite.resources.dll**.
- **ulnetclient16.dll**, befindet sich in **%SQLANY16%\UltraLite\UltraLite.NET\CE\Arm50** für Windows Mobile. Für Windows befindet es sich in **%SQLANY16%\UltraLite\UltraLite.NET\x64** oder **%SQLANY16%\UltraLite\UltraLite.NET\win32**.

7. Deployment der für Ihre Anwendung erforderlichen Dateien:

- *uleng16.exe*.
- Wenn Sie ZLIB-Komprimierung verwenden, *mlczlib16.dll*.
- Wenn Sie RSA TLS, RSA HTTPS oder RSA E2EE verwenden, *mlcrsa16.dll*.
- Bei Verwendung von FIPS 140-2 AES-Verschlüsselung *ulfips16.dll* und *sbgse2.dll*.
- Wenn Sie RSA FIPS 140-2 TLS, RSA FIPS 140-2 HTTPS oder RSA FIPS 140-2 E2EE verwenden, *sbgse2.dll* und *mlcrsafips16.dll*.

Für Windows Mobile befinden sich die Dateien in %SQLANY16%\UltraLite\CE\Arm.50. Für Windows befinden sich die Dateien in %SQLANY16%\UltraLite\Windows\x64 oder %SQLANY16%\UltraLite\Windows\x86.

Ergebnisse

Die UltraLite.NET-Anwendung, die die UltraLite-Engine verwendet, wird erfolgreich auf dem Windows-Desktop-Computer oder dem Windows Mobile-Gerät ausgeführt, auf dem sie bereitgestellt wurde.

Nächste Schritte

Führen Sie das Deployment einer UltraLite-Datenbank auf einem Windows-Desktop-Computer oder einem Windows Mobile-Gerät durch, auf dem die Anwendung bereitgestellt wurde, oder erstellen Sie eine neue Datenbank mit der per Deployment bereitgestellten Anwendung.

Siehe auch

- „Kompilierungs- und Deploymentspezifikationen für UltraLite-Anwendungen“ [[UltraLite - Datenbankverwaltung](#)]
- „Datenbank-Deploymenttechniken für UltraLite und UltraLite Java Edition“ [[UltraLite - Datenbankverwaltung](#)]

Praktische Einführung: Erstellen einer Windows Mobile-Anwendung unter Verwendung von UltraLite.NET

Diese praktische Einführung beschreibt die Schritte, die Sie zum Erstellen einer UltraLite-Anwendung für Windows Mobile in Microsoft Visual Studio ausführen müssen. Sie verwendet die ADO.NET-Schnittstelle, die vom iAnywhere.Data.UltraLite-Namespace bereitgestellt wird und wird auf .NET Compact Framework 3.5 ausgeführt.

Diese praktische Einführung enthält Code für eine Visual Basic-Anwendung und für eine Visual C#-Anwendung.

Kenntnisse und Erfahrungen

In dieser praktischen Einführung wird von folgenden Voraussetzungen ausgegangen:

- Sie sind mit der Programmiersprache C# oder Visual Basic vertraut.
- Sie wissen, wie eine UltraLite-Datenbank mit dem UltraLite-Plugin in Sybase Central erstellt wird.
- Sie haben Microsoft Visual Studio auf Ihrem Computer installiert und sind mit Visual Studio vertraut. Diese praktische Einführung wurde unter Visual Studio 2005 getestet und bezieht sich möglicherweise auf Visual Studio-Aktionen oder Prozeduren, die sich ein wenig von anderen Versionen von Visual Studio unterscheiden.
- Sie haben Windows Mobile 5.0 SDK oder später von Microsoft installiert.
- Sie haben .NET Compact Framework 3.5 auf Ihrem mobilen Gerät installiert.

Ziele

Das Ziel dieser praktischen Einführung ist, dass Sie mit der Entwicklung von UltraLite-Anwendungen in der Visual Studio-Umgebung vertraut werden.

Installationshinweis

Wenn Sie UltraLite-Software auf einem Windows-Computer installieren, auf dem Visual Studio bereits installiert ist, erkennt der UltraLite-Installationprozess, dass Visual Studio vorhanden ist und führt die erforderlichen Integrationsschritte aus. Wenn Sie Visual Studio nach der Installation von UltraLite installieren oder eine neue Version von Visual Studio installieren, muss der Integrationsprozess von UltraLite und Visual Studio wie folgt manuell in der Eingabeaufforderung ausgeführt werden:

- Stellen Sie sicher, dass Visual Studio nicht läuft.
- Für Visual Studio 2005 oder später: Führen Sie *installULNet.exe* in dem Verzeichnis *%SQLANY16%\UltraLite\UltraLite.NET\Assembly\v2* aus. Für diese Aufgabe benötigen Sie möglicherweise Administratorprivilegien.

Siehe auch

- „Erstellen einer UltraLite-Datenbank mit dem Assistenten Datenbank erstellen“ [[UltraLite - Datenbankverwaltung](#)]

Lektion 1: Erstellen eines Visual Studio-Projekts

In dieser Lektion erstellen und konfigurieren Sie eine neue Visual Studio-Anwendung. Sie können als Programmiersprache entweder Visual Basic oder C# verwenden.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie die erforderliche Software installiert haben. Siehe „[Praktische Einführung: Erstellen einer Windows Mobile-Anwendung unter Verwendung von UltraLite.NET](#)“ auf Seite 25.

Kontext und Bemerkungen

Diese praktische Einführung geht davon aus, dass sich Ihre Dateien beim Entwurf einer C#-Anwendung im Verzeichnis `c:\tutorial\uldotnet\CSApp` und beim Entwurf einer Visual Basic-Anwendung im Verzeichnis `c:\tutorial\uldotnet\VBApp` befinden. Wenn Sie ein Verzeichnis mit einem anderen Namen verwenden möchten, müssen Sie dieses Verzeichnis während der gesamten praktischen Einführung verwenden.

Aufgabe

1. Erstellen Sie ein Visual Studio-Projekt.

- Klicken Sie im Visual Studio-Menü **Datei** auf **Neu » Projekt**.
- Es erscheint das Fenster **Neues Projekt**. Im linken Fensterausschnitt wählen Sie den Ordner der **Visual Basic**- oder der **Visual C#**-Projekte. Klicken Sie auf den Projekttyp **Smart Device**.
Klicken Sie im rechten Fensterausschnitt auf ein Smart Device-Projekt und nennen Sie Ihr Projekt entweder **VBApp** oder **CSApp**, je nachdem, ob Sie Visual Basic oder C# als Programmiersprache verwenden.
- Geben Sie den Speicherort `C:\tutorial\uldotnet` ein und klicken Sie auf **OK**.
- Klicken Sie auf die Zielplattform **Windows Mobile 5.0 Pocket PC SDK** und auf die .NET Compact Framework-Zielversion **.NET Compact Framework Version 3.5**. Klicken Sie auf **OK**.

2. Fügen Sie dem Projekt Verweise hinzu.

- Fügen Sie dem Projekt die iAnywhere.Data.UltraLite-Assembly und die zugeordneten Ressourcen hinzu.
 - a. Klicken Sie im Menü **Projekt** auf **Verweis hinzufügen**.
 - b. Klicken Sie in der Liste der verfügbaren Verweise auf **iAnywhere.Data.UltraLite** und **iAnywhere.Data.UltraLite EN** (für Englisch). Klicken Sie auf **OK**, um sie zur Liste ausgewählter Komponenten hinzuzufügen.

Wenn Sie nicht Englisch als Sprache wählen möchten, klicken Sie auf **Durchsuchen** und wählen die Datei **iAnywhere.Data.UltraLite xx** im Unterverzeichnis *UltraLite\UltraLite.NET\ce\Assembly\v2\xx* Ihrer SQL Anywhere-Installation aus, wobei xx eine aus zwei Buchstaben bestehende Abkürzung für die gewünschte Sprache ist (z.B. **en** für Englisch). Klicken Sie auf *iAnywhere.Data.UltraLite.resources.dll* und anschließend auf **Öffnen**.

- Verknüpfen Sie die UltraLite-Komponente mit dem Projekt.

Vergewissern Sie sich in diesem Schritt, dass Sie eine Verknüpfung mit dem Projekt erstellen und nicht die Komponente öffnen.

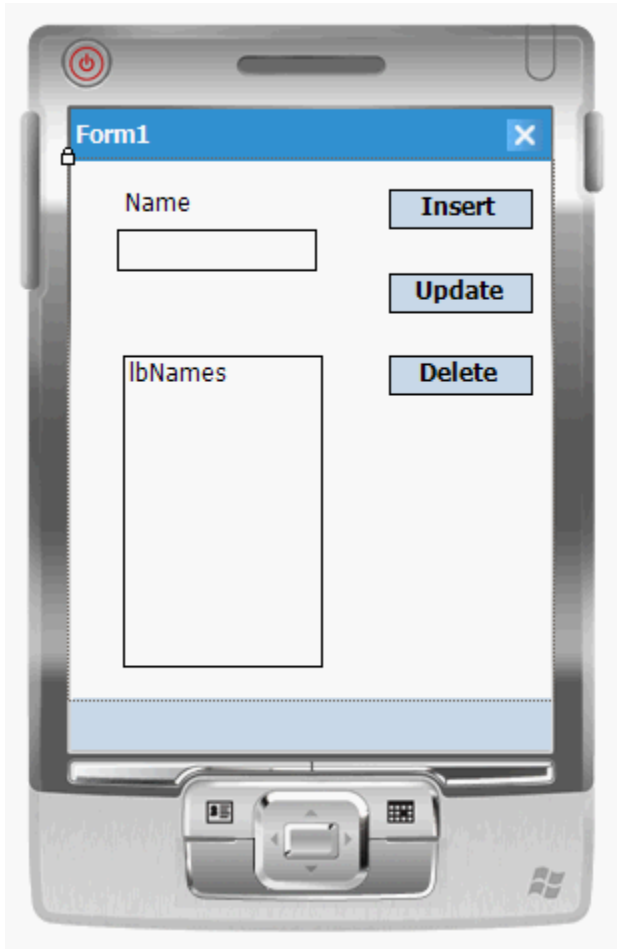
- Klicken Sie im Menü **Projekt** auf **Vorhandenes Element hinzufügen** und wechseln Sie in das Unterverzeichnis *UltraLite\UltraLite.NET\ce* Ihrer SQL Anywhere-Installation.
- Klicken Sie in der Liste **Dateityp** auf **Programmdateien**.
- Öffnen Sie den Ordner für den Prozessor des Windows Mobile-Geräts, das Sie verwenden. In Visual Studio 2005 und höher öffnen Sie den Ordner *arm.50*. Klicken Sie auf *ulnet16.dll*, danach auf den Pfeil in der Schaltfläche **Hinzufügen** und abschließend auf **Als Verknüpfung hinzufügen**.

3. Erstellen Sie ein Formular für Ihre Anwendung.

Wenn der Fensterausschnitt für die Visual Studio-Toolbox nicht angezeigt wird, klicken Sie im Hauptmenü auf **Ansicht » Toolbox**. Fügen Sie dem Formular die folgenden visuellen Komponenten hinzu, indem Sie das Objekt in der Toolbox markieren und an die gewünschte Position im Formular ziehen.

Typ	Name	Text
Button	btnInsert	Insert
Button	btnUpdate	Update
Button	btnDelete	Delete
TextBox	txtName	(kein Text)
ListBox	lbNames	(kein Text)
Label	laName	Name

Ihr Formular sollte nun der folgenden Abbildung ähneln:



4. Erstellen Sie Ihre Anwendung und führen Sie ein Deployment durch.

Durch das Kompilieren und das Deployment der Anwendung überprüfen Sie, ob Sie Ihr Visual Studio-Projekt richtig konfiguriert haben.

- a. Klicken Sie im Menü **Erstellen** auf **Projektmappe erstellen**. Überprüfen Sie, ob das Projekt erfolgreich erstellt wird. Wenn Sie eine Visual Basic-Anwendung erstellen, können Sie die folgende Warnung, die möglicherweise angezeigt wird, ignorieren:

Referenced assembly 'iAnywhere.Data.UltraLite.resources' is a localized satellite assembly

- b. Klicken Sie im Menü **Debug** auf **Debugging starten**.

Diese Aktion führt das Deployment Ihrer Anwendung für das mobile Gerät oder den Emulator durch und startet sie. Ein Deployment der Anwendung wird im Emulator oder im Gerät-Speicherort unter `\Programme\VBApp` oder `\Programme\CSApp` vorgenommen, abhängig vom jeweiligen Projektnamen.

Das Deployment kann einige Zeit dauern.

- c. Prüfen Sie, ob ein Deployment der Anwendung im Emulator oder auf dem Zielgerät vorgenommen und das von Ihnen entworfene Formular (**Form1**) richtig angezeigt wird.
- d. Fahren Sie den Emulator oder die Anwendung auf dem Zielgerät herunter.

Ergebnisse

Die UltraLite.NET API ist in der neuen Windows Mobile-Anwendung funktionsfähig.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 2: Erstellen einer UltraLite-Datenbank](#)“ auf Seite 29.

Lektion 2: Erstellen einer UltraLite-Datenbank

In dieser Lektion erstellen Sie eine UltraLite-Datenbank unter Verwendung von Sybase Central auf einem Desktopcomputer.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen eines Visual Studio-Projekts](#)“ auf Seite 26.

Aufgabe

1. Klicken Sie auf **Start » Programme » SQL Anywhere 16 » Administrationstools » Sybase Central**.
2. Erstellen Sie mit dem UltraLite-Plug-In für Sybase Central eine Datenbank in demselben Verzeichnis, in dem sich auch Ihre Anwendung befindet.

Klicken Sie im Menü **Extras** auf **UltraLite 16 » Datenbank erstellen**.

Im Allgemeinen reichen die Standard-Datenbankeigenschaften aus, die von Sybase Central eingerichtet werden. Beachten Sie folgende Eigenschaften:

- **Datenbankdateiname** `c:\tutorial\uldotnet\VBApp\VBApp.udb` oder `c:\tutorial\uldotnet\CSApp\CSApp.udb`, abhängig von Ihrem Anwendungstyp.
- **DBA-Benutzer-ID und Kennwort** Setzen Sie diese Werte für die Beispiele in dieser Dokumentation auf DBA und sql.
- **Kollationssequenz** Standardkollation verwenden.
- **Groß-/Kleinschreibung bei Zeichenfolgenvergleichen berücksichtigen** Diese Option sollte nicht aktiviert sein.

Klicken Sie auf **Fertig stellen** und stellen Sie eine Verbindung mit der UltraLite-Datenbank her.

3. Erstellen Sie eine neue UltraLite-Tabelle, indem Sie das Ordnersymbol **Tabellen** in der Strukturdarstellung von Sybase Central markieren. Klicken Sie anschließend auf **Datei » Neue Tabelle**. Beachten Sie folgende Eigenschaften:

- **Tabellenname** Geben Sie **Names** ein.
- **Spalten** Erstellen Sie in der Tabelle **Names** Spalten mit den folgenden Attributen:

Spaltenname	Datentyp (Größe)	Nullwerte	Eindeutig	Standardwert
ID	Integer	Nein	Ja (Primärschlüssel)	Global autoincrement
Name	Varchar(30)	Nein	Nein	Keine

- **Primärschlüssel** Legen Sie die ID-Spalte als Primärschlüssel fest.
4. Beenden Sie Sybase Central und überprüfen Sie, ob die Datenbankdatei im gewünschten Verzeichnis erstellt wurde.
 5. Verknüpfen Sie die initialisierte (leere) Datenbankdatei mit Ihrem Visual Basic Projekt, sodass ein Deployment der Datenbankdatei auf dem Gerät zusammen mit dem Anwendungscode erfolgt:
 - Klicken Sie im Menü **Visual Studio** auf **Projekt » Vorhandenes Element hinzufügen**.
 - Achten Sie darauf, dass **Objekttyp** auf **Alle Dateien** eingestellt ist. Wechseln Sie zu dem Verzeichnis, in dem Sie die Datenbankdatei erstellt haben, und klicken Sie auf die Datei *VBApp.udb* bzw. *CSApp.udb*, je nach Anwendungstyp.
 - Klicken Sie auf den Pfeil in der Schaltfläche **Hinzufügen** und auf **Als Verweis hinzufügen**.
 - Rechtsklicken Sie im Solution Explorer-Fenster auf den Namen der Datenbankdatei, die soeben zum Projekt hinzugefügt wurde, und klicken Sie auf **Eigenschaften**.
Setzen Sie die Eigenschaft **Buildvorgang** auf **Inhalt** und die Eigenschaft **In Ausgabeverzeichnis kopieren** auf **Immer kopieren**.

Ergebnisse

Eine UltraLite-Datenbank wird erstellt.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 3: Datenbankverbindungs-Steuerelemente zur Anwendung hinzufügen](#)“ auf Seite 30.

Siehe auch

- „Erstellen einer UltraLite-Datenbank mit dem Assistenten Datenbank erstellen“ [[UltraLite - Datenbankverwaltung](#)]

Lektion 3: Datenbankverbindungs-Steuerelemente zur Anwendung hinzufügen

In dieser Lektion fügen Sie Ihrer UltraLite.NET-Anwendung ein Steuerelement hinzu, das eine Verbindung zu einer UltraLite-Datenbank herstellt.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen eines Visual Studio-Projekts](#)“ auf Seite 26.

Aufgabe

1. Doppelklicken Sie auf das Formular, um die Quelldatei (*Form1.cs* oder *Form1.vb* zu öffnen).
2. Fügen Sie den Code zum Importieren des `iAnywhere.Data.UltraLite`-Namespaces hinzu.

Fügen Sie die folgende Anweisung in die erste Zeile der Datei ein.

```
//Visual C#  
using iAnywhere.Data.UltraLite;  
  
'Visual Basic  
Imports iAnywhere.Data.UltraLite
```

3. Fügen Sie der Formulardeklaration globale Variable hinzu.

Bei Visual C# fügen Sie nach dem Code, der die Formalkomponenten beschreibt, und vor der ersten Methodendeklaration den folgenden Code ein.

```
//Visual C#  
private ULConnection Conn;  
private int[] ids;
```

Bei Visual Basic fügen Sie den folgenden Code zu Beginn der `Form1`-Klasse ein.

```
'Visual Basic  
Dim Conn As ULConnection  
Dim ids() As Integer
```

Diese Variablen werden wie folgt benutzt:

- **ULConnection** Ein `Connection`-Objekt ist das Stammobjekt für alle Vorgänge, die für eine Verbindung mit einer Datenbank ausgeführt werden.
- **ids** Das `ids`-Array enthält die ID-Spaltenwerte, die nach Ausführung der Abfrage zurückgegeben werden.

Auch wenn Sie mit dem `ListBox`-Steuerelement auf sequenzielle Nummern zugreifen können, unterscheiden sich diese Nummern vom Wert der ID-Spalte, sobald eine Zeile gelöscht wurde. Daher müssen die ID-Spaltenwerte separat gespeichert werden.

4. Doppelklicken Sie auf einen leeren Bereich des Formulars, um die Methode `Form1_Load` zu erstellen.

Diese Methode führt die folgenden Aufgaben aus:

- Sie öffnet eine Verbindung zur Datenbank mithilfe der Verbindungsparameter, die im Steuerelement `ulConnectionParms1` festgelegt wurden.
- Sie ruft die Methode `RefreshListBox` (weiter unten in dieser praktischen Einführung definiert) auf.

- Sie gibt eine Fehlermeldung aus, wenn ein Fehler auftritt. Bei SQL Anywhere-Fehlern wird auch der Fehlercode ausgegeben. Siehe [Fehlermeldungen](#).

Bei C# fügen Sie der Methode Form1_Load folgenden Code hinzu:

```
//Visual C#
try {
    String ConnString = "dbf=\\Program Files\\CSApp\\CSApp.udb";
    Conn = new ULConnection( ConnString );
    Conn.Open();
    Conn.DatabaseID = 1;
    RefreshListBox();
}
catch ( System.Exception t ) {
    MessageBox.Show( "Exception: " + t.Message);
}
```

Bei Visual Basic fügen Sie der Methode Form1_Load folgenden Code hinzu.

```
'Visual Basic
Try
    Dim ConnString as String = "dbf=\\Program Files\\VBApp\\VBApp.udb"
    Conn = New ULConnection( ConnString )
    Conn.Open()
    Conn.DatabaseID = 1
    RefreshListBox()
Catch
    MsgBox("Exception: " + err.Description)
End Try
```

5. Erstellen Sie das Projekt.

Klicken Sie im Menü **Erstellen** auf **Projektmappe erstellen**. An dieser Stelle kann ein einzelner Fehler gemeldet werden, z.B. in C#: error CS0103: The name 'RefreshListBox' does not exist in the current context., da RefreshListBox noch nicht deklariert ist. Die nächste Lektion fügt diese Funktion hinzu.

Wenn Sie andere Fehlermeldungen erhalten, müssen Sie diese beheben, bevor Sie fortfahren. Suchen Sie nach üblichen Fehlern wie z.B. die widersprüchliche Verwendung der Groß- und Kleinschreibung in C#. Beispielsweise muss diese bei **UltraLite** und **ULConnection** genau übereinstimmen. In Visual Basic ist es absolut erforderlich, die **Imports iAnywhere.Data.UltraLite**-Anweisung wie in Lektion 3 beschrieben zu inkludieren.

Ergebnisse

Die Anwendung wird für eine Verbindung zu einer UltraLite-Datenbank eingerichtet.

Nächste Schritte

Gehen Sie weiter zu „[Lektion 4: Einfügen, Aktualisieren und Löschen von Daten](#)“ auf Seite 33.

Lektion 4: Einfügen, Aktualisieren und Löschen von Daten

In dieser Lektion fügen Sie Ihrer Anwendung Code hinzu, der Dynamic SQL zum Ändern der Daten in Ihrer Datenbank verwendet.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe „[Lektion 1: Erstellen eines Visual Studio-Projekts](#)“ auf Seite 26.

Kontext und Bemerkungen

In dieser Lektion erstellen Sie eine unterstützende Methode für die Verwaltung des Listenfelds. Diese Methode ist für die Datenmanipulationsmethode erforderlich, die in den später beschriebenen Verfahren verwendet wird.

Aufgabe

1. Rechtsklicken Sie auf das Formular und klicken Sie auf **Code anzeigen**.
2. Fügen Sie eine Methode der Form1-Klasse hinzu, mit der das Listenfeld aktualisiert und Daten in das Listenfeld geladen werden. Diese Methode führt die folgenden Aufgaben aus:
 - Sie löscht das Listenfeld.
 - Sie instanziert ein ULCommand-Objekt und weist es einer SELECT-Abfrage zu, die Daten aus der Tabelle "Namen" in der Datenbank ausgibt.
 - Sie führt eine Abfrage aus und gibt eine Ergebnismenge als ULDataReader zurück.
 - Sie instanziert ein Ganzzahl-Array, dessen Länge der Anzahl der Zeilen in der Ergebnismenge entspricht.
 - Sie lädt die in ULDataReader zurückgegebenen Namen in das Listenfeld und die in ULDataReader zurückgegebenen IDs in das Ganzzahl-Array.
 - Sie schließt ULDataReader.
 - Sie gibt eine Fehlermeldung aus, falls ein Fehler auftritt. Bei SQL-Fehlern wird auch der Fehlercode ausgegeben.

Bei einem C#-Projekt fügen Sie Ihrer Anwendung den folgenden Code als eine Methode der Form1-Klasse hinzu.

```
//Visual C#
private void RefreshListBox(){
    try{
        long NumRows;
        int I = 0;
        lbNames.Items.Clear();
        using( ULCommand cmd = Conn.CreateCommand() ){
            cmd.CommandText = "SELECT ID, Name FROM Names";
            using( ULDataReader dr = cmd.ExecuteReader()){
```

```
        dr.MoveBeforeFirst();
        NumRows = dr.RowCount;
        ids = new int[ NumRows ];
        while (dr.MoveNext())
        {
            lbNames.Items.Add(
                dr.GetString(1));
            ids[ I ] = dr.GetInt32(0);
            I++;
        }
        txtName.Text = " ";
    }
}
catch( Exception err ){
    MessageBox.Show(
        "Exception in RefreshListBox: " + err.Message );
}
}
```

Bei einem Visual Basic-Projekt fügen Sie Ihrer Anwendung den folgenden Code als eine Methode der Form1-Klasse hinzu.

```
'Visual Basic
Private Sub RefreshListBox()
    Try
        Dim cmd As ULCommand = Conn.CreateCommand()
        Dim I As Integer = 0
        lbNames.Items.Clear()
        cmd.CommandText = "SELECT ID, Name FROM Names"
        Dim dr As ULDataReader = cmd.ExecuteReader()
        ReDim ids(dr.RowCount)
        While (dr.MoveNext)
            lbNames.Items.Add(dr.GetString(1))
            ids(I) = dr.GetInt32(0)
            I = I + 1
        End While
        dr.Close()
        txtName.Text = " "
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub
```

3. Erstellen Sie das Projekt.

Bei der Erstellung des Projekts sollten keine Fehlermeldungen ausgegeben werden.

4. Auf der Registerkarte für den Entwurf des Formulars doppelklicken Sie auf die Schaltfläche **Insert**, um eine btnInsert_Click-Methode zu erstellen. Diese Methode führt die folgenden Aufgaben aus:

- Sie instanziert ein ULCommand-Objekt und weist es einer INSERT-Anweisung zu, die den Wert in das Textfeld in der Datenbank einfügt.
- Die Anweisung ausführen.
- Das ULCommand-Objekt entfernen.
- Das Listenfeld aktualisieren.
- Sie gibt eine Fehlermeldung aus, falls ein Fehler auftritt. Bei SQL-Fehlern wird auch der Fehlercode ausgegeben.

Bei C# fügen Sie der Methode btnInsert_Click folgenden Code hinzu.

```
//Visual C#
try {
    long RowsInserted;
    using( ULCommand cmd = Conn.CreateCommand() ) {
        cmd.CommandText =
            "INSERT INTO Names(name) VALUES (?)";
        cmd.Parameters.Add("", txtName.Text);
        RowsInserted = cmd.ExecuteNonQuery();
    }
    RefreshListBox();
}
catch( Exception err ) {
    MessageBox.Show("Exception: " + err.Message );
}
```

Bei Visual Basic fügen Sie der Methode btnInsert_Click folgenden Code hinzu.

```
'Visual Basic
Try
    Dim RowsInserted As Long
    Dim cmd As ULCommand = Conn.CreateCommand()
    cmd.CommandText = "INSERT INTO Names(name) VALUES (?)"
    cmd.Parameters.Add("", txtName.Text)
    RowsInserted = cmd.ExecuteNonQuery()
    cmd.Dispose()
    RefreshListBox()
Catch
    MsgBox("Exception: " + Err.Description)
End Try
```

5. Auf der Registerkarte für den Entwurf des Formulars doppelklicken Sie auf die Schaltfläche **Update**, um eine btnUpdate_Click-Methode zu erstellen. Diese Methode führt die folgenden Aufgaben aus:

- Sie instanziert ein ULCommand-Objekt und weist es einer UPDATE-Anweisung zu, die entsprechend der zugehörigen ID den Wert in das Textfeld in der Datenbank einfügt.
- Die Anweisung ausführen.
- Das ULCommand-Objekt entfernen.
- Das Listenfeld aktualisieren.
- Sie gibt eine Fehlermeldung aus, falls ein Fehler auftritt. Bei SQL-Fehlern wird auch der Fehlercode ausgegeben.

Bei C# fügen Sie der Methode btnUpdate_Click folgenden Code hinzu.

```
//Visual C#
try {
    long RowsUpdated;
    int updateID = ids[ lbNames.SelectedIndex ];
    using( ULCommand cmd = Conn.CreateCommand() ) {
        cmd.CommandText =
            "UPDATE Names SET name = ? WHERE id = ?" ;
        cmd.Parameters.Add("", txtName.Text );
        cmd.Parameters.Add("", updateID);
        RowsUpdated = cmd.ExecuteNonQuery();
    }
    RefreshListBox();
}
```

```
}  
catch( Exception err ) {  
    MessageBox.Show(  
        "Exception: " + err.Message);  
}
```

Bei Visual Basic fügen Sie der Methode btnUpdate_Click folgenden Code hinzu.

```
'Visual Basic  
Try  
    Dim RowsUpdated As Long  
    Dim updateID As Integer = ids(lbNames.SelectedIndex)  
    Dim cmd As ULCommand = Conn.CreateCommand()  
    cmd.CommandText = "UPDATE Names SET name = ? WHERE id = ?"  
    cmd.Parameters.Add("", txtName.Text)  
    cmd.Parameters.Add("", updateID)  
    RowsUpdated = cmd.ExecuteNonQuery()  
    cmd.Dispose()  
    RefreshListBox()  
Catch  
    MsgBox("Exception: " + Err.Description)  
End Try
```

6. Auf der Registerkarte für den Entwurf des Formulars doppelklicken Sie auf die Schaltfläche **Delete**, um eine btnDelete_Click-Methode zu erstellen. Fügen Sie Code hinzu, um die folgenden Aufgaben auszuführen:
- Ein ULCommand-Objekt instanzieren und ihm eine DELETE-Anweisung zuweisen. Mit der DELETE-Anweisung wird die ausgewählte Zeile basierend auf der verknüpften ID der Ganzzahl-Array IDs aus der Datenbank gelöscht.
 - Die Anweisung ausführen.
 - Das ULCommand-Objekt entfernen.
 - Das Listefeld aktualisieren.
 - Eine Fehlermeldung ausgeben, falls ein Fehler auftritt. Bei SQL-Fehlern wird auch der Fehlercode ausgegeben.

Bei C# fügen Sie der Methode btnDelete_Click folgenden Code hinzu.

```
//Visual C#  
try{  
    long RowsDeleted;  
    int deleteID = ids[lbNames.SelectedIndex];  
    using( ULCommand cmd = Conn.CreateCommand() ){  
        cmd.CommandText =  
            "DELETE From Names WHERE id = ?" ;  
        cmd.Parameters.Add("", deleteID);  
        RowsDeleted = cmd.ExecuteNonQuery ();  
    }  
    RefreshListBox();  
}  
catch( Exception err ) {  
    MessageBox.Show("Exception: " + err.Message );  
}
```

Bei Visual Basic fügen Sie der Methode btnDelete_Click folgenden Code hinzu.

```
'Visual Basic  
Try
```

```
Dim RowsDeleted As Long
Dim deleteID As Integer = ids(lbNames.SelectedIndex)
Dim cmd As ULCommand = Conn.CreateCommand()
cmd.CommandText = "DELETE From Names WHERE id = ?"
cmd.Parameters.Add("", deleteID)
RowsDeleted = cmd.ExecuteNonQuery()
cmd.Dispose()
RefreshListBox()
Catch
    MsgBox("Exception: " + Err.Description)
End Try
```

7. Erstellen Sie Ihre Anwendung, um zu überprüfen, dass sie korrekt kompiliert.

Ergebnisse

Die Windows Mobile-Anwendung wird so eingerichtet, dass Sie Datenvorgänge in der UltraLite-Datenbank durchführt.

Nächste Schritte

Gehen Sie weiter zu [„Lektion 5: Erstellung und Deployment der Anwendung“](#) auf Seite 37.

Siehe auch

- [„Datenerstellung und -änderung mit der ULTable-Klasse“](#) auf Seite 10
- [Fehlermeldungen](#)

Lektion 5: Erstellung und Deployment der Anwendung

In dieser Lektion entwickeln Sie Ihre Anwendung und führen das Deployment auf ein entferntes Gerät oder einen Emulator durch.

Voraussetzungen

In dieser Lektion wird davon ausgegangen, dass Sie bereits alle vorherigen Lektionen abgeschlossen haben. Siehe [„Lektion 1: Erstellen eines Visual Studio-Projekts“](#) auf Seite 26.

Aufgabe

1. Erstellen Sie die Lösung.

Stellen Sie sicher, dass die Anwendung ohne Fehler erstellt wird.

2. Wählen Sie das Deployment-Ziel aus.

Das Deployment-Ziel muss mit der Version von *ulnet16.dll* übereinstimmen, die Sie in Ihre Anwendung einbezogen haben.

3. Klicken Sie auf **Debug » Start**.

Es wird eine ausführbare Datei erstellt, die die Anwendung enthält und diese in den Emulator einsetzt. Der Prozess kann etwas Zeit in Anspruch nehmen, insbesondere dann, wenn .NET Compact Framework vor der Ausführung der Anwendung implementiert werden muss.

4. Wenn Fehler gemeldet werden, können Sie mit der folgenden Checkliste überprüfen, ob die Implementierung erfolgreich war.
 - Bestätigen Sie, dass die Anwendung in `\Programme\appname` eingerichtet wurde, wobei gilt, dass *appname* der Name ist, den Sie Ihrer Anwendung in Lektion 1 gegeben haben (CSApp oder VBApp).
 - Überprüfen Sie, ob der Pfad zur Datenbankdatei im Anwendungscode richtig ist.
 - Vergewissern Sie sich, dass Sie das Verknüpfen der Datei definiert haben, als die Datenbankdatei dem Projekt hinzugefügt wurde. Außerdem müssen die Eigenschaften "Buildvorgang" auf "Inhalt" und "In Ausgabeverzeichnis kopieren" auf "Immer kopieren" eingestellt werden. Andernfalls wird das Deployment der Dateien im Gerät nicht durchgeführt.
 - Achten Sie darauf, dass Sie einen Verweis zur richtigen Version von *ulnet16.dll* für Ihre Zielplattform angegeben oder den Windows Mobile-Installer aufgerufen haben. Wenn Sie bei früheren Versionen von Windows Mobile als Windows Mobile 5.0 zwischen dem Emulator und einem echten Gerät wechseln, müssen Sie die Version der verwendeten Bibliothek ändern.
 - Es kann sein, dass Sie den Emulator ohne Speicherung des Emulatorstatus beenden möchten. Bei einem erneuten Deployment der Anwendung werden alle erforderlichen Dateien in den Emulator kopiert, wodurch sichergestellt wird, dass es keine Versionsprobleme gibt.
5. Testen der Anwendung:
 - a. Fügen Sie Daten in die Datenbank ein.

Geben Sie einen Namen in das Textfeld ein, und klicken Sie auf **Insert (Einfügen)**. Der Name muss nun im Listefeld angezeigt werden.
 - b. Aktualisieren Sie die Daten in der Datenbank.

Klicken Sie im Listefeld auf einen Namen. Geben Sie einen neuen Namen in das Textfeld ein. Klicken Sie auf **Update (Aktualisieren)**. Anstelle des alten Namens muss nun der neue Name im Listefeld angezeigt werden.
 - c. Löschen Sie Daten aus der Datenbank.

Klicken Sie in der Liste auf einen Namen. Klicken Sie auf **Delete (Löschen)**. Der Name erscheint nicht mehr in der Liste.

Ergebnisse

Die Anwendung wird getestet und kann bereitgestellt werden.

Code der praktischen Einführung für C#

Im Folgenden ist der vollständige Code des Programms aus der praktischen Einführung, das in den vorangehenden Abschnitten beschrieben wurde, aufgelistet.

```

using iAnywhere.Data.UltraLite;
using System;
using System.Linq;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace CSApp
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private ULConnection Conn;
        private int[] ids;

        private void Form1_Load(object sender, EventArgs e)
        {
            try
            {
                String ConnString = "dbf=\\Program Files\\CSApp\\CSApp.udb";
                Conn = new ULConnection(ConnString);
                Conn.Open();
                Conn.DatabaseID = 1;
                RefreshListBox();
            }
            catch (System.Exception t)
            {
                MessageBox.Show("Exception: " + t.Message);
            }
        }
        private void RefreshListBox()
        {
            try
            {
                long NumRows;
                int I = 0;
                lbNames.Items.Clear();
                using (ULCommand cmd = Conn.CreateCommand())
                {
                    cmd.CommandText = "SELECT ID, Name FROM Names";
                    using (ULDataReader dr = cmd.ExecuteReader())
                    {
                        dr.MoveBeforeFirst();
                        NumRows = dr.RowCount;
                        ids = new int[NumRows];
                        while (dr.MoveNext())
                        {
                            lbNames.Items.Add(
                                dr.GetString(1));
                            ids[I] = dr.GetInt32(0);
                            I++;
                        }
                    }
                    txtName.Text = " ";
                }
            }
            catch (Exception err)
            {
            }
        }
    }
}

```

```
        MessageBox.Show(
            "Exception in RefreshListBox: " + err.Message);
    }
}

private void btnInsert_Click(object sender, EventArgs e)
{
    try
    {
        long RowsInserted;
        using (ULCommand cmd = Conn.CreateCommand())
        {
            cmd.CommandText =
                "INSERT INTO Names(name) VALUES (?)";
            cmd.Parameters.Add("", txtName.Text);
            RowsInserted = cmd.ExecuteNonQuery();
        }
        RefreshListBox();
    }
    catch (Exception err)
    {
        MessageBox.Show("Exception: " + err.Message);
    }
}

private void btnUpdate_Click(object sender, EventArgs e)
{
    try
    {
        long RowsUpdated;
        int updateID = ids[lbNames.SelectedIndex];
        using (ULCommand cmd = Conn.CreateCommand())
        {
            cmd.CommandText =
                "UPDATE Names SET name = ? WHERE id = ?";
            cmd.Parameters.Add("", txtName.Text);
            cmd.Parameters.Add("", updateID);
            RowsUpdated = cmd.ExecuteNonQuery();
        }
        RefreshListBox();
    }
    catch (Exception err)
    {
        MessageBox.Show(
            "Exception: " + err.Message);
    }
}

private void btnDelete_Click(object sender, EventArgs e)
{
    try
    {
        long RowsDeleted;
        int deleteID = ids[lbNames.SelectedIndex];
        using (ULCommand cmd = Conn.CreateCommand())
        {
            cmd.CommandText =
                "DELETE From Names WHERE id = ?";
            cmd.Parameters.Add("", deleteID);
            RowsDeleted = cmd.ExecuteNonQuery();
        }
        RefreshListBox();
    }
    catch (Exception err)
```

```

        {
            MessageBox.Show("Exception: " + err.Message);
        }
    }
}

```

Programmcode der praktischen Einführung für Visual Basic

```

Imports iAnywhere.Data.UltraLite
Public Class Form1
    Dim Conn As ULConnection
    Dim ids() As Integer
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles MyBase.Load
        Try
            Dim ConnString As String = "dbf=\Program Files\VBApp\VBApp.udb"
            Conn = New ULConnection(ConnString)
            Conn.Open()
            Conn.DatabaseID = 1
            RefreshListBox()
        Catch
            MsgBox("Exception: " + Err.Description)
        End Try
    End Sub
    Private Sub RefreshListBox()
        Try
            Dim cmd As ULCommand = Conn.CreateCommand()
            Dim I As Integer = 0
            lbNames.Items.Clear()
            cmd.CommandText = "SELECT ID, Name FROM Names"
            Dim dr As ULDataReader = cmd.ExecuteReader()
            ReDim ids(dr.RowCount)
            While (dr.MoveNext)
                lbNames.Items.Add(dr.GetString(1))
                ids(I) = dr.GetInt32(0)
                I = I + 1
            End While
            dr.Close()
            txtName.Text = " "
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
    End Sub

    Private Sub btnInsert_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles btnInsert.Click
        Try
            Dim RowsInserted As Long
            Dim cmd As ULCommand = Conn.CreateCommand()
            cmd.CommandText = "INSERT INTO Names(name) VALUES (?)"
            cmd.Parameters.Add("", txtName.Text)
            RowsInserted = cmd.ExecuteNonQuery()
            cmd.Dispose()
            RefreshListBox()
        Catch
            MsgBox("Exception: " + Err.Description)
        End Try
    End Sub

```

```
Private Sub btnUpdate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnUpdate.Click
    Try
        Dim RowsUpdated As Long
        Dim updateID As Integer = ids(lbNames.SelectedIndex)
        Dim cmd As ULCommand = Conn.CreateCommand()
        cmd.CommandText = "UPDATE Names SET name = ? WHERE id = ?"
        cmd.Parameters.Add("", txtName.Text)
        cmd.Parameters.Add("", updateID)
        RowsUpdated = cmd.ExecuteNonQuery()
        cmd.Dispose()
        RefreshListBox()
    Catch
        MsgBox("Exception: " + Err.Description)
    End Try
End Sub

Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDelete.Click
    Try
        Dim RowsDeleted As Long
        Dim deleteID As Integer = ids(lbNames.SelectedIndex)
        Dim cmd As ULCommand = Conn.CreateCommand()
        cmd.CommandText = "DELETE From Names WHERE id = ?"
        cmd.Parameters.Add("", deleteID)
        RowsDeleted = cmd.ExecuteNonQuery()
        cmd.Dispose()
        RefreshListBox()
    Catch
        MsgBox("Exception: " + Err.Description)
    End Try
End Sub
End Class
```

UltraLite.NET-API-Referenz

Dieses Kapitel beschreibt die API für den UltraLite.NET-Datenprovider für .NET Framework 2.0 und .NET Compact Framework 2.0.

UltraLite.NET-Erweiterungen, die im SQL Anywhere-Datenprovider für ADO.NET nicht verfügbar sind, werden in dieser API-Referenz mit **UL-Erw.:** bezeichnet.

Um die UltraLite-Laufzeit-Engine von UltraLite.NET zu verwenden, setzen Sie die RuntimeType-Eigenschaft auf den zutreffenden Wert, bevor Sie eine andere UltraLite.NET-API verwenden.

Die iAnywhere.Data.UltraLite-Assembly verwendet eine Ressourcen-Satelliten-Assembly namens iAnywhere.Data.UltraLite.resources. Die Haupt-Assembly sucht in der folgenden Reihenfolge anhand von Kulturen nach dieser Ressourcen-Assembly:

- CultureInfo.CurrentUICulture
- CultureInfo.CurrentCulture
- EN

Die folgende Liste beschreibt einige der gebräuchlichen Klassen der höheren Ebene für den iAnywhere.Data.UltraLite-Namespace:

- **ULConnection** Jedes ULConnection-Objekt stellt eine Verbindung zu einer UltraLite-Datenbank dar. Sie können ein oder mehrere ULConnection-Objekte erstellen.
- **ULTable** Jedes ULTable-Objekt bietet Zugriff auf die Daten in einer einzelnen Tabelle.
- **ULCommand** Jedes ULCommand-Objekt enthält eine SQL-Anweisung, die in der Datenbank ausgeführt werden soll.
- **ULDataReader** Jedes ULDataReader-Objekt enthält die Ergebnismenge einer einzelnen Abfrage.
- **ULSyncParms** Sie verwenden das ULSyncParms-Objekt, um Ihre UltraLite-Datenbank mit einem MobiLink-Server zu synchronisieren.

Viele in diesem Kapitel enthaltenen Eigenschaften und Methoden sind dem .NET Framework Datenprovider für OLE DB (System.Data.OleDb) sehr ähnlich. Weitere Informationen und Beispiele finden Sie in der Dokumentation zu Microsoft .NET Framework.

Namespace

iAnywhere.Data.UltraLite

Siehe auch

- <http://msdn.microsoft.com/en-us/library/System.Globalization.CultureInfo.CurrentCulture.aspx>
- <http://msdn.microsoft.com/en-us/library/System.Globalization.CultureInfo.CurrentCulture.aspx>
- [ULConnection-Klasse \[UltraLite.NET\] auf Seite 121](#)
- [ULTable-Klasse \[UltraLite.NET\] auf Seite 414](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULDataReader-Klasse \[UltraLite.NET\] auf Seite 237](#)
- [ULSyncParms-Klasse \[UltraLite.NET\] auf Seite 387](#)
- [ULDatabaseManager.RuntimeType-Eigenschaft \[UltraLite.NET\] auf Seite 229](#)

ULActiveSyncListener-Schnittstelle

UL-Erw.: Die Listener-Schnittstelle für den Empfang von ActiveSync-Ereignissen.

Visual Basic-Syntax

```
Public Interface ULActiveSyncListener
```

C#-Syntax

```
public interface ULActiveSyncListener
```

Mitglieder

Alle Mitglieder der ULActiveSyncListener-Schnittstelle, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ActiveSyncInvoked-Methode	Wird aufgerufen, wenn der MobiLink-Provider für ActiveSync die Anwendung aufruft, um die Synchronisation auszuführen

ActiveSyncInvoked-Methode

Wird aufgerufen, wenn der MobiLink-Provider für ActiveSync die Anwendung aufruft, um die Synchronisation auszuführen

Visual Basic-Syntax

```
Public Sub ActiveSyncInvoked(ByVal launchedByProvider As Boolean)
```

C#-Syntax

```
public void ActiveSyncInvoked(bool launchedByProvider)
```

Parameter

- **launchedByProvider** TRUE, wenn die Anwendung vom MobiLink-Provider gestartet wurde, um die ActiveSync-Synchronisation durchzuführen. Die Anwendung muss sich dann selbst beenden, nachdem sie die Synchronisation abgeschlossen hat. FALSE, wenn die Anwendung bereits lief, als sie vom MobiLink-Provider für ActiveSync aufgerufen wurde.

Bemerkungen

Diese Methode wird von einem eigenen Thread aufgerufen. Um Multi-Thread-Probleme zu vermeiden, sollte sie ein Ereignis an die Benutzerschnittstelle senden. Bei der Verwendung von Multi-Threading sollten Sie eine separate Verbindung sowie das Sperren-Schlüsselwort benutzen, um auf die Objekte zuzugreifen, die gemeinsam mit dem Rest der Anwendung verwendet werden.

Nach Abschluss der Synchronisation müssen Anwendungen die `ULDatabaseManager.SignalSyncIsComplete`-Methode aufrufen, um dem MobiLink-Provider für ActiveSync zu signalisieren.

Siehe auch

- [ULDatabaseManager.SignalSyncIsComplete-Methode \[UltraLite.NET\] auf Seite 228](#)

Beispiel

Die folgenden Codefragmente zeigen, wie eine ActiveSync-Anforderung empfangen und eine Synchronisation auf dem UI-Thread ausgeführt wird:

```
' Visual Basic
Imports iAnywhere.Data.UltraLite
Public Class MainWindow
    Inherits System.Windows.Forms.Form
    Implements ULActiveSyncListener

    Private conn As ULConnection

    Public Sub New(ByVal args() As String)
        MyBase.New()

        ' This call is required by the Windows Form Designer.
        InitializeComponent()

        ' Add any initialization after the InitializeComponent call.
        ULConnection.DatabaseManager.SetActiveSyncListener( _
            "myCompany.myapp", Me _
        )

        ' Create Connection
        ...
    End Sub

    Protected Overrides Sub OnClosing( _
        ByVal e As System.ComponentModel.CancelEventArgs _
    )
        ULConnection.DatabaseManager.SetActiveSyncListener( _
            Nothing, Nothing _
        )
        MyBase.OnClosing(e)
    End Sub

    Public Sub ActiveSyncInvoked( _
        ByVal launchedByProvider As Boolean _
    ) Implements ULActiveSyncListener.ActiveSyncInvoked
        Me.Invoke(New EventHandler(AddressOf Me.ActiveSyncAction))
    End Sub

    Public Sub ActiveSyncAction( _
        ByVal sender As Object, ByVal e As EventArgs _
    )
```

```
        ' Perform active sync.
        conn.Synchronize()
        ULConnection.DatabaseManager.SignalSyncIsComplete()
    End Sub
End Class
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
using iAnywhere.Data.UltraLite;

public class Form1 : System.Windows.Forms.Form, ULActiveSyncListener
{
    private System.Windows.Forms.MainMenu mainMenu1;
    private ULConnection conn;

    public Form1()
    {
        //
        // Required for Windows Form Designer support.
        // InitializeComponent();

        //
        // TODO: Add any constructor code after the
        // InitializeComponent call.
        //
        ULDatabaseManager.SetActiveSyncListener(
            "myCompany.myapp", this
        );

        // Create connection
        ...
    }

    protected override void Dispose( bool disposing )
    {
        base.Dispose( disposing );
    }

    protected override void OnClosing(
        System.ComponentModel.CancelEventArgs e )
    {
        ULDatabaseManager.SetActiveSyncListener(null, null);
        base.OnClosing(e);
    }

    public void ActiveSyncInvoked(bool launchedByProvider)
    {
        this.Invoke( new EventHandler( ActiveSyncHandler ) );
    }

    internal void ActiveSyncHandler(object sender, EventArgs e)
    {
        conn.Synchronize();
        ULDatabaseManager.SignalSyncIsComplete();
    }
}
```

ULBulkCopy-Klasse

Lädt eine UltraLite-Tabelle mit großen Datenmengen effizient aus einer anderen Quelle.

Visual Basic-Syntax

```
Public NotInheritable Class ULBulkCopy Implements System.IDisposable
```

C#-Syntax

```
public sealed class ULBulkCopy : System.IDisposable
```

Basisklassen

- [System.IDisposable](#)

Mitglieder

Alle Mitglieder der ULBulkCopy-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ULBulkCopy-Konstruktor	Initialisiert ein ULBulkCopy-Objekt mit dem angegebenen ULConnecti-on-Objekt.
Close-Methode	Schließt das ULBulkCopy-Objekt.
Dispose-Methode	Beseitigt das ULBulkCopy-Objekt.
WriteToServer-Methode	Kopiert alle Zeilen im bereitgestellten Array von System.Data.DataRow-Objekten in eine Zieltabelle, die vom DestinationTableName-Feld des ULBulkCopy-Objekts angegeben wird.
BatchSize-Eigenschaft	Ruft die Anzahl der Zeilen in den einzelnen Batches ab oder setzt sie fest.
BulkCopyTimeout-Eigenschaft	Ruft die Anzahl der Sekunden ab bzw. legt die Anzahl der Sekunden fest, nach deren Ablauf der Vorgang abgebrochen wird.
ColumnMappings-Eigen-schaft	Gibt eine Sammlung von ULBulkCopyColumnMapping-Einträgen zurück.
DestinationTableName-Eigenschaft	Ruft den Namen der Zieltabelle auf dem Server ab bzw. legt ihn fest.
NotifyAfter-Eigenschaft	Legt die Anzahl der Zeilen fest, die vor der Erzeugung eines Benachrichti-gungsereignisses verarbeitet werden sollen.
ULRowsCopied-Ereignis	Dieses Ereignis tritt ein, wenn die von der Eigenschaft NotifyAfter ange-gebene Anzahl der Zeilen verarbeitet wurde.

Bemerkungen

Die ULBulkCopy-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

ULBulkCopy-Konstruktor

Initialisiert ein ULBulkCopy-Objekt mit dem angegebenen ULConnection-Objekt.

Überladungsliste

Name	Beschreibung
ULBulkCopy(string)-Konstruktor	Initialisiert ein ULBulkCopy-Objekt mit der angegebenen Verbindungszeichenfolge.
ULBulkCopy(string, ULBulkCopyOptions)-Konstruktor	Initialisiert ein ULBulkCopy-Objekt mit der angegebenen Verbindungszeichenfolge und den angegebenen Kopieroptionen.
ULBulkCopy(ULConnection)-Konstruktor	Initialisiert ein ULBulkCopy-Objekt mit dem angegebenen ULConnection-Objekt.
ULBulkCopy(ULConnection, ULBulkCopyOptions, ULTransaction)-Konstruktor	Initialisiert ein ULBulkCopy-Objekt mit dem angegebenen ULConnection-Objekt, den angegebenen Kopieroptionen und dem angegebenen ULTransaction-Objekt.

ULBulkCopy(string)-Konstruktor

Initialisiert ein ULBulkCopy-Objekt mit der angegebenen Verbindungszeichenfolge.

Visual Basic-Syntax

```
Public Sub New(ByVal connectionString As String)
```

C#-Syntax

```
public ULBulkCopy(string connectionString)
```

Parameter

- **connectionString** Die Zeichenfolge, die die Verbindung festlegt, die zur Verwendung durch das ULBulkCopy-Objekt geöffnet werden soll. Eine Verbindungszeichenfolge ist eine durch Semikola getrennte Liste von Schlüsselwort=Wert-Paaren.

Bemerkungen

Die ULBulkCopy-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

Diese Syntax öffnet eine Verbindung während eines WriteToServer-Methodenaufrufs mit dem connectionString-Wert. Die Verbindung wird am Ende des WriteToServer-Aufrufs geschlossen.

Die Verbindungszeichenfolge kann mit einem ULConnectionParms-Objekt bereitgestellt werden.

Siehe auch

- [ULConnection.ConnectionString-Eigenschaft \[UltraLite.NET\] auf Seite 160](#)
- [ULConnectionParms-Klasse \[UltraLite.NET\] auf Seite 169](#)
- [System.IDisposable](#)

ULBulkCopy(string, ULBulkCopyOptions)-Konstruktor

Initialisiert ein ULBulkCopy-Objekt mit der angegebenen Verbindungszeichenfolge und den angegebenen Kopieroptionen.

Visual Basic-Syntax

```
Public Sub New(  
    ByVal connectionString As String,  
    ByVal copyOptions As ULBulkCopyOptions  
)
```

C#-Syntax

```
public ULBulkCopy(  
    string connectionString,  
    ULBulkCopyOptions copyOptions  
)
```

Parameter

- **connectionString** Die Zeichenfolge, die die Verbindung festlegt, die zur Verwendung durch das ULBulkCopy-Objekt geöffnet werden soll. Eine Verbindungszeichenfolge ist eine durch Semikola getrennte Liste von Schlüsselwort=Wert-Paaren.
- **copyOptions** Eine Kombination von Werten aus der ULBulkCopyOptions-Enumeration, die festlegt, wie Datenquellenzeilen in die Zieltabelle kopiert werden

Bemerkungen

Die ULBulkCopy-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

Diese Syntax öffnet eine Verbindung während eines WriteToServer-Methodenaufrufs mit dem connectionString-Wert. Die Verbindung wird am Ende des WriteToServer-Aufrufs geschlossen.

Siehe auch

- [ULConnection.ConnectionString-Eigenschaft \[UltraLite.NET\] auf Seite 160](#)
- [ULBulkCopyOptions-Enumeration \[UltraLite.NET\] auf Seite 457](#)

ULBulkCopy(ULConnection)-Konstruktor

Initialisiert ein ULBulkCopy-Objekt mit dem angegebenen ULConnection-Objekt.

Visual Basic-Syntax

```
Public Sub New(ByVal connection As ULConnection)
```

C#-Syntax

```
public ULBulkCopy(ULConnection connection)
```

Parameter

- **connection** Das bereits geöffnete ULConnection-Objekt, das für den Massenexport/-import der Daten verwendet wird. Wenn die Verbindung nicht geöffnet ist, wird während eines WriteToServer-Methodenaufrufs eine Ausnahmebedingung ausgegeben.

Bemerkungen

Die ULBulkCopy-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

Siehe auch

- [ULConnection-Klasse \[UltraLite.NET\] auf Seite 121](#)

ULBulkCopy(ULConnection, ULBulkCopyOptions, ULTransaction)-Konstruktor

Initialisiert ein ULBulkCopy-Objekt mit dem angegebenen ULConnection-Objekt, den angegebenen Kopieroptionen und dem angegebenen ULTransaction-Objekt.

Visual Basic-Syntax

```
Public Sub New(  
    ByVal connection As ULConnection,  
    ByVal copyOptions As ULBulkCopyOptions,  
    ByVal externalTransaction As ULTransaction  
)
```

C#-Syntax

```
public ULBulkCopy(  
    ULConnection connection,  
    ULBulkCopyOptions copyOptions,  
    ULTransaction externalTransaction  
)
```

Parameter

- **connection** Das bereits geöffnete ULConnection-Objekt, das für den Massenexport/-import der Daten verwendet wird. Wenn die Verbindung nicht geöffnet ist, wird während eines WriteToServer-Methodenaufrufs eine Ausnahmebedingung ausgegeben.
- **copyOptions** Eine Kombination von Werten aus der ULBulkCopyOptions-Enumeration, die festlegt, wie Datenquellenzeilen in die Zieltabelle kopiert werden
- **externalTransaction** Ein vorhandenes ULTransaction-Objekt, unter dem der Massenexport/-import der Daten ausgeführt wird. Wenn dieser Wert keine Nullreferenz ist ("Nothing" in Visual Basic), wird der Massenexport/-import der Daten darin ausgeführt. Es wird ein Fehler ausgegeben, wenn sowohl eine externe Transaktion als auch die Option ULBulkCopyOptions.UseInternalTransaction festgelegt wird.

Bemerkungen

Die ULBulkCopy-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

Siehe auch

- [ULConnection-Klasse \[UltraLite.NET\] auf Seite 121](#)
- [ULTransaction-Klasse \[UltraLite.NET\] auf Seite 450](#)

Close-Methode

Schließt das ULBulkCopy-Objekt.

Visual Basic-Syntax

```
Public Sub Close()
```

C#-Syntax

```
public void Close()
```

Dispose-Methode

Beseitigt das ULBulkCopy-Objekt.

Visual Basic-Syntax

```
Public Sub Dispose()
```

C#-Syntax

```
public void Dispose()
```

WriteToServer-Methode

Kopiert alle Zeilen im bereitgestellten Array von System.Data.DataRow-Objekten in eine Zieltabelle, die vom DestinationTableName-Feld des ULBulkCopy-Objekts angegeben wird.

Überladungsliste

Name	Beschreibung
WriteToServer(DataRow[])-Methode	Kopiert alle Zeilen im bereitgestellten Array von System.Data.DataRow-Objekten in eine Zieltabelle, die vom DestinationTableName-Feld des ULBulkCopy-Objekts angegeben wird.
WriteToServer(DataTable)-Methode	Kopiert alle Zeilen in der mit System.Data.DataTable bereitgestellten Tabelle in eine Zieltabelle, die durch die ULBulkCopy.DestinationTableName-Eigenschaft angegeben wird.

Name	Beschreibung
WriteToServer(DataTable, DataRowState)-Methode	Kopiert alle Zeilen, die in der mit System.Data.DataTable bereitgestellten Tabelle den angegebenen Zeilenstatus aufweisen, in eine Zieltabelle, die durch die ULBulkCopy.DestinationTableName-Eigenschaft angegeben wird.
WriteToServer(IDataReader)-Methode	Kopiert alle Zeilen in dem mit System.Data.IDataReader bereitgestellten Leser in eine Zieltabelle, die durch die ULBulkCopy.DestinationTableName-Eigenschaft angegeben wird.

WriteToServer(DataRow[])-Methode

Kopiert alle Zeilen im bereitgestellten Array von System.Data.DataRow-Objekten in eine Zieltabelle, die vom DestinationTableName-Feld des ULBulkCopy-Objekts angegeben wird.

Visual Basic-Syntax

```
Public Sub WriteToServer(ByVal rows As DataRow())
```

C#-Syntax

```
public void WriteToServer(DataRow[] rows)
```

Parameter

- **rows** Ein Array von System.Data.DataRow-Objekten, die in die Zieltabelle kopiert werden.

Siehe auch

- [ULBulkCopy.DestinationTableName-Eigenschaft \[UltraLite.NET\] auf Seite 55](#)
- [System.Data.DataRow](#)

WriteToServer(DataTable)-Methode

Kopiert alle Zeilen in der mit System.Data.DataTable bereitgestellten Tabelle in eine Zieltabelle, die durch die ULBulkCopy.DestinationTableName-Eigenschaft angegeben wird.

Visual Basic-Syntax

```
Public Sub WriteToServer(ByVal table As DataTable)
```

C#-Syntax

```
public void WriteToServer(DataTable table)
```

Parameter

- **table** Eine System.Data.DataTable-Tabelle, deren Zeilen in die Zieltabelle kopiert werden sollen.

Siehe auch

- [ULBulkCopy.DestinationTableName-Eigenschaft \[UltraLite.NET\] auf Seite 55](#)
- [System.Data.DataTable](#)

WriteToServer(DataTable, DataRowState)-Methode

Kopiert alle Zeilen, die in der mit System.Data.DataTable bereitgestellten Tabelle den angegebenen Zeilenstatus aufweisen, in eine Zieltabelle, die durch die ULBulkCopy.DestinationTableName-Eigenschaft angegeben wird.

Visual Basic-Syntax

```
Public Sub WriteToServer(  
    ByVal table As DataTable,  
    ByVal rowState As DataRowState  
)
```

C#-Syntax

```
public void WriteToServer(DataTable table, DataRowState rowState)
```

Parameter

- **table** Eine System.Data.DataTable-Tabelle, deren Zeilen in die Zieltabelle kopiert werden sollen.
- **rowState** Ein Wert aus der System.Data.DataRowState-Enumeration. Nur Zeilen mit diesem Zeilenstatus werden in die Zieltabelle kopiert.

Bemerkungen

Wenn der rowState-Parameter angegeben ist, werden nur die Zeilen kopiert, die denselben Zeilenstatus haben.

Siehe auch

- [ULBulkCopy.DestinationTableName-Eigenschaft \[UltraLite.NET\] auf Seite 55](#)
- [System.Data.DataTable](#)
- [System.Data.DataRowState](#)

WriteToServer(IDataReader)-Methode

Kopiert alle Zeilen in dem mit System.Data.IDataReader bereitgestellten Leser in eine Zieltabelle, die durch die ULBulkCopy.DestinationTableName-Eigenschaft angegeben wird.

Visual Basic-Syntax

```
Public Sub WriteToServer(ByVal reader As IDataReader)
```

C#-Syntax

```
public void WriteToServer(IDataReader reader)
```

Parameter

- **reader** Ein System.Data.IDataReader-Objekt, dessen Zeilen in die Zieltabelle kopiert werden sollen.

Siehe auch

- [ULBulkCopy.DestinationTableName-Eigenschaft \[UltraLite.NET\] auf Seite 55](#)
- [System.Data.IDataReader](#)

BatchSize-Eigenschaft

Ruft die Anzahl der Zeilen in den einzelnen Batches ab oder setzt sie fest.

Visual Basic-Syntax

```
Public Property BatchSize As Integer
```

C#-Syntax

```
public int BatchSize {get;set;}
```

Bemerkungen

Am Ende der einzelnen Batches werden die enthaltenen Zeilen an den Server gesendet.

Die Anzahl von Zeilen in jedem Batch. Standardwert ist "0".

Wenn Sie den Wert auf Null setzen, werden alle Zeilen in einem Batch gesendet.

Die Verwendung eines negativen Werts wird als Fehler behandelt.

Wenn dieser Wert geändert wird, während ein Batch ausgeführt wird, wird der aktuelle Batch abgeschlossen und alle folgenden Batches verwenden den neuen Wert.

BulkCopyTimeout-Eigenschaft

Ruft die Anzahl der Sekunden ab bzw. legt die Anzahl der Sekunden fest, nach deren Ablauf der Vorgang abgebrochen wird.

Visual Basic-Syntax

```
Public Property BulkCopyTimeout As Integer
```

C#-Syntax

```
public int BulkCopyTimeout {get;set;}
```

Bemerkungen

Standardwert ist 30 Sekunden.

Der Wert Null bedeutet, dass keine Beschränkung festgelegt wird. Dies sollte vermieden werden, weil es zu endlosen Wartezeiten führen kann.

Wenn die Zeit für den Vorgang überschritten wird, werden alle Zeilen in der aktuellen Transaktion zurückgesetzt und ein `SAException`-Fehler wird ausgegeben.

Wenn Sie einen Wert festlegen, der kleiner ist als Null, wird ein Fehler ausgegeben.

ColumnMappings-Eigenschaft

Gibt eine Sammlung von `ULBulkCopyColumnMapping`-Einträgen zurück.

Visual Basic-Syntax

```
Public ReadOnly Property ColumnMappings As  
    ULBulkCopyColumnMappingCollection
```

C#-Syntax

```
public ULBulkCopyColumnMappingCollection ColumnMappings {get;}
```

Bemerkungen

Spaltenzuordnungen definieren die Beziehungen zwischen Spalten in der Datenquelle und Spalten im Datenziel.

Standardmäßig ist dies eine leere Sammlung.

Die Eigenschaft kann nicht geändert werden, während der `WriteToServer`-Methodenaufruf ausgeführt wird.

Wenn das `ColumnMappings`-Objekt beim Ausführen der `WriteToServer`-Methode leer ist, wird die erste Spalte in der Quelle der ersten Spalte im Ziel zugeordnet, die zweite Spalte der zweiten Spalte usw. Dieser Vorgang wird ausgeführt, sofern die Spaltentypen konvertierbar sind, mindestens ebenso viele Zielspalten wie Quellspalten vorhanden sind und eventuelle zusätzliche Zielspalten nullwertfähig sein können.

Siehe auch

- [ULBulkCopyColumnMapping-Klasse \[UltraLite.NET\] auf Seite 57](#)

DestinationTableName-Eigenschaft

Ruft den Namen der Zieltabelle auf dem Server ab bzw. legt ihn fest.

Visual Basic-Syntax

```
Public Property DestinationTableName As String
```

C#-Syntax

```
public string DestinationTableName {get;set;}
```

Bemerkungen

Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Wenn der Wert während der Ausführung eines WriteToServer-Aufrufs geändert wird, hat die Änderung keine Wirkung.

Wenn der Wert vor einem Aufruf der WriteToServer-Methode nicht bereits gesetzt wurde, wird ein InvalidOperationException-Fehler ausgegeben.

Es ist ein Fehler, den Wert auf Null ("Nothing" in Visual Basic) oder die leere Zeichenfolge zu setzen.

NotifyAfter-Eigenschaft

Legt die Anzahl der Zeilen fest, die vor der Erzeugung eines Benachrichtigungsereignisses verarbeitet werden sollen.

Visual Basic-Syntax

```
Public Property NotifyAfter As Integer
```

C#-Syntax

```
public int NotifyAfter {get;set;}
```

Bemerkungen

Eine Ganzzahl, die die Anzahl von Zeilen repräsentiert, die verarbeitet werden müssen, bevor ein Benachrichtigungsereignis generiert wird, oder Null, wenn die Eigenschaft nicht festgelegt wurde.

Änderungen an dieser Eigenschaft, die während des Ausführens der WriteToServer-Methode vorgenommen wurden, werden erst nach der nächsten Benachrichtigung wirksam.

Wenn dieser Wert auf einen kleineren Wert als Null gesetzt wurde, wird ein Fehler ausgegeben.

Die Werte der Eigenschaften NotifyAfter und BulkCopyTimeOut schließen einander aus, sodass das Ereignis auch dann ausgelöst werden kann, wenn keine Zeilen an die Datenbank gesendet oder festgeschrieben wurden.

Siehe auch

- [ULBulkCopy.BulkCopyTimeout-Eigenschaft \[UltraLite.NET\] auf Seite 54](#)

ULRowsCopied-Ereignis

Dieses Ereignis tritt ein, wenn die von der Eigenschaft NotifyAfter angegebene Anzahl der Zeilen verarbeitet wurde.

Visual Basic-Syntax

```
Public Event ULRowsCopied As ULRowsCopiedEventHandler
```

C#-Syntax

```
public event ULRowsCopiedEventHandler ULRowsCopied;
```

Bemerkungen

Der Empfang eines ULRowsCopied-Ereignisses bedeutet nicht, dass Zeilen festgeschrieben wurden. Es ist nicht möglich, die Methode "Close" von diesem Ereignis aus aufzurufen.

Siehe auch

- [ULBulkCopy.NotifyAfter-Eigenschaft \[UltraLite.NET\] auf Seite 56](#)

ULBulkCopyColumnMapping-Klasse

Definiert die Zuordnung zwischen einer Spalte in der Datenquelle einer ULBulkCopy-Instanz und einer Spalte in der Zieltabelle der Instanz.

Visual Basic-Syntax

```
Public NotInheritable Class ULBulkCopyColumnMapping
```

C#-Syntax

```
public sealed class ULBulkCopyColumnMapping
```

Mitglieder

Alle Mitglieder der ULBulkCopyColumnMapping-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ULBulkCopyColumnMapping-Konstruktor	Erstellt eine neue Spaltenzuordnung.
DestinationColumn-Eigenschaft	Legt den Namen der Spalte fest, die in der Tabelle der Zieldatenbank zugeordnet wird.
DestinationOrdinal-Eigenschaft	Legt den Ordinalwert der Spalte fest, die in der Tabelle der Zieldatenbank zugeordnet wird.
SourceColumn-Eigenschaft	Legt den Namen der in der Datenquelle zugeordneten Spalte fest.
SourceOrdinal-Eigenschaft	Legt die Ordinalposition der Quellspalte innerhalb der Datenquelle fest.

Bemerkungen

Die ULBulkCopyColumnMapping-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

Siehe auch

- [ULBulkCopy-Klasse \[UltraLite.NET\] auf Seite 46](#)

ULBulkCopyColumnMapping-Konstruktor

Erstellt eine neue Spaltenzuordnung.

Überladungsliste

Name	Beschreibung
ULBulkCopyColumnMapping()-Konstruktor	Erstellt eine neue Spaltenzuordnung.
ULBulkCopyColumnMapping(int, int)-Konstruktor	Erstellt eine neue Spaltenzuordnung, wobei Ordinalzahlen oder Namen zur Angabe von Quell- und Zielspalten verwendet werden.
ULBulkCopyColumnMapping(int, string)-Konstruktor	Erstellt eine neue Spaltenzuordnung unter Verwendung einer Spaltenordinalzahl, um auf die Quellspalte zu verweisen, und eines Spaltennamens, um auf die Zielspalte zu verweisen.
ULBulkCopyColumnMapping(string, int)-Konstruktor	Erstellt eine neue Spaltenzuordnung unter Verwendung eines Spaltennamens, um auf die Quellspalte zu verweisen, und einer Spaltenordinalzahl, um auf die Zielspalte zu verweisen.
ULBulkCopyColumnMapping(string, string)-Konstruktor	Erstellt eine neue Spaltenzuordnung, wobei Spaltennamen zur Angabe von Quell- und Zielspalten verwendet werden.

ULBulkCopyColumnMapping()-Konstruktor

Erstellt eine neue Spaltenzuordnung.

Visual Basic-Syntax

```
Public Sub New()
```

C#-Syntax

```
public ULBulkCopyColumnMapping()
```

Bemerkungen

Die ULBulkCopyColumnMapping-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

ULBulkCopyColumnMapping(int, int)-Konstruktor

Erstellt eine neue Spaltenzuordnung, wobei Ordinalzahlen oder Namen zur Angabe von Quell- und Zielspalten verwendet werden.

Visual Basic-Syntax

```
Public Sub New(  
    ByVal sourceColumnOrdinal As Integer,
```

```
        ByVal destinationColumnOrdinal As Integer  
    )
```

C#-Syntax

```
public ULBulkCopyColumnMapping(  
    int sourceColumnOrdinal,  
    int destinationColumnOrdinal  
)
```

Parameter

- **sourceColumnOrdinal** Die Ordinalposition der Quellspalte innerhalb der Datenquelle. Die erste Spalte in einer Datenquelle hat die Ordinalposition Null.
- **destinationColumnOrdinal** Die Ordinalposition der Zielspalte innerhalb der Zieltabelle. Die erste Spalte in einer Tabelle hat die Ordinalposition Null.

Bemerkungen

Die ULBulkCopyColumnMapping-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

ULBulkCopyColumnMapping(int, string)-Konstruktor

Erstellt eine neue Spaltenzuordnung unter Verwendung einer Spaltenordinalzahl, um auf die Quellspalte zu verweisen, und eines Spaltennamens, um auf die Zielspalte zu verweisen.

Visual Basic-Syntax

```
Public Sub New(  
    ByVal sourceColumnOrdinal As Integer,  
    ByVal destinationColumn As String  
)
```

C#-Syntax

```
public ULBulkCopyColumnMapping(  
    int sourceColumnOrdinal,  
    string destinationColumn  
)
```

Parameter

- **sourceColumnOrdinal** Die Ordinalposition der Quellspalte innerhalb der Datenquelle. Die erste Spalte in einer Datenquelle hat die Ordinalposition Null.
- **destinationColumn** Der Name der Zielspalte innerhalb der Zieltabelle.

Bemerkungen

Die ULBulkCopyColumnMapping-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

ULBulkCopyColumnMapping(string, int)-Konstruktor

Erstellt eine neue Spaltenzuordnung unter Verwendung eines Spaltennamens, um auf die Quellspalte zu verweisen, und einer Spaltenordinalzahl, um auf die Zielspalte zu verweisen.

Visual Basic-Syntax

```
Public Sub New(  
    ByVal sourceColumn As String,  
    ByVal destinationColumnOrdinal As Integer  
)
```

C#-Syntax

```
public ULBulkCopyColumnMapping(  
    string sourceColumn,  
    int destinationColumnOrdinal  
)
```

Parameter

- **sourceColumn** Der Name der Quellspalte innerhalb der Datenquelle.
- **destinationColumnOrdinal** Die Ordinalposition der Zielspalte innerhalb der Zieltabelle. Die erste Spalte in einer Tabelle hat die Ordinalposition Null.

Bemerkungen

Die ULBulkCopyColumnMapping-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

ULBulkCopyColumnMapping(string, string)-Konstruktor

Erstellt eine neue Spaltenzuordnung, wobei Spaltennamen zur Angabe von Quell- und Zielspalten verwendet werden.

Visual Basic-Syntax

```
Public Sub New(  
    ByVal sourceColumn As String,  
    ByVal destinationColumn As String  
)
```

C#-Syntax

```
public ULBulkCopyColumnMapping(  
    string sourceColumn,  
    string destinationColumn  
)
```

Parameter

- **sourceColumn** Der Name der Quellspalte innerhalb der Datenquelle.
- **destinationColumn** Der Name der Zielspalte innerhalb der Zieltabelle.

Bemerkungen

Die ULBulkCopyColumnMapping-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

DestinationColumn-Eigenschaft

Legt den Namen der Spalte fest, die in der Tabelle der Zieldatenbank zugeordnet wird.

Visual Basic-Syntax

```
Public Property DestinationColumn As String
```

C#-Syntax

```
public string DestinationColumn {get;set;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen der Spalte in der Zieltabelle oder eine Nullreferenz ("Nothing" in Visual Basic) angibt, wenn die DestinationOrdinal-Eigenschaft Priorität hat.

Die Eigenschaften DestinationColumn und DestinationOrdinal schließen einander aus. Der zuletzt festgelegte Wert hat Vorrang.

Die Festlegung der Eigenschaft DestinationColumn bewirkt, dass die Eigenschaft DestinationOrdinal auf -1 gesetzt wird. Wenn die Eigenschaft DestinationOrdinal festgelegt wird, erhält die Eigenschaft DestinationColumn den Wert einer Nullreferenz ("Nothing" in Visual Basic).

Es ist ein Fehler, die DestinationColumn-Eigenschaft auf Null oder die leere Zeichenfolge zu setzen.

Siehe auch

- [ULBulkCopyColumnMapping.DestinationOrdinal-Eigenschaft \[UltraLite.NET\] auf Seite 61](#)

DestinationOrdinal-Eigenschaft

Legt den Ordinalwert der Spalte fest, die in der Tabelle der Zieldatenbank zugeordnet wird.

Visual Basic-Syntax

```
Public Property DestinationOrdinal As Integer
```

C#-Syntax

```
public int DestinationOrdinal {get;set;}
```

Bemerkungen

Eine Ganzzahl, die den Ordinalwert der Spalte angibt, die in der Zieltabelle zugeordnet wird, oder -1, wenn die Eigenschaft nicht festgelegt wurde.

Die Eigenschaften DestinationColumn und DestinationOrdinal schließen einander aus. Der zuletzt festgelegte Wert hat Vorrang.

Die Festlegung der Eigenschaft `DestinationColumn` bewirkt, dass die Eigenschaft `DestinationOrdinal` auf -1 gesetzt wird. Wenn die Eigenschaft `DestinationOrdinal` festgelegt wird, erhält die Eigenschaft `DestinationColumn` den Wert einer Nullreferenz ("Nothing" in Visual Basic).

Siehe auch

- [ULBulkCopyColumnMapping.DestinationColumn-Eigenschaft \[UltraLite.NET\] auf Seite 61](#)

SourceColumn-Eigenschaft

Legt den Namen der in der Datenquelle zugeordneten Spalte fest.

Visual Basic-Syntax

```
Public Property SourceColumn As String
```

C#-Syntax

```
public string SourceColumn {get;set;}
```

Bemerkungen

Eine Zeichenfolge, die den Name der Spalte in der Datenquelle oder eine Nullreferenz ("Nothing" in Visual Basic) angibt, wenn `SourceOrdinal` Priorität hat.

Die Eigenschaften `SourceColumn` und `SourceOrdinal` schließen einander aus. Der zuletzt festgelegte Wert hat Vorrang.

Die Festlegung der Eigenschaft `SourceColumn` bewirkt, dass die Eigenschaft `SourceOrdinal` auf -1 gesetzt wird. Wenn die Eigenschaft `SourceOrdinal` festgelegt wird, erhält die Eigenschaft `SourceColumn` den Wert einer Nullreferenz ("Nothing" in Visual Basic).

Es ist ein Fehler, die `SourceColumn`-Eigenschaft auf Null oder die leere Zeichenfolge zu setzen.

Siehe auch

- [ULBulkCopyColumnMapping.SourceOrdinal-Eigenschaft \[UltraLite.NET\] auf Seite 62](#)

SourceOrdinal-Eigenschaft

Legt die Ordinalposition der Quellspalte innerhalb der Datenquelle fest.

Visual Basic-Syntax

```
Public Property SourceOrdinal As Integer
```

C#-Syntax

```
public int SourceOrdinal {get;set;}
```

Bemerkungen

Eine Ganzzahl, die den Ordinalwert der Spalte in der Datenquelle angibt, oder -1, wenn die Eigenschaft nicht festgelegt wurde.

Die Eigenschaften SourceColumn und SourceOrdinal schließen einander aus. Der zuletzt festgelegte Wert hat Vorrang.

Die Festlegung der Eigenschaft SourceColumn bewirkt, dass die Eigenschaft SourceOrdinal auf -1 gesetzt wird. Wenn die Eigenschaft SourceOrdinal festgelegt wird, erhält die Eigenschaft SourceColumn den Wert einer Nullreferenz ("Nothing" in Visual Basic).

Siehe auch

- [ULBulkCopyColumnMapping.SourceColumn-Eigenschaft \[UltraLite.NET\] auf Seite 62](#)

ULBulkCopyColumnMappingCollection-Klasse

Eine Sammlung von ULBulkCopyColumnMapping-Objekten, die Werte von System.Collections.CollectionBase erbt.

Visual Basic-Syntax

```
Public NotInheritable Class ULBulkCopyColumnMappingCollection
    Inherits System.Collections.CollectionBase
```

C#-Syntax

```
public sealed class ULBulkCopyColumnMappingCollection :
    System.Collections.CollectionBase
```

Basisklassen

- [System.Collections.CollectionBase](#)

Mitglieder

Alle Mitglieder der ULBulkCopyColumnMappingCollection-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
Add-Methode	Fügt das angegebene ULBulkCopyColumnMapping-Objekt der Sammlung hinzu.
Clear-Methode (geerbt aus System.Collections.CollectionBase)	Entfernt alle Objekte aus der CollectionBase -Instanz.
Contains-Methode	Gibt zurück, ob das angegebenen ULBulkCopyColumnMapping-Objekt in der Sammlung vorhanden ist.

Name	Beschreibung
CopyTo-Methode	Kopiert die Elemente des ULBulkCopyColumnMappingCollection-Objekts in ein Array von ULBulkCopyColumnMapping-Objekten, beginnend an einem bestimmten Index.
GetEnumerator-Methode (geerbt aus System.Collections.CollectionBase)	Gibt einen Enumerator zurück, der die Objekte der System.Collections.CollectionBase -Instanz durchläuft.
IndexOf-Methode	Gibt den Index des angegebenen ULBulkCopyColumnMapping-Objekts in der Sammlung zurück.
OnClear-Methode (geerbt aus System.Collections.CollectionBase)	Führt beim Löschen des Inhalts der System.Collections.CollectionBase -Instanz zusätzliche benutzerdefinierte Prozesse aus.
OnClearComplete-Methode (geerbt aus System.Collections.CollectionBase)	Führt nach dem Löschen des Inhalts der System.Collections.CollectionBase -Instanz zusätzliche benutzerdefinierte Prozesse aus.
OnInsert-Methode (geerbt aus System.Collections.CollectionBase)	Führt vor dem Einfügen eines neuen Elements in die System.Collections.CollectionBase -Instanz zusätzliche benutzerdefinierte Prozesse aus.
OnInsertComplete-Methode (geerbt aus System.Collections.CollectionBase)	Führt nach dem Einfügen eines neuen Elements in die System.Collections.CollectionBase -Instanz zusätzliche benutzerdefinierte Prozesse aus.
OnRemove-Methode (geerbt aus System.Collections.CollectionBase)	Führt beim Entfernen eines Elements aus der System.Collections.CollectionBase -Instanz zusätzliche benutzerdefinierte Prozesse aus.
OnRemoveComplete-Methode (geerbt aus System.Collections.CollectionBase)	Führt nach dem Entfernen eines Elements aus der System.Collections.CollectionBase -Instanz zusätzliche benutzerdefinierte Prozesse aus.
OnSet-Methode (geerbt aus System.Collections.CollectionBase)	Führt vor dem Einstellen eines Werts in der System.Collections.CollectionBase -Instanz zusätzliche benutzerdefinierte Prozesse aus.
OnSetComplete-Methode (geerbt aus System.Collections.CollectionBase)	Führt nach Einstellung eines Werts in der System.Collections.CollectionBase -Instanz zusätzliche benutzerdefinierte Prozesse aus.
OnValidate-Methode (geerbt aus System.Collections.CollectionBase)	Führt beim Validieren eines Werts zusätzliche benutzerdefinierte Prozesse aus.

Name	Beschreibung
Remove-Methode	Entfernt das angegebene ULBulkCopyColumnMapping-Objekt aus dem ULBulkCopyColumnMappingCollection-Objekt.
RemoveAt-Methode	Entfernt die Zuordnung am angegebenen Index aus der Sammlung.
Capacity-Eigenschaft (geerbt aus System.Collections.CollectionBase)	Ruft die Anzahl der Elemente ab, die System.Collections.CollectionBase enthalten kann, oder legt sie fest.
Count-Eigenschaft (geerbt aus System.Collections.CollectionBase)	Ruft die Anzahl der Elemente ab, die in der System.Collections.CollectionBase -Instanz enthalten sind.
InnerList-Eigenschaft (geerbt aus System.Collections.CollectionBase)	Bezieht eine System.Collections.ArrayList mit der Liste der Elemente in der System.Collections.CollectionBase -Instanz.
List-Eigenschaft (geerbt aus System.Collections.CollectionBase)	Bezieht eine System.Collections.IList mit der Liste der Elemente in der System.Collections.CollectionBase -Instanz.
this-Eigenschaft	Ruft das ULBulkCopyColumnMapping-Objekt am angegebenen Index ab.

Bemerkungen

Die ULBulkCopyColumnMappingCollection-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

Siehe auch

- [ULBulkCopyColumnMapping-Klasse \[UltraLite.NET\] auf Seite 57](#)

Add-Methode

Fügt das angegebene ULBulkCopyColumnMapping-Objekt der Sammlung hinzu.

Überladungsliste

Name	Beschreibung
Add(Int, Int)-Methode	Erstellt ein neues ULBulkCopyColumnMapping-Objekt unter Verwendung der Ordinalzahlen für Quell- und Zielspalte und fügt die Zuordnung zur Sammlung hinzu.

Name	Beschreibung
Add(Int, String)-Methode	Erstellt ein neues ULBulkCopyColumnMapping-Objekt, in dem mit einer Spaltenordinalzahl auf die Quellspalte und mit einem Spaltennamen auf die Zielspalte verwiesen wird, und fügt die Zuordnung zur Sammlung hinzu.
Add(String, Int)-Methode	Erstellt ein neues ULBulkCopyColumnMapping-Objekt, in dem mit einem Spaltennamen auf die Quellspalte und mit einer Spaltenordinalzahl auf die Zielspalte verwiesen wird, und fügt die Zuordnung zur Sammlung hinzu.
Add(String, String)-Methode	Erstellt ein neues ULBulkCopyColumnMapping-Objekt unter Verwendung der Spaltennamen für Quell- und Zielspalte und fügt die Zuordnung zur Sammlung hinzu.
Add(ULBulkCopyColumnMapping)-Methode	Fügt das angegebene ULBulkCopyColumnMapping-Objekt der Sammlung hinzu.

Add(Int, Int)-Methode

Erstellt ein neues ULBulkCopyColumnMapping-Objekt unter Verwendung der Ordinalzahlen für Quell- und Zielspalte und fügt die Zuordnung zur Sammlung hinzu.

Visual Basic-Syntax

```
Public Function Add(  
    ByVal sourceColumnOrdinal As Integer,  
    ByVal destinationColumnOrdinal As Integer  
) As ULBulkCopyColumnMapping
```

C#-Syntax

```
public ULBulkCopyColumnMapping Add(  
    int sourceColumnOrdinal,  
    int destinationColumnOrdinal  
)
```

Parameter

- **sourceColumnOrdinal** Die Ordinalposition der Quellspalte innerhalb der Datenquelle. Die erste Spalte in einer Datenquelle hat die Ordinalposition Null.
- **destinationColumnOrdinal** Die Ordinalposition der Zielspalte innerhalb der Zieltabelle. Die erste Spalte in einer Tabelle hat die Ordinalposition Null.

Bemerkungen

Die ULBulkCopyColumnMappingCollection-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

Siehe auch

- [ULBulkCopyColumnMapping-Klasse \[UltraLite.NET\] auf Seite 57](#)

Add(Int, String)-Methode

Erstellt ein neues ULBulkCopyColumnMapping-Objekt, in dem mit einer Spaltenordinalzahl auf die Quellspalte und mit einem Spaltennamen auf die Zielspalte verwiesen wird, und fügt die Zuordnung zur Sammlung hinzu.

Visual Basic-Syntax

```
Public Function Add(  
    ByVal sourceColumnOrdinal As Integer,  
    ByVal destinationColumn As String  
) As ULBulkCopyColumnMapping
```

C#-Syntax

```
public ULBulkCopyColumnMapping Add(  
    int sourceColumnOrdinal,  
    string destinationColumn  
)
```

Parameter

- **sourceColumnOrdinal** Die Ordinalposition der Quellspalte innerhalb der Datenquelle. Die erste Spalte in einer Datenquelle hat die Ordinalposition Null.
- **destinationColumn** Der Name der Zielspalte innerhalb der Zieltabelle.

Bemerkungen

Die ULBulkCopyColumnMappingCollection-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

Siehe auch

- [ULBulkCopyColumnMapping-Klasse \[UltraLite.NET\] auf Seite 57](#)

Add(String, Int)-Methode

Erstellt ein neues ULBulkCopyColumnMapping-Objekt, in dem mit einem Spaltennamen auf die Quellspalte und mit einer Spaltenordinalzahl auf die Zielspalte verwiesen wird, und fügt die Zuordnung zur Sammlung hinzu.

Visual Basic-Syntax

```
Public Function Add(  
    ByVal sourceColumn As String,  
    ByVal destinationColumnOrdinal As Integer  
) As ULBulkCopyColumnMapping
```

C#-Syntax

```
public ULBulkCopyColumnMapping Add(  
    string sourceColumn,  
    int destinationColumnOrdinal  
)
```

Parameter

- **sourceColumn** Der Name der Quellspalte innerhalb der Datenquelle.
- **destinationColumnOrdinal** Die Ordinalposition der Zielspalte innerhalb der Zieltabelle. Die erste Spalte in einer Tabelle hat die Ordinalposition Null.

Bemerkungen

Erstellt eine neue Spaltenzuordnung, wobei Ordinalzahlen oder Namen zur Angabe von Quell- und Zielspalten verwendet werden.

Die ULBulkCopyColumnMappingCollection-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

Siehe auch

- [ULBulkCopyColumnMapping-Klasse \[UltraLite.NET\] auf Seite 57](#)

Add(String, String)-Methode

Erstellt ein neues ULBulkCopyColumnMapping-Objekt unter Verwendung der Spaltennamen für Quell- und Zielspalte und fügt die Zuordnung zur Sammlung hinzu.

Visual Basic-Syntax

```
Public Function Add(  
    ByVal sourceColumn As String,  
    ByVal destinationColumn As String  
) As ULBulkCopyColumnMapping
```

C#-Syntax

```
public ULBulkCopyColumnMapping Add(  
    string sourceColumn,  
    string destinationColumn  
)
```

Parameter

- **sourceColumn** Der Name der Quellspalte innerhalb der Datenquelle.
- **destinationColumn** Der Name der Zielspalte innerhalb der Zieltabelle.

Bemerkungen

Die ULBulkCopyColumnMappingCollection-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

Siehe auch

- [ULBulkCopyColumnMapping-Klasse \[UltraLite.NET\] auf Seite 57](#)

Add(ULBulkCopyColumnMapping)-Methode

Fügt das angegebene ULBulkCopyColumnMapping-Objekt der Sammlung hinzu.

Visual Basic-Syntax

```
Public Function Add(  
    ByVal bulkCopyColumnMapping As ULBulkCopyColumnMapping  
) As ULBulkCopyColumnMapping
```

C#-Syntax

```
public ULBulkCopyColumnMapping Add(  
    ULBulkCopyColumnMapping bulkCopyColumnMapping  
)
```

Parameter

- **bulkCopyColumnMapping** Das ULBulkCopyColumnMapping-Objekt, das die Zuordnung beschreibt, die zur Sammlung hinzugefügt werden soll.

Bemerkungen

Die ULBulkCopyColumnMappingCollection-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

Siehe auch

- [ULBulkCopyColumnMapping-Klasse \[UltraLite.NET\] auf Seite 57](#)

Contains-Methode

Gibt zurück, ob das angegebenen ULBulkCopyColumnMapping-Objekt in der Sammlung vorhanden ist.

Visual Basic-Syntax

```
Public Function Contains(  
    ByVal value As ULBulkCopyColumnMapping  
) As Boolean
```

C#-Syntax

```
public bool Contains(ULBulkCopyColumnMapping value)
```

Parameter

- **value** Ein gültiges ULBulkCopyColumnMapping-Objekt

Rückgabe

TRUE, wenn die angegebene Zuordnung in der Sammlung vorhanden ist, ansonsten FALSE.

Siehe auch

- [ULBulkCopyColumnMapping-Klasse \[UltraLite.NET\] auf Seite 57](#)

CopyTo-Methode

Kopiert die Elemente des ULBulkCopyColumnMappingCollection-Objekts in ein Array von ULBulkCopyColumnMapping-Objekten, beginnend an einem bestimmten Index.

Visual Basic-Syntax

```
Public Sub CopyTo(  
    ByVal array As ULBulkCopyColumnMapping(),  
    ByVal index As Integer  
)
```

C#-Syntax

```
public void CopyTo(ULBulkCopyColumnMapping[] array, int index)
```

Parameter

- **array** Das eindimensionale ULBulkCopyColumnMapping-Array, das das Ziel für die aus diesem ULBulkCopyColumnMappingCollection-Objekt kopierten Elemente ist. Das Array muss eine auf Null basierende Indizierung haben.
- **index** Der auf Null basierende Index im Array, an dem der Kopiervorgang beginnt

Siehe auch

- [ULBulkCopyColumnMapping-Klasse \[UltraLite.NET\] auf Seite 57](#)

IndexOf-Methode

Gibt den Index des angegebenen ULBulkCopyColumnMapping-Objekts in der Sammlung zurück.

Visual Basic-Syntax

```
Public Function IndexOf(  
    ByVal value As ULBulkCopyColumnMapping  
) As Integer
```

C#-Syntax

```
public int IndexOf(ULBulkCopyColumnMapping value)
```

Parameter

- **value** Das ULBulkCopyColumnMapping-Objekt, das gesucht werden soll

Rückgabe

Der Rückgabewert ist der auf Null basierende Index der Spaltenzuordnung. Wenn die Spaltenzuordnung nicht in der Sammlung gefunden wird, wird -1 zurückgegeben.

Siehe auch

- [ULBulkCopyColumnMapping-Klasse \[UltraLite.NET\] auf Seite 57](#)

Remove-Methode

Entfernt das angegebene ULBulkCopyColumnMapping-Objekt aus dem ULBulkCopyColumnMappingCollection-Objekt.

Visual Basic-Syntax

```
Public Sub Remove(ByVal value As ULBulkCopyColumnMapping)
```

C#-Syntax

```
public void Remove(ULBulkCopyColumnMapping value)
```

Parameter

- **value** Das ULBulkCopyColumnMapping-Objekt, das aus der Sammlung entfernt werden soll

Siehe auch

- [ULBulkCopyColumnMapping-Klasse \[UltraLite.NET\] auf Seite 57](#)

RemoveAt-Methode

Entfernt die Zuordnung am angegebenen Index aus der Sammlung.

Visual Basic-Syntax

```
Public Shadows Sub RemoveAt(ByVal index As Integer)
```

C#-Syntax

```
public new void RemoveAt(int index)
```

Parameter

- **index** Der auf Null basierende Index des ULBulkCopyColumnMapping-Objekts, der aus der Sammlung entfernt werden soll

this-Eigenschaft

Ruft das ULBulkCopyColumnMapping-Objekt am angegebenen Index ab.

Visual Basic-Syntax

```
Public ReadOnly Property Item(  
    ByVal index As Integer  
) As ULBulkCopyColumnMapping
```

C#-Syntax

```
public ULBulkCopyColumnMapping this[int index] {get;}
```

Parameter

- **index** Der auf Null basierende Index des zu suchenden ULBulkCopyColumnMapping-Objekts.

Bemerkungen

Es wird ein ULBulkCopyColumnMapping-Objekt zurückgegeben.

Siehe auch

- [ULBulkCopyColumnMapping-Klasse \[UltraLite.NET\] auf Seite 57](#)

ULCommand-Klasse

Repräsentiert eine vorkompilierte SQL-Anweisung oder Abfrage mit oder ohne IN-Parameter.

Visual Basic-Syntax

```
Public NotInheritable Class ULCommand  
    Inherits System.Data.Common.DbCommand  
    Implements System.ICloneable
```

C#-Syntax

```
public sealed class ULCommand :  
    System.Data.Common.DbCommand,  
    System.ICloneable
```

Basisklassen

- [System.Data.Common.DbCommand](#)
- [System.ICloneable](#)

Mitglieder

Alle Mitglieder der ULCommand-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ULCommand-Konstruktor	Initialisiert ein ULCommand-Objekt.
BeginExecuteNonQuery-Methode	Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses ULCommand-Objekt beschrieben wird.
BeginExecuteReader-Methode	Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses ULCommand-Objekt beschrieben wird, und ruft die Ergebnismenge ab.
Cancel-Methode	Diese Methode wird in UltraLite.NET nicht unterstützt.

Name	Beschreibung
CreateParameter-Methode	Stellt ein ULParameter-Objekt bereit, um Parameter an ULCommand-Objekte übergeben zu können.
EndExecuteNonQuery-Methode	Beendet die asynchrone Ausführung einer SQL-Anweisung.
EndExecuteReader-Methode	Beendet die asynchrone Ausführung einer SQL-Anweisung und gibt das angeforderte ULDataReader-Objekt zurück.
ExecuteDbDataReader-Methode (geerbt aus System.Data.Common.DbCommand)	Führt den Befehlstext für die Verbindung aus.
ExecuteDbDataReader-Methode (geerbt aus System.Data.Common.DbCommand)	Provider sollten diese Methode implementieren, um eine nicht standardmäßige Implementierung für Overload: System.Data.Common.DbCommand.ExecuteReader-Overloads zu schaffen. Die Standardimplementierung ruft die synchrone System.Data.Common.DbCommand.ExecuteReader -Methode auf und gibt eine abgeschlossene Aufgabe zurück, die den aufrufenden Thread blockiert.
ExecuteNonQuery-Methode	Führt eine Anweisung aus, die keine Ergebnismenge zurückgibt, wie etwa eine SQL INSERT-, DELETE- oder UPDATE-Anweisung.
ExecuteNonQueryAsync-Methode (geerbt aus System.Data.Common.DbCommand)	Eine asynchrone Version von System.Data.Common.DbCommand.ExecuteNonQuery , die eine SQL-Anweisung auf einem Verbindungsobjekt ausführt.
ExecuteReader-Methode	Führt eine SQL SELECT-Anweisung aus und gibt die Ergebnismenge zurück.
ExecuteReaderAsync-Methode (geerbt aus System.Data.Common.DbCommand)	Eine asynchrone Version von Overload: System.Data.Common.DbCommand.ExecuteReader, die System.Data.Common.DbCommand.CommandText mit System.Data.Common.DbCommand.Connection ausführt und System.Data.Common.DbDataReader zurückgibt. Sie ruft System.Data.Common.DbCommand.ExecuteReaderAsync(System.Data.Common.DbCommandBehavior, System.Threading.CancellationToken) mit CancellationTokens.None auf.
ExecuteResultSet-Methode	UL-Erw.: Führt eine SQL SELECT-Anweisung aus und gibt die Ergebnismenge als ULResultSet-Objekt zurück.
ExecuteScalar-Methode	Führt eine SQL SELECT-Anweisung aus und gibt einen einzelnen Wert zurück.

Name	Beschreibung
ExecuteScalarAsync-Methode (geerbt aus <code>System.Data.Common.DbCommand</code>)	Eine asynchrone Version von System.Data.Common.DbCommand.ExecuteScalar . Sie führt eine Abfrage aus und gibt die erste Spalte der ersten Zeile in der Ergebnismenge zurück, die von der Abfrage zurückgegeben wird.
ExecuteTable-Methode	UL-Erw.: Ruft eine Datenbanktabelle in einem <code>ULTable</code> -Objekt zur direkten Bearbeitung ab.
Prepare-Methode	Präkompiliert die SQL-Anweisung dieses Befehls und speichert sie.
CommandText-Eigenschaft	Gibt den Text der SQL-Anweisung oder den Namen der Tabelle an, wenn die <code>ULCommand.CommandType</code> -Eigenschaft den Wert <code>System.Data.CommandType.TableDirect</code> hat.
CommandTimeout-Eigenschaft	Diese Funktion wird von UltraLite.NET nicht unterstützt.
CommandType-Eigenschaft	Gibt den Typ des auszuführenden Befehls an.
Connection-Eigenschaft	Das Verbindungsobjekt, auf dem das <code>ULCommand</code> -Objekt ausgeführt werden soll.
DesignTimeVisible-Eigenschaft	Gibt an, ob das <code>ULCommand</code> -Objekt in einem benutzerdefinierten Steuerelement des Windows Form-Designers sichtbar sein soll.
IndexName-Eigenschaft	UL-Erw.: Gibt den Namen des Indexes an, mit dem die Tabelle geöffnet (sortiert) werden soll, wenn die <code>ULCommand.CommandType</code> -Eigenschaft den Wert <code>System.Data.CommandType.TableDirect</code> hat.
Parameters-Eigenschaft	Legt die Parameter für die aktuelle Anweisung fest.
Plan-Eigenschaft	UL-Erw.: Gibt den Zugriffsplan zurück, den UltraLite.NET zum Ausführen einer Abfrage verwendet.
Transaction-Eigenschaft	Gibt das <code>ULTransaction</code> -Objekt an, in dem das <code>ULCommand</code> -Objekt ausgeführt wird
UpdatedRowSource-Eigenschaft	Legt fest, wie Ergebnisse von Befehlen auf das <code>DataRow</code> -Objekt angewendet werden, wenn es von der <code>ULDataAdapterUpdate</code> -Methode verwendet wird.

Bemerkungen

Dieses Objekt kann dann dazu verwendet werden, eine Anweisung oder Abfrage auf effiziente Weise wiederholt auszuführen.

ULCommand-Objekte können direkt oder mit der ULConnection.CreateCommand-Methode erstellt werden. Diese Methode stellt sicher, dass der Befehl die richtige Transaktion zur Ausführung von Anweisungen über die betreffende Verbindung enthält.

Die ULCommand.Transaction-Methode muss zurückgesetzt werden, nachdem die aktuelle Transaktion festgeschrieben oder zurückgesetzt wurde.

Die ULCommand-Klasse verfügt über die folgenden Methoden zum Ausführen von Befehlen in einer UltraLite.NET-Datenbank:

Methoden	Beschreibung
ULCommand.ExecuteNonQuery	Führt eine Anweisung aus, die keine Ergebnismenge zurückgibt, wie etwa eine SQL INSERT-, DELETE- oder UPDATE-Anweisung.
ULCommand.ExecuteReader()	Führt eine SQL SELECT-Anweisung aus und gibt die Ergebnismenge in einem ULDataReader-Objekt zurück. Verwenden Sie diese Methode zum Erstellen schreibgeschützter Ergebnismengen.
ULCommand.ExecuteResultSet()	UL-Erw.: Führt eine SQL SELECT-Anweisung aus und gibt die Ergebnismenge in einem ULResultSet-Objekt zurück. Verwenden Sie diese Methode zum Erstellen veränderlicher Ergebnismengen.
ULCommand.ExecuteScalar	Führt eine SQL SELECT-Anweisung aus und gibt einen einzelnen Wert zurück.
ULCommand.ExecuteTable()	UL-Erw.: Ruft eine Datenbanktabelle in einem ULTable-Objekt zur direkten Bearbeitung ab. Die ULCommand.CommandText-Eigenschaft wird als Name der Tabelle interpretiert und die ULCommand.IndexName-Eigenschaft kann verwendet werden, um eine Sortierreihenfolge für die Tabelle festzulegen. Die ULCommand.CommandType-Eigenschaft muss den Wert System.Data.CommandType.TableDirect haben.

Sie können die meisten Eigenschaften, einschließlich der ULCommand.CommandText-Eigenschaft, zurücksetzen und das ULCommand-Objekt erneut verwenden.

Aus Gründen der Ressourcenverwaltung wird empfohlen, Anweisungen explizit zu schließen, wenn sie abgeschlossen wurden. In C# können Sie die System.ComponentModel.Component.Dispose-Methode mit einer using-Anweisung automatisch aufrufen oder die System.ComponentModel.Component.Dispose-Methode explizit aufrufen. In Visual Basic müssen Sie die System.ComponentModel.Component.Dispose-Methode explizit aufrufen.

Siehe auch

- [ULConnection.CreateCommand-Methode \[UltraLite.NET\] auf Seite 136](#)
- [ULCommand.Transaction-Eigenschaft \[UltraLite.NET\] auf Seite 108](#)
- [ULCommand.ExecuteNonQuery-Methode \[UltraLite.NET\] auf Seite 93](#)
- [ULCommand.ExecuteReader-Methode \[UltraLite.NET\] auf Seite 94](#)
- [ULCommand.ExecuteResultSet-Methode \[UltraLite.NET\] auf Seite 97](#)
- [ULCommand.ExecuteScalar-Methode \[UltraLite.NET\] auf Seite 99](#)
- [ULCommand.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 100](#)
- [ULTable-Klasse \[UltraLite.NET\] auf Seite 414](#)
- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand.CommandType-Eigenschaft \[UltraLite.NET\] auf Seite 105](#)
- [ULCommand.IndexName-Eigenschaft \[UltraLite.NET\] auf Seite 107](#)
- [ULResultSet-Klasse \[UltraLite.NET\] auf Seite 351](#)
- [ULDataReader-Klasse \[UltraLite.NET\] auf Seite 237](#)
- [System.Data.Common.DbCommand](#)
- [System.Data.IDbCommand](#)
- [System.IDisposable](#)
- [System.Data.CommandType](#)
- [System.ComponentModel.Component.Dispose](#)

ULCommand-Konstruktor

Initialisiert ein ULCommand-Objekt.

Überladungsliste

Name	Beschreibung
ULCommand()-Konstruktor	Initialisiert ein ULCommand-Objekt.
ULCommand(string)-Konstruktor	Initialisiert ein ULCommand-Objekt mit dem angegebenen Befehlstext.
ULCommand(string, ULConnection)-Konstruktor	Initialisiert ein ULCommand-Objekt mit dem angegebenen Befehlstext und der angegebenen Verbindung.
ULCommand(string, ULConnection, ULTransaction)-Konstruktor	Initialisiert ein ULCommand-Objekt mit dem angegebenen Befehlstext sowie der angegebenen Verbindung und Transaktion.

ULCommand()-Konstruktor

Initialisiert ein ULCommand-Objekt.

Visual Basic-Syntax

```
Public Sub New()
```

C#-Syntax

```
public ULCommand()
```

Bemerkungen

Für das ULCommand-Objekt müssen die Eigenschaften ULCommand.CommandText, ULCommand.Connection und ULCommand.Transaction gesetzt werden, damit eine Anweisung ausgeführt werden kann.

Siehe auch

- [ULConnection.CreateCommand-Methode \[UltraLite.NET\] auf Seite 136](#)
- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand.Connection-Eigenschaft \[UltraLite.NET\] auf Seite 106](#)
- [ULCommand.Transaction-Eigenschaft \[UltraLite.NET\] auf Seite 108](#)

ULCommand(string)-Konstruktor

Initialisiert ein ULCommand-Objekt mit dem angegebenen Befehlstext.

Visual Basic-Syntax

```
Public Sub New(ByVal cmdText As String)
```

C#-Syntax

```
public ULCommand(string cmdText)
```

Parameter

- **cmdText** Gibt den Text der SQL-Anweisung oder den Namen der Tabelle an, wenn die ULCommand.CommandType-Eigenschaft den Wert System.Data.CommandType.TableDirect hat. Bei Anweisungen mit Parametern benutzen Sie als Platzhalter ein Fragezeichen (?), um Parameter zu übergeben.

Bemerkungen

Für das ULCommand-Objekt müssen die Eigenschaften ULCommand.Connection und ULCommand.Transaction gesetzt werden, damit eine Anweisung ausgeführt werden kann.

Siehe auch

- [ULConnection.CreateCommand-Methode \[UltraLite.NET\] auf Seite 136](#)
- [ULCommand.ULCommand-Konstruktor \[UltraLite.NET\] auf Seite 76](#)
- [ULCommand.Connection-Eigenschaft \[UltraLite.NET\] auf Seite 106](#)
- [ULCommand.Transaction-Eigenschaft \[UltraLite.NET\] auf Seite 108](#)
- [ULCommand.CommandType-Eigenschaft \[UltraLite.NET\] auf Seite 105](#)
- [System.Data.CommandType](#)

ULCommand(string, ULConnection)-Konstruktor

Initialisiert ein ULCommand-Objekt mit dem angegebenen Befehlstext und der angegebenen Verbindung.

Visual Basic-Syntax

```
Public Sub New(  
    ByVal cmdText As String,  
    ByVal connection As ULConnection  
)
```

C#-Syntax

```
public ULCommand(string cmdText, ULConnection connection)
```

Parameter

- **cmdText** Gibt den Text der SQL-Anweisung oder den Namen der Tabelle an, wenn die ULCommand.CommandType-Eigenschaft den Wert System.Data.CommandType.TableDirect hat. Bei Anweisungen mit Parametern benutzen Sie als Platzhalter ein Fragezeichen (?), um Parameter zu übergeben.
- **connection** Das ULConnection-Objekt, das die aktuelle Verbindung repräsentiert.

Bemerkungen

Für das ULCommand-Objekt muss möglicherweise die ULCommand.Transaction-Eigenschaft gesetzt werden, damit eine Anweisung ausgeführt werden kann.

Siehe auch

- [ULConnection-Klasse \[UltraLite.NET\] auf Seite 121](#)
- [ULConnection.CreateCommand-Methode \[UltraLite.NET\] auf Seite 136](#)
- [ULCommand.ULCommand-Konstruktor \[UltraLite.NET\] auf Seite 76](#)
- [ULCommand.Transaction-Eigenschaft \[UltraLite.NET\] auf Seite 108](#)
- [ULCommand.CommandType-Eigenschaft \[UltraLite.NET\] auf Seite 105](#)
- [System.Data.CommandType](#)

ULCommand(string, ULConnection, ULTransaction)-Konstruktor

Initialisiert ein ULCommand-Objekt mit dem angegebenen Befehlstext sowie der angegebenen Verbindung und Transaktion.

Visual Basic-Syntax

```
Public Sub New(  
    ByVal cmdText As String,  
    ByVal connection As ULConnection,  
    ByVal transaction As ULTransaction  
)
```

C#-Syntax

```
public ULCommand(  
    string cmdText,  
    ULConnection connection,  
    ULTransaction transaction  
)
```

Parameter

- **cmdText** Gibt den Text der SQL-Anweisung oder den Namen der Tabelle an, wenn die ULCommand.CommandType-Eigenschaft den Wert System.Data.CommandType.TableDirect hat. Bei Anweisungen mit Parametern benutzen Sie als Platzhalter ein Fragezeichen (?), um Parameter zu übergeben.
- **connection** Das ULConnection-Objekt, das die aktuelle Verbindung repräsentiert.
- **transaction** Das ULTransaction-Objekt, in dem das ULCommand-Objekt ausgeführt wird.

Siehe auch

- [ULConnection.CreateCommand-Methode \[UltraLite.NET\] auf Seite 136](#)
- [ULCommand.ULCommand-Konstruktor \[UltraLite.NET\] auf Seite 76](#)
- [ULCommand.CommandType-Eigenschaft \[UltraLite.NET\] auf Seite 105](#)
- [ULTransaction-Klasse \[UltraLite.NET\] auf Seite 450](#)
- [System.Data.CommandType](#)

BeginExecuteNonQuery-Methode

Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses ULCommand-Objekt beschrieben wird.

Überladungsliste

Name	Beschreibung
BeginExecuteNonQuery()-Methode	Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses ULCommand-Objekt beschrieben wird.
BeginExecuteNonQuery(AsyncCallback, Object)-Methode	Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses ULCommand-Objekt beschrieben wird, sofern eine Callback-Prozedur und Statusinformationen vorhanden sind.

BeginExecuteNonQuery()-Methode

Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses ULCommand-Objekt beschrieben wird.

Visual Basic-Syntax

```
Public Function BeginExecuteNonQuery() As IAsyncResult
```

C#-Syntax

```
public IAsyncResult BeginExecuteNonQuery()
```

Rückgabe

Es wird ein System.IAsyncResult-Wert zurückgegeben, der für Abrufe, zum Warten auf Ergebnisse oder für beides verwendet werden kann. Dieser Wert ist auch beim Aufrufen der EndExecuteNonQuery(IAsyncResult)-Methode erforderlich, die die Anzahl der betroffenen Zeilen zurückgibt.

Ausnahmebedingungen

- **ULException-Klasse** Jeder Fehler, der während der Ausführung des Befehlstexts aufgetreten ist.

Siehe auch

- [ULCommand.EndExecuteNonQuery-Methode \[UltraLite.NET\] auf Seite 86](#)
- [System.IAsyncResult](#)

BeginExecuteNonQuery(AsyncCallback, Object)-Methode

Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses ULCommand-Objekt beschrieben wird, sofern eine Callback-Prozedur und Statusinformationen vorhanden sind.

Visual Basic-Syntax

```
Public Function BeginExecuteNonQuery(  
    ByVal callback As AsyncCallback,  
    ByVal stateObject As Object  
) As IAsyncResult
```

C#-Syntax

```
public IAsyncResult BeginExecuteNonQuery(  
    AsyncCallback callback,  
    object stateObject  
)
```

Parameter

- **callback** Ein System.AsyncCallback-Delegat, der aufgerufen wird, wenn die Ausführung des Befehls abgeschlossen ist. Mit der Übergabe von Null (in Microsoft Visual Basic "Nothing") legen Sie fest, dass keine Callback-Prozedur erforderlich ist.
- **stateObject** Ein benutzerdefiniertes Statusobjekt, das an die Callback-Prozedur übergeben wird. Sie rufen dieses Objekt innerhalb der Callback-Prozedur unter Verwendung der System.IAsyncResult.AsyncState-Eigenschaft ab.

Rückgabe

Es wird ein System.IAsyncResult-Wert zurückgegeben, der für Abrufe, zum Warten auf Ergebnisse oder für beides verwendet werden kann. Dieser Wert ist auch beim Aufrufen der EndExecuteNonQuery(IAsyncResult)-Methode erforderlich, die die Anzahl der betroffenen Zeilen zurückgibt.

Ausnahmebedingungen

- **ULException-Klasse** Jeder Fehler, der während der Ausführung des Befehlstexts aufgetreten ist.

Siehe auch

- [ULCommand.EndExecuteNonQuery-Methode \[UltraLite.NET\] auf Seite 86](#)
- [System.IAsyncResult.AsyncState](#)
- [System.AsyncCallback](#)
- [System.IAsyncResult](#)

BeginExecuteReader-Methode

Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses ULCommand-Objekt beschrieben wird, und ruft die Ergebnismenge ab.

Überladungsliste

Name	Beschreibung
BeginExecuteReader()-Methode	Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses ULCommand-Objekt beschrieben wird, und ruft die Ergebnismenge ab.
BeginExecuteReader(AsyncCallback, Object)-Methode	Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses ULCommand-Objekt beschrieben wird, und ruft die Ergebnismenge ab, sofern eine Callback-Prozedur und Statusinformationen vorhanden sind.
BeginExecuteReader(AsyncCallback, Object, CommandBehavior)-Methode	Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses ULCommand-Objekt beschrieben wird, unter Verwendung eines der CommandBehavior-Werte und ruft die Ergebnismenge ab, sofern eine Callback-Prozedur und Statusinformationen vorhanden sind.
BeginExecuteReader(CommandBehavior)-Methode	Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses ULCommand-Objekt beschrieben wird, unter Verwendung eines der CommandBehavior-Werte und ruft die Ergebnismenge ab.

BeginExecuteReader()-Methode

Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses ULCommand-Objekt beschrieben wird, und ruft die Ergebnismenge ab.

Visual Basic-Syntax

```
Public Function BeginExecuteReader() As IAsyncResult
```

C#-Syntax

```
public IAsyncResult BeginExecuteReader()
```

Rückgabe

Es wird ein System.IAsyncResult-Wert zurückgegeben, der für Abrufe, zum Warten auf Ergebnisse oder für beides verwendet werden kann. Dieser Wert ist auch beim Aufrufen der EndExecuteReader(IAsyncResult)-Methode erforderlich, die ein ULDataReader-Objekt zurückgibt, das zum Abrufen der zurückgegebenen Zeilen verwendet werden kann.

Ausnahmebedingungen

- **ULException-Klasse** Jeder Fehler, der während der Ausführung des Befehlstexts aufgetreten ist.

Siehe auch

- [ULCommand.EndExecuteReader-Methode \[UltraLite.NET\] auf Seite 89](#)
- [ULDataReader-Klasse \[UltraLite.NET\] auf Seite 237](#)
- [System.IAsyncResult](#)

BeginExecuteReader(AsyncCallback, Object)-Methode

Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses ULCommand-Objekt beschrieben wird, und ruft die Ergebnismenge ab, sofern eine Callback-Prozedur und Statusinformationen vorhanden sind.

Visual Basic-Syntax

```
Public Function BeginExecuteReader(  
    ByVal callback As AsyncCallback,  
    ByVal stateObject As Object  
) As IAsyncResult
```

C#-Syntax

```
public IAsyncResult BeginExecuteReader(  
    AsyncCallback callback,  
    object stateObject  
)
```

Parameter

- **callback** Ein System.AsyncCallback-Delegat, der aufgerufen wird, wenn die Ausführung des Befehls abgeschlossen ist. Mit der Übergabe von Null (in Microsoft Visual Basic "Nothing") legen Sie fest, dass keine Callback-Prozedur erforderlich ist.
- **stateObject** Ein benutzerdefiniertes Statusobjekt, das an die Callback-Prozedur übergeben wird. Sie rufen dieses Objekt innerhalb der Callback-Prozedur unter Verwendung der System.IAsyncResult.AsyncState-Eigenschaft ab.

Rückgabe

Es wird ein System.IAsyncResult-Wert zurückgegeben, der für Abrufe, zum Warten auf Ergebnisse oder für beides verwendet werden kann. Dieser Wert ist auch beim Aufrufen der EndExecuteReader(IAsyncResult)-Methode erforderlich, die ein ULDataReader-Objekt zurückgibt, das zum Abrufen der zurückgegebenen Zeilen verwendet werden kann.

Ausnahmebedingungen

- **ULException-Klasse** Jeder Fehler, der während der Ausführung des Befehlstexts aufgetreten ist.

Siehe auch

- [ULDataReader-Klasse \[UltraLite.NET\] auf Seite 237](#)
- [ULCommand.EndExecuteReader-Methode \[UltraLite.NET\] auf Seite 89](#)
- [System.AsyncCallback](#)
- [System.IAsyncResult.AsyncState](#)
- [System.IAsyncResult.AsyncState](#)
- [System.IAsyncResult](#)

BeginExecuteReader(AsyncCallback, Object, CommandBehavior)-Methode

Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses ULCommand-Objekt beschrieben wird, unter Verwendung eines der CommandBehavior-Werte und ruft die Ergebnismenge ab, sofern eine Callback-Prozedur und Statusinformationen vorhanden sind.

Visual Basic-Syntax

```
Public Function BeginExecuteReader(  
    ByVal callback As AsyncCallback,  
    ByVal stateObject As Object,  
    ByVal cmdBehavior As CommandBehavior  
) As IAsyncResult
```

C#-Syntax

```
public IAsyncResult BeginExecuteReader(  
    AsyncCallback callback,  
    object stateObject,  
    CommandBehavior cmdBehavior  
)
```

Parameter

- **callback** Ein System.AsyncCallback-Delegat, der aufgerufen wird, wenn die Ausführung des Befehls abgeschlossen ist. Mit der Übergabe von Null (in Microsoft Visual Basic "Nothing") legen Sie fest, dass keine Callback-Prozedur erforderlich ist.
- **stateObject** Ein benutzerdefiniertes Statusobjekt, das an die Callback-Prozedur übergeben wird. Sie rufen dieses Objekt innerhalb der Callback-Prozedur unter Verwendung der System.IAsyncResult.AsyncState-Eigenschaft ab.
- **cmdBehavior** Eine bitweise Kombination von System.Data.CommandBehavior-Parametern, die das Ergebnis der Abfrage und ihre Auswirkung auf die Verbindung beschreibt. UltraLite.NET respektiert nur die Parameter System.Data.CommandBehavior.Default, System.Data.CommandBehavior.CloseConnection und System.Data.CommandBehavior.SchemaOnly.

Rückgabe

Es wird ein `System.IAsyncResult`-Wert zurückgegeben, der für Abrufe, zum Warten auf Ergebnisse oder für beides verwendet werden kann. Dieser Wert ist auch beim Aufrufen der `EndExecuteReader(IAsyncResult)`-Methode erforderlich, die ein `ULDataReader`-Objekt zurückgibt, das zum Abrufen der zurückgegebenen Zeilen verwendet werden kann.

Ausnahmebedingungen

- **ULException-Klasse** Jeder Fehler, der während der Ausführung des Befehlstexts aufgetreten ist.

Siehe auch

- [ULCommand.EndExecuteReader-Methode \[UltraLite.NET\] auf Seite 89](#)
- [ULDataReader-Klasse \[UltraLite.NET\] auf Seite 237](#)
- [System.AsyncCallback](#)
- [System.IAsyncResult](#)
- [System.IAsyncResult.AsyncState](#)
- [System.Data.CommandBehavior](#)

BeginExecuteReader(CommandBehavior)-Methode

Startet die asynchrone Ausführung einer SQL-Anweisung, die durch dieses `ULCommand`-Objekt beschrieben wird, unter Verwendung eines der `CommandBehavior`-Werte und ruft die Ergebnismenge ab.

Visual Basic-Syntax

```
Public Function BeginExecuteReader(  
    ByVal cmdBehavior As CommandBehavior  
) As IAsyncResult
```

C#-Syntax

```
public IAsyncResult BeginExecuteReader(CommandBehavior cmdBehavior)
```

Parameter

- **cmdBehavior** Eine bitweise Kombination von `System.Data.CommandBehavior`-Parametern, die das Ergebnis der Abfrage und ihre Auswirkung auf die Verbindung beschreibt. `UltraLite.NET` respektiert nur die Parameter `System.Data.CommandBehavior.Default`, `System.Data.CommandBehavior.CloseConnection` und `System.Data.CommandBehavior.SchemaOnly`.

Rückgabe

Es wird ein `System.IAsyncResult`-Wert zurückgegeben, der für Abrufe, zum Warten auf Ergebnisse oder für beides verwendet werden kann. Dieser Wert ist auch beim Aufrufen der `EndExecuteReader(IAsyncResult)`-Methode erforderlich, die ein `ULDataReader`-Objekt zurückgibt, das zum Abrufen der zurückgegebenen Zeilen verwendet werden kann.

Ausnahmebedingungen

- **ULException-Klasse** Jeder Fehler, der während der Ausführung des Befehlstexts aufgetreten ist.

Siehe auch

- [ULCommand.EndExecuteReader-Methode \[UltraLite.NET\] auf Seite 89](#)
- [ULDataReader-Klasse \[UltraLite.NET\] auf Seite 237](#)
- [System.Data.CommandBehavior](#)
- [System.IAsyncResult](#)

Cancel-Methode

Diese Methode wird in UltraLite.NET nicht unterstützt.

Visual Basic-Syntax

```
Public Overrides Sub Cancel()
```

C#-Syntax

```
public override void Cancel()
```

Bemerkungen

Diese Methode führt keine Aktion aus. UltraLite.NET-Befehle können während ihrer Ausführung nicht unterbrochen werden.

CreateParameter-Methode

Stellt ein ULParameter-Objekt bereit, um Parameter an ULCommand-Objekte übergeben zu können.

Visual Basic-Syntax

```
Public Shadows Function CreateParameter() As ULParameter
```

C#-Syntax

```
public new ULParameter CreateParameter()
```

Rückgabe

Ein neuer Parameter als ULParameter-Objekt

Bemerkungen

Einige SQL-Anweisungen können Parameter übernehmen, die im Text einer Anweisung durch ein Fragezeichen (?) angegeben werden. Die CreateParameter-Methode stellt ein ULParameter-Objekt bereit. Sie können Eigenschaften für das ULParameter-Objekt festlegen, um den Wert für den Parameter anzugeben.

Dies ist die stark typisierte Version von System.Data.IDbCommand.CreateParameter und System.Data.Common.DbCommand.CreateParameter.

Siehe auch

- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)
- [System.Data.IDbCommand.CreateCommand](#)
- [System.Data.Common.DbCommand.CreateCommand](#)

EndExecuteNonQuery-Methode

Beendet die asynchrone Ausführung einer SQL-Anweisung.

Visual Basic-Syntax

```
Public Function EndExecuteNonQuery(  
    ByVal asyncResult As IAsyncResult  
) As Integer
```

C#-Syntax

```
public int EndExecuteNonQuery(IAsyncResult asyncResult)
```

Parameter

- **asyncResult** Das System.IAsyncResult-Objekt, das vom Aufruf der BeginExecuteNonQuery-Methode zurückgegeben wird.

Rückgabe

Die Anzahl der betroffenen Zeilen (d.h. dasselbe Verhalten wie bei der ExecuteNonQuery-Methode).

Ausnahmebedingungen

- **ArgumentException** Der asyncResult-Parameter ist Null ("Nothing" in Microsoft Visual Basic).
- **InvalidOperationException** Die EndExecuteNonQuery(IAsyncResult)-Methode wurde für eine einzelne Befehlsausführung mehrmals aufgerufen oder die Methode stimmte nicht mit der Ausführungsmethode überein.

Bemerkungen

Sie müssen die EndExecuteNonQuery-Methode für jeden BeginExecuteNonQuery-Aufruf einmal aufrufen. Der Aufruf muss erfolgen, nachdem das Ergebnis des BeginExecuteNonQuery-Aufrufs zurückgegeben wurde. ADO.NET ist nicht threadsicher. Sie müssen sicherstellen, dass das Ergebnis des BeginExecuteNonQuery-Aufrufs zurückgegeben wurde. Der System.IAsyncResult-Wert, der an die EndExecuteNonQuery-Methode übergeben wird, muss mit demjenigen übereinstimmen, der vom abgeschlossenen BeginExecuteNonQuery-Aufruf zurückgegeben wurde. Es ist ein Fehler, wenn die EndExecuteNonQuery-Methode aufgerufen wird, um einen Aufruf der BeginExecuteReader-Methode zu beenden, und umgekehrt.

Wenn während der Ausführung des Befehls ein Fehler auftritt, wird beim Aufrufen der EndExecuteNonQuery-Methode eine Ausnahmebedingung ausgegeben.

Es gibt vier Möglichkeiten, auf den Abschluss der Ausführung zu warten:

EndExecuteNonQuery aufrufen Der Aufruf von EndExecuteNonQuery wird blockiert, bis der Befehl ausgeführt wurde. Beispiel:

```
' Visual Basic
Dim cmd As ULCommand = new ULCommand( _
    "UPDATE Departments" _
    + " SET DepartmentName = 'Engineering'" _
    + " WHERE DepartmentID=100", _
    conn _
)

Dim res As IAsyncResult res = _
    cmd.BeginExecuteNonQuery()

' Perform other work.

' This blocks until the command completes.
Dim rowCount As Integer = _
    cmd.EndExecuteNonQuery( res )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULCommand cmd = new ULCommand(
    "UPDATE Departments"
    + " SET DepartmentName = 'Engineering'"
    + " WHERE DepartmentID=100",
    conn
);

IAsyncResult res = cmd.BeginExecuteNonQuery();

// Perform other work.

// This blocks until the command completes.
int rowCount = cmd.EndExecuteNonQuery( res );
```

Eigenschaft IsCompleted von IAsyncResult abrufen Sie können die Eigenschaft IsCompleted von IAsyncResult abrufen. Beispiel:

```
' Visual Basic
Dim cmd As ULCommand = new ULCommand( _
    "UPDATE Departments" _
    + " SET DepartmentName = 'Engineering'" _
    + " WHERE DepartmentID=100", _
    conn _
)

Dim res As IAsyncResult res = _
    cmd.BeginExecuteNonQuery()
While( !res.IsCompleted )
    ' Perform other work.
End While

' This blocks until the command completes.
Dim rowCount As Integer = _
    cmd.EndExecuteNonQuery( res )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULCommand cmd = new ULCommand(
```

```
        "UPDATE Departments"
        + " SET DepartmentName = 'Engineering'"
        + " WHERE DepartmentID=100",
        conn
    );

    IAsyncResult res = cmd.BeginExecuteNonQuery();
    while( !res.IsCompleted ) {
        // Perform other work.
    }

    // This blocks until the command completes.
    int rowCount = cmd.EndExecuteNonQuery( res );
```

Eigenschaft `IAsyncResult.AsyncWaitHandle` verwenden, um ein Synchronisationsobjekt abzurufen
Sie können die Eigenschaft `IAsyncResult.AsyncWaitHandle` verwenden, um ein Synchronisationsobjekt abzurufen und darauf zu warten. Beispiel:

```
' Visual Basic
Dim cmd As ULCommand = new ULCommand( _
    "UPDATE Departments" _
    + " SET DepartmentName = 'Engineering'" _
    + " WHERE DepartmentID=100", _
    conn _
)
Dim res As IAsyncResult res = _
    cmd.BeginExecuteNonQuery()

' Perform other work.

Dim wh As WaitHandle = res.AsyncWaitHandle
wh.WaitOne()
' This does not block because the command is finished.
Dim rowCount As Integer = _
    cmd.EndExecuteNonQuery( res )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULCommand cmd = new ULCommand(
    "UPDATE Departments"
    + " SET DepartmentName = 'Engineering'"
    + " WHERE DepartmentID=100",
    conn
);
IAsyncResult res = cmd.BeginExecuteNonQuery();
// perform other work
WaitHandle wh = res.AsyncWaitHandle;
wh.WaitOne();
// This does not block because the command is finished.
int rowCount = cmd.EndExecuteNonQuery( res );
```

Callback-Funktion beim Aufrufen der `BeginExecuteNonQuery`-Methode angeben Sie können beim Aufrufen der `BeginExecuteNonQuery`-Methode eine Callback-Funktion angeben. Beispiel:

```
' Visual Basic
Private Sub callbackFunction(ByVal ar As IAsyncResult)
    Dim cmd As ULCommand = _
        CType(ar.AsyncState, ULCommand)
    ' This won't block since the command has completed.
    Dim rowCount As Integer = _
        cmd.EndExecuteNonQuery( res )
```

```

End Sub

' Elsewhere in the code
Private Sub DoStuff()
    Dim cmd As ULCommand = new ULCommand( _
        "UPDATE Departments" _
        + " SET DepartmentName = 'Engineering'" _
        + " WHERE DepartmentID=100", _
        conn _
    )
    Dim res As IAsyncResult = _
        cmd.BeginExecuteNonQuery( _
            callbackFunction, cmd _
        )
    ' Perform other work. The callback function
    ' is called when the command completes.
End Sub

```

Dies entspricht dem folgenden Code in der Sprache C#:

```

// C#
private void callbackFunction( IAsyncResult ar )
{
    ULCommand cmd = (ULCommand) ar.AsyncState;
    // This won't block since the command has completed.
    int rowCount = cmd.EndExecuteNonQuery();
}

// Elsewhere in the code
private void DoStuff()
{
    ULCommand cmd = new ULCommand(
        "UPDATE Departments"
        + " SET DepartmentName = 'Engineering'"
        + " WHERE DepartmentID=100",
        conn
    );
    IAsyncResult res = cmd.BeginExecuteNonQuery(
        callbackFunction, cmd
    );
    // Perform other work. The callback function
    // is called when the command completes.
}

```

Die Callback-Funktion wird in einem separaten Thread ausgeführt, sodass die üblichen Warnungen bezüglich der Aktualisierung der Benutzeroberfläche in einem Programm mit mehreren Threads gelten.

Siehe auch

- [ULCommand.BeginExecuteNonQuery-Methode \[UltraLite.NET\] auf Seite 79](#)
- [System.IAsyncResult](#)

EndExecuteReader-Methode

Beendet die asynchrone Ausführung einer SQL-Anweisung und gibt das angeforderte ULDataReader-Objekt zurück.

Visual Basic-Syntax

```
Public Function EndExecuteReader(  
    ByVal asyncResult As IAsyncResult  
) As ULDataReader
```

C#-Syntax

```
public ULDataReader EndExecuteReader(IAsyncResult asyncResult)
```

Parameter

- **asyncResult** Das System.IAsyncResult-Objekt, das vom BeginExecuteReader-Aufruf zurückgegeben wurde.

Rückgabe

Ein ULDataReader-Objekt, das zum Abrufen der angeforderten Zeilen verwendet werden kann (d.h. dasselbe Verhalten wie bei der ExecuteReader-Methode).

Ausnahmebedingungen

- **ArgumentException** Der asyncResult-Parameter ist Null ("Nothing" in Microsoft Visual Basic).
- **InvalidOperationException** Die EndExecuteReader-Methode wurde für eine einzelne Befehlsausführung mehrmals aufgerufen oder die Methode stimmte nicht mit der Ausführungsmethode überein.

Bemerkungen

Die EndExecuteReader-Methode muss für jeden Aufruf der BeginExecuteReader-Methode ein Mal aufgerufen werden. Der Aufruf muss erfolgen, nachdem das Ergebnis des BeginExecuteReader-Aufrufs zurückgegeben wurde. ADO.NET ist nicht threadsicher. Sie müssen sicherstellen, dass das Ergebnis der BeginExecuteReader-Methode zurückgegeben wurde. Der System.IAsyncResult-Wert, der an die EndExecuteReader-Methode übergeben wird, muss mit demjenigen übereinstimmen, der vom abgeschlossenen BeginExecuteReader-Aufruf zurückgegeben wurde. Es ist ein Fehler, wenn die EndExecuteReader-Methode aufgerufen wird, um einen BeginExecuteNonQuery-Aufruf zu beenden, und umgekehrt.

Wenn während der Ausführung des Befehls ein Fehler auftritt, wird beim Aufrufen der EndExecuteReader-Methode eine Ausnahmebedingung ausgegeben.

Es gibt vier Möglichkeiten, auf den Abschluss der Ausführung zu warten:

Call the EndExecuteReader method Das Aufrufen der EndExecuteReader-Methode wird blockiert, bis der Befehl ausgeführt wurde. Beispiel:

```
' Visual Basic  
Dim cmd As ULCommand = new ULCommand( _  
    "SELECT * FROM Departments", conn _  
)  
Dim res As IAsyncResult res = _  
    cmd.BeginExecuteReader()  
' Perform other work  
' This blocks until the command completes.  
Dim reader As ULDataReader = _  
    cmd.EndExecuteReader( res )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULCommand cmd = new ULCommand(
    "SELECT * FROM Departments", conn
);
IAsyncResult res = cmd.BeginExecuteReader();

// Perform other work
// This blocks until the command completes
ULDataReader reader = cmd.EndExecuteReader( res );
```

Eigenschaft IsCompleted von IAsyncResult abrufen Sie können die Eigenschaft IsCompleted von IAsyncResult abrufen. Beispiel:

```
' Visual Basic
Dim cmd As ULCommand = new ULCommand( _
    "SELECT * FROM Departments", conn _
)
Dim res As IAsyncResult res = _
    cmd.BeginExecuteReader()
While( !res.IsCompleted )
    ' Perform other work
End While
' This blocks until the command completes.
Dim reader As ULDataReader = _
    cmd.EndExecuteReader( res )

// C#
ULCommand cmd = new ULCommand(
    "SELECT * FROM Departments", conn
);
IAsyncResult res = cmd.BeginExecuteReader();
while( !res.IsCompleted ) {
    // Perform other work.
}
// This blocks until the command completes.
ULDataReader reader = cmd.EndExecuteReader( res );
```

Eigenschaft IAsyncResult.AsyncWaitHandle verwenden, um ein Synchronisationsobjekt abzurufen Sie können die Eigenschaft IAsyncResult.AsyncWaitHandle verwenden, um ein Synchronisationsobjekt abzurufen und darauf zu warten. Beispiel:

```
' Visual Basic
Dim cmd As ULCommand = new ULCommand( _
    "SELECT * FROM Departments", conn _
)
Dim res As IAsyncResult res = _
    cmd.BeginExecuteReader()
' Perform other work.
Dim wh As WaitHandle = res.AsyncWaitHandle
wh.WaitOne()
' This does not block because the command is finished.
Dim reader As ULDataReader = _
    cmd.EndExecuteReader( res )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULCommand cmd = new ULCommand(
```

```
        "SELECT * FROM Departments", conn
    );
    IAsyncResult res = cmd.BeginExecuteReader();
    // Perform other work.
    WaitHandle wh = res.AsyncWaitHandle;
    wh.WaitOne();
    // This does not block because the command is finished.
    ULDataReader reader = cmd.EndExecuteReader( res );
```

Specify a callback function when calling the BeginExecuteReader method Sie können beim Aufrufen der BeginExecuteReader-Methode eine Callback-Funktion angeben. Beispiel:

```
' Visual Basic
Private Sub callbackFunction(ByVal ar As IAsyncResult)
    Dim cmd As ULCommand = _
        CType(ar.AsyncState, ULCommand)
    ' This won't block since the command has completed.
    Dim reader As ULDataReader = cmd.EndExecuteReader()
End Sub

' Elsewhere in the code
Private Sub DoStuff()
    Dim cmd As ULCommand = new ULCommand( _
        "SELECT * FROM Departments", conn _
    )
    Dim res As IAsyncResult = _
        cmd.BeginExecuteReader( _
            callbackFunction, cmd _
        )
    ' Perform other work. The callback function
    ' is called when the command completes.
End Sub
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
private void callbackFunction( IAsyncResult ar )
{
    ULCommand cmd = (ULCommand) ar.AsyncState;
    // This won't block since the command has completed.
    ULDataReader reader = cmd.EndExecuteReader();
}

// Elsewhere in the code.
private void DoStuff()
{
    ULCommand cmd = new ULCommand(
        "SELECT * FROM Departments", conn
    );
    IAsyncResult res = cmd.BeginExecuteReader(callbackFunction, cmd);

    // Perform other work. The callback function
    // is called when the command completes.
}
```

Die Callback-Funktion wird in einem separaten Thread ausgeführt, sodass die üblichen Warnungen bezüglich der Aktualisierung der Benutzeroberfläche in einem Programm mit mehreren Threads gelten.

Siehe auch

- [ULCommand.BeginExecuteReader-Methode \[UltraLite.NET\] auf Seite 81](#)
- [ULDataReader-Klasse \[UltraLite.NET\] auf Seite 237](#)
- [System.IAsyncResult](#)

ExecuteNonQuery-Methode

Führt eine Anweisung aus, die keine Ergebnismenge zurückgibt, wie etwa eine SQL INSERT-, DELETE- oder UPDATE-Anweisung.

Visual Basic-Syntax

```
Public Overrides Function ExecuteNonQuery() As Integer
```

C#-Syntax

```
public override int ExecuteNonQuery()
```

Rückgabe

Die Anzahl der betroffenen Zeilen

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.
- **InvalidOperationException** Der Befehl hat einen ungültigen Status. Entweder fehlt das ULCommand.Connection-Objekt oder ist geschlossen, der ULCommand.Transaction-Wert stimmt nicht mit dem aktuellen Transaktionsstatus der Verbindung überein oder der ULCommand.CommandText-Wert ist ungültig.

Bemerkungen

Die Anweisung ist das aktuelle ULCommand-Objekt, mit dem ULCommand.CommandText-Wert und den erforderlichen ULCommand.Parameters-Werten.

Für UPDATE-, INSERT- und DELETE-Anweisungen ist der Rückgabewert die Anzahl von Zeilen, die vom Befehl betroffen sind. Für alle anderen Anweisungstypen und für Rollback-Anweisungen ist der Rückgabewert -1.

Die ULCommand.CommandType-Eigenschaft kann nicht den Wert System.Data.CommandType.TableDirect haben.

Siehe auch

- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand.CommandType-Eigenschaft \[UltraLite.NET\] auf Seite 105](#)
- [ULCommand.Connection-Eigenschaft \[UltraLite.NET\] auf Seite 106](#)
- [ULCommand.Parameters-Eigenschaft \[UltraLite.NET\] auf Seite 107](#)
- [ULCommand.Transaction-Eigenschaft \[UltraLite.NET\] auf Seite 108](#)
- [System.Data.CommandType](#)

ExecuteReader-Methode

Führt eine SQL SELECT-Anweisung aus und gibt die Ergebnismenge zurück.

Überladungsliste

Name	Beschreibung
ExecuteReader()-Methode	Führt eine SQL SELECT-Anweisung aus und gibt die Ergebnismenge zurück.
ExecuteReader(CommandBehavior)-Methode	Führt eine SQL SELECT-Anweisung mit dem angegebenen Befehlsverhalten aus und gibt die Ergebnismenge zurück

ExecuteReader()-Methode

Führt eine SQL SELECT-Anweisung aus und gibt die Ergebnismenge zurück.

Visual Basic-Syntax

```
Public Shadows Function ExecuteReader() As ULDataReader
```

C#-Syntax

```
public new ULDataReader ExecuteReader()
```

Rückgabe

Die Ergebnismenge als ULDataReader-Objekt

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.
- **InvalidOperationException** Der Befehl hat einen ungültigen Status. Entweder fehlt der ULCommand.Connection-Wert oder die Verbindung ist geschlossen, der ULCommand.Transaction-Wert stimmt nicht mit dem aktuellen Transaktionsstatus der Verbindung überein oder der ULCommand.CommandText-Wert ist ungültig.

Bemerkungen

Die Anweisung ist das aktuelle ULCommand-Objekt, mit dem ULCommand.CommandText-Wert und den erforderlichen ULCommand.Parameters-Werten. Das ULDataReader-Objekt ist eine schreibgeschützte Ergebnismenge. Verwenden Sie für bearbeitbare Ergebnismengen die ULCommand.ExecuteResultSet-Methode, die ULCommand.ExecuteTable-Methode oder ein ULDataAdapter-Objekt.

Wenn ULCommand.CommandType den Wert System.Data.CommandType.TableDirect hat, führt die ExecuteReader-Methode einen ULCommand.ExecuteTable-Aufruf aus und gibt einen ULTable-Objekt-Downcast als ULDataReader-Objekt zurück.

SELECT-Anweisungen werden aus Performancegründen standardmäßig als schreibgeschützt gekennzeichnet. Falls die Abfrage für Aktualisierungen verwendet werden soll, muss die Anweisung mit "FOR UPDATE" enden.

Dies ist die stark typisierte Version der Methoden `System.Data.IDbCommand.ExecuteReader` und `System.Data.Common.DbCommand.ExecuteReader`.

Siehe auch

- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand.CommandType-Eigenschaft \[UltraLite.NET\] auf Seite 105](#)
- [ULCommand.Connection-Eigenschaft \[UltraLite.NET\] auf Seite 106](#)
- [ULCommand.Parameters-Eigenschaft \[UltraLite.NET\] auf Seite 107](#)
- [ULCommand.ExecuteResultSet-Methode \[UltraLite.NET\] auf Seite 97](#)
- [ULCommand.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 100](#)
- [ULCommand.Transaction-Eigenschaft \[UltraLite.NET\] auf Seite 108](#)
- [ULDataAdapter-Klasse \[UltraLite.NET\] auf Seite 212](#)
- [ULDataReader-Klasse \[UltraLite.NET\] auf Seite 237](#)
- [ULTable-Klasse \[UltraLite.NET\] auf Seite 414](#)
- [System.Data.CommandType](#)
- [System.Data.IDbCommand.ExecuteReader](#)
- [System.Data.Common.DbCommand.ExecuteReader](#)

ExecuteReader(CommandBehavior)-Methode

Führt eine SQL SELECT-Anweisung mit dem angegebenen Befehlsverhalten aus und gibt die Ergebnismenge zurück

Visual Basic-Syntax

```
Public Shadows Function ExecuteReader(  
    ByVal cmdBehavior As CommandBehavior  
) As ULDataReader
```

C#-Syntax

```
public new ULDataReader ExecuteReader(CommandBehavior cmdBehavior)
```

Parameter

- **cmdBehavior** Eine bitweise Kombination von `System.Data.CommandBehavior`-Parametern, die das Ergebnis der Abfrage und ihre Auswirkung auf die Verbindung beschreibt. UltraLite.NET respektiert nur die Parameter `System.Data.CommandBehavior.Default`, `System.Data.CommandBehavior.CloseConnection` und `System.Data.CommandBehavior.SchemaOnly`.

Rückgabe

Die Ergebnismenge als `ULDataReader`-Objekt

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.
- **InvalidOperationException** Der Befehl hat einen ungültigen Status. Entweder fehlt der `ULCommand.Connection`-Wert oder die Verbindung ist geschlossen, der `ULCommand.Transaction`-Wert stimmt nicht mit dem aktuellen Transaktionsstatus der Verbindung überein oder der `ULCommand.CommandText`-Wert ist ungültig.

Bemerkungen

Die Anweisung ist das aktuelle `ULCommand`-Objekt, mit dem `ULCommand.CommandText`-Wert und den erforderlichen `ULCommand.Parameters`-Werten. Das `ULDataReader`-Objekt ist eine schreibgeschützte Ergebnismenge. Verwenden Sie für bearbeitbare Ergebnismengen die `ULCommand.ExecuteResultSet(CommandBehavior)`-Methode, die `ULCommand.ExecuteTable(CommandBehavior)`-Methode oder ein `ULDataAdapter`-Objekt.

Wenn `ULCommand.CommandType` den Wert `System.Data.CommandType.TableDirect` hat, führt die `ExecuteReader`-Methode einen `ULCommand.ExecuteTable(CommandBehavior)`-Aufruf aus und gibt einen `ULTable`-Objekt-Downcast als `ULDataReader`-Objekt zurück.

SELECT-Anweisungen werden aus Performancegründen standardmäßig als schreibgeschützt gekennzeichnet. Falls die Abfrage für Aktualisierungen verwendet werden soll, muss die Anweisung mit "FOR UPDATE" enden.

Dies ist die stark typisierte Version der Methoden `System.Data.IDbCommand.ExecuteReader(System.Data.CommandBehavior)` und `System.Data.Common.DbCommand.ExecuteReader(System.Data.CommandBehavior)`.

Siehe auch

- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand.Connection-Eigenschaft \[UltraLite.NET\] auf Seite 106](#)
- [ULCommand.ExecuteReader-Methode \[UltraLite.NET\] auf Seite 94](#)
- [ULCommand.ExecuteResultSet-Methode \[UltraLite.NET\] auf Seite 97](#)
- [ULCommand.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 100](#)
- [ULCommand.Parameters-Eigenschaft \[UltraLite.NET\] auf Seite 107](#)
- [ULCommand.Transaction-Eigenschaft \[UltraLite.NET\] auf Seite 108](#)
- [ULDataAdapter-Klasse \[UltraLite.NET\] auf Seite 212](#)
- [ULDataReader-Klasse \[UltraLite.NET\] auf Seite 237](#)
- [ULTable-Klasse \[UltraLite.NET\] auf Seite 414](#)
- [ULDataReader-Klasse \[UltraLite.NET\] auf Seite 237](#)
- [System.Data.IDbCommand.ExecuteReader](#)
- [System.Data.Common.DbCommand.ExecuteReader](#)
- [System.Data.CommandType](#)
- [System.Data.CommandBehavior](#)

ExecuteResultSet-Methode

UL-Erw.: Führt eine SQL SELECT-Anweisung aus und gibt die Ergebnismenge als ULResultSet-Objekt zurück.

Überladungsliste

Name	Beschreibung
ExecuteResultSet()-Methode	UL-Erw.: Führt eine SQL SELECT-Anweisung aus und gibt die Ergebnismenge als ULResultSet-Objekt zurück.
ExecuteResultSet(CommandBehavior)-Methode	UL-Erw.: Führt eine SQL SELECT-Anweisung mit dem angegebenen Befehlsverhalten aus und gibt die Ergebnismenge als ULResultSet-Objekt zurück.

ExecuteResultSet()-Methode

UL-Erw.: Führt eine SQL SELECT-Anweisung aus und gibt die Ergebnismenge als ULResultSet-Objekt zurück.

Visual Basic-Syntax

```
Public Function ExecuteResultSet() As ULResultSet
```

C#-Syntax

```
public ULResultSet ExecuteResultSet()
```

Rückgabe

Die Ergebnismenge als ULResultSet-Objekt

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.
- **InvalidOperationException** Der Befehl hat einen ungültigen Status. Entweder fehlt der ULCommand.Connection-Wert oder die Verbindung ist geschlossen, der ULCommand.Transaction-Wert stimmt nicht mit dem aktuellen Transaktionsstatus der Verbindung überein oder der ULCommand.CommandText-Wert ist ungültig.

Bemerkungen

Die Anweisung ist das aktuelle ULCommand-Objekt, mit dem ULCommand.CommandText-Wert und den erforderlichen ULCommand.Parameters-Werten. Das ULResultSet-Objekt ist eine bearbeitbare Ergebnismenge, für die Sie positionsbasierte Aktualisierungen und Löschungen durchführen können. Verwenden Sie für vollständig bearbeitbare Ergebnismengen die ULCommand.ExecuteTable-Methode oder ein ULDataAdapter-Objekt.

Wenn `ULCommand.CommandType` den Wert `System.Data.CommandType.TableDirect` hat, führt die `ExecuteReader`-Methode einen `ULCommand.ExecuteTable`-Aufruf aus und gibt einen `ULTable`-Objekt-Downcast als `ULResultSet`-Objekt zurück.

Diese Methode unterstützt positionierte Aktualisierungen und Löschungen mit Dynamic SQL.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand.Parameters-Eigenschaft \[UltraLite.NET\] auf Seite 107](#)
- [ULCommand.CommandType-Eigenschaft \[UltraLite.NET\] auf Seite 105](#)
- [ULResultSet-Klasse \[UltraLite.NET\] auf Seite 351](#)
- [ULTable-Klasse \[UltraLite.NET\] auf Seite 414](#)
- [System.Data.CommandType](#)

Beispiel

```
cmd.CommandText = "SELECT id, season, price FROM OurProducts";
ULResultSet rs = cmd.ExecuteResultSet();
while( rs.Read() ) {
    string season = rs.GetString( 1 );
    double price = rs.GetDouble( 2 );
    if( season.Equals( "summer" ) ) {
        rs.UpdateBegin();
        rs.SetDouble( 2, price * .5 );
        rs.Update();
    }
    if( season.Equals( "discontinued" ) ) {
        rs.Delete();
    }
}
rs.Close();
```

ExecuteResultSet(CommandBehavior)-Methode

UL-Erw.: Führt eine SQL SELECT-Anweisung mit dem angegebenen Befehlsverhalten aus und gibt die Ergebnismenge als `ULResultSet`-Objekt zurück.

Visual Basic-Syntax

```
Public Function ExecuteResultSet(  
    ByVal cmdBehavior As CommandBehavior  
) As ULResultSet
```

C#-Syntax

```
public ULResultSet ExecuteResultSet(CommandBehavior cmdBehavior)
```

Parameter

- **cmdBehavior** Eine bitweise Kombination von `System.Data.CommandBehavior`-Parametern, die das Ergebnis der Abfrage und ihre Auswirkung auf die Verbindung beschreibt. UltraLite.NET respektiert nur die Parameter `System.Data.CommandBehavior.Default`,

System.Data.CommandBehavior.CloseConnection und
System.Data.CommandBehavior.SchemaOnly.

Rückgabe

Die Ergebnismenge als ULResultSet-Objekt

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.
- **InvalidOperationException** Der Befehl hat einen ungültigen Status. Entweder fehlt der ULCommand.Connection-Wert oder die Verbindung ist geschlossen, der ULCommand.Transaction-Wert stimmt nicht mit dem aktuellen Transaktionsstatus der Verbindung überein oder der ULCommand.CommandText-Wert ist ungültig.

Bemerkungen

Die Anweisung ist das aktuelle ULCommand-Objekt, mit dem ULCommand.CommandText-Wert und den erforderlichen ULCommand.Parameters-Werten. Das ULResultSet-Objekt ist eine bearbeitbare Ergebnismenge, für die Sie positionsbasierte Aktualisierungen und Löschungen durchführen können. Verwenden Sie für vollständig bearbeitbare Ergebnismengen die ULCommand.ExecuteTable(CommandBehavior)-Methode oder ein ULDataAdapter-Objekt.

Wenn ULCommand.CommandType den Wert System.Data.CommandType.TableDirect hat, führt die ExecuteReader-Methode einen ULCommand.ExecuteTable(CommandBehavior)-Aufruf aus und gibt einen ULTable-Objekt-Downcast als ULResultSet-Objekt zurück.

Diese Methode unterstützt positionierte Aktualisierungen und Löschungen mit Dynamic SQL.

Siehe auch

- [ULResultSet-Klasse \[UltraLite.NET\] auf Seite 351](#)
- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand.CommandType-Eigenschaft \[UltraLite.NET\] auf Seite 105](#)
- [ULCommand.Connection-Eigenschaft \[UltraLite.NET\] auf Seite 106](#)
- [ULCommand.ExecuteReader-Methode \[UltraLite.NET\] auf Seite 94](#)
- [ULCommand.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 100](#)
- [ULCommand.Parameters-Eigenschaft \[UltraLite.NET\] auf Seite 107](#)
- [ULCommand.Transaction-Eigenschaft \[UltraLite.NET\] auf Seite 108](#)
- [ULDataAdapter-Klasse \[UltraLite.NET\] auf Seite 212](#)
- [ULTable-Klasse \[UltraLite.NET\] auf Seite 414](#)
- [System.Data.CommandType](#)
- [System.Data.CommandBehavior](#)

ExecuteScalar-Methode

Führt eine SQL SELECT-Anweisung aus und gibt einen einzelnen Wert zurück.

Visual Basic-Syntax

```
Public Overrides Function ExecuteScalar() As Object
```

C#-Syntax

```
public override object ExecuteScalar()
```

Rückgabe

Die erste Spalte der ersten Zeile in der Ergebnismenge oder eine Nullreferenz ("Nothing" in Visual Basic), wenn die Ergebnismenge leer ist

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.
- **InvalidOperationException** Der Befehl hat einen ungültigen Status. Entweder fehlt der ULCommand.Connection-Wert oder die Verbindung ist geschlossen, der ULCommand.Transaction-Wert stimmt nicht mit dem aktuellen Transaktionsstatus der Verbindung überein oder der ULCommand.CommandText-Wert ist ungültig.

Bemerkungen

Die Anweisung ist das aktuelle ULCommand-Objekt, mit dem ULCommand.CommandText-Wert und den erforderlichen ULCommand.Parameters-Werten.

Wenn diese Methode in einer Abfrage aufgerufen wird, die mehrere Zeilen und Spalten zurückgibt, wird nur die erste Spalte der ersten Zeile zurückgegeben.

Wenn ULCommand.CommandType den Wert System.Data.CommandType.TableDirect hat, führt die ExecuteScalar-Methode einen ULCommand.ExecuteTable-Aufruf aus und gibt die erste Spalte der ersten Zeile zurück.

SELECT-Anweisungen werden aus Performancegründen standardmäßig als schreibgeschützt gekennzeichnet. Falls die Abfrage für Aktualisierungen verwendet werden soll, muss die Anweisung mit "FOR UPDATE" enden.

Siehe auch

- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand.CommandType-Eigenschaft \[UltraLite.NET\] auf Seite 105](#)
- [ULCommand.Connection-Eigenschaft \[UltraLite.NET\] auf Seite 106](#)
- [ULCommand.IndexName-Eigenschaft \[UltraLite.NET\] auf Seite 107](#)
- [ULCommand.Parameters-Eigenschaft \[UltraLite.NET\] auf Seite 107](#)
- [ULCommand.Transaction-Eigenschaft \[UltraLite.NET\] auf Seite 108](#)
- [System.Data.CommandType](#)
- [System.Data.CommandBehavior](#)

ExecuteTable-Methode

UL-Erw.: Ruft eine Datenbanktabelle in einem ULTable-Objekt zur direkten Bearbeitung ab.

Überladungsliste

Name	Beschreibung
ExecuteTable()-Methode	UL-Erw.: Ruft eine Datenbanktabelle in einem ULTable-Objekt zur direkten Bearbeitung ab.
ExecuteTable(CommandBehavior)-Methode	UL-Erw.: Ruft mit dem festgelegten Befehlsverhalten eine Datenbanktabelle zur direkten Bearbeitung ab.

ExecuteTable()-Methode

UL-Erw.: Ruft eine Datenbanktabelle in einem ULTable-Objekt zur direkten Bearbeitung ab.

Visual Basic-Syntax

```
Public Function ExecuteTable() As ULTable
```

C#-Syntax

```
public ULTable ExecuteTable()
```

Rückgabe

Die Tabelle als ULTable-Objekt

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.
- **InvalidOperationException** Der Befehl hat einen ungültigen Status. Entweder fehlt der ULCommand.Connection-Wert oder die Verbindung ist geschlossen, der ULCommand.Transaction-Wert stimmt nicht mit dem aktuellen Transaktionsstatus der Verbindung überein oder der ULCommand.CommandText-Wert ist ungültig.

Bemerkungen

Der ULCommand.CommandText-Wert wird als Name der Tabelle interpretiert und der ULCommand.IndexName-Wert kann verwendet werden, um eine Sortierreihenfolge für die Tabelle festzulegen.

ULCommand.CommandType muss auf den Wert System.Data.CommandType.TableDirect gesetzt werden.

Wenn der ULCommand.IndexName-Wert eine Nullreferenz ist ("Nothing" in Visual Basic), wird die Tabelle mithilfe des Primärschlüssels geöffnet. Anderenfalls wird die Tabelle mithilfe des Werts von ULCommand.IndexName als Name des Sortierindexes geöffnet.

Siehe auch

- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand.CommandType-Eigenschaft \[UltraLite.NET\] auf Seite 105](#)
- [ULCommand.Connection-Eigenschaft \[UltraLite.NET\] auf Seite 106](#)
- [ULCommand.IndexName-Eigenschaft \[UltraLite.NET\] auf Seite 107](#)
- [ULCommand.Transaction-Eigenschaft \[UltraLite.NET\] auf Seite 108](#)
- [ULTable-Klasse \[UltraLite.NET\] auf Seite 414](#)
- [System.Data.CommandBehavior](#)
- [System.Data.CommandType](#)

ExecuteTable(CommandBehavior)-Methode

UL-Erw.: Ruft mit dem festgelegten Befehlsverhalten eine Datenbanktabelle zur direkten Bearbeitung ab.

Visual Basic-Syntax

```
Public Function ExecuteTable(  
    ByVal cmdBehavior As CommandBehavior  
) As ULTable
```

C#-Syntax

```
public ULTable ExecuteTable(CommandBehavior cmdBehavior)
```

Parameter

- **cmdBehavior** Eine bitweise Kombination von System.Data.CommandBehavior-Parametern, die das Ergebnis der Abfrage und ihre Auswirkung auf die Verbindung beschreibt. UltraLite.NET respektiert nur die Parameter System.Data.CommandBehavior.Default, System.Data.CommandBehavior.CloseConnection und System.Data.CommandBehavior.SchemaOnly.

Rückgabe

Die Tabelle als ULTable-Objekt

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.
- **InvalidOperationException** Der Befehl hat einen ungültigen Status. Entweder fehlt der ULCommand.Connection-Wert oder die Verbindung ist geschlossen, der ULCommand.Transaction-Wert stimmt nicht mit dem aktuellen Transaktionsstatus der Verbindung überein oder der ULCommand.CommandText-Wert ist ungültig.

Bemerkungen

Der ULCommand.CommandText-Wert wird als Name der Tabelle interpretiert und der ULCommand.IndexName-Wert kann verwendet werden, um eine Sortierreihenfolge für die Tabelle festzulegen.

ULCommand.CommandType muss auf den Wert System.Data.CommandType.TableDirect gesetzt werden.

Wenn der ULCommand.IndexName-Wert eine Nullreferenz ist ("Nothing" in Visual Basic), wird die Tabelle mithilfe des Primärschlüssels geöffnet. Anderenfalls wird die Tabelle mithilfe des Werts von ULCommand.IndexName als Name des Sortierindexes geöffnet.

Siehe auch

- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand.CommandType-Eigenschaft \[UltraLite.NET\] auf Seite 105](#)
- [ULCommand.Connection-Eigenschaft \[UltraLite.NET\] auf Seite 106](#)
- [ULCommand.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 100](#)
- [ULCommand.IndexName-Eigenschaft \[UltraLite.NET\] auf Seite 107](#)
- [ULCommand.Transaction-Eigenschaft \[UltraLite.NET\] auf Seite 108](#)
- [System.Data.CommandBehavior](#)
- [System.Data.CommandType](#)

Prepare-Methode

Präkompiliert die SQL-Anweisung dieses Befehls und speichert sie.

Visual Basic-Syntax

```
Public Overrides Sub Prepare()
```

C#-Syntax

```
public override void Prepare()
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.
- **InvalidOperationException** Der Befehl hat einen ungültigen Status. Entweder fehlt der ULCommand.Connection-Wert oder die Verbindung ist geschlossen, der ULCommand.Transaction-Wert stimmt nicht mit dem aktuellen Transaktionsstatus der Verbindung überein oder der ULCommand.CommandText-Wert ist ungültig.

Bemerkungen

Präkompilier-Anweisungen gestatten eine effiziente wiederholte Verwendung von Anweisungen, wenn sich nur die Parameterwerte geändert haben. Das Ändern irgendeiner anderen Eigenschaft in diesem Befehl hebt die Vorbereitung der Anweisung auf.

UltraLite.NET setzt nicht voraus, dass Anweisungen explizit vorbereitet werden, da alle nicht vorbereiteten Befehle bei Aufrufen der verschiedenen Execute-Methoden vorbereitet werden.

Siehe auch

- [ULCommand.Connection-Eigenschaft \[UltraLite.NET\] auf Seite 106](#)
- [ULCommand.Transaction-Eigenschaft \[UltraLite.NET\] auf Seite 108](#)
- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)

CommandText-Eigenschaft

Gibt den Text der SQL-Anweisung oder den Namen der Tabelle an, wenn die ULCommand.CommandType-Eigenschaft den Wert System.Data.CommandType.TableDirect hat.

Visual Basic-Syntax

```
Public Overrides Property CommandText As String
```

C#-Syntax

```
public override string CommandText {get;set;}
```

Bemerkungen

Bei Anweisungen mit Parametern benutzen Sie als Platzhalter ein Fragezeichen (?), um Parameter zu übergeben.

Eine Zeichenfolge, die den Text der SQL-Anweisung oder den Tabellennamen angibt. Standardwert ist eine leere Zeichenfolge (ungültiger Befehl).

SELECT-Anweisungen werden aus Performancegründen standardmäßig als schreibgeschützt gekennzeichnet. Falls die Abfrage für Aktualisierungen verwendet werden soll, muss die Anweisung mit "FOR UPDATE" enden.

Siehe auch

- [ULCommand.ExecuteNonQuery-Methode \[UltraLite.NET\] auf Seite 93](#)
- [ULCommand.ExecuteReader-Methode \[UltraLite.NET\] auf Seite 94](#)
- [ULCommand.ExecuteResultSet-Methode \[UltraLite.NET\] auf Seite 97](#)
- [ULCommand.ExecuteScalar-Methode \[UltraLite.NET\] auf Seite 99](#)
- [ULCommand.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 100](#)
- [ULCommand.CommandType-Eigenschaft \[UltraLite.NET\] auf Seite 105](#)
- [System.Data.CommandType](#)

Beispiel

```
' Visual Basic myCmd.CommandText = "SELECT * FROM Customers WHERE CustomerID = ?"
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#  
myCmd.CommandText = "SELECT * FROM Customers WHERE CustomerID = ?";
```

CommandTimeout-Eigenschaft

Diese Funktion wird von UltraLite.NET nicht unterstützt.

Visual Basic-Syntax

```
Public Overrides Property CommandTimeout As Integer
```

C#-Syntax

```
public override int CommandTimeout {get;set;}
```

Ausnahmen

- **ULException-Klasse** Die Festlegung des Werts wird in UltraLite.NET nicht unterstützt.

Bemerkungen

Der Wert ist stets Null.

CommandType-Eigenschaft

Gibt den Typ des auszuführenden Befehls an.

Visual Basic-Syntax

```
Public Overrides Property CommandType As CommandType
```

C#-Syntax

```
public override CommandType CommandType {get;set;}
```

Ausnahmen

- **ArgumentException** CommandType.StoredProcedure wird in UltraLite.NET nicht unterstützt.

Bemerkungen

Einer der System.Data.CommandType-Werte. Der Standardwert ist System.Data.CommandType.Text.

Folgende Befehlstypen werden unterstützt:

- System.Data.CommandType.TableDirect - **UL-Erw.:** Wenn Sie diese CommandType-Eigenschaft angeben, muss die ULCommand.CommandText-Eigenschaft der Name einer Datenbanktabelle sein. Sie können auch den Index angeben, der zum Öffnen (Sortieren) der Tabelle mit der ULCommand.IndexName-Eigenschaft verwendet werden soll. Verwenden Sie für den Zugriff auf die Tabelle die Methode ULCommand.ExecuteTable oder ULCommand.ExecuteReader.
- System.Data.CommandType.Text - Wenn Sie diese CommandType-Eigenschaft angeben, muss die ULCommand.CommandText-Eigenschaft eine SQL-Anweisung oder Abfrage sein. Mit der ULCommand.ExecuteNonQuery-Methode können Sie eine SQL-Anweisung ausführen, bei der es sich nicht um eine Abfrage handelt, mit der Methode ULCommand.ExecuteReader oder ULCommand.ExecuteScalar eine Abfrage.

Siehe auch

- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand.IndexName-Eigenschaft \[UltraLite.NET\] auf Seite 107](#)
- [ULCommand.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 100](#)
- [ULCommand.ExecuteReader-Methode \[UltraLite.NET\] auf Seite 94](#)
- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand.ExecuteNonQuery-Methode \[UltraLite.NET\] auf Seite 93](#)
- [ULCommand.ExecuteReader-Methode \[UltraLite.NET\] auf Seite 94](#)
- [ULCommand.ExecuteScalar-Methode \[UltraLite.NET\] auf Seite 99](#)
- [System.Data.CommandType](#)

Connection-Eigenschaft

Das Verbindungsobjekt, auf dem das ULCommand-Objekt ausgeführt werden soll.

Visual Basic-Syntax

```
Public Shadows Property Connection As ULConnection
```

C#-Syntax

```
public new ULConnection Connection {get;set;}
```

Bemerkungen

Das ULConnection-Objekt, für das der Befehl ausgeführt werden soll.

ULCommand-Objekte müssen eine offene Verbindung haben, damit sie ausgeführt werden können.

Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Dies ist die stark typisierte Version von System.Data.IDbCommand.Connection und System.Data.Common.DbCommand.Connection.

Siehe auch

- [ULConnection-Klasse \[UltraLite.NET\] auf Seite 121](#)
- [System.Data.IDbCommand.Connection](#)
- [System.Data.Common.DbCommand.Connection](#)

DesignTimeVisible-Eigenschaft

Gibt an, ob das ULCommand-Objekt in einem benutzerdefinierten Steuerelement des Windows Form-Designers sichtbar sein soll.

Visual Basic-Syntax

```
Public Overrides Property DesignTimeVisible As Boolean
```

C#-Syntax

```
public override bool DesignTimeVisible {get;set;}
```

Bemerkungen

TRUE, wenn dieses ULCommand-Objekt sichtbar sein soll, andernfalls FALSE. Der Standardwert ist FALSE.

IndexName-Eigenschaft

UL-Erw.: Gibt den Namen des Indexes an, mit dem die Tabelle geöffnet (sortiert) werden soll, wenn die ULCommand.CommandType-Eigenschaft den Wert System.Data.CommandType.TableDirect hat.

Visual Basic-Syntax

```
Public Property IndexName As String
```

C#-Syntax

```
public string IndexName {get;set;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen des Indexes angibt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic). Das bedeutet, die Tabelle wird mit ihrem Primärschlüssel geöffnet.

Siehe auch

- [ULCommand.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 100](#)
- [ULCommand.ExecuteReader-Methode \[UltraLite.NET\] auf Seite 94](#)
- [ULCommand.CommandType-Eigenschaft \[UltraLite.NET\] auf Seite 105](#)
- [System.Data.CommandType](#)

Parameters-Eigenschaft

Legt die Parameter für die aktuelle Anweisung fest.

Visual Basic-Syntax

```
Public ReadOnly Shadows Property Parameters As ULParameterCollection
```

C#-Syntax

```
public new ULParameterCollection Parameters {get;}
```

Bemerkungen

Ein ULParameterCollection-Objekt mit den Parametern der SQL-Anweisung. Der Standardwert ist eine leere Sammlung.

Verwenden Sie im Wert der ULCommand.CommandText-Eigenschaft Fragezeichen, um Parameter anzugeben. Die Parameter in der Sammlung werden in derselben Reihenfolge angegeben wie die Fragezeichen-Platzhalter. Der erste Parameter in der Sammlung entspricht dem ersten Fragezeichen in der SQL-Anweisung, der zweite Parameter in der Sammlung dem zweiten Fragezeichen in der SQL-Anweisung, usw. Es müssen mindestens ebenso viele Fragezeichen in der ULCommand.CommandText-Eigenschaft enthalten sein wie Parameter in dieser Sammlung.

Dies ist die stark typisierte Version von `System.Data.IDbCommand.Parameters` und `System.Data.Common.DbCommand.Parameters`.

Siehe auch

- [ULParameterCollection-Klasse \[UltraLite.NET\] auf Seite 333](#)
- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)
- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [System.Data.IDbCommand.Connection](#)
- [System.Data.Common.DbCommand.Connection](#)

Plan-Eigenschaft

UL-Erw.: Gibt den Zugriffsplan zurück, den UltraLite.NET zum Ausführen einer Abfrage verwendet.

Visual Basic-Syntax

```
Public ReadOnly Property Plan As String
```

C#-Syntax

```
public string Plan {get;}
```

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Diese Eigenschaft ist in erster Linie für die Entwicklung gedacht.

Eine Zeichenfolge mit der auf Text basierenden Beschreibung des Abfrageausführungsplans

Transaction-Eigenschaft

Gibt das `ULTransaction`-Objekt an, in dem das `ULCommand`-Objekt ausgeführt wird

Visual Basic-Syntax

```
Public Shadows Property Transaction As ULTransaction
```

C#-Syntax

```
public new ULTransaction Transaction {get;set;}
```

Bemerkungen

Das `ULTransaction`-Objekt, in dem das `ULCommand`-Objekt ausgeführt wird. Dies sollte die aktuelle Transaktion der Verbindung sein, die durch das `ULCommand.Connection`-Objekt angegeben wird. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Wenn ein Befehl erneut verwendet wird, nachdem eine Transaktion festgeschrieben oder zurückgesetzt wurde, muss diese Eigenschaft zurückgesetzt werden.

Dies ist die stark typisierte Version von `System.Data.IDbCommand.Transaction` und `System.Data.Common.DbCommand.Transaction`.

Siehe auch

- [ULConnection.BeginTransaction-Methode \[UltraLite.NET\] auf Seite 129](#)
- [ULTransaction-Klasse \[UltraLite.NET\] auf Seite 450](#)
- [ULCommand.Connection-Eigenschaft \[UltraLite.NET\] auf Seite 106](#)
- [System.Data.IDbCommand.Transaction](#)
- [System.Data.Common.DbCommand.Transaction](#)

UpdatedRowSource-Eigenschaft

Legt fest, wie Ergebnisse von Befehlen auf das `DataRow`-Objekt angewendet werden, wenn es von der `ULDataAdapterUpdate`-Methode verwendet wird.

Visual Basic-Syntax

```
Public Overrides Property UpdatedRowSource As UpdateRowSource
```

C#-Syntax

```
public override UpdateRowSource UpdatedRowSource {get;set;}
```

Bemerkungen

Einer der `System.Data.UpdateRowSource`-Werte. Der Standardwert ist `System.Data.UpdateRowSource.Both`.

Siehe auch

- [System.Data.UpdateRowSource](#)

ULCommandBuilder-Klasse

Generiert automatisch Befehle für einzelne Tabellen, mit denen Änderungen in einem `System.Data.DataSet`-Objekt mit der zugeordneten Datenbank abgeglichen werden

Visual Basic-Syntax

```
Public Class ULCommandBuilder  
    Inherits System.Data.Common.DbCommandBuilder
```

C#-Syntax

```
public class ULCommandBuilder : System.Data.Common.DbCommandBuilder
```

Basisklassen

- [System.Data.Common.DbCommandBuilder](#)

Mitglieder

Alle Mitglieder der ULCommandBuilder-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ULCommandBuilder-Konstruktor	Initialisiert ein ULCommandBuilder-Objekt.
ApplyParameterInfo-Methode (geerbt aus System.Data.Common.DbCommandBuilder)	Ermöglicht es der Provider-Implementierung der Klasse System.Data.Common.DbCommandBuilder , zusätzliche Parametereigenschaften zu verarbeiten.
Dispose-Methode (geerbt aus System.Data.Common.DbCommandBuilder)	Gibt die unverwalteten Ressourcen frei, die von System.Data.Common.DbCommandBuilder verwendet werden, und optional auch die verwalteten Ressourcen.
GetDeleteCommand-Methode	Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Löschvorgängen in der Datenbank erforderlich ist
GetInsertCommand-Methode	Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Einfügungen in der Datenbank erforderlich ist.
GetParameterName-Methode (geerbt aus System.Data.Common.DbCommandBuilder)	Gibt den Namen des angegebenen Parameters im Format @p# zurück.
GetParameterPlaceholder-Methode (geerbt aus System.Data.Common.DbCommandBuilder)	Gibt den Platzhalter für den Parameter in der zugeordneten SQL-Anweisung zurück.
GetSchemaTable-Methode (geerbt aus System.Data.Common.DbCommandBuilder)	Gibt die Schematabelle für System.Data.Common.DbCommandBuilder zurück.
GetUpdateCommand-Methode	Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Aktualisierungen in der Datenbank erforderlich ist.
InitializeCommand-Methode (geerbt aus System.Data.Common.DbCommandBuilder)	Setzt die Eigenschaften System.Data.Common.DbCommand.CommandTimeout , System.Data.Common.DbCommand.Transaction , System.Data.Common.DbCommand.CommandType und System.Data.UpdateRowSource für das System.Data.Common.DbCommand -Objekt zurück.

Name	Beschreibung
QuoteIdentifizier-Methode (Geerbt von System.Data.Common.DbCommandBuilder)	Wenn ein nicht in Anführungszeichen eingebetteter Bezeichner in der korrekten Katalogschreibweise übergeben wird, gibt das Objekt die korrekte, in Anführungszeichen gesetzte Form dieses Bezeichners zurück, wobei eventuelle Anführungszeichen innerhalb des Bezeichners durch Escapezeichen ausgenommen werden.
RefreshSchema-Methode (geerbt aus System.Data.Common.DbCommandBuilder)	Löscht die zugeordneten Befehle für dieses System.Data.Common.DbCommandBuilder -Objekt.
RowUpdatingHandler-Methode (geerbt aus System.Data.Common.DbCommandBuilder)	Fügt einen Event-Handler für das System.Data.OleDb.OleDbDataAdapter.RowUpdating -Ereignis hinzu.
SetRowUpdatingHandler-Methode (geerbt aus System.Data.Common.DbCommandBuilder)	Registriert die System.Data.Common.DbCommandBuilder für die Verarbeitung des System.Data.OleDb.OleDbDataAdapter.RowUpdating -Ereignisses für ein System.Data.Common.DbDataAdapter -Objekt
UnquoteIdentifizier-Methode (Geerbt von System.Data.Common.DbCommandBuilder)	Wenn ein in Anführungszeichen eingebetteter Bezeichner übergeben wird, gibt das Objekt die korrekte, nicht in Anführungszeichen gesetzte Schreibweise dieses Bezeichners zurück, wobei die Escapezeichen von etwaigen, in den Bezeichner eingebetteten Anführungszeichen entfernt werden.
CatalogLocation-Eigenschaft (geerbt aus System.Data.Common.DbCommandBuilder)	Legt das CatalogLocation -Objekt für eine Instanz der Klasse System.Data.Common.DbCommandBuilder fest bzw. ruft es ab.
CatalogSeparator-Eigenschaft (geerbt aus System.Data.Common.DbCommandBuilder)	Legt eine Zeichenfolge fest, die als Katalogtrennzeichen für eine Instanz der Klasse System.Data.Common.DbCommandBuilder verwendet wird, bzw. ruft sie ab.
ConflictOption-Eigenschaft (geerbt aus System.Data.Common.DbCommandBuilder)	Gibt an, welches System.Data.ConflictOption -Objekt von System.Data.Common.DbCommandBuilder verwendet werden soll.
DataAdapter-Eigenschaft	Ruft ein ULDataAdapter -Objekt ab, für das SQL-Anweisungen automatisch generiert werden, oder legt das Objekt fest.
QuotePrefix-Eigenschaft (geerbt aus System.Data.Common.DbCommandBuilder)	Ruft das oder die Anfangszeichen ab, die bei der Angabe von Datenbankobjekten (wie z.B. Tabellen oder Spalten), deren Namen z.B. Leerzeichen oder reservierte Tokens enthalten, verwendet werden sollen, bzw. legt die betreffenden Zeichen fest.

Name	Beschreibung
QuoteSuffix-Eigenschaft (geerbt aus System.Data.Common.DbCommandBuilder)	Ruft das oder die Endezeichen ab, die bei der Angabe von Datenbankobjekten (wie z.B. Tabellen oder Spalten), deren Namen z.B. Leerzeichen oder reservierte Tokens enthalten, verwendet werden sollen, bzw. legt die betreffenden Zeichen fest.
SchemaSeparator-Eigenschaft (geerbt aus System.Data.Common.DbCommandBuilder)	Ruft das Zeichen ab, das als Trennzeichen zwischen dem Schema-bezeichner und anderen Bezeichnern verwendet werden soll, oder legt es fest.
SetAllValues-Eigenschaft (geerbt aus System.Data.Common.DbCommandBuilder)	Gibt an, ob alle Spaltenwerte in einer UPDATE-Anweisung oder nur die geänderten Werte aufgenommen werden.

Bemerkungen

Das ULDataAdapter-Objekt generiert nicht automatisch die SQL-Anweisungen, die erforderlich sind, um Änderungen an einem System.Data.DataSet-Objekt mit der zugeordneten Datenquelle abzugleichen. Sie können jedoch ein ULCommandBuilder-Objekt erstellen, um automatisch SQL-Anweisungen für Aktualisierungen einzelner Tabellen zu generieren, wenn Sie die SelectCommand-Eigenschaft des ULDataAdapter-Objekts festlegen. Danach werden zusätzliche SQL-Anweisungen, die Sie nicht festlegen, vom ULCommandBuilder-Objekt generiert.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULConnection-Klasse \[UltraLite.NET\] auf Seite 121](#)
- [ULDataAdapter-Klasse \[UltraLite.NET\] auf Seite 212](#)
- [System.Data.DataSet](#)
- [System.ComponentModel.Component](#)
- [System.IDisposable](#)

Beispiel

Im folgenden Beispiel wird das ULCommand-Objekt zusammen mit den Objekten ULDataAdapter und ULConnection verwendet, um Zeilen aus einer Datenquelle auszuwählen. Das Beispiel wird einer Verbindungszeichenfolge übergeben, bei der es sich um eine Abfragenzeichenfolge mit einer SQL SELECT-Anweisung und eine Zeichenfolge mit dem Namen der Datenbanktabelle handelt. Anschließend wird im Beispiel ein ULCommandBuilder-Objekt erstellt.

```
' Visual Basic
Public Shared Function SelectULRows(ByVal connectionString As String, _
    ByVal queryString As String, ByVal tableName As String)

    Dim connection As ULConnection = New ULConnection(connectionString)
    Dim adapter As ULDataAdapter = New ULDataAdapter()

    adapter.SelectCommand = New ULCommand(queryString, connection)

    Dim builder As ULCommandBuilder = New ULCommandBuilder(adapter)

    connection.Open()
```

```

Dim dataSet As DataSet = New DataSet()
adapter.Fill(dataSet, tableName)

'Insert code to modify data in DataSet.

'Without the ULCommandBuilder this line would fail
adapter.Update(dataSet, tableName)

Return dataSet
End Function

```

Dies entspricht dem folgenden Code in der Sprache C#:

```

// C#
public static DataSet SelectULRows(string connectionString,
    string queryString, string tableName)
{
    using (ULConnection connection = new ULConnection(connectionString))
    {
        ULDataAdapter adapter = new ULDataAdapter();
        adapter.SelectCommand = new ULCommand(queryString, connection);
        ULCommandBuilder builder = new ULCommandBuilder(adapter);

        connection.Open();

        DataSet dataSet = new DataSet();
        adapter.Fill(dataSet, tableName);

        // Insert code to modify data in DataSet.

        // Without the ULCommandBuilder this line would fail
        adapter.Update(dataSet, tableName);

        return dataSet;
    }
}

```

ULCommandBuilder-Konstruktor

Initialisiert ein ULCommandBuilder-Objekt.

Überladungsliste

Name	Beschreibung
ULCommandBuilder()-Konstruktor	Initialisiert ein ULCommandBuilder-Objekt.
ULCommandBuilder(ULDataAdapter)-Konstruktor	Initialisiert ein ULCommandBuilder-Objekt mit dem angegebenen ULDataAdapter-Objekt.

ULCommandBuilder()-Konstruktor

Initialisiert ein ULCommandBuilder-Objekt.

Visual Basic-Syntax

```
Public Sub New()
```

C#-Syntax

```
public ULCommandBuilder()
```

Siehe auch

- [ULCommandBuilder.ULCommandBuilder-Konstruktor \[UltraLite.NET\] auf Seite 113](#)

ULCommandBuilder(ULDataAdapter)-Konstruktor

Initialisiert ein ULCommandBuilder-Objekt mit dem angegebenen ULDataAdapter-Objekt.

Visual Basic-Syntax

```
Public Sub New(ByVal adapter As ULDataAdapter)
```

C#-Syntax

```
public ULCommandBuilder(ULDataAdapter adapter)
```

Parameter

- **adapter** Ein ULDataAdapter-Objekt

Siehe auch

- [ULDataAdapter-Klasse \[UltraLite.NET\] auf Seite 212](#)

GetDeleteCommand-Methode

Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Löschvorgängen in der Datenbank erforderlich ist

Überladungsliste

Name	Beschreibung
GetDeleteCommand()-Methode	Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Löschvorgängen in der Datenbank erforderlich ist
GetDeleteCommand(bool)-Methode	Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Löschvorgängen in der Datenbank erforderlich ist

GetDeleteCommand()-Methode

Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Löschvorgängen in der Datenbank erforderlich ist

Visual Basic-Syntax

```
Public Shadows Function GetDeleteCommand() As ULCommand
```

C#-Syntax

```
public new ULCommand GetDeleteCommand()
```

Rückgabe

Das automatisch generierte ULCommand-Objekt, das für die Durchführung von Löschvorgängen erforderlich ist

Ausnahmen

- **InvalidOperationException** Die DbCommandBuilder.DataAdapter-Eigenschaft wurde nicht initialisiert. Die DataAdapter.SelectCommand-Eigenschaft wurde nicht initialisiert. Die DataAdapter.SelectCommand.Connection-Eigenschaft wurde nicht initialisiert. Die Dynamic SQL-Generierung für mehrere Basistabellen wird nicht unterstützt. Die Dynamic SQL-Generierung wird für einen SelectCommand-Wert, der doppelte Spalten enthält, nicht unterstützt. Die Dynamic SQL-Generierung für die DeleteCommand-Eigenschaft wird für einen SelectCommand-Wert, der keine Schlüsselspalteninformationen zurückgibt, nicht unterstützt.

Bemerkungen

Nachdem die SQL-Anweisung erstmals generiert wurde, muss die Anwendung explizit die DbCommandBuilder.RefreshSchema-Methode aufrufen, sofern diese den ULDataAdapter.SelectCommand-Wert ändert. Sonst verwendet die GetDeleteCommand-Methode weiterhin Informationen aus der vorhergehenden Anweisung, die nicht unbedingt richtig sein müssen. Die SQL-Anweisungen werden erstmals generiert, wenn die Anwendung entweder die DbDataAdapter.Update(System.Data.DataSet)-Methode oder die GetDeleteCommand-Methode aufruft.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULDataAdapter.SelectCommand-Eigenschaft \[UltraLite.NET\] auf Seite 219](#)
- [System.Data.Common.DbCommandBuilder.DataAdapter](#)
- [System.Data.Common.DbCommandBuilder.RefreshSchema](#)

GetDeleteCommand(bool)-Methode

Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Löschvorgängen in der Datenbank erforderlich ist

Visual Basic-Syntax

```
Public Shadows Function GetDeleteCommand(  
    ByVal useColumnsForParameterNames As Boolean  
) As ULCommand
```

C#-Syntax

```
public new ULCommand GetDeleteCommand(bool useColumnsForParameterNames)
```

Parameter

- **useColumnsForParameterNames** Wenn der Wert TRUE ergibt, werden Parameternamen generiert, die mit dem Spaltennamen übereinstimmen, sofern möglich. Wenn der Wert FALSE ergibt, werden @p1, @p2 usw. generiert.

Rückgabe

Das automatisch generierte ULCommand-Objekt, das für die Durchführung von Löschvorgängen erforderlich ist

Ausnahmen

- **InvalidOperationException** Die DbCommandBuilder.DataAdapter-Eigenschaft wurde nicht initialisiert. Die DataAdapter.SelectCommand-Eigenschaft wurde nicht initialisiert. Die DataAdapter.SelectCommand.Connection-Eigenschaft wurde nicht initialisiert. Die Dynamic SQL-Generierung für mehrere Basistabellen wird nicht unterstützt. Die Dynamic SQL-Generierung wird für einen SelectCommand-Wert, der doppelte Spalten enthält, nicht unterstützt. Die Dynamic SQL-Generierung für die DeleteCommand-Eigenschaft wird für einen SelectCommand-Wert, der keine Schlüsselspalteninformationen zurückgibt, nicht unterstützt.

Bemerkungen

Nachdem die SQL-Anweisung erstmals generiert wurde, muss die Anwendung explizit die DbCommandBuilder.RefreshSchema-Methode aufrufen, sofern diese den ULDataAdapter.SelectCommand-Wert ändert. Sonst verwendet die GetDeleteCommand-Methode weiterhin Informationen aus der vorhergehenden Anweisung, die nicht unbedingt richtig sein müssen. Die SQL-Anweisungen werden erstmals generiert, wenn die Anwendung entweder die DbDataAdapter.Update(System.Data.DataSet)-Methode oder die GetDeleteCommand-Methode aufruft.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULDataAdapter.SelectCommand-Eigenschaft \[UltraLite.NET\] auf Seite 219](#)
- [System.Data.Common.DbCommandBuilder.DataAdapter](#)
- [System.Data.Common.DbCommandBuilder.RefreshSchema](#)

GetInsertCommand-Methode

Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Einfügungen in der Datenbank erforderlich ist.

Überladungsliste

Name	Beschreibung
GetInsertCommand()-Methode	Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Einfügungen in der Datenbank erforderlich ist.
GetInsertCommand(bool)-Methode	Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Einfügungen in der Datenbank erforderlich ist.

GetInsertCommand()-Methode

Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Einfügungen in der Datenbank erforderlich ist.

Visual Basic-Syntax

```
Public Shadows Function GetInsertCommand() As ULCommand
```

C#-Syntax

```
public new ULCommand GetInsertCommand()
```

Rückgabe

Das automatisch generierte ULCommand-Objekt, das für die Durchführung von Einfügungen erforderlich ist.

Ausnahmen

- **InvalidOperationException** Die DbCommandBuilder.DataAdapter-Eigenschaft wurde nicht initialisiert. Die DataAdapter.SelectCommand-Eigenschaft wurde nicht initialisiert. Die DataAdapter.SelectCommand.Connection-Eigenschaft wurde nicht initialisiert. Die Dynamic SQL-Generierung für die InsertCommand-Eigenschaft wird für einen SelectCommand-Wert, der keine änderbaren Spalten zurückgibt, nicht unterstützt. Die Dynamic SQL-Generierung für mehrere Basistabellen wird nicht unterstützt. Die Dynamic SQL-Generierung wird für einen SelectCommand-Wert, der doppelte Spalten enthält, nicht unterstützt.

Bemerkungen

Nachdem die SQL-Anweisung erstmals generiert wurde, muss die Anwendung explizit die DbCommandBuilder.RefreshSchema-Methode aufrufen, sofern diese den ULDataAdapter.SelectCommand-Wert ändert. Sonst verwendet die GetInsertCommand-Methode weiterhin Informationen aus der vorhergehenden Anweisung, die nicht unbedingt richtig sein müssen. Die SQL-Anweisungen werden erstmals generiert, wenn die Anwendung entweder die DbDataAdapter.Update(System.Data.DataSet)-Methode oder die GetInsertCommand-Methode aufruft.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULDataAdapter.SelectCommand-Eigenschaft \[UltraLite.NET\] auf Seite 219](#)
- [System.Data.Common.DbCommandBuilder.RefreshSchema](#)
- [System.Data.Common.DbCommandBuilder.DataAdapter](#)

GetInsertCommand(bool)-Methode

Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Einfügungen in der Datenbank erforderlich ist.

Visual Basic-Syntax

```
Public Shadows Function GetInsertCommand(  
    ByVal useColumnsForParameterNames As Boolean  
) As ULCommand
```

C#-Syntax

```
public new ULCommand GetInsertCommand(bool useColumnsForParameterNames)
```

Parameter

- **useColumnsForParameterNames** Wenn der Wert TRUE ergibt, werden Parameternamen generiert, die mit dem Spaltennamen übereinstimmen, sofern möglich. Wenn der Wert FALSE ergibt, werden @p1, @p2 usw. generiert.

Rückgabe

Das automatisch generierte ULCommand-Objekt, das für die Durchführung von Einfügungen erforderlich ist.

Ausnahmen

- **InvalidOperationException** Die DbCommandBuilder.DataAdapter-Eigenschaft wurde nicht initialisiert. Die DataAdapter.SelectCommand-Eigenschaft wurde nicht initialisiert. Die DataAdapter.SelectCommand.Connection-Eigenschaft wurde nicht initialisiert. Die Dynamic SQL-Generierung für die InsertCommand-Eigenschaft wird für einen SelectCommand-Wert, der keine änderbaren Spalten zurückgibt, nicht unterstützt. Die Dynamic SQL-Generierung für mehrere Basistabellen wird nicht unterstützt. Die Dynamic SQL-Generierung wird für einen SelectCommand-Wert, der doppelte Spalten enthält, nicht unterstützt.

Bemerkungen

Nachdem die SQL-Anweisung erstmals generiert wurde, muss die Anwendung explizit die DbCommandBuilder.RefreshSchema-Methode aufrufen, sofern diese den ULDataAdapter.SelectCommand-Wert ändert. Sonst verwendet die GetInsertCommand-Methode weiterhin Informationen aus der vorhergehenden Anweisung, die nicht unbedingt richtig sein müssen. Die SQL-Anweisungen werden erstmals generiert, wenn die Anwendung entweder die DbDataAdapter.Update(System.Data.DataSet)-Methode oder die GetInsertCommand-Methode aufruft.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULDataAdapter.SelectCommand-Eigenschaft \[UltraLite.NET\] auf Seite 219](#)
- [System.Data.Common.DbCommandBuilder.DataAdapter](#)
- [System.Data.Common.DbCommandBuilder.RefreshSchema](#)

GetUpdateCommand-Methode

Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Aktualisierungen in der Datenbank erforderlich ist.

Überladungsliste

Name	Beschreibung
GetUpdateCommand()-Methode	Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Aktualisierungen in der Datenbank erforderlich ist.
GetUpdateCommand(bool)-Methode	Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Aktualisierungen in der Datenbank erforderlich ist.

GetUpdateCommand()-Methode

Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Aktualisierungen in der Datenbank erforderlich ist.

Visual Basic-Syntax

```
Public Shadows Function GetUpdateCommand() As ULCommand
```

C#-Syntax

```
public new ULCommand GetUpdateCommand()
```

Rückgabe

Das automatisch generierte ULCommand-Objekt, das für die Durchführung von Aktualisierungen erforderlich ist.

Ausnahmen

- **InvalidOperationException** Die DbCommandBuilder.DataAdapter-Eigenschaft wurde nicht initialisiert. Die DataAdapter.SelectCommand-Eigenschaft wurde nicht initialisiert. Die DataAdapter.SelectCommand.Connection-Eigenschaft wurde nicht initialisiert. Die Dynamic SQL-Generierung für die UpdateCommand-Eigenschaft wird für einen SelectCommand-Wert, der keine änderbaren Spalten zurückgibt, nicht unterstützt. Die Dynamic SQL-Generierung für mehrere Basistabellen wird nicht unterstützt. Die Dynamic SQL-Generierung wird für einen SelectCommand-Wert, der doppelte Spalten enthält, nicht unterstützt. Die Dynamic SQL-Generierung für die UpdateCommand-Eigenschaft wird für einen SelectCommand-Wert, der keine Schlüsselspalteninformationen zurückgibt, nicht unterstützt.

Bemerkungen

Nachdem die SQL-Anweisung erstmals generiert wurde, muss die Anwendung explizit die DbCommandBuilder.RefreshSchema-Methode aufrufen, sofern diese den ULDataAdapter.SelectCommand-Wert ändert. Sonst verwendet die GetUpdateCommand-Methode weiterhin Informationen aus der vorhergehenden Anweisung, die nicht unbedingt richtig sein müssen. Die SQL-Anweisungen werden erstmals generiert, wenn die Anwendung entweder die DbDataAdapter.Update(System.Data.DataSet)-Methode oder die GetUpdateCommand-Methode aufruft.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULDataAdapter.SelectCommand-Eigenschaft \[UltraLite.NET\] auf Seite 219](#)
- [System.Data.Common.DbCommandBuilder.DataAdapter](#)
- [System.Data.Common.DbCommandBuilder.RefreshSchema](#)

GetUpdateCommand(bool)-Methode

Ruft das automatisch generierte ULCommand-Objekt ab, das zur Ausführung von Aktualisierungen in der Datenbank erforderlich ist.

Visual Basic-Syntax

```
Public Shadows Function GetUpdateCommand(  
    ByVal useColumnsForParameterNames As Boolean  
) As ULCommand
```

C#-Syntax

```
public new ULCommand GetUpdateCommand(bool useColumnsForParameterNames)
```

Parameter

- **useColumnsForParameterNames** Wenn der Wert TRUE ergibt, werden Parameternamen generiert, die mit dem Spaltennamen übereinstimmen, sofern möglich. Wenn der Wert FALSE ergibt, werden @p1, @p2 usw. generiert.

Rückgabe

Das automatisch generierte ULCommand-Objekt, das für die Durchführung von Aktualisierungen erforderlich ist.

Ausnahmen

- **InvalidOperationException** Die DbCommandBuilder.DataAdapter-Eigenschaft wurde nicht initialisiert. Die DataAdapter.SelectCommand-Eigenschaft wurde nicht initialisiert. Die DataAdapter.SelectCommand.Connection-Eigenschaft wurde nicht initialisiert. Die Dynamic SQL-Generierung für die UpdateCommand-Eigenschaft wird für einen SelectCommand-Wert, der keine änderbaren Spalten zurückgibt, nicht unterstützt. Die Dynamic SQL-Generierung für mehrere Basistabellen wird nicht unterstützt. Die Dynamic SQL-Generierung wird für einen SelectCommand-Wert, der doppelte Spalten enthält, nicht unterstützt. Die Dynamic SQL-Generierung für die UpdateCommand-Eigenschaft wird für einen SelectCommand-Wert, der keine Schlüsselspalteninformationen zurückgibt, nicht unterstützt.

Bemerkungen

Nachdem die SQL-Anweisung generiert wurde, muss die Anwendung explizit DbCommandBuilder.RefreshSchema aufrufen, sofern sie ULDataAdapter.SelectCommand ändert. Sonst verwendet die GetUpdateCommand-Methode weiterhin Informationen aus der vorhergehenden Anweisung, die nicht unbedingt richtig sein müssen. Die SQL-Anweisungen werden erstmals generiert, wenn die Anwendung entweder die DbDataAdapter.Update(System.Data.DataSet)-Methode oder die GetUpdateCommand-Methode aufruft.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULDataAdapter.SelectCommand-Eigenschaft \[UltraLite.NET\] auf Seite 219](#)
- [System.Data.Common.DbCommandBuilder.DataAdapter](#)
- [System.Data.Common.DbCommandBuilder.RefreshSchema](#)

DataAdapter-Eigenschaft

Ruft ein ULDataAdapter-Objekt ab, für das SQL-Anweisungen automatisch generiert werden, oder legt das Objekt fest.

Visual Basic-Syntax

```
Public Shadows Property DataAdapter As ULDataAdapter
```

C#-Syntax

```
public new ULDataAdapter DataAdapter {get;set;}
```

Bemerkungen

Ein ULDataAdapter-Objekt

Siehe auch

- [ULDataAdapter-Klasse \[UltraLite.NET\] auf Seite 212](#)

ULConnection-Klasse

Repräsentiert eine Verbindung zu einer UltraLite.NET-Datenbank.

Visual Basic-Syntax

```
Public NotInheritable Class ULConnection  
    Inherits System.Data.Common.DbConnection
```

C#-Syntax

```
public sealed class ULConnection : System.Data.Common.DbConnection
```

Basisklassen

- [System.Data.Common.DbConnection](#)

Mitglieder

Alle Mitglieder der ULConnection-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ULConnection-Konstruktor	Initialisiert ein ULConnection-Objekt.

Name	Beschreibung
BeginDbTransaction-Methode (geerbt aus System.Data.Common.DbConnection)	Startet eine Datenbanktransaktion.
BeginSynchronize-Methode	UL-Erw.: Startet asynchron eine Synchronisation unter Verwendung des aktuellen SyncParms-Objekts.
BeginTransaction-Methode	Gibt ein Transaktionsobjekt zurück.
CancelGetNotification-Methode	UL Ext: Bricht ausstehende getNotification-Abrufe in allen Warteschlangen ab, die mit dem angegebenen Namen übereinstimmen.
CancelSynchronize-Methode	UL-Erw.: Bewirkt, dass eine laufende Synchronisation bei der nächsten Gelegenheit abgebrochen wird.
ChangeDatabase-Methode	Ändert die aktuelle Datenbank für eine geöffnete Verbindung.
ChangeEncryptionKey-Methode	UL-Erw.: Ändert den Chiffrierschlüssel der Datenbank zum angegebenen neuen Schlüssel.
ChangePassword-Methode	Ändert das Kennwort für den Benutzer, der in der Verbindungszeichenfolge für das bereitgestellte neue Kennwort angegeben ist.
Close-Methode	Schließt die Datenbankverbindung.
CountUploadRows-Methode	UL-Erw.: Gibt die Anzahl der Zeilen zurück, die hochgeladen werden müssen, wenn die nächste Synchronisation ausgeführt wird.
CreateCommand-Methode	Erstellt und initialisiert ein ULCommand-Objekt, das seiner Verbindung und seiner aktuellen Transaktion zugeordnet ist.
CreateNotificationQueue-Methode	UL Ext: Erstellt eine Ereigniswarteschlange.
DeclareEvent-Method	UL Ext: Deklariert ein benanntes Ereignis.
DestroyNotificationQueue-Methode	UL Ext: Vernichtet eine Ereigniswarteschlange.
EndSynchronize-Methode	UL-Erw.: Wird blockiert, bis eine asynchron gestartete Synchronisation beendet ist.
EnlistTransaction-Methode (Geerbt von System.Data.Common.DbConnection)	Tritt in die angegebene Transaktion ein.

Name	Beschreibung
ExecuteTable-Methode	UL-Erw.: Ruft eine Datenbanktabelle in einem ULTable-Objekt zur direkten Bearbeitung ab.
GetLastDownloadTime-Methode	UL-Erw.: Gibt die Uhrzeit des letzten Downloads der angegebenen Publikation zurück.
GetNewUUID-Methode	UL-Erw.: Generiert eine neue UUID (System.Guid).
GetNotification-Methode	UL Ext: Blockiert für eine Benachrichtigung oder einen Timeout.
GetNotificationParameter-Methode	UL Ext: Ruft den Wert eines Parameters für ein Ereignis ab, das gerade von der GetNotification-Methode gelesen wurde.
GetSchema-Methode	Gibt die Liste der unterstützten Schemasammlungen zurück.
GrantConnectTo-Methode	UL-Erw.: Erteilt einer Benutzer-ID mit einem bestimmten Kennwort die Zugriffsberechtigung für eine UltraLite-Datenbank.
OnStateChange-Methode (geerbt aus System.Data.Common.DbConnection))	Löst das System.Data.Common.DbConnection.StateChange -Ereignis aus.
Open-Methode	Öffnet eine Verbindung mit einer Datenbank unter Verwendung der vorher definierten Verbindungszeichenfolge.
OpenAsync-Methode (geerbt aus System.Data.Common.DbConnection))	Eine asynchrone Version von System.Data.Common.DbConnection.Open , die eine Datenbankverbindung mit den durch System.Data.Common.DbConnection.ConnectionString angegebenen Einstellungen öffnet.
RegisterForEvent-Methode	UL Ext: Registriert eine Warteschlange, um Ereignisse von einem Objekt zu beziehen.
ResetLastDownloadTime-Methode	UL-Erw.: Setzt den Zeitpunkt des letzten Downloads zurück.
RevokeConnectFrom-Methode	UL-Erw.: Entzieht der angegebenen Benutzer-ID die Zugriffsberechtigung auf eine UltraLite-Datenbank.
RollbackPartialDownload-Methode	UL-Erw.: Setzt noch nicht festgeschriebene Änderungen aus einem Teil-Download in der Datenbank zurück.
SendNotification-Methode	UL Ext: Sendet eine Benachrichtigung an passende Warteschlangen.

Name	Beschreibung
SetSyncListener-Methode	Gibt das Listener-Objekt an, mit dem Synchronisationsnachrichten verarbeitet werden.
StartSynchronizationDelete-Methode	UL-Erw.: Markiert alle nachfolgenden Löschungen, die von dieser Verbindung für die Synchronisation vorgenommen wurden.
StopSynchronizationDelete-Methode	UL-Erw.: Verhindert, dass Löschvorgänge synchronisiert werden.
Synchronize-Methode	UL-Erw.: Synchronisiert die Datenbank unter Verwendung des aktuellen ULConnection.SyncParms-Objekts.
TriggerEvent-Methode	UL Ext: Löst ein Ereignis aus.
ValidateDatabase-Methode	UL Ext: Führt die Validierung der aktuellen Datenbank durch.
ConnectionString-Eigenschaft	Legt die Parameter zum Öffnen einer Verbindung zu einer UltraLite.NET-Datenbank fest.
ConnectionTimeout-Eigenschaft	Diese Funktion wird von UltraLite.NET nicht unterstützt.
Database-Eigenschaft	Gibt den Namen der Datenbank zurück, für die die Verbindung geöffnet wird.
DatabaseID-Eigenschaft	UL-Erw.: Legt den Datenbank-ID-Wert fest, der für global autoincrement-Spalten verwendet wird.
DataSource-Eigenschaft	Diese Funktion wird von UltraLite.NET nicht unterstützt.
DbProviderFactory-Eigenschaft (geerbt aus System.Data.Common.DbConnection)	Ruft das System.Data.Common.DbProviderFactory -Objekt für diese System.Data.Common.DbConnection ab.
GlobalAutoIncrementUsage-Eigenschaft	UL-Erw.: Gibt den Prozentsatz noch verfügbarer Werte für global autoincrement an.
LastIdentity-Eigenschaft	UL-Erw.: Gibt den zuletzt verwendeten Identity-Wert zurück.
Schema-Eigenschaft	UL-Erw.: Stellt Zugriff auf das Schema der aktuellen Datenbank bereit, die dieser Verbindung zugeordnet ist.
ServerVersion-Eigenschaft	Diese Funktion wird von UltraLite.NET nicht unterstützt.
State-Eigenschaft	Gibt den aktuellen Status der Verbindung zurück.

Name	Beschreibung
SyncParms-Eigenschaft	UL-Erw.: Gibt die Synchronisationseinstellungen für diese Verbindung an.
SyncResult-Eigenschaft	UL-Erw.: Gibt die Ergebnisse der letzten Synchronisation für diese Verbindung zurück.
InfoMessage-Ereignis	Tritt ein, wenn UltraLite.NET eine Warnung oder eine Informationsnachricht auf dieser Verbindung absetzt
StateChange-Ereignis	Tritt ein, wenn sich der Status dieser Verbindung ändert
INVALID_DATABASE_ID-Feld	UL-Erw.: Eine Datenbank-ID-Konstante, die angibt, dass die ULConnection.DatabaseID-Eigenschaft nicht festgelegt wurde.
SYNC_ALL_DB-Feld	Leere Publikationsliste, steht für die gesamte Datenbank.
SYNC_ALL_PUBS-Feld	Publikationsname "*", steht für alle Publikationen.

Bemerkungen

Um die UltraLite-Engine-Laufzeitumgebung von UltraLite.NET zu verwenden, setzen Sie die ULDatabaseManager.RuntimeType-Eigenschaft auf den zutreffenden Wert, bevor Sie eine andere UltraLite.NET-API verwenden.

Eine Verbindung zu einer vorhandenen Datenbank wird mit der ULConnection.Open-Methode geöffnet.

Sie müssen eine Verbindung öffnen, bevor Sie andere Vorgänge ausführen können, und Sie müssen die Verbindung schließen, nachdem Sie alle Vorgänge in der Verbindung abgeschlossen haben und bevor Sie Ihre Anwendung beenden. Außerdem müssen Sie alle in einer Verbindung geöffneten Ergebnismengen und Tabellen schließen, bevor Sie die Verbindung schließen.

Auf das Schema der Datenbank kann mit dem ULConnection.Schema-Wert einer offenen Verbindung zugegriffen werden.

Siehe auch

- [ULConnection.Open-Methode \[UltraLite.NET\] auf Seite 150](#)
- [ULConnection.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 164](#)
- [System.Data.IDbConnection](#)
- [System.IDisposable](#)

ULConnection-Konstruktor

Initialisiert ein ULConnection-Objekt.

Überladungsliste

Name	Beschreibung
ULConnection()-Konstruktor	Initialisiert ein ULConnection-Objekt.
ULConnection(string)-Konstruktor	Initialisiert ein ULConnection-Objekt mit der angegebenen Verbindungszeichenfolge.

ULConnection()-Konstruktor

Initialisiert ein ULConnection-Objekt.

Visual Basic-Syntax

```
Public Sub New()
```

C#-Syntax

```
public ULConnection()
```

Bemerkungen

Die Verbindung muss geöffnet werden, damit Sie Vorgänge mit der Datenbank ausführen können.

Um die UltraLite-Engine-Laufzeitumgebung von UltraLite.NET zu verwenden, setzen Sie die `ULDatabaseManager.RuntimeType`-Eigenschaft auf den zutreffenden Wert, bevor Sie eine andere UltraLite.NET-API verwenden.

Damit das `ULConnection`-Objekt geöffnet werden kann, muss seine `ULConnection.ConnectionString`-Eigenschaft festgelegt sein.

Siehe auch

- [ULConnection.Open-Methode \[UltraLite.NET\] auf Seite 150](#)
- [ULConnection.ConnectionString-Eigenschaft \[UltraLite.NET\] auf Seite 160](#)

ULConnection(string)-Konstruktor

Initialisiert ein ULConnection-Objekt mit der angegebenen Verbindungszeichenfolge.

Visual Basic-Syntax

```
Public Sub New(ByVal connectionString As String)
```

C#-Syntax

```
public ULConnection(string connectionString)
```

Parameter

- **connectionString** Eine UltraLite.NET-Verbindungszeichenfolge. Eine Verbindungszeichenfolge ist eine durch Semikola getrennte Liste von Schlüsselwort-Wert-Paaren.

Ausnahmebedingungen

- **ArgumentException** Die bereitgestellte Verbindungszeichenfolge ist ungültig.

Bemerkungen

Die Verbindung muss geöffnet werden, damit Sie Vorgänge mit der Datenbank ausführen können.

Um die UltraLite-Engine-Laufzeitumgebung von UltraLite.NET zu verwenden, setzen Sie die `ULDatabaseManager.RuntimeType`-Eigenschaft auf den zutreffenden Wert, bevor Sie eine andere UltraLite.NET-API verwenden.

Die Verbindungszeichenfolge kann mit einem `ULConnectionParms`-Objekt bereitgestellt werden.

Siehe auch

- [ULConnection.Open-Methode \[UltraLite.NET\] auf Seite 150](#)
- [ULConnection.ULConnection-Konstruktor \[UltraLite.NET\] auf Seite 125](#)
- [ULConnectionParms-Klasse \[UltraLite.NET\] auf Seite 169](#)
- [ULConnection.ConnectionString-Eigenschaft \[UltraLite.NET\] auf Seite 160](#)

Beispiel

Der folgende Code erstellt und öffnet eine Verbindung zur vorhandenen Datenbank `\UltraLite\MyDatabase.udb` auf einem Windows Mobile-Gerät.

```
' Visual Basic
Dim openParms As ULConnectionParms = New ULConnectionParms
openParms.DatabaseOnDevice = "\UltraLite\MyDatabase.udb"
Dim conn As ULConnection = _
    New ULConnection( openParms.ToString() )
conn.Open()
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnDevice = @"\UltraLite\MyDatabase.udb";
ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```

BeginSynchronize-Methode

UL-Erw.: Startet asynchron eine Synchronisation unter Verwendung des aktuellen `SyncParms`-Objekts.

Überladungsliste

Name	Beschreibung
BeginSynchronize()-Methode	UL-Erw.: Startet asynchron eine Synchronisation unter Verwendung des aktuellen <code>SyncParms</code> -Objekts.

Name	Beschreibung
BeginSynchronize(Control, ULSyncProgressedDlg, object)-Methode	UL-Erw.: Startet asynchron eine Synchronisation unter Verwendung des aktuellen SyncParms-Objekts.

BeginSynchronize()-Methode

UL-Erw.: Startet asynchron eine Synchronisation unter Verwendung des aktuellen SyncParms-Objekts.

Visual Basic-Syntax

```
Public Function BeginSynchronize() As IAsyncResult
```

C#-Syntax

```
public IAsyncResult BeginSynchronize()
```

Rückgabe

Ein IAsyncResult-Objekt, mit dem ermittelt werden kann, ob die Synchronisation abgeschlossen ist oder sie blockiert werden kann, bis sie abgeschlossen ist.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Diese Methode erstellt einen neuen Thread, um die Synchronisation durchzuführen und danach sofort zurückzugeben. Rufen Sie die EndSynchronize-Methode auf, um die Synchronisation zu blockieren, bis sie abgeschlossen ist.

Siehe auch

- [ULConnection.BeginSynchronize-Methode \[UltraLite.NET\] auf Seite 127](#)
- [ULConnection.EndSynchronize-Methode \[UltraLite.NET\] auf Seite 138](#)
- [ULConnection.CancelSynchronize-Methode \[UltraLite.NET\] auf Seite 133](#)
- [ULSyncProgressedDlg-Delegat \[UltraLite.NET\] auf Seite 455](#)

BeginSynchronize(Control, ULSyncProgressedDlg, object)-Methode

UL-Erw.: Startet asynchron eine Synchronisation unter Verwendung des aktuellen SyncParms-Objekts.

Visual Basic-Syntax

```
Public Function BeginSynchronize(  
    ByVal control As Control,  
    ByVal dlg As ULSyncProgressedDlg,  
    ByVal state As Object  
) As IAsyncResult
```

C#-Syntax

```
public IAsyncResult BeginSynchronize(
    Control control,
    ULSyncProgressedDlg dlg,
    object state
)
```

Parameter

- **control** Ein System.Windows.Forms.Ctrl-Objekt, das der Synchronisations-Thread verwendet, um ULSyncProgressedDlg-Aufrufe aufzurufen.
- **dlg** Eine ULSyncProgressedDlg Methode, die regelmäßig mit Aktualisierungen des Synchronisationsfortschritts aufgerufen wird.
- **state** Auf diesen Benutzerkontext kann in der ULSyncProgressedDlg-Methode mithilfe von IAsyncResult.AsyncState zugegriffen werden.

Rückgabe

Ein IAsyncResult-Objekt, mit dem ermittelt werden kann, ob die Synchronisation abgeschlossen ist oder sie blockiert werden kann, bis sie abgeschlossen ist.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Diese Methode erstellt einen neuen Thread, um die Synchronisation durchzuführen und danach sofort zurückzugeben. Dabei wird die angegebene ULSyncProgressedDlg-Methode regelmäßig mit Aktualisierungen des Synchronisationsfortschritts aufgerufen. Rufen Sie EndSynchronize auf, um die Synchronisation zu blockieren, bis sie abgeschlossen ist.

Siehe auch

- [ULConnection.BeginSynchronize-Methode \[UltraLite.NET\] auf Seite 127](#)
- [ULConnection.EndSynchronize-Methode \[UltraLite.NET\] auf Seite 138](#)
- [ULConnection.CancelSynchronize-Methode \[UltraLite.NET\] auf Seite 133](#)
- [ULSyncProgressedDlg-Delegat \[UltraLite.NET\] auf Seite 455](#)

BeginTransaction-Methode

Gibt ein Transaktionsobjekt zurück.

Überladungsliste

Name	Beschreibung
BeginTransaction()-Methode	Gibt ein Transaktionsobjekt zurück.

Name	Beschreibung
BeginTransaction(IsolationLevel)-Methode	Gibt ein Transaction-Objekt mit der angegebenen Isolationsstufe zurück.

BeginTransaction()-Methode

Gibt ein Transaktionsobjekt zurück.

Visual Basic-Syntax

```
Public Shadows Function BeginTransaction() As ULTransaction
```

C#-Syntax

```
public new ULTransaction BeginTransaction()
```

Rückgabe

Ein ULTransaction-Objekt, das die neue Transaktion darstellt.

Ausnahmen

- [ULException-Klasse](#) Die Verbindung ist geschlossen.
- [InvalidOperationException](#) Die ULConnection-Klasse unterstützt keine parallelen Transaktionen.

Bemerkungen

Befehle, die mit einem Transaction-Objekt verbunden sind, werden als einzelne Transaktion ausgeführt. Die Transaktion wird mit der Methode ULTransaction.Commit oder ULTransaction.Rollback beendet.

Die Transaktion wird mit dem IsolationLevel.ReadCommitted-Wert erstellt.

Um einen Befehl mit einem Transaktionsobjekt zu verbinden, verwenden Sie die ULCommand.Transaction-Eigenschaft. Die aktuelle Transaktion wird automatisch Befehlen zugeordnet, die von der ULConnection.CreateCommand-Methode erstellt wurden.

Standardmäßig verwendet die Verbindung keine Transaktionen, und alle Befehle werden automatisch nach ihrer Ausführung festgeschrieben. Wenn die aktuelle Transaktion festgeschrieben oder zurückgesetzt wurde, kehrt die Verbindung bis zum nächsten Aufruf der BeginTransaction-Methode in den automatischen Festschreibemodus und zur vorherigen Isolationsstufe zurück.

Die UltraLite-Definition jeder Isolationsstufe unterscheidet sich geringfügig von der ADO.NET-Dokumentation des IsolationLevel-Objekts.

Dies ist die stark typisierte Version der Methoden System.Data.IDbConnection.BeginTransaction und System.Data.Common.DbConnection.BeginTransaction().

Siehe auch

- [ULTransaction.Commit-Methode \[UltraLite.NET\] auf Seite 451](#)
- [ULTransaction.Rollback-Methode \[UltraLite.NET\] auf Seite 452](#)
- [ULCommand.Transaction-Eigenschaft \[UltraLite.NET\] auf Seite 108](#)
- [ULConnection.CreateCommand-Methode \[UltraLite.NET\] auf Seite 136](#)
- [System.Data.IDbConnection.BeginTransaction](#)
- [System.Data.Common.DbConnection.BeginTransaction](#)
- „Isolationsstufen“ [*UltraLite - Datenbankverwaltung*]

BeginTransaction(IsolationLevel)-Methode

Gibt ein Transaction-Objekt mit der angegebenen Isolationsstufe zurück.

Visual Basic-Syntax

```
Public Shadows Function BeginTransaction(  
    ByVal isolationLevel As IsolationLevel  
) As ULTransaction
```

C#-Syntax

```
public new ULTransaction BeginTransaction(IsolationLevel isolationLevel)
```

Parameter

- **isolationLevel** Die erforderliche Isolationsstufe für die Transaktion. UltraLite.NET unterstützt nur die Werte System.Data.IsolationLevel.ReadUncommitted und ReadCommitted.

Rückgabe

Ein ULTransaction-Objekt, das die neue Transaktion darstellt.

Ausnahmen

- **ULException-Klasse** Die Verbindung ist geschlossen oder es wurde eine nicht unterstützte Isolationsstufe angegeben.
- **InvalidOperationException** Die ULConnection-Klasse unterstützt keine parallelen Transaktionen.

Bemerkungen

Befehle, die mit einem Transaction-Objekt verbunden sind, werden als einzelne Transaktion ausgeführt. Die Transaktion wird mit der Methode ULTransaction.Commit oder ULTransaction.Rollback beendet.

Um einen Befehl mit einem Transaktionsobjekt zu verbinden, verwenden Sie die ULCommand.Transaction-Eigenschaft. Die aktuelle Transaktion wird automatisch Befehlen zugeordnet, die von der ULConnection.CreateCommand-Methode erstellt wurden.

Standardmäßig verwendet die Verbindung keine Transaktionen, und alle Befehle werden automatisch nach ihrer Ausführung festgeschrieben. Wenn die aktuelle Transaktion festgeschrieben oder zurückgesetzt wurde, kehrt die Verbindung bis zum nächsten Aufruf der BeginTransaction-Methode in den automatischen Festschreibemodus und zur vorherigen Isolationsstufe zurück.

Die UltraLite-Definition jeder Isolationsstufe unterscheidet sich geringfügig von der ADO.NET-Dokumentation des IsolationLevel-Objekts.

Dies ist die stark typisierte Version der Methoden
System.Data.IDbConnection.BeginTransaction(System.Data.IsolationLevel) und
System.Data.Common.DbConnection.BeginTransaction(System.Data.IsolationLevel).

Siehe auch

- [ULTransaction-Klasse \[UltraLite.NET\] auf Seite 450](#)
- [ULConnection.BeginTransaction-Methode \[UltraLite.NET\] auf Seite 129](#)
- [ULTransaction.Commit-Methode \[UltraLite.NET\] auf Seite 451](#)
- [ULTransaction.Rollback-Methode \[UltraLite.NET\] auf Seite 452](#)
- [ULCommand.Transaction-Eigenschaft \[UltraLite.NET\] auf Seite 108](#)
- [ULConnection.CreateCommand-Methode \[UltraLite.NET\] auf Seite 136](#)
- [System.Data.IDbConnection.BeginTransaction](#)
- [System.Data.Common.DbConnection.BeginTransaction](#)
- [System.Data.IsolationLevel](#)
- „Isolationsstufen“ [*UltraLite - Datenbankverwaltung*]

CancelGetNotification-Methode

UL Ext: Bricht ausstehende getNotification-Abrufe in allen Warteschlangen ab, die mit dem angegebenen Namen übereinstimmen.

Visual Basic-Syntax

```
Public Function CancelGetNotification(  
    ByVal queueName As String  
) As Integer
```

C#-Syntax

```
public int CancelGetNotification(string queueName)
```

Parameter

- **queueName** Der Ausdruck, zu dem Warteschlangennamen passen müssen.

Rückgabe

Die Anzahl der betroffenen Warteschlangen (nicht notwendigerweise die Anzahl der blockierten Lesevorgänge).

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Diese Methode bricht ausstehende getNotification-Abrufe in allen Warteschlangen ab, die mit dem angegebenen Namen übereinstimmen.

Siehe auch

- [ULConnection.GetNotification-Methode \[UltraLite.NET\] auf Seite 145](#)
- [ULException-Klasse \[UltraLite.NET\] auf Seite 275](#)

CancelSynchronize-Methode

UL-Erw.: Bewirkt, dass eine laufende Synchronisation bei der nächsten Gelegenheit abgebrochen wird.

Visual Basic-Syntax

```
Public Sub CancelSynchronize(ByVal asyncResult As IAsyncResult)
```

C#-Syntax

```
public void CancelSynchronize(IAsyncResult asyncResult)
```

Parameter

- **asyncResult** Der von der BeginSynchronize-Methode zurückgegebene IAsyncResult-Wert.

Bemerkungen

Diese Methode weist die Beendigung des Synchronisations-Threads an und kehrt dann sofort zurück. Rufen Sie die EndSynchronize-Methode auf, um die Synchronisation zu blockieren, bis sie erfolgreich abgeschlossen wurde.

Siehe auch

- [ULConnection.BeginSynchronize-Methode \[UltraLite.NET\] auf Seite 127](#)
- [ULConnection.EndSynchronize-Methode \[UltraLite.NET\] auf Seite 138](#)

ChangeDatabase-Methode

Ändert die aktuelle Datenbank für eine geöffnete Verbindung.

Visual Basic-Syntax

```
Public Overrides Sub ChangeDatabase(ByVal connectionString As String)
```

C#-Syntax

```
public override void ChangeDatabase(string connectionString)
```

Parameter

- **connectionString** Eine vollständige Verbindungszeichenfolge zum Öffnen der Verbindung zu einer neuen Datenbank

Bemerkungen

Die Verbindung zur aktuellen Datenbank wird geschlossen, selbst wenn Parameterfehler vorliegen.

UL-Erw.: *connectionString* ist eine vollständige Verbindungszeichenfolge, kein dbn- oder dbf-Wert.

Siehe auch

- [ULConnection.ConnectionString-Eigenschaft \[UltraLite.NET\] auf Seite 160](#)

ChangeEncryptionKey-Methode

UL-Erw.: Ändert den Chiffrierschlüssel der Datenbank zum angegebenen neuen Schlüssel.

Visual Basic-Syntax

```
Public Sub ChangeEncryptionKey(ByVal newKey As String)
```

C#-Syntax

```
public void ChangeEncryptionKey(string newKey)
```

Parameter

- **newKey** Der neue Chiffrierschlüssel für die Datenbank.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Wenn der Chiffrierschlüssel verloren geht, ist es nicht möglich, die Datenbank zu öffnen.

Siehe auch

- [ULConnectionParms.EncryptionKey-Eigenschaft \[UltraLite.NET\] auf Seite 177](#)

ChangePassword-Methode

Ändert das Kennwort für den Benutzer, der in der Verbindungszeichenfolge für das bereitgestellte neue Kennwort angegeben ist.

Visual Basic-Syntax

```
Public Shared Sub ChangePassword(  
    ByVal connectionString As String,  
    ByVal newPassword As String  
)
```

C#-Syntax

```
public static void ChangePassword(  
    string connectionString,  
    string newPassword  
)
```

Parameter

- **connectionString** Die Verbindungszeichenfolge, die genügend Informationen enthält, damit eine Verbindung mit der gewünschten Datenbank hergestellt werden kann. Die Verbindungszeichenfolge kann die Benutzer-ID und das aktuelle Kennwort enthalten.
- **newPassword** Das gewünschte neue Kennwort.

Ausnahmebedingungen

- **ArgumentNullException** Entweder der connectionString-Parameter oder der newPassword-Parameter ist Null.
- **ArgumentException** Die Verbindungszeichenfolge umfasst die Option zur Verwendung der integrierten Sicherheit.
- **ULException-Klasse** Beim Versuch die Datenbank zu öffnen, ist ein SQL-Fehler aufgetreten.

Close-Methode

Schließt die Datenbankverbindung.

Visual Basic-Syntax

```
Public Overrides Sub Close()
```

C#-Syntax

```
public override void Close()
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Close-Methode setzt noch nicht festgeschriebene Transaktionen zurück und schließt dann die Verbindung. Eine Anwendung kann diese Methode mehrmals aufrufen.

CountUploadRows-Methode

UL-Erw.: Gibt die Anzahl der Zeilen zurück, die hochgeladen werden müssen, wenn die nächste Synchronisation ausgeführt wird.

Visual Basic-Syntax

```
Public Function CountUploadRows(  
    ByVal pubs As String,  
    ByVal threshold As Long  
) As Long
```

C#-Syntax

```
public long CountUploadRows(string pubs, long threshold)
```

Parameter

- **pubs** Eine kommagetrennte Liste von Publikationen, in denen Zeilen zu suchen sind.
- **threshold** Die maximale Anzahl der zu zählenden Zeilen, die die von CountUploadRows genommene Zeitdauer begrenzt. Der Wert 0 entspricht der Obergrenze. Der Wert 1 bestimmt, ob Zeilen synchronisiert werden müssen.

Rückgabe

Die Anzahl der Zeilen, die bei der nächsten Synchronisation aus den angegebenen Publikationen übertragen werden müssen

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

CreateCommand-Methode

Erstellt und initialisiert ein ULCommand-Objekt, das seiner Verbindung und seiner aktuellen Transaktion zugeordnet ist.

Visual Basic-Syntax

```
Public Shadows Function CreateCommand() As ULCommand
```

C#-Syntax

```
public new ULCommand CreateCommand()
```

Rückgabe

Ein neues ULCommand-Objekt

Bemerkungen

Sie können die Eigenschaften des ULCommand-Objekts benutzen, um sein Verhalten zu steuern.

Damit der Befehl ausgeführt werden kann, muss die ULCommand.CommandText-Eigenschaft festgelegt werden.

Dies ist die stark typisierte Version der Methoden System.Data.IDbConnection.CreateCommand und System.Data.Common.DbConnection.CreateCommand.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [System.Data.IDbConnection.CreateCommand](#)
- [System.Data.Common.DbConnection.CreateCommand](#)

CreateNotificationQueue-Methode

UL Ext: Erstellt eine Ereigniswarteschlange.

Visual Basic-Syntax

```
Public Sub CreateNotificationQueue(  
    ByVal queueName As String,  
    ByVal parameters As String  
)
```

C#-Syntax

```
public void CreateNotificationQueue(string queueName, string parameters)
```

Parameter

- **queueName** Der Name der neuen Warteschlange.
- **parameters** Erstellungsparameter, derzeit nicht verwendet, auf NULL gesetzt.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Diese Methode erstellt eine Ereignisbenachrichtigungs-Warteschlange für diese Verbindung. Der Bereich von Warteschlangennamen gilt für die jeweilige Verbindung, daher können verschiedene Verbindungen Warteschlangen mit demselben Namen erstellen. Wenn eine Ereignisbenachrichtigung gesendet wird, empfangen alle Warteschlangen in der Datenbank mit einem übereinstimmenden Namen eine (separate Instanz der) Benachrichtigung. Namen reagieren nicht auf Groß- und Kleinschreibung. Beim Aufrufen des RegisterForEvent-Ereignisses wird für jede Verbindung eine Standard-Warteschlange erstellt, falls keine Warteschlange angegeben ist.

Siehe auch

- [ULConnection.DestroyNotificationQueue-Methode \[UltraLite.NET\] auf Seite 138](#)
- [ULException-Klasse \[UltraLite.NET\] auf Seite 275](#)

DeclareEvent-Method

UL Ext: Deklariert ein benanntes Ereignis.

Visual Basic-Syntax

```
Public Sub DeclareEvent(ByVal eventName As String)
```

C#-Syntax

```
public void DeclareEvent(string eventName)
```

Parameter

- **eventName** Der Ereignisname

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Deklariert ein Ereignis, für das man sich anschließend registrieren kann und das ausgelöst werden kann. In UltraLite gibt es vordefinierte Systemereignisse, die von Vorgängen in der Datenbank oder in der Umgebung ausgelöst werden. Der Ereignisname muss eindeutig sein. Namen berücksichtigen nicht die Groß- und Kleinschreibung. Gibt einen Fehler aus, wenn der Name bereits verwendet wird oder ungültig ist.

Siehe auch

- [ULConnection.CreateNotificationQueue-Methode \[UltraLite.NET\] auf Seite 137](#)
- [ULException-Klasse \[UltraLite.NET\] auf Seite 275](#)

DestroyNotificationQueue-Methode

UL Ext: Vernichtet eine Ereigniswarteschlange.

Visual Basic-Syntax

```
Public Sub DestroyNotificationQueue(ByVal queueName As String)
```

C#-Syntax

```
public void DestroyNotificationQueue(string queueName)
```

Parameter

- **queueName** Der Name der Warteschlange.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Vernichtet die Ereignisbenachrichtigungs-Warteschlange. Eine Warnung wird signalisiert, wenn ungelesene Benachrichtigungen in der Warteschlange verbleiben. Ungelesene Benachrichtigungen werden verworfen. Die Standard-Ereigniswarteschlange einer Verbindung, falls erstellt, wird vernichtet, wenn die Verbindung geschlossen wird.

Siehe auch

- [ULConnection.CreateNotificationQueue-Methode \[UltraLite.NET\] auf Seite 137](#)
- [ULException-Klasse \[UltraLite.NET\] auf Seite 275](#)

EndSynchronize-Methode

UL-Erw.: Wird blockiert, bis eine asynchron gestartete Synchronisation beendet ist.

Visual Basic-Syntax

```
Public Sub EndSynchronize(ByVal asyncResult As IAsyncResult)
```

C#-Syntax

```
public void EndSynchronize(IAsyncResult asyncResult)
```

Parameter

- **asyncResult** Der von der BeginSynchronize-Methode zurückgegebene IAsyncResult-Wert.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Wenn während der Synchronisation ein Fehler aufgetreten ist, wird eine ULException-Ausnahmebedingung ausgegeben.

Siehe auch

- [ULConnection.BeginSynchronize-Methode \[UltraLite.NET\] auf Seite 127](#)
- [ULConnection.CancelSynchronize-Methode \[UltraLite.NET\] auf Seite 133](#)

ExecuteTable-Methode

UL-Erw.: Ruft eine Datenbanktabelle in einem ULTable-Objekt zur direkten Bearbeitung ab.

Überladungsliste

Name	Beschreibung
ExecuteTable(string)-Methode	UL-Erw.: Ruft eine Datenbanktabelle in einem ULTable-Objekt zur direkten Bearbeitung ab.
ExecuteTable(string, string)-Methode	UL-Erw.: Ruft eine Datenbanktabelle in einem ULTable-Objekt zur direkten Bearbeitung ab.
ExecuteTable(string, string, CommandBehavior)-Methode	UL-Erw.: Ruft mit dem festgelegten Befehlsverhalten eine Datenbanktabelle zur direkten Bearbeitung ab.

ExecuteTable(string)-Methode

UL-Erw.: Ruft eine Datenbanktabelle in einem ULTable-Objekt zur direkten Bearbeitung ab.

Visual Basic-Syntax

```
Public Function ExecuteTable(ByVal tableName As String) As ULTable
```

C#-Syntax

```
public UTable ExecuteTable(string tableName)
```

Parameter

- **tableName** Der Name der Tabelle, die geöffnet werden soll

Rückgabe

Die Tabelle als UTable-Objekt

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.
- **InvalidOperationException** Der Tabellenname *tableName* ist ungültig.

Bemerkungen

Die Tabelle wird mit dem Primärschlüssel der Tabelle geöffnet (sortiert).

Diese Methode ist ein Shortcut für die ULCommand.ExecuteTable-Methode, der kein ULCommand-Objekt erfordert. Sie wird bereitgestellt, um Benutzer bei der Portierung aus früheren Versionen von UltraLite.NET zu unterstützen (und ersetzt die Methoden iAnywhere.UltraLite.Connection.GetTable und iAnywhere.UltraLite.Table.Open).

Siehe auch

- [ULConnection.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 139](#)
- [ULTable-Klasse \[UltraLite.NET\] auf Seite 414](#)
- [ULCommand.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 100](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)

Beispiel

Der folgende Code öffnet die Tabelle MyTable mit dem Primärschlüssel der Tabelle. Es wird davon ausgegangen, dass die ULConnection-Instanz "conn" geöffnet ist.

```
' Visual Basic
Dim t As UTable = conn.ExecuteTable("MyTable")

' The line above is equivalent to
' Dim cmd As ULCommand = conn.CreateCommand()
' cmd.CommandText = "MyTable"
' cmd.CommandType = CommandType.TableDirect
' Dim t As UTable = cmd.ExecuteTable()
' cmd.Dispose()
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULTable t = conn.ExecuteTable("MyTable");

// The line above is equivalent to
// UTable t;
// using(ULCommand cmd = conn.CreateCommand())
// {
//     cmd.CommandText = "MyTable";
```

```
//      cmd.CommandType = CommandType.TableDirect;
//      t = cmd.ExecuteTable();
//  }
```

ExecuteTable(string, string)-Methode

UL-Erw.: Ruft eine Datenbanktabelle in einem ULTable-Objekt zur direkten Bearbeitung ab.

Visual Basic-Syntax

```
Public Function ExecuteTable(
    ByVal tableName As String,
    ByVal indexName As String
) As ULTable
```

C#-Syntax

```
public ULTable ExecuteTable(string tableName, string indexName)
```

Parameter

- **tableName** Der Name der Tabelle, die geöffnet werden soll
- **indexName** Der Name des Indexes, mit dem die Tabelle geöffnet (sortiert) werden soll.

Rückgabe

Die Tabelle als ULTable-Objekt

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.
- **InvalidOperationException** Der Tabellename *tableName* ist ungültig.

Bemerkungen

Die Tabelle wird mit dem angegebenen Index geöffnet (sortiert).

Diese Methode ist ein Shortcut für die ULCommand.ExecuteTable-Methode, der kein ULCommand-Objekt erfordert. Sie wird bereitgestellt, um Benutzer bei der Portierung aus früheren Versionen von UltraLite.NET zu unterstützen (und ersetzt die Methoden iAnywhere.UltraLite.Connection.GetTable und iAnywhere.UltraLite.Table.Open).

Siehe auch

- [ULConnection.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 139](#)
- [ULTable-Klasse \[UltraLite.NET\] auf Seite 414](#)
- [ULCommand.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 100](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)

Beispiel

Der folgende Code öffnet die Tabelle MyTable mit dem Index MyIndex. Es wird davon ausgegangen, dass das ULConnection-Objekt "conn" geöffnet ist.

```
' Visual Basic
Dim t As ULTable = conn.ExecuteTable("MyTable", "MyIndex")

' The line above is equivalent to
' Dim cmd As ULCommand = conn.CreateCommand()
' cmd.CommandText = "MyTable"
' cmd.IndexName = "MyIndex"
' cmd.CommandType = CommandType.TableDirect
' Dim t As ULTable = cmd.ExecuteTable()
' cmd.Dispose()
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULTable t = conn.ExecuteTable("MyTable", "MyIndex");

// The line above is equivalent to
// ULTable t;
// using(ULCommand cmd = conn.CreateCommand())
// {
//     cmd.CommandText = "MyTable";
//     cmd.IndexName = "MyIndex";
//     cmd.CommandType = CommandType.TableDirect;
//     t = cmd.ExecuteTable();
// }
```

ExecuteTable(string, string, CommandBehavior)-Methode

UL-Erw.: Ruft mit dem festgelegten Befehlsverhalten eine Datenbanktabelle zur direkten Bearbeitung ab.

Visual Basic-Syntax

```
Public Function ExecuteTable(
    ByVal tableName As String,
    ByVal indexName As String,
    ByVal cmdBehavior As CommandBehavior
) As ULTable
```

C#-Syntax

```
public ULTable ExecuteTable(
    string tableName,
    string indexName,
    CommandBehavior cmdBehavior
)
```

Parameter

- **tableName** Der Name der Tabelle, die geöffnet werden soll
- **indexName** Der Name des Indexes, mit dem die Tabelle geöffnet (sortiert) werden soll.
- **cmdBehavior** Eine bitweise Kombination von System.Data.CommandBehavior-Parametern, die das Ergebnis der Abfrage und ihre Auswirkung auf die Verbindung beschreibt. UltraLite.NET respektiert nur die Parameter System.Data.CommandBehavior.Default, System.Data.CommandBehavior.CloseConnection und System.Data.CommandBehavior.SchemaOnly.

Rückgabe

Die Tabelle als ULTable-Objekt

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.
- **InvalidOperationException** Der Tabellenname *tableName* ist ungültig.

Bemerkungen

Die Tabelle wird mit dem angegebenen Index geöffnet (sortiert).

Diese Methode ist ein Shortcut für die ULCommand.ExecuteTable(System.Data.CommandBehavior)-Methode, der kein ULCommand-Objekt erfordert. Sie wird bereitgestellt, um Benutzer bei der Portierung aus früheren Versionen von UltraLite.NET zu unterstützen (und ersetzt die Methoden iAnywhere.UltraLite.Connection.GetTable und iAnywhere.UltraLite.Table.Open).

Siehe auch

- [ULConnection.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 139](#)
- [ULTable-Klasse \[UltraLite.NET\] auf Seite 414](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [System.Data.CommandBehavior](#)

Beispiel

Der folgende Code öffnet die Tabelle MyTable mit dem Index MyIndex. Es wird davon ausgegangen, dass das ULConnection-Objekt "conn" geöffnet ist.

```
' Visual Basic
Dim t As ULTable = conn.ExecuteTable( _
    "MyTable", "MyIndex", CommandBehavior.Default _
)

' The line above is equivalent to the following code:
' Dim cmd As ULCommand = conn.CreateCommand()
' cmd.CommandText = "MyTable"
' cmd.IndexName = "MyIndex"
' cmd.CommandType = CommandType.TableDirect
' Dim t As ULTable = cmd.ExecuteTable(CommandBehavior.Default)
' cmd.Dispose()
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULTable t = conn.ExecuteTable(
    "MyTable", "MyIndex", CommandBehavior.Default
);

// The line above is equivalent to the following code:
// ULTable t;
// using(ULCommand cmd = conn.CreateCommand())
// {
//     cmd.CommandText = "MyTable";
//     cmd.IndexName = "MyIndex";
//     cmd.CommandType = CommandType.TableDirect;
//     t = cmd.ExecuteTable(CommandBehavior.Default);
// }
```

GetLastDownloadTime-Methode

UL-Erw.: Gibt die Uhrzeit des letzten Downloads der angegebenen Publikation zurück.

Visual Basic-Syntax

```
Public Function GetLastDownloadTime(ByVal publication As String) As Date
```

C#-Syntax

```
public DateTime GetLastDownloadTime(string publication)
```

Parameter

- **publication** Die zu prüfende Publikation.

Rückgabe

Der Zeitstempel des letzten Downloads. Wenn für die Publikation die SYNC_ALL_DB-Konstante verwendet wird, gibt dieses Objekt den Zeitpunkt des letzten Downloads der gesamten Datenbank zurück.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Der Parameter *publication* steht für den Namen der zu prüfenden Publikation.

Siehe auch

- [ULConnection.ResetLastDownloadTime-Methode \[UltraLite.NET\] auf Seite 152](#)
- [ULConnection.SYNC_ALL_DB-Feld \[UltraLite.NET\] auf Seite 168](#)

GetNewUUID-Methode

UL-Erw.: Generiert eine neue UUID (System.Guid).

Visual Basic-Syntax

```
Public Function GetNewUUID() As Guid
```

C#-Syntax

```
public Guid GetNewUUID()
```

Rückgabe

Eine neue UUID als System.Guid

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Diese Methode wird hier bereitgestellt, da sie im .NET Compact Framework nicht enthalten ist.

Siehe auch

- [System.Guid](#)

GetNotification-Methode

UL Ext: Blockiert für eine Benachrichtigung oder einen Timeout.

Visual Basic-Syntax

```
Public Function GetNotification(  
    ByVal queueName As String,  
    ByVal wait_ms As Integer  
) As String
```

C#-Syntax

```
public string GetNotification(string queueName, int wait_ms)
```

Parameter

- **queueName** Der Name der Warteschlange.
- **wait_ms** Die Wartezeit in Millisekunden. Verwenden Sie System.Threading.Timeout.Infinite (-1) für einen unbegrenzten Wartezeitraum.

Rückgabe

Gibt Null zurück, wenn die Wartezeit abgelaufen ist oder abgebrochen wurde. Andernfalls wird der Ereignisname zurückgegeben.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Diese Methode liest eine Ereignisbenachrichtigung. Dieser Aufruf wird blockiert, bis eine Benachrichtigung empfangen wird oder die angegebene Wartezeit abgelaufen ist. Um einen unbegrenzten Wartezeitraum festzulegen, übergeben Sie für den wait_ms-Parameter den Wert System.Threading.Timeout.Infinite. Um den Wartezustand zu beenden, senden Sie eine weitere Benachrichtigung an die angegebene Warteschlange oder verwenden Sie die CancelGetNotification-Methode. Verwenden Sie nach dem Lesen einer Benachrichtigung die ReadNotificationParameter-Methode, um zusätzliche Parameter abzurufen.

Siehe auch

- [ULConnection.SendNotification-Methode \[UltraLite.NET\] auf Seite 153](#)
- [ULConnection.GetNotificationParameter-Methode \[UltraLite.NET\] auf Seite 146](#)
- [ULConnection.CancelGetNotification-Methode \[UltraLite.NET\] auf Seite 132](#)
- [ULException-Klasse \[UltraLite.NET\] auf Seite 275](#)
- [System.Threading.Timeout](#)

GetNotificationParameter-Methode

UL Ext: Ruft den Wert eines Parameters für ein Ereignis ab, das gerade von der GetNotification-Methode gelesen wurde.

Visual Basic-Syntax

```
Public Function GetNotificationParameter(  
    ByVal queueName As String,  
    ByVal parameterName As String  
) As String
```

C#-Syntax

```
public string GetNotificationParameter(  
    string queueName ,  
    string parameterName  
)
```

Parameter

- **queueName** Der Name der Warteschlange.
- **parameterName** Der Name des Parameters, dessen Wert zurückgegeben werden soll.

Rückgabe

Gibt den Parameterwert zurück, wenn der Parameter gefunden wurde. Andernfalls wird NULL zurückgegeben.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Diese Methode ruft den Wert eines Parameters für die Ereignisbenachrichtigung ab, die soeben von der ULGetNotification-Methode gelesen wurde. Es sind nur die Parameter aus der zuletzt gelesenen Benachrichtigung in der angegebenen Warteschlange verfügbar.

Siehe auch

- [ULConnection.GetNotification-Methode \[UltraLite.NET\] auf Seite 145](#)
- [ULException-Klasse \[UltraLite.NET\] auf Seite 275](#)

GetSchema-Methode

Gibt die Liste der unterstützten Schemasammlungen zurück.

Überladungsliste

Name	Beschreibung
GetSchema()-Methode	Gibt die Liste der unterstützten Schemasammlungen zurück.
GetSchema(String)-Methode	Gibt Informationen für die angegebene Metadatensammlung für dieses ULConnection-Objekt zurück.
GetSchema(String, String[])-Methode	Gibt Schemainformationen für die Datenquelle dieses ULConnection-Objekts zurück und verwendet ggf. die angegebene Zeichenfolge für den Schemanamen und das angegebene Zeichenfolgen-Array für die Einschränkungswerte.

GetSchema()-Methode

Gibt die Liste der unterstützten Schemasammlungen zurück.

Visual Basic-Syntax

```
Public Overrides Function GetSchema() As DataTable
```

C#-Syntax

```
public override DataTable GetSchema()
```

GetSchema(String)-Methode

Gibt Informationen für die angegebene Metadatensammlung für dieses ULConnection-Objekt zurück.

Visual Basic-Syntax

```
Public Overrides Function GetSchema(  
    ByVal collection As String  
) As DataTable
```

C#-Syntax

```
public override DataTable GetSchema(string collection)
```

Parameter

- **collection** Der Name der Metadatensammlung. Wenn kein Name angegeben ist, wird der MetaDataCollections-Wert verwendet.

Siehe auch

- [ULConnection.GetSchema-Methode \[UltraLite.NET\] auf Seite 147](#)
- [ULConnection-Klasse \[UltraLite.NET\] auf Seite 121](#)

GetSchema(String, String[])-Methode

Gibt Schemainformationen für die Datenquelle dieses ULConnection-Objekts zurück und verwendet ggf. die angegebene Zeichenfolge für den Schemanamen und das angegebene Zeichenfolgen-Array für die Einschränkungswerte.

Visual Basic-Syntax

```
Public Overrides Function GetSchema(  
    ByVal collection As String,  
    ByVal restrictions As String()  
) As DataTable
```

C#-Syntax

```
public override DataTable GetSchema(  
    string collection,  
    string[] restrictions  
)
```

Parameter

- **collection** Der Name der Metadatenammlung. Wenn kein Name angegeben ist, wird der MetaDataCollections-Wert verwendet.
- **restrictions** Eine Gruppe von Einschränkungswerten für das angeforderte Schema.

Rückgabe

Ein DataTable-Objekt, das Schemainformationen enthält.

Bemerkungen

Diese Methode wird zur Abfrage der Datenbank für verschiedene Metadaten verwendet. Jeder Metadaten-Typ erhält einen Sammlungsnamen, der für den Empfang der Daten übergeben werden muss. Der Standard-Sammlungsname ist MetaDataCollections.

Sie können den .NET-Datenprovider abfragen, um die Liste der unterstützten Schemasammlungen zu erhalten, indem Sie die Methode GetSchema ohne Argumente oder mit dem Schemasammlungsnamen **MetaDataCollections** übergeben. Gibt eine DataTable-Tabelle mit einer Liste der unterstützten Schemasammlungen (CollectionName), der Anzahl der Einschränkungen, die sie jeweils unterstützen (NumberOfRestrictions) und der Anzahl der ID-Teile, die sie verwenden (NumberOfIdentifierParts), zurück.

Sammlung	Metadaten
Columns	Gibt Informationen über alle Spalten in der Datenbank zurück.

Sammlung	Metadaten
DataSourceInformation	Gibt Informationen über einen Datenbankprovider zurück
DataTypes	Gibt eine Liste der unterstützten Datentypen zurück.
ForeignKeys	Gibt Informationen über alle Fremdschlüssel in der Datenbank zurück.
IndexColumns	Gibt Informationen über alle Indexspalten in der Datenbank zurück.
Indexes	Gibt Informationen über alle Indizes in der Datenbank zurück.
MetaDataCollections	Gibt eine Liste aller Sammlungsnamen zurück.
Publications	Gibt Informationen über alle Publikationen in der Datenbank zurück.
ReservedWords	Gibt eine Liste der von UltraLite verwendeten reservierten Wörter zurück
Restrictions	Gibt Informationen über die in GetSchema verwendeten Einschränkungen zurück.
Tables	Gibt Informationen über alle Tabellen in der Datenbank zurück.

Diese Sammlungsnamen sind auch als schreibgeschützte Eigenschaften in der ULMetaDataCollectionNames-Klasse verfügbar.

Die zurückgegebenen Ergebnisse können durch Angeben eines Arrays von Einschränkungen im Aufruf der GetSchema-Methode gefiltert werden.

Die für jede Sammlung verfügbaren Einschränkungen können mit folgendem Aufruf abgefragt werden:

```
GetSchema( "Restrictions" )
```

Wenn die Sammlung vier Einschränkungen erfordert, muss der Restrictions-Parameter NULL oder eine Zeichenfolge von 4 Werten sein.

Zum Filtern einer bestimmten Einschränkung positionieren Sie die Zeichenfolge, anhand der gefiltert werden soll, an die betreffende Position im Array und geben für alle nicht verwendeten Stellen NULL an. Die Sammlung "Tables" hat z. B. 3 Einschränkungen: Table, TableType, SyncType.

So filtern Sie die Sammlung "Table" anhand des Tabellennamens:

GetSchema("Tables", new string[] { "my_table", NULL, NULL }) Gibt Informationen über alle Tabellen mit dem Namen my_table zurück.

GetSchema("Tables", new string[] { NULL, "User", NULL }) Gibt Informationen über alle Benutzertabellen zurück.

Siehe auch

- [ULConnection-Klasse \[UltraLite.NET\] auf Seite 121](#)
- [ULMetaDataCollectionNames-Klasse \[UltraLite.NET\] auf Seite 310](#)

GrantConnectTo-Methode

UL-Erw.: Erteilt einer Benutzer-ID mit einem bestimmten Kennwort die Zugriffsberechtigung für eine UltraLite-Datenbank.

Visual Basic-Syntax

```
Public Sub GrantConnectTo(ByVal uid As String, ByVal pwd As String)
```

C#-Syntax

```
public void GrantConnectTo(string uid, string pwd)
```

Parameter

- **uid** Die Benutzer-ID, die die Zugriffsberechtigung für die Datenbank erhalten soll.
- **pwd** Das Kennwort, das der Benutzer-ID zugeordnet werden soll.

Bemerkungen

Wenn eine bestehende Benutzer-ID angegeben wird, aktualisiert diese Funktion das Kennwort für den Benutzer. UltraLite unterstützt maximal 4 Benutzer. Diese Methode ist nur aktiviert, wenn die Benutzer-Authentifizierung beim Öffnen der Verbindung aktiviert wurde.

Siehe auch

- [ULConnectionParms.UserID-Eigenschaft \[UltraLite.NET\] auf Seite 178](#)
- [ULConnectionParms.Password-Eigenschaft \[UltraLite.NET\] auf Seite 177](#)
- [ULConnection.ConnectionString-Eigenschaft \[UltraLite.NET\] auf Seite 160](#)

Open-Methode

Öffnet eine Verbindung mit einer Datenbank unter Verwendung der vorher definierten Verbindungszeichenfolge.

Visual Basic-Syntax

```
Public Overrides Sub Open()
```

C#-Syntax

```
public override void Open()
```

Ausnahmen

- **InvalidOperationException** Die Verbindung ist bereits geöffnet oder die Verbindungszeichenfolge ist in der ULConnection.ConnectionString-Eigenschaft nicht angegeben.

- **ULException-Klasse** Beim Versuch die Datenbank zu öffnen, ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Verbindung muss explizit geschlossen oder entfernt werden, wenn sie nicht mehr benötigt wird.

Siehe auch

- [ULConnection.ConnectionString-Eigenschaft \[UltraLite.NET\] auf Seite 160](#)
- [ULConnection.State-Eigenschaft \[UltraLite.NET\] auf Seite 165](#)

RegisterForEvent-Methode

UL Ext: Registriert eine Warteschlange, um Ereignisse von einem Objekt zu beziehen.

Visual Basic-Syntax

```
Public Sub RegisterForEvent(  
    ByVal eventName As String,  
    ByVal objectName As String,  
    ByVal queueName As String,  
    ByVal registerNotUnReg As Boolean  
)
```

C#-Syntax

```
public void RegisterForEvent(  
    string eventName,  
    string objectName,  
    string queueName,  
    bool registerNotUnReg  
)
```

Parameter

- **eventName** Der Ereignisname
- **objectName** Der Name des Objekts, für das das Ereignis gilt. Z.B. ein Tabellename.
- **queueName** Der zu verwendende Ereigniswarteschlangenname.
- **registerNotUnReg** TRUE, um die Warteschlange zu registrieren, FALSE, um die Registrierung aufzuheben.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Diese Methode registriert eine Warteschlange, um Benachrichtigungen über ein Ereignis zu empfangen. Der Standardwert für die Verbindungswarteschlange ist implizit gegeben. Wenn kein Warteschlangenname angegeben wird, wird sie damit erstellt. Bestimmte Systemereignisse lassen die Angabe eines Objektnamens zu, für den das Ereignis gilt. Das TableModified-Ereignis z.B. kann den Tabellennamen spezifizieren. Im Gegensatz zur SendNotification-Methode empfängt nur die spezifische

registrierte Warteschlange Benachrichtigungen für das Ereignis, andere Warteschlangen mit demselben Namen auf anderen Verbindungen erhalten keine (es sei denn, sie sind ebenfalls explizit registriert). Diese Methode gibt einen Fehler aus, wenn die Warteschlange oder das Ereignis nicht existieren.

Siehe auch

- [ULConnection.DeclareEvent-Methode \[UltraLite.NET\] auf Seite 137](#)
- [ULConnection.CreateNotificationQueue-Methode \[UltraLite.NET\] auf Seite 137](#)
- [ULException-Klasse \[UltraLite.NET\] auf Seite 275](#)

ResetLastDownloadTime-Methode

UL-Erw.: Setzt den Zeitpunkt des letzten Downloads zurück.

Visual Basic-Syntax

```
Public Sub ResetLastDownloadTime(ByVal pubs As String)
```

C#-Syntax

```
public void ResetLastDownloadTime(string pubs)
```

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULConnection.GetLastDownloadTime-Methode \[UltraLite.NET\] auf Seite 144](#)

RevokeConnectFrom-Methode

UL-Erw.: Entzieht der angegebenen Benutzer-ID die Zugriffsberechtigung auf eine UltraLite-Datenbank.

Visual Basic-Syntax

```
Public Sub RevokeConnectFrom(ByVal uid As String)
```

C#-Syntax

```
public void RevokeConnectFrom(string uid)
```

Parameter

- **uid** Die Benutzer-ID, der die Zugriffsberechtigung auf die Datenbank entzogen wird

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULConnection.GrantConnectTo-Methode \[UltraLite.NET\] auf Seite 150](#)

RollbackPartialDownload-Methode

UL-Erw.: Setzt noch nicht festgeschriebene Änderungen aus einem Teil-Download in der Datenbank zurück.

Visual Basic-Syntax

```
Public Sub RollbackPartialDownload()
```

C#-Syntax

```
public void RollbackPartialDownload()
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULSyncParms.KeepPartialDownload-Eigenschaft \[UltraLite.NET\] auf Seite 392](#)
- [ULSyncParms.ResumePartialDownload-Eigenschaft \[UltraLite.NET\] auf Seite 395](#)

SendNotification-Methode

UL Ext: Sendet eine Benachrichtigung an passende Warteschlangen.

Visual Basic-Syntax

```
Public Function SendNotification(  
    ByVal queueName As String,  
    ByVal eventName As String,  
    ByVal parameters As String  
) As Integer
```

C#-Syntax

```
public int SendNotification(  
    string queueName,  
    string eventName,  
    string parameters  
)
```

Parameter

- **queueName** Der zu verwendende Ereigniswarteschlangenname.
- **eventName** Der Ereignisname
- **parameters** Zu übergebende Parameter.

Rückgabe

Anzahl der gesendeten Benachrichtigungen (die Anzahl der übereinstimmenden Warteschlangen).

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Gibt die Anzahl passender Warteschlangen zurück.

Diese Methode sendet eine Benachrichtigung an alle Warteschlangen, die dem angegebenen Namen entsprechen (einschließlich etwaiger Warteschlangen auf der aktuellen Verbindung). Dieser Aufruf bewirkt keine Blockierung. Verwenden Sie den speziellen Warteschlangennamen "*", um Benachrichtigungen an alle Warteschlangen zu senden.

Siehe auch

- [ULConnection.DeclareEvent-Methode \[UltraLite.NET\] auf Seite 137](#)
- [ULConnection.RegisterForEvent-Methode \[UltraLite.NET\] auf Seite 151](#)
- [ULException-Klasse \[UltraLite.NET\] auf Seite 275](#)

SetSyncListener-Methode

Gibt das Listener-Objekt an, mit dem Synchronisationsnachrichten verarbeitet werden.

Visual Basic-Syntax

```
Public Sub SetSyncListener(ByVal listener As ULSyncProgressListener)
```

C#-Syntax

```
public void SetSyncListener(ULSyncProgressListener listener)
```

Parameter

- **listener** Das ULSyncProgressListener-Objekt, das die SyncProgressed-Methode implementiert, die für Synchronisationsnachrichten auf dieser Verbindung aufgerufen wird.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Wenn die SQL-Anweisung SYNCHRONIZE *profileName* ausgeführt wird, werden deren Meldungen zum Verarbeitungsfortschritt an ein syncListener-Objekt weitergeleitet, wenn dieser nicht NULL ("Nothing" in Visual Basic) ist.

Um den Listener zu entfernen, übergeben Sie in einem Aufruf der SetSyncListener-Methode eine Nullreferenz.

Siehe auch

- [ULSyncProgressListener-Schnittstelle \[UltraLite.NET\] auf Seite 409](#)

StartSynchronizationDelete-Methode

UL-Erw.: Markiert alle nachfolgenden Löschungen, die von dieser Verbindung für die Synchronisation vorgenommen wurden.

Visual Basic-Syntax

```
Public Sub StartSynchronizationDelete()
```

C#-Syntax

```
public void StartSynchronizationDelete()
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Wenn diese Methode aufgerufen wird, werden alle Löschvorgänge erneut synchronisiert. Dadurch werden die aus der UltraLite-Datenbank gelöschten Zeilen auch aus der konsolidierten Datenbank entfernt.

Siehe auch

- [ULConnection.StopSynchronizationDelete-Methode \[UltraLite.NET\] auf Seite 155](#)
- [ULTable.Truncate-Methode \[UltraLite.NET\] auf Seite 436](#)

StopSynchronizationDelete-Methode

UL-Erw.: Verhindert, dass Löschvorgänge synchronisiert werden.

Visual Basic-Syntax

```
Public Sub StopSynchronizationDelete()
```

C#-Syntax

```
public void StopSynchronizationDelete()
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Diese Methode ist nützlich, um alte Informationen aus Platzgründen aus einer UltraLite-Datenbank zu löschen, während diese Informationen in der konsolidierten Datenbank erhalten bleiben.

Siehe auch

- [ULConnection.StartSynchronizationDelete-Methode \[UltraLite.NET\] auf Seite 155](#)

Synchronize-Methode

UL-Erw.: Synchronisiert die Datenbank unter Verwendung des aktuellen `ULConnection.SyncParms`-Objekts.

Überladungsliste

Name	Beschreibung
Synchronize()-Methode	UL-Erw.: Synchronisiert die Datenbank unter Verwendung des aktuellen <code>ULConnection.SyncParms</code> -Objekts.
Synchronize(ULSyncProgressListener)-Methode	UL-Erw.: Synchronisiert die Datenbank unter Verwendung des aktuellen <code>ULConnection.SyncParms</code> -Objekts, wobei die Fortschrittsereignisse an den angegebenen Listener gesendet werden.

Synchronize()-Methode

UL-Erw.: Synchronisiert die Datenbank unter Verwendung des aktuellen `ULConnection.SyncParms`-Objekts.

Visual Basic-Syntax

```
Public Sub Synchronize()
```

C#-Syntax

```
public void Synchronize()
```

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Ein ausführlicher Ergebnisstatusbericht wird in der `ULConnection.SyncResult`-Eigenschaft dieser Verbindung ausgegeben.

Siehe auch

- [ULConnection.Synchronize-Methode \[UltraLite.NET\] auf Seite 156](#)
- [ULConnection.SyncParms-Eigenschaft \[UltraLite.NET\] auf Seite 165](#)
- [ULConnection.SyncResult-Eigenschaft \[UltraLite.NET\] auf Seite 166](#)

Synchronize(ULSyncProgressListener)-Methode

UL-Erw.: Synchronisiert die Datenbank unter Verwendung des aktuellen `ULConnection.SyncParms`-Objekts, wobei die Fortschrittsereignisse an den angegebenen Listener gesendet werden.

Visual Basic-Syntax

```
Public Sub Synchronize(ByVal listener As ULSyncProgressListener)
```

C#-Syntax

```
public void Synchronize(ULSyncProgressListener listener)
```

Parameter

- **listener** Das Objekt, das Ereignisse zum Synchronisationsfortschritt empfängt.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Ein während der Synchronisation aufgetretener Fehler wird als `ULSyncProgressState.STATE_ERROR`-Ereignis gesendet und dann als `ULException` ausgegeben.

Ein ausführlicher Ergebnisstatusbericht wird in der `ULConnection.SyncResult`-Eigenschaft dieser Verbindung ausgegeben.

Siehe auch

- [ULSyncProgressListener-Schnittstelle \[UltraLite.NET\] auf Seite 409](#)
- [ULConnection.Synchronize-Methode \[UltraLite.NET\] auf Seite 156](#)
- [ULConnection.SyncParms-Eigenschaft \[UltraLite.NET\] auf Seite 165](#)
- [ULException-Klasse \[UltraLite.NET\] auf Seite 275](#)
- [ULConnection.SyncResult-Eigenschaft \[UltraLite.NET\] auf Seite 166](#)

TriggerEvent-Methode

UL Ext: Löst ein Ereignis aus.

Visual Basic-Syntax

```
Public Function TriggerEvent(  
    ByVal eventName As String,  
    ByVal parameters As String  
) As Integer
```

C#-Syntax

```
public int TriggerEvent(string eventName, string parameters)
```

Parameter

- **eventName** Der Name des Ereignisses, das ausgelöst werden soll.
- **parameters** Zu übergebende Parameter.

Rückgabe

Die Anzahl der gesendeten Ereignisbenachrichtigungen.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Gibt die Anzahl der gesendeten Benachrichtigungen zurück.

Diese Methode löst ein Ereignis aus (und sendet eine Benachrichtigung an alle registrierten Warteschlangen).

Siehe auch

- [ULConnection.DeclareEvent-Methode \[UltraLite.NET\] auf Seite 137](#)
- [ULConnection.RegisterForEvent-Methode \[UltraLite.NET\] auf Seite 151](#)
- [ULException-Klasse \[UltraLite.NET\] auf Seite 275](#)

ValidateDatabase-Methode

UL-Erw.: Führt die Validierung der aktuellen Datenbank durch.

Überladungsliste

Name	Beschreibung
ValidateDatabase(ULDBValid)-Methode	UL Ext: Führt die Validierung der aktuellen Datenbank durch.
ValidateDatabase(ULDBValid, string)-Methode	UL Ext: Führt die Validierung der aktuellen Datenbank durch.

ValidateDatabase(ULDBValid)-Methode

UL Ext: Führt die Validierung der aktuellen Datenbank durch.

Visual Basic-Syntax

```
Public Sub ValidateDatabase(ByVal how As ULDBValid)
```

C#-Syntax

```
public void ValidateDatabase(ULDBValid how)
```

Parameter

- **how** Beschreibt, wie die Validierung der Datenbank erfolgen soll.

Ausnahmen

- **ULException-Klasse** Ein SQLE_CORRUPT_ULTRALITE_INDEX- oder SQLE_CORRUPT_ULTRALITE_DATABASE-Fehler kann auftreten, wenn die Datenbank beschädigt ist.

Siehe auch

- [ULDatabaseManager.ValidateDatabase-Methode \[UltraLite.NET\] auf Seite 228](#)
- [ULDBValid-Enumeration \[UltraLite.NET\] auf Seite 458](#)
- [„Indexvalidierung für Tabelle %1, Index %2 ist fehlgeschlagen, Code: %3“ \[Fehlermeldungen\]](#)
- [„Validierung der Datenbankseite ist fehlgeschlagen mit Code: %1“ \[Fehlermeldungen\]](#)

Beispiel

Der folgende Code validiert die aktuelle Datenbank.

```
' Visual Basic
conn.ValidateDatabase( iAnywhere.Data.UltraLite.ULVF_INDEX )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
conn.ValidateDatabase( iAnywhere.Data.UltraLite.ULVF_INDEX )
```

ValidateDatabase(ULDBValid, string)-Methode

UL Ext: Führt die Validierung der aktuellen Datenbank durch.

Visual Basic-Syntax

```
Public Sub ValidateDatabase(
    ByVal how As ULDBValid,
    ByVal tableName As String
)
```

C#-Syntax

```
public void ValidateDatabase(ULDBValid how, string tableName)
```

Parameter

- **how** Beschreibt, wie die Validierung der Datenbank erfolgen soll.
- **tableName** Wenn NULL ("Nothing" in Visual Basic), wird die gesamte Datenbank validiert, sonst nur die benannte Tabelle.

Ausnahmen

- **ULException-Klasse** Ein SQLE_CORRUPT_ULTRALITE_INDEX- oder SQLE_CORRUPT_ULTRALITE_DATABASE-Fehler kann auftreten, wenn die Datenbank beschädigt ist.

Siehe auch

- [ULDatabaseManager.ValidateDatabase-Methode \[UltraLite.NET\] auf Seite 228](#)
- [ULDBValid-Enumeration \[UltraLite.NET\] auf Seite 458](#)
- [„Indexvalidierung für Tabelle %1, Index %2 ist fehlgeschlagen, Code: %3“ \[Fehlermeldungen\]](#)
- [„Validierung der Datenbankseite ist fehlgeschlagen mit Code: %1“ \[Fehlermeldungen\]](#)

Beispiel

Der folgende Code validiert die aktuelle Datenbank.

```
' Visual Basic  
conn.ValidateDatabase( iAnywhere.Data.UltraLite.ULVF_INDEX, Nothing )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#  
conn.ValidateDatabase( iAnywhere.Data.UltraLite.ULVF_INDEX, null )
```

ConnectionString-Eigenschaft

Legt die Parameter zum Öffnen einer Verbindung zu einer UltraLite.NET-Datenbank fest.

Visual Basic-Syntax

```
Public Overrides Property ConnectionString As String
```

C#-Syntax

```
public override string ConnectionString {get;set;}
```

Ausnahmen

- **InvalidOperationException** Der Wert kann nicht festgelegt werden, während die Verbindung geöffnet ist.
- **ArgumentException** Die bereitgestellte Verbindungszeichenfolge ist ungültig.

Bemerkungen

Die Verbindungszeichenfolge kann mit einem ULConnectionParms-Objekt bereitgestellt werden.

Die Parameter zum Öffnen dieser Verbindung sollten eine durch Semikola getrennten Liste von Schlüsselwort-Wert-Paaren sein. Standardwert ist eine leere Zeichenfolge (eine ungültige Verbindungszeichenfolge).

UL-Erw.: Die von UltraLite.NET verwendeten Parameter sind für UltraLite.NET-Datenbanken spezifisch. Daher ist die Verbindungszeichenfolge nicht mit SQL Anywhere-Verbindungszeichenfolgen kompatibel.

Parameterwerte können zwischen Apostrophe oder Anführungszeichen gesetzt werden, sofern der betreffende Text keine Anführungszeichen desselben Typs enthält. Werte müssen zwischen Anführungszeichen oder Apostrophe gesetzt werden, wenn sie Semikola enthalten, mit einem Apostroph oder Anführungszeichen beginnen oder führende oder nachgestellte Leerzeichen erfordern.

Wenn Sie Parameterwerte nicht zwischen Anführungszeichen oder Apostrophe setzen, müssen Sie sicherstellen, dass sie keine Semikola enthalten und nicht mit einem Apostroph oder einem Anführungszeichen beginnen. Führende und nachgestellte Leerzeichen in Werten werden ignoriert.

Standardmäßig werden Verbindungen mit UID=DBA und PWD=sql geöffnet. Um die Datenbank sicherer zu machen, ändern Sie das DBA-Kennwort oder erstellen Sie neue Benutzer (mit der `GrantConnectTo`-Methode) und entfernen Sie den DBA-Benutzer (mit `RevokeConnectFrom`).

Siehe auch

- [ULConnection.Open-Methode \[UltraLite.NET\] auf Seite 150](#)
- [ULConnectionParms-Klasse \[UltraLite.NET\] auf Seite 169](#)
- [ULConnection.GrantConnectTo-Methode \[UltraLite.NET\] auf Seite 150](#)
- „UltraLite-Verbindungsparameter“ [*UltraLite - Datenbankverwaltung*]

Beispiel

Der folgende Code erstellt und öffnet eine Verbindung zur vorhandenen Datenbank \UltraLite\MyDatabase.udb auf einem Windows Mobile-Gerät.

```
' Visual Basic
Dim openParms As ULConnectionParms = New ULConnectionParms
openParms.DatabaseOnDevice = "\UltraLite\MyDatabase.udb"
Dim conn As ULConnection = New ULConnection
conn.ConnectionString = openParms.ToString()
conn.Open()
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnDevice = @"\UltraLite\MyDatabase.udb";
ULConnection conn = new ULConnection();
conn.ConnectionString = openParms.ToString();
conn.Open();
```

ConnectionTimeout-Eigenschaft

Diese Funktion wird von UltraLite.NET nicht unterstützt.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property ConnectionTimeout As Integer
```

C#-Syntax

```
public override int ConnectionTimeout {get;}
```

Ausnahmen

- [ULException-Klasse](#) Die Festlegung des Werts wird in UltraLite.NET nicht unterstützt.

Bemerkungen

Der Wert ist stets Null.

Database-Eigenschaft

Gibt den Namen der Datenbank zurück, für die die Verbindung geöffnet wird.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property Database As String
```

C#-Syntax

```
public override string Database {get;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen der Datenbank enthält

Auf Windows Mobile-Geräten durchsucht das ULConnection-Objekt die Verbindungszeichenfolge in dieser Reihenfolge: dbn, ce_file.

Auf Desktopcomputern durchsucht das ULConnection-Objekt die Verbindungszeichenfolge in dieser Reihenfolge: dbn, nt_file.

DatabaseID-Eigenschaft

UL-Erw.: Legt den Datenbank-ID-Wert fest, der für global autoincrement-Spalten verwendet wird.

Visual Basic-Syntax

```
Public Property DatabaseID As Long
```

C#-Syntax

```
public long DatabaseID {get;set;}
```

Ausnahmen

- **ULException-Klasse** Die angegebene neue Datenbank-ID ist ungültig.

Bemerkungen

Der Wert der Datenbank-ID der aktuellen Datenbankverbindung

Der Wert der Datenbank-ID muss im Bereich [0, System.UInt32.MaxValue] liegen. Mit dem Wert ULConnection.INVALID_DATABASE_ID angezeigt, dass die Datenbank-ID für die aktuelle Datenbank nicht festgelegt wurde.

Siehe auch

- [ULDatabaseSchema.GetDatabaseProperty-Methode \[UltraLite.NET\] auf Seite 230](#)
- [ULDatabaseSchema.SetDatabaseOption-Methode \[UltraLite.NET\] auf Seite 234](#)
- [ULConnection.INVALID_DATABASE_ID-Feld \[UltraLite.NET\] auf Seite 168](#)
- [System.UInt32.MaxValue](#)

DataSource-Eigenschaft

Diese Funktion wird von UltraLite.NET nicht unterstützt.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property DataSource As String
```

C#-Syntax

```
public override string DataSource {get;}
```

Bemerkungen

Der Wert ist immer eine leere Zeichenfolge.

GlobalAutoIncrementUsage-Eigenschaft

UL-Erw.: Gibt den Prozentsatz noch verfügbarer Werte für global autoincrement an.

Visual Basic-Syntax

```
Public ReadOnly Property GlobalAutoIncrementUsage As Short
```

C#-Syntax

```
public short GlobalAutoIncrementUsage {get;}
```

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Der Prozentwert der verfügbaren globalen Werte für Autoincrement, die verwendet wurden. Dies ist eine Ganzzahl im Bereich [0-100] (inklusive).

Wenn der Prozentsatz sich dem Wert 100 nähert, sollte Ihre Anwendung mithilfe des `ULConnection.DatabaseID`-Werts einen neuen Wert für die globale Datenbank-ID setzen.

Siehe auch

- [ULDatabaseManager-Klasse \[UltraLite.NET\] auf Seite 222](#)
- [ULConnection.DatabaseID-Eigenschaft \[UltraLite.NET\] auf Seite 162](#)

LastIdentity-Eigenschaft

UL-Erw.: Gibt den zuletzt verwendeten Identity-Wert zurück.

Visual Basic-Syntax

```
Public ReadOnly Property LastIdentity As ULong
```

C#-Syntax

```
public ulong LastIdentity {get;}
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Der zuletzt verwendete Identitätswert als Datentyp "unsigned long"

Der zuletzt verwendete Identity-Wert. Diese Eigenschaft entspricht folgender SQL Anywhere-Anweisung:

```
SELECT @identity
```

Die LastIdentity-Eigenschaft ist besonders im Kontext von global autoincrement-Spalten nützlich.

Da Sie mit dieser Eigenschaft nur den zuletzt zugewiesenen Standardwert erfahren, sollten Sie den Wert möglichst bald nach der Ausführung der INSERT-Anweisung abfragen, um falsche Ergebnisse zu vermeiden.

Eine einzelne INSERT-Anweisung kann gelegentlich mehrere Spalten mit GLOBAL AUTOINCREMENT umfassen. In diesem Fall ist die LastIdentity-Eigenschaft einer der generierten Standardwerte, aber es gibt keine zuverlässige Methode, um zu ermitteln, aus welcher Spalte der Wert stammt. Aus diesem Grund sollten Sie Ihre Datenbank so planen und Ihre INSERT-Anweisungen so schreiben, dass diese Situation vermieden wird.

Schema-Eigenschaft

UL-Erw.: Stellt Zugriff auf das Schema der aktuellen Datenbank bereit, die dieser Verbindung zugeordnet ist.

Visual Basic-Syntax

```
Public ReadOnly Property Schema As ULDatabaseSchema
```

C#-Syntax

```
public ULDatabaseSchema Schema {get;}
```

Bemerkungen

Eine Referenz zum ULDatabaseSchema-Objekt, das das Schema der Datenbank repräsentiert, für die diese Verbindung geöffnet wurde.

Diese Eigenschaft ist nur gültig, während die Verbindung geöffnet ist.

Siehe auch

- [ULDatabaseSchema-Klasse \[UltraLite.NET\] auf Seite 229](#)

ServerVersion-Eigenschaft

Diese Funktion wird von UltraLite.NET nicht unterstützt.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property ServerVersion As String
```

C#-Syntax

```
public override string ServerVersion {get;}
```

Bemerkungen

Der Wert ist immer eine leere Zeichenfolge.

State-Eigenschaft

Gibt den aktuellen Status der Verbindung zurück.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property State As ConnectionState
```

C#-Syntax

```
public override ConnectionState State {get;}
```

Bemerkungen

Gibt System.Data.ConnectionState.Open zurück, wenn die Verbindung geöffnet ist, und System.Data.ConnectionState.Closed, wenn sie geschlossen ist.

Siehe auch

- [ULConnection.StateChange-Ereignis \[UltraLite.NET\] auf Seite 167](#)
- [System.Data.ConnectionState](#)

SyncParms-Eigenschaft

UL-Erw.: Gibt die Synchronisationseinstellungen für diese Verbindung an.

Visual Basic-Syntax

```
Public ReadOnly Property SyncParms As ULSyncParms
```

C#-Syntax

```
public ULSyncParms SyncParms {get;}
```

Bemerkungen

Eine Referenz zum ULSyncParms-Objekt, das die Parameter repräsentiert, die von dieser Verbindung für die Synchronisation verwendet werden. Änderungen der Parameter haben Auswirkungen auf die nächste Synchronisation, die über diese Verbindung ausgeführt wird.

Siehe auch

- [ULConnection.Synchronize-Methode \[UltraLite.NET\] auf Seite 156](#)
- [ULConnection.SyncResult-Eigenschaft \[UltraLite.NET\] auf Seite 166](#)
- [ULSyncParms-Klasse \[UltraLite.NET\] auf Seite 387](#)

SyncResult-Eigenschaft

UL-Erw.: Gibt die Ergebnisse der letzten Synchronisation für diese Verbindung zurück.

Visual Basic-Syntax

```
Public ReadOnly Property SyncResult As ULSyncResult
```

C#-Syntax

```
public ULSyncResult SyncResult {get;}
```

Bemerkungen

Eine Referenz zum ULSyncResult-Objekt, das die Ergebnisse der letzten Synchronisation für diese Verbindung repräsentiert.

Siehe auch

- [ULConnection.Synchronize-Methode \[UltraLite.NET\] auf Seite 156](#)
- [ULConnection.SyncParms-Eigenschaft \[UltraLite.NET\] auf Seite 165](#)
- [ULConnection.SyncResult-Eigenschaft \[UltraLite.NET\] auf Seite 166](#)

InfoMessage-Ereignis

Tritt ein, wenn UltraLite.NET eine Warnung oder eine Informationsnachricht auf dieser Verbindung absetzt

Visual Basic-Syntax

```
Public Event InfoMessage As ULInfoMessageEventHandler
```

C#-Syntax

```
public event ULInfoMessageEventHandler InfoMessage;
```

Bemerkungen

Zur Verarbeitung von UltraLite.NET-Warnungen oder Informationsnachrichten müssen Sie einen ULInfoMessageEventHandler-Delegaten erstellen und diesem Ereignis zuordnen.

Siehe auch

- [ULInfoMessageEventHandler-Delegat \[UltraLite.NET\] auf Seite 453](#)

Beispiel

Der folgende Code definiert einen Event-Handler für Informationsnachrichten:

```

' Visual Basic
Private Sub MyInfoMessageHandler( _
    obj As Object, args As ULInfoMessageEventArgs _
)
    System.Console.WriteLine( _
        "InfoMessageHandler: " + args.NativeError + ", " _
        + args.Message _
    )
End Sub

```

Dies entspricht dem folgenden Code in der Sprache C#:

```

// C#
private void MyInfoMessageHandler(
    object obj, ULInfoMessageEventArgs args
)
{
    System.Console.WriteLine(
        "InfoMessageHandler: " + args.NativeError + ", " _
        + args.Message
    );
}

```

Der folgende Code fügt der Verbindung "conn" die MyInfoMessageHandler-Methode hinzu.

```

' Visual Basic
AddHandler conn.InfoMessage, AddressOf MyInfoMessageHandler

```

Dies entspricht dem folgenden Code in der Sprache C#:

```

// C#
conn.InfoMessage +=
    new ULInfoMessageEventHandler(MyInfoMessageHandler);

```

StateChange-Ereignis

Tritt ein, wenn sich der Status dieser Verbindung ändert

Visual Basic-Syntax

```
Public Event StateChange As StateChangeEventHandler
```

C#-Syntax

```
public event override StateChangeEventHandler StateChange;
```

Bemerkungen

Zur Verarbeitung von Statusänderungsnachrichten müssen Sie einen System.Data.StateChangeEventHandler-Delegaten erstellen und diesem Ereignis zuordnen.

Siehe auch

- [System.Data.StateChangeEventHandler](#)

Beispiel

Der folgende Code definiert einen Event-Handler für Statusänderungen.

```
' Visual Basic
Private Sub MyStateHandler( _
    obj As Object, args As StateChangeEventArgs _
)
    System.Console.WriteLine( _
        "StateHandler: " + args.OriginalState + " to " _
        + args.CurrentState _
    )
End Sub
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
private void MyStateHandler(
    object obj, StateChangeEventArgs args
)
{
    System.Console.WriteLine(
        "StateHandler: " + args.OriginalState + " to "
        + args.CurrentState
    );
}
```

Der folgende Code fügt den Message-Handler MyStateHandler der Verbindung "conn" hinzu.

```
' Visual Basic
AddHandler conn.StateChange, AddressOf MyStateHandler
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
conn.StateChange += new StateChangeEventHandler(MyStateHandler);
```

INVALID_DATABASE_ID-Feld

UL-Erw.: Eine Datenbank-ID-Konstante, die angibt, dass die ULConnection.DatabaseID-Eigenschaft nicht festgelegt wurde.

Visual Basic-Syntax

```
Public Const INVALID_DATABASE_ID As Long
```

C#-Syntax

```
public const long INVALID_DATABASE_ID;
```

Siehe auch

- [ULConnection.DatabaseID-Eigenschaft \[UltraLite.NET\] auf Seite 162](#)

SYNC_ALL_DB-Feld

Leere Publikationsliste, steht für die gesamte Datenbank.

Visual Basic-Syntax

```
Public Const SYNC_ALL_DB As String
```

C#-Syntax

```
public const String SYNC_ALL_DB;
```

SYNC_ALL_PUBS-Feld

Publikationsname "*", steht für alle Publikationen.

Visual Basic-Syntax

```
Public Const SYNC_ALL_PUBS As String
```

C#-Syntax

```
public const String SYNC_ALL_PUBS;
```

ULConnectionParms-Klasse

UL-Erw.: Erstellt eine Verbindungszeichenfolge zum Öffnen einer Verbindung zu einer UltraLite-Datenbank.

Visual Basic-Syntax

```
Public Class ULConnectionParms Inherits System.ComponentModel.Component
```

C#-Syntax

```
public class ULConnectionParms : System.ComponentModel.Component
```

Basisklassen

- [System.ComponentModel.Component](#)

Mitglieder

Alle Mitglieder der ULConnectionParms-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ULConnectionParms-Konstruktor	Initialisiert eine ULConnectionParms-Instanz mit ihren Standardwerten.
Dispose-Methode (Geerbt von System.ComponentModel.Component)	Gibt alle Ressourcen frei, die von der System.ComponentModel.Component verwendet werden.
Finalize-Methode (Geerbt von System.ComponentModel.Component)	Gibt unverwaltete Ressourcen frei und führt andere Aufräuvorgänge aus, bevor die System.ComponentModel.Component von der Garbage Collection bereinigt wird.
GetService-Methode (Geerbt von System.ComponentModel.Component)	Gibt ein Objekt zurück, das für einen Dienst steht, der von der System.ComponentModel.Component oder ihrem System.ComponentModel.Container bereitgestellt wird.

Name	Beschreibung
ToString-Methode	Gibt die Zeichenfolge zurück, die diese Instanz repräsentiert.
AdditionalParms-Eigenschaft	Legt zusätzliche Parameter als semikolongetrennte Liste von Paaren der Form Name=Wert fest.
CacheSize-Eigenschaft	Legt die Größe des Caches fest.
CanRaiseEvents-Eigenschaft (Geerbt von System.ComponentModel.Component)	Ruft einen Wert ab, der anzeigt, ob die Komponente ein Ereignis auslösen kann.
ConnectionString-Eigenschaft	Legt einen Namen für die Verbindung fest.
Container-Eigenschaft (Geerbt von System.ComponentModel.Component)	Ruft den System.ComponentModel.IContainer ab, der die System.ComponentModel.Component enthält.
DatabaseOnDesktop-Eigenschaft	Legt den Pfad und den Dateinamen der UltraLite-Datenbank auf Windows PC-Plattformen fest.
DatabaseOnDevice-Eigenschaft	Legt den Pfad und den Dateinamen für die UltraLite-Datenbank unter Windows Mobile fest.
DesignMode-Eigenschaft (Geerbt von System.ComponentModel.Component)	Ruft einen Wert ab, der angibt, ob System.ComponentModel.Component derzeit im Planungsmodus ist.
EncryptionKey-Eigenschaft	Legt einen Schlüssel für die Verschlüsselung der Datenbank fest
Disposed (Geerbt von System.ComponentModel.Component) Bezieht die Liste der Event-Handler, die dieser System.ComponentModel.Component angefügt wurden. Events-Eigenschaft (Geerbt von System.ComponentModel.Component)	Erscheint, wenn die Komponente durch einen Aufruf der System.ComponentModel.Component.Dispose -Methode gelöscht wird.
Password-Eigenschaft	Gibt das Kennwort für den authentifizierten Benutzer an.
Site-Eigenschaft (Geerbt von System.ComponentModel.Component)	Ruft das System.ComponentModel.ISite -Objekt der System.ComponentModel.Component ab bzw. legt es fest.
UserID-Eigenschaft	Legt einen authentifizierten Benutzer für die Datenbank fest.

Bemerkungen

Die häufig verwendeten Verbindungszeichenfolgen sind individuelle Eigenschaften des Objekts ULConnectionParms.

Mit einem ULConnectionParms-Objekt werden die Parameter zum Öffnen einer Verbindung (mit der ULConnection.Open-Methode) oder zum Löschen einer Datenbank (mit der ULDatabaseManager.DropDatabase-Methode) angegeben.

Führende und nachgestellte Leerzeichen werden in allen Werten ignoriert. Werte dürfen keine führenden oder nachgestellten Leerzeichen oder ein Semikolon enthalten oder mit einem Apostroph oder Anführungszeichen beginnen.

Beim Erstellen einer Verbindungszeichenfolge müssen Sie die Datenbank angeben und alle optionalen Verbindungseinstellungen festlegen. Wenn Sie alle Verbindungsparameter durch Festlegen der geeigneten Eigenschaften für ein ULConnectionParms-Objekt angegeben haben, können Sie mit der ULConnectionParms.ToString-Methode eine Verbindungszeichenfolge erstellen. Die resultierende Zeichenfolge wird verwendet, um ein neues ULConnection-Objekt mit dem ULConnection(String)-Konstruktor zu erstellen oder die ULConnection.ConnectionString-Eigenschaft für ein vorhandenes ULConnection-Objekt festzulegen.

Datenbank identifizieren

Jede Instanz enthält plattformspezifische Pfade zu der Datenbank. Es wird nur der Wert für die ausführende Plattform verwendet. Im folgenden Code wird der Pfad \UltraLite\mydb1.udb unter Windows Mobile verwendet, während mydb2.udb auf anderen Plattformen verwendet wird.

```
' Visual Basic
Dim dbName As ULConnectionParms = new ULConnectionParms
dbName.DatabaseOnDevice = "\\UltraLite\mydb1.udb"
dbName.DatabaseOnDesktop = "somedir\mydb2.udb"
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULConnectionParms dbName = new ULConnectionParms();
dbName.DatabaseOnDevice = "\\UltraLite\mydb1.udb";
dbName.DatabaseOnDesktop = "somedir\mydb2.udb";
```

Die empfohlene Erweiterung für UltraLite-Datenbankdateien ist .udb. Auf Windows Mobile-Geräten ist die Standarddatenbank \UltraLiteDB\ulstore.udb. Auf anderen Windows-Plattformen ist die Standarddatenbank ulstore.udb. In C# muss jedem Backslashzeichen in einem Pfad ein Escapezeichen vorangestellt oder eine von @-Zeichen umschlossene Literal-Zeichenfolge verwendet werden.

Bei der Verwendung mehrerer Datenbanken muss für jede Datenbank ein Datenbankname angegeben werden.

Optionale Verbindungseinstellungen

Abhängig von den Anforderungen Ihrer Anwendung und davon, wie die Datenbank erstellt wurde, müssen Sie möglicherweise einen vom Standardwert abweichenden ULConnectionParms.UserID-Wert, einen vom Standardwert abweichenden ULConnectionParms.Password-Wert, einen ULConnectionParms.EncryptionKey-Wert für die Datenbank und den ULConnectionParms.CacheSize-

Wert angeben. Wenn Ihre Anwendung mehrere Verbindungen verwendet, müssen Sie für jede Verbindung einen eindeutigen `ULConnectionParms.ConnectionName`-Wert angeben.

Datenbanken werden mit der einfach authentifizierten Benutzer-ID DBA und dem anfänglichen Kennwort `sql` erstellt. Standardmäßig werden die Verbindungen mit der Benutzer-ID DBA und dem Kennwort `sql` geöffnet. Um den Standardbenutzer zu deaktivieren, verwenden Sie die `ULConnection.RevokeConnectFrom`-Methode. Um einen Benutzer hinzuzufügen oder das Kennwort für einen Benutzer zu ändern, verwenden Sie die `ULConnection.GrantConnectTo`-Methode.

Wenn bei der Erstellung der Datenbank ein Chiffrierschlüssel angegeben wurde, müssen alle nachfolgenden Verbindungen mit der Datenbank denselben Chiffrierschlüssel verwenden. Um den Chiffrierschlüssel einer Datenbank ändern, verwenden Sie die `ULConnection.ChangeEncryptionKey`-Methode.

Siehe auch

- [ULConnection.Open-Methode \[UltraLite.NET\] auf Seite 150](#)
- [ULDatabaseManager.DropDatabase-Methode \[UltraLite.NET\] auf Seite 225](#)
- [ULConnectionParms.ToString-Methode \[UltraLite.NET\] auf Seite 172](#)
- [ULConnection-Klasse \[UltraLite.NET\] auf Seite 121](#)
- [ULConnection.ConnectionString-Eigenschaft \[UltraLite.NET\] auf Seite 160](#)
- [ULConnectionParms.UserID-Eigenschaft \[UltraLite.NET\] auf Seite 178](#)
- [ULConnectionParms.Password-Eigenschaft \[UltraLite.NET\] auf Seite 177](#)
- [ULConnectionParms.EncryptionKey-Eigenschaft \[UltraLite.NET\] auf Seite 177](#)
- [ULConnectionParms.CacheSize-Eigenschaft \[UltraLite.NET\] auf Seite 175](#)
- [ULConnectionParms.ConnectionName-Eigenschaft \[UltraLite.NET\] auf Seite 175](#)
- [ULConnectionParms.AdditionalParms-Eigenschaft \[UltraLite.NET\] auf Seite 173](#)
- [ULConnection.RevokeConnectFrom-Methode \[UltraLite.NET\] auf Seite 152](#)
- [ULConnection.GrantConnectTo-Methode \[UltraLite.NET\] auf Seite 150](#)
- [ULConnection.ChangeEncryptionKey-Methode \[UltraLite.NET\] auf Seite 134](#)
- „UltraLite-Verbindungsparameter“ [*UltraLite - Datenbankverwaltung*]

ULConnectionParms-Konstruktor

Initialisiert eine `ULConnectionParms`-Instanz mit ihren Standardwerten.

Visual Basic-Syntax

```
Public Sub New()
```

C#-Syntax

```
public ULConnectionParms()
```

ToString-Methode

Gibt die Zeichenfolge zurück, die diese Instanz repräsentiert.

Visual Basic-Syntax

```
Public Overrides Function ToString() As String
```

C#-Syntax

```
public override string ToString()
```

Rückgabe

Die Zeichenfolge für diese Instanz ist eine durch Semikola getrennte Liste von Schlüsselwort=Wert-Paaren.

AdditionalParms-Eigenschaft

Legt zusätzliche Parameter als semikolongetrennte Liste von Paaren der Form Name=Wert fest.

Visual Basic-Syntax

```
Public Property AdditionalParms As String
```

C#-Syntax

```
public string AdditionalParms {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt eine ungültige Verbindungszeichenfolge.

Bemerkungen

Diese Parameter werden seltener verwendet.

Eine durch Semikolons getrennte Liste von zusätzlichen Parametern in der Form Schlüsselwort=Wert-Paaren. Die Werte der Schlüsselwort=Wert-Liste müssen den Regeln für ULConnection.ConnectionString entsprechen. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Die Werte für die Parameter für die Seitengröße und die Größe der Reserve werden in der Einheit Byte angegeben. Verwenden Sie das Suffix k oder K und das Suffix M oder m, um die Einheiten kB bzw. MB anzugeben.

Folgende zusätzliche Parameter sind verfügbar:

Schlüsselwort	Beschreibung
dbn	Kennzeichnet eine geladene Datenbank, mit der eine Verbindung hergestellt werden muss. Wenn eine Datenbank auf einem Server gestartet wird, wird ihr ein Datenbankname zugewiesen, und zwar entweder explizit mit der dbn-Option oder von UltraLite unter Verwendung des Dateinamens, wobei Erweiterung und Pfad weggelassen werden. Beim Öffnen von Verbindungen sucht UltraLite zunächst nach einer laufenden Datenbank mit einem übereinstimmenden dbn-Wert. Wenn keine gefunden wird, startet UltraLite eine neue Datenbank mit dem geeigneten Datenbank-Dateinamenparameter (mit der Eigenschaft DatabaseOnDevice oder DatabaseOnDesktop). Dieser Parameter ist erforderlich, wenn die Anwendung (oder UltraLite-Engine) auf zwei unterschiedliche Datenbanken zugreifen muss, die denselben Basis-Dateinamen haben. Dieser Parameter wird nur verwendet, wenn eine Verbindung mit der <code>ULConnection.Open</code> -Methode geöffnet wird.
reserve_size	Reserviert Speicher im Dateisystem für die Speicherung von beständigen UltraLite-Daten. Mit dem Parameter <code>reserve_size</code> können Sie den für die UltraLite-Datenbank erforderlichen Speicher im Dateisystem vorab zuweisen, ohne Daten einzufügen. Durch das Reservieren von Speicher im Dateisystem können Sie die Performance geringfügig steigern und Fehler aufgrund von zu wenig Speicherplatz verhindern. Standardmäßig wächst die Datei für den beständigen Speicher nur bei Bedarf, wenn die Anwendung die Datenbank aktualisiert. Der <code>reserve_size</code> -Parameter reserviert Speicherplatz im Dateisystem, wobei auch die Metadaten in der Datei für den beständigen Speicher einbezogen werden und nicht nur die unformatierten Daten. Der Metadaten-Overhead und die Datenkomprimierung müssen bei der Ermittlung des erforderlichen Speichers im Dateisystem anhand des Umfangs der Datenbankdaten berücksichtigt werden. Der Parameter <code>reserve_size</code> reserviert Speicher, indem die Datei für den beständigen Speicher beim Start auf die betreffende Reservierungsgröße gebracht wird, und zwar unabhängig von der zuvor vorhandenen Datei. Die Datei wird niemals gekürzt. Die folgende Parameterzeichenfolge stellt sicher, dass die Datei für den beständigen Speicher beim Start mindestens 2 MB groß ist: <code>create-Parms.AdditionalParms = "reserve_size=2m"</code> . Dieser Parameter wird nur verwendet, wenn eine Verbindung mit der <code>ULConnection.Open</code> -Methode geöffnet wird.
start	Gibt den Speicherort an und startet dann die UltraLite-Engine. Geben Sie den Verbindungsparameter <code>StartLine</code> (START) nur beim Verbinden mit einer Engine an, die aktuell nicht läuft. Der Speicherort ist nur erforderlich, wenn sich die UltraLite-Engine nicht im Systempfad befindet.

Siehe auch

- [ULDatabaseManager.RuntimeType-Eigenschaft \[UltraLite.NET\] auf Seite 229](#)
- [ULConnection.ConnectionString-Eigenschaft \[UltraLite.NET\] auf Seite 160](#)
- [ULConnectionParms.DatabaseOnDevice-Eigenschaft \[UltraLite.NET\] auf Seite 176](#)
- [ULConnectionParms.DatabaseOnDesktop-Eigenschaft \[UltraLite.NET\] auf Seite 176](#)
- [ULConnection.Open-Methode \[UltraLite.NET\] auf Seite 150](#)
- „UltraLite-Verbindungsparameter“ [*UltraLite - Datenbankverwaltung*]

CacheSize-Eigenschaft

Legt die Größe des Caches fest.

Visual Basic-Syntax

```
Public Property CacheSize As String
```

C#-Syntax

```
public string CacheSize {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die die Cachegröße festlegt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic). Das bedeutet, dass der Standardwert von 16 Seiten verwendet wird.

Die Werte für die Cachegröße werden in Byte angegeben. Verwenden Sie das Suffix k oder K und das Suffix M oder m, um die Einheiten kB bzw. MB anzugeben.

Folgende Anweisung legt z.B. die Cachegröße auf 128 kB fest.

```
connParms.CacheSize = "128k"
```

Die Standardgröße für den Cachespeicher ist 16 Seiten. Bei einer standardmäßigen Seitengröße von 4 kB beträgt die Standard-Cachegröße somit 64 kB. Die minimale Cachegröße ist plattformabhängig.

Die Standardgröße für den Cachespeicher ist vorsichtig festgelegt. Wenn Ihre Tests ergeben, dass Sie eine bessere Performance benötigen, müssen Sie die Cachegröße erhöhen.

Ein Erhöhen der Cachegröße über die Größe der Datenbank hinaus bietet keine Performance-Steigerung. Große Cachegrößen können sich jedoch negativ auf die Anzahl der anderen verwendbaren Anwendungen auswirken.

Wenn die Cachegröße nicht oder falsch festgelegt ist, wird die anfängliche Cachegröße verwendet.

ConnectionString-Eigenschaft

Legt einen Namen für die Verbindung fest.

Visual Basic-Syntax

```
Public Property ConnectionString As String
```

C#-Syntax

```
public string ConnectionString {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Wird nur benötigt, wenn Sie mehr als eine Verbindung zur Datenbank herstellen.

Eine Zeichenfolge, die den Namen der Verbindung angibt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

DatabaseOnDesktop-Eigenschaft

Legt den Pfad und den Dateinamen der UltraLite-Datenbank auf Windows PC-Plattformen fest.

Visual Basic-Syntax

```
Public Property DatabaseOnDesktop As String
```

C#-Syntax

```
public string DatabaseOnDesktop {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die den absoluten oder relativen Pfad zur Datenbank angibt. Wenn der Wert eine Nullreferenz ist ("Nothing" in Visual Basic), wird die Datenbank ulstore.udb verwendet. In C# muss jedem Backslashzeichen in einem Pfad ein Escapezeichen vorangestellt oder eine von @-Zeichen umschlossene Literal-Zeichenfolge verwendet werden. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

DatabaseOnDevice-Eigenschaft

Legt den Pfad und den Dateinamen für die UltraLite-Datenbank unter Windows Mobile fest.

Visual Basic-Syntax

```
Public Property DatabaseOnDevice As String
```

C#-Syntax

```
public string DatabaseOnDevice {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die den vollständigen Pfad zur Datenbank angibt. Wenn der Wert eine Nullreferenz ist ("Nothing" in Visual Basic), wird die Datenbank \\UltraLiteDB\ulstore.udb verwendet. In C# muss jedem Backslashzeichen in einem Pfad ein Escapezeichen vorangestellt oder eine von @-Zeichen umschlossene Literal-Zeichenfolge verwendet werden. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

EncryptionKey-Eigenschaft

Legt einen Schlüssel für die Verschlüsselung der Datenbank fest

Visual Basic-Syntax

```
Public Property EncryptionKey As String
```

C#-Syntax

```
public string EncryptionKey {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die den Chiffrierschlüssel angibt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic). Das bedeutet, dass keine Verschlüsselung verwendet wird.

Alle Verbindungen müssen den Schlüssel verwenden, der bei der Erstellung der Datenbank angegeben wurde. Wenn Sie den Schlüssel verlieren, können Sie nie mehr auf die Datenbank zugreifen.

Wie bei allen Kennwörtern sollte ein Schlüsselwert gewählt werden, der nur schwer erraten werden kann. Der Schlüssel kann von beliebiger Länge sein. Im Allgemeinen gilt: Je länger der Schlüssel, desto schwieriger lässt er sich erraten. Das Kombinieren von Zahlen, Buchstaben und Sonderzeichen reduziert die Gefahr des Erratens.

Siehe auch

- [ULConnection.ChangeEncryptionKey-Methode \[UltraLite.NET\] auf Seite 134](#)

Password-Eigenschaft

Gibt das Kennwort für den authentifizierten Benutzer an.

Visual Basic-Syntax

```
Public Property Password As String
```

C#-Syntax

```
public string Password {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die eine Datenbankbenutzer-ID angibt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Bei Kennwörtern wird die Groß- und Kleinschreibung berücksichtigt.

Bei der Erstellung einer Datenbank wird sql als Kennwort für die Benutzer-ID DBA festgelegt.

Siehe auch

- [ULConnectionParms.UserID-Eigenschaft \[UltraLite.NET\] auf Seite 178](#)

UserID-Eigenschaft

Legt einen authentifizierten Benutzer für die Datenbank fest.

Visual Basic-Syntax

```
Public Property UserID As String
```

C#-Syntax

```
public string UserID {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die eine Datenbankbenutzer-ID angibt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Benutzer-IDs beachten die Groß- und Kleinschreibung.

Datenbanken werden zunächst mit dem authentifizierten Benutzer DBA erstellt.

Wenn weder die Benutzer-ID noch das Kennwort angegeben werden, wird der Benutzer DBA mit dem Kennwort sql verwendet. Um die Datenbank sicherer zu machen, ändern Sie das Kennwort des DBA-Benutzers oder erstellen Sie neue Benutzer (mit der `ULConnection.GrantConnectTo`-Methode) und entfernen Sie den DBA-Benutzer (mit der `ULConnection.RevokeConnectFrom`-Methode).

Siehe auch

- [ULConnectionParms.Password-Eigenschaft \[UltraLite.NET\] auf Seite 177](#)
- [ULConnection.GrantConnectTo-Methode \[UltraLite.NET\] auf Seite 150](#)
- [ULConnection.RevokeConnectFrom-Methode \[UltraLite.NET\] auf Seite 152](#)

ULConnectionStringBuilder-Klasse

Erstellt eine Verbindungszeichenfolge zum Öffnen einer Verbindung zu einer UltraLite-Datenbank.

Visual Basic-Syntax

```
Public NotInheritable Class ULConnectionStringBuilder
    Inherits System.Data.Common.DbConnectionStringBuilder
```

C#-Syntax

```
public sealed class ULConnectionStringBuilder :
    System.Data.Common.DbConnectionStringBuilder
```

Basisklassen

- [System.Data.Common.DbConnectionStringBuilder](#)

Mitglieder

Alle Mitglieder der ULConnectionStringBuilder-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ULConnectionStringBuilder-Konstruktor	Initialisiert ein ULConnectionStringBuilder-Objekt mit dessen Standardwerten.
Add-Methode (geerbt aus System.Data.Common.DbConnectionStringBuilder)	Fügt dem System.Data.Common.DbConnectionStringBuilder -Objekt einen Eintrag mit dem angegebenen Schlüssel und Wert hinzu.
AppendKeyValuePair-Methode (geerbt aus System.Data.Common.DbConnectionStringBuilder)	Stellt eine effiziente und sichere Möglichkeit dar, einen Schlüssel und Wert an ein vorhandenes System.Text.StringBuilder -Objekt anzuhängen.
Clear-Methode (geerbt aus System.Data.Common.DbConnectionStringBuilder)	Löscht den Inhalt der System.Data.Common.DbConnectionStringBuilder -Instanz.
ClearPropertyDescriptors-Methode (geerbt aus System.Data.Common.DbConnectionStringBuilder)	Löscht die Sammlung von System.ComponentModel.PropertyDescriptor -Objekte auf dem verknüpften System.Data.Common.DbConnectionStringBuilder -Objekt.
ContainsKey-Methode	Ermittelt, ob das ULConnectionStringBuilder-Objekt ein spezifisches Schlüsselwort enthält.
EquivalentTo-Methode	Vergleicht die Verbindungsinformationen in diesem ULConnectionStringBuilder-Objekt mit den Verbindungsinformationen im angegebenen DbConnectionStringBuilder-Objekt.

Name	Beschreibung
GetProperties-Methode (geerbt aus <code>System.Data.Common.DbConnectionStringBuilder</code>)	Füllt ein angegebenes <code>System.Collections.Hashtable</code> -Objekt mit Informationen zu allen Eigenschaften dieses <code>System.Data.Common.DbConnectionStringBuilder</code> -Objekts.
GetShortName-Methode	Ruft die kurze Version des angegebenen Schlüsselworts ab.
Remove-Methode	Löscht den Eintrag mit dem angegebenen Schlüssel aus dem <code>ULConnectionStringBuilder</code> -Objekt.
ShouldSerialize-Methode (Geerbt von <code>System.Data.Common.DbConnectionStringBuilder</code>)	Gibt an, ob der angegebene Schlüssel in dieser <code>System.Data.Common.DbConnectionStringBuilder</code> -Instanz existiert.
ToString-Methode (geerbt aus <code>System.Data.Common.DbConnectionStringBuilder</code>)	Gibt die Verbindungszeichenfolge zurück, die diesem <code>System.Data.Common.DbConnectionStringBuilder</code> -Objekt zugeordnet ist.
TryGetValue-Methode	Ruft einen Wert ab, der dem angegebenen Schlüssel aus diesem <code>ULConnectionStringBuilder</code> -Objekt entspricht.
BrowsableConnectionString-Eigenschaft (geerbt aus <code>System.Data.Common.DbConnectionStringBuilder</code>)	Ruft einen Wert ab, der angibt, ob das <code>System.Data.Common.DbConnectionStringBuilder.ConnectionString</code> -Objekt in den Visual Studio-Designern sichtbar ist, bzw. legt ihn fest.
CacheSize-Eigenschaft	UL-Erw.: Legt die Größe des Caches fest.
ConnectionName-Eigenschaft	Legt einen Namen für die Verbindung fest.
ConnectionString-Eigenschaft (geerbt aus <code>System.Data.Common.DbConnectionStringBuilder</code>)	Ruft die Verbindungszeichenfolge ab, die dem <code>DbConnectionStringBuilder</code> -Objekt zugeordnet ist, bzw. legt sie fest.
Count-Eigenschaft (geerbt aus <code>System.Data.Common.DbConnectionStringBuilder</code>)	Ruft die aktuelle Anzahl von Schlüsseln ab, die in der <code>System.Data.Common.DbConnectionStringBuilder.ConnectionString</code> -Eigenschaft enthalten sind.
DatabaseKey-Eigenschaft	Legt einen Schlüssel für die Verschlüsselung der Datenbank fest
DatabaseName-Eigenschaft	Gibt einen Namen für die Datenbank oder einen Namen einer geladenen Datenbank an, mit der eine Verbindung hergestellt werden muss.
DatabaseOnDesktop-Eigenschaft	UL-Erw.: Legt den Pfad und den Dateinamen der UltraLite-Datenbank auf Windows PC-Plattformen fest.

Name	Beschreibung
DatabaseOnDevice-Eigenschaft	UL-Erw.: Legt den Pfad und den Dateinamen für die UltraLite-Datenbank unter Windows Mobile fest.
IsFixedSize-Eigenschaft (geerbt aus System.Data.Common.DbConnectionStringBuilder)	Ruft einen Wert ab, der angibt, ob das System.Data.Common.DbConnectionStringBuilder -Objekt eine feste Größe hat.
IsReadOnly-Eigenschaft (geerbt aus System.Data.Common.DbConnectionStringBuilder)	Ruft einen Wert ab, der angibt, ob das System.Data.Common.DbConnectionStringBuilder -Objekt schreibgeschützt ist.
Keys-Eigenschaft (Geerbt von System.Data.Common.DbConnectionStringBuilder)	Ruft ein System.Collections.ICollection -Objekt ab, das die Schlüssel aus dem System.Data.Common.DbConnectionStringBuilder -Objekt enthält.
OrderedTableScans-Eigenschaft	Legt fest, ob SQL-Abfragen ohne ORDER BY-Klauseln standardmäßig geordnete Table-Scans durchführen sollen.
Password-Eigenschaft	Gibt das Kennwort für den authentifizierten Benutzer an.
ReserveSize-Eigenschaft	UL-Erw.: Gibt den Reservierungsspeicherplatz des Dateisystems zur Speicherung von beständigen UltraLite-Daten an.
StartLine-Eigenschaft	Gibt den Speicherort an und startet dann die UltraLite-Engine.
this-Eigenschaft	Gibt den Wert des angegebenen Verbindungsschlüsselworts an.
UserID-Eigenschaft	Legt einen authentifizierten Benutzer für die Datenbank fest.
Values-Eigenschaft (geerbt aus System.Data.Common.DbConnectionStringBuilder)	Ruft ein System.Collections.ICollection -Objekt ab, das die Werte aus dem System.Data.Common.DbConnectionStringBuilder -Objekt enthält.

Bemerkungen

Die häufig verwendeten Verbindungszeichenfolgen sind individuelle Eigenschaften des Objekts ULConnectionStringBuilder.

Die ULConnectionStringBuilder-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

Mit einem ULConnectionStringBuilder-Objekt werden die Parameter zum Öffnen einer Verbindung (mit der ULConnection.Open-Methode) oder zum Löschen einer Datenbank (mit der ULDatabaseManager.DropDatabase-Methode) angegeben.

Führende und nachgestellte Leerzeichen werden in allen Werten ignoriert. Werte dürfen keine führenden oder nachgestellten Leerzeichen oder ein Semikolon enthalten oder mit einem Apostroph oder Anführungszeichen beginnen.

Beim Erstellen einer Verbindungszeichenfolge müssen Sie die Datenbank angeben und alle optionalen Verbindungseinstellungen festlegen. Wenn Sie alle Verbindungsparameter durch Festlegen der geeigneten Eigenschaften für ein `ULConnectionStringBuilder`-Objekt angegeben haben, können Sie mit `System.Data.Common.DbConnectionStringBuilder.ConnectionString` eine Verbindungszeichenfolge erstellen. Die resultierende Zeichenfolge wird verwendet, um ein neues `ULConnection`-Objekt mit dem `ULConnection(String)`-Konstruktor zu erstellen oder die `ULConnection.ConnectionString`-Eigenschaft für ein vorhandenes `ULConnection`-Objekt festzulegen.

Datenbank identifizieren

Jede Instanz enthält plattformspezifische Pfade zu der Datenbank. Es wird nur der Wert für die ausführende Plattform verwendet. Im folgenden Code wird der Pfad `\UltraLite\mydb1.udb` unter Windows Mobile verwendet, während `mydb2.udb` auf anderen Plattformen verwendet wird.

```
' Visual Basic
Dim dbName As ULConnectionStringBuilder = _
    new ULConnectionStringBuilder
dbName.DatabaseOnDevice = "\UltraLite\mydb1.udb"
dbName.DatabaseOnDesktop = "somedir\mydb2.udb"
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULConnectionStringBuilder dbName = new ULConnectionStringBuilder();
dbName.DatabaseOnDevice = "\\UltraLite\\mydb1.udb";
dbName.DatabaseOnDesktop = @"somedir\mydb2.udb";
```

Die empfohlene Erweiterung für UltraLite-Datenbankdateien ist `.udb`. Auf Windows Mobile-Geräten ist die Standarddatenbank `\UltraLiteDB\ulstore.udb`. Auf anderen Windows-Plattformen ist die Standarddatenbank `ulstore.udb`. In C# muss jedem Backslashzeichen in einem Pfad ein Escapezeichen vorangestellt oder eine von `@`-Zeichen umschlossene Literal-Zeichenfolge verwendet werden.

Bei der Verwendung mehrerer Datenbanken muss für jede Datenbank ein Datenbankname angegeben werden.

Optionale Verbindungseinstellungen

Abhängig von den Anforderungen Ihrer Anwendung und davon, wie die Datenbank erstellt wurde, müssen Sie möglicherweise einen vom Standardwert abweichenden `ULConnectionStringBuilder.UserID`-Wert, einen vom Standardwert abweichenden `ULConnectionStringBuilder.Password`-Wert, einen `ULConnectionStringBuilder.DatabaseKey`-Wert für die Datenbank und den `ULConnectionStringBuilder.CacheSize`-Wert angeben. Wenn Ihre Anwendung mehrere Verbindungen verwendet, müssen Sie für jede Verbindung einen eindeutigen `ULConnectionStringBuilder.ConnectionName`-Wert angeben.

Datenbanken werden mit der einfach authentifizierten Benutzer-ID `DBA` und dem anfänglichen Kennwort `sql` erstellt. Standardmäßig werden die Verbindungen mit der Benutzer-ID `DBA` und dem Kennwort `sql` geöffnet. Um den Standardbenutzer zu deaktivieren, verwenden Sie die `ULConnection.RevokeConnectFrom`-Methode. Um einen Benutzer hinzuzufügen oder das Kennwort für einen Benutzer zu ändern, verwenden Sie die `ULConnection.GrantConnectTo`-Methode.

Wenn bei der Erstellung der Datenbank ein Chiffrierschlüssel angegeben wurde, müssen alle nachfolgenden Verbindungen mit der Datenbank denselben Chiffrierschlüssel verwenden. Um den

Chiffrierschlüssel einer Datenbank ändern, verwenden Sie die `ULConnection.ChangeEncryptionKey`-Methode.

Siehe auch

- [ULConnection.Open-Methode \[UltraLite.NET\] auf Seite 150](#)
- [ULDatabaseManager.DropDatabase-Methode \[UltraLite.NET\] auf Seite 225](#)
- [ULConnection-Klasse \[UltraLite.NET\] auf Seite 121](#)
- [ULConnection.ConnectionString-Eigenschaft \[UltraLite.NET\] auf Seite 160](#)
- [ULConnectionStringBuilder.DatabaseName-Eigenschaft \[UltraLite.NET\] auf Seite 188](#)
- [ULConnectionStringBuilder.UserID-Eigenschaft \[UltraLite.NET\] auf Seite 194](#)
- [ULConnectionStringBuilder.Password-Eigenschaft \[UltraLite.NET\] auf Seite 191](#)
- [ULConnectionStringBuilder.DatabaseKey-Eigenschaft \[UltraLite.NET\] auf Seite 188](#)
- [ULConnectionStringBuilder.CacheSize-Eigenschaft \[UltraLite.NET\] auf Seite 186](#)
- [ULConnectionStringBuilder.ConnectionName-Eigenschaft \[UltraLite.NET\] auf Seite 187](#)
- [ULConnection.RevokeConnectFrom-Methode \[UltraLite.NET\] auf Seite 152](#)
- [ULConnection.GrantConnectTo-Methode \[UltraLite.NET\] auf Seite 150](#)
- [ULConnection.ChangeEncryptionKey-Methode \[UltraLite.NET\] auf Seite 134](#)
- [System.Data.Common.DbConnectionStringBuilder.ConnectionString](#)
- „UltraLite-Verbindungsparameter“ [*UltraLite - Datenbankverwaltung*]

ULConnectionStringBuilder-Konstruktor

Initialisiert ein `ULConnectionStringBuilder`-Objekt mit dessen Standardwerten.

Überladungsliste

Name	Beschreibung
ULConnectionStringBuilder()-Konstruktoren	Initialisiert ein <code>ULConnectionStringBuilder</code> -Objekt mit dessen Standardwerten.
ULConnectionStringBuilder(string)-Konstruktor	Initialisiert ein <code>ULConnectionStringBuilder</code> -Objekt mit der angegebenen Verbindungszeichenfolge.

ULConnectionStringBuilder()-Konstruktoren

Initialisiert ein `ULConnectionStringBuilder`-Objekt mit dessen Standardwerten.

Visual Basic-Syntax

```
Public Sub New()
```

C#-Syntax

```
public ULConnectionStringBuilder()
```

ULConnectionStringBuilder(string)-Konstruktor

Initialisiert ein ULConnectionStringBuilder-Objekt mit der angegebenen Verbindungszeichenfolge.

Visual Basic-Syntax

```
Public Sub New(ByVal connectionString As String)
```

C#-Syntax

```
public ULConnectionStringBuilder(string connectionString)
```

Parameter

- **connectionString** Eine UltraLite.NET-Verbindungszeichenfolge. Eine Verbindungszeichenfolge ist eine durch Semikola getrennte Liste von Schlüsselwort-Wert-Paaren.

ContainsKey-Methode

Ermittelt, ob das ULConnectionStringBuilder-Objekt ein spezifisches Schlüsselwort enthält.

Visual Basic-Syntax

```
Public Overrides Function ContainsKey(  
    ByVal keyword As String  
    ) As Boolean
```

C#-Syntax

```
public override bool ContainsKey(string keyword)
```

Parameter

- **keyword** Der Name des Verbindungsschlüsselworts.

Rückgabe

TRUE, wenn dieser Verbindungszeichenfolgen-BUILDER einen Wert für das angegebene Schlüsselwort enthält, andernfalls FALSE.

EquivalentTo-Methode

Vergleicht die Verbindungsinformationen in diesem ULConnectionStringBuilder-Objekt mit den Verbindungsinformationen im angegebenen DbConnectionStringBuilder-Objekt.

Visual Basic-Syntax

```
Public Overrides Function EquivalentTo(  
    ByVal connectionStringBuilder As DbConnectionStringBuilder  
    ) As Boolean
```

C#-Syntax

```
public override bool EquivalentTo(  
    DbConnectionStringBuilder connectionStringBuilder  
)
```

Parameter

- **connectionStringBuilder** Das andere DbConnectionStringBuilder-Objekt, mit dem dieses ULConnectionStringBuilder-Objekt verglichen werden soll.

Rückgabe

TRUE, wenn dieses Objekt dem angegebenen DbConnectionStringBuilder-Objekt entspricht, ansonsten FALSE.

Siehe auch

- [System.Data.Common.DbConnectionStringBuilder](#)

GetShortName-Methode

Ruft die kurze Version des angegebenen Schlüsselworts ab.

Visual Basic-Syntax

```
Public Shared Function GetShortName(ByVal keyword As String) As String
```

C#-Syntax

```
public static string GetShortName(string keyword)
```

Parameter

- **keyword** Der Schlüssel des abzurufenden Elements.

Rückgabe

Die Kurzversion des angegebenen Schlüsselworts, wenn das Schlüsselwort erkannt wird, anderenfalls NULL

Remove-Methode

Löscht den Eintrag mit dem angegebenen Schlüssel aus dem ULConnectionStringBuilder-Objekt.

Visual Basic-Syntax

```
Public Overrides Function Remove(ByVal keyword As String) As Boolean
```

C#-Syntax

```
public override bool Remove(string keyword)
```

Parameter

- **keyword** Der Name des Verbindungsschlüsselworts.

Rückgabe

TRUE, wenn der Schlüssel in der Verbindungszeichenfolge vorhanden war und entfernt wurde, andernfalls FALSE.

TryGetValue-Methode

Ruft einen Wert ab, der dem angegebenen Schlüssel aus diesem `ULConnectionStringBuilder`-Objekt entspricht.

Visual Basic-Syntax

```
Public Overrides Function TryGetValue(  
    ByVal keyword As String,  
    ByVal value As Object  
) As Boolean
```

C#-Syntax

```
public override bool TryGetValue(string keyword, out Object value)
```

Parameter

- **keyword** Der Schlüssel des abzurufenden Elements.
- **value** Der dem Schlüssel entsprechende Wert

Rückgabe

TRUE, wenn das Schlüsselwort in der Verbindungszeichenfolge gefunden wurde, anderenfalls FALSE.

Bemerkungen

Mit der `TryGetValue`-Methode können Entwickler einen Wert auf einem `ULConnectionStringBuilder`-Objekt sicher abrufen, ohne zuvor die `ContainsKey`-Methode aufrufen zu müssen. Da die `TryGetValue`-Methode beim Aufruf keine Ausnahmebedingung auslöst, wenn ein nicht vorhandener Schlüssel übergeben wird, müssen Sie nicht nach einem Schlüssel suchen, bevor Sie seinen Wert abrufen. Wenn `TryGetValue` mit einem nicht vorhandenen Schlüssel aufgerufen wird, erhält der `value`-Parameter den Wert NULL ("Nothing" in Visual Basic).

CacheSize-Eigenschaft

UL-Erw.: Legt die Größe des Caches fest.

Visual Basic-Syntax

```
Public Property CacheSize As String
```

C#-Syntax

```
public string CacheSize {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die die Cachegröße festlegt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic). Das bedeutet, dass der Standardwert von 16 Seiten verwendet wird.

Die Werte für die Cachegröße werden in Byte angegeben. Verwenden Sie das Suffix k oder K und das Suffix M oder m, um die Einheiten kB bzw. MB anzugeben.

Folgende Anweisung legt z.B. die Cachegröße auf 128 kB fest.

```
connParms.CacheSize = "128k"
```

Die Standardgröße für den Cachespeicher ist 16 Seiten. Bei einer standardmäßigen Seitengröße von 4 kB beträgt die Standard-Cachegröße somit 64 kB. Die minimale Cachegröße ist plattformabhängig.

Die Standardgröße für den Cachespeicher ist vorsichtig festgelegt. Wenn Ihre Tests ergeben, dass Sie eine bessere Performance benötigen, müssen Sie die Cachegröße erhöhen.

Ein Erhöhen der Cachegröße über die Größe der Datenbank hinaus bietet keine Performance-Steigerung. Große Cachegrößen können sich jedoch negativ auf die Anzahl der anderen verwendbaren Anwendungen auswirken.

Wenn die Cachegröße nicht oder falsch festgelegt ist, wird die anfängliche Cachegröße verwendet.

ConnectionString-Eigenschaft

Legt einen Namen für die Verbindung fest.

Visual Basic-Syntax

```
Public Property ConnectionString As String
```

C#-Syntax

```
public string ConnectionString {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Wird nur benötigt, wenn Sie mehr als eine Verbindung zur Datenbank herstellen.

Eine Zeichenfolge, die den Namen der Verbindung angibt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

DatabaseKey-Eigenschaft

Legt einen Schlüssel für die Verschlüsselung der Datenbank fest

Visual Basic-Syntax

```
Public Property DatabaseKey As String
```

C#-Syntax

```
public string DatabaseKey {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die den Chiffrierschlüssel angibt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic). Das bedeutet, dass keine Verschlüsselung verwendet wird.

Alle Verbindungen müssen den Schlüssel verwenden, der bei der Erstellung der Datenbank angegeben wurde. Wenn Sie den Schlüssel verlieren, können Sie nie mehr auf die Datenbank zugreifen.

Wie bei allen Kennwörtern sollte ein Schlüsselwert gewählt werden, der nur schwer erraten werden kann. Der Schlüssel kann von beliebiger Länge sein. Im Allgemeinen gilt: Je länger der Schlüssel, desto schwieriger lässt er sich erraten. Das Kombinieren von Zahlen, Buchstaben und Sonderzeichen reduziert die Gefahr des Erratens.

Siehe auch

- [ULConnection.ChangeEncryptionKey-Methode \[UltraLite.NET\] auf Seite 134](#)

DatabaseName-Eigenschaft

Gibt einen Namen für die Datenbank oder einen Namen einer geladenen Datenbank an, mit der eine Verbindung hergestellt werden muss.

Visual Basic-Syntax

```
Public Property DatabaseName As String
```

C#-Syntax

```
public string DatabaseName {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die den Namen der Datenbank angibt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Wenn eine Datenbank auf einem Server gestartet wird, wird ihr ein Datenbankname zugewiesen, und zwar entweder explizit mit der dbn-Option oder von UltraLite unter Verwendung des Dateinamens, wobei Erweiterung und Pfad weggelassen werden.

Beim Öffnen von Verbindungen sucht UltraLite zunächst nach einer laufenden Datenbank mit einem übereinstimmenden dbn-Parameter. Wenn keine gefunden wird, startet UltraLite eine neue Datenbank mit dem geeigneten Datenbank-Dateinamenparameter (mit der Eigenschaft DatabaseOnDevice oder DatabaseOnDesktop).

Dieser Parameter ist erforderlich, wenn die Anwendung (oder UltraLite-Engine) auf zwei unterschiedliche Datenbanken zugreifen muss, die denselben Basis-Dateinamen haben.

Siehe auch

- [ULConnectionStringBuilder.DatabaseOnDevice-Eigenschaft \[UltraLite.NET\] auf Seite 190](#)
- [ULConnectionStringBuilder.DatabaseOnDesktop-Eigenschaft \[UltraLite.NET\] auf Seite 189](#)

DatabaseOnDesktop-Eigenschaft

UL-Erw.: Legt den Pfad und den Dateinamen der UltraLite-Datenbank auf Windows PC-Plattformen fest.

Visual Basic-Syntax

```
Public Property DatabaseOnDesktop As String
```

C#-Syntax

```
public string DatabaseOnDesktop {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die den absoluten oder relativen Pfad zur Datenbank angibt. Wenn der Wert eine Nullreferenz ist ("Nothing" in Visual Basic), wird die Datenbank ulstore.udb verwendet. In C# muss jedem Backslashzeichen in einem Pfad ein Escapezeichen vorangestellt oder eine von @-Zeichen umschlossene Literal-Zeichenfolge verwendet werden. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

DatabaseOnDevice-Eigenschaft

UL-Erw.: Legt den Pfad und den Dateinamen für die UltraLite-Datenbank unter Windows Mobile fest.

Visual Basic-Syntax

```
Public Property DatabaseOnDevice As String
```

C#-Syntax

```
public string DatabaseOnDevice {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die den vollständigen Pfad zur Datenbank angibt. Wenn der Wert eine Nullreferenz ist ("Nothing" in Visual Basic), wird die Datenbank \\UltraLiteDB\ulstore.udb verwendet. In C# muss jedem Backslashzeichen in einem Pfad ein Escapezeichen vorangestellt oder eine von @-Zeichen umschlossene Literal-Zeichenfolge verwendet werden. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

OrderedTableScans-Eigenschaft

Legt fest, ob SQL-Abfragen ohne ORDER BY-Klauseln standardmäßig geordnete Table-Scans durchführen sollen.

Visual Basic-Syntax

```
Public Property OrderedTableScans As String
```

C#-Syntax

```
public string OrderedTableScans {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine boolesche Zeichenfolge, die angibt, ob sortierte Table Scans verwendet werden sollen. z.B. TRUE/FALSE, yes/no, 1/0 etc. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Bei der Verwendung von Dynamic SQL greift UltraLite, sofern die Reihenfolge für die Ausführung einer Abfrage nicht von Bedeutung ist, direkt von den Datenbankseiten auf die Zeilen zu, anstatt den Primärschlüsselindex zu verwenden. Dies verbessert die Performance beim Abrufen von Zeilen. Zur Verwendung dieser Optimierung muss die Abfrage schreibgeschützt sein und alle Zeilen durchsuchen.

Wenn Zeilen in einer bestimmten Reihenfolge erwartet werden, sollte die ORDER BY-Anweisung Teil der SQL-Abfrage sein. Es ist jedoch möglich, dass einige Anwendungen sich auf das Verhalten verlassen, bei dem standardmäßig Zeilen in der Primärschlüssel-Reihenfolge zurückgegeben werden. In diesem Fall sollten Benutzer den Parameter OrderedTableScans auf 1 (TRUE, yes, on) setzen, um beim Durchsuchen einer Tabelle zum alten Verhalten zurückzukehren.

Wenn der OrderedTableScans-Wert auf 1 (TRUE, yes, on) gesetzt ist und der Benutzer keine ORDER BY-Klausel verwendet oder wenn die Abfrageperformance durch einen Index nicht verbessert wird, verwendet UltraLite standardmäßig den Primärschlüssel.

Password-Eigenschaft

Gibt das Kennwort für den authentifizierten Benutzer an.

Visual Basic-Syntax

```
Public Property Password As String
```

C#-Syntax

```
public string Password {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die eine Datenbankbenutzer-ID angibt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Bei Kennwörtern wird die Groß- und Kleinschreibung berücksichtigt.

Bei der Erstellung einer Datenbank wird sql als Kennwort für die Benutzer-ID DBA festgelegt.

Siehe auch

- [ULConnectionStringBuilder.UserID-Eigenschaft \[UltraLite.NET\] auf Seite 194](#)

ReserveSize-Eigenschaft

UL-Erw.: Gibt den Reservierungsspeicherplatz des Dateisystems zur Speicherung von beständigen UltraLite-Daten an.

Visual Basic-Syntax

```
Public Property ReserveSize As String
```

C#-Syntax

```
public string ReserveSize {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die die reservierte Größe festlegt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Die Werte für den Reservierungsgröße-Parameter werden in Byte angegeben. Verwenden Sie das Suffix k oder K und das Suffix M oder m, um die Einheiten kB bzw. MB anzugeben.

Mit dem Parameter `reserve_size` können Sie den für die UltraLite-Datenbank erforderlichen Speicher im Dateisystem vorab zuweisen, ohne Daten einzufügen. Durch das Reservieren von Speicher im Dateisystem können Sie die Performance geringfügig steigern und Fehler aufgrund von zu wenig Speicherplatz verhindern. Standardmäßig wächst die Datei für den beständigen Speicher nur bei Bedarf, wenn die Anwendung die Datenbank aktualisiert.

Der `reserve_size`-Parameter reserviert Speicherplatz im Dateisystem, wobei auch die Metadaten in der Datei für den beständigen Speicher einbezogen werden und nicht nur die unformatierten Daten. Der Metadaten-Overhead und die Datenkomprimierung müssen bei der Ermittlung des erforderlichen Speichers im Dateisystem anhand des Umfangs der Datenbankdaten berücksichtigt werden.

Der Parameter `reserve_size` reserviert Speicher, indem die Datei für den beständigen Speicher beim Start auf die betreffende Reservierungsgröße gebracht wird, und zwar unabhängig von der zuvor vorhandenen Datei. Die Datei wird niemals gekürzt.

Die folgende Parameterzeichenfolge stellt sicher, dass die Datei für den beständigen Speicher beim Start mindestens 2 MB umfasst.

```
connParms.ReserveSize = "2m"
```

StartLine-Eigenschaft

Gibt den Speicherort an und startet dann die UltraLite-Engine.

Visual Basic-Syntax

```
Public Property StartLine As String
```

C#-Syntax

```
public string StartLine {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die den Speicherort der Programmdatei der UltraLite-Engine angibt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Geben Sie den Verbindungsparameter StartLine (START) nur beim Verbinden mit einer Engine an, die aktuell nicht läuft.

Siehe auch

- [ULDatabaseManager.RuntimeType-Eigenschaft \[UltraLite.NET\] auf Seite 229](#)

this-Eigenschaft

Gibt den Wert des angegebenen Verbindungsschlüsselworts an.

Visual Basic Syntax

```
Public Overrides Property Item(ByVal keyword As String) As Object
```

C#-Syntax

```
public override object this[string keyword] {get;set;}
```

Parameter

- **keyword** Der Name des Verbindungsschlüsselworts.

Bemerkungen

Ein Objekt, das den Wert des angegebenen Verbindungsschlüsselworts repräsentiert.

Verbindungsschlüsselwörter und die entsprechenden Eigenschaften der ULConnectionStringBuilder-Klasse werden in der folgenden Tabelle beschrieben:

Schlüsselwort	Entsprechende Eigenschaft
cache_size	ULConnectionStringBuilder.CacheSize
ce_file	ULConnectionStringBuilder.DatabaseOnDevice
con	ULConnectionStringBuilder.ConnectionName
dbkey	ULConnectionStringBuilder.DatabaseKey
dbn	ULConnectionStringBuilder.DatabaseName
nt_file	ULConnectionStringBuilder.DatabaseOnDesktop
pwd	ULConnectionStringBuilder.Password
reserve_size	ULConnectionStringBuilder.ReserveSize
start	ULConnectionStringBuilder.StartLine
uid	ULConnectionStringBuilder.UserID

Siehe auch

- [ULConnectionStringBuilder.CacheSize-Eigenschaft \[UltraLite.NET\] auf Seite 186](#)
- [ULConnectionStringBuilder.DatabaseOnDevice-Eigenschaft \[UltraLite.NET\] auf Seite 190](#)
- [ULConnectionStringBuilder.ConnectionName-Eigenschaft \[UltraLite.NET\] auf Seite 187](#)
- [ULConnectionStringBuilder.DatabaseKey-Eigenschaft \[UltraLite.NET\] auf Seite 188](#)
- [ULConnectionStringBuilder.DatabaseName-Eigenschaft \[UltraLite.NET\] auf Seite 188](#)
- [ULConnectionStringBuilder.DatabaseOnDesktop-Eigenschaft \[UltraLite.NET\] auf Seite 189](#)
- [ULConnectionStringBuilder.Password-Eigenschaft \[UltraLite.NET\] auf Seite 191](#)
- [ULConnectionStringBuilder.ReserveSize-Eigenschaft \[UltraLite.NET\] auf Seite 191](#)
- [ULConnectionStringBuilder.StartLine-Eigenschaft \[UltraLite.NET\] auf Seite 192](#)
- [ULConnectionStringBuilder.UserID-Eigenschaft \[UltraLite.NET\] auf Seite 194](#)

UserID-Eigenschaft

Legt einen authentifizierten Benutzer für die Datenbank fest.

Visual Basic-Syntax

```
Public Property UserID As String
```

C#-Syntax

```
public string UserID {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die eine Datenbankbenutzer-ID angibt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Benutzer-IDs beachten die Groß- und Kleinschreibung.

Datenbanken werden zunächst mit dem authentifizierten Benutzer DBA erstellt.

Wenn weder die Benutzer-ID noch das Kennwort angegeben werden, wird der Benutzer DBA mit dem Kennwort sql verwendet. Um die Datenbank sicherer zu machen, ändern Sie das Kennwort des DBA-Benutzers oder erstellen Sie neue Benutzer (mit der `ULConnection.GrantConnectTo`-Methode) und entfernen Sie den DBA-Benutzer (mit der `ULConnection.RevokeConnectFrom`-Methode).

Siehe auch

- [ULConnectionStringBuilder.Password-Eigenschaft \[UltraLite.NET\] auf Seite 191](#)
- [ULConnection.GrantConnectTo-Methode \[UltraLite.NET\] auf Seite 150](#)
- [ULConnection.RevokeConnectFrom-Methode \[UltraLite.NET\] auf Seite 152](#)

ULCreateParms-Klasse

UL-Erw.: Erstellt eine Zeichenfolge von Optionen für den Zeitpunkt der Erstellung einer UltraLite-Datenbank.

Visual Basic-Syntax

```
Public Class ULCreateParms
```

C#-Syntax

```
public class ULCreateParms
```

Mitglieder

Alle Mitglieder der ULCreateParms-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ULCreateParms-Konstruktor	Initialisiert ein ULCreateParms-Objekt mit dessen Standardwerten.
ToString-Methode	Gibt die Zeichenfolge zurück, die diese Instanz repräsentiert.
CaseSensitive-Eigenschaft	Legt fest, ob die neue Datenbank beim Vergleich von Zeichenfolgenwerten zwischen Groß- und Kleinschreibung unterscheidet.
ChecksumLevel-Eigenschaft	Gibt die Ebene der Prüfsummen für Datenbankseiten an, die für die neue Datenbank aktiviert ist.
DateFormat-Eigenschaft	Gibt das Datumsformat an, das für Zeichenfolgenkonvertierungen in der neuen Datenbank verwendet wird.
DateOrder-Eigenschaft	Gibt die Datumsreihenfolge an, die für Zeichenfolgenkonvertierungen in der neuen Datenbank verwendet wird.
FIPS-Eigenschaft	Legt fest, ob die neue Datenbank die AES_FIPS- Verschlüsselung oder die AES-Verschlüsselung verwendet.
MaxHashSize-Eigenschaft	Legt die maximale Anzahl von Byte fest, die für die Index-Hashmethode in der neuen Datenbank verwendet werden soll.
NearestCentury-Eigenschaft	Gibt das nächste Jahrhundert an, das für Zeichenfolgenkonvertierungen in der neuen Datenbank verwendet wird.
Obfuscate-Eigenschaft	Gibt an, ob die neue Datenbank Verschleierung verwenden soll, um die Datenbank zu verschlüsseln.
PageSize-Eigenschaft	Gibt die Seitengröße der neuen Datenbank in Byte oder kB an.

Name	Beschreibung
Precision-Eigenschaft	Legt die Gleitkomma-Gesamtstellenzahl fest, die von der Datenbank für Zeichenfolgenkonvertierungen verwendet wird.
Scale-Eigenschaft	Legt die minimale Anzahl der Stellen nach dem Dezimalzeichen fest, wenn ein arithmetisches Ergebnis während Zeichenfolgenkonvertierungen von der neuen Datenbank auf die maximale Gesamtstellenzahl gekürzt wird.
TimeFormat-Eigenschaft	Gibt das Uhrzeitformat an, das für Zeichenfolgenkonvertierungen in der neuen Datenbank verwendet wird.
TimestampFormat-Eigenschaft	Gibt das Zeitstempelformat an, das für Zeichenfolgenkonvertierungen in der neuen Datenbank verwendet wird.
TimestampIncrement-Eigenschaft	Legt die minimale Differenzmenge zwischen eindeutigen Zeitstempeln in Mikrosekunden (Millionstel einer Sekunde) fest.
UTF8Encoding-Eigenschaft	Legt fest, ob die neue Datenbank den UTF8-Zeichensatz oder den der Kollation zugeordneten Zeichensatz verwenden soll.

Bemerkungen

Mit einem `ULCreateParms`-Objekt werden die Parameter zum Erstellen einer Datenbank mit der `ULDatabaseManager.CreateDatabase`-Methode angegeben.

Führende und nachgestellte Leerzeichen werden in allen Zeichenfolgenwerten ignoriert. Werte dürfen keine führenden oder nachgestellten Leerzeichen oder ein Semikolon enthalten oder mit einem Apostroph oder Anführungszeichen beginnen.

Wenn Sie alle Erstellungsparameter durch Festlegen der geeigneten Eigenschaften für ein `ULCreateParms`-Objekt angegeben haben, können Sie mit der `ULCreateParms.ToString`-Methode eine Erstellungsparameter-Zeichenfolge erstellen. Die resultierende Zeichenfolge kann anschließend als `createParms`-Parameter der `ULDatabaseManager.CreateDatabase`-Methode verwendet werden.

Siehe auch

- [ULDatabaseSchema.GetDatabaseProperty](#)-Methode [UltraLite.NET] auf Seite 230
- [ULDatabaseManager.CreateDatabase](#)-Methode [UltraLite.NET] auf Seite 223
- [ULCreateParms.ToString](#)-Methode [UltraLite.NET] auf Seite 197
- „UltraLite-Verbindungsparameter“ [*UltraLite - Datenbankverwaltung*]

Beispiel

Der folgende Code erstellt die Datenbank `\UltraLite\MyDatabase.udb` auf einem Windows Mobile-Gerät. Die Datenbank soll zwischen Groß- und Kleinschreibung unterscheiden und den UTF8-Zeichensatz verwenden.

```
' Visual Basic
Dim createParms As ULCreateParms = New ULCreateParms
createParms.CaseSensitive = True
createParms.UTF8Encoding = True
```

```
Dim openParms As ULConnectionParms = New ULConnectionParms
openParms.DatabaseOnDevice = "\UltraLite\MyDatabase.udb"

ULConnection.DatabaseManager.CreateDatabase( _
    openParms.ToString(), _
    createParms.ToString() _
)

Dim conn As ULConnection = _
    New ULConnection( openParms.ToString() )
conn.Open()
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULCreateParms createParms = new ULCreateParms();
createParms.CaseSensitive = true;
createParms.UTF8Encoding = true;
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnDevice = ".udb";

ULConnection.DatabaseManager.CreateDatabase(
    openParms.ToString(),
    createParms.ToString()
);

ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```

ULCreateParms-Konstruktor

Initialisiert ein ULCreateParms-Objekt mit dessen Standardwerten.

Visual Basic-Syntax

```
Public Sub New()
```

C#-Syntax

```
public ULCreateParms()
```

ToString-Methode

Gibt die Zeichenfolge zurück, die diese Instanz repräsentiert.

Visual Basic-Syntax

```
Public Overrides Function ToString() As String
```

C#-Syntax

```
public override string ToString()
```

Rückgabe

Die Zeichenfolge für diese Instanz ist eine durch Semikola getrennte Liste von Schlüsselwort=Wert-Paaren.

CaseSensitive-Eigenschaft

Legt fest, ob die neue Datenbank beim Vergleich von Zeichenfolgenwerten zwischen Groß- und Kleinschreibung unterscheidet.

Visual Basic-Syntax

```
Public Property CaseSensitive As Boolean
```

C#-Syntax

```
public bool CaseSensitive {get;set;}
```

Bemerkungen

TRUE, wenn die Datenbank die Groß- und Kleinschreibung berücksichtigen soll, andernfalls FALSE. Der Standardwert ist FALSE.

Diese Methode wirkt sich nur darauf aus, wie Zeichenfolgendaten verglichen und sortiert werden. Datenbankbezeichner, wie Tabellennamen, Spaltennamen, Indexnamen und Verbindungsbenutzer-IDs, unterscheiden nie zwischen Groß- und Kleinschreibung. Verbindungskennwörter und Datenbankchiffrierschlüssel unterscheiden immer zwischen Groß- und Kleinschreibung.

ChecksumLevel-Eigenschaft

Gibt die Ebene der Prüfsummen für Datenbankseiten an, die für die neue Datenbank aktiviert ist.

Visual Basic-Syntax

```
Public Property ChecksumLevel As Integer
```

C#-Syntax

```
public int ChecksumLevel {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert ist ungültig.

Bemerkungen

Eine Ganzzahl, die die Ebene der Prüfsummenvalidierung festlegt. Gültige Werte sind 0, 1 und 2. Standardwert ist "0".

DateFormat-Eigenschaft

Gibt das Datumsformat an, das für Zeichenfolgenkonvertierungen in der neuen Datenbank verwendet wird.

Visual Basic-Syntax

```
Public Property DateFormat As String
```

C#-Syntax

```
public string DateFormat {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die das Datumsformat festlegt. Wenn der Wert eine Nullreferenz ist ("Nothing" in Visual Basic), verwendet die Datenbank "JJJJ-MM-TT". In C# muss jedem Backslashzeichen in einem Pfad ein Escapezeichen vorangestellt oder eine von @-Zeichen umschlossene Literal-Zeichenfolge verwendet werden. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

DateOrder-Eigenschaft

Gibt die Datumsreihenfolge an, die für Zeichenfolgenkonvertierungen in der neuen Datenbank verwendet wird.

Visual Basic-Syntax

```
Public Property DateOrder As UDateOrder
```

C#-Syntax

```
public UDateOrder DateOrder {get;set;}
```

Bemerkungen

Ein UDateOrder-Wert, der die Datumsreihenfolge für die Zeichenfolgenkonvertierung angibt. Die Standardeinstellung ist YMD.

Siehe auch

- [UDateOrder-Enumeration \[UltraLite.NET\] auf Seite 458](#)

FIPS-Eigenschaft

Legt fest, ob die neue Datenbank die AES_FIPS- Verschlüsselung oder die AES-Verschlüsselung verwendet.

Visual Basic-Syntax

```
Public Property FIPS As Boolean
```

C#-Syntax

```
public bool FIPS {get;set;}
```

Bemerkungen

TRUE, wenn die Datenbank mit AES_FIPS verschlüsselt werden soll, FALSE, wenn die Datenbank mit AES verschlüsselt werden soll. Der Standardwert ist FALSE.

Die Verschlüsselung muss aktiviert werden, indem beim Erstellen der neuen Datenbank ein Wert für den EncryptionKey-Verbindungsparameter angegeben wird. Wenn FIPS auf TRUE gesetzt wurde und kein Chiffrierschlüssel angegeben wird, schlägt die ULDatabaseManager.CreateDatabase-Methode fehl und es wird ein Fehler bezüglich des fehlenden Chiffrierschlüssels gemeldet.

Siehe auch

- [ULConnectionParms.EncryptionKey-Eigenschaft \[UltraLite.NET\] auf Seite 177](#)
- [ULDatabaseManager.CreateDatabase-Methode \[UltraLite.NET\] auf Seite 223](#)

MaxHashSize-Eigenschaft

Legt die maximale Anzahl von Byte fest, die für die Index-Hashmethode in der neuen Datenbank verwendet werden soll.

Visual Basic-Syntax

```
Public Property MaxHashSize As Integer
```

C#-Syntax

```
public int MaxHashSize {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert ist ungültig.

Bemerkungen

Eine Ganzzahl, die die maximale Hashgröße festlegt. Der Wert muss im Bereich [0,32] liegen. Standardwert ist "8".

NearestCentury-Eigenschaft

Gibt das nächste Jahrhundert an, das für Zeichenfolgenkonvertierungen in der neuen Datenbank verwendet wird.

Visual Basic-Syntax

```
Public Property NearestCentury As Integer
```

C#-Syntax

```
public int NearestCentury {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert ist ungültig.

Bemerkungen

Eine Ganzzahl, die das nächste Jahrhundert angibt. Der Wert muss im Bereich [0,100] liegen.
Standardwert ist "50".

Obfuscate-Eigenschaft

Gibt an, ob die neue Datenbank Verschleierung verwenden soll, um die Datenbank zu verschlüsseln.

Visual Basic-Syntax

```
Public Property Obfuscate As Boolean
```

C#-Syntax

```
public bool Obfuscate {get;set;}
```

Bemerkungen

TRUE, wenn die Datenbank unter Verwendung der Verschleierung verschlüsselt werden soll, andernfalls FALSE. Der Standardwert ist FALSE.

Diese Option wird ignoriert, wenn die FIPS-Verschlüsselung aktiviert ist (ULCreateParms.FIPS-Eigenschaft). Der Chiffrierschlüssel wird ignoriert, wenn die Verschleierung aktiviert ist und beim Erstellen der neuen Datenbank ein Wert für den EncryptionKey-Verbindungsparameter angegeben wird.

Siehe auch

- [ULCreateParms.FIPS-Eigenschaft \[UltraLite.NET\] auf Seite 199](#)

PageSize-Eigenschaft

Gibt die Seitengröße der neuen Datenbank in Byte oder kB an.

Visual Basic-Syntax

```
Public Property PageSize As Integer
```

C#-Syntax

```
public int PageSize {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert ist ungültig.

Bemerkungen

Eine Ganzzahl, die die Seitengröße in Byte festlegt. Gültige Werte sind 1024 (1 kB), 2048 (2 kB), 4096 (4 kB), 8192 (8 kB), 16384 (16 kB). Standardwert ist "4096".

Precision-Eigenschaft

Legt die Gleitkomma-Gesamtstellenzahl fest, die von der Datenbank für Zeichenfolgenkonvertierungen verwendet wird.

Visual Basic-Syntax

```
Public Property Precision As Integer
```

C#-Syntax

```
public int Precision {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert ist ungültig.

Bemerkungen

Eine Ganzzahl, die die Gesamtstellenzahl festlegt. Der Wert muss im Bereich [1,127] liegen. Standardwert ist "30".

Siehe auch

- [ULCreateParams.Scale-Eigenschaft \[UltraLite.NET\] auf Seite 202](#)

Scale-Eigenschaft

Legt die minimale Anzahl der Stellen nach dem Dezimalzeichen fest, wenn ein arithmetisches Ergebnis während Zeichenfolgenkonvertierungen von der neuen Datenbank auf die maximale Gesamtstellenzahl gekürzt wird.

Visual Basic-Syntax

```
Public Property Scale As Integer
```

C#-Syntax

```
public int Scale {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert ist ungültig.

Bemerkungen

Eine Ganzzahl, die die Dezimalstellen festlegt. Der Wert muss im Bereich [0,127] liegen. Standardwert ist "6".

Der Scale-Wert muss kleiner oder gleich dem Precision-Wert sein. Andernfalls tritt beim Erstellen der Datenbank ein Fehler auf.

TimeFormat-Eigenschaft

Gibt das Uhrzeitformat an, das für Zeichenfolgenkonvertierungen in der neuen Datenbank verwendet wird.

Visual Basic-Syntax

```
Public Property TimeFormat As String
```

C#-Syntax

```
public string TimeFormat {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die das Uhrzeitformat festlegt. Wenn der Wert eine Nullreferenz ist ("Nothing" in Visual Basic), verwendet die Datenbank "HH:NN:SS.SSS". In C# muss jedem Backslashzeichen in einem Pfad ein Escapezeichen vorangestellt oder eine von @-Zeichen umschlossene Literal-Zeichenfolge verwendet werden. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

TimestampFormat-Eigenschaft

Gibt das Zeitstempelformat an, das für Zeichenfolgenkonvertierungen in der neuen Datenbank verwendet wird.

Visual Basic-Syntax

```
Public Property TimestampFormat As String
```

C#-Syntax

```
public string TimestampFormat {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert enthielt ein Semikolon oder begann mit einem Apostroph oder Anführungszeichen.

Bemerkungen

Eine Zeichenfolge, die das Zeitstempelformat festlegt. Wenn der Wert eine Nullreferenz ist ("Nothing" in Visual Basic), verwendet die Datenbank "JJJJ-MM-TT HH:NN:SS.SSS". In C# muss jedem Backslashzeichen in einem Pfad ein Escapezeichen vorangestellt oder eine von @-Zeichen umschlossene Literal-Zeichenfolge verwendet werden. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

TimestampIncrement-Eigenschaft

Legt die minimale Differenzmenge zwischen eindeutigen Zeitstempeln in Mikrosekunden (Millionstel einer Sekunde) fest.

Visual Basic-Syntax

```
Public Property TimestampIncrement As Integer
```

C#-Syntax

```
public int TimestampIncrement {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert ist ungültig.

Bemerkungen

Eine Ganzzahl, die das Zeitstempel-Inkrement festlegt. Der Wert muss im Bereich [1,60000000] liegen. Standardwert ist "1".

UTF8Encoding-Eigenschaft

Legt fest, ob die neue Datenbank den UTF8-Zeichensatz oder den der Kollation zugeordneten Zeichensatz verwenden soll.

Visual Basic-Syntax

```
Public Property UTF8Encoding As Boolean
```

C#-Syntax

```
public bool UTF8Encoding {get;set;}
```

Bemerkungen

TRUE, wenn die Datenbank den UTF8-Zeichensatz verwenden soll. FALSE, wenn die Datenbank den der Kollation zugeordneten Zeichensatz verwenden soll. Der Standardwert ist FALSE.

Wählen Sie die Verwendung des UTF8-Zeichensatzes, um Zeichen zu speichern, die in dem der Kollation zugeordneten Zeichensatz nicht enthalten sind. Beispielsweise können Sie eine Datenbank mit der Kollation 1252LATIN1 erstellen, da Sie die US-amerikanische Sortierung verwenden wollen, aber UTF8Encoding auf TRUE setzen, da internationale Adressen in ihrer jeweiligen lokalen Schreibweise gespeichert werden sollen.

ULCursorSchema-Klasse

UL-Erw.: Repräsentiert das Schema eines UltraLite.NET-Cursors.

Visual Basic-Syntax

```
Public MustInherit Class ULCursorSchema
```

C#-Syntax

```
public abstract class ULCursorSchema
```

Abgeleitete Klassen

- [ULResultSetSchema-Klasse \[UltraLite.NET\] auf Seite 375](#)
- [ULTableSchema-Klasse \[UltraLite.NET\] auf Seite 437](#)

Mitglieder

Alle Mitglieder der ULCursorSchema-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
GetColumnID-Methode	Gibt die Spalten-ID der angegebenen Spalte zurück.
GetColumnName-Methode	Gibt den Namen der Spalte zurück, die von der Spalten-ID angegeben wird.
GetColumnPrecision-Methode	Gibt die Gesamtstellenzahl der Spalte zurück, die durch die angegebene Spalten-ID identifiziert wird, wenn die Spalte numerisch ist (SQL-Typ NUMERIC).
GetColumnScale-Methode	Gibt die Anzahl der Dezimalstellen der Spalte zurück, die durch die angegebene Spalten-ID identifiziert wird, wenn die Spalte numerisch ist (SQL-Typ NUMERIC).
GetColumnSize-Methode	Gibt die Größe der Spalte zurück, die durch die angegebene Spalten-ID identifiziert wird, wenn die Spalte eine Größenangabe hat (SQL-Typ BINARY oder CHAR).
GetColumnSQLName-Methode	Gibt den Namen der Spalte zurück, die von der Spalten-ID angegeben wird.
GetColumnULDbType-Methode	Gibt den UltraLite.NET-Datentyp der Spalte zurück, die von der angegebenen Spalten-ID identifiziert wird.
GetSchemaTable-Methode	Gibt ein System.Data.DataTable-Objekt zurück, das das Spaltenschema des ULDataReader-Objekts beschreibt.
ColumnCount-Eigenschaft	Gibt die Anzahl der Spalten im Cursor zurück.
IsOpen-Eigenschaft	Prüft, ob das Cursorschema aktuell geöffnet ist.
Name-Eigenschaft	Gibt den Namen des Cursors zurück.

Bemerkungen

Diese Klasse ist eine abstrakte Basisklasse der Klassen ULTableSchema und ULResultSetSchema.

Hinweis

Für Benutzer, die vom iAnywhere.UltraLite-Namespace portieren, sind Spalten-IDs 0-basiert, nicht 1-basiert, da sie sich im iAnywhere.UltraLite-Namespace befinden.

Siehe auch

- [ULTableSchema-Klasse \[UltraLite.NET\] auf Seite 437](#)
- [ULResultSetSchema-Klasse \[UltraLite.NET\] auf Seite 375](#)

GetColumnID-Methode

Gibt die Spalten-ID der angegebenen Spalte zurück.

Visual Basic-Syntax

```
Public Function GetColumnID(ByVal name As String) As Short
```

C#-Syntax

```
public short GetColumnID(string name)
```

Parameter

- **name** Der Name der Spalte.

Rückgabe

Die Spalten-ID der benannten Spalte.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Spalten-IDs reichen von 0 bis einschließlich ColumnCount-1.

In Ergebnismengen haben nicht alle Spalten Namen und nicht alle Spaltennamen sind eindeutig. Wenn Sie keine Aliasnamen verwenden, wird dem Namen einer nicht berechneten Spalte der Name der Tabelle vorangestellt, aus der die Spalte stammt. Beispiel: MyTable.ID ist der Name der einzigen Spalte in der Ergebnismenge für die Abfrage "SELECT ID FROM MyTable".

Spalten-IDs und -anzahl können sich während eines Schema-Upgrades ändern. Um eine Spalte korrekt zu identifizieren, greifen Sie auf sie anhand des Namens zu oder aktualisieren die IDs und Anzahl im Cache nach jedem Schema-Upgrade.

Siehe auch

- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)

GetColumnName-Methode

Gibt den Namen der Spalte zurück, die von der Spalten-ID angegeben wird.

Visual Basic-Syntax

```
Public Function GetColumnName(ByVal columnID As Integer) As String
```

C#-Syntax

```
public string GetColumnName(int columnID)
```

Parameter

- **columnID** Die ID der Spalte. Der Wert muss im Bereich [0,ColumnCount-1] liegen.

Rückgabe

Der Name der Spalte oder eine Nullreferenz ("Nothing" in Visual Basic), wenn die Spalte keinen Namen hat. Wenn die Spalte in der SQL-Abfrage einen Aliasnamen hat, wird der Aliasname zurückgegeben.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

In Ergebnismengen haben nicht alle Spalten Namen und nicht alle Spaltennamen sind eindeutig. Wenn Sie keine Aliasnamen verwenden, wird dem Namen einer nicht berechneten Spalte der Name der Tabelle vorangestellt, aus der die Spalte stammt. Beispiel: MyTable.ID ist der Name der einzigen Spalte in der Ergebnismenge für die Abfrage "SELECT ID FROM MyTable".

Spalten-IDs und -anzahl können sich während eines Schema-Upgrades ändern. Um eine Spalte korrekt zu identifizieren, greifen Sie auf sie anhand des Namens zu oder aktualisieren die IDs und Anzahl im Cache nach jedem Schema-Upgrade.

Siehe auch

- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)

GetColumnPrecision-Methode

Gibt die Gesamtstellenzahl der Spalte zurück, die durch die angegebene Spalten-ID identifiziert wird, wenn die Spalte numerisch ist (SQL-Typ NUMERIC).

Visual Basic-Syntax

```
Public Function GetColumnPrecision(ByVal columnID As Integer) As Integer
```

C#-Syntax

```
public int GetColumnPrecision(int columnID)
```

Parameter

- **columnID** Die ID der Spalte. Der Wert muss im Bereich [0,ColumnCount-1] liegen.

Rückgabe

Die Anzahl der Dezimalstellen der angegebenen numerischen Spalte

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULCursorSchema.GetColumnULDbType-Methode \[UltraLite.NET\] auf Seite 210](#)
- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)

GetColumnScale-Methode

Gibt die Anzahl der Dezimalstellen der Spalte zurück, die durch die angegebene Spalten-ID identifiziert wird, wenn die Spalte numerisch ist (SQL-Typ NUMERIC).

Visual Basic-Syntax

```
Public Function GetColumnScale(ByVal columnID As Integer) As Integer
```

C#-Syntax

```
public int GetColumnScale(int columnID)
```

Parameter

- **columnID** Die ID der Spalte. Der Wert muss im Bereich [0,ColumnCount-1] liegen.

Rückgabe

Die Dezimalstellen der angegebenen numerischen Spalte

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULCursorSchema.GetColumnULDbType-Methode \[UltraLite.NET\] auf Seite 210](#)
- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)

GetColumnSize-Methode

Gibt die Größe der Spalte zurück, die durch die angegebene Spalten-ID identifiziert wird, wenn die Spalte eine Größenangabe hat (SQL-Typ BINARY oder CHAR).

Visual Basic-Syntax

```
Public Function GetColumnSize(ByVal columnID As Integer) As Integer
```

C#-Syntax

```
public int GetColumnSize(int columnID)
```

Parameter

- **columnID** Die ID der Spalte. Der Wert muss im Bereich [0,ColumnCount-1] liegen.

Rückgabe

Die Größe der angegebenen Spalte mit Größenangabe.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULCursorSchema.GetColumnULDbType-Methode \[UltraLite.NET\] auf Seite 210](#)
- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)

GetColumnSQLName-Methode

Gibt den Namen der Spalte zurück, die von der Spalten-ID angegeben wird.

Visual Basic-Syntax

```
Public Function GetColumnSQLName(ByVal columnID As Integer) As String
```

C#-Syntax

```
public string GetColumnSQLName(int columnID)
```

Parameter

- **columnID** Die ID der Spalte. Der Wert muss im Bereich [0,ColumnCount-1] liegen.

Rückgabe

Der Name der Spalte oder eine Nullreferenz ("Nothing" in Visual Basic), wenn die Spalte keinen Namen hat. Wenn die Spalte in der SQL-Abfrage einen Aliasnamen hat, wird der Aliasname zurückgegeben.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

In Ergebnismengen haben nicht alle Spalten Namen und nicht alle Spaltennamen sind eindeutig. Wenn Sie Aliase verwenden, ist der Name der Spalte der Alias.

Die GetColumnSQLName-Methode unterscheidet sich darin von der GetColumnName-Methode, dass die GetColumnSQLName-Methode für nicht berechnete Spalten ohne Alias immer nur den Spaltennamen zurückgibt (ohne den Tabellennamen als Präfix). Während dieses Verhalten dem Verhalten anderer ADO.NET-Provider ähnlich ist, produziert es wahrscheinlich häufiger nicht-eindeutige Namen.

Spalten-IDs und -anzahl können sich während eines Schema-Upgrades ändern. Um eine Spalte korrekt zu identifizieren, greifen Sie auf sie anhand des Namens zu oder aktualisieren die IDs und Anzahl im Cache nach jedem Schema-Upgrade.

Siehe auch

- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)
- [ULCursorSchema.GetColumnName-Methode \[UltraLite.NET\] auf Seite 207](#)
- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)

GetColumnULDbType-Methode

Gibt den UltraLite.NET-Datentyp der Spalte zurück, die von der angegebenen Spalten-ID identifiziert wird.

Visual Basic-Syntax

```
Public Function GetColumnULDbType(ByVal columnID As Integer) As ULDbType
```

C#-Syntax

```
public ULDbType GetColumnULDbType(int columnID)
```

Parameter

- **columnID** Die ID der Spalte. Der Wert muss im Bereich [0,ColumnCount-1] liegen.

Rückgabe

Eine enumerierte ULDbType-Ganzzahl.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)
- [ULDbType-Enumeration \[UltraLite.NET\] auf Seite 459](#)

GetSchemaTable-Methode

Gibt ein System.Data.DataTable-Objekt zurück, das das Spaltenschema des ULDataReader-Objekts beschreibt.

Visual Basic-Syntax

```
Public Function GetSchemaTable() As DataTable
```

C#-Syntax

```
public DataTable GetSchemaTable()
```

Rückgabe

Ein System.Data.DataTable-Objekt, das das Spaltenschema beschreibt

Siehe auch

- [ULDataReader.GetSchemaTable-Methode \[UltraLite.NET\] auf Seite 258](#)
- [ULDataReader-Klasse \[UltraLite.NET\] auf Seite 237](#)
- [System.Data.DataTable](#)

ColumnCount-Eigenschaft

Gibt die Anzahl der Spalten im Cursor zurück.

Visual Basic-Syntax

```
Public ReadOnly Property ColumnCount As Short
```

C#-Syntax

```
public short ColumnCount {get;}
```

Bemerkungen

Die Anzahl der Spalten im Cursor oder 0, wenn das Cursorschema geschlossen ist

Spalten-IDs reichen von 0 bis einschließlich ColumnCount-1.

Spalten-IDs und -anzahl können sich während eines Schema-Upgrades ändern. Um eine Spalte korrekt zu identifizieren, greifen Sie auf sie anhand des Namens zu oder aktualisieren die IDs und Anzahl im Cache nach jedem Schema-Upgrade.

IsOpen-Eigenschaft

Prüft, ob das Cursorschema aktuell geöffnet ist.

Visual Basic-Syntax

```
Public ReadOnly Property IsOpen As Boolean
```

C#-Syntax

```
public bool IsOpen {get;}
```

Bemerkungen

TRUE, wenn das Cursorschema aktuell geöffnet ist, andernfalls FALSE.

Name-Eigenschaft

Gibt den Namen des Cursors zurück.

Visual Basic-Syntax

```
Public ReadOnly Property Name As String
```

C#-Syntax

```
public abstract string Name {get;}
```

Bemerkungen

Der Name des Cursors als Zeichenfolge.

ULDataAdapter-Klasse

Stellt eine Gruppe von Befehlen und eine Datenbankverbindung dar, die verwendet werden, um ein System.Data.DataSet-Objekt aufzufüllen und eine Datenbank zu aktualisieren.

Visual Basic-Syntax

```
Public NotInheritable Class ULDataAdapter  
    Inherits System.Data.Common.DbDataAdapter  
    Implements System.ICloneable
```

C#-Syntax

```
public sealed class ULDataAdapter :  
    System.Data.Common.DbDataAdapter,  
    System.ICloneable
```

Basisklassen

- [System.Data.Common.DbDataAdapter](#)
- [System.ICloneable](#)

Mitglieder

Alle Mitglieder der ULDataAdapter-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ULDataAdapter-Konstruktor	Initialisiert ein ULDataAdapter-Objekt
AddToBatch-Methode (geerbt aus System.Data.Common.DbDataAdapter)	Fügt dem aktuellen Batch ein System.Data.IDbCommand -Objekt hinzu.
ClearBatch-Methode (Geerbt von System.Data.Common.DbDataAdapter)	Entfernt alle System.Data.IDbCommand -Objekte aus dem Batch.
CreateRowUpdatedEvent-Methode (geerbt aus System.Data.Common.DbDataAdapter)	Initialisiert eine neue Instanz der Klasse System.Data.Common.RowUpdatedEventArgs .

Name	Beschreibung
CreateRowUpdatingEvent-Methode (geerbt aus System.Data.Common.DbDataAdapter)	Initialisiert eine neue Instanz der Klasse System.Data.Common.RowUpdatingEventArgs .
Dispose-Methode (geerbt aus System.Data.Common.DbDataAdapter)	Gibt die unverwalteten Ressourcen frei, die von System.Data.Common.DbDataAdapter verwendet werden, und optional auch die verwalteten Ressourcen.
ExecuteBatch-Methode (geerbt aus System.Data.Common.DbDataAdapter)	Führt den aktuellen Batch aus.
Fill-Methode (geerbt aus System.Data.Common.DbDataAdapter)	Fügt Zeilen zum System.Data.DataSet -Objekt hinzu bzw. aktualisiert sie.
FillSchema-Methode (geerbt aus System.Data.Common.DbDataAdapter)	Fügt ein System.Data.DataTable -Objekt namens "Table" zum angegebenen System.Data.DataSet -Objekt hinzu und konfiguriert das Schema, damit es dem Schema in der Datenquelle entspricht, basierend auf dem angegebenen System.Data.SchemaType -Wert.
GetBatchedParameter-Methode (geerbt aus System.Data.Common.DbDataAdapter)	Gibt einen System.Data.IDataParameter -Wert aus einem der Befehle im aktuellen Batch zurück.
GetBatchedRecordsAffected-Methode (geerbt aus System.Data.Common.DbDataAdapter)	Gibt Informationen über einen einzelnen Aktualisierungsversuche innerhalb einer größeren Batch-Aktualisierung zurück.
GetFillParameters-Methode	Gibt die Parameter zurück, die vom Benutzer definiert werden, wenn er eine SELECT-Anweisung ausführt.
InitializeBatching-Methode (Geerbt von System.Data.Common.DbDataAdapter)	Initialisiert die Batchverarbeitung für den System.Data.Common.DbDataAdapter .
OnRowUpdated-Methode (geerbt aus System.Data.Common.DbDataAdapter)	Löst das RowUpdated-Ereignis in eines .NET Framework-Datenproviders aus.
OnRowUpdating-Methode (geerbt aus System.Data.Common.DbDataAdapter)	Löst das RowUpdating-Ereignis in eines .NET Framework-Datenproviders aus.

Name	Beschreibung
TerminateBatching-Methode (Geerbt von System.Data.Common.DbDataAdapter)	Beendet die Batchverarbeitung für den System.Data.Common.DbDataAdapter .
Update-Methode (geerbt aus System.Data.Common.DbDataAdapter)	Ruft die entsprechenden INSERT-, UPDATE- oder DELETE-Anweisungen für jede eingefügte, aktualisierte oder gelöschte Zeile im angegebenen Array von System.Data.DataRow -Objekten auf.
DeleteCommand-Eigenschaft	Gibt ein ULCommand-Objekt an, das in der Datenbank ausgeführt wird, wenn die DbDataAdapter.Update-Methode aufgerufen wurde, um Zeilen aus der Datenbank zu löschen, die gelöschten Zeilen im System.Data.DataSet-Objekt entsprechen.
FillCommandBehavior-Eigenschaft (geerbt aus System.Data.Common.DbDataAdapter)	Ruft das Verhalten des Befehls ab, der zum Füllen des Datenadapters verwendet wird, bzw. legt es fest.
InsertCommand-Eigenschaft	Gibt ein ULCommand-Objekt an, das in der Datenbank ausgeführt wird, wenn die DbDataAdapter.Update-Methode aufgerufen wurde, um Zeilen in die Datenbank einzufügen, die eingefügten Zeilen im System.Data.DataSet-Objekt entsprechen.
SelectCommand-Eigenschaft	Gibt ein ULCommand-Objekt an, das während eines Aufrufs der Methode System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet) oder System.Data.Common.DbDataAdapter.FillSchema(System.Data.DataSet, System.Data.SchemaType) verwendet wird, um eine Ergebnismenge aus der Datenbank abzurufen, die in das System.Data.DataSet-Objekt kopiert werden soll.
TableMappings-Eigenschaft	Gibt eine Sammlung zurück, die die Hauptzuordnung zwischen einer Quelltable und einem System.Data.DataTable-Objekt bereitstellt.
UpdateBatchSize-Eigenschaft (Geerbt von System.Data.Common.DbDataAdapter)	Ruft einen Wert ab, der die Unterstützung für die Batchverarbeitung aktiviert oder deaktiviert und die Anzahl der Befehle festlegt, die in einem Batch verarbeitet werden können, oder legt diesen Wert fest.
UpdateCommand-Eigenschaft	Gibt ein ULCommand-Objekt an, das in der Datenbank ausgeführt wird, wenn die System.Data.Common.DbDataAdapter.Update-Methode aufgerufen wurde, um Zeilen in der Datenbank zu aktualisieren, die aktualisierten Zeilen im System.Data.DataSet-Objekt entsprechen.

Name	Beschreibung
RowUpdated-Ereignis	Tritt während einer Aktualisierung auf, nachdem ein Befehl in einer Datenquelle ausgeführt wurde.
RowUpdating-Ereignis	Tritt während einer Aktualisierung auf, bevor ein Befehl in einer Datenquelle ausgeführt wurde.
DefaultSourceTableName-Feld (Geerbt von System.Data.Common.DbDataAdapter)	Der Standardname, der vom System.Data.Common.DataAdapter -Objekt für Tabellenzuordnungen verwendet wird.

Bemerkungen

Das System.Data.DataSet-Objekt ermöglicht die Verarbeitung von Daten offline, d.h. wenn keine Verbindung zur UltraLite-Datenbank besteht. Die ULDataAdapter-Klasse bietet Methoden, um ein System.Data.DataSet-Objekt einer Gruppe von SQL-Anweisungen zuzuordnen.

Da UltraLite eine lokale Datenbank ist und MobiLink mit Konfliktlösung arbeitet, kann ULDataAdapter nur beschränkt eingesetzt werden. Für die meisten Anwendungszwecke ermöglicht die ULDataReader- oder ULTable-Klasse einen effizienteren Zugriff auf Daten.

Siehe auch

- [ULDataReader-Klasse \[UltraLite.NET\] auf Seite 237](#)
- [ULTable-Klasse \[UltraLite.NET\] auf Seite 414](#)
- [System.Data.DataSet](#)
- [System.Data.Common.DbDataAdapter](#)
- [System.Data.IDbDataAdapter](#)
- [System.Data.IDataAdapter](#)
- [System.IDisposable](#)

ULDataAdapter-Konstruktor

Initialisiert ein ULDataAdapter-Objekt

Überladungsliste

Name	Beschreibung
ULDataAdapter()-Konstruktor	Initialisiert ein ULDataAdapter-Objekt
ULDataAdapter(string, string)-Konstruktor	Initialisiert ein ULDataAdapter-Objekt mit dem angegebenen SELECT-Befehl und der Verbindungszeichenfolge.
ULDataAdapter(string, ULConnection)-Konstruktor	Initialisiert ein ULDataAdapter-Objekt mit dem angegebenen SELECT-Befehl und der Verbindung

Name	Beschreibung
ULDataAdapter(ULCommand)-Konstruktor	Initialisiert ein ULDataAdapter-Objekt mit dem angegebenen SELECT-Befehl.

ULDataAdapter()-Konstruktor

Initialisiert ein ULDataAdapter-Objekt

Visual Basic-Syntax

```
Public Sub New()
```

C#-Syntax

```
public ULDataAdapter()
```

Siehe auch

- [ULDataAdapter.ULDataAdapter-Konstruktor \[UltraLite.NET\] auf Seite 215](#)

ULDataAdapter(string, string)-Konstruktor

Initialisiert ein ULDataAdapter-Objekt mit dem angegebenen SELECT-Befehl und der Verbindungszeichenfolge.

Visual Basic-Syntax

```
Public Sub New(  
    ByVal selectCommandText As String,  
    ByVal selectConnectionString As String  
)
```

C#-Syntax

```
public ULDataAdapter(  
    string selectCommandText,  
    string selectConnectionString  
)
```

Parameter

- **selectCommandText** Eine SELECT-Anweisung, die von der ULDataAdapter.SelectCommand-Methode verwendet werden soll.
- **selectConnectionString** Eine Verbindungszeichenfolge für eine UltraLite.NET-Datenbank.

Siehe auch

- [ULDataAdapter.ULDataAdapter-Konstruktor \[UltraLite.NET\] auf Seite 215](#)
- [ULDataAdapter.SelectCommand-Eigenschaft \[UltraLite.NET\] auf Seite 219](#)

ULDataAdapter(string, ULConnection)-Konstruktor

Initialisiert ein ULDataAdapter-Objekt mit dem angegebenen SELECT-Befehl und der Verbindung

Visual Basic-Syntax

```
Public Sub New(  
    ByVal selectCommandText As String,  
    ByVal selectConnection As ULConnection  
)
```

C#-Syntax

```
public ULDataAdapter(  
    string selectCommandText,  
    ULConnection selectConnection  
)
```

Parameter

- **selectCommandText** Eine SELECT-Anweisung, die von der ULDataAdapter.SelectCommand-Methode des ULDataAdapter-Objekts verwendet werden soll.
- **selectConnection** Ein ULConnection-Objekt, das eine Verbindung mit einer Datenbank festlegt.

Siehe auch

- [ULDataAdapter.ULDataAdapter-Konstruktor \[UltraLite.NET\] auf Seite 215](#)
- [ULDataAdapter.SelectCommand-Eigenschaft \[UltraLite.NET\] auf Seite 219](#)
- [ULConnection-Klasse \[UltraLite.NET\] auf Seite 121](#)

ULDataAdapter(ULCommand)-Konstruktor

Initialisiert ein ULDataAdapter-Objekt mit dem angegebenen SELECT-Befehl.

Visual Basic-Syntax

```
Public Sub New(ByVal selectCommand As ULCommand)
```

C#-Syntax

```
public ULDataAdapter(ULCommand selectCommand)
```

Parameter

- **selectCommand** Ein ULCommand-Objekt, das während der Ausführung von System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet) verwendet wird, um Datensätze aus der Datenquelle für die Platzierung im System.Data.DataSet-Objekt auszuwählen.

Siehe auch

- [ULDataAdapter.ULDataAdapter-Konstruktor \[UltraLite.NET\] auf Seite 215](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [System.Data.DataSet](#)

GetFillParameters-Methode

Gibt die Parameter zurück, die vom Benutzer definiert werden, wenn er eine SELECT-Anweisung ausführt.

Visual Basic-Syntax

```
Public Shadows Function GetFillParameters() As ULParameter()
```

C#-Syntax

```
public new ULParameter[] GetFillParameters()
```

Rückgabe

Ein Array von ULParameter-Objekten, das die vom Benutzer definierten Parameter enthält.

Bemerkungen

Dies ist die stark typisierte Version der System.Data.Common.DbDataAdapter.GetFillParameters-Methode.

Siehe auch

- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)
- [System.Data.Common.DbDataAdapter.GetFillParameters](#)

DeleteCommand-Eigenschaft

Gibt ein ULCommand-Objekt an, das in der Datenbank ausgeführt wird, wenn die DbDataAdapter.Update-Methode aufgerufen wurde, um Zeilen aus der Datenbank zu löschen, die gelöschten Zeilen im System.Data.DataSet-Objekt entsprechen.

Visual Basic-Syntax

```
Public Shadows Property DeleteCommand As ULCommand
```

C#-Syntax

```
public new ULCommand DeleteCommand {get;set;}
```

Bemerkungen

Ein ULCommand-Objekt, das ausgeführt wird, um Zeilen in der Datenbank zu löschen, die gelöschten Zeilen im System.Data.DataSet-Objekt entsprechen.

Wenn die DeleteCommand-Eigenschaft einem vorhandenen ULCommand-Objekt zugeordnet ist, wird das ULCommand-Objekt nicht geklont. Die DeleteCommand-Eigenschaft behält eine Referenz zum vorhandenen ULCommand-Objekt bei.

Dies ist die stark typisierte Version der Eigenschaften System.Data.IDbDataAdapter.DeleteCommand und System.Data.Common.DbDataAdapter.DeleteCommand.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [System.Data.DataSet](#)
- [System.Data.IDbDataAdapter.DeleteCommand](#)
- [System.Data.Common.DbDataAdapter.DeleteCommand](#)

InsertCommand-Eigenschaft

Gibt ein ULCommand-Objekt an, das in der Datenbank ausgeführt wird, wenn die DbDataAdapter.Update-Methode aufgerufen wurde, um Zeilen in die Datenbank einzufügen, die eingefügten Zeilen im System.Data.DataSet-Objekt entsprechen.

Visual Basic-Syntax

```
Public Shadows Property InsertCommand As ULCommand
```

C#-Syntax

```
public new ULCommand InsertCommand {get;set;}
```

Bemerkungen

Ein ULCommand-Objekt, das ausgeführt wird, um Zeilen in die Datenbank einzufügen, die eingefügten Zeilen im System.Data.DataSet-Objekt entsprechen

Wenn die InsertCommand-Eigenschaft einem vorhandenen ULCommand-Objekt zugeordnet ist, wird das ULCommand-Objekt nicht geklont. Die InsertCommand-Eigenschaft behält eine Referenz zum vorhandenen ULCommand-Objekt bei.

Dies ist die stark typisierte Version der Eigenschaften System.Data.IDbDataAdapter.InsertCommand und System.Data.Common.DbDataAdapter.InsertCommand.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [System.Data.DataSet](#)
- [System.Data.IDbDataAdapter.InsertCommand](#)
- [System.Data.Common.DbDataAdapter.InsertCommand](#)

SelectCommand-Eigenschaft

Gibt ein ULCommand-Objekt an, das während eines Aufrufs der Methode System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet) oder System.Data.Common.DbDataAdapter.FillSchema(System.Data.DataSet, System.Data.SchemaType) verwendet wird, um eine Ergebnismenge aus der Datenbank abzurufen, die in das System.Data.DataSet-Objekt kopiert werden soll.

Visual Basic-Syntax

```
Public Shadows Property SelectCommand As ULCommand
```

C#-Syntax

```
public new ULCommand SelectCommand {get;set;}
```

Bemerkungen

Ein ULCommand-Objekt, das ausgeführt wird, um das System.Data.DataSet-Objekt aufzufüllen.

Wenn die SelectCommand-Eigenschaft einem vorhandenen ULCommand-Objekt zugeordnet ist, wird das ULCommand-Objekt nicht geklont. Die SelectCommand-Eigenschaft behält eine Referenz zum vorhandenen ULCommand-Objekt bei.

Wenn die SelectCommand-Eigenschaft keine Zeilen zurückgibt, werden dem System.Data.DataSet-Objekt keine Tabellen hinzugefügt und es wird keine Ausnahmebedingung ausgelöst. Die SELECT-Anweisung kann auch in den Konstruktoren ULDataAdapter(ULCommand), ULDataAdapter(String,ULConnection) oder ULDataAdapter(String,String) angegeben werden.

Dies ist die stark typisierte Version der Eigenschaften System.Data.IDbDataAdapter.SelectCommand und System.Data.Common.DbDataAdapter.SelectCommand.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULDataAdapter.ULDataAdapter-Konstruktor \[UltraLite.NET\] auf Seite 215](#)
- [System.Data.DataSet](#)
- [System.Data.IDbDataAdapter.SelectCommand](#)
- [System.Data.Common.DbDataAdapter.SelectCommand](#)

TableMappings-Eigenschaft

Gibt eine Sammlung zurück, die die Hauptzuordnung zwischen einer Quelltable und einem System.Data.DataTable-Objekt bereitstellt.

Visual Basic-Syntax

```
Public ReadOnly Shadows Property TableMappings As  
DataTableMappingCollection
```

C#-Syntax

```
public new DataTableMappingCollection TableMappings {get;}
```

Bemerkungen

Eine Sammlung von System.Data.Common.DataTableMapping-Objekten, die die Hauptzuordnung zwischen Quelltabellen und System.Data.DataTables bereitstellen. Der Standardwert ist eine leere Sammlung.

Beim Abgleichen von Änderungen verwendet das ULDataAdapter-Objekt die System.Data.Common.DataTableMappingCollection-Sammlung, um die von der Datenquelle verwendeten Spaltennamen den vom System.Data.DataSet-Objekt verwendeten Spaltennamen zuzuordnen.

Dies ist die stark typisierte Version der `System.Data.IDataAdapter.TableMappings`-Eigenschaft.

Siehe auch

- [System.Data.DataTable](#)
- [System.Data.Common.DataTableMapping](#)
- [System.Data.DataTable](#)
- [System.Data.Common.DataTableMappingCollection](#)
- [System.Data.DataSet](#)
- [System.Data.IDataAdapter.TableMappings](#)

UpdateCommand-Eigenschaft

Gibt ein `ULCommand`-Objekt an, das in der Datenbank ausgeführt wird, wenn die `System.Data.Common.DbDataAdapter.Update`-Methode aufgerufen wurde, um Zeilen in der Datenbank zu aktualisieren, die aktualisierten Zeilen im `System.Data.DataSet`-Objekt entsprechen.

Visual Basic-Syntax

```
Public Shadows Property UpdateCommand As ULCommand
```

C#-Syntax

```
public new ULCommand UpdateCommand {get;set;}
```

Bemerkungen

Ein `ULCommand`-Objekt, das ausgeführt wird, um Zeilen in der Datenbank zu aktualisieren, die aktualisierten Zeilen im `System.Data.DataSet`-Objekt entsprechen

Wenn `UpdateCommand` einem bestehenden `ULCommand`-Objekt zugewiesen wird, wird das `ULCommand`-Objekt nicht geklont. Die `UpdateCommand`-Eigenschaft behält eine Referenz zum vorhandenen `ULCommand`-Objekt bei.

Wenn die Ausführung dieses Befehls Zeilen zurückgibt, können diese Zeilen mit dem `System.Data.DataSet`-Objekt zusammengeführt werden, je nachdem, wie Sie die `ULCommand.UpdatedRowSource`-Eigenschaft für das `ULCommand`-Objekt festlegen.

Dies ist die stark typisierte Version der Eigenschaften `System.Data.IDbDataAdapter.UpdateCommand` und `System.Data.Common.DbDataAdapter.UpdateCommand`.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULCommand.UpdatedRowSource-Eigenschaft \[UltraLite.NET\] auf Seite 109](#)
- [System.Data.DataSet](#)
- [System.Data.IDbDataAdapter.UpdateCommand](#)
- [System.Data.Common.DbDataAdapter.DeleteCommand](#)

RowUpdated-Ereignis

Tritt während einer Aktualisierung auf, nachdem ein Befehl in einer Datenquelle ausgeführt wurde.

Visual Basic-Syntax

```
Public Event RowUpdated As ULRowUpdatedEventHandler
```

C#-Syntax

```
public event ULRowUpdatedEventHandler RowUpdated;
```

Bemerkungen

Das Ereignis wird ausgelöst, wenn eine Aktualisierung versucht wird.

Zur Verarbeitung von Ereignissen zur Zeilenaktualisierung müssen Sie einen ULRowUpdatedEventHandler-Delegaten erstellen und diesem Ereignis zuordnen.

Siehe auch

- [ULRowUpdatedEventHandler-Delegat \[UltraLite.NET\] auf Seite 454](#)

RowUpdating-Ereignis

Tritt während einer Aktualisierung auf, bevor ein Befehl in einer Datenquelle ausgeführt wurde.

Visual Basic-Syntax

```
Public Event RowUpdating As ULRowUpdatingEventHandler
```

C#-Syntax

```
public event ULRowUpdatingEventHandler RowUpdating;
```

Bemerkungen

Das Ereignis wird ausgelöst, wenn eine Aktualisierung versucht wird.

Zur Verarbeitung von Ereignissen zur Zeilenaktualisierung müssen Sie einen ULRowUpdatingEventHandler-Delegaten erstellen und diesem Ereignis zuordnen.

Siehe auch

- [ULRowUpdatedEventHandler-Delegat \[UltraLite.NET\] auf Seite 454](#)

ULDatabaseManager-Klasse

UL-Erw.: Stellt statische Methoden für das Erstellen, Löschen und Validieren von Datenbanken bereit.

Visual Basic-Syntax

```
Public NotInheritable Class ULDatabaseManager
```

C#-Syntax

```
public sealed class ULDatabaseManager
```

Mitglieder

Alle Mitglieder der ULDatabaseManager-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
CreateDatabase-Methode	Erstellt eine neue UltraLite-Datenbank.
DropDatabase-Methode	Löscht die angegebene Datenbank
SetActiveSyncListener-Methode	Gibt das Listener-Objekt an, das ActiveSync-Aufrufe vom MobiLink-Provider für ActiveSync verarbeiten soll
SetServerSyncListener-Methode	Gibt das Listener-Objekt an, das zur Verarbeitung der angegebenen Nachricht der Serversynchronisation verwendet wird
SignalSyncIsComplete-Methode	Signalisiert dem MobiLink-Provider für ActiveSync, dass eine Anwendung die Synchronisation abgeschlossen hat.
ValidateDatabase-Methode	Führt eine Validierung der Basisdaten und eine Indexvalidierung in einer Datenbank durch.
RuntimeType-Eigenschaft	Gibt den Typ der UltraLite.NET-Laufzeitumgebung an.

Bemerkungen

Um die UltraLite-Engine-Laufzeitumgebung von UltraLite.NET zu verwenden, setzen Sie die ULDatabaseManager.RuntimeType-Eigenschaft auf den zutreffenden Wert, bevor Sie eine andere UltraLite.NET-API verwenden.

Beispiel

Das folgende Beispiel wählt die UltraLite-Laufzeit-Engine und erstellt eine Verbindung:

```
' Visual Basic
ULDatabaseManager.RuntimeType = ULRuntimeType.UL_ENGINE_CLIENT
Dim conn As ULConnection = new ULConnection
' The RuntimeType is now locked
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULDatabaseManager.RuntimeType = ULRuntimeType.UL_ENGINE_CLIENT;
ULConnection conn = new ULConnection();
// The RuntimeType is now locked
```

CreateDatabase-Methode

Erstellt eine neue UltraLite-Datenbank.

Visual Basic-Syntax

```
Public Shared Sub CreateDatabase(  
    ByVal connString As String,  
    ByVal createParms As String  
)
```

C#-Syntax

```
public static void CreateDatabase(string connString, string createParms)
```

Parameter

- **connString** Die Parameter zum Identifizieren einer Datenbank in Form einer durch Semikolons getrennten Liste von Schlüsselwort-Wert-Paaren.
- **createParms** Die Parameter zum Konfigurieren der neuen Datenbank in Form einer durch Semikolons getrennten Liste von Schlüsselwort-Wert-Paaren.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULConnection.Open-Methode \[UltraLite.NET\] auf Seite 150](#)
- [ULConnectionParms-Klasse \[UltraLite.NET\] auf Seite 169](#)
- [ULCreateParms-Klasse \[UltraLite.NET\] auf Seite 195](#)

Beispiel

Der folgende Code erstellt die Datenbank \UltraLite\MyDatabase.udb auf einem Windows Mobile-Gerät und öffnet dann eine Verbindung zu ihr.

```
' Visual Basic  
Dim openParms As ULConnectionParms = New ULConnectionParms  
openParms.DatabaseOnDevice = "\UltraLite\MyDatabase.udb"  
ULConnection.DatabaseManager.CreateDatabase( _  
    openParms.ToString(), _  
    "" _  
)  
Dim conn As ULConnection = _  
    New ULConnection( openParms.ToString() )  
conn.Open()
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#  
ULConnectionParms openParms = new ULConnectionParms();  
openParms.DatabaseOnDevice = ".udb";  
  
ULConnection.DatabaseManager.CreateDatabase(  
    openParms.ToString(),  
    ""  
);  
ULConnection conn = new ULConnection( openParms.ToString() );  
conn.Open();
```

DropDatabase-Methode

Löscht die angegebene Datenbank

Visual Basic-Syntax

```
Public Shared Sub DropDatabase(ByVal connString As String)
```

C#-Syntax

```
public static void DropDatabase(string connString)
```

Parameter

- **connString** Die Parameter zum Identifizieren einer Datenbank in Form einer durch Semikolons getrennten Liste von Schlüsselwort-Wert-Paaren.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Sie können keine Datenbank löschen, die offene Verbindungen hat.

Siehe auch

- [ULConnection.Open-Methode \[UltraLite.NET\] auf Seite 150](#)
- [ULConnectionParms-Klasse \[UltraLite.NET\] auf Seite 169](#)

Beispiel

Der folgende Code erstellt die Datenbank \UltraLite\MyDatabase.udb auf einem Windows Mobile-Gerät und öffnet dann eine Verbindung zu ihr:

```
' Visual Basic
Dim connParms As ULConnectionParms = New ULConnectionParms
connParms.DatabaseOnDevice = "\UltraLite\MyDatabase.udb"
ULConnection.DatabaseManager.DropDatabase( _
    connParms.ToString() _
)
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULConnectionParms connParms = new ULConnectionParms();
connParms.DatabaseOnDevice = ".udb";
ULConnection.DatabaseManager.DropDatabase(
    connParms.ToString()
);
ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```

SetActiveSyncListener-Methode

Gibt das Listener-Objekt an, das ActiveSync-Aufrufe vom MobiLink-Provider für ActiveSync verarbeiten soll

Visual Basic-Syntax

```
Public Shared Sub SetActiveSyncListener(  
    ByVal appClassName As String,  
    ByVal listener As UActiveSyncListener  
)
```

C#-Syntax

```
public static void SetActiveSyncListener(  
    string appClassName,  
    UActiveSyncListener listener  
)
```

Parameter

- **appClassName** Der eindeutige Klassenname für die Anwendung. Dies ist der Klassenname, der bei der Registrierung der Anwendung für die Verwendung mit ActiveSync verwendet wird.
- **listener** Das UActiveSyncListener-Objekt. Verwenden Sie Null ("Nothing" in Visual Basic), um den vorherigen Listener zu entfernen.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Der *appClassName*-Parameter ist ein eindeutiger Bezeichner zum Identifizieren der Anwendung. Die Anwendung kann jeweils nur einen *appClassName*-Wert verwenden. Wenn ein Listener mit einem bestimmten *appClassName*-Wert registriert ist, schlagen Aufrufe der Methode SetServerSyncListener oder SetActiveSyncListener mit einem anderen *appClassName*-Wert fehl.

Um den ActiveSync-Listener zu entfernen, rufen Sie die SetActiveSyncListener-Methode mit einer Nullreferenz ("Nothing" in Visual Basic) als *listener*-Parameter auf.

Um alle Listener zu entfernen, rufen Sie die SetServerSyncListener-Methode mit einer Nullreferenz ("Nothing" in Visual Basic) für alle Parameter auf.

Anwendungen sollten vor ihrer Beendigung alle Listener entfernen.

Siehe auch

- [ULDatabaseManager.SetServerSyncListener-Methode \[UltraLite.NET\] auf Seite 226](#)
- [ULDatabaseManager.SetActiveSyncListener-Methode \[UltraLite.NET\] auf Seite 225](#)
- [UActiveSyncListener-Schnittstelle \[UltraLite.NET\] auf Seite 44](#)
- [UActiveSyncListener.ActiveSyncInvoked-Methode \[UltraLite.NET\] auf Seite 44](#)

SetServerSyncListener-Methode

Gibt das Listener-Objekt an, das zur Verarbeitung der angegebenen Nachricht der Serversynchronisation verwendet wird

Visual Basic-Syntax

```
Public Shared Sub SetServerSyncListener(  
    ByVal messageName As String,  
    ByVal appClassName As String,  
    ByVal listener As ULServerSyncListener  
)
```

C#-Syntax

```
public static void SetServerSyncListener(  
    string messageName,  
    string appClassName,  
    ULServerSyncListener listener  
)
```

Parameter

- **messageName** Der Name der Nachricht
- **appClassName** Der eindeutige Klassenname für die Anwendung. Dies ist ein eindeutiger Bezeichner zur Kennzeichnung der Anwendung.
- **listener** Das ULServerSyncListener-Objekt. Verwenden Sie Null ("Nothing" in Visual Basic), um den vorherigen Listener zu entfernen.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Der *appClassName*-Parameter ist ein eindeutiger Bezeichner zum Identifizieren der Anwendung. Die Anwendung kann jeweils nur einen *appClassName*-Wert verwenden. Wenn ein Listener mit einem bestimmten *appClassName*-Wert registriert ist, schlagen Aufrufe der Methode `SetServerSyncListener` oder `SetActiveSyncListener` mit einem anderen *appClassName*-Wert fehl.

Wenn der Listener für eine bestimmte Nachricht entfernt werden soll, rufen Sie die `SetServerSyncListener`-Methode mit einer Nullreferenz ("Nothing" in Visual Basic) als *listener*-Parameter auf.

Um alle Listener zu entfernen, rufen Sie die `SetServerSyncListener`-Methode mit einer Nullreferenz ("Nothing" in Visual Basic) für alle Parameter auf.

Anwendungen sollten vor ihrer Beendigung alle Listener entfernen.

Siehe auch

- [ULDatabaseManager.SetServerSyncListener-Methode \[UltraLite.NET\] auf Seite 226](#)
- [ULDatabaseManager.SetActiveSyncListener-Methode \[UltraLite.NET\] auf Seite 225](#)
- [ULServerSyncListener.ServerSyncInvoked-Methode \[UltraLite.NET\] auf Seite 384](#)

SignalSyncIsComplete-Methode

Signalisiert dem MobiLink-Provider für ActiveSync, dass eine Anwendung die Synchronisation abgeschlossen hat.

Visual Basic-Syntax

```
Public Shared Sub SignalSyncIsComplete()
```

C#-Syntax

```
public static void SignalSyncIsComplete()
```

Siehe auch

- [ULDatabaseManager.SignalSyncIsComplete-Methode \[UltraLite.NET\] auf Seite 228](#)
- [ULActiveSyncListener.ActiveSyncInvoked-Methode \[UltraLite.NET\] auf Seite 44](#)

ValidateDatabase-Methode

Führt eine Validierung der Basisdaten und eine Indexvalidierung in einer Datenbank durch.

Visual Basic-Syntax

```
Public Shared Sub ValidateDatabase(  
    ByVal start_parms As String,  
    ByVal how As ULDBValid  
)
```

C#-Syntax

```
public static void ValidateDatabase(string start_parms, ULDBValid how)
```

Parameter

- **start_parms** Die Parameter zum Identifizieren einer Datenbank in Form einer durch Semikolons getrennten Liste von Schlüsselwort-Wert-Paaren.
- **how** Beschreibt, wie die Validierung der Datenbank erfolgen soll.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULConnection.ValidateDatabase-Methode \[UltraLite.NET\] auf Seite 158](#)
- [ULDBValid-Enumeration \[UltraLite.NET\] auf Seite 458](#)
- [ULConnectionParms-Klasse \[UltraLite.NET\] auf Seite 169](#)

Beispiel

Der folgende Code validiert Indizes für die Datenbank \UltraLite\MyDatabase.udb unter Windows Mobile.

```
' Visual Basic
Dim openParms As ULConnectionParms = New ULConnectionParms
openParms.DatabaseOnDevice = "\UltraLite\MyDatabase.udb"
ULConnection.DatabaseManager.ValidateDatabase( _
    openParms.ToString(), iAnywhere.Data.UltraLite.ULVF_INDEX )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnDevice = ".udb";
ULConnection.DatabaseManager.ValidateDatabase(
    openParms.ToString(), iAnywhere.Data.UltraLite.ULVF_INDEX );
```

RuntimeType-Eigenschaft

Gibt den Typ der UltraLite.NET-Laufzeitumgebung an.

Visual Basic-Syntax

```
Public Shared Property RuntimeType As ULRuntimeType
```

C#-Syntax

```
public ULRuntimeType RuntimeType {get;set;}
```

Bemerkungen

Der Laufzeittyp muss vor der Verwendung einer anderen UltraLite.NET API ausgewählt werden.

Ein ULRuntimeType-Wert, der den Typ der nicht verwalteten UltraLite.NET-Laufzeitumgebung kennzeichnet.

Siehe auch

- [ULRuntimeType-Enumeration \[UltraLite.NET\] auf Seite 463](#)

Beispiel

Das folgende Beispiel wählt die UltraLite-Laufzeit-Engine und erstellt eine Verbindung:

```
' Visual Basic
ULDatabaseManager.RuntimeType = ULRuntimeType.UL_ENGINE_CLIENT
Dim conn As ULConnection = new ULConnection
' The RuntimeType is now locked
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULDatabaseManager.RuntimeType = ULRuntimeType.UL_ENGINE_CLIENT;
ULConnection conn = new ULConnection();
// The RuntimeType is now locked
```

ULDatabaseSchema-Klasse

UL-Erw.: Repräsentiert das Schema einer UltraLite.NET-Datenbank.

Visual Basic-Syntax

```
Public NotInheritable Class ULDatabaseSchema
```

C#-Syntax

```
public sealed class ULDatabaseSchema
```

Mitglieder

Alle Mitglieder der ULDatabaseSchema-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
GetDatabaseProperty-Methode	Gibt den Wert der angegebenen Datenbankeigenschaften zurück.
GetPublicationName-Methode	Gibt den Namen der Publikation, die mithilfe der Publikations-ID angegeben wurde, zurück.
GetTableName-Methode	Gibt den Namen der Tabelle zurück, deren Tabellen-ID angegeben wurde.
SetDatabaseOption-Methode	Setzt den Wert für den angegebenen Datenbankparameter fest.
IsCaseSensitive-Eigenschaft	Prüft, ob die Datenbank zwischen Groß- und Kleinschreibung unterscheidet.
IsOpen-Eigenschaft	Ermittelt, ob das Datenbankschema geöffnet ist.
PublicationCount-Eigenschaft	Ruft die Anzahl der Publikationen in der Datenbank ab.
TableCount-Eigenschaft	Ruft die Anzahl der Tabellen in der Datenbank ab.

Bemerkungen

Es gibt keinen Konstruktor für diese Klasse. Ein ULDatabaseSchema-Objekt ist einer Verbindung als ihr ULConnection.Schema-Objekt zugeordnet und nur gültig, wenn diese Verbindung geöffnet ist.

Siehe auch

- [ULDatabaseSchema-Klasse \[UltraLite.NET\] auf Seite 229](#)
- [ULConnection.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 164](#)

GetDatabaseProperty-Methode

Gibt den Wert der angegebenen Datenbankeigenschaften zurück.

Visual Basic-Syntax

```
Public Function GetDatabaseProperty(ByVal name As String) As String
```

C#-Syntax

```
public string GetDatabaseProperty(string name)
```

Parameter

- **name** Der Name der Datenbankeigenschaft, dessen Wert Sie erhalten wollen. Eigenschaftsnamen reagieren nicht auf Groß- und Kleinschreibung.

Rückgabe

Der Wert der Eigenschaft als Zeichenfolge.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Erkannte Eigenschaften sind:

Eigenschaft	Beschreibung
CaseSensitive	Der Status der Funktion für die Berücksichtigung der Groß- und Kleinschreibung. Gibt ON zurück, wenn in der Datenbank die Groß- und Kleinschreibung aktiviert ist. Andernfalls wird OFF zurückgegeben. Die Berücksichtigung der Groß- und Kleinschreibung von Datenbanken hat Auswirkungen darauf, wie Indizes für Tabellen und Ergebnismengen sortiert werden. Die Berücksichtigung der Groß- und Kleinschreibung hat auch Auswirkungen darauf, wie die ULConnection-Parms.UserID- und ULConnectionParms.Password-Werte einer Verbindung überprüft werden. Benutzer-IDs berücksichtigen nie die Groß- und Kleinschreibung, während Kennwörter die Groß- und Kleinschreibung immer berücksichtigen.
CharSet	Der Zeichensatz der Datenbank.
ChecksumLevel	Die Ebene der für die Datenbank aktivierten Datenbankseiten-Prüfsummen.
Collation	Der Name der Kollationssequenz der Datenbank
ConnCount	Die Anzahl von Verbindungen zur Datenbank
date_format	Das Datumsformat, das für Zeichenfolgenkonvertierungen von der Datenbank verwendet wird. Dieses Format ist nicht unbedingt identisch mit dem Format von System.DateTime.
date_order	Die Datumsreihenfolge, die für Zeichenfolgenkonvertierungen von der Datenbank verwendet wird
Encryption	Der Typ der für die Datenbank angewendeten Verschlüsselung. Gibt None, Simple, AES oder AES_FIPS zurück.
File	Der Name der Datenbankdatei

Eigenschaft	Beschreibung
global_database_id	Der Wert der Option global_database_id, der für globalAutoincrement-Spalten verwendet wird
isolation_level	Der Wert der Option isolation_level, die benutzt wird, um zu steuern, wie stark Vorgänge in einer Transaktion für Vorgänge in anderen gleichzeitigen Transaktionen sichtbar sind. Dieser Wert wird pro Verbindung eingestellt.
MaxHashSize	Die standardmäßige maximale Anzahl von Byte für Index-Hashing. Diese Eigenschaft kann pro Index festgelegt werden.
ml_remote_id	Der Wert der Option ml_remote_id, der für die Identifizierung der Datenbank während der Synchronisation verwendet wird
Name	Der Name der Datenbankdatei (DBN)
nearest_century	Das nächste Jahrhundert, das für Zeichenfolgenkonvertierungen von der Datenbank verwendet wird
PageSize	Die Seitengröße der Datenbank in Byte
precision	Die Gleitkomma-Gesamtstellenzahl, die für Zeichenfolgenkonvertierungen von der Datenbank verwendet wird
scale	Die minimale Anzahl der Stellen nach dem Dezimalzeichen, wenn ein arithmetisches Ergebnis auf den maximalen PRECISION-Wert während Zeichenfolgenkonvertierungen von der Datenbank gekürzt wird
time_format	Das Zeitformat, das für Zeichenfolgenkonvertierungen in der Datenbank verwendet wird. Dieses Format ist nicht unbedingt identisch mit dem Format von System.TimeSpan.
timestamp_format	Das Zeitstempelformat, das für Zeichenfolgenkonvertierungen in der Datenbank verwendet wird. Dieses Format ist nicht unbedingt identisch mit dem Format von System.DateTime.
timestamp_increment	Die minimale Differenzmenge zwischen eindeutigen Zeitstempeln, in Mikrosekunden (Millionstel einer Sekunde).

Siehe auch

- [ULDatabaseSchema.SetDatabaseOption-Methode \[UltraLite.NET\] auf Seite 234](#)
- [ULConnectionParms.UserID-Eigenschaft \[UltraLite.NET\] auf Seite 178](#)
- [ULConnectionParms.Password-Eigenschaft \[UltraLite.NET\] auf Seite 177](#)
- [System.TimeSpan](#)
- [System.DateTime](#)

GetPublicationName-Methode

Gibt den Namen der Publikation, die mithilfe der Publikations-ID angegeben wurde, zurück.

Visual Basic-Syntax

```
Public Function GetPublicationName(ByVal pubID As Integer) As String
```

C#-Syntax

```
public string GetPublicationName(int pubID)
```

Parameter

- **pubID** Die ID der Publikation. Der Wert muss im Bereich [1,PublicationCount] liegen.

Rückgabe

Der Publikationsname als Zeichenfolge

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Hinweis

Publikations-IDs und Publikationsanzahl können sich während eines Schema-Upgrades ändern. Um eine Publikation korrekt zu identifizieren, greifen Sie auf sie anhand des Namens zu oder aktualisieren die IDs und Anzahlwerte im Cache nach jedem Schema-Upgrade.

Siehe auch

- [ULDatabaseSchema.PublicationCount-Eigenschaft \[UltraLite.NET\] auf Seite 236](#)

GetTableName-Methode

Gibt den Namen der Tabelle zurück, deren Tabellen-ID angegeben wurde.

Visual Basic-Syntax

```
Public Function GetTableName(ByVal tableID As Integer) As String
```

C#-Syntax

```
public string GetTableName(int tableID)
```

Parameter

- **tableID** Die ID der Tabelle. Der Wert muss im Bereich [1,TableCount] liegen.

Rückgabe

Der Tabellenname als Zeichenfolge

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Tabellen-IDs können sich während eines Schema-Upgrades ändern. Um eine Tabelle korrekt zu identifizieren, greifen Sie auf sie anhand des Namens zu oder aktualisieren die IDs im Cache nach jedem Schema-Upgrade.

Siehe auch

- [ULDatabaseSchema.TableCount-Eigenschaft \[UltraLite.NET\] auf Seite 237](#)

SetDatabaseOption-Methode

Setzt den Wert für den angegebenen Datenbankparameter fest.

Visual Basic-Syntax

```
Public Sub SetDatabaseOption(  
    ByVal name As String,  
    ByVal value As String  
)
```

C#-Syntax

```
public void SetDatabaseOption(string name, string value)
```

Parameter

- **name** Der Name der Datenbankoption. Optionsnamen unterscheiden nicht zwischen Groß- und Kleinschreibung.
- **value** Der neue Wert für die Option

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Nach dem Festlegen einer Datenbankoption wird ein Festschreibevorgang ausgeführt.

Die Verwendung dieser Methode, während eine Transaktion aktiv ist, kann zu unvorhersehbaren Ergebnissen führen und wird nicht empfohlen. Der Aufruf ändert zwar die Isolationsstufe der Verbindung, aktualisiert aber nicht den `ULTransaction.IsolationLevel`-Wert.

Folgende Optionen werden erkannt:

Option	Beschreibung
global_database_id	Der Wert, der für globalAutoincrement-Spalten verwendet wird. Der Wert muss im Bereich [0, System.UInt32.MaxValue] liegen. Der Standardwert ist der ULConnection.INVALID_DATABASE_ID-Wert (wird verwendet, um anzuzeigen, dass die Datenbank-ID für die aktuelle Datenbank nicht festgelegt wurde).
isolation_level	Der Wert, mit dem Sie kontrollieren, inwieweit Vorgänge in einer Transaktion für die Vorgänge in anderen gleichzeitigen Transaktionen sichtbar sind. Der Wert muss read_uncommitted (Nicht festgeschriebene Anweisungen lesen) oder read_committed (Festgeschriebene Anweisungen lesen) sein. Der Standardwert ist read_committed. Wenn Sie isolation_level für eine Verbindung auf "read_uncommitted" setzen, entspricht dies einem Einbetten aller Vorgänge auf dieser Verbindung in Aufrufe von BeginTransaction(System.Data.IsolationLevel.ReadUncommitted) und Commit(). Die Verwendung von read_committed entspricht indessen System.Data.IsolationLevel.ReadCommitted. SetDatabaseOption() darf nicht verwendet werden, um die Isolationsstufe der aktuellen Transaktion einzustellen. Verwenden Sie statt dessen BeginTransaction(IsolationLevel). Die UltraLite-Definition jeder Isolationsstufe unterscheidet sich geringfügig von der ADO.NET-Dokumentation des IsolationLevel-Objekts. Dieser Wert wird pro Verbindung eingestellt.
ml_remote_id	Der Wert, der zur Identifizierung der Datenbank während der Synchronisation verwendet wird. Mit einer Nullreferenz ("Nothing" in Visual Basic) als Wert entfernen Sie die Option ml_remote_id aus der Datenbank.

Siehe auch

- [ULDatabaseSchema.GetDatabaseProperty-Methode \[UltraLite.NET\] auf Seite 230](#)
- [ULConnection.INVALID_DATABASE_ID-Feld \[UltraLite.NET\] auf Seite 168](#)
- [System.UInt32.MaxValue](#)
- „Isolationsstufen“ [[UltraLite - Datenbankverwaltung](#)]

IsCaseSensitive-Eigenschaft

Prüft, ob die Datenbank zwischen Groß- und Kleinschreibung unterscheidet.

Visual Basic-Syntax

```
Public ReadOnly Property IsCaseSensitive As Boolean
```

C#-Syntax

```
public bool IsCaseSensitive {get;}
```

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

TRUE, wenn die Datenbank die Groß- und Kleinschreibung berücksichtigt, andernfalls FALSE.

Die Berücksichtigung der Groß- und Kleinschreibung von Datenbanken hat Auswirkungen darauf, wie Indizes für Tabellen und Ergebnismengen sortiert werden. Die Berücksichtigung der Groß- und Kleinschreibung hat auch Auswirkungen darauf, wie die `ULConnectionParms.UserID`- und `ULConnectionParms.Password`-Werte überprüft werden.

Siehe auch

- [ULDatabaseSchema.GetDatabaseProperty-Methode \[UltraLite.NET\] auf Seite 230](#)
- [ULConnectionParms.UserID-Eigenschaft \[UltraLite.NET\] auf Seite 178](#)
- [ULConnectionParms.Password-Eigenschaft \[UltraLite.NET\] auf Seite 177](#)

IsOpen-Eigenschaft

Ermittelt, ob das Datenbankschema geöffnet ist.

Visual Basic-Syntax

```
Public ReadOnly Property IsOpen As Boolean
```

C#-Syntax

```
public bool IsOpen {get;}
```

Bemerkungen

TRUE, wenn dieses Datenbankschema aktuell geöffnet ist, andernfalls FALSE.

Ein `ULDatabaseSchema`-Objekt ist nur geöffnet, wenn die zugeordnete Verbindung geöffnet ist.

PublicationCount-Eigenschaft

Ruft die Anzahl der Publikationen in der Datenbank ab.

Visual Basic-Syntax

```
Public ReadOnly Property PublicationCount As Integer
```

C#-Syntax

```
public int PublicationCount {get;}
```

Bemerkungen

Die Anzahl von Publikationen in der Datenbank.

Publikations-IDs reichen von 1 bis zum `PublicationCount`-Wert einschließlich.

Hinweis

Publikations-IDs und Publikationsanzahl können sich während eines Schema-Upgrades ändern. Um eine Publikation korrekt zu identifizieren, greifen Sie auf sie anhand des Namens zu oder aktualisieren die IDs und Anzahlwerte im Cache nach jedem Schema-Upgrade.

Siehe auch

- [ULDatabaseSchema.GetPublicationName-Methode \[UltraLite.NET\] auf Seite 233](#)

TableCount-Eigenschaft

Ruft die Anzahl der Tabellen in der Datenbank ab.

Visual Basic-Syntax

```
Public ReadOnly Property TableCount As Integer
```

C#-Syntax

```
public int TableCount {get;}
```

Bemerkungen

Die Anzahl der Tabellen in der Datenbank.

Tabellen-IDs reichen von 1 bis zum TableCount-Wert einschließlich.

Hinweis

Tabellen-IDs und Zähler können sich während eines Schema-Upgrades ändern. Um eine Tabelle korrekt zu identifizieren, greifen Sie auf sie anhand des Namens zu oder aktualisieren die IDs und Anzahl im Cache nach jedem Schema-Upgrade.

ULDataReader-Klasse

Repräsentiert einen schreibgeschützten bidirektionalen Cursor in einer UltraLite-Datenbank.

Visual Basic-Syntax

```
Public Class ULDataReader
    Inherits System.Data.Common.DbDataReader
    Implements System.ComponentModel.IListSource
```

C#-Syntax

```
public class ULDataReader :
    System.Data.Common.DbDataReader,
    System.ComponentModel.IListSource
```

Basisklassen

- [System.Data.Common.DbDataReader](#)
- [System.ComponentModel.IListSource](#)

Abgeleitete Klassen.

- [ULResultSet-Klasse \[UltraLite.NET\] auf Seite 351](#)

Mitglieder

Alle Mitglieder der ULDataReader-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
Close-Methode	Schließt den Cursor.
Dispose-Methode (geerbt aus System.Data.Common.DbDataReader)	Gibt alle Ressourcen frei, die von der aktuellen Instanz der Klasse System.Data.Common.DbDataReader verwendet werden.
GetBoolean-Methode	Gibt den Wert für die angegebene Spalte als System.Boolean zurück.
GetByte-Methode	Gibt den Wert für die angegebene Spalte als 8-Bit-Wert ohne Vorzeichen zurück (System.Byte).
GetBytes-Methode	UL-Erw.: Gibt den Wert für die angegebene Spalte als Array von System.Bytes -Werten zurück.
GetChar-Methode	Diese Methode wird in UltraLite.NET nicht unterstützt.
GetChars-Methode	Kopiert eine Untermenge des Werts für die angegebene ULDb-Type.Long Varchar-Spalte, beginnend beim angegebenen Offset, in das angegebene Offset des System.Char -Zielarrays.
GetData-Methode (Geerbt von System.Data.Common.DbDataReader)	Gibt ein System.Data.Common.DbDataReader -Objekt für die angeforderte Spalten-Ordinalzahl zurück.
GetDataTypeName-Methode	Gibt den Namen des Provider-Datentyps der angegebenen Spalte zurück.
GetDateTime-Methode	Gibt den Wert für die angegebene Spalte als System.DateTime -Typ mit Millisekunden-Genauigkeit zurück.
GetDbDataReader-Methode (geerbt aus System.Data.Common.DbDataReader)	Gibt ein System.Data.Common.DbDataReader -Objekt für die angeforderte Spaltenordinalzahl zurück, das mit einer providerspezifischen Implementierung aufgehoben werden kann.

Name	Beschreibung
GetDecimal-Methode	Gibt den Wert für die angegebene Spalte als System.Decimal-Typ zurück.
GetDouble-Methode	Gibt den Wert für die angegebene Spalte als System.Double-Typ zurück.
GetEnumerator-Methode	Gibt einen System.Collections.IEnumerator-Wert zurück, der das ULDataReader-Objekt durchläuft.
GetFieldType-Methode	Gibt den System.Type-Wert zurück, der für die angegebene Spalte am besten geeignet ist.
GetFieldValue(T)-Methode (geerbt aus System.Data.Common.DbDataReader)	Ruft den Wert der angegebenen Spalte synchron als Typ ab.
GetFieldValueAsync-Methode (geerbt aus System.Data.Common.DbDataReader)	Ruft den Wert der angegebenen Spalte asynchron als Typ ab.
GetFloat-Methode	Gibt den Wert für die angegebene Spalte als System.Single-Typ zurück.
GetGuid-Methode	Gibt den Wert für die angegebene Spalte als UUID-Typ zurück (System.Guid).
GetInt16-Methode	Gibt den Wert für die angegebene Spalte als System.Int16-Typ zurück.
GetInt32-Methode	Gibt den Wert für die angegebene Spalte als System.Int32-Typ zurück.
GetInt64-Methode	Gibt den Wert für die angegebene Spalte als System.Int64-Typ zurück.
GetName-Methode	Gibt den Namen der angegebenen Spalte zurück.
GetOrdinal-Methode	Gibt die Spalten-ID der angegebenen Spalte zurück.
GetProviderSpecificFieldType-Methode (geerbt aus System.Data.Common.DbDataReader)	Gibt den providerspezifischen Feldtyp der angegebenen Spalte zurück.

Name	Beschreibung
GetProviderSpecificValue-Methode (geerbt aus <code>System.Data.Common.DbDataReader</code>)	Ruft den Wert der angegebenen Spalte als Instanz von <code>System.Object</code> ab.
GetProviderSpecificValues-Methode (geerbt aus <code>System.Data.Common.DbDataReader</code>)	Ruft alle providerspezifischen Attributspalten in der Sammlung für die aktuelle Zeile ab.
GetRowCount-Methode	UL-Erw.: Gibt die Anzahl der Zeilen im Cursor innerhalb des Schwellenwerts zurück.
GetSchemaTable-Methode	Gibt einen <code>System.Data.DataTable</code> -Wert zurück, der die Spaltenmetadaten des <code>ULDataReader</code> -Objekts beschreibt.
GetStream-Methode (geerbt aus <code>System.Data.Common.DbDataReader</code>)	Ruft Daten als <code>System.IO.Stream</code> ab.
GetString-Methode	Gibt den Wert für die angegebene Spalte als <code>System.String</code> -Typ zurück.
GetTextReader-Methode (geerbt aus <code>System.Data.Common.DbDataReader</code>)	Ruft Daten als <code>System.IO.TextReader</code> ab.
GetTimeSpan-Methode	Gibt den Wert für die angegebene Spalte als <code>System.TimeSpan</code> -Typ mit Millisekunden-Genauigkeit zurück.
GetUInt16-Methode	Gibt den Wert für die angegebene Spalte als <code>System.UInt16</code> -Typ zurück.
GetUInt32-Methode	Gibt den Wert für die angegebene Spalte als <code>System.UInt32</code> -Typ zurück.
GetUInt64-Methode	Gibt den Wert für die angegebene Spalte als <code>System.UInt64</code> -Typ zurück.
GetValue-Methode	Gibt den Wert der angegebenen Spalte in ihrem nativen Format zurück.
GetValues-Methode	Gibt alle Spaltenwerte für die aktuelle Zeile zurück.
IsDBNull-Methode	Prüft, ob der Wert aus der angegebenen Spalte NULL ist.

Name	Beschreibung
IsDBNullAsync-Methode (geerbt aus System.Data.Common.DbDataReader)	Eine asynchrone Version von System.Data.Common.DbDataReader.IsDBNull(System.Int32) , die einen Wert abrufen, der angibt, ob die Spalte nicht vorhandene oder fehlende Werte enthält.
MoveAfterLast-Methode	UL-Erw.: Positioniert den Cursor hinter die letzte Zeile des Cursors.
MoveBeforeFirst-Methode	UL-Erw.: Positioniert den Cursor vor die erste Zeile des Cursors.
MoveFirst-Methode	UL-Erw.: Positioniert den Cursor auf die erste Zeile des Cursors.
MoveLast-Methode	UL-Erw.: Positioniert den Cursor auf die letzte Zeile des Cursors.
MoveNext-Methode	UL-Erw.: Positioniert den Cursor in die nächste Zeile oder nach der letzten Zeile, wenn der Cursor bereits in der letzten Zeile war.
MovePrevious-Methode	UL-Erw.: Positioniert den Cursor in die vorhergehende Zeile oder vor die erste Zeile.
MoveRelative-Methode	UL-Erw.: Positioniert den Cursor relativ zur aktuellen Zeile.
NextResult-Methode	Rückt das ULDataReader-Objekt zum nächsten Ergebnis vor, wenn Ergebnisse von Batch-SQL-Anweisungen gelesen werden.
NextResultAsync-Methode (geerbt aus System.Data.Common.DbDataReader)	Eine asynchrone Version von System.Data.Common.DbDataReader.NextResult , die den Leser beim Lesen der Ergebnisse von Anweisungsfolgen von statements.Invokes System.Data.Common.DbDataReader.NextResultAsync(System.Threading.CancellationToken) mit CancellationToken.None zum nächsten Ergebnis vorrückt.
Read-Methode	Positioniert den Cursor in die nächste Zeile oder nach der letzten Zeile, wenn der Cursor bereits in der letzten Zeile war.
ReadAsync-Methode (geerbt aus System.Data.Common.DbDataReader)	Eine asynchrone Version von System.Data.Common.DbDataReader.Read , die den Leser zum nächsten Datensatz in der Ergebnismenge vorrückt.
Depth-Eigenschaft	Gibt die Tiefe der Verschachtelung für die aktuelle Zeile zurück.
FieldCount-Eigenschaft	Gibt die Anzahl der Spalten im Cursor zurück.
HasRows-Eigenschaft	Prüft, ob das ULDataReader-Objekt eine oder mehrere Zeilen hat.
IsBOF-Eigenschaft	UL-Erw.: Gibt an, ob sich die aktuelle Zeilenposition vor der ersten Zeile befindet.

Name	Beschreibung
IsClosed-Eigenschaft	Prüft, ob der Cursor aktuell offen ist.
IsEOF-Eigenschaft	UL-Erw.: Gibt an, ob sich die aktuelle Zeilenposition nach der letzten Zeile befindet.
RecordsAffected-Eigenschaft	Gibt die Anzahl der durch eine SQL-Anweisung geänderten, eingefügten oder gelöschten Zeilen zurück.
RowCount-Eigenschaft	UL-Erw.: Gibt die Anzahl der Zeilen im Cursor zurück.
Schema-Eigenschaft	UL-Erw.: Enthält das Schema dieses Cursors.
this-Eigenschaft	Gibt den Wert der angegebenen Spalte in ihrem nativen Format zurück.
VisibleFieldCount-Eigenschaft (geerbt aus System.Data.Common.DbDataReader)	Ruft die Anzahl der Felder in System.Data.Common.DbDataReader ab, die nicht ausgeblendet sind.

Bemerkungen

Cursor sind Gruppen von Zeilen aus einer Tabelle oder die Ergebnismenge aus einer Abfrage.

Es gibt keinen Konstruktor für die ULDataReader-Klasse. Wenn Sie ein ULDataReader-Objekt abrufen möchten, führen Sie ein ULCommand-Objekt aus:

```
' Visual Basic
Dim cmd As ULCommand = new ULCommand( _
    "SELECT emp_id FROM employee", conn _
)
Dim reader As ULDataReader = cmd.ExecuteReader()
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULCommand cmd = new ULCommand(
    "SELECT emp_id FROM employee", conn
);
ULDataReader reader = cmd.ExecuteReader();
```

UL-Erw.: Der ADO.NET-Standard erfordert nur die Weitergabe durch die Ergebnismenge, aber ULDataReader-Objekte sind bidirektional. Die Move-Methoden von ULDataReader bieten volle Flexibilität bei der Navigation durch Ergebnisse.

Ein ULDataReader-Objekt ist eine schreibgeschützte Ergebnismenge. Wenn Sie ein flexibleres Objekt für die Verarbeitung von Ergebnissen benötigen, verwenden Sie die ULCommand.ExecuteResultSet-Methode, ULCommand.ExecuteTable-Methode oder eine ULDataAdapter-Klasse. Die ULDataReader-Klasse ruft Zeilen nach Bedarf ab, während die ULDataAdapter-Klasse alle Zeilen einer Ergebnismenge abrufen muss, bevor Sie mit dem Objekt weitere Vorgänge durchführen können. Bei großen Ergebnismengen bietet die ULDataReader-Klasse daher viel kürzere Antwortzeiten.

UL-Erw.: Alle Spalten eines ULDataReader-Objekts können mit der GetString-Methode abgerufen werden.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULCommand.ExecuteResultSet-Methode \[UltraLite.NET\] auf Seite 97](#)
- [ULCommand.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 100](#)
- [ULDataAdapter-Klasse \[UltraLite.NET\] auf Seite 212](#)
- [ULDataReader.GetString-Methode \[UltraLite.NET\] auf Seite 260](#)
- [System.Data.Common.DbDataReader](#)
- [System.Data.IDataReader](#)
- [System.Data.IDataRecord](#)
- [System.IDisposable](#)

Close-Methode

Schließt den Cursor.

Visual Basic-Syntax

```
Public Overrides Sub Close()
```

C#-Syntax

```
public override void Close()
```

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Es ist kein Fehler, einen Cursor zu schließen, der bereits geschlossen wurde.

GetBoolean-Methode

Gibt den Wert für die angegebene Spalte als System.Boolean zurück.

Visual Basic-Syntax

```
Public Overrides Function GetBoolean(ByVal colID As Integer) As Boolean
```

C#-Syntax

```
public override bool GetBoolean(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als System.Boolean

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Boolean](#)

GetByte-Methode

Gibt den Wert für die angegebene Spalte als 8-Bit-Wert ohne Vorzeichen zurück (System.Byte).

Visual Basic-Syntax

```
Public Overrides Function GetByte(ByVal colID As Integer) As Byte
```

C#-Syntax

```
public override byte GetByte(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als System.Byte

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Byte](#)

GetBytes-Methode

UL-Erw.: Gibt den Wert für die angegebene Spalte als Array von System.Bytes-Werten zurück.

Überladungsliste

Name	Beschreibung
GetBytes(int)-Methode	UL-Erw.: Gibt den Wert für die angegebene Spalte als Array von System.Bytes-Werten zurück.
GetBytes(int, long, byte[], int, int)-Methode	Kopiert eine Untermenge des Werts für die angegebene ULDbType.LongBinary-Spalte, beginnend beim angegebenen Offset, in das angegebene Offset des System.Byte-Zielarrays.

GetBytes(int)-Methode

UL-Erw.: Gibt den Wert für die angegebene Spalte als Array von System.Bytes-Werten zurück.

Visual Basic-Syntax

```
Public Function GetBytes(ByVal colID As Integer) As Byte()
```

C#-Syntax

```
public byte[] GetBytes(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als Array von System.Bytes-Typen.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Nur gültig für Spalten der Typen ULDbType.Binary, ULDbType.LongBinary oder ULDbType.UniqueIdentifier.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.GetBytes-Methode \[UltraLite.NET\] auf Seite 244](#)
- [System.Byte](#)

GetBytes(int, long, byte[], int, int)-Methode

Kopiert eine Untermenge des Werts für die angegebene ULDbType.LongBinary-Spalte, beginnend beim angegebenen Offset, in das angegebene Offset des System.Byte-Zielarrays.

Visual Basic-Syntax

```
Public Overrides Function GetBytes(  
    ByVal colID As Integer,  
    ByVal srcOffset As Long,  
    ByVal dst As Byte(),  
    ByVal dstOffset As Integer,  
    ByVal count As Integer  
) As Long
```

C#-Syntax

```
public override long GetBytes(  
    int colID,  
    long srcOffset,  
    byte[] dst,  
    int dstOffset,  
    int count  
)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **srcOffset** Die Startposition im Spaltenwert. Null ist der Anfang des Werts.
- **dst** Das Ziel-Array
- **dstOffset** Die Startposition im Ziel-Array
- **count** Die Anzahl der zu kopierenden Bytes.

Rückgabe

Die tatsächlich zu kopierende Anzahl von Bytes.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Wenn Sie einen *dst*-Puffer übergeben, der eine Nullreferenz hat ("Nothing" in Visual Basic), gibt die **GetBytes**-Methode die Länge des Feldes in Byte zurück.

Die Bytes an den Positionen *srcOffset* bis *srcOffset* + *count* - 1 des Werts werden an die Positionen *dstOffset* bis *dstOffset* + *count* - 1 des Ziel-Arrays kopiert. Falls das Ende des Werts angetroffen wird, bevor *count* Byte kopiert wurden, bleibt der Rest des Ziel-Arrays unverändert.

Falls eine der folgenden Aussagen zutrifft, wird ein **ULException**-Objekt mit **ULSQLCode.SQLE_INVALID_PARAMETER**-Code ausgegeben und das Ziel wird nicht geändert:

- *srcOffset* ist negativ.
- *dstOffset* ist negativ.

- *count* ist negativ.
- *dstOffset* + *count* ist größer als die *dst*-Länge.

Bei anderen Fehlern wird ein `ULException`-Objekt mit dem entsprechenden Fehlercode ausgegeben.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.GetBytes-Methode \[UltraLite.NET\] auf Seite 244](#)
- [ULException-Klasse \[UltraLite.NET\] auf Seite 275](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Byte](#)

GetChar-Methode

Diese Methode wird in UltraLite.NET nicht unterstützt.

Visual Basic-Syntax

```
Public Overrides Function GetChar(ByVal colID As Integer) As Char
```

C#-Syntax

```
public override char GetChar(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Diese Methode wird in UltraLite.NET nicht unterstützt.

Ausnahmen

- [ULException-Klasse](#) Diese Methode wird in UltraLite.NET nicht unterstützt.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.GetString-Methode \[UltraLite.NET\] auf Seite 260](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)

GetChars-Methode

Kopiert eine Untermenge des Werts für die angegebene `ULDbType.LongVarchar`-Spalte, beginnend beim angegebenen Offset, in das angegebene Offset des `System.Char`-Zielarrays.

Visual Basic-Syntax

```
Public Overrides Function GetChars(  
    ByVal colID As Integer,  
    ByVal srcOffset As Long,  
    ByVal dst As Char(),  
    ByVal dstOffset As Integer,  
    ByVal count As Integer  
) As Long
```

C#-Syntax

```
public override long GetChars(  
    int colID,  
    long srcOffset,  
    char[] dst,  
    int dstOffset,  
    int count  
)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **srcOffset** Die Startposition im Spaltenwert. Null ist der Anfang des Werts.
- **dst** Das Ziel-Array
- **dstOffset** Die Startposition im Ziel-Array
- **count** Die Anzahl der zu kopierenden Zeichen.

Rückgabe

Die tatsächliche Anzahl der kopierten Zeichen

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Wenn Sie einen *dst*-Puffer übergeben, der eine Nullreferenz hat ("Nothing" in Visual Basic), gibt die *GetChars*-Methode die Länge des Feldes in Zeichen zurück.

Die Zeichen an den Positionen *srcOffset* bis *srcOffset* + *count* - 1 des Werts werden an die Positionen *dstOffset* bis *dstOffset* + *count* - 1 des Ziel-Arrays kopiert. Falls das Ende des Werts angetroffen wird, bevor *count* Zeichen kopiert wurden, bleibt der Rest des Ziel-Arrays unverändert.

Falls eine der folgenden Aussagen zutrifft, wird ein *ULException*-Objekt mit der *ULSQLCode.SQLE_INVALID_PARAMETER*-Konstanten ausgegeben und das Ziel wird nicht verändert:

- Der *srcOffset*-Wert ist negativ.

- Der *dstOffset*-Wert ist negativ.
- Der *count*-Wert ist negativ.
- Der Wert *dstOffset + count* ist größer als die *dst*-Länge.

Bei anderen Fehlern wird ein `ULException`-Objekt mit dem entsprechenden Fehlercode ausgegeben.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULException-Klasse \[UltraLite.NET\] auf Seite 275](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Char](#)

GetDataTypeName-Methode

Gibt den Namen des Provider-Datentyps der angegebenen Spalte zurück.

Visual Basic-Syntax

```
Public Overrides Function GetDataTypeName(  
    ByVal colID As Integer  
) As String
```

C#-Syntax

```
public override string GetDataTypeName(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Eine Zeichenfolge, die dem `ULDbType`-Typ der Spalte entspricht.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULCursorSchema.GetColumnULDbType-Methode \[UltraLite.NET\] auf Seite 210](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [ULDbType-Enumeration \[UltraLite.NET\] auf Seite 459](#)

GetDateTime-Methode

Gibt den Wert für die angegebene Spalte als System.DateTime-Typ mit Millisekunden-Genauigkeit zurück.

Visual Basic-Syntax

```
Public Overrides Function GetDateTime(ByVal colID As Integer) As Date
```

C#-Syntax

```
public override DateTime GetDateTime(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als System.DateTime-Typ.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.DateTime](#)

GetDecimal-Methode

Gibt den Wert für die angegebene Spalte als System.Decimal-Typ zurück.

Visual Basic-Syntax

```
Public Overrides Function GetDecimal(ByVal colID As Integer) As Decimal
```

C#-Syntax

```
public override decimal GetDecimal(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als System.Decimal-Typ.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Decimal](#)

GetDouble-Methode

Gibt den Wert für die angegebene Spalte als System.Double-Typ zurück.

Visual Basic-Syntax

```
Public Overrides Function GetDouble(ByVal colID As Integer) As Double
```

C#-Syntax

```
public override double GetDouble(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als System.Double-Typ.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Double](#)

GetEnumerator-Methode

Gibt einen System.Collections.IEnumerator-Wert zurück, der das ULDataReader-Objekt durchläuft.

Visual Basic-Syntax

```
Public Overrides Function GetEnumerator()  
    As System.Collections.IEnumerator
```

C#-Syntax

```
public override IEnumerator GetEnumerator()
```

Rückgabe

Ein System.Collections.IEnumerator für das ULDataReader-Objekt.

Siehe auch

- [System.Collections.IEnumerator](#)

GetFieldType-Methode

Gibt den System.Type-Wert zurück, der für die angegebene Spalte am besten geeignet ist.

Visual Basic-Syntax

```
Public Overrides Function GetFieldType(ByVal colID As Integer) As Type
```

C#-Syntax

```
public override Type GetFieldType(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Ein System.Type-Wert für die Spalte

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetDataTypeName-Methode \[UltraLite.NET\] auf Seite 249](#)
- [ULCursorSchema.GetColumnULDbType-Methode \[UltraLite.NET\] auf Seite 210](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Type](#)

GetFloat-Methode

Gibt den Wert für die angegebene Spalte als System.Single-Typ zurück.

Visual Basic-Syntax

```
Public Overrides Function GetFloat(ByVal colID As Integer) As Single
```

C#-Syntax

```
public override float GetFloat(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als System.Single-Typ.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Single](#)

GetGuid-Methode

Gibt den Wert für die angegebene Spalte als UUID-Typ zurück (System.Guid).

Visual Basic-Syntax

```
Public Overrides Function GetGuid(ByVal colID As Integer) As Guid
```

C#-Syntax

```
public override Guid GetGuid(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als GUID-Typ.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Diese Methode ist nur für Spalten des ULDbType.UniqueIdentifier-Typs oder des ULDbType.Binary-Typs mit der Länge 16 zulässig.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULCursorSchema.GetColumnULDbType-Methode \[UltraLite.NET\] auf Seite 210](#)
- [ULCursorSchema.GetColumnSize-Methode \[UltraLite.NET\] auf Seite 208](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Guid](#)

GetInt16-Methode

Gibt den Wert für die angegebene Spalte als System.Int16-Typ zurück.

Visual Basic-Syntax

```
Public Overrides Function GetInt16(ByVal colID As Integer) As Short
```

C#-Syntax

```
public override short GetInt16(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als System.Int16-Typ.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Int16](#)

GetInt32-Methode

Gibt den Wert für die angegebene Spalte als System.Int32-Typ zurück.

Visual Basic-Syntax

```
Public Overrides Function GetInt32(ByVal colID As Integer) As Integer
```

C#-Syntax

```
public override int GetInt32(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als System.Int32-Typ.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Int32](#)

GetInt64-Methode

Gibt den Wert für die angegebene Spalte als System.Int64-Typ zurück.

Visual Basic-Syntax

```
Public Overrides Function GetInt64(ByVal colID As Integer) As Long
```

C#-Syntax

```
public override long GetInt64(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als System.Int64-Typ.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Int64](#)

GetName-Methode

Gibt den Namen der angegebenen Spalte zurück.

Visual Basic-Syntax

```
Public Overrides Function GetName(ByVal colID As Integer) As String
```

C#-Syntax

```
public override string GetName(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Name der Spalte oder eine Nullreferenz ("Nothing" in Visual Basic), wenn die Spalte keinen Namen hat. Wenn die Spalte in der SQL-Abfrage einen Aliasnamen hat, wird der Aliasname zurückgegeben.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

In Ergebnismengen haben nicht alle Spalten Namen und nicht alle Spaltennamen sind eindeutig. Wenn Sie keine Aliasnamen verwenden, wird dem Namen einer nicht berechneten Spalte der Name der Tabelle vorangestellt, aus der die Spalte stammt. Beispiel: Der MyTable.ID-Wert ist der Name der einzigen Spalte in der Ergebnismenge für die Abfrage "SELECT ID FROM MyTable".

Diese Methode ist identisch mit der ULCursorSchema.GetColumnName-Methode.

Siehe auch

- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [ULDataReader.GetSchemaTable-Methode \[UltraLite.NET\] auf Seite 258](#)
- [ULCursorSchema.GetColumnName-Methode \[UltraLite.NET\] auf Seite 207](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)

GetOrdinal-Methode

Gibt die Spalten-ID der angegebenen Spalte zurück.

Visual Basic-Syntax

```
Public Overrides Function GetOrdinal(  
    ByVal columnName As String  
) As Integer
```

C#-Syntax

```
public override int GetOrdinal(string columnName)
```

Parameter

- **columnName** Der Name der Spalte.

Rückgabe

Die Spalten-ID der benannten Spalte.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Spalten-IDs liegen im Bereich von 0 bis einschließlich `ULDataReader.FieldCount-1`.

In Ergebnismengen haben nicht alle Spalten Namen und nicht alle Spaltennamen sind eindeutig. Wenn Sie keine Aliasnamen verwenden, wird dem Namen einer nicht berechneten Spalte der Name der Tabelle vorangestellt, aus der die Spalte stammt. Beispiel: Der `MyTable.ID`-Wert ist der Name der einzigen Spalte in der Ergebnismenge für die Abfrage "SELECT ID FROM MyTable".

Spalten-IDs und Zähler können sich während eines Schema-Upgrades ändern. Um eine Spalte korrekt zu identifizieren, greifen Sie auf sie anhand des Namens zu oder aktualisieren die IDs und Anzahl im Cache nach jedem Schema-Upgrade.

Diese Methode ist identisch mit der `ULCursorSchema.GetColumnID`-Methode.

Siehe auch

- [ULDataReader.GetSchemaTable-Methode \[UltraLite.NET\] auf Seite 258](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [ULCursorSchema.GetColumnID-Methode \[UltraLite.NET\] auf Seite 206](#)

GetRowCount-Methode

UL-Erw.: Gibt die Anzahl der Zeilen im Cursor innerhalb des Schwellenwerts zurück.

Visual Basic-Syntax

```
Public Function GetRowCount(ByVal threshold As Integer) As Integer
```

C#-Syntax

```
public int GetRowCount(int threshold)
```

Parameter

- **threshold** Schwellenwert für die Zeilenanzahl.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Anzahl der Zeilen im Cursor.

Bei komplexen Abfragen ist die RowCount-Eigenschaft aufwendig, da sie das Durchlaufen von Cursorzeilen erfordert. Mit der GetRowCount(threshold)-Methode kann der Aufrufer ermitteln, ob zumindest Schwellenwertzeilen vorhanden sind. Wenn die Anzahl der Zeilen unter dem Schwellenwert liegt, wird diese Zahl zurückgegeben; sonst wird der Schwellenwert zurückgegeben. Dieser Aufruf kann mit einem höheren Schwellenwert wiederholt werden.

Gibt die RowCount-Eigenschaft zurück, wenn der Schwellenwert 0 ist.

Siehe auch

- [ULDataReader.RowCount-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)

GetSchemaTable-Methode

Gibt einen System.Data.DataTable-Wert zurück, der die Spaltenmetadaten des ULDataReader-Objekts beschreibt.

Visual Basic-Syntax

```
Public Overrides Function GetSchemaTable() As DataTable
```

C#-Syntax

```
public override DataTable GetSchemaTable()
```

Rückgabe

Ein System.Data.DataTable-Objekt, das das Schema der einzelnen Spalten im ULDataReader-Objekt beschreibt.

Bemerkungen

Die GetSchemaTable-Methode gibt Metadaten über jede Spalte in der nachstehenden Reihenfolge zurück:

DataTable-Spalte	Beschreibung
ColumnName	Der Name der Spalte oder eine Nullreferenz ("Nothing" in Visual Basic), wenn die Spalte keinen Namen hat. Wenn die Spalte in der SQL-Abfrage einen Aliasnamen hat, wird der Aliasname zurückgegeben. In Ergebnismengen haben nicht alle Spalten Namen und nicht alle Spaltennamen sind eindeutig.
ColumnOrdinal	Die ID der Spalte. Der Wert ist im Bereich [0, FieldCount -1].
ColumnSize	Für Spalten mit Größenangaben ist dies die maximale Länge eines Werts in der Spalte. Für andere Spalten ist dies die Größe des Datentyps in Bytes.

DataTable-Spalte	Beschreibung
NumericPrecision	Die Gesamtstellenzahl einer numerischen Spalte (ProviderType ULDbType.Decimal oder ULDbType.Numeric) oder DBNull, wenn die Spalte nicht numerisch ist
NumericScale	Die Anzahl der Dezimalstellen einer numerischen Spalte (ProviderType ULDbType.Decimal oder ULDbType.Numeric) oder DBNull, wenn die Spalte nicht numerisch ist
IsUnique	TRUE, wenn die Spalte eine nicht berechnete eindeutige Spalte in der Tabelle (BaseTableName) ist, aus der sie stammt.
IsKey	TRUE, wenn die Spalte Teil einer Gruppe von Spalten in der Ergebnismenge ist, die zusammen aus einem eindeutigen Schlüssel für die Ergebnismenge genommen werden. Die Gruppe der Spalten, für die der IsKey-Wert TRUE ist, muss nicht die Mindestgruppe sein, die eine Zeile in der Ergebnismenge eindeutig kennzeichnet.
BaseCatalogName	Der Name des Katalogs in der Datenbank, die die Spalte enthält. Für UltraLite.NET ist dieser Wert immer DBNull.
BaseColumnName	Der ursprüngliche Name der Spalte in der BaseTableName-Tabelle der Datenbank oder DBNull, wenn die Spalte berechnet wird oder wenn die Informationen nicht ermittelt werden können.
BaseSchemaName	Der Name des Schemas in der Datenbank, das die Spalte enthält. Für UltraLite.NET ist dieser Wert immer DBNull.
BaseTableName	Der Name der Tabelle in der Datenbank, die die Spalte enthält, oder DBNull, wenn die Spalte berechnet wird oder diese Information nicht ermittelt werden kann.
DataType	Der .NET-Datentyp, der für diesen Spaltentyp am besten geeignet ist.
AllowDBNull	TRUE, wenn die Spalte nullwertfähig ist. FALSE, wenn die Spalte nicht nullwertfähig ist oder diese Information nicht ermittelt werden kann.
ProviderType	Der ULDbType-Wert der Spalte.
IsIdentity	TRUE, wenn die Spalte eine Identity-Spalte ist, andernfalls FALSE. Für UltraLite.NET ist dieser Wert immer FALSE.
IsAutoIncrement	TRUE, wenn die Spalte eine Autoincrement- oder globalAutoincrement-Spalte ist, andernfalls FALSE (oder wenn diese Information nicht ermittelt werden kann)
IsRowVersion	TRUE, wenn die Spalte einen beständigen Zeilenbezeichner enthält, der nicht beschrieben werden kann und einen Wert ohne Bedeutung hat, der lediglich die Zeile kennzeichnet. Für UltraLite.NET ist dieser Wert immer FALSE.

DataTable-Spalte	Beschreibung
IsLong	TRUE, wenn die Spalte eine ULDbType.LongVarchar- oder ULDbType.LongBinary-Spalte ist, anderenfalls FALSE
IsReadOnly	TRUE, wenn die Spalte schreibgeschützt ist. FALSE, wenn sie geändert werden kann oder die Zugriffsberechtigung nicht ermittelt werden kann.
IsAliased	TRUE, wenn der Spaltenname ein Aliasname ist; andernfalls FALSE.
IsExpression	TRUE, wenn die Spalte ein Ausdruck ist. FALSE, wenn Sie einen Wert enthält.

Siehe auch

- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [ULDbType-Enumeration \[UltraLite.NET\] auf Seite 459](#)
- [System.Data.DataTable](#)

GetString-Methode

Gibt den Wert für die angegebene Spalte als System.String-Typ zurück.

Visual Basic-Syntax

```
Public Overrides Function GetString(ByVal colID As Integer) As String
```

C#-Syntax

```
public override String GetString(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als System.String-Typ.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.String](#)

GetTimeSpan-Methode

Gibt den Wert für die angegebene Spalte als System.TimeSpan-Typ mit Millisekunden-Genauigkeit zurück.

Visual Basic-Syntax

```
Public Function GetTimeSpan(ByVal colID As Integer) As TimeSpan
```

C#-Syntax

```
public TimeSpan GetTimeSpan(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als System.TimeSpan-Typ.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.TimeSpan](#)

GetUInt16-Methode

Gibt den Wert für die angegebene Spalte als System.UInt16-Typ zurück.

Visual Basic-Syntax

```
Public Function GetUInt16(ByVal colID As Integer) As UShort
```

C#-Syntax

```
public ushort GetUInt16(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als System.UInt16-Typ.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.UInt16](#)

GetUInt32-Methode

Gibt den Wert für die angegebene Spalte als System.UInt32-Typ zurück.

Visual Basic-Syntax

```
Public Function GetUInt32(ByVal colID As Integer) As UInteger
```

C#-Syntax

```
public uint GetUInt32(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als System.UInt32-Typ.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.UInt32](#)

GetUInt64-Methode

Gibt den Wert für die angegebene Spalte als System.UInt64-Typ zurück.

Visual Basic-Syntax

```
Public Function GetUInt64(ByVal colID As Integer) As ULong
```

C#-Syntax

```
public ulong GetUInt64(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als System.UInt64-Typ.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.UInt64](#)

GetValue-Methode

Gibt den Wert der angegebenen Spalte in ihrem nativen Format zurück.

Visual Basic-Syntax

```
Public Overrides Function GetValue(ByVal colID As Integer) As Object
```

C#-Syntax

```
public override object GetValue(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als .NET-Typ, der am besten für die Spalte geeignet ist, oder als DBNull-Typ, wenn die Spalte NULL ist.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Diese Methode ist identisch mit der ULDataReader.this[int]-Methode.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)

GetValues-Methode

Gibt alle Spaltenwerte für die aktuelle Zeile zurück.

Visual Basic-Syntax

```
Public Overrides Function GetValues(ByVal values As Object()) As Integer
```

C#-Syntax

```
public override int GetValues(object[] values)
```

Parameter

- **values** Das Array von System.Objects-Objekten zum Speichern der gesamten Zeile

Rückgabe

Die Anzahl der abgerufenen Spaltenwerte. Wenn die Länge des Arrays größer ist als die Anzahl der Spalten (ULDataReader.FieldCount), können nur FieldCount-Elemente abgerufen werden und der Rest des Arrays bleibt unverändert.

Ausnahmebedingungen

- **ArgumentNullException** Das *values*-Array ist NULL oder hat die Länge Null.
- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Für die meisten Anwendungen bietet die GetValues-Methode eine effiziente Möglichkeit für den Abruf aller Spalten anstelle des Abrufs jeweils einzelner Spalten.

Sie können ein System.Object-Array übergeben, das weniger Spalten enthält als in der Ergebniszeile vorhanden sind. Nur die Menge von Daten, die im System.Object-Array enthalten sind, wird in das Array kopiert. Sie können auch ein System.Object-Array übergeben, dessen Länge geringer ist als die Anzahl von Spalten in der Ergebniszeile.

Diese Methode gibt den DBNull-Typ für Datenbankspalten mit NULL zurück. Für andere Spalten gibt sie den Wert der Spalte in ihrem nativen Format zurück.

Siehe auch

- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.GetValue-Methode \[UltraLite.NET\] auf Seite 263](#)
- [System.Object](#)

IsDBNull-Methode

Prüft, ob der Wert aus der angegebenen Spalte NULL ist.

Visual Basic-Syntax

```
Public Overrides Function IsDBNull(ByVal colID As Integer) As Boolean
```

C#-Syntax

```
public override bool IsDBNull(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

TRUE, wenn der Wert NULL ist, andernfalls FALSE

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)

MoveAfterLast-Methode

UL-Erw.: Positioniert den Cursor hinter die letzte Zeile des Cursors.

Visual Basic-Syntax

```
Public Sub MoveAfterLast()
```

C#-Syntax

```
public void MoveAfterLast()
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

MoveBeforeFirst-Methode

UL-Erw.: Positioniert den Cursor vor die erste Zeile des Cursors.

Visual Basic-Syntax

```
Public Sub MoveBeforeFirst()
```

C#-Syntax

```
public void MoveBeforeFirst()
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

MoveFirst-Methode

UL-Erw.: Positioniert den Cursor auf die erste Zeile des Cursors.

Visual Basic-Syntax

```
Public Function MoveFirst() As Boolean
```

C#-Syntax

```
public bool MoveFirst()
```

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE. Die Methode schlägt zum Beispiel fehl, wenn keine Zeilen vorhanden sind.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

MoveLast-Methode

UL-Erw.: Positioniert den Cursor auf die letzte Zeile des Cursors.

Visual Basic-Syntax

```
Public Function MoveLast() As Boolean
```

C#-Syntax

```
public bool MoveLast()
```

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE. Die Methode schlägt zum Beispiel fehl, wenn keine Zeilen vorhanden sind.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

MoveNext-Methode

UL-Erw.: Positioniert den Cursor in die nächste Zeile oder nach der letzten Zeile, wenn der Cursor bereits in der letzten Zeile war.

Visual Basic-Syntax

```
Public Function MoveNext() As Boolean
```

C#-Syntax

```
public bool MoveNext()
```

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE. Die Methode schlägt zum Beispiel fehl, wenn keine weiteren Zeilen vorhanden sind.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Diese Methode ist identisch mit der ULDataReader.Read-Methode.

Siehe auch

- [ULDataReader.Read-Methode \[UltraLite.NET\] auf Seite 269](#)

MovePrevious-Methode

UL-Erw.: Positioniert den Cursor in die vorhergehende Zeile oder vor die erste Zeile.

Visual Basic-Syntax

```
Public Function MovePrevious() As Boolean
```

C#-Syntax

```
public bool MovePrevious()
```

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE. Die Methode schlägt zum Beispiel fehl, wenn keine weiteren Zeilen vorhanden sind.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

MoveRelative-Methode

UL-Erw.: Positioniert den Cursor relativ zur aktuellen Zeile.

Visual Basic-Syntax

```
Public Function MoveRelative(ByVal offset As Integer) As Boolean
```

C#-Syntax

```
public bool MoveRelative(int offset)
```

Parameter

- **offset** Die Anzahl Zeilen, um die die Position verschoben wird. Bei negativen Werten werden die Zeilen nach hinten verschoben.

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE. Die Methode schlägt zum Beispiel fehl, wenn sie den Cursor vor die erste oder nach der letzten Zeile positioniert.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Falls die Zeile nicht vorhanden ist, gibt die Methode FALSE zurück und die Cursorposition ist nach der letzten Zeile (die ULDataReader.IsEOF-Methode), wenn der *offset*-Wert positiv ist, und vor der ersten Zeile (die ULDataReader.IsBOF-Methode), wenn der *offset*-Wert negativ ist.

Siehe auch

- [ULDataReader.IsEOF-Eigenschaft \[UltraLite.NET\] auf Seite 271](#)
- [ULDataReader.IsBOF-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)

NextResult-Methode

Rückt das ULDataReader-Objekt zum nächsten Ergebnis vor, wenn Ergebnisse von Batch-SQL-Anweisungen gelesen werden.

Visual Basic-Syntax

```
Public Overrides Function NextResult() As Boolean
```

C#-Syntax

```
public override bool NextResult()
```

Rückgabe

TRUE, wenn mehrere Ergebnismengen vorhanden sind, andernfalls FALSE. Für UltraLite.NET wird immer FALSE zurückgegeben.

Ausnahmen

- **ULException-Klasse** Das ULDataReader-Objekt ist nicht geöffnet.

Bemerkungen

UL-Erw.: UltraLite.NET unterstützt keine Batch-SQL-Anweisungen, sodass das ULDataReader-Objekt immer in der ersten und einzigen Ergebnismenge positioniert wird. NextResult-Methodenaufrufe haben keine Auswirkung.

Read-Methode

Positioniert den Cursor in die nächste Zeile oder nach der letzten Zeile, wenn der Cursor bereits in der letzten Zeile war.

Visual Basic-Syntax

```
Public Overrides Function Read() As Boolean
```

C#-Syntax

```
public override bool Read()
```

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE. Die Methode schlägt zum Beispiel fehl, wenn keine weiteren Zeilen vorhanden sind.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Diese Methode ist identisch mit der ULDataReader.MoveNext-Methode.

Siehe auch

- [ULDataReader.MoveNext-Methode \[UltraLite.NET\] auf Seite 266](#)

Depth-Eigenschaft

Gibt die Tiefe der Verschachtelung für die aktuelle Zeile zurück.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property Depth As Integer
```

C#-Syntax

```
public override int Depth {get;}
```

Ausnahmen

- [ULException-Klasse](#) Das ULDataReader-Objekt ist nicht geöffnet.

Bemerkungen

Die äußerste Tabelle hat die Tiefe Null.

Alle UltraLite.NET-Ergebnismengen haben die Tiefe Null.

FieldCount-Eigenschaft

Gibt die Anzahl der Spalten im Cursor zurück.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property FieldCount As Integer
```

C#-Syntax

```
public override int FieldCount {get;}
```

Rückgabe

Die Anzahl der Spalten im Cursor als Ganzzahl. Gibt 0 zurück, wenn der Cursor geschlossen ist.

Bemerkungen

Diese Methode ist identisch mit der `ULCursorSchema.ColumnCount`-Methode.

Siehe auch

- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)

HasRows-Eigenschaft

Prüft, ob das `ULDataReader`-Objekt eine oder mehrere Zeilen hat.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property HasRows As Boolean
```

C#-Syntax

```
public override bool HasRows {get;}
```

Bemerkungen

TRUE, wenn die Ergebnismenge mindestens eine Zeile enthält, andernfalls FALSE.

IsBOF-Eigenschaft

UL-Erw.: Gibt an, ob sich die aktuelle Zeilenposition vor der ersten Zeile befindet.

Visual Basic-Syntax

```
Public ReadOnly Property IsBOF As Boolean
```

C#-Syntax

```
public bool IsBOF {get;}
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

TRUE, wenn sich die aktuelle Zeilenposition vor der ersten Zeile befindet, andernfalls FALSE.

IsClosed-Eigenschaft

Prüft, ob der Cursor aktuell offen ist.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property IsClosed As Boolean
```

C#-Syntax

```
public override bool IsClosed {get;}
```

Bemerkungen

TRUE, wenn dieser Cursor aktuell geöffnet ist, andernfalls FALSE.

IsEOF-Eigenschaft

UL-Erw.: Gibt an, ob sich die aktuelle Zeilenposition nach der letzten Zeile befindet.

Visual Basic-Syntax

```
Public ReadOnly Property IsEOF As Boolean
```

C#-Syntax

```
public bool IsEOF {get;}
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

TRUE, wenn sich die aktuelle Zeilenposition nach der letzten Zeile befindet, andernfalls FALSE.

RecordsAffected-Eigenschaft

Gibt die Anzahl der durch eine SQL-Anweisung geänderten, eingefügten oder gelöschten Zeilen zurück.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property RecordsAffected As Integer
```

C#-Syntax

```
public override int RecordsAffected {get;}
```

Bemerkungen

Für SELECT-Anweisungen oder CommandType.TableDirect-Tabellen ist dieser Wert -1.

Die Anzahl der durch eine SQL-Anweisung geänderten, eingefügten oder gelöschten Zeilen.

Siehe auch

- [System.Data.CommandType](#)

RowCount-Eigenschaft

UL-Erw.: Gibt die Anzahl der Zeilen im Cursor zurück.

Visual Basic-Syntax

```
Public ReadOnly Property RowCount As Integer
```

C#-Syntax

```
public int RowCount {get;}
```

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Anzahl der Zeilen im Cursor.

RowCount kann z.B. eingesetzt werden, um zu entscheiden, ob die alten Zeilen gelöscht werden sollen, um Platz zu sparen. Alte Zeilen können mit der ULConnection.StopSynchronizationDelete-Methode aus der UltraLite-Datenbank gelöscht werden, ohne dass sie aus der konsolidierten Datenbank gelöscht werden.

Siehe auch

- [ULConnection.StartSynchronizationDelete-Methode \[UltraLite.NET\] auf Seite 155](#)
- [ULConnection.StopSynchronizationDelete-Methode \[UltraLite.NET\] auf Seite 155](#)

Schema-Eigenschaft

UL-Erw.: Enthält das Schema dieses Cursors.

Visual Basic-Syntax

```
Public ReadOnly Property Schema As ULCursorSchema
```

C#-Syntax

```
public ULCursorSchema Schema {get;}
```

Bemerkungen

Für Ergebnismengen stellt das `ULResultSetSchema`-Objekt das Schema der Ergebnismenge dar. Für Tabellen stellt das `ULTableSchema`-Objekt das Schema der Tabelle dar.

Diese Eigenschaft stellt das vollständige Schema des Cursors dar, einschließlich erweiterter UltraLite.NET-Informationen, die in den Ergebnissen der `ULDataReader.GetSchemaTable`-Methode nicht enthalten sind.

Siehe auch

- [ULTableSchema-Klasse \[UltraLite.NET\] auf Seite 437](#)
- [ULDataReader.GetSchemaTable-Methode \[UltraLite.NET\] auf Seite 258](#)
- [ULResultSetSchema-Klasse \[UltraLite.NET\] auf Seite 375](#)

this-Eigenschaft

Gibt den Wert der angegebenen Spalte in ihrem nativen Format zurück.

Überladungsliste

Name	Beschreibung
this[Int]-Eigenschaft	Gibt den Wert der angegebenen Spalte in ihrem nativen Format zurück.
this[String]-Eigenschaft	Gibt den Wert der angegebenen benannten Spalte in ihrem nativen Format zurück.

this[Int]-Eigenschaft

Gibt den Wert der angegebenen Spalte in ihrem nativen Format zurück.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property Item(  
    ByVal colID As Integer  
) As Object
```

C#-Syntax

```
public override object this[int colID] {get;}
```

Parameter

- colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Rückgabe

Der Spaltenwert als .NET-Typ, der am besten für die Spalte geeignet ist, oder `DBNull`, wenn die Spalte `NULL` ist

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

In C# ist diese Eigenschaft der Indexersteller für die ULDataReader-Klasse.

Diese Methode hat dieselbe Funktion wie die ULDataReader.GetValue(int)-Methode.

Siehe auch

- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.GetValue-Methode \[UltraLite.NET\] auf Seite 263](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)

this[String]-Eigenschaft

Gibt den Wert der angegebenen benannten Spalte in ihrem nativen Format zurück.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property Item(ByVal name As String) As Object
```

C#-Syntax

```
public override object this[string name] {get;}
```

Parameter

- **name** Der Name der Spalte.

Rückgabe

Der Spaltenwert als .NET-Typ, der am besten für die Spalte geeignet ist, oder DBNull, wenn die Spalte NULL ist

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

In C# ist diese Eigenschaft der Indexer für das ULDataReader-Objekt.

In Ergebnismengen haben nicht alle Spalten Namen und nicht alle Spaltennamen sind eindeutig. Wenn Sie keine Aliasnamen verwenden, wird dem Namen einer nicht berechneten Spalte der Name der Tabelle vorangestellt, aus der die Spalte stammt. Beispiel: Der MyTable.ID-Wert ist der Name der einzigen Spalte in der Ergebnismenge für die Abfrage "SELECT ID FROM MyTable".

Beim wiederholten Zugriff auf Spalten ist es effizienter, mithilfe der Spalten-ID anstelle des Spaltennamens zuzugreifen.

Diese Methode ist äquivalent zu Folgendem:

```
dataReader.GetValue( dataReader.GetOrdinal( name ) )
```

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetValue-Methode \[UltraLite.NET\] auf Seite 263](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)

ULException-Klasse

Repräsentiert einen SQL-Fehler, der von der UltraLite.NET-Datenbank zurückgegeben wurde.

Visual Basic-Syntax

```
Public NotInheritable Class ULException
    Inherits System.ApplicationException
```

C#-Syntax

```
public sealed class ULException : System.ApplicationException
```

Basisklassen

- [System.ApplicationException](#)

Mitglieder

Alle Mitglieder der ULException-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
GetObjectData-Methode	Füllt ein SerializationInfo-Objekt mit den Daten, die erforderlich sind, um dieses ULException-Objekt zu serialisieren.
NativeError-Eigenschaft	Gibt den von der Datenbank zurückgegebenen SQLCODE-Wert zurück
Source-Eigenschaft	Gibt den Namen des Providers zurück, der den Fehler generiert hat.

Bemerkungen

Der SQL-Code, der den Fehler angibt, wird in der NativeError-Eigenschaft zurückgegeben.

Diese Klasse ist unter .NET Compact Framework nicht serialisierbar.

Siehe auch

- [ULException.NativeError-Eigenschaft \[UltraLite.NET\] auf Seite 276](#)
- „SQL Anywhere-Fehlermeldungen - sortiert nach SQLCODE“ [[Fehlermeldungen](#)]

GetObjectData-Methode

Füllt ein `SerializationInfo`-Objekt mit den Daten, die erforderlich sind, um dieses `ULException`-Objekt zu serialisieren.

Visual Basic-Syntax

```
Public Overrides Sub GetObjectData(  
    ByVal info As SerializationInfo,  
    ByVal context As StreamingContext  
)
```

C#-Syntax

```
public override void GetObjectData(  
    SerializationInfo info,  
    StreamingContext context  
)
```

Parameter

- **info** Das `SerializationInfo`-Objekt, das mit Daten gefüllt werden soll
- **context** Das Ziel für diese Serialisierung

Bemerkungen

Diese Klasse wird unter .NET Compact Framework nicht unterstützt.

NativeError-Eigenschaft

Gibt den von der Datenbank zurückgegebenen `SQLCODE`-Wert zurück

Visual Basic-Syntax

```
Public ReadOnly Property NativeError As ULSQLCode
```

C#-Syntax

```
public ULSQLCode NativeError {get;}
```

Bemerkungen

Der `ULSQLCode`-Wert, der von der Datenbank zurückgegeben wurde

Source-Eigenschaft

Gibt den Namen des Providers zurück, der den Fehler generiert hat.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property Source As String
```

C#-Syntax

```
public override string Source {get;}
```

Bemerkungen

Der Zeichenfolgenwert, der UltraLite.NET als Provider kennzeichnet.

ULFactory-Klasse

Repräsentiert eine Gruppe von Methoden zum Erstellen der Instanzen der Implementierung der Datenquellenklassen des iAnywhere.Data.UltraLite-Providers.

Visual Basic-Syntax

```
Public Class ULFactory Inherits System.Data.Common.DbProviderFactory
```

C#-Syntax

```
public class ULFactory : System.Data.Common.DbProviderFactory
```

Basisklassen

- [System.Data.Common.DbProviderFactory](#)

Mitglieder

Alle Mitglieder der ULFactory-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
CreateCommand-Methode	Gibt eine stark typisierte System.Data.Common.DbCommand-Instanz zurück
CreateCommandBuilder-Methode	Gibt eine stark typisierte System.Data.Common.DbCommandBuilder-Instanz zurück
CreateConnection-Methode	Gibt eine stark typisierte System.Data.Common.DbConnection-Instanz zurück
CreateConnectionStringBuilder-Methode	Gibt eine stark typisierte System.Data.Common.DbConnectionStringBuilder-Instanz zurück
CreateDataAdapter-Methode	Gibt eine stark typisierte System.Data.Common.DbDataAdapter-Instanz zurück
CreateDataSourceEnumerator-Methode (Geerbt von System.Data.Common.DbProviderFactory)	Gibt eine neue Instanz der Klasse des Providers zurück, die die System.Data.Common.DbDataSourceEnumerator -Klasse implementiert.
CreateParameter-Methode	Gibt eine stark typisierte System.Data.Common.DbParameter-Instanz zurück.

Name	Beschreibung
CreatePermission-Methode (Geerbt von System.Data.Common.DbProviderFactory)	Gibt eine neue Instanz der Klasse des Providers zurück, die die Version des Providers der System.Security.CodeAccessPermission -Klasse implementiert.
CanCreateDataSourceEnumerator-Eigenschaft	Gibt FALSE zurück, um anzuzeigen, dass UltraLite.NET die DbDataSourceEnumerator-Klasse nicht unterstützt.
Instance-Feld	Repräsentiert die einzelne Instanz der ULFactory-Klasse.

Bemerkungen

Die Klasse ULFactory ist in .NET Compact Framework 2.0 nicht verfügbar.

ADO.NET 2.0 führt die beiden neuen Klassen System.Data.Common.DbProviderFactories und System.Data.Common.DbProviderFactory ein, die das Schreiben von Provider-unabhängigem Code vereinfachen. Um diese Klassen mit UltraLite.NET zu verwenden, übergeben Sie iAnywhere.Data.UltraLite als Provider-unveränderlichen Namen an die GetFactory-Methode. Beispiel:

```
' Visual Basic
Dim factory As DbProviderFactory = _
    DbProviderFactories.GetFactory( "iAnywhere.Data.UltraLite" )
Dim conn As DbConnection = _
    factory.CreateConnection()
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
DbProviderFactory factory =
    DbProviderFactories.GetFactory( "iAnywhere.Data.UltraLite" );
DbConnection conn = factory.CreateConnection();
```

In diesem Beispiel wird conn als ein ULConnection-Objekt erstellt.

Eine Beschreibung von Provider-Factorys und generischer Programmierung in ADO.NET 2.0 finden Sie unter [. UltraLite.NET unterstützt die Methoden CreateCommandBuilder\(\), CreateDataSourceEnumerator\(\) und CreatePermission\(\) nicht.](#) <http://msdn.microsoft.com/en-us/library/System.Data.Common.DbProviderFactories.aspx> <http://msdn.microsoft.com/en-us/library/System.Data.Common.DbProviderFactory.aspx> Custom Attribute: sealed

CreateCommand-Methode

Gibt eine stark typisierte System.Data.Common.DbCommand-Instanz zurück

Visual Basic-Syntax

```
Public Overrides Function CreateCommand() As DbCommand
```

C#-Syntax

```
public override DbCommand CreateCommand()
```

Rückgabe

Eine neue ULCommand-Instanz mit dem Datentyp DbCommand.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [System.Data.Common.DbCommand](#)

CreateCommandBuilder-Methode

Gibt eine stark typisierte System.Data.Common.DbCommandBuilder-Instanz zurück

Visual Basic-Syntax

```
Public Overrides Function CreateCommandBuilder() As DbCommandBuilder
```

C#-Syntax

```
public override DbCommandBuilder CreateCommandBuilder()
```

Rückgabe

Eine neue ULCommandBuilder-Instanz mit dem Datentyp DbCommandBuilder.

Siehe auch

- [ULCommandBuilder-Klasse \[UltraLite.NET\] auf Seite 109](#)
- [System.Data.Common.DbCommandBuilder](#)

CreateConnection-Methode

Gibt eine stark typisierte System.Data.Common.DbConnection-Instanz zurück

Visual Basic-Syntax

```
Public Overrides Function CreateConnection() As DbConnection
```

C#-Syntax

```
public override DbConnection CreateConnection()
```

Rückgabe

Eine neue ULConnection-Instanz mit dem Datentyp DbConnection

Siehe auch

- [ULConnection-Klasse \[UltraLite.NET\] auf Seite 121](#)
- [System.Data.Common.DbConnection](#)

CreateConnectionStringBuilder-Methode

Gibt eine stark typisierte System.Data.Common.DbConnectionStringBuilder-Instanz zurück

Visual Basic-Syntax

```
Public Overrides Function CreateConnectionStringBuilder()  
    As DbConnectionStringBuilder
```

C#-Syntax

```
public override DbConnectionStringBuilder CreateConnectionStringBuilder()
```

Rückgabe

Eine neue ULConnectionStringBuilder-Instanz mit dem Datentyp DbConnectionStringBuilder.

Siehe auch

- [ULConnectionStringBuilder-Klasse \[UltraLite.NET\] auf Seite 179](#)
- [System.Data.Common.DbConnectionStringBuilder](#)

CreateDataAdapter-Methode

Gibt eine stark typisierte System.Data.Common.DbDataAdapter-Instanz zurück

Visual Basic-Syntax

```
Public Overrides Function CreateDataAdapter() As DbDataAdapter
```

C#-Syntax

```
public override DbDataAdapter CreateDataAdapter()
```

Rückgabe

Eine neue ULDataAdapter-Instanz mit dem Datentyp von DbDataAdapter.

Siehe auch

- [ULDataAdapter-Klasse \[UltraLite.NET\] auf Seite 212](#)
- [System.Data.Common.DbDataAdapter](#)

CreateParameter-Methode

Gibt eine stark typisierte System.Data.Common.DbParameter-Instanz zurück

Visual Basic-Syntax

```
Public Overrides Function CreateParameter() As DbParameter
```

C#-Syntax

```
public override DbParameter CreateParameter()
```

Rückgabe

Eine neue ULParameter-Instanz mit dem Datentyp DbParameter.

Siehe auch

- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)
- [System.Data.Common.DbParameter](#)

CanCreateDataSourceEnumerator-Eigenschaft

Gibt FALSE zurück, um anzuzeigen, dass UltraLite.NET die DbDataSourceEnumerator-Klasse nicht unterstützt.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property CanCreateDataSourceEnumerator As  
Boolean
```

C#-Syntax

```
public override bool CanCreateDataSourceEnumerator {get;}
```

Bemerkungen

FALSE, um anzuzeigen, dass die ULFactory-Klasse die CreateDataSourceEnumerator-Methode nicht implementiert.

Instance-Feld

Repräsentiert die einzelne Instanz der ULFactory-Klasse.

Visual Basic-Syntax

```
Public Shared ReadOnly Instance As ULFactory
```

C#-Syntax

```
public static readonly ULFactory Instance;
```

Bemerkungen

Die Klasse ULFactory ist in .NET Compact Framework 2.0 nicht verfügbar.

ULFactory ist eine einzelne Klasse. Das heißt nur, dass diese Instanz dieser Klasse vorhanden sein kann.

Normalerweise wird dieses Feld nicht direkt verwendet. Stattdessen erhalten Sie eine Referenz dieser Instanz von der ULFactory-Klasse mithilfe der
System.Data.Common.DbProviderFactories.GetFactory(String)-Methode.

Siehe auch

- [ULFactory-Klasse \[UltraLite.NET\] auf Seite 277](#)

ULFileTransfer-Klasse

UL-Erw.: Überträgt eine Datei aus einer entfernten Datenbank mithilfe des MobiLink-Servers.

Visual Basic-Syntax

```
Public NotInheritable Class ULFileTransfer
```

C#-Syntax

```
public sealed class ULFileTransfer
```

Mitglieder

Alle Mitglieder der ULFileTransfer-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ULFileTransfer-Konstruktor	Initialisiert ein ULFileTransfer-Objekt.
DownloadFile-Methode	Lädt die Datei herunter, die durch die Eigenschaften dieses Objekts festgelegt wird.
UploadFile-Methode	Lädt die Datei hoch, die durch die Eigenschaften dieses Objekts festgelegt wird.
AuthenticationParms-Eigenschaft	Legt Parameter für ein angepasstes Benutzerauthentifizierungsskript (MobiLink authenticate_parameters-Verbindungsereignis) fest.
AuthStatus-Eigenschaft	Gibt den Autorisierungsstatuscode für den letzten Dateiübertragungsversuch zurück.
AuthValue-Eigenschaft	Gibt den Rückgabewert aus Synchronisationsskripten mit angepasster Benutzerauthentifizierung zurück.
FileAuthCode-Eigenschaft	Gibt den Rückgabewert aus dem Skript authenticate_file_transfer für den letzten Dateiübertragungsversuch zurück.
FileName-Eigenschaft	Gibt den Namen der Spalte an, die heruntergeladen werden soll.
LocalFileName-Eigenschaft	Legt den Namen der lokalen Datei für die heruntergeladene Datei fest.
LocalPath-Eigenschaft	Legt fest, an welchen Ort die Datei heruntergeladen wird.
Password-Eigenschaft	Das MobiLink-Kennwort für den mit dem UserName-Wert angegebenen Benutzer.
RemoteKey-Eigenschaft	Der Schlüssel, der den MobiLink-Client für den MobiLink-Server eindeutig kennzeichnet.

Name	Beschreibung
ResumePartialDownload-Eigenschaft	Legt fest, ob ein früherer Teil-Download wieder aufgenommen oder entfernt werden soll.
Stream-Eigenschaft	Gibt den MobiLink-Synchronisationsdatenstrom für die Verwendung bei der Dateiübertragung an.
StreamErrorCode-Eigenschaft	Gibt den Fehler zurück, der vom Datenstrom selbst für den letzten Dateiübertragungsversuch gemeldet wird.
StreamErrorSystem-Eigenschaft	Gibt den systemspezifischen Code für den Datenstromfehler zurück.
StreamParms-Eigenschaft	Gibt die Parameter zur Konfiguration des Synchronisationsdatenstroms an.
TransferredFile-Eigenschaft	Prüft, ob die Datei tatsächlich während des letzten Dateiübertragungsversuchs heruntergeladen wurde.
UserName-Eigenschaft	Der Benutzername, der den MobiLink-Client für den MobiLink-Server kennzeichnet.
Version-Eigenschaft	Legt fest, welches Synchronisationsskript verwendet werden soll.

Bemerkungen

Es ist nicht erforderlich, für eine Datenübertragung eine Datenbankverbindung herzustellen. Wenn die Anwendung jedoch eine UltraLite-Datenbank mit der UltraLite-Laufzeit-Engine verwendet, müssen Sie den `ULDatabaseManager.RuntimeType`-Wert auf den entsprechenden Wert setzen, bevor Sie diese API-Schnittstelle oder eine andere UltraLite.NET API-Schnittstelle verwenden.

Um eine Datei zu übertragen, müssen Sie die Werte `ULFileTransfer.FileName`, `ULFileTransfer.Stream`, `ULFileTransfer.UserName` und `ULFileTransfer.Version` festlegen.

Siehe auch

- [ULFileTransfer.FileName-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)
- [ULFileTransfer.Stream-Eigenschaft \[UltraLite.NET\] auf Seite 294](#)
- [ULFileTransfer.UserName-Eigenschaft \[UltraLite.NET\] auf Seite 296](#)
- [ULFileTransfer.Version-Eigenschaft \[UltraLite.NET\] auf Seite 297](#)

ULFileTransfer-Konstruktor

Initialisiert ein `ULFileTransfer`-Objekt.

Visual Basic-Syntax

```
Public Sub New()
```

C#-Syntax

```
public ULFileTransfer()
```

Bemerkungen

Die Verbindung muss geöffnet werden, damit Sie Vorgänge mit der Datenbank ausführen können.

Es ist nicht erforderlich, für eine Datenübertragung eine Datenbankverbindung herzustellen. Wenn die Anwendung jedoch eine UltraLite-Datenbank mit der UltraLite-Laufzeit-Engine verwendet, müssen Sie den ULDatabaseManager.RuntimeType-Wert auf den entsprechenden Wert setzen, bevor Sie diese API-Schnittstelle oder eine andere UltraLite.NET API-Schnittstelle verwenden.

Für das ULFileTransfer-Objekt müssen die Werte ULFileTransfer.FileName, ULFileTransfer.Stream, ULFileTransfer.UserName und ULFileTransfer.Version gesetzt werden, damit es eine Datei übertragen kann.

Siehe auch

- [ULFileTransfer.DownloadFile-Methode \[UltraLite.NET\] auf Seite 284](#)
- [ULFileTransfer.UploadFile-Methode \[UltraLite.NET\] auf Seite 287](#)
- [ULFileTransfer.FileName-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)
- [ULFileTransfer.Stream-Eigenschaft \[UltraLite.NET\] auf Seite 294](#)
- [ULFileTransfer.UserName-Eigenschaft \[UltraLite.NET\] auf Seite 296](#)
- [ULFileTransfer.Version-Eigenschaft \[UltraLite.NET\] auf Seite 297](#)

DownloadFile-Methode

Lädt die Datei herunter, die durch die Eigenschaften dieses Objekts festgelegt wird.

Überladungsliste

Name	Beschreibung
DownloadFile()-Methode	Lädt die Datei herunter, die durch die Eigenschaften dieses Objekts festgelegt wird.
DownloadFile(ULFileTransferProgressListener)-Methode	Lädt die Datei herunter, die durch die Eigenschaften dieser Datei festgelegt wird, wobei an den angegebenen Listener Fortschrittsereignisse gesendet werden

DownloadFile()-Methode

Lädt die Datei herunter, die durch die Eigenschaften dieses Objekts festgelegt wird.

Visual Basic-Syntax

```
Public Function DownloadFile() As Boolean
```

C#-Syntax

```
public bool DownloadFile()
```

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE. (Prüfen Sie den ULFileTransfer.StreamErrorCode-Wert und andere Statureigenschaften auf Gründe.)

Bemerkungen

Die vom ULFileTransfer.FileName-Wert angegebene Datei wird vom MobiLink-Server auf den ULFileTransfer.LocalPath-Wert unter Verwendung der Werte ULFileTransfer.Stream, ULFileTransfer.UserName, ULFileTransfer.Password und ULFileTransfer.Version heruntergeladen.

Um eine Beschädigung der Datei zu vermeiden, lädt UltraLite.NET sie in eine temporäre Datei und ersetzt die lokale Datei erst, wenn der Download abgeschlossen ist. Andere Eigenschaften, die den Download beeinflussen, sind ULFileTransfer.LocalFileName, ULFileTransfer.AuthenticationParms und ULFileTransfer.ResumePartialDownload.

Ein ausführlicher Ergebnisstatusbericht wird in den Werten ULFileTransfer.AuthStatus, ULFileTransfer.AuthValue, ULFileTransfer.FileAuthCode, ULFileTransfer.TransferredFile, ULFileTransfer.StreamErrorCode und ULFileTransfer.StreamErrorSystem dieses Objekts ausgewiesen.

Siehe auch

- [ULFileTransfer.DownloadFile-Methode \[UltraLite.NET\] auf Seite 284](#)
- [ULFileTransfer.FileName-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)
- [ULFileTransfer.LocalPath-Eigenschaft \[UltraLite.NET\] auf Seite 292](#)
- [ULFileTransfer.Stream-Eigenschaft \[UltraLite.NET\] auf Seite 294](#)
- [ULFileTransfer.UserName-Eigenschaft \[UltraLite.NET\] auf Seite 296](#)
- [ULFileTransfer.Password-Eigenschaft \[UltraLite.NET\] auf Seite 292](#)
- [ULFileTransfer.Version-Eigenschaft \[UltraLite.NET\] auf Seite 297](#)
- [ULFileTransfer.LocalFileName-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)
- [ULFileTransfer.AuthenticationParms-Eigenschaft \[UltraLite.NET\] auf Seite 289](#)
- [ULFileTransfer.ResumePartialDownload-Eigenschaft \[UltraLite.NET\] auf Seite 293](#)
- [ULFileTransfer.AuthStatus-Eigenschaft \[UltraLite.NET\] auf Seite 290](#)
- [ULFileTransfer.AuthValue-Eigenschaft \[UltraLite.NET\] auf Seite 290](#)
- [ULFileTransfer.FileAuthCode-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)
- [Die ULFileTransfer.TransferredFile-Eigenschaft \[UltraLite.NET\] auf Seite 296](#)
- [ULFileTransfer.StreamErrorCode-Eigenschaft \[UltraLite.NET\] auf Seite 295](#)
- [ULFileTransfer.StreamErrorSystem-Eigenschaft \[UltraLite.NET\] auf Seite 295](#)
- [ULFileTransfer.StreamErrorCode-Eigenschaft \[UltraLite.NET\] auf Seite 295](#)
- [ULFileTransfer.UploadFile-Methode \[UltraLite.NET\] auf Seite 287](#)

DownloadFile(ULFileTransferProgressListener)-Methode

Lädt die Datei herunter, die durch die Eigenschaften dieser Datei festgelegt wird, wobei an den angegebenen Listener Fortschrittsereignisse gesendet werden

Visual Basic-Syntax

```
Public Function DownloadFile(  
    ByVal listener As ULFileTransferProgressListener  
) As Boolean
```

C#-Syntax

```
public bool DownloadFile(ULFileTransferProgressListener listener)
```

Parameter

- **listener** Das Objekt, das Ereignisse zum Dateiübertragungsfortschritt empfängt.

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE. (Prüfen Sie den ULFileTransfer.StreamErrorCode-Wert und andere Statureigenschaften auf Gründe.)

Bemerkungen

Die vom ULFileTransfer.FileName-Wert angegebene Datei wird vom MobiLink-Server auf den ULFileTransfer.LocalPath-Wert unter Verwendung der Werte ULFileTransfer.Stream, ULFileTransfer.UserName, ULFileTransfer.Password und ULFileTransfer.Version heruntergeladen.

Um eine Beschädigung der Datei zu vermeiden, lädt UltraLite.NET sie in eine temporäre Datei und ersetzt die lokale Datei erst, wenn der Download abgeschlossen ist. Andere Eigenschaften, die den Download beeinflussen, sind ULFileTransfer.LocalFileName, ULFileTransfer.AuthenticationParms und ULFileTransfer.ResumePartialDownload.

Ein ausführlicher Ergebnisstatusbericht wird in den Werten ULFileTransfer.AuthStatus, ULFileTransfer.AuthValue, ULFileTransfer.FileAuthCode, ULFileTransfer.TransferredFile, ULFileTransfer.StreamErrorCode und ULFileTransfer.StreamErrorSystem dieses Objekts ausgewiesen.

Fehler können dazu führen, dass keine Daten an den Listener gesendet werden.

Siehe auch

- [ULFileTransfer.FileName-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)
- [ULFileTransfer.LocalPath-Eigenschaft \[UltraLite.NET\] auf Seite 292](#)
- [ULFileTransfer.Stream-Eigenschaft \[UltraLite.NET\] auf Seite 294](#)
- [ULFileTransfer.UserName-Eigenschaft \[UltraLite.NET\] auf Seite 296](#)
- [ULFileTransfer.Password-Eigenschaft \[UltraLite.NET\] auf Seite 292](#)
- [ULFileTransfer.Version-Eigenschaft \[UltraLite.NET\] auf Seite 297](#)
- [ULFileTransfer.LocalFileName-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)
- [ULFileTransfer.AuthenticationParms-Eigenschaft \[UltraLite.NET\] auf Seite 289](#)
- [ULFileTransfer.ResumePartialDownload-Eigenschaft \[UltraLite.NET\] auf Seite 293](#)
- [ULFileTransfer.AuthStatus-Eigenschaft \[UltraLite.NET\] auf Seite 290](#)
- [ULFileTransfer.AuthValue-Eigenschaft \[UltraLite.NET\] auf Seite 290](#)
- [ULFileTransfer.FileAuthCode-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)
- [Die ULFileTransfer.TransferredFile-Eigenschaft \[UltraLite.NET\] auf Seite 296](#)
- [ULFileTransfer.StreamErrorCode-Eigenschaft \[UltraLite.NET\] auf Seite 295](#)
- [ULFileTransfer.StreamErrorSystem-Eigenschaft \[UltraLite.NET\] auf Seite 295](#)
- [ULFileTransfer.StreamErrorCode-Eigenschaft \[UltraLite.NET\] auf Seite 295](#)
- [ULFileTransfer.UploadFile-Methode \[UltraLite.NET\] auf Seite 287](#)

UploadFile-Methode

Lädt die Datei hoch, die durch die Eigenschaften dieses Objekts festgelegt wird.

Überladungsliste

Name	Beschreibung
UploadFile()-Methode	Lädt die Datei hoch, die durch die Eigenschaften dieses Objekts festgelegt wird.
UploadFile(ULFileTransfer-ProgressListener)-Methode	Lädt die Datei hoch, die durch die Eigenschaften dieser Datei festgelegt wird, wobei Fortschrittsereignisse an den angegebenen Listener gesendet werden.

UploadFile()-Methode

Lädt die Datei hoch, die durch die Eigenschaften dieses Objekts festgelegt wird.

Visual Basic-Syntax

```
Public Function UploadFile() As Boolean
```

C#-Syntax

```
public bool UploadFile()
```

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE. (Prüfen Sie den `ULFileTransfer.StreamErrorCode`-Wert und andere Statureigenschaften auf Gründe.)

Bemerkungen

Die vom `ULFileTransfer.FileName`-Wert angegebene Datei wird auf den MobiLink-Server vom `ULFileTransfer.LocalPath`-Wert hochgeladen, unter Verwendung der Werte `ULFileTransfer.Stream`, `ULFileTransfer.UserName`, `ULFileTransfer.Password` und `ULFileTransfer.Version`.

Ein ausführlicher Ergebnisstatusbericht wird in den Werten `ULFileTransfer.AuthStatus`, `ULFileTransfer.AuthValue`, `ULFileTransfer.FileAuthCode`, `ULFileTransfer.TransferredFile`, `ULFileTransfer.StreamErrorCode` und `ULFileTransfer.StreamErrorSystem` dieses Objekts ausgewiesen.

Siehe auch

- [ULFileTransfer.UploadFile-Methode \[UltraLite.NET\] auf Seite 287](#)
- [ULFileTransfer.FileName-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)
- [ULFileTransfer.LocalPath-Eigenschaft \[UltraLite.NET\] auf Seite 292](#)
- [ULFileTransfer.Stream-Eigenschaft \[UltraLite.NET\] auf Seite 294](#)
- [ULFileTransfer.UserName-Eigenschaft \[UltraLite.NET\] auf Seite 296](#)
- [ULFileTransfer.Password-Eigenschaft \[UltraLite.NET\] auf Seite 292](#)
- [ULFileTransfer.Version-Eigenschaft \[UltraLite.NET\] auf Seite 297](#)
- [ULFileTransfer.LocalFileName-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)
- [ULFileTransfer.AuthenticationParms-Eigenschaft \[UltraLite.NET\] auf Seite 289](#)
- [ULFileTransfer.ResumePartialDownload-Eigenschaft \[UltraLite.NET\] auf Seite 293](#)
- [ULFileTransfer.AuthStatus-Eigenschaft \[UltraLite.NET\] auf Seite 290](#)
- [ULFileTransfer.AuthValue-Eigenschaft \[UltraLite.NET\] auf Seite 290](#)
- [ULFileTransfer.FileAuthCode-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)
- [Die ULFileTransfer.TransferredFile-Eigenschaft \[UltraLite.NET\] auf Seite 296](#)
- [ULFileTransfer.StreamErrorCode-Eigenschaft \[UltraLite.NET\] auf Seite 295](#)
- [ULFileTransfer.StreamErrorSystem-Eigenschaft \[UltraLite.NET\] auf Seite 295](#)
- [ULFileTransfer.StreamErrorCode-Eigenschaft \[UltraLite.NET\] auf Seite 295](#)

UploadFile(ULFileTransferProgressListener)-Methode

Lädt die Datei hoch, die durch die Eigenschaften dieser Datei festgelegt wird, wobei Fortschrittsereignisse an den angegebenen Listener gesendet werden.

Visual Basic-Syntax

```
Public Function UploadFile(  
    ByVal listener As ULFileTransferProgressListener  
) As Boolean
```

C#-Syntax

```
public bool UploadFile(ULFileTransferProgressListener listener)
```

Parameter

- **listener** Das Objekt, das Ereignisse zum Dateiübertragungsfortschritt empfängt.

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE. (Prüfen Sie den ULFileTransfer.StreamErrorCode-Wert und andere Statureigenschaften auf Gründe.)

Bemerkungen

Die vom ULFileTransfer.FileName-Wert angegebene Datei wird auf den MobiLink-Server vom ULFileTransfer.LocalPath-Wert hochgeladen, unter Verwendung der Werte ULFileTransfer.Stream, ULFileTransfer.UserName, ULFileTransfer.Password und ULFileTransfer.Version.

Ein ausführlicher Ergebnisstatusbericht wird in den Werten ULFileTransfer.AuthStatus, ULFileTransfer.AuthValue, ULFileTransfer.FileAuthCode, ULFileTransfer.TransferredFile, ULFileTransfer.StreamErrorCode und ULFileTransfer.StreamErrorSystem dieses Objekts ausgewiesen.

Fehler können dazu führen, dass keine Daten an den Listener gesendet werden.

Siehe auch

- [ULFileTransfer.UploadFile-Methode \[UltraLite.NET\] auf Seite 287](#)
- [ULFileTransfer.FileName-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)
- [ULFileTransfer.LocalPath-Eigenschaft \[UltraLite.NET\] auf Seite 292](#)
- [ULFileTransfer.Stream-Eigenschaft \[UltraLite.NET\] auf Seite 294](#)
- [ULFileTransfer.UserName-Eigenschaft \[UltraLite.NET\] auf Seite 296](#)
- [ULFileTransfer.Password-Eigenschaft \[UltraLite.NET\] auf Seite 292](#)
- [ULFileTransfer.Version-Eigenschaft \[UltraLite.NET\] auf Seite 297](#)
- [ULFileTransfer.LocalFileName-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)
- [ULFileTransfer.AuthenticationParms-Eigenschaft \[UltraLite.NET\] auf Seite 289](#)
- [ULFileTransfer.ResumePartialDownload-Eigenschaft \[UltraLite.NET\] auf Seite 293](#)
- [ULFileTransfer.AuthStatus-Eigenschaft \[UltraLite.NET\] auf Seite 290](#)
- [ULFileTransfer.AuthValue-Eigenschaft \[UltraLite.NET\] auf Seite 290](#)
- [ULFileTransfer.FileAuthCode-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)
- [Die ULFileTransfer.TransferredFile-Eigenschaft \[UltraLite.NET\] auf Seite 296](#)
- [ULFileTransfer.StreamErrorCode-Eigenschaft \[UltraLite.NET\] auf Seite 295](#)
- [ULFileTransfer.StreamErrorSystem-Eigenschaft \[UltraLite.NET\] auf Seite 295](#)
- [ULFileTransfer.StreamErrorCode-Eigenschaft \[UltraLite.NET\] auf Seite 295](#)

AuthenticationParms-Eigenschaft

Legt Parameter für ein angepasstes Benutzerauthentifizierungsskript (MobiLink authenticate_parameters-Verbindungsereignis) fest.

Visual Basic-Syntax

```
Public Property AuthenticationParms As String()
```

C#-Syntax

```
public String[] AuthenticationParms {get;set;}
```

Bemerkungen

Ein Array von Zeichenfolgen, die jeweils einen Authentifizierungsparameter enthalten (null Array-Einträge werden einen Synchronisationsfehler ergeben). Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic). Das bedeutet, dass keine Authentifizierungsparameter verwendet werden.

Nur die ersten 255 Zeichenfolgen werden verwendet und keine Zeichenfolge sollte die Grenze des MobiLink-Servers für Authentifizierungsparameter überschreiten (derzeit 4000 UTF8-Byte).

AuthStatus-Eigenschaft

Gibt den Autorisierungsstatuscode für den letzten Dateiübertragungsversuch zurück.

Visual Basic-Syntax

```
Public ReadOnly Property AuthStatus As ULAuthStatusCode
```

C#-Syntax

```
public ULAuthStatusCode AuthStatus {get;}
```

Bemerkungen

Einer der ULAuthStatusCode-Werte, die den Autorisierungsstatus für den letzten Dateiübertragungsversuch angeben.

Siehe auch

- [ULAuthStatusCode-Enumeration \[UltraLite.NET\] auf Seite 456](#)

AuthValue-Eigenschaft

Gibt den Rückgabewert aus Synchronisationsskripten mit angepasster Benutzerauthentifizierung zurück.

Visual Basic-Syntax

```
Public ReadOnly Property AuthValue As Long
```

C#-Syntax

```
public long AuthValue {get;}
```

Bemerkungen

Eine Ganzzahl (long integer) aus Synchronisationsskripten mit angepasster Benutzerauthentifizierung.

FileAuthCode-Eigenschaft

Gibt den Rückgabewert aus dem Skript `authenticate_file_transfer` für den letzten Dateiübertragungsversuch zurück.

Visual Basic-Syntax

```
Public ReadOnly Property FileAuthCode As UShort
```

C#-Syntax

```
public ushort FileAuthCode {get;}
```

Bemerkungen

Eine kurze Ganzzahl ohne Vorzeichen, die vom Skript `authenticate_file_transfer` für den letzten Dateiübertragungsversuch zurückgegeben wird.

FileName-Eigenschaft

Gibt den Namen der Spalte an, die heruntergeladen werden soll.

Visual Basic-Syntax

```
Public Property FileName As String
```

C#-Syntax

```
public string FileName {get;set;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen der Datei zurückgibt, der vom MobiLink-Server erkannt wurde. Diese Eigenschaft hat keinen Standardwert und muss explizit festgelegt werden.

Der FileName-Wert ist der Name der Datei auf dem Server, auf dem MobiLink ausgeführt wird. MobiLink sucht zunächst im UserName-Unterverzeichnis und dann im Stammverzeichnis nach dieser Datei (das Stammverzeichnis wird mit der MobiLink-Serveroption `-ftr` festgelegt). Der FileName-Wert darf keine Laufwerks- oder Pfadinformationen enthalten, damit sie vom MobiLink-Server gefunden werden kann. Beispiel: `"myfile.txt"` ist gültig, doch `"somedir\myfile.txt"`, `"..\myfile.txt"` und `"c:\myfile.txt"` sind jeweils ungültig.

Siehe auch

- [ULFileTransfer.LocalFileName-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)
- [ULFileTransfer.DownloadFile-Methode \[UltraLite.NET\] auf Seite 284](#)
- [ULFileTransfer.UploadFile-Methode \[UltraLite.NET\] auf Seite 287](#)

LocalFileName-Eigenschaft

Legt den Namen der lokalen Datei für die heruntergeladene Datei fest.

Visual Basic-Syntax

```
Public Property LocalFileName As String
```

C#-Syntax

```
public string LocalFileName {get;set;}
```

Bemerkungen

Eine Zeichenfolge, die den lokalen Dateinamen für die heruntergeladene Datei festlegt. Der FileName-Wert wird verwendet, wenn der Wert eine Nullreferenz ist ("Nothing" in Visual Basic). Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Siehe auch

- [ULFileTransfer.FileName-Eigenschaft \[UltraLite.NET\] auf Seite 291](#)

LocalPath-Eigenschaft

Legt fest, an welchen Ort die Datei heruntergeladen wird.

Visual Basic-Syntax

```
Public Property LocalPath As String
```

C#-Syntax

```
public string LocalPath {get;set;}
```

Bemerkungen

Eine Zeichenfolge, die das lokale Verzeichnis der Datei angibt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Das lokale Standardverzeichnis variiert je nach Betriebssystem des Geräts:

- Wenn der LocalPath-Wert bei Windows Mobile-Geräten eine Nullreferenz ist ("Nothing" in Visual Basic), wird die Datei im Stammverzeichnis (\) gespeichert.
- Wenn der LocalPath-Wert bei PC-Anwendungen eine Nullreferenz ist ("Nothing" in Visual Basic), wird die Datei im aktuellen Verzeichnis gespeichert.

Siehe auch

- [ULFileTransfer.DownloadFile-Methode \[UltraLite.NET\] auf Seite 284](#)
- [ULFileTransfer.UploadFile-Methode \[UltraLite.NET\] auf Seite 287](#)

Password-Eigenschaft

Das MobiLink-Kennwort für den mit dem UserName-Wert angegebenen Benutzer.

Visual Basic-Syntax

```
Public Property Password As String
```

C#-Syntax

```
public string Password {get;set;}
```

Bemerkungen

Eine Zeichenfolge, die das MobiLink-Kennwort festlegt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic). Das bedeutet, dass kein Kennwort angegeben wurde.

Der MobiLink-Benutzername und das Kennwort unterscheiden sich von der ID und dem Kennwort des Datenbankbenutzers und dienen der Kennzeichnung und Authentifizierung der Anwendung beim MobiLink-Server.

Siehe auch

- [ULFileTransfer.UserName-Eigenschaft \[UltraLite.NET\] auf Seite 296](#)
- [ULFileTransfer.DownloadFile-Methode \[UltraLite.NET\] auf Seite 284](#)
- [ULFileTransfer.UploadFile-Methode \[UltraLite.NET\] auf Seite 287](#)

RemoteKey-Eigenschaft

Der Schlüssel, der den MobiLink-Client für den MobiLink-Server eindeutig kennzeichnet.

Visual Basic-Syntax

```
Public Property RemoteKey As String
```

C#-Syntax

```
public string RemoteKey {get;set;}
```

Bemerkungen

Eine Zeichenfolge, die den Remote-Schlüssel angibt. Diese Eigenschaft hat keinen Standardwert und muss explizit festgelegt werden.

Der MobiLink-Server übergibt diesen Wert an verschiedene Skripten, um diesen Client eindeutig zu identifizieren.

Siehe auch

- [ULFileTransfer.DownloadFile-Methode \[UltraLite.NET\] auf Seite 284](#)
- [ULFileTransfer.UploadFile-Methode \[UltraLite.NET\] auf Seite 287](#)

ResumePartialDownload-Eigenschaft

Legt fest, ob ein früherer Teil-Download wieder aufgenommen oder entfernt werden soll.

Visual Basic-Syntax

Public Property **ResumePartialDownload** As Boolean

C#-Syntax

```
public bool ResumePartialDownload {get;set;}
```

Bemerkungen

TRUE, wenn ein früherer Teil-Download wiederaufgenommen werden soll. FALSE, wenn er entfernt werden soll. Der Standardwert ist FALSE.

UltraLite.NET kann das ULFileTransferListener-Objekt verwenden, um Downloads fortzusetzen, die aufgrund von Kommunikationsfehlern oder Abbruch durch den Benutzer fehlgeschlagen sind. UltraLite.NET verarbeitet den Download so, wie er empfangen wird. Wenn ein Download unterbrochen wird, bleibt die Teil-Download-Datei erhalten und kann bei der nächsten Dateiübertragung wieder aufgenommen werden.

Wenn die Datei auf dem Server aktualisiert wird, wird ein Teil-Download entfernt und ein neuer Download gestartet.

Siehe auch

- [ULFileTransfer.DownloadFile-Methode \[UltraLite.NET\] auf Seite 284](#)
- [ULFileTransfer.UploadFile-Methode \[UltraLite.NET\] auf Seite 287](#)

Stream-Eigenschaft

Gibt den MobiLink-Synchronisationsdatenstrom für die Verwendung bei der Dateiübertragung an.

Visual Basic-Syntax

Public Property **Stream** As ULStreamType

C#-Syntax

```
public ULStreamType Stream {get;set;}
```

Bemerkungen

Einer der ULStreamType-Werte, die den Typ des zu verwendenden Synchronisationsdatenstroms festlegen. Der Standardwert ist der ULStreamType.TCPIP-Wert.

Die meisten Synchronisationsdatenströme benötigen Parameter, um die Adresse des MobiLink-Servers zu bestimmen und anderes Verhalten zu steuern. Diese Parameter werden durch den ULFileTransfer.StreamParms-Wert bereitgestellt.

Wenn der Datenstromtyp auf einen für die Plattform unzulässigen Wert gesetzt wird, wird der Datenstromparameter auf den ULStreamType.TCPIP-Wert gesetzt.

Siehe auch

- [ULStreamType-Enumeration \[UltraLite.NET\] auf Seite 464](#)
- [ULFileTransfer.StreamParams-Eigenschaft \[UltraLite.NET\] auf Seite 295](#)
- [ULFileTransfer.DownloadFile-Methode \[UltraLite.NET\] auf Seite 284](#)
- [ULFileTransfer.UploadFile-Methode \[UltraLite.NET\] auf Seite 287](#)

StreamErrorCode-Eigenschaft

Gibt den Fehler zurück, der vom Datenstrom selbst für den letzten Dateiübertragungsversuch gemeldet wird.

Visual Basic-Syntax

```
Public ReadOnly Property StreamErrorCode As ULStreamErrorCode
```

C#-Syntax

```
public ULStreamErrorCode StreamErrorCode {get;}
```

Bemerkungen

Einer der ULStreamErrorCode-Werte, die den Fehler angeben, der vom Datenstrom selbst gemeldet wurde. Der ULStreamErrorCode.NONE-Wert, wenn kein Fehler aufgetreten ist.

StreamErrorSystem-Eigenschaft

Gibt den systemspezifischen Code für den Datenstromfehler zurück.

Visual Basic-Syntax

```
Public ReadOnly Property StreamErrorSystem As Integer
```

C#-Syntax

```
public int StreamErrorSystem {get;}
```

Bemerkungen

Eine Ganzzahl, die den systemspezifischen Code des Datenstromfehlers angibt.

StreamParams-Eigenschaft

Gibt die Parameter zur Konfiguration des Synchronisationsdatenstroms an.

Visual Basic-Syntax

```
Public Property StreamParams As String
```

C#-Syntax

```
public string StreamParams {get;set;}
```

Bemerkungen

Eine Zeichenfolge in der Form einer durch Semikola getrennten Liste von Schlüsselwort-Wert-Paaren, die die Parameter für den Datenstrom festlegt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Der StreamParms-Wert ist eine Zeichenfolge, die alle Parameter enthält, die für Synchronisationsdatenströme verwendet werden. Parameter werden in einer durch Semikola getrennten Liste mit Paaren der Form Name=Wert ("Param1=Wert1;Param2=Wert2") angegeben.

Siehe auch

- [ULFileTransfer.Stream-Eigenschaft \[UltraLite.NET\] auf Seite 294](#)
- [ULFileTransfer.DownloadFile-Methode \[UltraLite.NET\] auf Seite 284](#)
- [ULFileTransfer.UploadFile-Methode \[UltraLite.NET\] auf Seite 287](#)
- [ULStreamType-Enumeration \[UltraLite.NET\] auf Seite 464](#)
- „UltraLite-Netzwerkprotokolloptionen für dbmlsync“ [[UltraLite - Datenbankverwaltung](#)]

TransferredFile-Eigenschaft

Prüft, ob die Datei tatsächlich während des letzten Dateiübertragungsversuchs heruntergeladen wurde.

Visual Basic-Syntax

```
Public ReadOnly Property TransferredFile As Boolean
```

C#-Syntax

```
public bool TransferredFile {get;}
```

Bemerkungen

TRUE, wenn die Datei heruntergeladen wurde, andernfalls FALSE.

Falls die Datei bereits aktuell ist, wenn die DownloadFile- oder die UploadFile-Methode aufgerufen wird, wird TRUE zurückgegeben, aber TransferredFile ist FALSE. Wenn ein Fehler auftritt und die DownloadFile- oder die UploadFile-Methode FALSE zurückgibt, ist TransferredFile FALSE.

Siehe auch

- [ULFileTransfer.DownloadFile-Methode \[UltraLite.NET\] auf Seite 284](#)
- [ULFileTransfer.UploadFile-Methode \[UltraLite.NET\] auf Seite 287](#)

UserName-Eigenschaft

Der Benutzername, der den MobiLink-Client für den MobiLink-Server kennzeichnet.

Visual Basic-Syntax

```
Public Property UserName As String
```

C#-Syntax

```
public string UserName {get;set;}
```

Bemerkungen

Eine Zeichenfolge, die den Benutzernamen festlegt. Diese Eigenschaft hat keinen Standardwert und muss explizit festgelegt werden.

Der MobiLink-Server verwendet diesen Wert, um die herunterzuladende Datei zu finden. Der MobiLink-Benutzername und das Kennwort unterscheiden sich von der ID und dem Kennwort des Datenbankbenutzers und dienen der Kennzeichnung und Authentifizierung der Anwendung beim MobiLink-Server.

Siehe auch

- [ULFileTransfer.Password-Eigenschaft \[UltraLite.NET\] auf Seite 292](#)
- [ULFileTransfer.DownloadFile-Methode \[UltraLite.NET\] auf Seite 284](#)
- [ULFileTransfer.UploadFile-Methode \[UltraLite.NET\] auf Seite 287](#)

Version-Eigenschaft

Legt fest, welches Synchronisationsskript verwendet werden soll.

Visual Basic-Syntax

```
Public Property Version As String
```

C#-Syntax

```
public string Version {get;set;}
```

Bemerkungen

Eine Zeichenfolge, die die Version des zu verwendenden Synchronisationsskripts angibt. Diese Eigenschaft hat keinen Standardwert und muss explizit festgelegt werden.

Jedes Synchronisationsskript in der konsolidierten Datenbank wird mit einer Versionszeichenfolge markiert. Die Versionszeichenfolge gestattet es einer UltraLite-Anwendung, zwischen einer Reihe von Synchronisationsskripten zu wählen.

Siehe auch

- [ULFileTransfer.DownloadFile-Methode \[UltraLite.NET\] auf Seite 284](#)
- [ULFileTransfer.UploadFile-Methode \[UltraLite.NET\] auf Seite 287](#)

ULFileTransferProgressData-Klasse

UL-Erw.: Gibt Überwachungsdaten zum Dateiübertragungsfortschritt zurück.

Visual Basic-Syntax

```
Public Class ULFileTransferProgressData
```

C#-Syntax

```
public class ULFileTransferProgressData
```

Mitglieder

Alle Mitglieder der ULFileTransferProgressData-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
BytesReceived-Eigenschaft	Gibt die Anzahl der bislang empfangenen Bytes zurück.
FileSize-Eigenschaft	Gibt die Größe der Datei zurück, die übertragen wird.
Flags-Eigenschaft	Gibt die aktuellen Optionen für die Dateiübertragung zurück, die weitere Informationen über den aktuellen Status liefern.
ResumedAtSize-Eigenschaft	Gibt die Position in der Datei zurück, an der die Übertragung wieder aufgenommen wurde.
FLAG_IS_BLOCKING-Feld	Eine Option, die anzeigt, dass die Dateiübertragung blockiert ist, während auf eine Antwort vom MobiLink-Server gewartet wird.

Siehe auch

- [ULFileTransferProgressListener-Schnittstelle \[UltraLite.NET\] auf Seite 300](#)

BytesReceived-Eigenschaft

Gibt die Anzahl der bislang empfangenen Bytes zurück.

Visual Basic-Syntax

```
Public ReadOnly Property BytesReceived As ULong
```

C#-Syntax

```
public ulong BytesReceived {get;}
```

Bemerkungen

Die Anzahl der bislang empfangen Bytes.

FileSize-Eigenschaft

Gibt die Größe der Datei zurück, die übertragen wird.

Visual Basic-Syntax

```
Public ReadOnly Property FileSize As ULong
```

C#-Syntax

```
public ulong FileSize {get;}
```

Bemerkungen

Die Größe der Datei in Byte.

Flags-Eigenschaft

Gibt die aktuellen Optionen für die Dateiübertragung zurück, die weitere Informationen über den aktuellen Status liefern.

Visual Basic-Syntax

```
Public ReadOnly Property Flags As Integer
```

C#-Syntax

```
public int Flags {get;}
```

Bemerkungen

Eine Ganzzahl, die eine durch OR verknüpfte Kombination von Optionen enthält.

Siehe auch

- [ULFileTransferProgressData.FLAG_IS_BLOCKING-Feld \[UltraLite.NET\] auf Seite 299](#)

ResumedAtSize-Eigenschaft

Gibt die Position in der Datei zurück, an der die Übertragung wieder aufgenommen wurde.

Visual Basic-Syntax

```
Public ReadOnly Property ResumedAtSize As ULong
```

C#-Syntax

```
public ulong ResumedAtSize {get;}
```

Bemerkungen

Die Anzahl der zuvor übertragenen Bytes.

FLAG_IS_BLOCKING-Feld

Eine Option, die anzeigt, dass die Dateiübertragung blockiert ist, während auf eine Antwort vom MobiLink-Server gewartet wird.

Visual Basic-Syntax

```
Public Const FLAG_IS_BLOCKING As Integer
```

C#-Syntax

```
public const int FLAG_IS_BLOCKING;
```

ULFileTransferProgressListener-Schnittstelle

UL-Erw.: Die Listener-Schnittstelle für den Empfang von Dateiübertragungsfortschritt-Ereignissen.

Visual Basic-Syntax

```
Public Interface ULFileTransferProgressListener
```

C#-Syntax

```
public interface ULFileTransferProgressListener
```

Mitglieder

Alle Mitglieder der ULFileTransferProgressListener-Schnittstelle, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
FileTransferProgressed-Methode	Wird während einer Dateiübertragung aufgerufen, um den Benutzer über den Fortschritt zu informieren.

Siehe auch

- [ULFileTransfer.DownloadFile-Methode \[UltraLite.NET\] auf Seite 284](#)

FileTransferProgressed-Methode

Wird während einer Dateiübertragung aufgerufen, um den Benutzer über den Fortschritt zu informieren.

Visual Basic-Syntax

```
Public Function FileTransferProgressed(  
    ByVal data As ULFileTransferProgressData  
) As Boolean
```

C#-Syntax

```
public bool FileTransferProgressed(ULFileTransferProgressData data)
```

Parameter

- **data** Ein ULFileTransferProgressData-Objekt, das die neuesten Fortschrittsdaten zur Dateiübertragung enthält.

Rückgabe

Diese Methode muss TRUE zurückgeben, um die Übertragung abubrechen, oder FALSE, um fortzufahren.

Bemerkungen

Diese Methode muss TRUE zurückgeben, um die Übertragung abubrechen, oder FALSE, um fortzufahren.

Während eines FileTransferProgressed-Aufrufs dürfen keine UltraLite.NET API-Methoden aufgerufen werden.

Siehe auch

- [ULFileTransferProgressData-Klasse \[UltraLite.NET\] auf Seite 297](#)

ULIndexSchema-Klasse

UL-Erw.: Stellt das Schema eines UltraLite-Tabellenindexes dar.

Visual Basic-Syntax

```
Public NotInheritable Class ULIndexSchema
```

C#-Syntax

```
public sealed class ULIndexSchema
```

Mitglieder

Alle Mitglieder der ULIndexSchema-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
GetColumnName-Methode	Gibt den Namen der <i>colOrdinalInIndex</i> -ten Spalte in diesem Index zurück.
IsColumnDescending-Methode	Prüft, ob die benannte Spalte von diesem Index in absteigender Reihenfolge verwendet wird.
ColumnCount-Eigenschaft	Gibt die Anzahl der Spalten des Indexes zurück.
IsForeignKey-Eigenschaft	Prüft, ob der Index ein Fremdschlüssel ist.
IsForeignKeyCheckOnCommit-Eigenschaft	Prüft, ob die referenzielle Integrität für den Fremdschlüssel bei Festzeichnungen oder bei Einfügungen und Aktualisierungen überprüft wird.
IsForeignKeyNullable-Eigenschaft	Prüft, ob der Fremdschlüssel nullwertfähig ist.
IsOpen-Eigenschaft	Ermittelt, ob das Indexschema geöffnet oder geschlossen ist.
IsPrimaryKey-Eigenschaft	Prüft, ob der Index der Primärschlüssel ist.

Name	Beschreibung
IsUniqueIndex-Eigenschaft	Prüft, ob der Index eindeutig ist.
IsUniqueKey-Eigenschaft	Prüft, ob der Index ein eindeutiger Schlüssel ist.
Name-Eigenschaft	Gibt den Namen des Indexes zurück.
ReferencedIndexName-Eigenschaft	Der Name des referenzierten Primärindexes, wenn der Index ein Fremdschlüssel ist
ReferencedTableName-Eigenschaft	Der Name der referenzierten Primärtabelle, wenn der Index ein Fremdschlüssel ist.

Bemerkungen

Es gibt keinen Konstruktor für diese Klasse. Indexschemata werden unter Verwendung der Methoden `ULTableSchema.PrimaryKey`, `ULTableSchema.GetIndex(string)` und `ULTableSchema.GetOptimalIndex(int)` erstellt.

Siehe auch

- [ULTableSchema.PrimaryKey-Eigenschaft \[UltraLite.NET\] auf Seite 449](#)
- [ULTableSchema.GetIndex-Methode \[UltraLite.NET\] auf Seite 441](#)
- [ULTableSchema.GetOptimalIndex-Methode \[UltraLite.NET\] auf Seite 442](#)
- [ULTableSchema-Klasse \[UltraLite.NET\] auf Seite 437](#)

GetColumnName-Methode

Gibt den Namen der *colOrdinalInIndex*-ten Spalte in diesem Index zurück.

Visual Basic-Syntax

```
Public Function GetColumnName(  
    ByVal colOrdinalInIndex As Short  
) As String
```

C#-Syntax

```
public string GetColumnName(short colOrdinalInIndex)
```

Parameter

- **colOrdinalInIndex** Die Ordinalzahl der gewünschten Spalte im Index. Der Wert muss im Bereich [1,ColumnCount] liegen.

Rückgabe

Der Name der Spalte.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Ordinalzahlen und Anzahl der Spalten können sich während eines Schema-Upgrades ändern. Spalten-Ordinalzahlen aus einem Index unterscheiden sich von den Spalten-IDs in einem Tabellenindex oder einem anderen Index, selbst wenn sie sich auf dieselbe physische Spalte in einer bestimmten Tabelle beziehen.

Siehe auch

- [ULIndexSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 303](#)

IsColumnDescending-Methode

Prüft, ob die benannte Spalte von diesem Index in absteigender Reihenfolge verwendet wird.

Visual Basic-Syntax

```
Public Function IsColumnDescending(ByVal name As String) As Boolean
```

C#-Syntax

```
public bool IsColumnDescending(string name)
```

Parameter

- **name** Der Name der Spalte.

Rückgabe

TRUE, wenn die Spalte in absteigender Reihenfolge verwendet wird. FALSE, wenn sie in aufsteigender Reihenfolge verwendet wird

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULIndexSchema.GetColumnName-Methode \[UltraLite.NET\] auf Seite 302](#)
- [ULIndexSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 303](#)

ColumnCount-Eigenschaft

Gibt die Anzahl der Spalten des Indexes zurück.

Visual Basic-Syntax

```
Public ReadOnly Property ColumnCount As Short
```

C#-Syntax

```
public short ColumnCount {get;}
```

Bemerkungen

Die Anzahl der Spalten im Index

Spalten-Ordinalzahlen in Indizes reichen von 1 bis zum ColumnCount-Wert einschließlich.

Ordinalzahlen und Anzahl der Spalten können sich während eines Schema-Upgrades ändern. Spalten-Ordinalzahlen aus einem Index unterscheiden sich von den Spalten-IDs in einem Tabellenindex oder einem anderen Index, selbst wenn sie sich auf dieselbe physische Spalte in einer bestimmten Tabelle beziehen.

IsForeignKey-Eigenschaft

Prüft, ob der Index ein Fremdschlüssel ist.

Visual Basic-Syntax

```
Public ReadOnly Property IsForeignKey As Boolean
```

C#-Syntax

```
public bool IsForeignKey {get;}
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

TRUE, wenn der Index der Fremdschlüssel ist, andernfalls FALSE.

Spalten in einem Fremdschlüssel können einen eindeutigen Nicht-Null-Index einer anderen Tabelle referenzieren.

IsForeignKeyCheckOnCommit-Eigenschaft

Prüft, ob die referenzielle Integrität für den Fremdschlüssel bei Festschreibungen oder bei Einfügungen und Aktualisierungen überprüft wird.

Visual Basic-Syntax

```
Public ReadOnly Property IsForeignKeyCheckOnCommit As Boolean
```

C#-Syntax

```
public bool IsForeignKeyCheckOnCommit {get;}
```

Ausnahmen

- **ULException-Klasse** Ein SQL-Fehler ist aufgetreten (einschließender Index ist kein Fremdschlüssel).

Bemerkungen

TRUE, wenn die referenzielle Integrität bei Commits geprüft wird; FALSE, wenn sie bei Einfügungen und Aktualisierungen geprüft wird.

Siehe auch

- [ULIndexSchema.IsForeignKey-Eigenschaft \[UltraLite.NET\] auf Seite 304](#)

IsForeignKeyNullable-Eigenschaft

Prüft, ob der Fremdschlüssel nullwertfähig ist.

Visual Basic-Syntax

```
Public ReadOnly Property IsForeignKeyNullable As Boolean
```

C#-Syntax

```
public bool IsForeignKeyNullable {get;}
```

Ausnahmen

- **ULException-Klasse** Ein SQL-Fehler ist aufgetreten (einschließender Index ist kein Fremdschlüssel).

Bemerkungen

TRUE, wenn der Fremdschlüssel nullwertfähig ist, sonst FALSE

Siehe auch

- [ULIndexSchema.IsForeignKey-Eigenschaft \[UltraLite.NET\] auf Seite 304](#)

IsOpen-Eigenschaft

Ermittelt, ob das Indexschema geöffnet oder geschlossen ist.

Visual Basic-Syntax

```
Public ReadOnly Property IsOpen As Boolean
```

C#-Syntax

```
public bool IsOpen {get;}
```

Bemerkungen

TRUE, wenn das Indexschema geöffnet ist, andernfalls FALSE.

IsPrimaryKey-Eigenschaft

Prüft, ob der Index der Primärschlüssel ist.

Visual Basic-Syntax

```
Public ReadOnly Property IsPrimaryKey As Boolean
```

C#-Syntax

```
public bool IsPrimaryKey {get;}
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

TRUE, wenn der Index der Primärschlüssel ist, andernfalls FALSE.

Spalten im Primärschlüssel dürfen nicht NULL sein.

IsUniqueIndex-Eigenschaft

Prüft, ob der Index eindeutig ist.

Visual Basic-Syntax

```
Public ReadOnly Property IsUniqueIndex As Boolean
```

C#-Syntax

```
public bool IsUniqueIndex {get;}
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

TRUE, wenn der Index eindeutig ist, andernfalls FALSE.

Spalten in einem eindeutigen Index können Null sein.

IsUniqueKey-Eigenschaft

Prüft, ob der Index ein eindeutiger Schlüssel ist.

Visual Basic-Syntax

```
Public ReadOnly Property IsUniqueKey As Boolean
```

C#-Syntax

```
public bool IsUniqueKey {get;}
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

TRUE, wenn der Index ein eindeutiger Schlüssel ist, andernfalls FALSE.

Spalten in einem eindeutigen Schlüssel dürfen nicht NULL sein.

Name-Eigenschaft

Gibt den Namen des Indexes zurück.

Visual Basic-Syntax

```
Public ReadOnly Property Name As String
```

C#-Syntax

```
public string Name {get;}
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Eine Zeichenfolge, die den Namen des Indexes angibt.

ReferencedIndexName-Eigenschaft

Der Name des referenzierten Primärexes, wenn der Index ein Fremdschlüssel ist

Visual Basic-Syntax

```
Public ReadOnly Property ReferencedIndexName As String
```

C#-Syntax

```
public string ReferencedIndexName {get;}
```

Ausnahmen

- **ULException-Klasse** Ein SQL-Fehler ist aufgetreten (einschließender Index ist kein Fremdschlüssel).

Bemerkungen

Eine Zeichenfolge, die den Namen des referenzierten Primärexes angibt

Siehe auch

- [ULIndexSchema.IsForeignKey-Eigenschaft \[UltraLite.NET\] auf Seite 304](#)

ReferencedTableName-Eigenschaft

Der Name der referenzierten Primärtabelle, wenn der Index ein Fremdschlüssel ist.

Visual Basic-Syntax

```
Public ReadOnly Property ReferencedTableName As String
```

C#-Syntax

```
public string ReferencedTableName {get;}
```

Ausnahmen

- **ULException-Klasse** Ein SQL-Fehler ist aufgetreten (einschließender Index ist kein Fremdschlüssel).

Bemerkungen

Eine Zeichenfolge, die den Namen der referenzierten Primärtabelle angibt

Siehe auch

- [ULIndexSchema.IsForeignKey-Eigenschaft \[UltraLite.NET\] auf Seite 304](#)

ULInfoMessageEventArgs-Klasse

Stellt Daten für das ULConnection.InfoMessage-Ereignis bereit.

Visual Basic-Syntax

```
Public NotInheritable Class ULInfoMessageEventArgs  
    Inherits System.EventArgs
```

C#-Syntax

```
public sealed class ULInfoMessageEventArgs : System.EventArgs
```

Basisklassen

- [System.EventArgs](#)

Mitglieder

Alle Mitglieder der ULInfoMessageEventArgs-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ToString-Methode	Die Zeichenfolge des ULConnection.InfoMessage-Ereignisses.
Message-Eigenschaft	Die von der Datenbank zurückgegebene Zeichenfolge der Informationsnachricht oder Warnung.

Name	Beschreibung
NativeError-Eigenschaft	Der SQLCODE, der der Informationsnachricht oder Warnung entspricht, die von der Datenbank zurückgegeben wurden.
Source-Eigenschaft	Der Name des ADO.NET-Datenproviders, der die Nachricht zurückgibt.
Empty-Feld (geerbt aus System.EventArgs)	Repräsentiert ein Ereignis ohne Ereignisdaten.

Siehe auch

- [ULConnection.InfoMessage-Ereignis \[UltraLite.NET\] auf Seite 166](#)

ToString-Methode

Die Zeichenfolge des ULConnection.InfoMessage-Ereignisses.

Visual Basic-Syntax

```
Public Overrides Function ToString() As String
```

C#-Syntax

```
public override string ToString()
```

Rückgabe

Die Zeichenfolge der Informationsnachricht oder Warnung.

Bemerkungen

Die Zeichenfolge des ULConnection.InfoMessage-Ereignisses.

Siehe auch

- [ULConnection.InfoMessage-Ereignis \[UltraLite.NET\] auf Seite 166](#)

Message-Eigenschaft

Die von der Datenbank zurückgegebene Zeichenfolge der Informationsnachricht oder Warnung.

Visual Basic-Syntax

```
Public ReadOnly Property Message As String
```

C#-Syntax

```
public string Message {get;}
```

Bemerkungen

Eine Zeichenfolge mit der Informationsnachricht oder Warnung

NativeError-Eigenschaft

Der SQLCODE, der der Informationsnachricht oder Warnung entspricht, die von der Datenbank zurückgegeben wurden.

Visual Basic-Syntax

```
Public ReadOnly Property NativeError As ULSQLCode
```

C#-Syntax

```
public ULSQLCode NativeError {get;}
```

Bemerkungen

Ein ULSQLCode-Wert einer Informationsnachricht oder Warnung

Source-Eigenschaft

Der Name des ADO.NET-Datenproviders, der die Nachricht zurückgibt.

Visual Basic-Syntax

```
Public ReadOnly Property Source As String
```

C#-Syntax

```
public string Source {get;}
```

Bemerkungen

Die Zeichenfolge "UltraLite.NET Data Provider".

ULMetaDataCollectionNames-Klasse

Stellt eine Liste von Konstanten für die ULConnection.GetSchema(String,String[])-Methode bereit, um Metadaten-Sammlungen abzurufen.

Visual Basic-Syntax

```
Public NotInheritable Class ULMetaDataCollectionNames
```

C#-Syntax

```
public sealed class ULMetaDataCollectionNames
```

Mitglieder

Alle Mitglieder der ULMetaDataCollectionNames-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
Columns-Eigenschaft	Stellt eine Konstante für die ULConnection.GetSchema(String)-Methode bereit, die die Columns-Sammlung repräsentiert.
DataSourceInformation-Eigenschaft	Stellt eine Konstante für die Verwendung mit der ULConnection.GetSchema(String)-Methode bereit, die die DataSourceInformation-Sammlung repräsentiert.
DataTypes-Eigenschaft	Stellt eine Konstante für die ULConnection.GetSchema(String)-Methode bereit, die die DataTypes-Sammlung repräsentiert.
ForeignKeys-Eigenschaft	Stellt eine Konstante für die ULConnection.GetSchema(String)-Methode bereit, die die ForeignKeys-Sammlung repräsentiert.
IndexColumns-Eigenschaft	Stellt eine Konstante für die ULConnection.GetSchema(String)-Methode bereit, die die IndexColumns-Sammlung repräsentiert.
Indexes-Eigenschaft	Stellt eine Konstante für die ULConnection.GetSchema(String)-Methode bereit, die die Indexes-Sammlung repräsentiert.
MetaDataCollections-Eigenschaft	Stellt eine Konstante für die ULConnection.GetSchema(String)-Methode bereit, die die MetaDataCollections-Sammlung repräsentiert.
Publications-Eigenschaft	Stellt eine Konstante für die ULConnection.GetSchema(String)-Methode bereit, die die Publications-Sammlung repräsentiert.
ReservedWords-Eigenschaft	Stellt eine Konstante mit der ULConnection.GetSchema(String)-Methode bereit, die die ReservedWords-Sammlung repräsentiert.
Restrictions-Eigenschaft	Stellt eine Konstante für die ULConnection.GetSchema(String)-Methode bereit, die die Restrictions-Sammlung repräsentiert.
Tables-Eigenschaft	Stellt eine Konstante für die ULConnection.GetSchema(String)-Methode bereit, die die Tables-Sammlung repräsentiert.

Columns-Eigenschaft

Stellt eine Konstante für die ULConnection.GetSchema(String)-Methode bereit, die die Columns-Sammlung repräsentiert.

Visual Basic-Syntax

```
Public Shared ReadOnly Property Columns As String
```

C#-Syntax

```
public string Columns {get;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen der Columns-Sammlung repräsentiert.

Siehe auch

- [ULConnection.GetSchema-Methode \[UltraLite.NET\] auf Seite 147](#)

Beispiel

Der folgende Code füllt ein DataTable-Objekt mit der Columns-Sammlung.

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.Columns )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.Columns );
```

DataSourceInformation-Eigenschaft

Stellt eine Konstante für die Verwendung mit der ULConnection.GetSchema(String)-Methode bereit, die die DataSourceInformation-Sammlung repräsentiert.

Visual Basic-Syntax

```
Public Shared ReadOnly Property DataSourceInformation As String
```

C#-Syntax

```
public string DataSourceInformation {get;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen der DataSourceInformation-Sammlung repräsentiert.

Siehe auch

- [ULConnection.GetSchema-Methode \[UltraLite.NET\] auf Seite 147](#)

Beispiel

Der folgende Code füllt ein DataTable-Objekt mit der DataSourceInformation-Sammlung.

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.DataSourceInformation )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.DataSourceInformation );
```

DataTypes-Eigenschaft

Stellt eine Konstante für die `ULConnection.GetSchema(String)`-Methode bereit, die die `DataTypes`-Sammlung repräsentiert.

Visual Basic-Syntax

```
Public Shared ReadOnly Property DataTypes As String
```

C#-Syntax

```
public string DataTypes {get;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen der `DataTypes`-Sammlung repräsentiert.

Siehe auch

- [ULConnection.GetSchema-Methode \[UltraLite.NET\] auf Seite 147](#)

Beispiel

Der folgende Code füllt ein `DataTable`-Objekt mit der `DataTypes`-Sammlung.

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.DataTypes )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.DataTypes );
```

ForeignKeys-Eigenschaft

Stellt eine Konstante für die `ULConnection.GetSchema(String)`-Methode bereit, die die `ForeignKeys`-Sammlung repräsentiert.

Visual Basic-Syntax

```
Public Shared ReadOnly Property ForeignKeys As String
```

C#-Syntax

```
public string ForeignKeys {get;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen der `ForeignKeys`-Sammlung repräsentiert.

Siehe auch

- [ULConnection.GetSchema-Methode \[UltraLite.NET\] auf Seite 147](#)

Beispiel

Der folgende Code füllt ein DataTable-Objekt mit der ForeignKeys-Sammlung.

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.ForeignKeys )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.ForeignKeys );
```

IndexColumns-Eigenschaft

Stellt eine Konstante für die ULConnection.GetSchema(String)-Methode bereit, die die IndexColumns-Sammlung repräsentiert.

Visual Basic-Syntax

```
Public Shared ReadOnly Property IndexColumns As String
```

C#-Syntax

```
public string IndexColumns {get;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen der IndexColumns-Sammlung repräsentiert.

Siehe auch

- [ULConnection.GetSchema-Methode \[UltraLite.NET\] auf Seite 147](#)

Beispiel

Der folgende Code füllt ein DataTable-Objekt mit der IndexColumns-Sammlung.

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.IndexColumns )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.IndexColumns );
```

Indexes-Eigenschaft

Stellt eine Konstante für die ULConnection.GetSchema(String)-Methode bereit, die die Indexes-Sammlung repräsentiert.

Visual Basic-Syntax

```
Public Shared ReadOnly Property Indexes As String
```

C#-Syntax

```
public string Indexes {get;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen der Indexes-Sammlung repräsentiert.

Siehe auch

- [ULConnection.GetSchema-Methode \[UltraLite.NET\] auf Seite 147](#)

Beispiel

Der folgende Code füllt ein DataTable-Objekt mit der Indexes-Sammlung.

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.Indexes )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.Indexes );
```

MetaDataCollections-Eigenschaft

Stellt eine Konstante für die ULConnection.GetSchema(String)-Methode bereit, die die MetaDataCollections-Sammlung repräsentiert.

Visual Basic-Syntax

```
Public Shared ReadOnly Property MetaDataCollections As String
```

C#-Syntax

```
public string MetaDataCollections {get;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen der MetaDataCollections-Sammlung repräsentiert.

Siehe auch

- [ULConnection.GetSchema-Methode \[UltraLite.NET\] auf Seite 147](#)

Beispiel

Der folgende Code füllt ein DataTable-Objekt mit der MetaDataCollections-Sammlung.

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.MetaDataCollections )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.MetadataCollections );
```

Publications-Eigenschaft

Stellt eine Konstante für die `ULConnection.GetSchema(String)`-Methode bereit, die die Publications-Sammlung repräsentiert.

Visual Basic-Syntax

```
Public Shared ReadOnly Property Publications As String
```

C#-Syntax

```
public string Publications {get;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen der Publications-Sammlung repräsentiert.

Siehe auch

- [ULConnection.GetSchema-Methode \[UltraLite.NET\] auf Seite 147](#)

Beispiel

Der folgende Code füllt ein `DataTable`-Objekt mit der Publications-Sammlung.

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.Publications )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.Publications );
```

ReservedWords-Eigenschaft

Stellt eine Konstante mit der `ULConnection.GetSchema(String)`-Methode bereit, die die ReservedWords-Sammlung repräsentiert.

Visual Basic-Syntax

```
Public Shared ReadOnly Property ReservedWords As String
```

C#-Syntax

```
public string ReservedWords {get;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen der ReservedWords-Sammlung repräsentiert.

Siehe auch

- [ULConnection.GetSchema-Methode \[UltraLite.NET\] auf Seite 147](#)

Beispiel

Der folgende Code füllt ein DataTable-Objekt mit der ReservedWords-Sammlung.

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.ReservedWords )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.ReservedWords );
```

Restrictions-Eigenschaft

Stellt eine Konstante für die ULConnection.GetSchema(String)-Methode bereit, die die Restrictions-Sammlung repräsentiert.

Visual Basic-Syntax

```
Public Shared ReadOnly Property Restrictions As String
```

C#-Syntax

```
public string Restrictions {get;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen der Restrictions-Sammlung repräsentiert.

Siehe auch

- [ULConnection.GetSchema-Methode \[UltraLite.NET\] auf Seite 147](#)

Beispiel

Der folgende Code füllt ein DataTable-Objekt mit der Restrictions-Sammlung.

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.Restrictions )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.Restrictions );
```

Tables-Eigenschaft

Stellt eine Konstante für die ULConnection.GetSchema(String)-Methode bereit, die die Tables-Sammlung repräsentiert.

Visual Basic-Syntax

```
Public Shared ReadOnly Property Tables As String
```

C#-Syntax

```
public string Tables {get;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen der Tables-Sammlung repräsentiert.

Siehe auch

- [ULConnection.GetSchema-Methode \[UltraLite.NET\] auf Seite 147](#)

Beispiel

Der folgende Code füllt ein DataTable-Objekt mit der Tables-Sammlung.

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.Tables )

// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.Tables );
```

ULParameter-Klasse

Stellt einen Parameter für ein ULCommand-Objekt dar.

Visual Basic-Syntax

```
Public NotInheritable Class ULParameter
    Inherits System.Data.Common.DbParameter
    Implements System.ICloneable
```

C#-Syntax

```
public sealed class ULParameter :
    System.Data.Common.DbParameter,
    System.ICloneable
```

Basisklassen

- [System.Data.Common.DbParameter](#)
- [System.ICloneable](#)

Mitglieder

Alle Mitglieder der ULParameter-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ULParameter-Konstruktor	Initialisiert ein ULParameter-Objekt mit dem Wert Null ("Nothing" in Visual Basic).
ResetDbType-Methode	Diese Methode wird in UltraLite.NET nicht unterstützt.
ToString-Methode	Gibt die Zeichenfolge zurück, die diese Instanz repräsentiert.
DbType-Eigenschaft	Legt den System.Data.DbType des Parameters fest.
Direction-Eigenschaft	Ein Wert, der anzeigt, ob der Parameter nur für die Eingabe, nur für die Ausgabe, bidirektional oder der Rückgabewertparameter einer gespeicherten Prozedur ist.
IsNullable-Eigenschaft	Gibt an, ob der Parameter NULL akzeptiert.
Offset-Eigenschaft	Legt den Offset des ULParameter.Value-Objekts fest.
ParameterName-Eigenschaft	Gibt den Namen des Parameters an.
Precision-Eigenschaft	Gibt die maximale Anzahl von Stellen an, die verwendet werden, um die ULParameter.Value-Eigenschaft darzustellen.
Scale-Eigenschaft	Gibt die Anzahl der Dezimalstellen an, mit denen die ULParameter.Value-Eigenschaft aufgelöst wird.
Size-Eigenschaft	Gibt die maximale Größe der Daten in der Spalte an.
SourceColumn-Eigenschaft	Gibt den Namen der Quellspalte an, die dem DataSet-Objekt zugeordnet ist und zum Laden oder Zurückgeben des Werts verwendet wird.
SourceColumnNullMapping-Eigenschaft	Gibt an, ob die angegebene Spalte nullwertfähig ist.
SourceVersion-Eigenschaft	Der System.Data.DataRowVersion-Wert, der beim Laden der ULParameter.Value-Eigenschaft verwendet wird.
ULDbType-Eigenschaft	Legt den iAnywhere.Data.UltraLite.ULDbType des Parameters fest.
Value-Eigenschaft	Gibt den Wert des Parameters an.

Bemerkungen

Ein ULParameter-Objekt kann mit einem seiner zahlreichen Konstruktoren oder mit der ULCommand.CreateParameter-Methode erstellt werden. Aufgrund der speziellen Behandlung der 0- und 0.0-Konstanten und der Art und Weise, wie überladene Methoden gelöst werden, wird dringend

empfohlen, Konstantenwerte explizit auf den Objekttyp festzulegen, wenn der ULParameter(string,object)-Konstruktor verwendet wird. Beispiel:

```
' Visual Basic
Dim p As ULParameter = New ULParameter( "", CType( 0, Object ) )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULParameter p = new ULParameter( "", (object)0 );
```

Parameter (einschließlich jener, die durch die ULCommand.CreateParameter-Methode erstellt werden) müssen einer ULCommand.Parameters-Sammlung hinzugefügt werden, um sie verwenden zu können. Alle Parameter werden als positionsbasierte Parameter behandelt und von einem Befehl in der Reihenfolge verwendet, in der sie hinzugefügt wurden.

In UltraLite.NET können Parameter nur als IN-Parameter verwendet werden und alle Zuordnungsinformationen werden ignoriert. Nur die ULParameter.Value-Eigenschaft ist wichtig.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULCommand.CreateParameter-Methode \[UltraLite.NET\] auf Seite 85](#)
- [ULParameter.ULParameter-Konstruktor \[UltraLite.NET\] auf Seite 320](#)
- [ULCommand.Parameters-Eigenschaft \[UltraLite.NET\] auf Seite 107](#)
- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)
- [System.Data.Common.DbParameter](#)
- [System.Data.IDbDataParameter](#)
- [System.Data.IDataParameter](#)

ULParameter-Konstruktor

Initialisiert ein ULParameter-Objekt mit dem Wert Null ("Nothing" in Visual Basic).

Überladungsliste

Name	Beschreibung
ULParameter()-Konstruktor	Initialisiert ein ULParameter-Objekt mit dem Wert Null ("Nothing" in Visual Basic).
ULParameter(string, object)-Konstruktor	Initialisiert ein ULParameter-Objekt mit dem angegebenen Parameternamen und -wert.
ULParameter(string, ULDbType)-Konstruktor	Initialisiert ein ULParameter-Objekt mit dem angegebenen Parameternamen und -datentyp.
ULParameter(string, ULDbType, int)-Konstruktor	Initialisiert ein ULParameter-Objekt mit dem angegebenen Parameternamen und -datentyp.

Name	Beschreibung
ULParameter(string, ULDb-Type, int, ParameterDirection, bool, byte, byte, string, DataRowVersion, object)-Konstruktor	Initialisiert ein ULParameter-Objekt mit dem angegebenen Namen, dem Datentyp, der Länge, der Richtung, der Nullwertfähigkeit, der numerischen Gesamtstellenzahl, der Anzahl der Dezimalstellen, der Quellspalte und dem Wert des Parameters.
ULParameter(string, ULDb-Type, int, string)-Konstruktor	Initialisiert ein ULParameter-Objekt mit dem angegebenen Namen, dem Datentyp und der Länge des Parameters.

ULParameter()-Konstruktor

Initialisiert ein ULParameter-Objekt mit dem Wert Null ("Nothing" in Visual Basic).

Visual Basic-Syntax

```
Public Sub New()
```

C#-Syntax

```
public ULParameter()
```

Siehe auch

- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)
- [ULParameter.ULParameter-Konstruktor \[UltraLite.NET\] auf Seite 320](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)

Beispiel

Der folgende Code erstellt einen ULParameter-Wert von 3 und fügt ihn dem ULCommand-Objekt cmd hinzu.

```
' Visual Basic
Dim p As ULParameter = New ULParameter
p.Value = 3
cmd.Parameters.Add( p )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULParameter p = new ULParameter();
p.Value = 3;
cmd.Parameters.Add( p );
```

ULParameter(string, object)-Konstruktor

Initialisiert ein ULParameter-Objekt mit dem angegebenen Parameternamen und -wert.

Visual Basic-Syntax

```
Public Sub New(ByVal parameterName As String, ByVal value As Object)
```

C#-Syntax

```
public ULParameter(string parameterName, object value)
```

Parameter

- **parameterName** Der Name des Parameters. Verwenden Sie für nicht benannte Parameter eine leere Zeichenfolge ("") oder eine Nullreferenz ("Nothing" in Visual Basic) für diesen Wert. In UltraLite.NET werden Parameternamen nicht vom ULCommand-Objekt verwendet.
- **value** Übergeben Sie eine System.Object-Klasse, um den Wert des Parameters zu erstellen.

Bemerkungen

Aufgrund der speziellen Behandlung der 0- und 0.0-Konstanten und der Weise, wie überladene Methoden gelöst werden, wird dringend empfohlen, explizit Konstantenwerte für Objekttypen festzulegen, wenn dieser Konstruktor verwendet wird.

Siehe auch

- [ULParameter.ULParameter-Konstruktor \[UltraLite.NET\] auf Seite 320](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [System.Object](#)

Beispiel

Der folgende Code erstellt einen ULParameter-Wert mit dem Wert 0 und fügt ihn dem ULCommand-Objekt cmd hinzu.

```
' Visual Basic  
cmd.Parameters.Add( New ULParameter( "", CType( 0, Object ) ) )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#  
cmd.Parameters.Add( new ULParameter( "", (object)0 ) );
```

ULParameter(string, ULDbType)-Konstruktor

Initialisiert ein ULParameter-Objekt mit dem angegebenen Parameternamen und -datentyp.

Visual Basic-Syntax

```
Public Sub New(ByVal parameterName As String, ByVal dbType As ULDbType)
```

C#-Syntax

```
public ULParameter(string parameterName, ULDbType dbType)
```

Parameter

- **parameterName** Der Name des Parameters. Verwenden Sie für nicht benannte Parameter eine leere Zeichenfolge ("") oder eine Nullreferenz ("Nothing" in Visual Basic) für diesen Wert. In UltraLite.NET werden Parameternamen nicht vom ULCommand-Objekt verwendet.

- **dbType** Einer der iAnywhere.Data.UltraLite.ULDbType-Werte.

Bemerkungen

Dieser Konstruktor wird nicht empfohlen. Er wird aus Gründen der Kompatibilität mit anderen Datenprovidern bereitgestellt.

In UltraLite.NET können Parameter nur als IN-Parameter verwendet werden und alle Zuordnungsinformationen werden ignoriert. Nur die `ULParameter.Value`-Eigenschaft ist wichtig.

Siehe auch

- [ULParameter.ULParameter-Konstruktor \[UltraLite.NET\] auf Seite 320](#)
- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)

ULParameter(string, ULDbType, int)-Konstruktor

Initialisiert ein `ULParameter`-Objekt mit dem angegebenen Parameternamen und -datentyp.

Visual Basic-Syntax

```
Public Sub New(  
    ByVal parameterName As String,  
    ByVal dbType As ULDbType,  
    ByVal size As Integer  
)
```

C#-Syntax

```
public ULParameter(string parameterName, ULDbType dbType, int size)
```

Parameter

- **parameterName** Der Name des Parameters. Verwenden Sie für nicht benannte Parameter eine leere Zeichenfolge (""), oder eine Nullreferenz ("Nothing" in Visual Basic) für diesen Wert. In UltraLite.NET werden Parameternamen nicht vom `ULCommand`-Objekt verwendet.
- **dbType** Einer der iAnywhere.Data.UltraLite.ULDbType-Werte.
- **size** Die Länge des Parameters.

Bemerkungen

Dieser Konstruktor wird nicht empfohlen. Er wird aus Gründen der Kompatibilität mit anderen Datenprovidern bereitgestellt.

In UltraLite.NET können Parameter nur als IN-Parameter verwendet werden und alle Zuordnungsinformationen werden ignoriert. Nur die `ULParameter.Value`-Eigenschaft ist wichtig.

Siehe auch

- [ULParameter.ULParameter-Konstruktor \[UltraLite.NET\] auf Seite 320](#)
- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)

ULParameter(string, ULDbType, int, ParameterDirection, bool, byte, byte, string, DataRowVersion, object)-Konstruktor

Initialisiert ein ULParameter-Objekt mit dem angegebenen Namen, dem Datentyp, der Länge, der Richtung, der Nullwertfähigkeit, der numerischen Gesamtstellenzahl, der Anzahl der Dezimalstellen, der Quellspalte und dem Wert des Parameters.

Visual Basic-Syntax

```
Public Sub New(  
    ByVal parameterName As String,  
    ByVal dbType As ULDbType,  
    ByVal size As Integer,  
    ByVal direction As ParameterDirection,  
    ByVal isNullable As Boolean,  
    ByVal precision As Byte,  
    ByVal scale As Byte,  
    ByVal sourceColumn As String,  
    ByVal sourceVersion As DataRowVersion,  
    ByVal value As Object  
)
```

C#-Syntax

```
public ULParameter(  
    string parameterName,  
    ULDbType dbType,  
    int size,  
    ParameterDirection direction,  
    bool isNullable,  
    byte precision,  
    byte scale,  
    string sourceColumn,  
    DataRowVersion sourceVersion,  
    object value  
)
```

Parameter

- **parameterName** Der Name des Parameters. Verwenden Sie für nicht benannte Parameter eine leere Zeichenfolge (""), oder eine Nullreferenz ("Nothing" in Visual Basic) für diesen Wert. In UltraLite.NET werden Parameternamen nicht vom ULCommand-Objekt verwendet.
- **dbType** Einer der iAnywhere.Data.UltraLite.ULDbType-Werte.
- **size** Die Länge des Parameters.
- **direction** Einer der System.Data.ParameterDirection-Werte.
- **isNullable** TRUE, wenn der Wert des Feldes NULL enthalten darf, sonst FALSE
- **precision** Die Gesamtanzahl der Stellen links und rechts vom Dezimalzeichen, in die die Value-Eigenschaft aufgelöst wird.
- **scale** Die Gesamtanzahl der Dezimalstellen, in die die Value-Eigenschaft aufgelöst wird.

- **sourceColumn** Der Name der Quellspalte, die zuzuordnen ist.
- **sourceVersion** Einer der System.Data.DataRowVersion-Werte
- **value** Übergeben Sie ein System.Object, um den Wert des Parameters zu erstellen.

Ausnahmen

- **ULException-Klasse** Nur die System.Data.ParameterDirection.Input-Richtung wird in UltraLite.NET unterstützt.

Bemerkungen

Dieser Konstruktor wird nicht empfohlen. Er wird aus Gründen der Kompatibilität mit anderen Datenprovidern bereitgestellt.

Siehe auch

- [ULParameter.ULParameter-Konstruktor \[UltraLite.NET\] auf Seite 320](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [System.Data.DataRowVersion](#)
- [System.Object](#)
- [System.Data.ParameterDirection](#)

ULParameter(string, ULDbType, int, string)-Konstruktor

Initialisiert ein ULParameter-Objekt mit dem angegebenen Namen, dem Datentyp und der Länge des Parameters.

Visual Basic-Syntax

```
Public Sub New(  
    ByVal parameterName As String,  
    ByVal dbType As ULDbType,  
    ByVal size As Integer,  
    ByVal sourceColumn As String  
)
```

C#-Syntax

```
public ULParameter(  
    string parameterName,  
    ULDbType dbType,  
    int size,  
    string sourceColumn  
)
```

Parameter

- **parameterName** Der Name des Parameters. Verwenden Sie für nicht benannte Parameter eine leere Zeichenfolge ("") oder eine Nullreferenz ("Nothing" in Visual Basic) für diesen Wert. In UltraLite.NET werden Parameternamen nicht vom ULCommand-Objekt verwendet.
- **dbType** Einer der iAnywhere.Data.UltraLite.ULDbType-Werte.

- **size** Die Länge des Parameters.
- **sourceColumn** Der Name der Quellspalte, die zuzuordnen ist.

Bemerkungen

Dieser Konstruktor wird nicht empfohlen. Er wird aus Gründen der Kompatibilität mit anderen Datenprovidern bereitgestellt.

In UltraLite.NET können Parameter nur als IN-Parameter verwendet werden und alle Zuordnungsinformationen werden ignoriert. Nur die `ULParameter.Value`-Eigenschaft ist wichtig.

Siehe auch

- [ULParameter-ULParameter-Konstruktor \[UltraLite.NET\] auf Seite 320](#)
- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)

ResetDbType-Methode

Diese Methode wird in UltraLite.NET nicht unterstützt.

Visual Basic-Syntax

```
Public Overrides Sub ResetDbType()
```

C#-Syntax

```
public override void ResetDbType()
```

Bemerkungen

In UltraLite.NET können Parameter nur als IN-Parameter verwendet werden und alle Zuordnungsinformationen werden ignoriert. Nur die `ULParameter.Value`-Eigenschaft ist wichtig.

Siehe auch

- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)

ToString-Methode

Gibt die Zeichenfolge zurück, die diese Instanz repräsentiert.

Visual Basic-Syntax

```
Public Overrides Function ToString() As String
```

C#-Syntax

```
public override string ToString()
```

Rückgabe

Der Name des Parameters.

DbType-Eigenschaft

Legt den System.Data.DbType des Parameters fest.

Visual Basic-Syntax

```
Public Overrides Property DbType As DbType
```

C#-Syntax

```
public override DbType DbType {get;set;}
```

Ausnahmen

- **ArgumentException** Es gibt keine Zuordnung des angegebenen Werts zu einem iAnywhere.Data.UltraLite.ULDbType-Wert. Deshalb wird der angegebene Wert nicht unterstützt.

Bemerkungen

Einer der System.Data.DbType-Werte.

Die Eigenschaften ULParameter.ULDbType und DbType sind verknüpft. Durch Festlegen von der DbType-Eigenschaft wird die ULParameter.ULDbType-Eigenschaft auf einen unterstützenden iAnywhere.Data.UltraLite.ULDbType-Wert geändert.

Siehe auch

- [ULParameter.ULDbType-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)
- [System.Data.DbType](#)

Direction-Eigenschaft

Ein Wert, der anzeigt, ob der Parameter nur für die Eingabe, nur für die Ausgabe, bidirektional oder der Rückgabewertparameter einer gespeicherten Prozedur ist.

Visual Basic-Syntax

```
Public Overrides Property Direction As ParameterDirection
```

C#-Syntax

```
public override ParameterDirection Direction {get;set;}
```

Ausnahmen

- **ULException-Klasse** Nur die System.Data.ParameterDirection.Input-Richtung wird in UltraLite.NET unterstützt.

Bemerkungen

Einer der System.Data.ParameterDirection-Werte.

In UltraLite.NET können Parameter nur als IN-Parameter verwendet werden und alle Zuordnungsinformationen werden ignoriert. Nur die ULParameter.Value-Eigenschaft ist wichtig.

Siehe auch

- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)
- [System.Data.ParameterDirection](#)

IsNullable-Eigenschaft

Gibt an, ob der Parameter NULL akzeptiert.

Visual Basic-Syntax

```
Public Overrides Property IsNullable As Boolean
```

C#-Syntax

```
public override bool IsNullable {get;set;}
```

Bemerkungen

TRUE, wenn NULL akzeptiert wird, andernfalls FALSE. Der Standardwert ist FALSE. NULL wird unter Verwendung der DBNull-Klasse verarbeitet.

In UltraLite.NET können Parameter nur als IN-Parameter verwendet werden und alle Zuordnungsinformationen werden ignoriert. Nur die `ULParameter.Value`-Eigenschaft ist wichtig.

Siehe auch

- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)

Offset-Eigenschaft

Legt den Offset des `ULParameter.Value`-Objekts fest.

Visual Basic-Syntax

```
Public Property Offset As Integer
```

C#-Syntax

```
public int Offset {get;set;}
```

Bemerkungen

Der Offset des Werts. Standardwert ist "0".

In UltraLite.NET können Parameter nur als IN-Parameter verwendet werden und alle Zuordnungsinformationen werden ignoriert. Nur die `ULParameter.Value`-Eigenschaft ist wichtig.

Siehe auch

- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)

ParameterName-Eigenschaft

Gibt den Namen des Parameters an.

Visual Basic-Syntax

```
Public Overrides Property ParameterName As String
```

C#-Syntax

```
public override string ParameterName {get;set;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen des Parameters repräsentiert, oder eine leere Zeichenfolge (""), für unbenannte Parameter. Die Festlegung einer Nullreferenz ("Nothing" in Visual Basic) bewirkt, dass eine leere Zeichenfolge verwendet wird.

In UltraLite.NET werden Parameternamen nicht von einem ULCommand-Objekt verwendet. Alle Parameter werden als positionsbasierte Parameter behandelt und von einem Befehl in der Reihenfolge verwendet, in der sie hinzugefügt wurden.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)

Precision-Eigenschaft

Gibt die maximale Anzahl von Stellen an, die verwendet werden, um die ULParameter.Value-Eigenschaft darzustellen.

Visual Basic-Syntax

```
Public Property Precision As Byte
```

C#-Syntax

```
public byte Precision {get;set;}
```

Ausnahmen

- **ArgumentException** Der Wert ist größer als 38.

Bemerkungen

Die maximale Anzahl von Stellen, die verwendet werden, um die ULParameter.Value-Eigenschaft darzustellen. Der Standardwert ist 0, was bedeutet, dass der Datenprovider die Gesamtstellenzahl für die ULParameter.Value-Eigenschaft definiert.

In UltraLite.NET können Parameter nur als IN-Parameter verwendet werden und alle Zuordnungsinformationen werden ignoriert. Nur die ULParameter.Value-Eigenschaft ist wichtig.

Siehe auch

- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)

Scale-Eigenschaft

Gibt die Anzahl der Dezimalstellen an, mit denen die `ULParameter.Value`-Eigenschaft aufgelöst wird.

Visual Basic-Syntax

```
Public Property Scale As Byte
```

C#-Syntax

```
public byte scale {get;set;}
```

Bemerkungen

Die Anzahl der Dezimalstellen, mit denen die `ULParameter.Value`-Eigenschaft aufgelöst wird. Standardwert ist "0".

In UltraLite.NET können Parameter nur als IN-Parameter verwendet werden und alle Zuordnungsinformationen werden ignoriert. Nur die `ULParameter.Value`-Eigenschaft ist wichtig.

Siehe auch

- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)

Size-Eigenschaft

Gibt die maximale Größe der Daten in der Spalte an.

Visual Basic-Syntax

```
Public Overrides Property Size As Integer
```

C#-Syntax

```
public override int size {get;set;}
```

Bemerkungen

Die maximale Größe der Daten in der Spalte. Der Standardwert wird aus dem Parameterwert abgerufen. Die `Size`-Eigenschaft wird für die Datentypen "binary" und "string" verwendet.

In UltraLite.NET können Parameter nur als IN-Parameter verwendet werden und alle Zuordnungsinformationen werden ignoriert. Nur die `ULParameter.Value`-Eigenschaft ist wichtig.

Siehe auch

- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)

SourceColumn-Eigenschaft

Gibt den Namen der Quellspalte an, die dem `DataSet`-Objekt zugeordnet ist und zum Laden oder Zurückgeben des Werts verwendet wird.

Visual Basic-Syntax

```
Public Overrides Property SourceColumn As String
```

C#-Syntax

```
public override string SourceColumn {get;set;}
```

Bemerkungen

Eine Zeichenfolge, die den Namen der Quellspalte angibt, die dem DataSet-Objekt zugeordnet ist und zum Laden oder Zurückgeben des Werts verwendet wird.

In UltraLite.NET können Parameter nur als IN-Parameter verwendet werden und alle Zuordnungsinformationen werden ignoriert. Nur die ULParameter.Value-Eigenschaft ist wichtig.

Siehe auch

- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)

SourceColumnNullMapping-Eigenschaft

Gibt an, ob die angegebene Spalte nullwertfähig ist.

Visual Basic-Syntax

```
Public Overrides Property SourceColumnNullMapping As Boolean
```

C#-Syntax

```
public override bool SourceColumnNullMapping {get;set;}
```

Bemerkungen

TRUE, wenn die Quellspalte nullwertfähig ist, sonst FALSE.

In UltraLite.NET können Parameter nur als IN-Parameter verwendet werden und alle Zuordnungsinformationen werden ignoriert. Nur die ULParameter.Value-Eigenschaft ist wichtig.

Siehe auch

- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)

SourceVersion-Eigenschaft

Der System.Data.DataRowVersion-Wert, der beim Laden der ULParameter.Value-Eigenschaft verwendet wird.

Visual Basic-Syntax

```
Public Overrides Property SourceVersion As DataRowVersion
```

C#-Syntax

```
public override DataRowVersion SourceVersion {get;set;}
```

Siehe auch

- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)
- [System.Data.DataRowVersion](#)

ULDbType-Eigenschaft

Legt den iAnywhere.Data.UltraLite.ULDbType des Parameters fest.

Visual Basic-Syntax

```
Public Property ULDbType As ULDbType
```

C#-Syntax

```
public ULDbType ULDbType {get;set;}
```

Bemerkungen

Einer der iAnywhere.Data.UltraLite.ULDbType-Werte.

Die ULDbType-Eigenschaft und die ULParameter.DbType-Eigenschaft sind verknüpft. Durch Festlegen der ULDbType-Eigenschaft wird die ULParameter.DbType-Eigenschaft auf einen unterstützenden System.Data.DbType-Wert geändert.

In UltraLite.NET können Parameter nur als IN-Parameter verwendet werden und alle Zuordnungsinformationen werden ignoriert. Nur die ULParameter.Value-Eigenschaft ist wichtig.

Siehe auch

- [ULParameter.Value-Eigenschaft \[UltraLite.NET\] auf Seite 332](#)
- [ULParameter.DbType-Eigenschaft \[UltraLite.NET\] auf Seite 327](#)
- [System.Data.DbType](#)

Value-Eigenschaft

Gibt den Wert des Parameters an.

Visual Basic-Syntax

```
Public Overrides Property Value As Object
```

C#-Syntax

```
public override object Value {get;set;}
```

Bemerkungen

Übergeben Sie eine System.Object-Klasse, die den Wert des Parameters angibt

Der Wert wird unverändert, d.h. ohne Typkonvertierung oder Zuordnung, an den Datenprovider gesendet. Während der Ausführung versucht der Befehl, den Wert in den erforderlichen Typ zu konvertieren. Wenn

dies fehlschlägt, wird ein `ULException`-Objekt mit `ULSQLCode.SQLE_CONVERSION_ERROR` gemeldet.

Siehe auch

- [ULException-Klasse \[UltraLite.NET\] auf Seite 275](#)
- [System.Object](#)

ULParameterCollection-Klasse

Stellt alle Parameter für ein `ULCommand`-Objekt dar.

Visual Basic-Syntax

```
Public NotInheritable Class ULParameterCollection  
    Inherits System.Data.Common.DbParameterCollection
```

C#-Syntax

```
public sealed class ULParameterCollection :  
    System.Data.Common.DbParameterCollection
```

Basisklassen

- [System.Data.Common.DbParameterCollection](#)

Mitglieder

Alle Mitglieder der `ULParameterCollection`-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
Add-Methode	Fügt ein <code>ULParameter</code> -Objekt zur Sammlung hinzu.
AddRange-Methode	Fügt ein Array von Werten an das Ende der <code>ULParameterCollection</code> -Sammlung an.
Clear-Methode	Entfernt alle Parameter aus der Sammlung.
Contains-Methode	Prüft, ob ein <code>ULParameter</code> -Objekt in der Sammlung vorhanden ist.
CopyTo-Methode	Kopiert <code>ULParameter</code> -Objekte aus dem <code>ULParameterCollection</code> -Objekt in das angegebene Array.
GetEnumerator-Methode	Gibt einen Enumerator für die Sammlung zurück.
GetParameter-Methode (geerbt aus <code>System.Data.Common.DbParameterCollection</code>)	Gibt das System.Data.Common.DbParameter -Objekt am angegebenen Index der Sammlung zurück.

Name	Beschreibung
IndexOf-Methode	Gibt die Position des ULParameter-Objekts in der Sammlung zurück.
Insert-Methode	Fügt ein ULParameter-Objekt am angegebenen Index in die Sammlung ein.
Remove-Methode	Entfernt ein ULParameter-Objekt aus der Sammlung.
RemoveAt-Methode	Entfernt den Parameter des angegebenen Indexes aus der Sammlung.
SetParameter-Methode (geerbt aus System.Data.Common.DbParameterCollection)	Setzt das System.Data.Common.DbParameter -Objekt am angegebenen Index auf einen neuen Wert.
Count-Eigenschaft	Gibt die Anzahl der ULParameter-Objekte in der Sammlung zurück.
IsFixedSize-Eigenschaft	Gibt an, ob das ULParameterCollection-Objekt eine feste Größe hat.
IsReadOnly-Eigenschaft	Gibt an, ob das ULParameterCollection-Objekt schreibgeschützt ist.
IsSynchronized-Eigenschaft	Gibt an, ob das ULParameterCollection-Objekt synchronisiert ist.
SyncRoot-Eigenschaft	Gibt ein Objekt zurück, das verwendet werden kann, um den Zugriff auf das SAPParameterCollection-Objekt zu synchronisieren.
this-Eigenschaft	Gibt das ULParameter-Objekt am angegebenen Index zurück.

Bemerkungen

Alle Parameter in der Sammlung werden als positionsbasierte Parameter behandelt und in derselben Reihenfolge angegeben wie die Fragezeichen-Platzhalter im ULCommand.CommandText-Wert. Der erste Parameter in der Sammlung entspricht dem ersten Fragezeichen in der SQL-Anweisung, der zweite Parameter in der Sammlung dem zweiten Fragezeichen in der SQL-Anweisung, usw. Es müssen mindestens ebenso viele Fragezeichen im ULCommand.CommandText-Wert enthalten sein wie in der Sammlung. Für fehlende Parameter wird NULL verwendet.

Es gibt keinen Konstruktor für diese ULParameterCollection-Klasse. Von der ULCommand.Parameters-Eigenschaft erhalten Sie ein ULParameterCollection-Objekt.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand.Parameters-Eigenschaft \[UltraLite.NET\] auf Seite 107](#)
- [System.Data.Common.DbParameterCollection](#)
- [System.Data.IDataParameterCollection](#)

Add-Methode

Fügt ein ULParameter-Objekt zur Sammlung hinzu.

Überladungsliste

Name	Beschreibung
Add(Object)-Methode	Fügt ein ULParameter-Objekt zur Sammlung hinzu.
Add(String, Object)-Methode	Fügt der Sammlung ein neues ULParameter-Objekt hinzu, das mit dem angegebenen Parameternamen und -wert erstellt wurde.
Add(string, ULDbType)-Methode	Fügt der Sammlung ein neues ULParameter-Objekt hinzu, das mit dem angegebenen Parameternamen und -datentyp erstellt wurde.
Add(string, ULDbType, int)-Methode	Fügt der Sammlung ein neues ULParameter-Objekt hinzu, das mit dem angegebenen Parameternamen sowie Datentyp und Länge des Parameters erstellt wurde.
Add(string, ULDbType, int, string)-Methode	Fügt der Sammlung ein neues ULParameter-Objekt hinzu, das mit dem angegebenen Parameternamen sowie Datentyp, Länge und Quellspaltenname des Parameters erstellt wurde.
Add(ULParameter)-Methode	Fügt ein ULParameter-Objekt zur Sammlung hinzu.

Add(Object)-Methode

Fügt ein ULParameter-Objekt zur Sammlung hinzu.

Visual Basic-Syntax

```
Public Overrides Function Add(ByVal value As Object) As Integer
```

C#-Syntax

```
public override int Add(object value)
```

Parameter

- **value** Das ULParameter-Objekt, das der Sammlung hinzugefügt werden soll.

Rückgabe

Der Index des neuen ULParameter-Objekts.

Ausnahmebedingungen

- **ArgumentNullException** Der Wert kann nicht NULL sein ("Nothing" in Visual Basic).
- **InvalidCastException** Der angegebene Wert muss ein ULParameter-Objekt sein.
- **ArgumentException** Das ULParameter-Objekt kann der Sammlung nur einmal hinzugefügt werden.

Bemerkungen

Alle Parameter in der Sammlung werden als positionsbasierte Parameter behandelt und müssen der Sammlung in der Reihenfolge der Fragezeichen-Platzhalter im ULCommand.CommandText-Wert hinzugefügt werden. Der erste Parameter in der Sammlung entspricht dem ersten Fragezeichen in der SQL-Anweisung, der zweite Parameter in der Sammlung dem zweiten Fragezeichen in der SQL-Anweisung, usw. Es müssen mindestens ebenso viele Fragezeichen im ULCommand.CommandText-Wert enthalten sein wie in der Sammlung. Für fehlende Parameter wird NULL verwendet.

Siehe auch

- [ULParameterCollection.Add-Methode \[UltraLite.NET\] auf Seite 335](#)
- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)

Add(String, Object)-Methode

Fügt der Sammlung ein neues ULParameter-Objekt hinzu, das mit dem angegebenen Parameternamen und -wert erstellt wurde.

Visual Basic-Syntax

```
Public Function Add(  
    ByVal parameterName As String,  
    ByVal value As Object  
) As ULParameter
```

C#-Syntax

```
public ULParameter Add(string parameterName, object value)
```

Parameter

- **parameterName** Der Name des Parameters. Verwenden Sie für nicht benannte Parameter eine leere Zeichenfolge (""), oder eine Nullreferenz ("Nothing" in Visual Basic) für diesen Wert. In UltraLite.NET werden Parameternamen nicht vom ULCommand-Objekt verwendet.
- **value** Ein System.Object-Objekt, das der Wert des Parameters sein soll.

Rückgabe

Das neue ULParameter-Objekt.

Bemerkungen

Alle Parameter in der Sammlung werden als positionsbasierte Parameter behandelt und müssen der Sammlung in der Reihenfolge der Fragezeichen-Platzhalter im ULCommand.CommandText-Wert hinzugefügt werden. Der erste Parameter in der Sammlung entspricht dem ersten Fragezeichen in der SQL-Anweisung, der zweite Parameter in der Sammlung dem zweiten Fragezeichen in der SQL-Anweisung, usw. Es müssen mindestens ebenso viele Fragezeichen im ULCommand.CommandText-Wert enthalten sein wie in der Sammlung. Für fehlende Parameter wird NULL verwendet.

Aufgrund der speziellen Behandlung der 0- und 0.0-Konstanten und der Art, in der überladene Methoden gelöst werden, wird dringend empfohlen, bei Verwendung dieser Methode Konstantenwerte explizit in den Objekttyp umzuwandeln.

Siehe auch

- [ULParameterCollection.Add-Methode \[UltraLite.NET\] auf Seite 335](#)
- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)
- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [System.Object](#)

Beispiel

Der folgende Code fügt einen ULParameter-Wert von 0 zu einem ULCommand-Objekt namens cmd hinzu.

```
' Visual Basic
cmd.Parameters.Add( "", CType( 0, Object ) )
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
cmd.Parameters.Add( "", (object)0 );
```

Add(string, ULDbType)-Methode

Fügt der Sammlung ein neues ULParameter-Objekt hinzu, das mit dem angegebenen Parameternamen und -datentyp erstellt wurde.

Visual Basic-Syntax

```
Public Function Add(
    ByVal parameterName As String,
    ByVal ulDbType As ULDbType
) As ULParameter
```

C#-Syntax

```
public ULParameter Add(string parameterName, ULDbType ulDbType)
```

Parameter

- **parameterName** Der Name des Parameters. Verwenden Sie für nicht benannte Parameter eine leere Zeichenfolge (""), oder eine Nullreferenz ("Nothing" in Visual Basic) für diesen Wert. In UltraLite.NET werden Parameternamen nicht vom ULCommand-Objekt verwendet.

- **ulDbType** Einer der iAnywhere.Data.UltraLite.ULDbType-Werte.

Rückgabe

Das neue ULParameter-Objekt.

Bemerkungen

Alle Parameter in der Sammlung werden als positionsbasierte Parameter behandelt und müssen der Sammlung in der Reihenfolge der Fragezeichen-Platzhalter im ULCommand.CommandText-Wert hinzugefügt werden. Der erste Parameter in der Sammlung entspricht dem ersten Fragezeichen in der SQL-Anweisung, der zweite Parameter in der Sammlung dem zweiten Fragezeichen in der SQL-Anweisung, usw. Es müssen mindestens ebenso viele Fragezeichen im ULCommand.CommandText-Wert enthalten sein wie in der Sammlung. Für fehlende Parameter wird NULL verwendet.

Siehe auch

- [ULParameterCollection.Add-Methode \[UltraLite.NET\] auf Seite 335](#)
- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)
- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)

Add(string, ULDbType, int)-Methode

Fügt der Sammlung ein neues ULParameter-Objekt hinzu, das mit dem angegebenen Parameternamen sowie Datentyp und Länge des Parameters erstellt wurde.

Visual Basic-Syntax

```
Public Function Add(  
    ByVal parameterName As String,  
    ByVal ulDbType As ULDbType,  
    ByVal size As Integer  
) As ULParameter
```

C#-Syntax

```
public ULParameter Add(  
    string parameterName,  
    ULDbType ulDbType,  
    int size  
)
```

Parameter

- **parameterName** Der Name des Parameters. Verwenden Sie für nicht benannte Parameter eine leere Zeichenfolge (""), oder eine Nullreferenz ("Nothing" in Visual Basic) für diesen Wert. In UltraLite.NET werden Parameternamen nicht vom ULCommand-Objekt verwendet.
- **ulDbType** Einer der iAnywhere.Data.UltraLite.ULDbType-Werte.
- **size** Die Länge des Parameters.

Rückgabe

Das neue ULParameter-Objekt.

Bemerkungen

Alle Parameter in der Sammlung werden als positionsbasierte Parameter behandelt und müssen der Sammlung in der Reihenfolge der Fragezeichen-Platzhalter im ULCommand.CommandText-Wert hinzugefügt werden. Der erste Parameter in der Sammlung entspricht dem ersten Fragezeichen in der SQL-Anweisung, der zweite Parameter in der Sammlung dem zweiten Fragezeichen in der SQL-Anweisung, usw. Es müssen mindestens ebenso viele Fragezeichen im ULCommand.CommandText-Wert enthalten sein wie in der Sammlung. Für fehlende Parameter wird NULL verwendet.

Siehe auch

- [ULParameterCollection.Add-Methode \[UltraLite.NET\] auf Seite 335](#)
- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)
- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)

Add(string, ULDbType, int, string)-Methode

Fügt der Sammlung ein neues ULParameter-Objekt hinzu, das mit dem angegebenen Parameternamen sowie Datentyp, Länge und Quellspaltenname des Parameters erstellt wurde.

Visual Basic-Syntax

```
Public Function Add(  
    ByVal parameterName As String,  
    ByVal ulDbType As ULDbType,  
    ByVal size As Integer,  
    ByVal sourceColumn As String  
) As ULParameter
```

C#-Syntax

```
public ULParameter Add(  
    string parameterName,  
    ULDbType ulDbType,  
    int size,  
    string sourceColumn  
)
```

Parameter

- **parameterName** Der Name des Parameters. Verwenden Sie für nicht benannte Parameter eine leere Zeichenfolge (""), oder eine Nullreferenz ("Nothing" in Visual Basic) für diesen Wert. In UltraLite.NET werden Parameternamen nicht vom ULCommand-Objekt verwendet.
- **ulDbType** Einer der iAnywhere.Data.UltraLite.ULDbType-Werte.
- **size** Die Länge des Parameters.
- **sourceColumn** Der Name der Quellspalte, die zuzuordnen ist.

Rückgabe

Das neue ULParameter-Objekt.

Bemerkungen

Alle Parameter in der Sammlung werden als positionsbasierte Parameter behandelt und müssen der Sammlung in der Reihenfolge der Fragezeichen-Platzhalter im ULCommand.CommandText-Wert hinzugefügt werden. Der erste Parameter in der Sammlung entspricht dem ersten Fragezeichen in der SQL-Anweisung, der zweite Parameter in der Sammlung dem zweiten Fragezeichen in der SQL-Anweisung, usw. Es müssen mindestens ebenso viele Fragezeichen im ULCommand.CommandText-Wert enthalten sein wie in der Sammlung. Für fehlende Parameter wird NULL verwendet.

Siehe auch

- [ULParameterCollection.Add-Methode \[UltraLite.NET\] auf Seite 335](#)
- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)
- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)

Add(ULParameter)-Methode

Fügt ein ULParameter-Objekt zur Sammlung hinzu.

Visual Basic-Syntax

```
Public Function Add(ByVal value As ULParameter) As ULParameter
```

C#-Syntax

```
public ULParameter Add(ULParameter value)
```

Parameter

- **value** Das ULParameter-Objekt, das der Sammlung hinzugefügt werden soll.

Rückgabe

Das neue ULParameter-Objekt.

Ausnahmebedingungen

- **ArgumentNullException** Der Wert kann nicht NULL sein ("Nothing" in Visual Basic).
- **ArgumentException** Das ULParameter-Objekt kann der Sammlung nur einmal hinzugefügt werden.

Bemerkungen

Alle Parameter in der Sammlung werden als positionsbasierte Parameter behandelt und müssen der Sammlung in der Reihenfolge der Fragezeichen-Platzhalter im ULCommand.CommandText-Wert hinzugefügt werden. Der erste Parameter in der Sammlung entspricht dem ersten Fragezeichen in der SQL-Anweisung, der zweite Parameter in der Sammlung dem zweiten Fragezeichen in der SQL-

Anweisung, usw. Es müssen mindestens ebenso viele Fragezeichen im ULCommand.CommandText-Wert enthalten sein wie in der Sammlung. Für fehlende Parameter wird NULL verwendet.

Siehe auch

- [ULParameterCollection.Add-Methode \[UltraLite.NET\] auf Seite 335](#)
- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)
- [ULCommand.CommandText-Eigenschaft \[UltraLite.NET\] auf Seite 104](#)

AddRange-Methode

Fügt ein Array von Werten an das Ende der ULParameterCollection-Sammlung an.

Überladungsliste

Name	Beschreibung
AddRange(Array)-Methode	Fügt ein Array von Werten an das Ende der ULParameterCollection-Sammlung an.
AddRange(ULParameter[])-Methode	Fügt ein Array von Werten an das Ende der ULParameterCollection-Sammlung an.

AddRange(Array)-Methode

Fügt ein Array von Werten an das Ende der ULParameterCollection-Sammlung an.

Visual Basic-Syntax

```
Public Overrides Sub AddRange(ByVal values As Array)
```

C#-Syntax

```
public override void AddRange(Array values)
```

Parameter

- **values** Ein Array von ULParameter-Objekten, die an das Ende dieser Sammlung hinzugefügt werden sollen.

Siehe auch

- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)

AddRange(ULParameter[])-Methode

Fügt ein Array von Werten an das Ende der ULParameterCollection-Sammlung an.

Visual Basic-Syntax

```
Public Sub AddRange(ByVal values As ULParameter())
```

C#-Syntax

```
public void AddRange(ULParameter[] values)
```

Parameter

- **values** Ein Array von ULParameter-Objekten, die an das Ende dieser Sammlung hinzugefügt werden sollen.

Bemerkungen

Dies ist die stark typisierte Version der DbParameterCollection.AddRange(Array)-Methode.

Siehe auch

- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)
- [ULParameterCollection-Klasse \[UltraLite.NET\] auf Seite 333](#)

Clear-Methode

Entfernt alle Parameter aus der Sammlung.

Visual Basic-Syntax

```
Public Overrides Sub Clear()
```

C#-Syntax

```
public override void Clear()
```

Contains-Methode

Prüft, ob ein ULParameter-Objekt in der Sammlung vorhanden ist.

Überladungsliste

Name	Beschreibung
Contains(Object)-Methode	Prüft, ob ein ULParameter-Objekt in der Sammlung vorhanden ist.
Contains(String)-Methode	Prüft, ob ein ULParameter-Objekt mit dem angegebenen Namen in der Sammlung vorhanden ist.

Contains(Object)-Methode

Prüft, ob ein ULParameter-Objekt in der Sammlung vorhanden ist.

Visual Basic-Syntax

```
Public Overrides Function Contains(ByVal value As Object) As Boolean
```

C#-Syntax

```
public override bool Contains(object value)
```

Parameter

- **value** Das ULParameter-Objekt, auf das geprüft werden soll.

Rückgabe

TRUE, wenn die Sammlung das ULParameter-Objekt enthält, andernfalls FALSE.

Siehe auch

- [ULParameterCollection.Contains-Methode \[UltraLite.NET\] auf Seite 342](#)
- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)

Contains(String)-Methode

Prüft, ob ein ULParameter-Objekt mit dem angegebenen Namen in der Sammlung vorhanden ist.

Visual Basic-Syntax

```
Public Overrides Function Contains(ByVal value As String) As Boolean
```

C#-Syntax

```
public override bool Contains(string value)
```

Parameter

- **value** Der Name des zu suchenden Parameters.

Rückgabe

TRUE, wenn die Sammlung das ULParameter-Objekt enthält, andernfalls FALSE.

Siehe auch

- [ULParameterCollection.Contains-Methode \[UltraLite.NET\] auf Seite 342](#)
- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)

CopyTo-Methode

Kopiert ULParameter-Objekte aus dem ULParameterCollection-Objekt in das angegebene Array.

Visual Basic-Syntax

```
Public Overrides Sub CopyTo(  
    ByVal array As Array,  
    ByVal index As Integer  
)
```

C#-Syntax

```
public override void CopyTo(Array array, int index)
```

Parameter

- **array** Das Array, in das die ULParameter-Objekte kopiert werden sollen.
- **index** Der Startindex des Arrays.

Siehe auch

- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)

GetEnumerator-Methode

Gibt einen Enumerator für die Sammlung zurück.

Visual Basic-Syntax

```
Public Overrides Function GetEnumerator()  
    As System.Collections.IEnumerator
```

C#-Syntax

```
public override IEnumerator GetEnumerator()
```

Rückgabe

Ein ArrayList-Enumerator, der die Parameter in der Sammlung aufzählt.

IndexOf-Methode

Gibt die Position des ULParameter-Objekts in der Sammlung zurück.

Überladungsliste

Name	Beschreibung
IndexOf(Object)-Methode	Gibt die Position des ULParameter-Objekts in der Sammlung zurück.
IndexOf(String)-Methode	Gibt die Position des ULParameter-Objekts mit dem angegebenen Namen in der Sammlung zurück.

IndexOf(Object)-Methode

Gibt die Position des ULParameter-Objekts in der Sammlung zurück.

Visual Basic-Syntax

```
Public Overrides Function IndexOf(ByVal value As Object) As Integer
```

C#-Syntax

```
public override int IndexOf(object value)
```

Parameter

- **value** Das ULParameter-Objekt, das gefunden werden soll.

Rückgabe

Der nullbasierte Index des ULParameter-Objekts in der Sammlung oder -1, wenn der Parameter nicht gefunden wird.

Ausnahmen

- **InvalidCastException** Der angegebene Wert muss ein ULParameter-Objekt sein.

Siehe auch

- [ULParameterCollection.IndexOf-Methode \[UltraLite.NET\] auf Seite 344](#)
- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)

IndexOf(String)-Methode

Gibt die Position des ULParameter-Objekts mit dem angegebenen Namen in der Sammlung zurück.

Visual Basic-Syntax

```
Public Overrides Function IndexOf(  
    ByVal parameterName As String  
) As Integer
```

C#-Syntax

```
public override int IndexOf(string parameterName)
```

Parameter

- **parameterName** Der Name des Parameters, der gefunden werden soll.

Rückgabe

Der nullbasierte Index des ULParameter-Objekts in der Sammlung oder -1, wenn der Parameter nicht gefunden wird.

Siehe auch

- [ULParameterCollection.IndexOf-Methode \[UltraLite.NET\] auf Seite 344](#)
- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)

Insert-Methode

Fügt ein ULParameter-Objekt am angegebenen Index in die Sammlung ein.

Visual Basic-Syntax

```
Public Overrides Sub Insert(  
    ByVal index As Integer,
```

```
        ByVal value As Object  
    )
```

C#-Syntax

```
public override void Insert(int index, object value)
```

Parameter

- **index** Der auf Null basierende Index, in den der Parameter innerhalb der Sammlung eingefügt werden soll.
- **value** Das ULParameter-Objekt, das eingefügt werden soll.

Ausnahmen

- **IndexOutOfRangeException** Der Index ist ungültig.
- **ArgumentNullException** Ein Parameter kann nicht mit einer Nullreferenz ("Nothing" in Visual Basic) festgelegt werden.
- **InvalidCastException** Der angegebene Wert muss ein ULParameter-Objekt sein.

Siehe auch

- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)

Remove-Methode

Entfernt ein ULParameter-Objekt aus der Sammlung.

Visual Basic-Syntax

```
Public Overrides Sub Remove(ByVal value As Object)
```

C#-Syntax

```
public override void Remove(object value)
```

Parameter

- **value** Das ULParameter-Objekt, das entfernt werden soll.

Ausnahmebedingungen

- **ArgumentNullException** Ein Parameter kann nicht mit einer Nullreferenz ("Nothing" in Visual Basic) festgelegt werden.
- **InvalidCastException** Der angegebene Wert muss ein ULParameter-Objekt sein.
- **ArgumentException** Die Sammlung enthält den angegebenen Parameter nicht.

Siehe auch

- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)

RemoveAt-Methode

Entfernt den Parameter des angegebenen Indexes aus der Sammlung.

Überladungsliste

Name	Beschreibung
RemoveAt(Int)-Methode	Entfernt den Parameter des angegebenen Indexes aus der Sammlung.
RemoveAt(String)-Methode	Löscht den Parameter mit dem angegebenen Namen aus der Sammlung.

RemoveAt(Int)-Methode

Entfernt den Parameter des angegebenen Indexes aus der Sammlung.

Visual Basic-Syntax

```
Public Overrides Sub RemoveAt(ByVal index As Integer)
```

C#-Syntax

```
public override void RemoveAt(int index)
```

Parameter

- **index** Der auf Null basierende Index des zu entfernenden Parameters. Der Wert muss im Bereich [0,ULParameterCollection.Count-1] liegen. Der erste Parameter in der Sammlung hat den Indexwert Null.

Ausnahmen

- **IndexOutOfRangeException** Der Index ist ungültig.

Siehe auch

- [ULParameterCollection.RemoveAt-Methode \[UltraLite.NET\] auf Seite 347](#)
- [ULParameterCollection.Count-Eigenschaft \[UltraLite.NET\] auf Seite 348](#)

RemoveAt(String)-Methode

Löscht den Parameter mit dem angegebenen Namen aus der Sammlung.

Visual Basic-Syntax

```
Public Overrides Sub RemoveAt(ByVal parameterName As String)
```

C#-Syntax

```
public override void RemoveAt(string parameterName)
```

Parameter

- **parameterName** Der Name des abzurufenden Parameters.

Ausnahmen

- **IndexOutOfRangeException** Es gibt keinen Parameter mit dem angegebenen Namen.

Siehe auch

- [ULParameterCollection.RemoveAt-Methode \[UltraLite.NET\] auf Seite 347](#)

Count-Eigenschaft

Gibt die Anzahl der ULParameter-Objekte in der Sammlung zurück.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property Count As Integer
```

C#-Syntax

```
public override int Count {get;}
```

Bemerkungen

Die Anzahl von ULParameter-Objekten in der Sammlung.

IsFixedSize-Eigenschaft

Gibt an, ob das ULParameterCollection-Objekt eine feste Größe hat.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property IsFixedSize As Boolean
```

C#-Syntax

```
public override bool IsFixedSize {get;}
```

Bemerkungen

TRUE, wenn diese Sammlung eine feste Größe hat, andernfalls FALSE.

IsReadOnly-Eigenschaft

Gibt an, ob das ULParameterCollection-Objekt schreibgeschützt ist.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property IsReadOnly As Boolean
```

C#-Syntax

```
public override bool IsReadOnly {get;}
```

Bemerkungen

TRUE, wenn diese Sammlung schreibgeschützt ist, andernfalls FALSE.

IsSynchronized-Eigenschaft

Gibt an, ob das ULParameterCollection-Objekt synchronisiert ist.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property IsSynchronized As Boolean
```

C#-Syntax

```
public override bool IsSynchronized {get;}
```

Bemerkungen

TRUE, wenn diese Sammlung synchronisiert ist, andernfalls FALSE.

SyncRoot-Eigenschaft

Gibt ein Objekt zurück, das verwendet werden kann, um den Zugriff auf das SAParameterCollection-Objekt zu synchronisieren.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property SyncRoot As Object
```

C#-Syntax

```
public override object SyncRoot {get;}
```

Bemerkungen

Das Objekt, das verwendet werden soll, um den Zugriff auf diese Sammlung zu synchronisieren.

this-Eigenschaft

Gibt das ULParameter-Objekt am angegebenen Index zurück.

Überladungsliste

Name	Beschreibung
this[Int]-Eigenschaft	Gibt das ULParameter-Objekt am angegebenen Index zurück.
this[String]-Eigenschaft	Gibt das ULParameter-Objekt mit dem angegebenen Namen zurück.

this[Int]-Eigenschaft

Gibt das ULParameter-Objekt am angegebenen Index zurück.

Visual Basic-Syntax

```
Public Shadows Property Item(ByVal index As Integer) As ULParameter
```

C#-Syntax

```
public new ULParameter this[int index] {get;set;}
```

Parameter

- **index** Der auf Null basierende Index des abzurufenden Parameters. Der Wert muss im Bereich [0,ULParameterCollection.Count-1] liegen. Der erste Parameter in der Sammlung hat den Indexwert Null.

Rückgabe

Das ULParameter-Objekt an der Stelle des angegebenen Indexes.

Ausnahmen

- **IndexOutOfRangeException** Der Index ist ungültig.

Bemerkungen

In C# ist diese Eigenschaft der Indexersteller für die ULParameterCollection-Klasse.

Dies ist die stark typisierte Version der DbParameterCollection.this[int]-Eigenschaft.

Siehe auch

- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)

this[String]-Eigenschaft

Gibt das ULParameter-Objekt mit dem angegebenen Namen zurück.

Visual Basic-Syntax

```
Public Shadows Property Item(  
    ByVal parameterName As String  
) As ULParameter
```

C#-Syntax

```
public new ULParameter this[string parameterName] {get;set;}
```

Parameter

- **parameterName** Der Name des abzurufenden Parameters.

Rückgabe

Das ULParameter-Objekt mit dem angegebenen Namen.

Ausnahmen

- **IndexOutOfRangeException** Es gibt keinen Parameter mit dem angegebenen Namen.
- **ArgumentNullException** Ein Parameter kann nicht mit NULL als Parameternamen ("Nothing" in Visual Basic) festgelegt werden.

Bemerkungen

In C# ist diese Eigenschaft der Indexersteller für die ULParameterCollection-Klasse.

Dies ist die stark typisierte Version der DbParameterCollection.this[string]-Eigenschaft.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetValue-Methode \[UltraLite.NET\] auf Seite 263](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULParameter-Klasse \[UltraLite.NET\] auf Seite 318](#)

ULResultSet-Klasse

UL-Erw.: Stellt eine bearbeitbare Ergebnismenge in einer UltraLite-Datenbank dar.

Visual Basic-Syntax

```
Public Class ULResultSet Inherits ULDataReader
```

C#-Syntax

```
public class ULResultSet : ULDataReader
```

Basisklassen

- [ULDataReader-Klasse \[UltraLite.NET\] auf Seite 237](#)

Abgeleitete Klassen.

- [ULTable-Klasse \[UltraLite.NET\] auf Seite 414](#)

Mitglieder

Alle Mitglieder der ULResultSet-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
AppendBytes-Methode	Hängt die angegebene Teilmenge des angegebenen Arrays von System.Bytes an den neuen Wert für die angegebene ULDbType.Long-Binary-Spalte an.
AppendChars-Methode	Hängt die angegebene Teilmenge des angegebenen Arrays von System.Chars an den neuen Wert für die angegebene ULDbType.Long-Varchar-Spalte an.

Name	Beschreibung
Close-Methode	Schließt den Cursor.
Delete-Methode	Löscht die aktuelle Zeile.
Dispose-Methode (geerbt aus System.Data.Common.DbDataReader)	Gibt alle Ressourcen frei, die von der aktuellen Instanz der Klasse System.Data.Common.DbDataReader verwendet werden.
GetBoolean-Methode	Gibt den Wert für die angegebene Spalte als System.Boolean zurück.
GetByte-Methode	Gibt den Wert für die angegebene Spalte als 8-Bit-Wert ohne Vorzeichen zurück (System.Byte).
GetBytes-Methode	UL-Erw.: Gibt den Wert für die angegebene Spalte als Array von System.Bytes-Werten zurück.
GetChar-Methode	Diese Methode wird in UltraLite.NET nicht unterstützt.
GetChars-Methode	Kopiert eine Untermenge des Werts für die angegebene ULDb-Type.LongVarchar-Spalte, beginnend beim angegebenen Offset, in das angegebene Offset des System.Char-Zielarrays.
GetData-Methode (Geerbt von System.Data.Common.DbDataReader)	Gibt ein System.Data.Common.DbDataReader -Objekt für die angeforderte Spalten-Ordinalzahl zurück.
GetDataTypeName-Methode	Gibt den Namen des Provider-Datentyps der angegebenen Spalte zurück.
GetDateTime-Methode	Gibt den Wert für die angegebene Spalte als System.DateTime-Typ mit Millisekunden-Genauigkeit zurück.
GetDbDataReader-Methode (geerbt aus System.Data.Common.DbDataReader)	Gibt ein System.Data.Common.DbDataReader -Objekt für die angeforderte Spaltenordinalzahl zurück, das mit einer providerspezifischen Implementierung aufgehoben werden kann.
GetDecimal-Methode	Gibt den Wert für die angegebene Spalte als System.Decimal-Typ zurück.
GetDouble-Methode	Gibt den Wert für die angegebene Spalte als System.Double-Typ zurück.
GetEnumerator-Methode	Gibt einen System.Collections.IEnumerator-Wert zurück, der das ULDataReader-Objekt durchläuft.
GetFieldType-Methode	Gibt den System.Type-Wert zurück, der für die angegebene Spalte am besten geeignet ist.

Name	Beschreibung
GetFieldValue(T)-Methode (geerbt aus System.Data.Common.DbDataReader)	Ruft den Wert der angegebenen Spalte synchron als Typ ab.
GetFieldValueAsync(T)-Methode (geerbt aus System.Data.Common.DbDataReader)	Ruft den Wert der angegebenen Spalte asynchron als Typ ab.
GetFloat-Methode	Gibt den Wert für die angegebene Spalte als System.Single-Typ zurück.
GetGuid-Methode	Gibt den Wert für die angegebene Spalte als UUID-Typ zurück (System.Guid).
GetInt16-Methode	Gibt den Wert für die angegebene Spalte als System.Int16-Typ zurück.
GetInt32-Methode	Gibt den Wert für die angegebene Spalte als System.Int32-Typ zurück.
GetInt64-Methode	Gibt den Wert für die angegebene Spalte als System.Int64-Typ zurück.
GetName-Methode	Gibt den Namen der angegebenen Spalte zurück.
GetOrdinal-Methode	Gibt die Spalten-ID der angegebenen Spalte zurück.
GetProviderSpecificFieldType-Methode (geerbt aus System.Data.Common.DbDataReader)	Gibt den providerspezifischen Feldtyp der angegebenen Spalte zurück.
GetProviderSpecificValue-Methode (geerbt aus System.Data.Common.DbDataReader)	Ruft den Wert der angegebenen Spalte als Instanz von System.Object ab.
GetProviderSpecificValues-Methode (geerbt aus System.Data.Common.DbDataReader)	Ruft alle providerspezifischen Attributspalten in der Sammlung für die aktuelle Zeile ab.
GetRowCount-Methode	UL-Erw.: Gibt die Anzahl der Zeilen im Cursor innerhalb des Schwellenwerts zurück.

Name	Beschreibung
GetSchemaTable-Methode	Gibt einen System.Data.DataTable-Wert zurück, der die Spaltenmetadaten des ULDataReader-Objekts beschreibt.
GetStream-Methode (geerbt aus System.Data.Common.DbDataReader)	Ruft Daten als System.IO.Stream ab.
GetString-Methode	Gibt den Wert für die angegebene Spalte als System.String-Typ zurück.
GetTextReader-Methode (geerbt aus System.Data.Common.DbDataReader)	Ruft Daten als System.IO.TextReader ab.
GetTimeSpan-Methode	Gibt den Wert für die angegebene Spalte als System.TimeSpan-Typ mit Millisekunden-Genauigkeit zurück.
GetUInt16-Methode	Gibt den Wert für die angegebene Spalte als System.UInt16-Typ zurück.
GetUInt32-Methode	Gibt den Wert für die angegebene Spalte als System.UInt32-Typ zurück.
GetUInt64-Methode	Gibt den Wert für die angegebene Spalte als System.UInt64-Typ zurück.
GetValue-Methode	Gibt den Wert der angegebenen Spalte in ihrem nativen Format zurück.
GetValues-Methode	Gibt alle Spaltenwerte für die aktuelle Zeile zurück.
IsDBNull-Methode	Prüft, ob der Wert aus der angegebenen Spalte NULL ist.
IsDBNullAsync-Methode (geerbt aus System.Data.Common.DbDataReader)	Eine asynchrone Version von System.Data.Common.DbDataReader.IsDBNull(System.Int32) , die einen Wert abruft, der angibt, ob die Spalte nicht vorhandene oder fehlende Werte enthält.
MoveAfterLast-Methode	UL-Erw.: Positioniert den Cursor hinter die letzte Zeile des Cursors.
MoveBeforeFirst-Methode	UL-Erw.: Positioniert den Cursor vor die erste Zeile des Cursors.
MoveFirst-Methode	UL-Erw.: Positioniert den Cursor auf die erste Zeile des Cursors.
MoveLast-Methode	UL-Erw.: Positioniert den Cursor auf die letzte Zeile des Cursors.
MoveNext-Methode	UL-Erw.: Positioniert den Cursor in die nächste Zeile oder nach der letzten Zeile, wenn der Cursor bereits in der letzten Zeile war.

Name	Beschreibung
MovePrevious-Methode	UL-Erw.: Positioniert den Cursor in die vorhergehende Zeile oder vor die erste Zeile.
MoveRelative-Methode	UL-Erw.: Positioniert den Cursor relativ zur aktuellen Zeile.
NextResult-Methode	Rückt das ULDataReader-Objekt zum nächsten Ergebnis vor, wenn Ergebnisse von Batch-SQL-Anweisungen gelesen werden.
NextResultAsync-Methode (geerbt aus System.Data.Common.DbDataReader)	Eine asynchrone Version von System.Data.Common.DbDataReader.NextResult , die den Leser beim Lesen der Ergebnisse von Anweisungsfolgen von statements.Invokes System.Data.Common.DbDataReader.NextResultAsync(System.Threading.CancellationToken) mit CancellationToken.None zum nächsten Ergebnis vorrückt.
Read-Methode	Positioniert den Cursor in die nächste Zeile oder nach der letzten Zeile, wenn der Cursor bereits in der letzten Zeile war.
ReadAsync-Methode (geerbt aus System.Data.Common.DbDataReader)	Eine asynchrone Version von System.Data.Common.DbDataReader.Read , die den Leser zum nächsten Datensatz in der Ergebnismenge vorrückt.
SetBoolean-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Boolean-Objekts fest.
SetByte-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Byte-Objekts (8-Bit-Ganzzahl ohne Vorzeichen) fest.
SetBytes-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Bytes-Arrays fest.
SetDateTime-Methode	Legt den Wert für die angegebene Spalte mithilfe eines System.Date.Time-Objekts fest.
SetDBNull-Methode	Setzt eine Spalte auf NULL.
SetDecimal-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Decimal-Objekts fest.
SetDouble-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Double-Objekts fest.
SetFloat-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Single-Objekts fest.
SetGuid-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Guid-Objekts fest.

Name	Beschreibung
SetInt16-Methode	Legt den Wert für die angegebene Spalte unter Verwendung von System.Int16 fest.
SetInt32-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Int32 fest.
SetInt64-Methode	Setzt den Wert für die angegebene Spalte unter Verwendung eines Int64-Werts.
SetString-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.String-Objekts fest.
SetTimeSpan-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.TimeSpan-Objekts fest.
SetToDefault-Methode	Setzt den Wert bei der angegebenen Spalte auf ihren Standardwert.
SetUInt16-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.UInt16-Objekts fest.
SetUInt32-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.UInt32-Objekts fest.
SetUInt64-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.UInt64-Objekts fest.
Update-Methode	Aktualisiert die aktuelle Zeile mit den aktuellen Spaltenwerten (mit set-Methoden angegeben).
UpdateBegin-Methode	Bereitet die Aktualisierung der aktuellen Zeile vor.
Depth-Eigenschaft	Gibt die Tiefe der Verschachtelung für die aktuelle Zeile zurück.
FieldCount-Eigenschaft	Gibt die Anzahl der Spalten im Cursor zurück.
HasRows-Eigenschaft	Prüft, ob das ULDataReader-Objekt eine oder mehrere Zeilen hat.
IsBOF-Eigenschaft	UL-Erw.: Gibt an, ob sich die aktuelle Zeilenposition vor der ersten Zeile befindet.
IsClosed-Eigenschaft	Prüft, ob der Cursor aktuell offen ist.
IsEOF-Eigenschaft	UL-Erw.: Gibt an, ob sich die aktuelle Zeilenposition nach der letzten Zeile befindet.
RecordsAffected-Eigenschaft	Gibt die Anzahl der durch eine SQL-Anweisung geänderten, eingefügten oder gelöschten Zeilen zurück.

Name	Beschreibung
RowCount-Eigenschaft	UL-Erw.: Gibt die Anzahl der Zeilen im Cursor zurück.
Schema-Eigenschaft	UL-Erw.: Enthält das Schema dieses Cursors.
this-Eigenschaft	Gibt den Wert der angegebenen Spalte in ihrem nativen Format zurück.
VisibleFieldCount-Eigenschaft (geerbt aus System.Data.Common.DbDataReader)	Ruft die Anzahl der Felder in System.Data.Common.DbDataReader ab, die nicht ausgeblendet sind.

Bemerkungen

Es gibt keinen Konstruktor für diese Klasse. Ergebnismengen werden mit der `ULCommand.ExecuteResultSet`-Methode erstellt.

```
' Visual Basic
Dim cmd As ULCommand = new ULCommand( _
    "SELECT emp_id FROM employee", conn _
)
Dim resultSet As ULResultSet = cmd.ExecuteResultSet()
```

Dies entspricht dem folgenden Code in der Sprache C#:

```
// C#
ULCommand cmd = new ULCommand(
    "SELECT emp_id FROM employee", conn
);
ULResultSet resultSet = cmd.ExecuteResultSet();
```

Ein `ULResultSet`-Objekt stellt eine bearbeitbare Ergebnismenge dar, für die Sie positionsbasierte Aktualisierungen und Löschungen durchführen können. Verwenden Sie für vollständig bearbeitbare Ergebnismengen die `ULCommand.ExecuteTable`-Methode oder die `ULDataAdapter`-Klasse.

Siehe auch

- [ULCommand.ExecuteResultSet-Methode \[UltraLite.NET\] auf Seite 97](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULResultSet-Klasse \[UltraLite.NET\] auf Seite 351](#)
- [ULCommand.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 100](#)
- [ULDataAdapter-Klasse \[UltraLite.NET\] auf Seite 212](#)
- [ULDataReader-Klasse \[UltraLite.NET\] auf Seite 237](#)
- [System.Data.IDataReader](#)
- [System.Data.IDataRecord](#)
- [System.IDisposable](#)

AppendBytes-Methode

Hängt die angegebene Teilmenge des angegebenen Arrays von `System.Bytes` an den neuen Wert für die angegebene `ULDbType.LongBinary`-Spalte an.

Visual Basic-Syntax

```
Public Sub AppendBytes(  
    ByVal colID As Integer,  
    ByVal val As Byte(),  
    ByVal srcOffset As Integer,  
    ByVal count As Integer  
)
```

C#-Syntax

```
public void AppendBytes(int colID, byte[] val, int srcOffset, int count)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Startposition im Ausgangs-Array
- **srcOffset** Die Startposition im Ausgangs-Array.
- **count** Die Anzahl der zu kopierenden Bytes.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Bytes an Position *srcOffset* (beginnend bei 0) bis *srcOffset + count - 1* des *val*-Arrays werden an den Wert für die angegebene Spalte angehängt.

Bei Einfügungen initialisiert ULTable.InsertBegin den neuen Wert entsprechend dem Standardwert der Spalte. Die Daten in der Zeile werden erst geändert, wenn Sie ein ULTable.Insert ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Bei der Aktualisierung löscht die erste Anhängung an eine Spalte den aktuellen Spaltenwert, bevor der neue Wert angehängt wird.

Falls eine der folgenden Aussagen zutrifft, wird ein ULException-Objekt mit dem ULSQLCode.SQLE_INVALID_PARAMETER-Code ausgegeben und das Ziel wird nicht verändert:

- *val* ist Null.
- *srcOffset* ist negativ.
- *count* ist negativ.
- *srcOffset + count* ist größer als die *val*-Länge.

Bei anderen Fehlern wird ein ULException-Objekt mit dem entsprechenden Fehlercode ausgegeben.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.InsertBegin-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULException-Klasse \[UltraLite.NET\] auf Seite 275](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Byte](#)

AppendChars-Methode

Hängt die angegebene Teilmenge des angegebenen Arrays von System.Chars an den neuen Wert für die angegebene ULDbType.LongVarchar-Spalte an.

Visual Basic-Syntax

```
Public Sub AppendChars(
    ByVal colID As Integer,
    ByVal val As Char(),
    ByVal srcOffset As Integer,
    ByVal count As Integer
)
```

C#-Syntax

```
public void AppendChars(int colID, char[] val, int srcOffset, int count)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Startposition im Ausgangs-Array
- **srcOffset** Die Startposition im Ausgangs-Array.
- **count** Die Anzahl der zu kopierenden Bytes.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Zeichen an Position *srcOffset* (beginnend bei 0) bis *srcOffset + count - 1* des *val*-Arrays werden an den Wert für die angegebene Spalte angehängt. Bei Einfügungen initialisiert ULTable.InsertBegin den neuen Wert entsprechend dem Standardwert der Spalte. Die Daten in der Zeile werden erst geändert, wenn Sie ein ULTable.Insert ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Bei der Aktualisierung löscht die erste Anhängung an eine Spalte den aktuellen Spaltenwert, bevor der neue Wert angehängt wird.

Falls eine der folgenden Aussagen zutrifft, wird ein `ULException`-Objekt mit dem `ULSQLCode.SQLE_INVALID_PARAMETER`-Code ausgegeben und das Ziel wird nicht verändert:

- *val* ist Null.
- *srcOffset* ist negativ.
- *count* ist negativ.
- *srcOffset* + *count* ist größer als die *val*-Länge.

Bei anderen Fehlern wird ein `ULException`-Objekt mit dem entsprechenden Fehlercode ausgegeben.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.InsertBegin-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULException-Klasse \[UltraLite.NET\] auf Seite 275](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Char](#)

Delete-Methode

Löscht die aktuelle Zeile.

Visual Basic-Syntax

```
Public Sub Delete()
```

C#-Syntax

```
public void Delete()
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULConnection.StartSynchronizationDelete-Methode \[UltraLite.NET\] auf Seite 155](#)
- [ULConnection.StopSynchronizationDelete-Methode \[UltraLite.NET\] auf Seite 155](#)

SetBoolean-Methode

Legt den Wert für die angegebene Spalte unter Verwendung eines `System.Boolean`-Objekts fest.

Visual Basic-Syntax

```
Public Sub SetBoolean(ByVal colID As Integer, ByVal val As Boolean)
```

C#-Syntax

```
public void SetBoolean(int colID, bool val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Daten in der Zeile werden erst geändert, wenn Sie eine ULTable.Insert- oder Update-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Boolean](#)

SetByte-Methode

Legt den Wert für die angegebene Spalte unter Verwendung eines System.Byte-Objekts (8-Bit-Ganzzahl ohne Vorzeichen) fest.

Visual Basic-Syntax

```
Public Sub SetByte(ByVal colID As Integer, ByVal val As Byte)
```

C#-Syntax

```
public void SetByte(int colID, byte val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Daten in der Zeile werden erst geändert, wenn Sie eine `ULTable.Insert-` oder `Update-Methode` ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Byte](#)

SetBytes-Methode

Legt den Wert für die angegebene Spalte unter Verwendung eines `System.Bytes-Arrays` fest.

Visual Basic-Syntax

```
Public Sub SetBytes(ByVal colID As Integer, ByVal val As Byte())
```

C#-Syntax

```
public void SetBytes(int colID, byte[] val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich `[0,ULDataReader.FieldCount-1]` liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Nur geeignet für Spalten vom Typ `ULDbType.Binary` oder `ULDbType.LongBinary`, oder für Spalten vom Typ `ULDbType.UniqueIdentifier`, wenn der Wert die Länge 16 hat. Die Daten in der Zeile werden erst geändert, wenn Sie eine `ULTable.Insert-` oder `Update-Methode` ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Byte](#)

SetDateTime-Methode

Legt den Wert für die angegebene Spalte mithilfe eines System.Date.Time-Objekts fest.

Visual Basic-Syntax

```
Public Sub SetDateTime(ByVal colID As Integer, ByVal val As Date)
```

C#-Syntax

```
public void SetDateTime(int colID, DateTime val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Der festgelegte Wert ist auf die Millisekunde genau. Die Daten in der Zeile werden erst geändert, wenn Sie eine ULTable.Insert- oder Update-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.DateTime](#)

SetDBNull-Methode

Setzt eine Spalte auf NULL.

Visual Basic-Syntax

```
Public Sub SetDBNull(ByVal colID As Integer)
```

C#-Syntax

```
public void SetDBNull(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Daten werden erst dann tatsächlich geändert, wenn Sie ULTable.Insert oder Update ausführen. Diese Änderung ist wiederum erst dauerhaft, wenn sie festgeschrieben wurde.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULTableSchema.IsColumnNullable-Methode \[UltraLite.NET\] auf Seite 447](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)

SetDecimal-Methode

Legt den Wert für die angegebene Spalte unter Verwendung eines System.Decimal-Objekts fest.

Visual Basic-Syntax

```
Public Sub SetDecimal(ByVal colID As Integer, ByVal val As Decimal)
```

C#-Syntax

```
public void SetDecimal(int colID, decimal val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Daten in der Zeile werden erst geändert, wenn Sie eine ULTable.Insert- oder Update-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Decimal](#)

SetDouble-Methode

Legt den Wert für die angegebene Spalte unter Verwendung eines System.Double-Objekts fest.

Visual Basic-Syntax

```
Public Sub SetDouble(ByVal colID As Integer, ByVal val As Double)
```

C#-Syntax

```
public void SetDouble(int colID, double val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Daten in der Zeile werden erst geändert, wenn Sie eine ULTable.Insert- oder Update-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Double](#)

SetFloat-Methode

Legt den Wert für die angegebene Spalte unter Verwendung eines System.Single-Objekts fest.

Visual Basic-Syntax

```
Public Sub SetFloat(ByVal colID As Integer, ByVal val As Single)
```

C#-Syntax

```
public void SetFloat(int colID, float val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Daten in der Zeile werden erst geändert, wenn Sie eine ULTable.Insert- oder Update-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Single](#)

SetGuid-Methode

Legt den Wert für die angegebene Spalte unter Verwendung eines System.Guid-Objekts fest.

Visual Basic-Syntax

```
Public Sub SetGuid(ByVal colID As Integer, ByVal val As Guid)
```

C#-Syntax

```
public void SetGuid(int colID, Guid val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Daten in der Zeile werden erst geändert, wenn Sie eine ULTable.Insert- oder Update-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird. Nur gültig für Spalten vom Typ ULDbType.UniqueIdentifier oder Spalten vom Typ ULDbType.Binary mit der Länge 16.

Siehe auch

- [ULConnection.GetNewUUID-Methode \[UltraLite.NET\] auf Seite 144](#)
- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULCursorSchema.GetColumnSize-Methode \[UltraLite.NET\] auf Seite 208](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Guid](#)

SetInt16-Methode

Legt den Wert für die angegebene Spalte unter Verwendung von System.Int16 fest.

Visual Basic-Syntax

```
Public Sub SetInt16(ByVal colID As Integer, ByVal val As Short)
```

C#-Syntax

```
public void SetInt16(int colID, short val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Daten in der Zeile werden erst geändert, wenn Sie eine ULTable.Insert- oder Update-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Int16](#)

SetInt32-Methode

Legt den Wert für die angegebene Spalte unter Verwendung eines System.Int32 fest.

Visual Basic-Syntax

```
Public Sub SetInt32(ByVal colID As Integer, ByVal val As Integer)
```

C#-Syntax

```
public void SetInt32(int colID, int val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Daten in der Zeile werden erst geändert, wenn Sie eine ULTable.Insert- oder Update-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Int32](#)

SetInt64-Methode

Setzt den Wert für die angegebene Spalte unter Verwendung eines Int64-Werts.

Visual Basic-Syntax

```
Public Sub SetInt64(ByVal colID As Integer, ByVal val As Long)
```

C#-Syntax

```
public void SetInt64(int colID, long val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Daten in der Zeile werden erst geändert, wenn Sie eine ULTable.Insert- oder Update-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.Int64](#)

SetString-Methode

Legt den Wert für die angegebene Spalte unter Verwendung eines System.String-Objekts fest.

Visual Basic-Syntax

```
Public Sub SetString(ByVal colID As Integer, ByVal val As String)
```

C#-Syntax

```
public void SetString(int colID, string val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Daten in der Zeile werden erst geändert, wenn Sie eine ULTable.Insert- oder Update-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)

SetTimeSpan-Methode

Legt den Wert für die angegebene Spalte unter Verwendung eines System.TimeSpan-Objekts fest.

Visual Basic-Syntax

```
Public Sub SetTimeSpan(ByVal colID As Integer, ByVal val As TimeSpan)
```

C#-Syntax

```
public void SetTimeSpan(int colID, TimeSpan val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Der festgelegte Wert ist auf die Millisekunde genau und wird auf einen nicht-negativen Wert zwischen 0 und 24 Stunden normalisiert. Die Daten in der Zeile werden erst geändert, wenn Sie eine ULTable.Insert- oder Update-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.TimeSpan](#)

SetToDefault-Methode

Setzt den Wert bei der angegebenen Spalte auf ihren Standardwert.

Visual Basic-Syntax

```
Public Sub SetToDefault(ByVal colID As Integer)
```

C#-Syntax

```
public void SetToDefault(int colID)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Daten in der Zeile werden erst geändert, wenn Sie eine ULTable.Insert- oder Update-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTableSchema.GetColumnDefaultValue-Methode \[UltraLite.NET\] auf Seite 439](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)

SetUInt16-Methode

Legt den Wert für die angegebene Spalte unter Verwendung eines System.UInt16-Objekts fest.

Visual Basic-Syntax

```
Public Sub SetUInt16(ByVal colID As Integer, ByVal val As UShort)
```

C#-Syntax

```
public void SetUInt16(int colID, ushort val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Daten in der Zeile werden erst geändert, wenn Sie eine ULTable.Insert- oder Update-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.UInt16](#)

SetUInt32-Methode

Legt den Wert für die angegebene Spalte unter Verwendung eines System.UInt32-Objekts fest.

Visual Basic-Syntax

```
Public Sub SetUInt32(ByVal colID As Integer, ByVal val As UInteger)
```

C#-Syntax

```
public void SetUInt32(int colID, uint val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Daten in der Zeile werden erst geändert, wenn Sie eine ULTable.Insert- oder Update-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.UInt32](#)

SetUInt64-Methode

Legt den Wert für die angegebene Spalte unter Verwendung eines System.UInt64-Objekts fest.

Visual Basic-Syntax

```
Public Sub SetUInt64(ByVal colID As Integer, ByVal val As ULong)
```

C#-Syntax

```
public void SetUInt64(int colID, ulong val)
```

Parameter

- **colID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULDataReader.FieldCount-1] liegen. Die erste Spalte im Cursor hat den ID-Wert Null.
- **val** Der Wert, der an den aktuellen neuen Wert für die Spalte angehängt wird.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die Daten in der Zeile werden erst geändert, wenn Sie eine ULTable.Insert- oder Update-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULDataReader.GetOrdinal-Methode \[UltraLite.NET\] auf Seite 256](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULTable.Insert-Methode \[UltraLite.NET\] auf Seite 431](#)
- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)
- [ULDataReader.FieldCount-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [System.UInt64](#)

Update-Methode

Aktualisiert die aktuelle Zeile mit den aktuellen Spaltenwerten (mit set-Methoden angegeben).

Visual Basic-Syntax

```
Public Sub Update()
```

C#-Syntax

```
public void Update()
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULResultSet.UpdateBegin-Methode \[UltraLite.NET\] auf Seite 374](#)

UpdateBegin-Methode

Bereitet die Aktualisierung der aktuellen Zeile vor.

Visual Basic-Syntax

```
Public Sub UpdateBegin()
```

C#-Syntax

```
public void UpdateBegin()
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Spaltenwerte werden durch den Aufruf der geeigneten setType- oder AppendType-Methoden geändert. Die erste Anhängung an eine Spalte löscht den aktuellen Spaltenwert, bevor der neue Wert angehängt wird.

Die Daten in der Zeile werden erst geändert, wenn Sie Update aufrufen. Die Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Das Ändern von Spalten in dem Index, der zum Öffnen der Tabelle verwendet wird, kann sich in unvorhersehbarer Weise auf aktive Suchvorgänge auswirken. Spalten im Primärschlüssel der Tabelle können nicht aktualisiert werden.

Siehe auch

- [ULResultSet.Update-Methode \[UltraLite.NET\] auf Seite 374](#)

ULResultSetSchema-Klasse

UL-Erw.: Stellt das Schema einer UltraLite-Ergebnismenge dar.

Visual Basic-Syntax

```
Public NotInheritable Class ULResultSetSchema Inherits ULCursorSchema
```

C#-Syntax

```
public sealed class ULResultSetSchema : ULCursorSchema
```

Basisklassen

- [ULCursorSchema-Klasse \[UltraLite.NET\] auf Seite 204](#)

Mitglieder

Alle Mitglieder der ULResultSetSchema-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
GetColumnID-Methode	Gibt die Spalten-ID der angegebenen Spalte zurück.
GetColumnName-Methode	Gibt den Namen der Spalte zurück, die von der Spalten-ID angegeben wird.
GetColumnPrecision-Methode	Gibt die Gesamtstellenzahl der Spalte zurück, die durch die angegebene Spalten-ID identifiziert wird, wenn die Spalte numerisch ist (SQL-Typ NUMERIC).
GetColumnScale-Methode	Gibt die Anzahl der Dezimalstellen der Spalte zurück, die durch die angegebene Spalten-ID identifiziert wird, wenn die Spalte numerisch ist (SQL-Typ NUMERIC).
GetColumnSize-Methode	Gibt die Größe der Spalte zurück, die durch die angegebene Spalten-ID identifiziert wird, wenn die Spalte eine Größenangabe hat (SQL-Typ BINARY oder CHAR).
GetColumnSQLName-Methode	Gibt den Namen der Spalte zurück, die von der Spalten-ID angegeben wird.
GetColumnULDbType-Methode	Gibt den UltraLite.NET-Datentyp der Spalte zurück, die von der angegebenen Spalten-ID identifiziert wird.
GetSchemaTable-Methode	Gibt ein System.Data.DataTable-Objekt zurück, das das Spaltenschema des ULDataReader-Objekts beschreibt.
ColumnCount-Eigenschaft	Gibt die Anzahl der Spalten im Cursor zurück.

Name	Beschreibung
IsOpen-Eigenschaft	Prüft, ob das Cursorschema aktuell geöffnet ist.
Name-Eigenschaft	Gibt den Namen des Cursors zurück.

Bemerkungen

Es gibt keinen Konstruktor für diese Klasse. Ein `ULResultSetSchema`-Objekt ist einer Ergebnismenge als deren `ULDataReader.Schema`-Eigenschaft zugeordnet.

Ein Ergebnismenge-Schema ist nur gültig, während der `DataReader` geöffnet ist.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULDataReader-Klasse \[UltraLite.NET\] auf Seite 237](#)
- [ULResultSetSchema-Klasse \[UltraLite.NET\] auf Seite 375](#)
- [ULDataReader.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 272](#)
- [ULCursorSchema-Klasse \[UltraLite.NET\] auf Seite 204](#)

Name-Eigenschaft

Gibt den Namen des Cursors zurück.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property Name As String
```

C#-Syntax

```
public override string Name {get;}
```

Bemerkungen

Die SQL-Anweisung, die das `ULResultSetSchema`-Objekt generiert hat

ULRowsCopiedEventArgs-Klasse

Stellt die Menge der Argumente dar, die an das `ULRowsCopiedEventHandler`-Objekt übergeben werden.

Visual Basic-Syntax

```
Public NotInheritable Class ULRowsCopiedEventArgs
```

C#-Syntax

```
public sealed class ULRowsCopiedEventArgs
```

Mitglieder

Alle Mitglieder der `ULRowsCopiedEventArgs`-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ULRowsCopiedEventArgs-Konstruktor	Erstellt eine neue Instanz des ULRowsCopiedEventArgs-Objekts.
Abort-Eigenschaft	Ruft einen Wert ab bzw. legt einen Wert fest, der angibt, ob der Massenexport/-import-Vorgang abgebrochen werden soll.
RowsCopied-Eigenschaft	Gibt die Anzahl der Zeilen zurück, die während des aktuellen Massenexport/-import-Vorgangs kopiert werden.

Bemerkungen

Die ULRowsCopiedEventArgs-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

Siehe auch

- [ULRowsCopiedEventHandler-Delegat \[UltraLite.NET\] auf Seite 455](#)

ULRowsCopiedEventArgs-Konstruktor

Erstellt eine neue Instanz des ULRowsCopiedEventArgs-Objekts.

Visual Basic-Syntax

```
Public Sub New(ByVal rowsCopied As Long)
```

C#-Syntax

```
public ULRowsCopiedEventArgs(long rowsCopied)
```

Parameter

- rowsCopied** Ein 64-Bit-Ganzzahlwert, der die Anzahl der Zeilen angibt, die während des aktuellen Massenexport/-imports von Daten kopiert werden.

Bemerkungen

Die ULRowsCopiedEventArgs-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

Abort-Eigenschaft

Ruft einen Wert ab bzw. legt einen Wert fest, der angibt, ob der Massenexport/-import-Vorgang abgebrochen werden soll.

Visual Basic-Syntax

```
Public Property Abort As Boolean
```

C#-Syntax

```
public bool Abort {get;set;}
```

Bemerkungen

Die ULRowsCopiedEventArgs-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

RowsCopied-Eigenschaft

Gibt die Anzahl der Zeilen zurück, die während des aktuellen Massenexport/-import-Vorgangs kopiert werden.

Visual Basic-Syntax

```
Public ReadOnly Property RowsCopied As Long
```

C#-Syntax

```
public long RowsCopied {get;}
```

Bemerkungen

Eine lange Ganzzahl, die die Anzahl der kopierten Zeilen repräsentiert.

Die ULRowsCopiedEventArgs-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

ULRowUpdatedEventArgs-Klasse

Stellt Daten für das ULDataAdapter.RowUpdated-Ereignis bereit.

Visual Basic-Syntax

```
Public NotInheritable Class ULRowUpdatedEventArgs
    Inherits System.Data.Common.RowUpdatedEventArgs
```

C#-Syntax

```
public sealed class ULRowUpdatedEventArgs :
    System.Data.Common.RowUpdatedEventArgs
```

Basisklassen

- [System.Data.Common.RowUpdatedEventArgs](#)

Mitglieder

Alle Mitglieder der ULRowUpdatedEventArgs-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ULRowUpdatedEventArgs-Konstruktor	Initialisiert eine neue Instanz der ULRowUpdatedEventArgs-Klasse.

Name	Beschreibung
CopyToRows-Methode (geerbt aus System.Data.Common.RowUpdatedEventArgs)	Kopiert Referenzen zu den geänderten Zeilen in das angegebene Array.
Command-Eigenschaft	Gibt das ULCommand -Objekt zurück, das ausgeführt wird, wenn die DbDataAdapter.Update -Methode aufgerufen wird.
Errors-Eigenschaft (geerbt aus System.Data.Common.RowUpdatedEventArgs)	Ruft alle Fehler ab, die beim Ausführen von System.Data.Common.RowUpdatedEventArgs.Command vom .NET Framework-Datenprovider generiert wurden.
RecordsAffected-Eigenschaft	Gibt die Anzahl der durch eine SQL-Anweisung geänderten, eingefügten oder gelöschten Zeilen zurück.
Row-Eigenschaft (geerbt aus System.Data.Common.RowUpdatedEventArgs)	Ruft das System.Data.DataRow -Objekt ab, das durch System.Data.Common.DbDataAdapter.Update(System.Data.DataSet) gesendet wurde.
RowCount-Eigenschaft (geerbt aus System.Data.Common.RowUpdatedEventArgs)	Ruft die Anzahl der Zeilen ab, die in einem Batch von aktualisierten Datensätzen abgearbeitet wurden.
StatementType-Eigenschaft (geerbt aus System.Data.Common.RowUpdatedEventArgs)	Ruft den Typ einer ausgeführten SQL-Anweisung ab.
Status-Eigenschaft (geerbt aus System.Data.Common.RowUpdatedEventArgs)	Ruft das System.Data.UpdateStatus -Objekt der System.Data.Common.RowUpdatedEventArgs.Command -Eigenschaft ab.
TableMapping-Eigenschaft (geerbt aus System.Data.Common.RowUpdatedEventArgs)	Ruft das System.Data.Common.DataTableMapping -Objekt ab, das durch System.Data.Common.DbDataAdapter.Update(System.Data.DataSet) gesendet wurde.

Siehe auch

- [ULDataAdapter.RowUpdated-Ereignis \[UltraLite.NET\]](#) auf Seite 222
- [System.Data.Common.RowUpdatedEventArgs](#)

ULRowUpdatedEventArgs-Konstruktor

Initialisiert eine neue Instanz der [ULRowUpdatedEventArgs](#)-Klasse.

Visual Basic-Syntax

```
Public Sub New(  
    ByVal row As DataRow,
```

```
        ByVal command As IDbCommand,  
        ByVal statementType As StatementType,  
        ByVal tableMapping As DataTableMapping  
    )
```

C#-Syntax

```
public ULRowUpdatedEventArgs(  
    DataRow row,  
    IDbCommand command,  
    StatementType statementType,  
    DataTableMapping tableMapping  
)
```

Parameter

- **row** Das System.Data.DataRow-Objekt, das durch einen DbDataAdapter.Update-Aufruf gesendet wird.
- **command** Das System.Data.IDbCommand-Objekt, das beim Aufrufen der DbDataAdapter.Update-Methode ausgeführt wird.
- **statementType** Einer der System.Data.StatementType-Werte, die den Typ der angegebenen Abfrage definieren
- **tableMapping** Das System.Data.Common.DataTableMapping-Objekt, das durch einen DbDataAdapter.Update-Aufruf gesendet wird.

Siehe auch

- [System.Data.DataRow](#)
- [System.Data.IDbCommand](#)
- [System.Data.StatementType](#)
- [System.Data.Common.DataTableMapping](#)

Command-Eigenschaft

Gibt das ULCommand-Objekt zurück, das ausgeführt wird, wenn die DbDataAdapter.Update-Methode aufgerufen wird.

Visual Basic-Syntax

```
Public ReadOnly Shadows Property Command As ULCommand
```

C#-Syntax

```
public new ULCommand Command {get;}
```

Bemerkungen

Das ULCommand-Objekt, das von der Aktualisierung ausgeführt wird

Dies ist die stark typisierte Version von System.Data.Common.RowUpdatedEventArgs.Command.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [System.Data.Common.RowUpdatedEventArgs.Command](#)

RecordsAffected-Eigenschaft

Gibt die Anzahl der durch eine SQL-Anweisung geänderten, eingefügten oder gelöschten Zeilen zurück.

Visual Basic-Syntax

```
Public ReadOnly Shadows Property RecordsAffected As Integer
```

C#-Syntax

```
public new int RecordsAffected {get;}
```

Bemerkungen

Für SELECT-Anweisungen ist dieser Wert -1.

Die Anzahl von Zeilen, die geändert, eingefügt oder gelöscht wurden, 0, wenn keine Zeilen betroffen waren oder die Anweisung fehlgeschlagen ist, und -1 bei SELECT-Anweisungen.

ULRowUpdatingEventArgs-Klasse

Stellt Daten für das ULDataAdapter.RowUpdating-Ereignis bereit.

Visual Basic-Syntax

```
Public NotInheritable Class ULRowUpdatingEventArgs  
    Inherits System.Data.Common.RowUpdatingEventArgs
```

C#-Syntax

```
public sealed class ULRowUpdatingEventArgs :  
    System.Data.Common.RowUpdatingEventArgs
```

Basisklassen

- [System.Data.Common.RowUpdatingEventArgs](#)

Mitglieder

Alle Mitglieder der ULRowUpdatingEventArgs-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ULRowUpdatingEventArgs-Konstruktor	Initialisiert eine neue Instanz der ULRowUpdatedEventArgs-Klasse.

Name	Beschreibung
BaseCommand-Eigenschaft (geerbt aus System.Data.Common.RowUpdatingEventArgs)	Ruft das System.Data.IDbCommand -Objekt für eine Instanz dieser Klasse ab bzw. legt es fest.
Command-Eigenschaft	Gibt das ULCommand-Objekt an, das ausgeführt werden soll, wenn die DbDataAdapter.Update-Methode ausgeführt wird.
Errors-Eigenschaft (geerbt aus System.Data.Common.RowUpdatingEventArgs)	Ruft alle vom .NET Framework-Datenprovider generierten Fehler ab, wenn System.Data.Common.RowUpdatedEventArgs.Command ausgeführt wird
Row-Eigenschaft (geerbt aus System.Data.Common.RowUpdatingEventArgs)	Ruft das System.Data.DataRow -Objekt ab, das als Teil eines Einfügings-, Aktualisierungs- oder Löschvorgangs an den Datenbankserver gesendet wird.
StatementType-Eigenschaft (geerbt aus System.Data.Common.RowUpdatingEventArgs)	Ruft den Typ einer auszuführenden SQL-Anweisung ab.
Status-Eigenschaft (geerbt aus System.Data.Common.RowUpdatingEventArgs)	Ruft das System.Data.UpdateStatus -Objekt der System.Data.Common.RowUpdatedEventArgs.Command -Eigenschaft ab bzw. legt es fest.
TableMapping-Eigenschaft (geerbt aus System.Data.Common.RowUpdatingEventArgs)	Ruft das System.Data.Common.DataTableMapping ausgeführt werden soll-Objekt ab, das über System.Data.Common.DbDataAdapter.Update(System.Data.DataSet) gesendet werden soll.

Siehe auch

- [ULDataAdapter.RowUpdating-Ereignis \[UltraLite.NET\]](#) auf Seite 222
- [System.Data.Common.RowUpdatingEventArgs](#)

ULRowUpdatingEventArgs-Konstruktor

Initialisiert eine neue Instanz der ULRowUpdatedEventArgs-Klasse.

Visual Basic-Syntax

```
Public Sub New(
    ByVal row As DataRow,
    ByVal command As IDbCommand,
    ByVal statementType As StatementType,
    ByVal tableMapping As DataTableMapping
)
```

C#-Syntax

```
public ULRowUpdatingEventArgs(  
    DataRow row,  
    IDbCommand command,  
    StatementType statementType,  
    DataTableMapping tableMapping  
)
```

Parameter

- **row** Das zu aktualisierende System.Data.DataRow-Objekt
- **command** Das System.Data.IDbCommand-Objekt, das bei der Aktualisierung ausgeführt werden soll
- **statementType** Einer der System.Data.StatementType-Werte, die den Typ der angegebenen Abfrage definieren
- **tableMapping** Der System.Data.Common.DataTableMapping-Wert, der durch einen DbDataAdapter.Update-Aufruf gesendet wird.

Siehe auch

- [System.Data.DataRow](#)
- [System.Data.IDbCommand](#)
- [System.Data.StatementType](#)
- [System.Data.Common.DataTableMapping](#)

Command-Eigenschaft

Gibt das ULCommand-Objekt an, das ausgeführt werden soll, wenn die DbDataAdapter.Update-Methode ausgeführt wird.

Visual Basic-Syntax

```
Public Shadows Property Command As ULCommand
```

C#-Syntax

```
public new ULCommand Command {get;set;}
```

Bemerkungen

Das ULCommand-Objekt, das beim Update ausgeführt werden soll

Dies ist die stark typisierte Version des System.Data.Common.RowUpdatingEventArgs.Command-Werts.

Siehe auch

- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [System.Data.Common.RowUpdatingEventArgs.Command](#)

ULServerSyncListener-Schnittstelle

UL-Erw.: Die Listener-Schnittstelle für den Empfang von Server-Synchronisationsnachrichten.

Visual Basic-Syntax

```
Public Interface ULServerSyncListener
```

C#-Syntax

```
public interface ULServerSyncListener
```

Mitglieder

Alle Mitglieder der ULServerSyncListener-Schnittstelle, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
ServerSyncInvoked-Methode	Wird aufgerufen, wenn der MobiLink-Listener für serverinitiierte Synchronisationen die Anwendung aufruft, um die Synchronisation auszuführen.

ServerSyncInvoked-Methode

Wird aufgerufen, wenn der MobiLink-Listener für serverinitiierte Synchronisationen die Anwendung aufruft, um die Synchronisation auszuführen.

Visual Basic-Syntax

```
Public Sub ServerSyncInvoked(ByVal messageName As String)
```

C#-Syntax

```
public void ServerSyncInvoked(string messageName)
```

Parameter

- **messageName** Der Name der Nachricht, die an die Anwendung gesendet wurde.

Bemerkungen

Diese Methode wird von einem eigenen Thread aufgerufen. Um Multi-Thread-Probleme zu vermeiden, sollte sie ein Ereignis an die Benutzerschnittstelle senden. Bei der Verwendung von Multi-Threading sollten Sie eine separate Verbindung sowie das Sperren-Schlüsselwort benutzen, um auf die Objekte zuzugreifen, die gemeinsam mit dem Rest der Anwendung verwendet werden.

Beispiel

```
Importiert iAnywhere.Data.UltraLite

Public Class MainWindow Inherits System.Windows.Forms.Form Implements ULServerSyncListener

Private conn As ULConnection

Public Sub New(ByVal args() As String) MyBase.New()
```

'This call is required by the Windows Form Designer. InitializeComponent()

```
'Add any initialization after the InitializeComponent() call
ULConnection.DatabaseManager.SetServerSyncListener( _ "myCompany.mymsg",
"myCompany.myapp", Me _ ) 'Create Connection ... End Sub
```

```
Protected Overrides Sub OnClosing( _ ByVal e As System.ComponentModel.CancelEventArgs _ )
ULConnection.DatabaseManager.SetServerSyncListener( _ Nothing, Nothing, Nothing _ )
 MyBase.OnClosing(e) End Sub
```

```
Public Sub ServerSyncInvoked(ByVal messageName As String) _ Implements
ULServerSyncListener.ServerSyncInvoked
```

```
Me.Invoke(New EventHandler(AddressOf Me.ServerSyncAction)) End Sub
```

```
Public Sub ServerSyncAction( _ ByVal sender As Object, ByVal e As EventArgs _ ) ' Do Server sync
conn.Synchronize() End Sub End Class
```

Der folgende C#-Code zeigt, wie eine Anforderung zur Serversynchronisation empfangen und eine Synchronisation auf dem UI-Thread ausgeführt wird.

```
using iAnywhere.Data.UltraLite;
public class Form1 : System.Windows.Forms.Form, ULServerSyncListener
{
    private System.Windows.Forms.MainMenu mainMenu1;
    private ULConnection conn;

    public Form1()
    {
        //
        // Required for Windows Form Designer support
        //
        InitializeComponent();

        //
        // TODO: Add any constructor code after
        // InitializeComponent call
        //
        ULConnection.DatabaseManager.SetServerSyncListener(
            "myCompany.mymsg", "myCompany.myapp", this
        );

        // Create connection
        ...
    }

    protected override void Dispose( bool disposing )
    {
        base.Dispose( disposing );
    }

    protected override void OnClosing(
        System.ComponentModel.CancelEventArgs e)
    {
        ULConnection.DatabaseManager.SetServerSyncListener(
            null, null, null
        );
        base.OnClosing(e);
    }
}
```

```
public void ServerSyncInvoked( string messageName )
{
    this.Invoke( new EventHandler( ServerSyncHandler ) );
}

internal void ServerSyncHandler(object sender, EventArgs e)
{
    conn.Synchronize();
}
}
```

ULSqlProgressData-Klasse

UL-Erw.: Gibt Daten zum Verarbeitungsfortschritt des SQL-Passthrough-Skripts zurück.

Visual Basic-Syntax

```
Public Class ULSqlProgressData
```

C#-Syntax

```
public class ULSqlProgressData
```

Mitglieder

Alle Mitglieder der ULSqlProgressData-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
CurrentScript-Eigenschaft	Der Index der bisher ausgeführten Skripten.
ScriptCount-Eigenschaft	Gibt die Anzahl der gerade ausgeführten Skripten zurück.
State-Eigenschaft	Gibt den aktuellen Status des Verarbeitungsfortschritts zurück.

CurrentScript-Eigenschaft

Der Index der bisher ausgeführten Skripten.

Visual Basic-Syntax

```
Public ReadOnly Property CurrentScript As Long
```

C#-Syntax

```
public long CurrentScript {get;}
```

Bemerkungen

Der aktuelle Index der gerade ausgeführten Skripten.

ScriptCount-Eigenschaft

Gibt die Anzahl der gerade ausgeführten Skripten zurück.

Visual Basic-Syntax

```
Public ReadOnly Property ScriptCount As Long
```

C#-Syntax

```
public long ScriptCount {get;}
```

Bemerkungen

Die Anzahl der gerade ausgeführten Skripten.

State-Eigenschaft

Gibt den aktuellen Status des Verarbeitungsfortschritts zurück.

Visual Basic-Syntax

```
Public ReadOnly Property State As ULSqlProgressState
```

C#-Syntax

```
public ULSqlProgressState State {get;}
```

Bemerkungen

Einer der ULSqlProgressState-Werte, der den aktuellen Zustand des SQL-Passthrough-Callbacks angibt.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

ULSyncParms-Klasse

UL-Erw.: Stellt Synchronisationsparameter dar, die festlegen, wie eine UltraLite-Datenbank zu synchronisieren ist.

Visual Basic-Syntax

```
Public NotInheritable Class ULSyncParms
```

C#-Syntax

```
public sealed class ULSyncParms
```

Mitglieder

Alle Mitglieder der ULSyncParms-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
CopyFrom-Methode	Kopiert die Eigenschaften des angegebenen ULSyncParms-Objekts in dieses ULSyncParms-Objekt.
ToString-Methode	Gibt die Zeichenfolge zurück, die diese Instanz repräsentiert.
AdditionalParms-Eigenschaft	Legt zusätzliche Synchronisationsparameter als semikolongetrennte Liste von Paaren der Form Name=Wert fest.
AuthenticationParms-Eigenschaft	Legt Parameter für ein angepasstes Benutzerauthentifizierungsskript (MobiLink authenticate_parameters-Verbindungsereignis) fest.
DownloadOnly-Eigenschaft	Legt fest, ob Uploads während der Synchronisation deaktiviert oder aktiviert sind.
KeepPartialDownload-Eigenschaft	Legt fest, ob Teil-Downloads während der Synchronisation aktiviert oder deaktiviert werden sollen.
NewPassword-Eigenschaft	Legt für den mit UserName angegebenen Benutzer ein neues MobiLink-Kennwort fest.
Password-Eigenschaft	Das MobiLink-Kennwort für den mit UserName angegebenen Benutzer.
PingOnly-Eigenschaft	Legt fest, ob der Client nur den MobiLink-Server anpingen soll, anstatt eine echte Synchronisation durchzuführen.
Publications-Eigenschaft	Legt die zu synchronisierenden Publikationen fest.
ResumePartialDownload-Eigenschaft	Legt fest, ob ein früherer Teil-Download wieder aufgenommen oder entfernt werden soll.
SendDownloadAck-Eigenschaft	Legt fest, ob der Client während der Synchronisation eine Downloadbestätigung an den MobiLink-Server senden soll.
Stream-Eigenschaft	Gibt den MobiLink-Synchronisationsdatenstrom für die Verwendung bei der Synchronisation an.
StreamParms-Eigenschaft	Gibt die Parameter zur Konfiguration des Synchronisationsdatenstroms an.
UploadOnly-Eigenschaft	Legt fest, ob Downloads während der Synchronisation deaktiviert oder aktiviert sind.
UserName-Eigenschaft	Der Benutzername, der den MobiLink-Client für den MobiLink-Server eindeutig kennzeichnet.
Version-Eigenschaft	Legt fest, welches Synchronisationsskript verwendet werden soll.

Bemerkungen

Es gibt keinen Konstruktor für diese Klasse. Jede Verbindung besitzt ihre eigene ULSyncParms-Instanz, die ihr als ULConnection.SyncParms-Eigenschaft zugeordnet ist.

Es kann jeweils nur ein Synchronisationsbefehl (die Eigenschaft ULSyncParms.DownloadOnly, ULSyncParms.PingOnly, ULSyncParms.ResumePartialDownload oder ULSyncParms.UploadOnly) angegeben werden. Wenn mehrere dieser Parameter auf TRUE gesetzt sind, gibt die ULConnection.Synchronize-Methode die SQLException-Ausnahmebedingung ULSQLCode.SQLE_SYNC_INFO_INVALID aus.

Andere Quellen von ULSQLCode.SQLE_SYNC_INFO_INVALID-Fehlern sind z.B. das fehlende Festlegen eines ULSyncParms.Stream- oder ULSyncParms.Version-Werts.

Siehe auch

- [ULConnection-Klasse \[UltraLite.NET\] auf Seite 121](#)
- [ULConnection.SyncParms-Eigenschaft \[UltraLite.NET\] auf Seite 165](#)
- [ULConnection.Synchronize-Methode \[UltraLite.NET\] auf Seite 156](#)
- [ULSyncParms.DownloadOnly-Eigenschaft \[UltraLite.NET\] auf Seite 391](#)
- [ULSyncParms.PingOnly-Eigenschaft \[UltraLite.NET\] auf Seite 394](#)
- [ULSyncParms.ResumePartialDownload-Eigenschaft \[UltraLite.NET\] auf Seite 395](#)
- [ULSyncParms.UploadOnly-Eigenschaft \[UltraLite.NET\] auf Seite 397](#)
- [ULConnection.Synchronize-Methode \[UltraLite.NET\] auf Seite 156](#)
- [ULSyncParms.Stream-Eigenschaft \[UltraLite.NET\] auf Seite 396](#)
- [ULSyncParms.Version-Eigenschaft \[UltraLite.NET\] auf Seite 399](#)

CopyFrom-Methode

Kopiert die Eigenschaften des angegebenen ULSyncParms-Objekts in dieses ULSyncParms-Objekt.

Visual Basic-Syntax

```
Public Sub CopyFrom(ByVal src As ULSyncParms)
```

C#-Syntax

```
public void CopyFrom(ULSyncParms src)
```

Parameter

- **src** Das Objekt, aus dem kopiert werden soll

Siehe auch

- [ULSyncParms-Klasse \[UltraLite.NET\] auf Seite 387](#)

ToString-Methode

Gibt die Zeichenfolge zurück, die diese Instanz repräsentiert.

Visual Basic-Syntax

```
Public Overrides Function ToString() As String
```

C#-Syntax

```
public override string ToString()
```

Rückgabe

Die Zeichenfolgendarstellung dieser Instanz ist eine durch Semikola getrennte Liste von Schlüsselwort=Wert-Paaren.

AdditionalParms-Eigenschaft

Legt zusätzliche Synchronisationsparameter als semikolongetrennte Liste von Paaren der Form Name=Wert fest.

Visual Basic-Syntax

```
Public Property AdditionalParms As String
```

C#-Syntax

```
public string AdditionalParms {get;set;}
```

Rückgabe

Eine Zeichenfolge in der Form einer durch Semikola getrennten Liste von Name=Wert-Paaren.

Bemerkungen

Verwenden Sie diese Eigenschaft, um mehrere zusätzliche Synchronisationsparameter anzugeben, die nicht ohne Weiteres unter Verwendung anderer vordefinierter Parameter angegeben werden können.

Siehe auch

- „Synchronisationsparameter Additional Parameters“ [[UltraLite - Datenbankverwaltung](#)]

Beispiel

```
private ULSyncParms info;  
// ...  
info.AdditionalParms =  
    "AllowDownloadDupRows=1;  
    CheckpointStore=1;  
    DisableConcurrency=1;  
    TableOrder=Customer,Sales"
```

AuthenticationParms-Eigenschaft

Legt Parameter für ein angepasstes Benutzerauthentifizierungsskript (MobiLink authenticate_parameters-Verbindungsereignis) fest.

Visual Basic-Syntax

```
Public Property AuthenticationParms As String()
```

C#-Syntax

```
public string[] AuthenticationParms {get;set;}
```

Rückgabe

Ein Array von Zeichenfolgen, die jeweils einen Authentifizierungsparameter enthalten (null Array-Einträge werden einen Synchronisationsfehler ergeben). Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic). Das bedeutet, dass keine Authentifizierungsparameter verwendet werden.

Bemerkungen

Nur die ersten 255 Zeichenfolgen werden verwendet und keine Zeichenfolge sollte die Grenze des MobiLink-Servers für Authentifizierungsparameter überschreiten (derzeit 4000 UTF8-Byte).

DownloadOnly-Eigenschaft

Legt fest, ob Uploads während der Synchronisation deaktiviert oder aktiviert sind.

Visual Basic-Syntax

```
Public Property DownloadOnly As Boolean
```

C#-Syntax

```
public bool DownloadOnly {get;set;}
```

Rückgabe

TRUE, um Uploads bei der Synchronisation zu deaktivieren, sonst FALSE. Der Standardwert ist FALSE.

Bemerkungen

Es kann jeweils nur ein Synchronisationsbefehl (die Eigenschaft ULSyncParms.DownloadOnly, ULSyncParms.PingOnly, ULSyncParms.ResumePartialDownload oder ULSyncParms.UploadOnly) angegeben werden. Wenn mehrere dieser Parameter auf TRUE gesetzt sind, gibt die ULConnection.Synchronize-Methode die SQLException-Ausnahmebedingung ULSQLCode.SQLE_SYNC_INFO_INVALID aus.

Siehe auch

- [ULSyncParms.UploadOnly-Eigenschaft \[UltraLite.NET\] auf Seite 397](#)
- [ULSyncParms.DownloadOnly-Eigenschaft \[UltraLite.NET\] auf Seite 391](#)
- [ULSyncParms.PingOnly-Eigenschaft \[UltraLite.NET\] auf Seite 394](#)
- [ULSyncParms.ResumePartialDownload-Eigenschaft \[UltraLite.NET\] auf Seite 395](#)
- [ULSyncParms.UploadOnly-Eigenschaft \[UltraLite.NET\] auf Seite 397](#)
- [ULConnection.Synchronize-Methode \[UltraLite.NET\] auf Seite 156](#)

KeepPartialDownload-Eigenschaft

Legt fest, ob Teil-Downloads während der Synchronisation aktiviert oder deaktiviert werden sollen.

Visual Basic-Syntax

```
Public Property KeepPartialDownload As Boolean
```

C#-Syntax

```
public bool KeepPartialDownload {get;set;}
```

Rückgabe

Setzen Sie dies auf TRUE, um Teil-Downloads während der Synchronisation zu aktivieren und zu speichern, bzw. auf FALSE, um Teil-Downloads zu deaktivieren und Downloads zurückzusetzen, wenn ein Fehler auftritt. Der Standardwert ist FALSE.

Bemerkungen

Mit dem ULSyncProgressListener-Objekt, kann UltraLite.NET Teil-Downloads wieder aufnehmen, die aufgrund von Kommunikationsfehlern oder eines Abbruchs durch den Benutzer abgebrochen wurden. UltraLite.NET verarbeitet den Download so, wie er empfangen wird. Wenn ein Download unterbrochen wird, bleibt die Teil-Download-Transaktion in der Datenbank und kann bei der nächsten Synchronisation wieder aufgenommen werden.

Falls ein Teil-Download bewahrt wurde, wird die ULConnection.ULSyncResult.PartialDownloadRetained-Eigenschaft auf TRUE gesetzt, wenn die ULConnection.Synchronize-Methode vorhanden ist.

Wenn die PartialDownloadRetained-Eigenschaft gesetzt ist, können Sie einen Download wiederaufnehmen. Um das zu tun, rufen Sie die ULConnection.Synchronize-Methode mit der ULConnection.ULSyncParms.ResumePartialDownload-Eigenschaft auf, die auf TRUE gesetzt sein muss. Es wird empfohlen, den Wert TRUE für die KeepPartialDownload-Eigenschaft beizubehalten, auch für den Fall anderer Kommunikationsfehler. Wird ein Upload übersprungen, so wird kein Download ausgeführt.

Der Download, den Sie während eines wiederaufgenommenen Downloads empfangen, ist so alt wie der Download, der ursprünglich begonnen wurde. Wenn Sie die neuesten Daten benötigen, können Sie unmittelbar nach Abschluss des wiederaufgenommenen Downloads einen weiteren Download ausführen.

Bei der Wiederaufnahme eines Downloads sind einige der ULSyncParms-Eigenschaften ohne Bedeutung. Die Publications-Eigenschaft wird z.B. nicht verwendet. Sie erhalten die Publikationen, die Sie beim ursprünglichen Download angefordert haben. Die einzigen Eigenschaften, die festgelegt werden müssen, sind ResumePartialDownload und UserName. Die KeepPartialDownload-Eigenschaft kann auf Wunsch gesetzt werden und normal funktionieren.

Wenn ein Teil-Download existiert und nicht mehr benötigt wird, rufen Sie die ULConnection.RollbackPartialDownload-Methode auf, um die fehlgeschlagene Download-Transaktion zurückzusetzen. Wenn Sie versuchen, erneut zu synchronisieren, ohne die ResumePartialDownload-Eigenschaft festzulegen, wird der Teil-Download zurückgesetzt, bevor die nächste Synchronisation beginnt.

Siehe auch

- [ULSyncResult.PartialDownloadRetained-Eigenschaft \[UltraLite.NET\] auf Seite 412](#)
- [ULSyncParms.ResumePartialDownload-Eigenschaft \[UltraLite.NET\] auf Seite 395](#)
- [ULConnection.RollbackPartialDownload-Methode \[UltraLite.NET\] auf Seite 153](#)
- [ULSyncParms.ResumePartialDownload-Eigenschaft \[UltraLite.NET\] auf Seite 395](#)
- [ULSyncParms.UserName-Eigenschaft \[UltraLite.NET\] auf Seite 398](#)
- [ULConnection.RollbackPartialDownload-Methode \[UltraLite.NET\] auf Seite 153](#)
- „Wiederaufnahme fehlgeschlagener Downloads“ [*MobiLink - Serveradministration*]

NewPassword-Eigenschaft

Legt für den mit UserName angegebenen Benutzer ein neues MobiLink-Kennwort fest.

Visual Basic-Syntax

```
Public Property NewPassword As String
```

C#-Syntax

```
public string NewPassword {get;set;}
```

Rückgabe

Eine Zeichenfolge, die ein neues MobiLink-Kennwort festlegt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic). Das bedeutet, dass das Kennwort nicht geändert wurde.

Bemerkungen

Ein neues Kennwort wird nach der nächsten Synchronisation wirksam.

Siehe auch

- [ULSyncParms.UserName-Eigenschaft \[UltraLite.NET\] auf Seite 398](#)

Password-Eigenschaft

Das MobiLink-Kennwort für den mit UserName angegebenen Benutzer.

Visual Basic-Syntax

```
Public Property Password As String
```

C#-Syntax

```
public string Password {get;set;}
```

Rückgabe

Eine Zeichenfolge, die das MobiLink-Kennwort festlegt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic). Das bedeutet, dass kein Kennwort angegeben wurde.

Bemerkungen

Der MobiLink-Benutzername und das Kennwort unterscheiden sich von der ID und dem Kennwort des Datenbankbenutzers und dienen der Kennzeichnung und Authentifizierung der Anwendung beim MobiLink-Server.

Siehe auch

- [ULSyncParms.NewPassword-Eigenschaft \[UltraLite.NET\] auf Seite 393](#)
- [ULSyncParms.UserName-Eigenschaft \[UltraLite.NET\] auf Seite 398](#)

PingOnly-Eigenschaft

Legt fest, ob der Client nur den MobiLink-Server anpingen soll, anstatt eine echte Synchronisation durchzuführen.

Visual Basic-Syntax

```
Public Property PingOnly As Boolean
```

C#-Syntax

```
public bool PingOnly {get;set;}
```

Rückgabe

TRUE, wenn der Client den MobiLink-Server nur pingen soll. FALSE, wenn der Client eine echte Synchronisation ausführen soll. Der Standardwert ist FALSE.

Bemerkungen

Es kann jeweils nur ein Synchronisationsbefehl (die Eigenschaft ULSyncParms.DownloadOnly, ULSyncParms.PingOnly, ULSyncParms.ResumePartialDownload oder ULSyncParms.UploadOnly) angegeben werden. Wenn mehrere dieser Parameter auf TRUE gesetzt sind, gibt die ULConnection.Synchronize-Methode die SQLException-Ausnahmebedingung ULSQLCode.SQLE_SYNC_INFO_INVALID aus.

Siehe auch

- [ULSyncParms.DownloadOnly-Eigenschaft \[UltraLite.NET\] auf Seite 391](#)
- [ULSyncParms.PingOnly-Eigenschaft \[UltraLite.NET\] auf Seite 394](#)
- [ULSyncParms.ResumePartialDownload-Eigenschaft \[UltraLite.NET\] auf Seite 395](#)
- [ULSyncParms.UploadOnly-Eigenschaft \[UltraLite.NET\] auf Seite 397](#)
- [ULConnection.Synchronize-Methode \[UltraLite.NET\] auf Seite 156](#)

Publications-Eigenschaft

Legt die zu synchronisierenden Publikationen fest.

Visual Basic-Syntax

```
Public Property Publications As String
```

C#-Syntax

```
public string Publications {get;set;}
```

Rückgabe

Eine Zeichenfolge mit einer Liste von Publikationsnamen, getrennt durch Kommas (,) oder der Spezialwert ULConnection.SYNC_ALL_PUBS oder der Spezialwert ULConnection.SYNC_ALL_DB. Der Standardwert ist ULConnection.SYNC_ALL_DB.

Siehe auch

- [ULConnection.SYNC_ALL_PUBS-Feld \[UltraLite.NET\] auf Seite 169](#)
- [ULConnection.SYNC_ALL_DB-Feld \[UltraLite.NET\] auf Seite 168](#)

ResumePartialDownload-Eigenschaft

Legt fest, ob ein früherer Teil-Download wieder aufgenommen oder entfernt werden soll.

Visual Basic-Syntax

```
Public Property ResumePartialDownload As Boolean
```

C#-Syntax

```
public bool ResumePartialDownload {get;set;}
```

Rückgabe

TRUE, wenn ein früherer Teil-Download wiederaufgenommen werden soll. FALSE, wenn er entfernt werden soll. Der Standardwert ist FALSE.

Bemerkungen

Es kann jeweils nur ein Synchronisationsbefehl (die Eigenschaft ULSyncParms.DownloadOnly, ULSyncParms.PingOnly, ULSyncParms.ResumePartialDownload oder ULSyncParms.UploadOnly) angegeben werden. Wenn mehrere dieser Parameter auf TRUE gesetzt sind, gibt die ULConnection.Synchronize-Methode die SQLException-Ausnahmebedingung ULSQLCode.SQLE_SYNC_INFO_INVALID aus.

Siehe auch

- [ULSyncParms.KeepPartialDownload-Eigenschaft \[UltraLite.NET\] auf Seite 392](#)
- [ULSyncParms.DownloadOnly-Eigenschaft \[UltraLite.NET\] auf Seite 391](#)
- [ULSyncParms.PingOnly-Eigenschaft \[UltraLite.NET\] auf Seite 394](#)
- [ULSyncParms.ResumePartialDownload-Eigenschaft \[UltraLite.NET\] auf Seite 395](#)
- [ULSyncParms.UploadOnly-Eigenschaft \[UltraLite.NET\] auf Seite 397](#)
- [ULConnection.Synchronize-Methode \[UltraLite.NET\] auf Seite 156](#)
- [ULSyncResult.PartialDownloadRetained-Eigenschaft \[UltraLite.NET\] auf Seite 412](#)

SendDownloadAck-Eigenschaft

Legt fest, ob der Client während der Synchronisation eine Downloadbestätigung an den MobiLink-Server senden soll.

Visual Basic-Syntax

```
Public Property SendDownloadAck As Boolean
```

C#-Syntax

```
public bool SendDownloadAck {get;set;}
```

Rückgabe

Setzen Sie dies auf TRUE, um anzugeben, dass der Client eine Downloadbestätigung an den MobiLink-Server senden soll. Setzen Sie dies auf FALSE, um anzugeben, dass keine Downloadbestätigung gesendet wird. Der Standardwert ist FALSE.

Bemerkungen

Die Downloadbestätigung wird gesendet, nachdem der Download auf dem entfernten Server voll angewendet und festgeschrieben wurde (eine positive Bestätigung) oder nachdem der Download fehlgeschlagen ist (eine negative Bestätigung).

Wenn der Client eine Downloadbestätigung sendet, muss der Worker-Thread der MobiLink-Serverdatenbank darauf warten, dass der Client den Download durchführt und festschreibt. Wenn der Client keine Downloadbestätigung sendet, ist der MobiLink-Server schneller für seine nächste Synchronisation verfügbar.

Stream-Eigenschaft

Gibt den MobiLink-Synchronisationsdatenstrom für die Verwendung bei der Synchronisation an.

Visual Basic-Syntax

```
Public Property Stream As ULStreamType
```

C#-Syntax

```
public ULStreamType Stream {get;set;}
```

Rückgabe

Einer der ULStreamType-Werte, die den Typ des zu verwendenden Synchronisationsdatenstroms festlegen. Der Standardwert ist ULStreamType.TCPIP.

Bemerkungen

Die meisten Synchronisationsdatenströme benötigen Parameter, um die Adresse des MobiLink-Servers zu bestimmen und anderes Verhalten zu steuern. Diese Parameter werden von der ULSyncParms.StreamParms-Eigenschaft bereitgestellt.

Wenn der Datenstromtyp auf einen für die Plattform unzulässigen Wert gesetzt wird, wird der Datenstromparameter auf ULStreamType.TCPIP gesetzt.

Siehe auch

- [ULStreamType-Enumeration \[UltraLite.NET\] auf Seite 464](#)
- [ULSyncParms.StreamParms-Eigenschaft \[UltraLite.NET\] auf Seite 397](#)

StreamParms-Eigenschaft

Gibt die Parameter zur Konfiguration des Synchronisationsdatenstroms an.

Visual Basic-Syntax

```
Public Property StreamParms As String
```

C#-Syntax

```
public string StreamParms {get;set;}
```

Rückgabe

Eine Zeichenfolge in der Form einer durch Semikola getrennten Liste von Schlüsselwort-Wert-Paaren, die die Parameter für den Datenstrom festlegt. Der Standardwert ist eine Nullreferenz ("Nothing" in Visual Basic).

Bemerkungen

StreamParms ist eine Zeichenfolge, die alle Parameter enthält, die für Synchronisationsdatenströme verwendet werden. Parameter werden in einer durch Semikola getrennten Liste mit Paaren der Form Name=Wert ("Param1=Wert1;Param2=Wert2") angegeben.

Siehe auch

- [ULSyncParms.Stream-Eigenschaft \[UltraLite.NET\] auf Seite 396](#)
- [ULStreamType-Enumeration \[UltraLite.NET\] auf Seite 464](#)
- „UltraLite-Netzwerkprotokolloptionen für dbmlsync“ [*UltraLite - Datenbankverwaltung*]

UploadOnly-Eigenschaft

Legt fest, ob Downloads während der Synchronisation deaktiviert oder aktiviert sind.

Visual Basic-Syntax

```
Public Property UploadOnly As Boolean
```

C#-Syntax

```
public bool UploadOnly {get;set;}
```

Rückgabe

TRUE, um Downloads bei der Synchronisation zu deaktivieren, sonst FALSE. Der Standardwert ist FALSE.

Bemerkungen

Es kann jeweils nur ein Synchronisationsbefehl (die Eigenschaft `ULSyncParms.DownloadOnly`, `ULSyncParms.PingOnly`, `ULSyncParms.ResumePartialDownload` oder `ULSyncParms.UploadOnly`) angegeben werden. Wenn mehrere dieser Parameter auf TRUE gesetzt sind, gibt die `ULConnection.Synchronize`-Methode die `SQLException`-Ausnahmebedingung `ULSQLCode.SQLE_SYNC_INFO_INVALID` aus.

Siehe auch

- [ULSyncParms.DownloadOnly-Eigenschaft \[UltraLite.NET\] auf Seite 391](#)
- [ULSyncParms.PingOnly-Eigenschaft \[UltraLite.NET\] auf Seite 394](#)
- [ULSyncParms.ResumePartialDownload-Eigenschaft \[UltraLite.NET\] auf Seite 395](#)
- [ULSyncParms.UploadOnly-Eigenschaft \[UltraLite.NET\] auf Seite 397](#)
- [ULConnection.Synchronize-Methode \[UltraLite.NET\] auf Seite 156](#)

UserName-Eigenschaft

Der Benutzername, der den MobiLink-Client für den MobiLink-Server eindeutig kennzeichnet.

Visual Basic-Syntax

```
Public Property UserName As String
```

C#-Syntax

```
public string UserName {get;set;}
```

Rückgabe

Eine Zeichenfolge, die den Benutzernamen festlegt. Dieser Parameter hat keinen Standardwert und muss explizit festgelegt werden.

Bemerkungen

Der MobiLink-Server verwendet diesen Wert, um den herunterzuladenden Inhalt zu bestimmen, den Synchronisationsstatus aufzuzeichnen und den normalen Betrieb nach Unterbrechungen während der Synchronisation wieder aufzunehmen. Der Benutzername und das Kennwort unterscheiden sich von der ID und dem Kennwort des Datenbankbenutzers und dienen der Kennzeichnung und Authentifizierung der Anwendung beim MobiLink-Server.

Siehe auch

- [ULSyncParms.Password-Eigenschaft \[UltraLite.NET\] auf Seite 393](#)

Version-Eigenschaft

Legt fest, welches Synchronisationsskript verwendet werden soll.

Visual Basic-Syntax

```
Public Property Version As String
```

C#-Syntax

```
public string Version {get;set;}
```

Rückgabe

Eine Zeichenfolge, die die Version des zu verwendenden Synchronisationsskripts angibt. Dieser Parameter hat keinen Standardwert und muss explizit festgelegt werden.

Bemerkungen

Jedes Synchronisationsskript in der konsolidierten Datenbank wird mit einer Versionszeichenfolge markiert. Es können z.B. zwei verschiedene download_cursor-Skripten vorhanden sein, die jeweils durch eine unterschiedliche Versionszeichenfolge gekennzeichnet werden. Die Versionszeichenfolge gestattet es einer UltraLite-Anwendung, zwischen einer Reihe von Synchronisationsskripten zu wählen.

ULSyncProgressData-Klasse

UL-Erw.: Gibt die Überwachungsdaten zum Synchronisationsfortschritt zurück.

Visual Basic-Syntax

```
Public Class ULSyncProgressData
```

C#-Syntax

```
public class ULSyncProgressData
```

Mitglieder

Alle Mitglieder der Klasse ULSyncProgressData-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
CurrentDownloadRowCount-Eigenschaft	Die Anzahl der bis dahin heruntergeladenen Zeilen.
Flags-Eigenschaft	Gibt die aktuellen Optionen für die Synchronisation zurück, die weitere Informationen über den aktuellen Status liefern.
IgnoredDeletes-Eigenschaft	Gibt die Anzahl der bisher empfangenen Zeilen zurück, die bereits gelöscht wurden.

Name	Beschreibung
IgnoredUpdates-Eigenschaft	Gibt die Anzahl der bisher empfangenen Zeilen zurück, die bereits aktualisiert wurden.
IsFinalSyncProgress-Eigenschaft	Gibt TRUE zurück, wenn dies die letzte Meldung zum Synchronisationsfortschritt ist.
ReceivedBytes-Eigenschaft	Gibt die Anzahl der bislang empfangenen Bytes zurück.
ReceivedDeletes-Eigenschaft	Gibt die Anzahl der gelöschten Zeilen zurück, die bislang empfangen wurden.
ReceivedInserts-Eigenschaft	Gibt die Anzahl der eingefügten Zeilen zurück, die bislang empfangen wurden.
ReceivedUpdates-Eigenschaft	Gibt die Anzahl der aktualisierten Zeilen zurück, die bislang empfangen wurden.
SentBytes-Eigenschaft	Gibt die Anzahl der bislang gesendeten Bytes zurück.
SentDeletes-Eigenschaft	Gibt die Anzahl der gelöschten Zeilen zurück, die bislang gesendet wurden.
SentInserts-Eigenschaft	Gibt die Anzahl der eingefügten Zeilen zurück, die bislang gesendet wurden.
SentUpdates-Eigenschaft	Gibt die Anzahl der aktualisierten Zeilen zurück, die bislang gesendet wurden.
State-Eigenschaft	Gibt den aktuellen Synchronisationsstatus zurück.
SyncTableCount-Eigenschaft	Gibt die Anzahl der Tabellen an, die synchronisiert werden.
SyncTableIndex-Eigenschaft	Gibt den Index der Tabelle zurück, die aktuell synchronisiert wird, im Bereich von 1 bis zur Gesamtzahl der von der Synchronisation betroffenen Tabellen.
TableID-Eigenschaft	Gibt den Datenbankindex der Tabelle zurück, die gerade synchronisiert wird.
TableName-Eigenschaft	Gibt den Namen der gerade im Upload oder Download befindlichen Tabelle zurück.
TotalDownloadRowCount-Eigenschaft	Gibt die Gesamtzahl der in dem Download zu empfangenden Zeilen zurück.
TruncateDeletes-Eigenschaft	Gibt die Anzahl der Zeilen zurück, die von einem Kürzungsvorgang gelöscht wurden.

Name	Beschreibung
FLAG_IS_BLOCKING-Feld	Eine Option, die anzeigt, dass die Synchronisation blockiert ist, während auf eine Antwort vom MobiLink-Server gewartet wird.

Siehe auch

- [ULSyncProgressListener-Schnittstelle \[UltraLite.NET\] auf Seite 409](#)

CurrentDownloadRowCount-Eigenschaft

Die Anzahl der bisher heruntergeladenen Zeilen.

Visual Basic-Syntax

```
Public ReadOnly Property CurrentDownloadRowCount As Integer
```

C#-Syntax

```
public int CurrentDownloadRowCount {get;}
```

Rückgabe

Die Anzahl der bisher heruntergeladenen Zeilen.

Bemerkungen

Diese Anzahl enthält doppelte Zeilen, bei denen es sich nicht um in ReceivedInserts, ReceivedUpdates oder ReceivedDeletes handelt.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

Flags-Eigenschaft

Gibt die aktuellen Optionen für die Synchronisation zurück, die weitere Informationen über den aktuellen Status liefern.

Visual Basic-Syntax

```
Public ReadOnly Property Flags As Integer
```

C#-Syntax

```
public int Flags {get;}
```

Rückgabe

Eine Ganzzahl, die eine durch OR verknüpfte Kombination von Optionen enthält.

Siehe auch

- [ULSyncProgressData.FLAG_IS_BLOCKING-Feld \[UltraLite.NET\] auf Seite 409](#)

IgnoredDeletes-Eigenschaft

Gibt die Anzahl der bisher empfangenen Zeilen zurück, die bereits gelöscht wurden.

Visual Basic-Syntax

```
Public ReadOnly Property IgnoredDeletes As Integer
```

C#-Syntax

```
public int IgnoredDeletes {get;}
```

Rückgabe

Gibt die Anzahl der bisher empfangenen Zeilen zurück, die bereits gelöscht wurden.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

IgnoredUpdates-Eigenschaft

Gibt die Anzahl der bisher empfangenen Zeilen zurück, die bereits aktualisiert wurden.

Visual Basic-Syntax

```
Public ReadOnly Property IgnoredUpdates As Integer
```

C#-Syntax

```
public int IgnoredUpdates {get;}
```

Rückgabe

Gibt die Anzahl der bisher empfangenen Zeilen zurück, die bereits aktualisiert wurden.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

IsFinalSyncProgress-Eigenschaft

Gibt TRUE zurück, wenn dies die letzte Meldung zum Synchronisationsfortschritt ist.

Visual Basic-Syntax

```
Public ReadOnly Property IsFinalSyncProgress As Boolean
```

C#-Syntax

```
public bool IsFinalSyncProgress {get;}
```

Rückgabe

TRUE, wenn dies die letzte Meldung zum Synchronisationsfortschritt ist.

ReceivedBytes-Eigenschaft

Gibt die Anzahl der bislang empfangenen Bytes zurück.

Visual Basic-Syntax

```
Public ReadOnly Property ReceivedBytes As Long
```

C#-Syntax

```
public long ReceivedBytes {get;}
```

Rückgabe

Die Anzahl der bislang empfangen Bytes.

Bemerkungen

Diese Information wird für alle Statuswerte aktualisiert.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

ReceivedDeletes-Eigenschaft

Gibt die Anzahl der gelöschten Zeilen zurück, die bislang empfangen wurden.

Visual Basic-Syntax

```
Public ReadOnly Property ReceivedDeletes As Integer
```

C#-Syntax

```
public int ReceivedDeletes {get;}
```

Rückgabe

Die Anzahl der bislang empfangenen gelöschten Zeilen

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

ReceivedInserts-Eigenschaft

Gibt die Anzahl der eingefügten Zeilen zurück, die bislang empfangen wurden.

Visual Basic-Syntax

```
Public ReadOnly Property ReceivedInserts As Integer
```

C#-Syntax

```
public int ReceivedInserts {get;}
```

Rückgabe

Die Anzahl der bislang empfangenen eingefügten Zeilen.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

ReceivedUpdates-Eigenschaft

Gibt die Anzahl der aktualisierten Zeilen zurück, die bislang empfangen wurden.

Visual Basic-Syntax

```
Public ReadOnly Property ReceivedUpdates As Integer
```

C#-Syntax

```
public int ReceivedUpdates {get;}
```

Rückgabe

Die Anzahl der bislang empfangenen aktualisierten Zeilen.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

SentBytes-Eigenschaft

Gibt die Anzahl der bislang gesendeten Bytes zurück.

Visual Basic-Syntax

```
Public ReadOnly Property SentBytes As Long
```

C#-Syntax

```
public long SentBytes {get;}
```

Rückgabe

Die Anzahl der bislang gesendeten Bytes.

Bemerkungen

Diese Information wird für alle Statuswerte aktualisiert.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

SentDeletes-Eigenschaft

Gibt die Anzahl der gelöschten Zeilen zurück, die bislang gesendet wurden.

Visual Basic-Syntax

```
Public ReadOnly Property SentDeletes As Integer
```

C#-Syntax

```
public int SentDeletes {get;}
```

Rückgabe

Die Anzahl der bislang gesendeten gelöschten Zeilen.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

SentInserts-Eigenschaft

Gibt die Anzahl der eingefügten Zeilen zurück, die bislang gesendet wurden.

Visual Basic-Syntax

```
Public ReadOnly Property SentInserts As Integer
```

C#-Syntax

```
public int SentInserts {get;}
```

Rückgabe

Die Anzahl der bislang gesendeten eingefügten Zeilen.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

SentUpdates-Eigenschaft

Gibt die Anzahl der aktualisierten Zeilen zurück, die bislang gesendet wurden.

Visual Basic-Syntax

```
Public ReadOnly Property SentUpdates As Integer
```

C#-Syntax

```
public int SentUpdates {get;}
```

Rückgabe

Die Anzahl der bislang gesendeten aktualisierten Zeilen.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

State-Eigenschaft

Gibt den aktuellen Synchronisationsstatus zurück.

Visual Basic-Syntax

```
Public ReadOnly Property State As ULSyncProgressState
```

C#-Syntax

```
public ULSyncProgressState State {get;}
```

Rückgabe

Eine ULSyncProgressState, die den aktuellen Synchronisationsstatus festlegt.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

SyncTableCount-Eigenschaft

Gibt die Anzahl der Tabellen an, die synchronisiert werden.

Visual Basic-Syntax

```
Public ReadOnly Property SyncTableCount As Integer
```

C#-Syntax

```
public int SyncTableCount {get;}
```

Rückgabe

Die Anzahl der Tabellen, die synchronisiert werden. Für jede Tabelle gibt es eine Sende- und Empfangsphase, daher kann diese Zahl höher sein als die Anzahl der synchronisierten Tabellen.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

SyncTableIndex-Eigenschaft

Gibt den Index der Tabelle zurück, die aktuell synchronisiert wird, im Bereich von 1 bis zur Gesamtzahl der von der Synchronisation betroffenen Tabellen.

Visual Basic-Syntax

```
Public ReadOnly Property SyncTableIndex As Integer
```

C#-Syntax

```
public int SyncTableIndex {get;}
```

Rückgabe

Der Index der Tabelle, die gerade synchronisiert wird, im Bereich von 1 bis zum Wert der SyncTableCount-Eigenschaft.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

TableID-Eigenschaft

Gibt den Datenbankindex der Tabelle zurück, die gerade synchronisiert wird.

Visual Basic-Syntax

```
Public ReadOnly Property TableID As Integer
```

C#-Syntax

```
public int TableID {get;}
```

Rückgabe

Der Datenbankindex im Bereich von 1 bis zum Wert der ULDatabaseSchema.TableCount-Eigenschaft.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)
- [ULDatabaseSchema.TableCount-Eigenschaft \[UltraLite.NET\] auf Seite 237](#)

TableName-Eigenschaft

Gibt den Namen der gerade im Upload oder Download befindlichen Tabelle zurück.

Visual Basic-Syntax

```
Public ReadOnly Property TableName As String
```

C#-Syntax

```
public string TableName {get;}
```

Rückgabe

Der Name der aktuell synchronisierten Tabelle; NULL, wenn nicht anwendbar.

TotalDownloadRowCount-Eigenschaft

Gibt die Gesamtzahl der in dem Download zu empfangenden Zeilen zurück.

Visual Basic-Syntax

```
Public ReadOnly Property TotalDownloadRowCount As Integer
```

C#-Syntax

```
public int TotalDownloadRowCount {get;}
```

Rückgabe

Die Anzahl der in dem Download zu empfangenden Zeilen.

Bemerkungen

Diese Anzahl enthält doppelte Zeilen, bei denen es sich nicht um in ReceivedInserts, ReceivedUpdates oder ReceivedDeletes handelt. Dieser Wert ist nicht gesetzt, bis die Synchronisation für die erste Tabelle in den STATE_RECEIVING_TABLE-Status geht.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

TruncateDeletes-Eigenschaft

Gibt die Anzahl der Zeilen zurück, die von einem Kürzungsvorgang gelöscht wurden.

Visual Basic-Syntax

```
Public ReadOnly Property TruncateDeletes As Integer
```

C#-Syntax

```
public int TruncateDeletes {get;}
```

Rückgabe

Die Anzahl der durch einen Kürzungsvorgang gelöschten Zeilen.

Siehe auch

- [ULSyncProgressState-Enumeration \[UltraLite.NET\] auf Seite 465](#)

FLAG_IS_BLOCKING-Feld

Eine Option, die anzeigt, dass die Synchronisation blockiert ist, während auf eine Antwort vom MobiLink-Server gewartet wird.

Visual Basic-Syntax

```
Public Const FLAG_IS_BLOCKING As Integer
```

C#-Syntax

```
public const int FLAG_IS_BLOCKING;
```

ULSyncProgressListener-Schnittstelle

UL-Erw.: Die Listener-Schnittstelle für den Empfang von Synchronisationsfortschritt-Ereignissen.

Visual Basic-Syntax

```
Public Interface ULSyncProgressListener
```

C#-Syntax

```
public interface ULSyncProgressListener
```

Mitglieder

Alle Mitglieder der ULSyncProgressListener-Schnittstelle, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
SyncProgressed-Methode	Wird während der Synchronisation aufgerufen, um den Benutzer über den Fortschritt zu informieren.

Siehe auch

- [ULConnection.Synchronize-Methode \[UltraLite.NET\] auf Seite 156](#)

SyncProgressed-Methode

Wird während der Synchronisation aufgerufen, um den Benutzer über den Fortschritt zu informieren.

Visual Basic-Syntax

```
Public Function SyncProgressed(  
    ByVal data As ULSyncProgressData  
) As Boolean
```

C#-Syntax

```
public bool SyncProgressed(ULSyncProgressData data)
```

Parameter

- **data** Ein ULSyncProgressData-Objekt, das die neuesten Fortschrittsdaten zur Synchronisation enthält.

Rückgabe

Diese Methode muss TRUE zurückgeben, um die Synchronisation abubrechen, oder FALSE, um fortzufahren.

Bemerkungen

Diese Methode muss TRUE zurückgeben, um die Synchronisation abubrechen, oder FALSE, um fortzufahren.

Während eines SyncProgressed-Aufrufs dürfen keine UltraLite.NET API-Methoden aufgerufen werden.

Siehe auch

- [ULSyncProgressData-Klasse \[UltraLite.NET\] auf Seite 399](#)

ULSyncResult-Klasse

UL-Erw.: Repräsentiert den Status der letzten Synchronisation.

Visual Basic-Syntax

```
Public Class ULSyncResult
```

C#-Syntax

```
public class ULSyncResult
```

Mitglieder

Alle Mitglieder der ULSyncResult-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
AuthStatus-Eigenschaft	Gibt den Autorisierungsstatuscode für den letzten Synchronisationsversuch zurück.
AuthValue-Eigenschaft	Gibt den Rückgabewert aus Synchronisationsskripten mit angepasster Benutzerauthentifizierung zurück.
IgnoredRows-Eigenschaft	Prüft, ob ausgelesene Zeilen bei der letzten Synchronisation ignoriert wurden.
PartialDownloadRetained-Eigenschaft	Prüft, ob ein Teil-Download bei der letzten Synchronisation bewahrt wurde.

Name	Beschreibung
StreamErrorCode-Eigenschaft	Gibt den Fehler zurück, der vom Datenstrom selbst gemeldet wurde.
StreamErrorParameters-Eigenschaft	Gibt eine Liste mit Datenstrom-Fehlerparametern zurück, wobei Kom-mata als Trennzeichen verwendet werden.
StreamErrorSystem-Eigenschaft	Gibt den systemspezifischen Code für den Datenstromfehler zurück.
Timestamp-Eigenschaft	Gibt den Zeitstempel der letzten Synchronisation zurück.
UploadOK-Eigenschaft	Prüft, ob die letzte Upload-Synchronisation erfolgreich war.

Bemerkungen

Es gibt keinen Konstruktor für diese Klasse. Jede Verbindung besitzt ihre eigene ULSyncResult-Instanz, die ihr als ULConnection.SyncResult-Eigenschaft zugeordnet ist. Eine ULSyncResult-Instanz ist nur gültig, während die Verbindung geöffnet ist.

Siehe auch

- [ULConnection.SyncResult-Eigenschaft \[UltraLite.NET\] auf Seite 166](#)
- [ULConnection.Synchronize-Methode \[UltraLite.NET\] auf Seite 156](#)

AuthStatus-Eigenschaft

Gibt den Autorisierungsstatuscode für den letzten Synchronisationsversuch zurück.

Visual Basic-Syntax

```
Public ReadOnly Property AuthStatus As ULAuthStatusCode
```

C#-Syntax

```
public ULAuthStatusCode AuthStatus {get;}
```

Bemerkungen

Einer der ULAuthStatusCode-Werte, die den Autorisierungsstatus für den letzten Synchronisationsversuch angeben.

Siehe auch

- [ULAuthStatusCode-Enumeration \[UltraLite.NET\] auf Seite 456](#)

AuthValue-Eigenschaft

Gibt den Rückgabewert aus Synchronisationsskripten mit angepasster Benutzerauthentifizierung zurück.

Visual Basic-Syntax

```
Public ReadOnly Property AuthValue As Long
```

C#-Syntax

```
public long AuthValue {get;}
```

Bemerkungen

Eine Ganzzahl (long integer) aus Synchronisationsskripten mit angepasster Benutzerauthentifizierung.

IgnoredRows-Eigenschaft

Prüft, ob ausgelesene Zeilen bei der letzten Synchronisation ignoriert wurden.

Visual Basic-Syntax

```
Public ReadOnly Property IgnoredRows As Boolean
```

C#-Syntax

```
public bool IgnoredRows {get;}
```

Bemerkungen

TRUE, wenn gesendete Zeilen bei der letzten Synchronisation ignoriert wurden. FALSE, wenn keine Zeilen ignoriert wurden.

Siehe auch

- [ULSyncParms.DownloadOnly-Eigenschaft \[UltraLite.NET\] auf Seite 391](#)

PartialDownloadRetained-Eigenschaft

Prüft, ob ein Teil-Download bei der letzten Synchronisation bewahrt wurde.

Visual Basic-Syntax

```
Public ReadOnly Property PartialDownloadRetained As Boolean
```

C#-Syntax

```
public bool PartialDownloadRetained {get;}
```

Bemerkungen

TRUE, wenn ein Download unterbrochen wurde und der Teil-Download bewahrt wurde. FALSE, wenn der Download nicht unterbrochen oder der Teil-Download zurückgesetzt wurde.

Siehe auch

- [ULSyncParms.KeepPartialDownload-Eigenschaft \[UltraLite.NET\] auf Seite 392](#)

StreamErrorCode-Eigenschaft

Gibt den Fehler zurück, der vom Datenstrom selbst gemeldet wurde.

Visual Basic-Syntax

```
Public ReadOnly Property StreamErrorCode As UStreamErrorCode
```

C#-Syntax

```
public UStreamErrorCode StreamErrorCode {get;}
```

Bemerkungen

Einer der UStreamErrorCode-Werte, die den Fehler festlegen, der vom Datenstrom selbst gemeldet wurde. UStreamErrorCode.NONE, wenn kein Fehler aufgetreten ist.

StreamErrorParameters-Eigenschaft

Gibt eine Liste mit Datenstrom-Fehlerparametern zurück, wobei Kommata als Trennzeichen verwendet werden.

Visual Basic-Syntax

```
Public ReadOnly Property StreamErrorParameters As String
```

C#-Syntax

```
public string StreamErrorParameters {get;}
```

Bemerkungen

Enthält eine durch Kommas getrennte Liste mit Fehlerparametern für den Datenstromfehlercode, der in der StreamErrorCode-Eigenschaft gemeldet wird. Dies ist eine leere Zeichenfolge, wenn Fehler keine Parameter haben oder wenn kein Fehler festgelegt wurde.

Siehe auch

- [ULFileTransfer.StreamErrorCode-Eigenschaft \[UltraLite.NET\] auf Seite 295](#)

StreamErrorSystem-Eigenschaft

Gibt den systemspezifischen Code für den Datenstromfehler zurück.

Visual Basic-Syntax

```
Public ReadOnly Property StreamErrorSystem As Integer
```

C#-Syntax

```
public int StreamErrorSystem {get;}
```

Bemerkungen

Eine Ganzzahl, die den systemspezifischen Code des Datenstromfehlers angibt.

Timestamp-Eigenschaft

Gibt den Zeitstempel der letzten Synchronisation zurück.

Visual Basic-Syntax

```
Public ReadOnly Property Timestamp As Date
```

C#-Syntax

```
public DateTime Timestamp {get;}
```

Bemerkungen

Eine System.DateTime-Struktur, die den Zeitstempel der letzten Synchronisation angibt.

Siehe auch

- [System.DateTime](#)

UploadOK-Eigenschaft

Prüft, ob die letzte Upload-Synchronisation erfolgreich war.

Visual Basic-Syntax

```
Public ReadOnly Property UploadOK As Boolean
```

C#-Syntax

```
public bool UploadOK {get;}
```

Bemerkungen

TRUE, wenn die letzte Upload-Synchronisation erfolgreich war, sonst FALSE.

ULTable-Klasse

UL-Erw.: Stellt eine Tabelle in einer UltraLite-Datenbank dar.

Visual Basic-Syntax

```
Public Class ULTable Inherits ULResultSet
```

C#-Syntax

```
public class ULTable : ULResultSet
```

Basisklassen

- [ULResultSet-Klasse \[UltraLite.NET\] auf Seite 351](#)

Mitglieder

Alle Mitglieder der ULTable-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
AppendBytes-Methode	Hängt die angegebene Teilmenge des angegebenen Arrays von System.Bytes an den neuen Wert für die angegebene ULDbType.LongBinary-Spalte an.
AppendChars-Methode	Hängt die angegebene Teilmenge des angegebenen Arrays von System.Chars an den neuen Wert für die angegebene ULDbType.LongVarchar-Spalte an.
Close-Methode	Schließt den Cursor.
Delete-Methode	Löscht die aktuelle Zeile.
DeleteAllRows-Methode	Löscht alle Zeilen in der Tabelle.
Dispose-Methode (geerbt aus System.Data.Common.DbDataReader)	Gibt alle Ressourcen frei, die von der aktuellen Instanz der Klasse System.Data.Common.DbDataReader verwendet werden.
FindBegin-Methode	Bereitet die Durchführung einer neuen Suche in der Tabelle vor.
FindFirst-Methode	Verschiebt die aktuelle Position am Anfang beginnend vorwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer vollständigen Wertemenge im aktuellen Index übereinstimmt.
FindLast-Methode	Verschiebt die aktuelle Position ausgehend vom Ende rückwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer Gesamtmenge von Werten im aktuellen Index genau übereinstimmt.
FindNext-Methode	Setzt die ULTable.FindFirst-Suche mit einer von der aktuellen Position ausgehenden Vorwärtsbewegung durch die Tabelle fort, wobei untersucht wird, ob die nächste Zeile einem Wert oder einer Gesamtmenge von Werten im aktuellen Index entspricht.
FindPrevious-Methode	Setzt die ULTable.FindLast-Suche mit einer von der aktuellen Position ausgehenden Rückwärtsbewegung durch die Tabelle fort, wobei untersucht wird, ob die vorherige Zeile einem Wert oder einer Gesamtmenge von Werten im aktuellen Index entspricht.
GetBoolean-Methode	Gibt den Wert für die angegebene Spalte als System.Boolean zurück.

Name	Beschreibung
GetByte-Methode	Gibt den Wert für die angegebene Spalte als 8-Bit-Wert ohne Vorzeichen zurück (System.Byte).
GetBytes-Methode	UL-Erw.: Gibt den Wert für die angegebene Spalte als Array von System.Bytes-Werten zurück.
GetChar-Methode	Diese Methode wird in UltraLite.NET nicht unterstützt.
GetChars-Methode	Kopiert eine Untermenge des Werts für die angegebene ULDb-Type.LongVarchar-Spalte, beginnend beim angegebenen Offset, in das angegebene Offset des System.Char-Zielarrays.
GetData-Methode (Geerbt von System.Data.Common.DbDataReader)	Gibt ein System.Data.Common.DbDataReader -Objekt für die angeforderte Spalten-Ordinalzahl zurück.
GetDataTypeName-Methode	Gibt den Namen des Provider-Datentyps der angegebenen Spalte zurück.
GetDateTime-Methode	Gibt den Wert für die angegebene Spalte als System.DateTime-Typ mit Millisekunden-Genauigkeit zurück.
GetDbDataReader-Methode (geerbt aus System.Data.Common.DbDataReader)	Gibt ein System.Data.Common.DbDataReader -Objekt für die angeforderte Spaltenordinalzahl zurück, das mit einer providerspezifischen Implementierung aufgehoben werden kann.
GetDecimal-Methode	Gibt den Wert für die angegebene Spalte als System.Decimal-Typ zurück.
GetDouble-Methode	Gibt den Wert für die angegebene Spalte als System.Double-Typ zurück.
GetEnumerator-Methode	Gibt einen System.Collections.IEnumerator-Wert zurück, der das ULDataReader-Objekt durchläuft.
GetFieldType-Methode	Gibt den System.Type-Wert zurück, der für die angegebene Spalte am besten geeignet ist.
GetFieldValue(T)-Methode (geerbt aus System.Data.Common.DbDataReader)	Ruft den Wert der angegebenen Spalte synchron als Typ ab.

Name	Beschreibung
GetFieldValueAsync-Methode (geerbt aus System.Data.Common.DbDataReader)	Ruft den Wert der angegebenen Spalte asynchron als Typ ab.
GetFloat-Methode	Gibt den Wert für die angegebene Spalte als System.Single-Typ zurück.
GetGuid-Methode	Gibt den Wert für die angegebene Spalte als UUID-Typ zurück (System.Guid).
GetInt16-Methode	Gibt den Wert für die angegebene Spalte als System.Int16-Typ zurück.
GetInt32-Methode	Gibt den Wert für die angegebene Spalte als System.Int32-Typ zurück.
GetInt64-Methode	Gibt den Wert für die angegebene Spalte als System.Int64-Typ zurück.
GetName-Methode	Gibt den Namen der angegebenen Spalte zurück.
GetOrdinal-Methode	Gibt die Spalten-ID der angegebenen Spalte zurück.
GetProviderSpecificFieldType-Methode (geerbt aus System.Data.Common.DbDataReader)	Gibt den providerspezifischen Feldtyp der angegebenen Spalte zurück.
GetProviderSpecificValue-Methode (geerbt aus System.Data.Common.DbDataReader)	Ruft den Wert der angegebenen Spalte als Instanz von System.Object ab.
GetProviderSpecificValues-Methode (geerbt aus System.Data.Common.DbDataReader)	Ruft alle providerspezifischen Attributspalten in der Sammlung für die aktuelle Zeile ab.
GetRowCount-Methode	UL-Erw.: Gibt die Anzahl der Zeilen im Cursor innerhalb des Schwellenwerts zurück.
GetSchemaTable-Methode	Gibt einen System.Data.DataTable-Wert zurück, der die Spaltenmetadaten des ULDataReader-Objekts beschreibt.
GetStream-Methode (geerbt aus System.Data.Common.DbDataReader)	Ruft Daten als System.IO.Stream ab.

Name	Beschreibung
GetString-Methode	Gibt den Wert für die angegebene Spalte als System.String-Typ zurück.
GetTextReader-Methode (geerbt aus System.Data.Common.DbDataReader)	Ruft Daten als System.IO.TextReader ab.
GetTimeSpan-Methode	Gibt den Wert für die angegebene Spalte als System.TimeSpan-Typ mit Millisekunden-Genauigkeit zurück.
GetUInt16-Methode	Gibt den Wert für die angegebene Spalte als System.UInt16-Typ zurück.
GetUInt32-Methode	Gibt den Wert für die angegebene Spalte als System.UInt32-Typ zurück.
GetUInt64-Methode	Gibt den Wert für die angegebene Spalte als System.UInt64-Typ zurück.
GetValue-Methode	Gibt den Wert der angegebenen Spalte in ihrem nativen Format zurück.
GetValues-Methode	Gibt alle Spaltenwerte für die aktuelle Zeile zurück.
Insert-Methode	Fügt eine neue Zeile mit den aktuellen Spaltenwerten (mit den set-Methoden angegeben) ein.
InsertBegin-Methode	Bereitet das Einfügen einer neuen Zeile in die Tabelle vor, indem alle aktuellen Spaltenwerte auf ihre Standardwerte gesetzt werden.
IsDBNull-Methode	Prüft, ob der Wert aus der angegebenen Spalte NULL ist.
IsDBNullAsync-Methode (geerbt aus System.Data.Common.DbDataReader)	Eine asynchrone Version von System.Data.Common.DbDataReader.IsDBNull(System.Int32) , die einen Wert abruft, der angibt, ob die Spalte nicht vorhandene oder fehlende Werte enthält.
LookupBackward-Methode	Verschiebt die Cursorposition von hinten beginnend rückwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer vollständigen Wertemenge im aktuellen Index übereinstimmt oder kleiner als dieser Wert ist.
LookupBegin-Methode	Bereitet ein neues Nachschlagen in der Tabelle vor.

Name	Beschreibung
LookupForward-Methode	Verschiebt die aktuelle Position ausgehend vom Anfang vorwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer Gesamtmenge von Werten im aktuellen Index genau übereinstimmt oder größer ist.
MoveAfterLast-Methode	UL-Erw.: Positioniert den Cursor hinter die letzte Zeile des Cursors.
MoveBeforeFirst-Methode	UL-Erw.: Positioniert den Cursor vor die erste Zeile des Cursors.
MoveFirst-Methode	UL-Erw.: Positioniert den Cursor auf die erste Zeile des Cursors.
MoveLast-Methode	UL-Erw.: Positioniert den Cursor auf die letzte Zeile des Cursors.
MoveNext-Methode	UL-Erw.: Positioniert den Cursor in die nächste Zeile oder nach der letzten Zeile, wenn der Cursor bereits in der letzten Zeile war.
MovePrevious-Methode	UL-Erw.: Positioniert den Cursor in die vorhergehende Zeile oder vor die erste Zeile.
MoveRelative-Methode	UL-Erw.: Positioniert den Cursor relativ zur aktuellen Zeile.
NextResult-Methode	Rückt das ULDataReader-Objekt zum nächsten Ergebnis vor, wenn Ergebnisse von Batch-SQL-Anweisungen gelesen werden.
NextResultAsync-Methode (geerbt aus System.Data.Common.DbDataReader)	Eine asynchrone Version von System.Data.Common.DbDataReader.NextResult , die den Leser beim Lesen der Ergebnisse von Anweisungsfolgen von statements.Invokes System.Data.Common.DbDataReader.NextResultAsync(System.Threading.CancellationToken) mit CancellationToken.None zum nächsten Ergebnis vorrückt.
Read-Methode	Positioniert den Cursor in die nächste Zeile oder nach der letzten Zeile, wenn der Cursor bereits in der letzten Zeile war.
ReadAsync-Methode (geerbt aus System.Data.Common.DbDataReader)	Eine asynchrone Version von System.Data.Common.DbDataReader.Read , die den Leser zum nächsten Datensatz in der Ergebnismenge vorrückt.
SetBoolean-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Boolean-Objekts fest.
SetByte-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Byte-Objekts (8-Bit-Ganzzahl ohne Vorzeichen) fest.
SetBytes-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Bytes-Arrays fest.

Name	Beschreibung
SetDateTime-Methode	Legt den Wert für die angegebene Spalte mithilfe eines System.Date.Time-Objekts fest.
SetDBNull-Methode	Setzt eine Spalte auf NULL.
SetDecimal-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Decimal-Objekts fest.
SetDouble-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Double-Objekts fest.
SetFloat-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Single-Objekts fest.
SetGuid-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Guid-Objekts fest.
SetInt16-Methode	Legt den Wert für die angegebene Spalte unter Verwendung von System.Int16 fest.
SetInt32-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.Int32 fest.
SetInt64-Methode	Setzt den Wert für die angegebene Spalte unter Verwendung eines Int64-Werts.
SetString-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.String-Objekts fest.
SetTimeSpan-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.TimeSpan-Objekts fest.
SetToDefault-Methode	Setzt den Wert bei der angegebenen Spalte auf ihren Standardwert.
SetUInt16-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.UInt16-Objekts fest.
SetUInt32-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.UInt32-Objekts fest.
SetUInt64-Methode	Legt den Wert für die angegebene Spalte unter Verwendung eines System.UInt64-Objekts fest.
Truncate-Methode	Löscht alle Zeilen in der Tabelle, während gleichzeitig ein Stopp der Löschvorgänge bei der Synchronisation aktiviert wird.

Name	Beschreibung
Update-Methode	Aktualisiert die aktuelle Zeile mit den aktuellen Spaltenwerten (mit set-Methoden angegeben).
UpdateBegin-Methode	Bereitet die Aktualisierung der aktuellen Zeile vor.
Depth-Eigenschaft	Gibt die Tiefe der Verschachtelung für die aktuelle Zeile zurück.
FieldCount-Eigenschaft	Gibt die Anzahl der Spalten im Cursor zurück.
HasRows-Eigenschaft	Prüft, ob das ULDataReader-Objekt eine oder mehrere Zeilen hat.
IsBOF-Eigenschaft	UL-Erw.: Gibt an, ob sich die aktuelle Zeilenposition vor der ersten Zeile befindet.
IsClosed-Eigenschaft	Prüft, ob der Cursor aktuell offen ist.
IsEOF-Eigenschaft	UL-Erw.: Gibt an, ob sich die aktuelle Zeilenposition nach der letzten Zeile befindet.
RecordsAffected-Eigenschaft	Gibt die Anzahl der durch eine SQL-Anweisung geänderten, eingefügten oder gelöschten Zeilen zurück.
RowCount-Eigenschaft	UL-Erw.: Gibt die Anzahl der Zeilen im Cursor zurück.
Schema-Eigenschaft	Speichert das Tabellenschema.
this-Eigenschaft	Gibt den Wert der angegebenen Spalte in ihrem nativen Format zurück.
VisibleFieldCount-Eigenschaft (geerbt aus System.Data.Common.DbDataReader)	Ruft die Anzahl der Felder in System.Data.Common.DbDataReader ab, die nicht ausgeblendet sind.

Bemerkungen

Es gibt keinen Konstruktor für diese Klasse. Tabellen werden mit der `ULCommand.ExecuteTable`-Methode erstellt.

Siehe auch

- [ULCommand.ExecuteTable-Methode \[UltraLite.NET\] auf Seite 100](#)
- [ULCommand-Klasse \[UltraLite.NET\] auf Seite 72](#)
- [ULResultSet-Klasse \[UltraLite.NET\] auf Seite 351](#)
- [System.Data.IDataReader](#)
- [System.Data.IDataRecord](#)
- [System.IDisposable](#)

DeleteAllRows-Methode

Löscht alle Zeilen in der Tabelle.

Visual Basic-Syntax

```
Public Sub DeleteAllRows()
```

C#-Syntax

```
public void DeleteAllRows()
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

In einigen Anwendungen kann es nützlich sein, alle Zeilen aus einer Tabelle zu löschen, bevor eine neue Datenmenge in die Tabelle geladen wird. Zeilen können mit der `ULConnection.StopSynchronizationDelete`-Methode aus der UltraLite-Datenbank gelöscht werden, ohne dass sie aus der konsolidierten Datenbank gelöscht werden.

Siehe auch

- [ULTable.Truncate-Methode \[UltraLite.NET\] auf Seite 436](#)
- [ULConnection.StopSynchronizationDelete-Methode \[UltraLite.NET\] auf Seite 155](#)

FindBegin-Methode

Bereitet die Durchführung einer neuen Suche in der Tabelle vor.

Visual Basic-Syntax

```
Public Sub FindBegin()
```

C#-Syntax

```
public void FindBegin()
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die zu suchenden Werte werden angegeben, indem die zutreffendsten Type-Methoden zum Festlegen der Typen für die Spalten im Index aufgerufen werden, mit denen die Tabelle geöffnet wurde.

Siehe auch

- [ULTable.FindFirst-Methode \[UltraLite.NET\] auf Seite 423](#)
- [ULTable.FindLast-Methode \[UltraLite.NET\] auf Seite 425](#)

FindFirst-Methode

Verschiebt die aktuelle Position am Anfang beginnend vorwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer vollständigen Wertemenge im aktuellen Index übereinstimmt.

Überladungsliste

Name	Beschreibung
FindFirst()-Methode	Verschiebt die aktuelle Position am Anfang beginnend vorwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer vollständigen Wertemenge im aktuellen Index übereinstimmt.
FindFirst(short)-Methode	Verschiebt die Cursorposition von vorne beginnend vorwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einem Teil einer Wertemenge im aktuellen Index übereinstimmt.

FindFirst()-Methode

Verschiebt die aktuelle Position am Anfang beginnend vorwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer vollständigen Wertemenge im aktuellen Index übereinstimmt.

Visual Basic-Syntax

```
Public Function FindFirst() As Boolean
```

C#-Syntax

```
public bool FindFirst()
```

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Um den zu suchenden Wert festzulegen, legen Sie den Spaltenwert für die einzelnen Spalten im Index fest. Der Cursor bleibt in der ersten Zeile, die genau mit dem Indexwert übereinstimmt. Bei einem Fehlschlag ist die Cursorposition hinter der letzten Zeile (wie vom `ULDataReader.IsEOF`-Eigenschaft angezeigt).

Die `FindBegin`-Methode muss vor jeder Suche aufgerufen werden.

Siehe auch

- [ULTable.FindBegin-Methode \[UltraLite.NET\] auf Seite 422](#)
- [ULTable.FindNext-Methode \[UltraLite.NET\] auf Seite 427](#)
- [ULTable.FindPrevious-Methode \[UltraLite.NET\] auf Seite 429](#)
- [ULTable.FindFirst-Methode \[UltraLite.NET\] auf Seite 423](#)
- [ULDataReader.IsEOF-Eigenschaft \[UltraLite.NET\] auf Seite 271](#)
- [ULTable.FindBegin-Methode \[UltraLite.NET\] auf Seite 422](#)

FindFirst(short)-Methode

Verschiebt die Cursorposition von vorne beginnend vorwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einem Teil einer Wertemenge im aktuellen Index übereinstimmt.

Visual Basic-Syntax

```
Public Function FindFirst(ByVal numColumns As Short) As Boolean
```

C#-Syntax

```
public bool FindFirst(short numColumns)
```

Parameter

- **numColumns** Bei zusammengesetzten Indizes ist dies die Anzahl der Spalten, die bei der Suche verwendet werden soll. Wenn Sie z.B. einen dreispaltigen Index haben und einen Wert suchen, der lediglich in der ersten Spalte übereinstimmt, müssen Sie den Wert der ersten Spalte angeben und dann den Wert 1 festlegen.

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Um den zu suchenden Wert festzulegen, legen Sie den Spaltenwert für die einzelnen Spalten im Index fest. Der Cursor bleibt in der ersten Zeile, die genau mit dem Indexwert übereinstimmt. Bei einem Fehlschlag ist die Cursorposition hinter der letzten Zeile (wie vom ULDataReader.IsEOF-Eigenschaft angezeigt).

Die FindBegin-Methode muss vor jeder Suche aufgerufen werden.

Siehe auch

- [ULTable.FindBegin-Methode \[UltraLite.NET\] auf Seite 422](#)
- [ULTable.FindNext-Methode \[UltraLite.NET\] auf Seite 427](#)
- [ULTable.FindPrevious-Methode \[UltraLite.NET\] auf Seite 429](#)
- [ULTable.FindFirst-Methode \[UltraLite.NET\] auf Seite 423](#)
- [ULDataReader.IsEOF-Eigenschaft \[UltraLite.NET\] auf Seite 271](#)
- [ULTable.FindBegin-Methode \[UltraLite.NET\] auf Seite 422](#)

FindLast-Methode

Verschiebt die aktuelle Position ausgehend vom Ende rückwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer Gesamtmenge von Werten im aktuellen Index genau übereinstimmt.

Überladungsliste

Name	Beschreibung
FindLast()-Methode	Verschiebt die aktuelle Position ausgehend vom Ende rückwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer Gesamtmenge von Werten im aktuellen Index genau übereinstimmt.
FindLast(short)-Methode	Verschiebt die aktuelle Position ausgehend vom Ende rückwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer Teilmenge von Werten im aktuellen Index genau übereinstimmt.

FindLast()-Methode

Verschiebt die aktuelle Position ausgehend vom Ende rückwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer Gesamtmenge von Werten im aktuellen Index genau übereinstimmt.

Visual Basic-Syntax

```
Public Function FindLast() As Boolean
```

C#-Syntax

```
public bool FindLast()
```

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Um den zu suchenden Wert festzulegen, legen Sie den Spaltenwert für die einzelnen Spalten im Index fest. Der Cursor bleibt in der ersten Zeile, die genau mit dem Indexwert übereinstimmt. Bei einem

Fehlschlag ist die Cursorposition vor der ersten Zeile (wie von der `ULDataReader.IsBOF`-Eigenschaft angezeigt).

Die `FindBegin`-Methode muss vor jeder Suche aufgerufen werden.

Siehe auch

- [ULTable.FindBegin-Methode \[UltraLite.NET\] auf Seite 422](#)
- [ULTable.FindNext-Methode \[UltraLite.NET\] auf Seite 427](#)
- [ULTable.FindPrevious-Methode \[UltraLite.NET\] auf Seite 429](#)
- [ULTable.FindLast-Methode \[UltraLite.NET\] auf Seite 425](#)
- [ULDataReader.IsBOF-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [ULTable.FindBegin-Methode \[UltraLite.NET\] auf Seite 422](#)

FindLast(short)-Methode

Verschiebt die aktuelle Position ausgehend vom Ende rückwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer Teilmenge von Werten im aktuellen Index genau übereinstimmt.

Visual Basic-Syntax

```
Public Function FindLast(ByVal numColumns As Short) As Boolean
```

C#-Syntax

```
public bool FindLast(short numColumns)
```

Parameter

- **numColumns** Bei zusammengesetzten Indizes ist dies die Anzahl der Spalten, die bei der Suche verwendet werden soll. Wenn Sie z.B. einen dreispaltigen Index haben und einen Wert suchen, der lediglich in der ersten Spalte übereinstimmt, müssen Sie den Wert der ersten Spalte und dann den Wert 1 festlegen.

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Um den zu suchenden Wert festzulegen, legen Sie den Spaltenwert für die einzelnen Spalten im Index fest. Der Cursor bleibt in der ersten Zeile, die genau mit dem Indexwert übereinstimmt. Bei einem Fehlschlag ist die Cursorposition vor der ersten Zeile (wie von der `ULDataReader.IsBOF`-Eigenschaft angezeigt).

Die `FindBegin`-Methode muss vor jeder Suche aufgerufen werden.

Siehe auch

- [ULTable.FindBegin-Methode \[UltraLite.NET\] auf Seite 422](#)
- [ULTable.FindNext-Methode \[UltraLite.NET\] auf Seite 427](#)
- [ULTable.FindPrevious-Methode \[UltraLite.NET\] auf Seite 429](#)
- [ULTable.FindLast-Methode \[UltraLite.NET\] auf Seite 425](#)
- [ULDataReader.IsBOF-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)
- [ULTable.FindBegin-Methode \[UltraLite.NET\] auf Seite 422](#)

FindNext-Methode

Setzt die ULTable.FindFirst-Suche mit einer von der aktuellen Position ausgehenden Vorwärtsbewegung durch die Tabelle fort, wobei untersucht wird, ob die nächste Zeile einem Wert oder einer Gesamtmenge von Werten im aktuellen Index entspricht.

Überladungsliste

Name	Beschreibung
FindNext()-Methode	Setzt die ULTable.FindFirst-Suche mit einer von der aktuellen Position ausgehenden Vorwärtsbewegung durch die Tabelle fort, wobei untersucht wird, ob die nächste Zeile einem Wert oder einer Gesamtmenge von Werten im aktuellen Index entspricht.
FindNext(short)-Methode	Setzt die ULTable.FindFirst-Suche mit einer von der aktuellen Position ausgehenden Vorwärtsbewegung durch die Tabelle fort, wobei untersucht wird, ob die nächste Zeile einem Wert oder einer Teilmenge von Werten im aktuellen Index entspricht.

FindNext()-Methode

Setzt die ULTable.FindFirst-Suche mit einer von der aktuellen Position ausgehenden Vorwärtsbewegung durch die Tabelle fort, wobei untersucht wird, ob die nächste Zeile einem Wert oder einer Gesamtmenge von Werten im aktuellen Index entspricht.

Visual Basic-Syntax

```
Public Function FindNext() As Boolean
```

C#-Syntax

```
public bool FindNext()
```

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Der Cursor bleibt in der nächsten Zeile, wenn sie genau mit dem Indexwert übereinstimmt. Bei einem Fehlschlag ist die Cursorposition hinter der letzten Zeile (wie vom `ULDataReader.IsEOF`-Eigenschaft angezeigt).

Das Verhalten der `FindNext`-Methode ist nicht festgelegt, wenn die gesuchten Spaltenwerte während einer Zeilenaktualisierung geändert werden.

Siehe auch

- [ULTable.FindFirst-Methode \[UltraLite.NET\] auf Seite 423](#)
- [ULTable.FindNext-Methode \[UltraLite.NET\] auf Seite 427](#)
- [ULDataReader.IsEOF-Eigenschaft \[UltraLite.NET\] auf Seite 271](#)

FindNext(short)-Methode

Setzt die `ULTable.FindFirst`-Suche mit einer von der aktuellen Position ausgehenden Vorwärtsbewegung durch die Tabelle fort, wobei untersucht wird, ob die nächste Zeile einem Wert oder einer Teilmenge von Werten im aktuellen Index entspricht.

Visual Basic-Syntax

```
Public Function FindNext(ByVal numColumns As Short) As Boolean
```

C#-Syntax

```
public bool FindNext(short numColumns)
```

Parameter

- **numColumns** Bei zusammengesetzten Indizes ist dies die Anzahl der Spalten, die bei der Suche verwendet werden soll. Wenn Sie z.B. einen dreispaltigen Index haben und einen Wert suchen, der lediglich in der ersten Spalte übereinstimmt, müssen Sie den Wert der ersten Spalte und dann den Wert 1 festlegen.

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Der Cursor bleibt in der nächsten Zeile, wenn sie genau mit dem Indexwert übereinstimmt. Bei einem Fehlschlag ist die Cursorposition hinter der letzten Zeile (wie vom `ULDataReader.IsEOF`-Eigenschaft angezeigt).

Das Verhalten der `FindNext`-Methode ist nicht festgelegt, wenn die gesuchten Spaltenwerte während einer Zeilenaktualisierung geändert werden.

Siehe auch

- [ULTable.FindFirst-Methode \[UltraLite.NET\] auf Seite 423](#)
- [ULTable.FindNext-Methode \[UltraLite.NET\] auf Seite 427](#)
- [ULTable.FindFirst-Methode \[UltraLite.NET\] auf Seite 423](#)
- [ULDataReader.IsEOF-Eigenschaft \[UltraLite.NET\] auf Seite 271](#)

FindPrevious-Methode

Setzt die ULTable.FindLast-Suche mit einer von der aktuellen Position ausgehenden Rückwärtsbewegung durch die Tabelle fort, wobei untersucht wird, ob die vorherige Zeile einem Wert oder einer Gesamtmenge von Werten im aktuellen Index entspricht.

Überladungsliste

Name	Beschreibung
FindPrevious()-Methode	Setzt die ULTable.FindLast-Suche mit einer von der aktuellen Position ausgehenden Rückwärtsbewegung durch die Tabelle fort, wobei untersucht wird, ob die vorherige Zeile einem Wert oder einer Gesamtmenge von Werten im aktuellen Index entspricht.
FindPrevious(short)-Methode	Setzt die ULTable.FindLast-Suche mit einer von der aktuellen Position ausgehenden Rückwärtsbewegung durch die Tabelle fort, wobei untersucht wird, ob die vorherige Zeile einem Wert oder einer Teilmenge von Werten im aktuellen Index entspricht.

FindPrevious()-Methode

Setzt die ULTable.FindLast-Suche mit einer von der aktuellen Position ausgehenden Rückwärtsbewegung durch die Tabelle fort, wobei untersucht wird, ob die vorherige Zeile einem Wert oder einer Gesamtmenge von Werten im aktuellen Index entspricht.

Visual Basic-Syntax

```
Public Function FindPrevious( ) As Boolean
```

C#-Syntax

```
public bool FindPrevious( )
```

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Der Cursor bleibt in der vorherigen Zeile, wenn sie genau mit dem Indexwert übereinstimmt. Bei einem Fehlschlag ist die Cursorposition vor der ersten Zeile (wie von der `ULDataReader.IsBOF`-Eigenschaft angezeigt).

Das Verhalten der `FindPrevious`-Methode ist nicht festgelegt, wenn die gesuchten Spaltenwerte während einer Zeilenaktualisierung geändert werden.

Siehe auch

- [ULTable.FindLast-Methode \[UltraLite.NET\] auf Seite 425](#)
- [ULTable.FindPrevious-Methode \[UltraLite.NET\] auf Seite 429](#)
- [ULDataReader.IsBOF-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)

FindPrevious(short)-Methode

Setzt die `ULTable.FindLast`-Suche mit einer von der aktuellen Position ausgehenden Rückwärtsbewegung durch die Tabelle fort, wobei untersucht wird, ob die vorherige Zeile einem Wert oder einer Teilmenge von Werten im aktuellen Index entspricht.

Visual Basic-Syntax

```
Public Function FindPrevious(ByVal numColumns As Short) As Boolean
```

C#-Syntax

```
public bool FindPrevious(short numColumns)
```

Parameter

- **numColumns** Bei zusammengesetzten Indizes ist dies die Anzahl der Spalten, die bei der Suche verwendet werden soll. Wenn Sie z.B. einen dreispaltigen Index haben und einen Wert suchen, der lediglich in der ersten Spalte übereinstimmt, müssen Sie den Wert der ersten Spalte angeben und dann den Wert 1 festlegen.

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Der Cursor bleibt in der vorherigen Zeile, wenn sie genau mit dem Indexwert übereinstimmt. Bei einem Fehlschlag ist die Cursorposition vor der ersten Zeile (wie von der `ULDataReader.IsBOF`-Eigenschaft angezeigt).

Das Verhalten der `FindPrevious`-Methode ist nicht festgelegt, wenn die gesuchten Spaltenwerte während einer Zeilenaktualisierung geändert werden.

Siehe auch

- [ULTable.FindLast-Methode \[UltraLite.NET\] auf Seite 425](#)
- [ULTable.FindPrevious-Methode \[UltraLite.NET\] auf Seite 429](#)
- [ULDataReader.IsBOF-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)

Insert-Methode

Fügt eine neue Zeile mit den aktuellen Spaltenwerten (mit den set-Methoden angegeben) ein.

Visual Basic-Syntax

```
Public Sub Insert()
```

C#-Syntax

```
public void Insert()
```

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Jedem Einfügevorgang muss ein Aufruf der ULTable.InsertBegin-Methode vorangehen.

Siehe auch

- [ULTable.InsertBegin-Methode \[UltraLite.NET\] auf Seite 431](#)

InsertBegin-Methode

Bereitet das Einfügen einer neuen Zeile in die Tabelle vor, indem alle aktuellen Spaltenwerte auf ihre Standardwerte gesetzt werden.

Visual Basic-Syntax

```
Public Sub InsertBegin()
```

C#-Syntax

```
public void InsertBegin()
```

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Rufen Sie eine oder mehrere entsprechende SetType- oder AppendType-Methoden auf, um die Nicht-Standardwerte anzugeben, die eingefügt werden sollen.

Die Zeile wird erst eingefügt und die Daten in der Zeile werden erst geändert, wenn Sie die Insert-Methode ausführen. Diese Änderung ist erst dann dauerhaft, wenn sie festgeschrieben wird.

Siehe auch

- [ULTable.Insert-Methode \[UltraLite.NET\]](#) auf Seite 431

LookupBackward-Methode

Verschiebt die Cursorposition von hinten beginnend rückwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer vollständigen Wertemenge im aktuellen Index übereinstimmt oder kleiner als dieser Wert ist.

Überladungsliste

Name	Beschreibung
LookupBackward()-Methode	Verschiebt die Cursorposition von hinten beginnend rückwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer vollständigen Wertemenge im aktuellen Index übereinstimmt oder kleiner als dieser Wert ist.
LookupBackward(short)-Methode	Verschiebt die Cursorposition vom Ende beginnend rückwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einem Teil der Wertemenge im aktuellen Index übereinstimmt oder kleiner als dieser Wert ist.

LookupBackward()-Methode

Verschiebt die Cursorposition von hinten beginnend rückwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer vollständigen Wertemenge im aktuellen Index übereinstimmt oder kleiner als dieser Wert ist.

Visual Basic-Syntax

```
Public Function LookupBackward() As Boolean
```

C#-Syntax

```
public bool LookupBackward()
```

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Um den zu suchenden Wert festzulegen, legen Sie den Spaltenwert für die einzelnen Spalten im Index fest. Der Cursor bleibt in der ersten Zeile, die mit dem Indexwert übereinstimmt oder kleiner als dieser Wert ist. Bei einem Fehler (d.h. keine Zeile ist kleiner als der gesuchte Wert) befindet sich der Cursor vor der ersten Zeile (wie von der ULDataReader.IsBOF-Eigenschaft gezeigt).

Die LookupBegin-Methode muss vor jeder Suche aufgerufen werden.

Siehe auch

- [ULTable.LookupBegin-Methode \[UltraLite.NET\] auf Seite 434](#)
- [ULTable.LookupBackward-Methode \[UltraLite.NET\] auf Seite 432](#)
- [ULDataReader.IsBOF-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)

LookupBackward(short)-Methode

Verschiebt die Cursorposition vom Ende beginnend rückwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einem Teil der Wertemenge im aktuellen Index übereinstimmt oder kleiner als dieser Wert ist.

Visual Basic-Syntax

```
Public Function LookupBackward(ByVal numColumns As Short) As Boolean
```

C#-Syntax

```
public bool LookupBackward(short numColumns)
```

Parameter

- **numColumns** Bei zusammengesetzten Indizes ist dies die Anzahl der Spalten, die bei der Suche verwendet werden soll. Wenn Sie z.B. einen dreispaltigen Index haben und einen Wert suchen, der lediglich in der ersten Spalte übereinstimmt, müssen Sie den Wert der ersten Spalte angeben und dann den Wert 1 festlegen.

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Um den zu suchenden Wert festzulegen, legen Sie den Spaltenwert für die einzelnen Spalten im Index fest. Der Cursor bleibt in der ersten Zeile, die mit dem Indexwert übereinstimmt oder kleiner als dieser Wert ist. Bei einem Fehler (d.h. keine Zeile ist kleiner als der gesuchte Wert) befindet sich der Cursor vor der ersten Zeile (wie von der ULDataReader.IsBOF-Eigenschaft gezeigt).

Die LookupBegin-Methode muss vor jeder Suche aufgerufen werden.

Siehe auch

- [ULTable.LookupBegin-Methode \[UltraLite.NET\] auf Seite 434](#)
- [ULTable.LookupBackward-Methode \[UltraLite.NET\] auf Seite 432](#)
- [ULDataReader.IsBOF-Eigenschaft \[UltraLite.NET\] auf Seite 270](#)

LookupBegin-Methode

Bereitet ein neues Nachschlagen in der Tabelle vor.

Visual Basic-Syntax

```
Public Sub LookupBegin( )
```

C#-Syntax

```
public void LookupBegin( )
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die zu suchenden Werte werden angegeben, indem die zutreffendsten Type-Methoden zum Festlegen der Typen für die Spalten im Index aufgerufen werden, mit denen die Tabelle geöffnet wurde.

Siehe auch

- [ULTable.LookupForward-Methode \[UltraLite.NET\] auf Seite 434](#)
- [ULTable.LookupBackward-Methode \[UltraLite.NET\] auf Seite 432](#)

LookupForward-Methode

Verschiebt die aktuelle Position ausgehend vom Anfang vorwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer Gesamtmenge von Werten im aktuellen Index genau übereinstimmt oder größer ist.

Überladungsliste

Name	Beschreibung
LookupForward()-Methode	Verschiebt die aktuelle Position ausgehend vom Anfang vorwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer Gesamtmenge von Werten im aktuellen Index genau übereinstimmt oder größer ist.
LookupForward(short)-Methode	Verschiebt die Cursorposition von vorne beginnend vorwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer Teilwertemenge im aktuellen Index übereinstimmt oder größer als dieser Wert ist.

LookupForward()-Methode

Verschiebt die aktuelle Position ausgehend vom Anfang vorwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer Gesamtmenge von Werten im aktuellen Index genau übereinstimmt oder größer ist.

Visual Basic-Syntax

```
Public Function LookupForward() As Boolean
```

C#-Syntax

```
public bool LookupForward()
```

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Um den zu suchenden Wert festzulegen, legen Sie den Spaltenwert für die einzelnen Spalten im Index fest. Der Cursor bleibt in der ersten Zeile, die mit dem Indexwert übereinstimmt oder größer als dieser Wert ist. Bei einem Fehler (d.h. keine Zeile ist größer als der gesuchte Wert) befindet sich der Cursor hinter der letzten Zeile (wie von der `ULDataReader.IsEOF` gezeigt).

Die `LookupBegin`-Methode muss vor jeder Suche aufgerufen werden.

Siehe auch

- [ULTable.LookupBegin-Methode \[UltraLite.NET\] auf Seite 434](#)
- [ULTable.LookupForward-Methode \[UltraLite.NET\] auf Seite 434](#)
- [ULDataReader.IsEOF-Eigenschaft \[UltraLite.NET\] auf Seite 271](#)

LookupForward(short)-Methode

Verschiebt die Cursorposition von vorne beginnend vorwärts durch die Tabelle und sucht nach einer Zeile, die mit einem Wert oder einer Teilwertemenge im aktuellen Index übereinstimmt oder größer als dieser Wert ist.

Visual Basic-Syntax

```
Public Function LookupForward(ByVal numColumns As Short) As Boolean
```

C#-Syntax

```
public bool LookupForward(short numColumns)
```

Parameter

- **numColumns** Bei zusammengesetzten Indizes ist dies die Anzahl der Spalten, die bei der Suche verwendet werden soll. Wenn Sie z.B. einen dreispaltigen Index haben und einen Wert suchen, der lediglich in der ersten Spalte übereinstimmt, müssen Sie den Wert der ersten Spalte angeben und dann den Wert 1 festlegen.

Rückgabe

TRUE, wenn erfolgreich, andernfalls FALSE.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Um den zu suchenden Wert festzulegen, legen Sie den Spaltenwert für die einzelnen Spalten im Index fest. Der Cursor bleibt in der ersten Zeile, die mit dem Indexwert übereinstimmt oder größer als dieser Wert ist. Bei einem Fehler (d.h. keine Zeile ist größer als der gesuchte Wert) befindet sich der Cursor hinter der letzten Zeile (wie von der `ULDataReader.IsEOF` gezeigt).

Die `LookupBegin`-Methode muss vor jeder Suche aufgerufen werden.

Siehe auch

- [ULTable.LookupBegin-Methode \[UltraLite.NET\] auf Seite 434](#)
- [ULTable.LookupForward-Methode \[UltraLite.NET\] auf Seite 434](#)
- [ULDataReader.IsEOF-Eigenschaft \[UltraLite.NET\] auf Seite 271](#)
- [ULTable.LookupBegin-Methode \[UltraLite.NET\] auf Seite 434](#)

Truncate-Methode

Löscht alle Zeilen in der Tabelle, während gleichzeitig ein Stopp der Löschvorgänge bei der Synchronisation aktiviert wird.

Visual Basic-Syntax

```
Public Sub Truncate()
```

C#-Syntax

```
public void Truncate()
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULTable.DeleteAllRows-Methode \[UltraLite.NET\] auf Seite 422](#)

Schema-Eigenschaft

Speichert das Tabellenschema.

Visual Basic-Syntax

```
Public ReadOnly Shadows Property Schema As ULTableSchema
```

C#-Syntax

```
public new ULTableSchema Schema {get;}
```

Bemerkungen

Diese Eigenschaft ist nur gültig, während die Verbindung geöffnet ist.

Das ULTableSchema-Objekt, das das Tabellenschema repräsentiert.

Diese Eigenschaft stellt das vollständige Schema der Tabelle dar, einschließlich erweiterter UltraLite.NET-Informationen, die in den Ergebnissen vom Aufruf der ULDataReader.GetSchemaTable-Methode nicht enthalten sind.

Siehe auch

- [ULTableSchema-Klasse \[UltraLite.NET\] auf Seite 437](#)

ULTableSchema-Klasse

UL-Erw.: Stellt das Schema einer UltraLite-Tabelle dar.

Visual Basic-Syntax

```
Public NotInheritable Class ULTableSchema Inherits ULCursorSchema
```

C#-Syntax

```
public sealed class ULTableSchema : ULCursorSchema
```

Basisklassen

- [ULCursorSchema-Klasse \[UltraLite.NET\] auf Seite 204](#)

Mitglieder

Alle Mitglieder der ULTableSchema-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
GetColumnDefaultValue-Methode	Gibt den Standardwert der angegebenen Spalte zurück.
GetColumnID-Methode	Gibt die Spalten-ID der angegebenen Spalte zurück.
GetColumnName-Methode	Gibt den Namen der Spalte zurück, die von der Spalten-ID angegeben wird.
GetColumnPartitionSize-Methode	Gibt die Partitionsgröße für global autoincrement zurück, die der angegebenen Spalte zugeordnet ist.
GetColumnPrecision-Methode	Gibt die Gesamtstellenzahl der Spalte zurück, die durch die angegebene Spalten-ID identifiziert wird, wenn die Spalte numerisch ist (SQL-Typ NUMERIC).

Name	Beschreibung
GetColumnScale-Methode	Gibt die Anzahl der Dezimalstellen der Spalte zurück, die durch die angegebene Spalten-ID identifiziert wird, wenn die Spalte numerisch ist (SQL-Typ NUMERIC).
GetColumnSize-Methode	Gibt die Größe der Spalte zurück, die durch die angegebene Spalten-ID identifiziert wird, wenn die Spalte eine Größenangabe hat (SQL-Typ BINARY oder CHAR).
GetColumnSQLName-Methode	Gibt den Namen der Spalte zurück, die von der Spalten-ID angegeben wird.
GetColumnULDbType-Methode	Gibt den UltraLite.NET-Datentyp der Spalte zurück, die von der angegebenen Spalten-ID identifiziert wird.
GetIndex-Methode	Gibt das Indexschema des genannten Indexes zurück.
GetIndexName-Methode	Gibt den Namen des Indexes zurück, dessen Index-ID angegeben wurde.
GetOptimalIndex-Methode	Der optimale Index zum Suchen einer Tabelle unter Verwendung der angegebenen Spalte.
GetPublicationPredicate-Methode	Gibt das Publikationsprädikat für diese Tabelle in der benannten Publikation zurück.
GetSchemaTable-Methode	Gibt ein System.Data.DataTable-Objekt zurück, das das Spaltenschema des ULDataReader-Objekts beschreibt.
IsColumnAutoIncrement-Methode	Prüft, ob der Standardwert der angegebenen Spalte auf AUTOINCREMENT gesetzt ist.
IsColumnCurrentDate-Methode	Prüft, ob der Standardwert der angegebenen Spalte auf das aktuelle Datum gesetzt ist (einen ULDbType.Date-Wert).
IsColumnCurrentTime-Methode	Prüft, ob der Standardwert der angegebenen Spalte auf die aktuelle Uhrzeit gesetzt ist (einen ULDbType.Time-Wert).
IsColumnCurrentTimesamp-Methode	Prüft, ob der Standardwert der angegebenen Spalte auf den aktuellen Zeitstempel gesetzt ist (einen ULDbType.TimeStamp-Wert).
IsColumnCurrentUTCTimestamp-Methode	Prüft, ob der Standardwert der angegebenen Spalte auf den aktuellen UTC-Zeitstempel gesetzt ist (einen ULDbType.TimeStamp-Wert).
IsColumnGlobalAutoIncrement-Methode	Prüft, ob der Standardwert der angegebenen Spalte auf GLOBAL AUTOINCREMENT gesetzt ist.

Name	Beschreibung
IsColumnNewUUID-Methode	Prüft, ob eine neue UUID (ein System.Guid-Wert) als Standardwert für die angegebene Spalte festgelegt ist.
IsColumnNullable-Methode	Prüft, ob die angegebene Spalte nullwertfähig ist.
IsInPublication-Methode	Prüft, ob die Tabelle in der genannten Publikation enthalten ist.
ColumnCount-Eigenschaft	Gibt die Anzahl der Spalten im Cursor zurück.
IndexCount-Eigenschaft	Gibt die Anzahl der Indizes der Tabelle zurück.
IsNeverSynchronized-Eigenschaft	Prüft, ob die Tabelle als nie zu synchronisieren markiert ist.
IsOpen-Eigenschaft	Prüft, ob das Cursorschema aktuell geöffnet ist.
Name-Eigenschaft	Gibt den Namen der Tabelle zurück.
PrimaryKey-Eigenschaft	Gibt das Indexschema des Primärschlüssels der Tabelle zurück.
UploadUnchangedRows-Eigenschaft	Prüft, ob die Datenbank Zeilen ausliest, die nicht geändert wurden.

Bemerkungen

Es gibt keinen Konstruktor für diese Klasse. Ein ULTableSchema-Objekt ist einer Tabelle als ihre ULTable.Schema-Eigenschaft zugeordnet.

Siehe auch

- [ULTableSchema-Klasse \[UltraLite.NET\] auf Seite 437](#)
- [ULTable.Schema-Eigenschaft \[UltraLite.NET\] auf Seite 436](#)
- [ULCursorSchema-Klasse \[UltraLite.NET\] auf Seite 204](#)

GetColumnDefaultValue-Methode

Gibt den Standardwert der angegebenen Spalte zurück.

Visual Basic-Syntax

```
Public Function GetColumnDefaultValue(  
    ByVal columnID As Integer  
) As String
```

C#-Syntax

```
public string GetColumnDefaultValue(int columnID)
```

Parameter

- **columnID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULCursorSchema.ColumnCount-1] liegen. Die erste Spalte in einer Tabelle hat den ID-Wert Null.

Rückgabe

Der Standardwert der angegebenen Spalte als Zeichenfolge oder Nullreferenz ("Nothing" in Visual Basic), wenn der Standardwert Null ist.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)

GetColumnPartitionSize-Methode

Gibt die Partitionsgröße für global autoincrement zurück, die der angegebenen Spalte zugeordnet ist.

Visual Basic-Syntax

```
Public Function GetColumnPartitionSize(  
    ByVal columnID As Integer  
) As ULong
```

C#-Syntax

```
public ulong GetColumnPartitionSize(int columnID)
```

Parameter

- **columnID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULCursorSchema.ColumnCount-1] liegen. Die erste Spalte in der Tabelle hat den ID-Wert Null.

Rückgabe

Die globale Autoinkrement-Partitionsgröße der Spalte als eine System.UInt64-Struktur.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Alle global autoincrement-Spalten in einer gegebenen Tabelle verwenden dieselbe global autoincrement-Partition.

Siehe auch

- [ULTableSchema.IsColumnGlobalAutoIncrement-Methode \[UltraLite.NET\] auf Seite 446](#)
- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)
- [System.UInt64](#)

GetIndex-Methode

Gibt das Indexschema des genannten Indexes zurück.

Visual Basic-Syntax

```
Public Function GetIndex(ByVal name As String) As ULIndexSchema
```

C#-Syntax

```
public ULIndexSchema GetIndex(string name)
```

Parameter

- **name** Der Name des Indexes.

Rückgabe

Ein ULIndexSchema-Objekt, das den benannten Index darstellt.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULIndexSchema-Klasse \[UltraLite.NET\] auf Seite 301](#)

GetIndexName-Methode

Gibt den Namen des Indexes zurück, dessen Index-ID angegeben wurde.

Visual Basic-Syntax

```
Public Function GetIndexName(ByVal indexID As Integer) As String
```

C#-Syntax

```
public string GetIndexName(int indexID)
```

Parameter

- **indexID** Die ID des Indexes. Der Wert muss im Bereich [1,IndexCount] liegen.

Rückgabe

Der Name des Indexes als Zeichenfolge.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Index-IDs und Indexanzahl können sich während eines Schema-Upgrades ändern. Um einen Index korrekt zu identifizieren, greifen Sie auf ihn anhand des Namens zu oder aktualisieren die IDs und Anzahl im Cache nach jedem Schema-Upgrade.

Siehe auch

- [ULTableSchema.IndexCount-Eigenschaft \[UltraLite.NET\] auf Seite 448](#)

GetOptimalIndex-Methode

Der optimale Index zum Suchen einer Tabelle unter Verwendung der angegebenen Spalte.

Visual Basic-Syntax

```
Public Function GetOptimalIndex(ByVal columnID As Integer) As String
```

C#-Syntax

```
public string GetOptimalIndex(int columnID)
```

Parameter

- **columnID** Die ID-Nummer der Spalte. Die erste Spalte in der Tabelle hat den ID-Wert Null.

Rückgabe

Ein ULIndexSchema-Objekt, das den optimalen Index für die angegebene Spalte repräsentiert.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Die angegebene Spalte ist die erste Spalte im Index, doch der Index kann mehrere Spalten enthalten.

Siehe auch

- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)
- [ULIndexSchema-Klasse \[UltraLite.NET\] auf Seite 301](#)

GetPublicationPredicate-Methode

Gibt das Publikationsprädikat für diese Tabelle in der benannten Publikation zurück.

Visual Basic-Syntax

```
Public Function GetPublicationPredicate(  
    ByVal pubName As String  
) As String
```

C#-Syntax

```
public string GetPublicationPredicate(string pubName)
```

Parameter

- **pubName** Der Name der Publikation

Rückgabe

Das Publikationsprädikat als Zeichenfolge.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

IsColumnAutoIncrement-Methode

Prüft, ob der Standardwert der angegebenen Spalte auf AUTOINCREMENT gesetzt ist.

Visual Basic-Syntax

```
Public Function IsColumnAutoIncrement(  
    ByVal columnID As Integer  
) As Boolean
```

C#-Syntax

```
public bool IsColumnAutoIncrement(int columnID)
```

Parameter

- **columnID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULCursorSchema.ColumnCount-1] liegen. Die erste Spalte in der Tabelle hat den ID-Wert Null.

Rückgabe

TRUE, wenn der Spaltenwert automatisch erhöht wird, andernfalls FALSE.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)

IsColumnCurrentDate-Methode

Prüft, ob der Standardwert der angegebenen Spalte auf das aktuelle Datum gesetzt ist (einen ULDbType.Date-Wert).

Visual Basic-Syntax

```
Public Function IsColumnCurrentDate(  
    ByVal columnID As Integer  
) As Boolean
```

C#-Syntax

```
public bool IsColumnCurrentDate(int columnID)
```

Parameter

- **columnID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULCursorSchema.ColumnCount-1] liegen. Die erste Spalte in der Tabelle hat den ID-Wert Null.

Rückgabe

TRUE, wenn die Spalte standardmäßig das aktuelle Datum anzeigt, sonst FALSE.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)

IsColumnCurrentTime-Methode

Prüft, ob der Standardwert der angegebenen Spalte auf die aktuelle Uhrzeit gesetzt ist (einen ULDbType.Time-Wert).

Visual Basic-Syntax

```
Public Function IsColumnCurrentTime(  
    ByVal columnID As Integer  
) As Boolean
```

C#-Syntax

```
public bool IsColumnCurrentTime(int columnID)
```

Parameter

- **columnID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULCursorSchema.ColumnCount-1] liegen. Die erste Spalte in der Tabelle hat den ID-Wert Null.

Rückgabe

TRUE, wenn die Spalte standardmäßig die aktuelle Uhrzeit anzeigt, sonst FALSE.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)

IsColumnCurrentTimestamp-Methode

Prüft, ob der Standardwert der angegebenen Spalte auf den aktuellen Zeitstempel gesetzt ist (einen ULDbType.TimeStamp-Wert).

Visual Basic-Syntax

```
Public Function IsColumnCurrentTimestamp(  
    ByVal columnID As Integer  
) As Boolean
```

C#-Syntax

```
public bool IsColumnCurrentTimestamp(int columnID)
```

Parameter

- **columnID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULCursorSchema.ColumnCount-1] liegen. Die erste Spalte in der Tabelle hat den ID-Wert Null.

Rückgabe

TRUE, wenn die Spalte standardmäßig den aktuellen Zeitstempel anzeigt, sonst FALSE.

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)

IsColumnCurrentUTCTimestamp-Methode

Prüft, ob der Standardwert der angegebenen Spalte auf den aktuellen UTC-Zeitstempel gesetzt ist (einen ULDbType.TimeStamp-Wert).

Visual Basic-Syntax

```
Public Function IsColumnCurrentUTCTimestamp(  
    ByVal columnID As Integer  
) As Boolean
```

C#-Syntax

```
public bool IsColumnCurrentUTCTimestamp(int columnID)
```

Parameter

- **columnID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULCursorSchema.ColumnCount-1] liegen. Die erste Spalte in der Tabelle hat den ID-Wert Null.

Rückgabe

TRUE, wenn die Spalte standardmäßig den aktuellen UTC-Zeitstempel anzeigt, sonst FALSE.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)

IsColumnGlobalAutoIncrement-Methode

Prüft, ob der Standardwert der angegebenen Spalte auf GLOBAL AUTOINCREMENT gesetzt ist.

Visual Basic-Syntax

```
Public Function IsColumnGlobalAutoIncrement(  
    ByVal columnID As Integer  
) As Boolean
```

C#-Syntax

```
public bool IsColumnGlobalAutoIncrement(int columnID)
```

Parameter

- **columnID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULCursorSchema.ColumnCount-1] liegen. Die erste Spalte in der Tabelle hat den ID-Wert Null.

Rückgabe

TRUE, wenn für die Spalte global autoincrement festgelegt ist, andernfalls FALSE.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULTableSchema.GetColumnPartitionSize-Methode \[UltraLite.NET\] auf Seite 440](#)
- [ULConnection.DatabaseID-Eigenschaft \[UltraLite.NET\] auf Seite 162](#)
- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)

IsColumnNewUUID-Methode

Prüft, ob eine neue UUID (ein System.Guid-Wert) als Standardwert für die angegebene Spalte festgelegt ist.

Visual Basic-Syntax

```
Public Function IsColumnNewUUID(ByVal columnID As Integer) As Boolean
```

C#-Syntax

```
public bool IsColumnNewUUID(int columnID)
```

Parameter

- **columnID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULCursorSchema.ColumnCount-1] liegen. Die erste Spalte in der Tabelle hat den ID-Wert Null.

Rückgabe

TRUE, wenn die Spalte standardmäßig einen neuen UUID annimmt, sonst FALSE.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)
- [System.Guid](#)

IsColumnNullable-Methode

Prüft, ob die angegebene Spalte nullwertfähig ist.

Visual Basic-Syntax

```
Public Function IsColumnNullable(ByVal columnID As Integer) As Boolean
```

C#-Syntax

```
public bool IsColumnNullable(int columnID)
```

Parameter

- **columnID** Die ID-Nummer der Spalte. Der Wert muss im Bereich [0,ULCursorSchema.ColumnCount-1] liegen. Die erste Spalte in der Tabelle hat den ID-Wert Null.

Rückgabe

TRUE, wenn die Spalte nullwertfähig ist, sonst FALSE.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Siehe auch

- [ULCursorSchema.ColumnCount-Eigenschaft \[UltraLite.NET\] auf Seite 211](#)

IsInPublication-Methode

Prüft, ob die Tabelle in der genannten Publikation enthalten ist.

Visual Basic-Syntax

```
Public Function IsInPublication(ByVal pubName As String) As Boolean
```

C#-Syntax

```
public bool IsInPublication(string pubName)
```

Parameter

- **pubName** Der Name der Publikation

Rückgabe

TRUE, wenn die Tabelle in die Publikation aufgenommen ist, sonst FALSE.

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

IndexCount-Eigenschaft

Gibt die Anzahl der Indizes der Tabelle zurück.

Visual Basic-Syntax

```
Public ReadOnly Property IndexCount As Integer
```

C#-Syntax

```
public int IndexCount {get;}
```

Bemerkungen

Die Anzahl der Indizes für die Tabelle oder 0, wenn das Tabellenschema geschlossen ist.

Index-IDs reichen von 1 bis einschließlich zum IndexCount-Wert.

Hinweis

Index-IDs und -anzahl können sich während eines Schema-Upgrades ändern. Um einen Index korrekt zu identifizieren, greifen Sie auf ihn anhand des Namens zu oder aktualisieren die IDs und Anzahl im Cache nach jedem Schema-Upgrade.

IsNeverSynchronized-Eigenschaft

Prüft, ob die Tabelle als nie zu synchronisieren markiert ist.

Visual Basic-Syntax

```
Public ReadOnly Property IsNeverSynchronized As Boolean
```

C#-Syntax

```
public bool IsNeverSynchronized {get;}
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

TRUE, wenn diese Tabelle als nie zu synchronisieren markiert ist, sonst FALSE.

Tabellen, die als nie zu synchronisieren markiert sind, werden niemals in die Synchronisation einbezogen, selbst wenn sie in einer Publikation enthalten sind. Diese Tabellen werden manchmal "No-Sync-Tabellen" genannt.

Name-Eigenschaft

Gibt den Namen der Tabelle zurück.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property Name As String
```

C#-Syntax

```
public override string Name {get;}
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Der Name der Tabelle als Zeichenfolge.

PrimaryKey-Eigenschaft

Gibt das Indexschema des Primärschlüssels der Tabelle zurück.

Visual Basic-Syntax

```
Public ReadOnly Property PrimaryKey As ULIndexSchema
```

C#-Syntax

```
public ULIndexSchema PrimaryKey {get;}
```

Ausnahmen

- **ULException-Klasse** Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

Ein ULIndexSchema-Objekt, das den Primärschlüssel für die Tabelle repräsentiert.

Siehe auch

- [ULIndexSchema-Klasse \[UltraLite.NET\] auf Seite 301](#)

UploadUnchangedRows-Eigenschaft

Prüft, ob die Datenbank Zeilen ausliest, die nicht geändert wurden.

Visual Basic-Syntax

```
Public ReadOnly Property UploadUnchangedRows As Boolean
```

C#-Syntax

```
public bool UploadUnchangedRows {get;}
```

Ausnahmen

- [ULException-Klasse](#) Es ist ein SQL-Fehler aufgetreten.

Bemerkungen

TRUE, wenn die Tabelle für den Upload aller Zeilen bei der Synchronisation markiert ist; FALSE, wenn nur geänderte Zeilen ausgelesen werden sollen.

So markierte Tabellen senden bei der Synchronisation der Tabelle sowohl veränderte als auch unveränderte Zeilen. Diese Tabellen werden manchmal "All-Sync-Tabellen" genannt.

ULTransaction-Klasse

Repräsentiert eine SQL-Transaktion.

Visual Basic-Syntax

```
Public NotInheritable Class ULTransaction  
Inherits System.Data.Common.DbTransaction
```

C#-Syntax

```
public sealed class ULTransaction : System.Data.Common.DbTransaction
```

Basisklassen

- [System.Data.Common.DbTransaction](#)

Mitglieder

Alle Mitglieder der ULTransaction-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
Commit-Methode	Schreibt die Datenbanktransaktion fest.
Dispose-Methode (geerbt aus System.Data.Common.DbTransaction)	Gibt die nicht verwalteten Ressourcen frei, die von System.Data.Common.DbTransaction verwendet wurden

Name	Beschreibung
Rollback-Methode	Setzt die noch nicht festgeschriebenen Änderungen der Transaktion in der Datenbank zurück.
Connection-Eigenschaft	Gibt die dieser Transaktion zugeordnete Verbindung zurück.
IsolationLevel-Eigenschaft	Gibt die Isolationsstufe für die Transaktion zurück.

Bemerkungen

Es gibt keinen Konstruktor für die ULTransaction-Klasse. Ein ULTransaction-Objekt erhalten Sie mithilfe der ULConnection.BeginTransaction-Methode. Um einen Befehl mit einer Transaktion zu verbinden, verwenden Sie die ULCommand.Transaction-Eigenschaft.

Wenn eine Transaktion festgeschrieben oder zurückgesetzt wurde, kehrt die Verbindung in den Modus zurück, bei dem alle ausgeführten Vorgänge automatisch festgeschrieben werden. Um mehrere Vorgänge zu gruppieren, muss eine neue Transaktion erstellt werden.

Siehe auch

- [ULConnection.BeginTransaction-Methode \[UltraLite.NET\] auf Seite 129](#)
- [ULCommand.Transaction-Eigenschaft \[UltraLite.NET\] auf Seite 108](#)
- [System.Data.Common.DbTransaction](#)
- [System.Data.IDbTransaction](#)
- [System.IDisposable](#)

Commit-Methode

Schreibt die Datenbanktransaktion fest.

Visual Basic-Syntax

```
Public Overrides Sub Commit()
```

C#-Syntax

```
public override void Commit()
```

Bemerkungen

Wenn eine Transaktion festgeschrieben oder zurückgesetzt wurde, kehrt die Verbindung in den Modus zurück, bei dem alle ausgeführten Vorgänge automatisch festgeschrieben werden. Um mehrere Vorgänge zu gruppieren, muss eine neue Transaktion erstellt werden.

Wenn die Commit-Methode aufgrund eines Datenbankfehlers fehlschlägt (z.B. aufgrund eines Fehlers der referenziellen Integrität), bleibt die Transaktion aktiv. Beheben Sie den Fehler und rufen Sie die Commit-Methode erneut auf oder rufen Sie ULTransaction.Rollback-Methode auf, um die Transaktion abzuschließen.

Siehe auch

- [ULTransaction.Rollback-Methode \[UltraLite.NET\] auf Seite 452](#)

Rollback-Methode

Setzt die noch nicht festgeschriebenen Änderungen der Transaktion in der Datenbank zurück.

Visual Basic-Syntax

```
Public Overrides Sub Rollback()
```

C#-Syntax

```
public override void Rollback()
```

Bemerkungen

Wenn eine Transaktion festgeschrieben oder zurückgesetzt wurde, kehrt die Verbindung in den Modus zurück, bei dem alle ausgeführten Vorgänge automatisch festgeschrieben werden. Um mehrere Vorgänge zu gruppieren, muss eine neue Transaktion erstellt werden.

Siehe auch

- [ULTransaction.Commit-Methode \[UltraLite.NET\] auf Seite 451](#)

Connection-Eigenschaft

Gibt die dieser Transaktion zugeordnete Verbindung zurück.

Visual Basic-Syntax

```
Public ReadOnly Shadows Property Connection As ULConnection
```

C#-Syntax

```
public new ULConnection Connection {get;}
```

Bemerkungen

Das ULConnection-Objekt, das mit der Transaktion verbunden ist, oder eine Nullreferenz ("Nothing" in Visual Basic), wenn die Transaktion nicht mehr gültig ist.

Dies ist die stark typisierte Version der Eigenschaften System.Data.IDbTransaction.Connection und System.Data.Common.DbCommand.Connection.

Siehe auch

- [ULConnection.BeginTransaction-Methode \[UltraLite.NET\] auf Seite 129](#)
- [ULConnection-Klasse \[UltraLite.NET\] auf Seite 121](#)
- [System.Data.IDbTransaction.Connection](#)
- [System.Data.Common.DbCommand.Connection](#)

IsolationLevel-Eigenschaft

Gibt die Isolationsstufe für die Transaktion zurück.

Visual Basic-Syntax

```
Public ReadOnly Overrides Property IsolationLevel As IsolationLevel
```

C#-Syntax

```
public override IsolationLevel IsolationLevel {get;}
```

Bemerkungen

Einer der System.Data.IsolationLevel-Werte. UltraLite.NET unterstützt nur den System.Data.IsolationLevel.ReadUncommitted-Wert.

Siehe auch

- [ULConnection.BeginTransaction-Methode \[UltraLite.NET\] auf Seite 129](#)
- [System.Data.IsolationLevel](#)

ULInfoMessageEventHandler-Delegat

Stellt die Methode dar, die das ULConnection.InfoMessage-Ereignis verarbeitet.

Visual Basic-Syntax

```
Public Delegate Sub ULInfoMessageEventHandler(  
    ByVal obj As Object,  
    ByVal args As ULInfoMessageEventArgs  
)
```

C#-Syntax

```
public delegate void ULInfoMessageEventHandler(  
    object obj,  
    ULInfoMessageEventArgs args  
);
```

Parameter

- **obj** Die Verbindung, die das Ereignis sendet.
- **args** Das ULInfoMessageEventArgs-Objekt, das die Ereignisdaten enthält.

Siehe auch

- [ULConnection.InfoMessage-Ereignis \[UltraLite.NET\] auf Seite 166](#)
- [ULInfoMessageEventArgs-Klasse \[UltraLite.NET\] auf Seite 308](#)

ULRowUpdatedEventHandler-Delegat

Stellt die Methode dar, die das ULDataAdapter.RowUpdated-Ereignis verarbeitet.

Visual Basic-Syntax

```
Public Delegate Sub ULRowUpdatedEventHandler(  
    ByVal sender As Object,  
    ByVal e As ULRowUpdatedEventArgs  
)
```

C#-Syntax

```
public delegate void ULRowUpdatedEventHandler(  
    object sender,  
    ULRowUpdatedEventArgs e  
);
```

Parameter

- **sender** Die Verbindung, die das Ereignis sendet.
- **e** Das ULRowUpdatedEventArgs-Objekt, das die Ereignisdaten enthält.

Siehe auch

- [ULDataAdapter.RowUpdated-Ereignis \[UltraLite.NET\] auf Seite 222](#)
- [ULRowUpdatedEventArgs-Klasse \[UltraLite.NET\] auf Seite 378](#)

ULRowUpdatingEventHandler-Delegat

Stellt die Methode dar, die das ULDataAdapter.RowUpdating-Ereignis verarbeitet.

Visual Basic-Syntax

```
Public Delegate Sub ULRowUpdatingEventHandler(  
    ByVal sender As Object,  
    ByVal e As ULRowUpdatingEventArgs  
)
```

C#-Syntax

```
public delegate void ULRowUpdatingEventHandler(  
    object sender,  
    ULRowUpdatingEventArgs e  
);
```

Parameter

- **sender** Die Verbindung, die das Ereignis sendet.
- **e** Das ULRowUpdatingEventArgs-Objekt, das die Ereignisdaten enthält.

Siehe auch

- [ULDataAdapter.RowUpdating-Ereignis \[UltraLite.NET\] auf Seite 222](#)
- [ULRowUpdatingEventArgs-Klasse \[UltraLite.NET\] auf Seite 381](#)

ULRowsCopiedEventHandler-Delegat

Stellt die Methode dar, die das ULBulkCopy.ULRowsCopied-Ereignis verarbeitet.

Visual Basic-Syntax

```
Public Delegate Sub ULRowsCopiedEventHandler(  
    ByVal sender As Object,  
    ByVal rowsCopiedEventArgs As ULRowsCopiedEventArgs  
)
```

C#-Syntax

```
public delegate void ULRowsCopiedEventHandler(  
    object sender,  
    ULRowsCopiedEventArgs rowsCopiedEventArgs  
);
```

Bemerkungen

Der Delegat ULRowsCopiedEventHandler ist in .NET Compact Framework 2.0 nicht verfügbar.

Siehe auch

- [ULBulkCopy.ULRowsCopied-Ereignis \[UltraLite.NET\] auf Seite 56](#)
- [System.Object](#)

ULSyncProgressedDlg-Delegat

Repräsentiert die Methode, die während der Synchronisation mit Informationen zum Synchronisationsfortschritt aufgerufen wird.

Visual Basic-Syntax

```
Public Delegate Sub ULSyncProgressedDlg(  
    ByVal result As IAsyncResult,  
    ByVal data As ULSyncProgressData  
)
```

C#-Syntax

```
public delegate void ULSyncProgressedDlg(  
    IAsyncResult result,  
    ULSyncProgressData data  
);
```

Parameter

- **result** Das von der BeginSynchronize-Methode zurückgegebene IAsyncResult-Objekt. Verwenden Sie result.AsyncState, um auf das für die BeginSynchronize-Methode angegebene Objekt zuzugreifen.
- **data** Ein ULSyncProgressData-Objekt, das die neuesten Fortschrittsdaten zur Synchronisation enthält.

Bemerkungen

Es wird empfohlen, mit der Benutzeroberfläche zu arbeiten oder UltraLite.NET-API-Aufrufe nach dieser Methode durchzuführen. Die Synchronisation wird bei Aufrufen nach dieser Methode nicht aufrechterhalten.

Siehe auch

- [ULConnection.BeginSynchronize-Methode \[UltraLite.NET\] auf Seite 127](#)
- [ULSyncProgressData-Klasse \[UltraLite.NET\] auf Seite 399](#)

ULAuthStatusCode-Enumeration

UL-Erw.: Zählt die Statuscodes auf, die während der MobiLink-Benutzerauthentifizierung ausgegeben werden können.

Visual Basic-Syntax

```
Public Enum ULAuthStatusCode
```

C#-Syntax

```
public enum ULAuthStatusCode
```

Mitglieder

Mitgliedsname	Beschreibung	Wert
UNKNOWN	Der Autorisierungsstatus ist unbekannt, möglicherweise weil die Verbindung noch keine Synchronisation ausgeführt hat (UNKNOWN = 0).	0
VALID	Benutzer-ID und Kennwort waren zur Zeit der Synchronisation gültig (VALID = 1).	1
VALID_BUT_EXPIRES_SOON	Benutzer-ID und Kennwort waren zum Zeitpunkt der Synchronisation gültig, laufen aber demnächst ab (VALID_BUT_EXPIRES_SOON = 2).	2
EXPIRED	Die Benutzer-ID oder das Kennwort ist abgelaufen - die Autorisierung ist fehlgeschlagen (EXPIRED = 3).	3

Mitgliedsname	Beschreibung	Wert
INVALID	Falsche Benutzer-ID oder falsches Kennwort - die Autorisierung ist fehlgeschlagen (INVALID = 4).	4
IN_USE	Die Benutzer-ID wird bereits verwendet - die Autorisierung ist fehlgeschlagen (IN_USE = 5).	5

Siehe auch

- [ULSyncResult.AuthStatus-Eigenschaft \[UltraLite.NET\] auf Seite 411](#)

ULBulkCopyOptions-Enumeration

Ein bitweiser Parameter, der eine oder mehrere Optionen festlegt, die mit einer Instanz der ULBulkCopy-Klasse verwendet werden sollen.

Visual Basic-Syntax

```
Public Enum ULBulkCopyOptions
```

C#-Syntax

```
public enum ULBulkCopyOptions
```

Mitglieder

Mitgliedsname	Beschreibung	Wert
default	Wenn nur diese Option angegeben wird, wird das Standardverhalten angewendet.	0x0
KeepIdentity	Wenn diese Option angegeben wird, werden die Quellwerte, die in eine Identity-Spalte kopiert werden sollen, beibehalten. Standardmäßig werden in der Zieltabelle neue Identity-Werte generiert.	0x1
UseInternalTransaction	Wenn diese Option angegeben wird, wird jeder Batch des Massensimport/-export-Vorgangs innerhalb einer Transaktion ausgeführt. Wenn die Option nicht angegeben wird, werden keine Transaktionen verwendet. Wenn Sie diese Option verwenden und für den Konstruktor auch ein ULTransaction-Objekt bereitstellen, tritt eine System.ArgumentException-Ausnahmebedingung auf.	0x2

Bemerkungen

Die ULBulkCopyOptions-Klasse ist in .NET Compact Framework 2.0 nicht verfügbar.

Die ULBulkCopyOptions-Enumeration wird verwendet, wenn eine ULBulkCopy-Instanz erstellt werden soll, um festzulegen, wie sich WriteToServer-Methoden verhalten.

Siehe auch

- [ULBulkCopy-Klasse \[UltraLite.NET\] auf Seite 46](#)

ULDBValid-Enumeration

Enumeriert die UltraLite.NET-Datenbankvalidierungsmethoden.

Visual Basic-Syntax

```
Public Enum ULDBValid
```

C#-Syntax

```
public enum ULDBValid
```

Mitglieder

Mitgliedsname	Beschreibung	Wert
EXPRESS_VALIDATE	Führt eine schnellere, aber weniger gründliche Validierung aus.	0
FULL_VALIDATE	Validiert Tabellen, Indizes und alle Datenbankseiten.	1

Siehe auch

- [ULDatabaseManager.ValidateDatabase-Methode \[UltraLite.NET\] auf Seite 228](#)
- [ULConnection.ValidateDatabase-Methode \[UltraLite.NET\] auf Seite 158](#)

ULDateOrder-Enumeration

UL-Erw.: Enumeriert die Datumsreihenfolgen, die eine Datenbank unterstützen kann.

Visual Basic-Syntax

```
Public Enum ULDateOrder
```

C#-Syntax

```
public enum ULDateOrder
```

Mitglieder

Mitgliedsname	Beschreibung
YMD	Das Jahr, gefolgt vom Monat und vom Tag des Monats.
MDY	Der Monat, gefolgt vom Tag des Monats und vom Jahr.

Mitgliedsname	Beschreibung
DMY	Der Tag des Monats, gefolgt vom Monat und vom Jahr.

ULDbType-Enumeration

Zählt die Datentypen der UltraLite.NET-Datenbank auf.

Visual Basic-Syntax

```
Public Enum ULDbType
```

C#-Syntax

```
public enum ULDbType
```

Mitglieder

Mitgliedsname	Beschreibung	Wert
BigInt	64-Bit-Ganzzahl mit Vorzeichen.	5
Binary	Binärdaten mit einer festgelegten maximalen Länge. Die Enumerationswerte Binary und VarBinary sind Aliasnamen füreinander.	15
Bit	1-Bit-Parameter.	8
Char	Zeichendaten mit einer definierten Länge. In UltraLite.NET unterstützt dieser Typ immer Unicode-Zeichen. Die Typen Char und VarChar sind voll kompatibel.	0
Datum	Datumsinformationen.	10
Datum/Uhrzeit	Zeitstempelinformationen ("date", "time"). Die Enumerationswerte DateTime und TimeStamp sind Aliasnamen füreinander.	9
Decimal	Exakte numerische Daten mit festgelegter Gesamtstellenzahl und festgelegten Dezimalstellen. Die Enumerationswerte Decimal und Numeric sind Aliasnamen füreinander.	14

Mitgliedsname	Beschreibung	Wert
Double	Gleitkommazahl mit doppelter Genauigkeit (8 Byte).	12
Float	Gleitkommazahl mit einfacher Genauigkeit (4 Byte). Die Enumerationswerte Float und Real sind Aliasnamen füreinander.	13
Integer	32-Bit-Ganzzahl ohne Vorzeichen.	1
LongBinary	Binärdaten mit variabler Länge.	18
LongVarchar	Zeichendaten mit variabler Länge. In UltraLite.NET unterstützt dieser Typ immer Unicode-Zeichen.	17
Numeric	Exakte numerische Daten mit festgelegter Gesamtstellenzahl und festgelegten Dezimalstellen. Die Enumerationswerte Decimal und Numeric sind Aliasnamen füreinander.	14
Real	Gleitkommazahl mit einfacher Genauigkeit (4 Byte). Die Enumerationswerte Float und Real sind Aliasnamen füreinander.	13
SmallInt	16-Bit-Ganzzahl mit Vorzeichen.	3
STGeometry	ST_Geometry-Informationen.	ULNET_TYPE_ST_GEOMETRY
Time	Zeitinformationen.	11
TimeStamp	Zeitstempelinformationen ("date", "time"). Die Enumerationswerte DateTime und TimeStamp sind Aliasnamen füreinander.	9
TimeStampWith-TimeZone	Zeitstempelinformationen (Datum, Zeit) sowie Zeitzonen-Offset.	ULNET_TYPE_TIMESTAMP_WITH_TIME_ZONE
TinyInt	8-Bit-Ganzzahl ohne Vorzeichen.	7

Mitgliedsname	Beschreibung	Wert
UniqueIdentifier	Universally Unique Identifier (UUID/GUID)	19
UnsignedBigInt	64-Bit-Ganzzahl ohne Vorzeichen.	6
UnsignedInt	32-Bit-Ganzzahl ohne Vorzeichen.	2
UnsignedSmallInt	16-Bit-Ganzzahl ohne Vorzeichen.	4
VarBinary	Binärdaten mit einer festgelegten maximalen Länge. Die Enumerationswerte Binary und VarBinary sind Aliasnamen füreinander.	15
VarChar	Zeichendaten mit einer festgelegten maximalen Länge. In UltraLite.NET unterstützt dieser Typ immer Unicode-Zeichen. Die Typen Char und VarChar sind voll kompatibel.	16

Bemerkungen

Die folgende Tabelle listet auf, welche .NET-Typen mit den einzelnen ULDbType-Typen kompatibel sind. Bei integralen Typen können Tabellenspalten immer mit kleineren Integer-Typen und auch größeren Typen festgelegt werden, sofern der tatsächliche Wert innerhalb des Bereichs des Typs liegt.

ULDbType	Kompatibler .NET-Typ	Integrierter C#-Typ	Integrierter Visual Basic-Typ
Binary, VarBinary	System.Byte[] oder System.Guid, wenn die Größe 16 ist	byte[]	Byte()
Bit	System.Boolean	bool	Boolean
Char, VarChar	System.String	String	String
Date	System.DateTime	DateTime (kein integrierter Typ)	Date
Double	System.Double	double	Double
LongBinary	System.Byte[]	byte[]	Byte()
LongVarchar	System.String	String	String

ULDbType	Kompatibler .NET-Typ	Integrierter C#-Typ	Integrierter Visual Basic-Typ
Decimal, Numeric	System.Decimal	decimal	Decimal
Float, Real	System.Single	float	Single
BigInt	System.Int64	long	Long
Integer	System.Int32	int	Integer
SmallInt	System.Int16	short	Short
ST_Geometry	System.String	String	String
Time	System.TimeSpan	TimeSpan (kein integrierter Typ)	TimeSpan (kein integrierter Typ)
DateTime, Time-Stamp	System.DateTime	DateTime (kein integrierter Typ)	Date
TimeStampWithTimeZone	System.String	String	String
TinyInt	System.Byte	byte	Byte
UnsignedBigInt	System.UInt64	ulong	UInt64 (kein integrierter Typ)
UnsignedInt	System.UInt32	uint	UInt32 (kein integrierter Typ)
UnsignedSmallInt	System.UInt16	ushort	UInt16 (kein integrierter Typ)
UniqueIdentifier	System.Guid	Guid (kein integrierter Typ)	Guid (kein integrierter Typ)

Binärspalten der Länge 16 sind voll mit dem Typ UniqueIdentifier kompatibel.

Siehe auch

- [ULDataReader.GetFieldType-Methode \[UltraLite.NET\] auf Seite 252](#)
- [ULDataReader.GetDataTypeName-Methode \[UltraLite.NET\] auf Seite 249](#)
- [ULCursorSchema.GetColumnULDbType-Methode \[UltraLite.NET\] auf Seite 210](#)
- [System.Byte](#)
- [System.Guid](#)
- [System.Boolean](#)
- [System.String](#)
- [System.DateTime](#)
- [System.Single](#)
- [System.Int64](#)
- [System.Int32](#)
- [System.Int16](#)
- [System.TimeSpan](#)
- [System.UInt64](#)
- [System.UInt32](#)
- [System.UInt16](#)

ULRuntimeType-Enumeration

UL-Erw.: Zählt die Typen der UltraLite.NET-Laufzeitumgebungen auf.

Visual Basic-Syntax

```
Public Enum ULRuntimeType
```

C#-Syntax

```
public enum ULRuntimeType
```

Mitglieder

Mitgliedsname	Beschreibung	Wert
STANDALONE_UL	Wählt die Standalone-UltraLite.NET-Laufzeitumgebung aus. Die Standalone-Laufzeitumgebung greift direkt auf Datenbanken zu. Der Datenbankzugriff ist auf diese Weise schneller, doch es kann nicht gemeinsam auf sie zugegriffen werden.	0
UL_ENGINE_CLIENT	Wählt die UltraLite-Laufzeit-Engine Der UltraLite.NET-Engine-Client kommuniziert mit der UltraLite-Engine, um auf Datenbanken zuzugreifen. Datenbanken können von unterschiedlichen Anwendungen gemeinsam genutzt werden.	1

Siehe auch

- [ULDatabaseManager.RuntimeType-Eigenschaft \[UltraLite.NET\] auf Seite 229](#)

ULSqlProgressState-Enumeration

UL-Erw.: Enumeriert alle Zustände, die beim Ausführen von SQL-Passthrough-Skripten auftreten können.

Visual Basic-Syntax

```
Public Enum ULSqlProgressState
```

C#-Syntax

```
public enum ULSqlProgressState
```

Mitglieder

Mitgliedsname	Beschreibung	Wert
STATE_STARTING	Es wurden noch keine Skripten ausgeführt.	0
STATE_RUNNING_SCRIPT	Derzeit läuft ein SQL-Passthrough-Skript.	1
STATE_DONE	Skripten wurden erfolgreich abgeschlossen.	2
STATE_ERROR	Skripten wurden abgeschlossen, aber ein Fehler ist aufgetreten.	3

Siehe auch

- [ULSqlProgressData-Klasse \[UltraLite.NET\] auf Seite 386](#)

ULStreamType-Enumeration

UL-Erw.: Listet die Typen der MobiLink-Synchronisationsdatenströme für die Verwendung bei der Synchronisation auf.

Visual Basic-Syntax

```
Public Enum ULStreamType
```

C#-Syntax

```
public enum ULStreamType
```

Mitglieder

Mitgliedsname	Beschreibung
TCP/IP	Synchronisieren über TCP/IP.

Mitgliedsname	Beschreibung
HTTP	Synchronisieren über HTTP. Der HTTP-Datenstrom verwendet TCP/IP als zugrunde liegendes Transportmittel. UltraLite-Anwendungen fungieren als Webbrowser und der MobiLink-Server als Webserver. UltraLite-Anwendungen senden POST-Anforderungen, um Daten an den Server zu senden, und GET-Anforderungen, um Daten vom Server zu lesen.
HTTPS	Synchronisieren über HTTPS (HTTP mit Transportschichtssicherheit)
TLS	Synchronisieren über TCP/IP mit Transportschichtssicherheit

Bemerkungen

Hinweis

Erforderliche getrennt lizenzierbare Komponenten.

FIPS-zertifizierte Verschlüsselung erfordert eine separate Lizenz. Alle Technologien für starke Verschlüsselungen unterliegen Exportbestimmungen.

Siehe „Getrennt lizenzierbare Komponenten“ [[SQL Anywhere 16 - Einführung](#)].

Siehe auch

- [ULSyncParams.Stream-Eigenschaft \[UltraLite.NET\] auf Seite 396](#)
- „UltraLite-Netzwerkprotokolloptionen für dbmsync“ [[UltraLite - Datenbankverwaltung](#)]

ULSyncProgressState-Enumeration

UL-Erw.: Listet alle Statuswerte auf, die bei der Synchronisation auftreten können.

Visual Basic-Syntax

```
Public Enum ULSyncProgressState
```

C#-Syntax

```
public enum ULSyncProgressState
```

Mitglieder

Mitgliedsname	Beschreibung	Wert
STATE_STARTING	Es wurden noch keine Synchronisationsaktionen unternommen.	0
STATE_CONNECTING	Der Synchronisationsdatenstrom wurde erstellt, aber noch nicht geöffnet.	1

Mitgliedsname	Beschreibung	Wert
STATE_SENDING_HEADER	Der Synchronisationsdatenstrom wurde geöffnet und der Header wird demnächst gesendet.	2
STATE_SENDING_TABLE	Es wird gerade eine Tabelle gesendet. Der Verarbeitungsfortschritt kann mit den Eigenschaften <code>ULSyncProgressData.SyncTableIndex</code> und <code>ULSyncProgressData.SyncTableCount</code> überwacht werden.	3
STATE_SENDING_DATA	Es werden gerade Daten für die aktuelle Tabelle gesendet. Die Eigenschaften <code>ULSyncProgressData.SentBytes</code> , <code>ULSyncProgressData.SentInserts</code> , <code>ULSyncProgressData.SentUpdates</code> und <code>ULSyncProgressData.SentDeletes</code> wurden aktualisiert.	4
STATE_FINISHING_UPLOAD	Der Upload wird gerade abgeschlossen. Die endgültige Anzahl der gesendeten Zeilen wird in dieses Ereignis einbezogen.	5
STATE_RECEIVING_UPLOAD_ACK	Es wird gerade eine Bestätigung darüber empfangen, dass der Sendevorgang abgeschlossen wurde.	6
STATE_RECEIVING_TABLE	Es wird gerade eine Tabelle empfangen. Der Verarbeitungsfortschritt kann mit den Eigenschaften <code>ULSyncProgressData.SyncTableIndex</code> und <code>ULSyncProgressData.SyncTableCount</code> überwacht werden.	7
STATE_RECEIVING_DATA	Es werden gerade Daten für die aktuelle Tabelle empfangen. Die Eigenschaften <code>ULSyncProgressData.ReceivedBytes</code> , <code>ULSyncProgressData.ReceivedInserts</code> , <code>ULSyncProgressData.ReceivedUpdates</code> und <code>ULSyncProgressData.ReceivedDeletes</code> wurden aktualisiert.	8
STATE_COMMITTING_DOWNLOAD	Der Download wird gerade festgeschrieben. Die endgültige Anzahl der empfangenen Zeilen wird in dieses Ereignis einbezogen.	9
STATE_ROLLING_BACK_DOWNLOAD	Die Synchronisation setzt den Download zurück, weil während des Downloads ein Fehler aufgetreten ist. Der Fehler wird zusammen mit einem nachfolgenden <code>STATE_ERROR</code> -Fortschrittsbericht übermittelt.	10

Mitgliedsname	Beschreibung	Wert
STATE_SENDING_DOWNLOAD_ACK	Eine Bestätigung des Download-Abschlusses wird gesendet.	11
STATE_DISCONNECTING	Der Synchronisationsdatenstrom wird demnächst geschlossen.	12
STATE_DONE	Die Synchronisation wurde erfolgreich ausgeführt.	13
STATE_ERROR	Die Synchronisation wurde abgeschlossen, jedoch mit einem Fehler.	14
STATE_CANCELLED	Die Synchronisation wurde abgebrochen.	15

Siehe auch

- [ULSyncProgressData-Klasse \[UltraLite.NET\] auf Seite 399](#)
- [ULSyncProgressData.ReceivedBytes-Eigenschaft \[UltraLite.NET\] auf Seite 403](#)
- [ULSyncProgressData.ReceivedInserts-Eigenschaft \[UltraLite.NET\] auf Seite 404](#)
- [ULSyncProgressData.ReceivedUpdates-Eigenschaft \[UltraLite.NET\] auf Seite 404](#)
- [ULSyncProgressData.ReceivedDeletes-Eigenschaft \[UltraLite.NET\] auf Seite 403](#)
- [ULSyncProgressData.SentBytes-Eigenschaft \[UltraLite.NET\] auf Seite 404](#)
- [ULSyncProgressData.SentInserts-Eigenschaft \[UltraLite.NET\] auf Seite 405](#)
- [ULSyncProgressData.SentUpdates-Eigenschaft \[UltraLite.NET\] auf Seite 405](#)
- [ULSyncProgressData.SentDeletes-Eigenschaft \[UltraLite.NET\] auf Seite 405](#)
- [ULSyncProgressData.SyncTableIndex-Eigenschaft \[UltraLite.NET\] auf Seite 407](#)
- [ULSyncProgressData.SyncTableCount-Eigenschaft \[UltraLite.NET\] auf Seite 406](#)

Index

A

- Abort-Eigenschaft
 - ULRowsCopiedEventArgs-Klasse [UltraLite.NET-API],377
- ActiveSync, Synchronisation
 - UltraLite.NET,19
- ActiveSyncInvoked-Methode
 - ULActiveSyncListener-Schnittstelle [UltraLite.NET-API],44
- Add-Methode
 - ULBulkCopyColumnMappingCollection-Klasse [UltraLite.NET-API],65
 - ULParameterCollection-Klasse [UltraLite.NET-API],335
- AdditionalParms-Eigenschaft
 - ULConnectionParms-Klasse [UltraLite.NET-API],173
 - ULSyncParms-Klasse [UltraLite.NET-API],390
- AddRange-Methode
 - ULParameterCollection-Klasse [UltraLite.NET-API],341
- Aktualisierung
 - UltraLite.NET, Tabellenzeilen,12
- Aktualisierungsmodus
 - UltraLite.NET,11
- Anwendungen
 - entwickeln für Windows Mobile,3
- AppendBytes-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],357
- AppendChars-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],359
- Architekturen
 - UltraLite.Net,43
- Auswählen, Daten aus Datenbanktabellen
 - UltraLite.NET,8
- AuthenticationParms-Eigenschaft
 - ULFileTransfer-Klasse [UltraLite.NET-API],289
 - ULSyncParms-Klasse [UltraLite.NET-API],390
- AuthStatus-Eigenschaft
 - ULFileTransfer-Klasse [UltraLite.NET-API],290
 - ULSyncResult-Klasse [UltraLite.NET-API],411
- AuthValue-Eigenschaft
 - ULFileTransfer-Klasse [UltraLite.NET-API],290
 - ULSyncResult-Klasse [UltraLite.NET-API],411

- Autocommit, Modus
 - UltraLite.NET- Entwicklung,16

B

- BatchSize-Eigenschaft
 - ULBulkCopy-Klasse [UltraLite.NET-API],54
- BeginExecuteNonQuery-Methode
 - ULCommand-Klasse [UltraLite.NET-API],79
- BeginExecuteReader-Methode
 - ULCommand-Klasse [UltraLite.NET-API],81
- BeginSynchronize-Methode
 - ULConnection-Klasse [UltraLite.NET-API],127
- BeginTransaction-Methode
 - ULConnection-Klasse [UltraLite.NET-API],129
- Blättern
 - UltraLite.NET, Tabellen-API,10
- BulkCopyTimeout-Eigenschaft
 - ULBulkCopy-Klasse [UltraLite.NET-API],54
- BytesReceived-Eigenschaft
 - ULFileTransferProgressData-Klasse [UltraLite.NET-API],298

C

- CacheSize-Eigenschaft
 - ULConnectionParms-Klasse [UltraLite.NET-API],175
 - ULConnectionStringBuilder-Klasse [UltraLite.NET-API],186
- Cancel-Methode
 - ULCommand-Klasse [UltraLite.NET-API],85
- CancelGetNotification-Methode
 - ULConnection-Klasse [UltraLite.NET-API],132
- CancelSynchronize-Methode
 - ULConnection-Klasse [UltraLite.NET-API],133
- CanCreateDataSourceEnumerator-Eigenschaft
 - ULFactory-Klasse [UltraLite.NET-API],281
- CaseSensitive-Eigenschaft
 - ULCreateParms-Klasse [UltraLite.NET-API],198
- Casting
 - UltraLite.NET, Datentypen in,15
- ChangeDatabase-Methode
 - ULConnection-Klasse [UltraLite.NET-API],133
- ChangeEncryptionKey-Methode
 - ULConnection-Klasse [UltraLite.NET-API],134
- ChangePassword-Methode
 - ULConnection-Klasse [UltraLite.NET-API],134
- ChecksumLevel-Eigenschaft

- ULCreateParms-Klasse [UltraLite.NET-API],198
 - Clear-Methode
 - ULParameterCollection-Klasse [UltraLite.NET-API],342
 - Close-Methode
 - ULBulkCopy-Klasse [UltraLite.NET-API],51
 - ULConnection-Klasse [UltraLite.NET-API],135
 - ULDataReader-Klasse [UltraLite.NET-API],243
 - ColumnCount-Eigenschaft
 - ULCursorSchema-Klasse [UltraLite.NET-API],211
 - ULIndexSchema-Klasse [UltraLite.NET-API],303
 - ColumnMappings-Eigenschaft
 - ULBulkCopy-Klasse [UltraLite.NET-API],55
 - Columns-Eigenschaft
 - ULMetaDataCollectionNames-Klasse [UltraLite.NET-API],311
 - Command-Eigenschaft
 - ULRowUpdatedEventArgs-Klasse [UltraLite.NET-API],380
 - ULRowUpdatingEventArgs-Klasse [UltraLite.NET-API],383
 - CommandText-Eigenschaft
 - ULCommand-Klasse [UltraLite.NET-API],104
 - CommandTimeout-Eigenschaft
 - ULCommand-Klasse [UltraLite.NET-API],104
 - CommandType-Eigenschaft
 - ULCommand-Klasse [UltraLite.NET-API],105
 - Commit-Methode
 - UltraLite.NET-Transaktionen,16
 - ULTransaction-Klasse [UltraLite.NET-API],451
 - Connection, Klasse
 - UltraLite.NET,3
 - Connection-Eigenschaft
 - ULCommand-Klasse [UltraLite.NET-API],106
 - ULTransaction-Klasse [UltraLite.NET-API],452
 - ConnectionString-Eigenschaft
 - ULConnectionParms-Klasse [UltraLite.NET-API],175
 - ULConnectionStringBuilder-Klasse [UltraLite.NET-API],187
 - ConnectionString-Eigenschaft
 - ULConnection-Klasse [UltraLite.NET-API],160
 - ConnectionTimeout-Eigenschaft
 - ULConnection-Klasse [UltraLite.NET-API],161
 - Contains-Methode
 - ULBulkCopyColumnMappingCollection-Klasse [UltraLite.NET-API],69
 - ULParameterCollection-Klasse [UltraLite.NET-API],342
 - ContainsKey-Methode
 - ULConnectionStringBuilder-Klasse [UltraLite.NET-API],184
 - CopyFrom-Methode
 - ULSyncParms-Klasse [UltraLite.NET-API],389
 - CopyTo-Methode
 - ULBulkCopyColumnMappingCollection-Klasse [UltraLite.NET-API],70
 - ULParameterCollection-Klasse [UltraLite.NET-API],343
 - Count-Eigenschaft
 - ULParameterCollection-Klasse [UltraLite.NET-API],348
 - CountUploadRows-Methode
 - ULConnection-Klasse [UltraLite.NET-API],135
 - CreateCommand-Methode
 - ULConnection-Klasse [UltraLite.NET-API],136
 - ULFactory-Klasse [UltraLite.NET-API],278
 - CreateCommandBuilder-Methode
 - ULFactory-Klasse [UltraLite.NET-API],279
 - CreateConnection-Methode
 - ULFactory-Klasse [UltraLite.NET-API],279
 - CreateConnectionStringBuilder-Methode
 - ULFactory-Klasse [UltraLite.NET-API],280
 - CreateDataAdapter-Methode
 - ULFactory-Klasse [UltraLite.NET-API],280
 - CreateDatabase-Methode
 - ULDatabaseManager-Klasse [UltraLite.NET-API],223
 - CreateNotificationQueue-Methode
 - ULConnection-Klasse [UltraLite.NET-API],137
 - CreateParameter-Methode
 - ULCommand-Klasse [UltraLite.NET-API],85
 - ULFactory-Klasse [UltraLite.NET-API],280
 - CurrentDownloadRowCount-Eigenschaft
 - ULSyncProgressData-Klasse [UltraLite.NET-API],401
 - CurrentScript-Eigenschaft
 - ULSqlProgressData-Klasse [UltraLite.NET-API],386
- D**
- DataAdapter-Eigenschaft
 - ULCommandBuilder-Klasse [UltraLite.NET-API],121

- Database-Eigenschaft
 - ULConnection-Klasse [UltraLite.NET-API],161
- DatabaseID-Eigenschaft
 - ULConnection-Klasse [UltraLite.NET-API],162
- DatabaseKey-Eigenschaft
 - ULConnectionStringBuilder-Klasse [UltraLite.NET-API],188
- DatabaseManager, Klasse
 - UltraLite.NET,3
- DatabaseName-Eigenschaft
 - ULConnectionStringBuilder-Klasse [UltraLite.NET-API],188
- DatabaseOnDesktop-Eigenschaft
 - ULConnectionParms-Klasse [UltraLite.NET-API],176
 - ULConnectionStringBuilder-Klasse [UltraLite.NET-API],189
- DatabaseOnDevice-Eigenschaft
 - ULConnectionParms-Klasse [UltraLite.NET-API],176
 - ULConnectionStringBuilder-Klasse [UltraLite.NET-API],190
- DataSource-Eigenschaft
 - ULConnection-Klasse [UltraLite.NET-API],162
- DataSourceInformation-Eigenschaft
 - ULMetaDataCollectionNames-Klasse [UltraLite.NET-API],312
- DataTypes-Eigenschaft
 - ULMetaDataCollectionNames-Klasse [UltraLite.NET-API],313
- DateFormat-Eigenschaft
 - ULCreateParms-Klasse [UltraLite.NET-API],198
- Datenänderung
 - UltraLite.NET mit SQL ,5
- Datenbankschemata
 - UltraLite.NET, Zugriff,17
- Datentypen
 - UltraLite.NET, API-Zugriff und Casting,14
- Datenverarbeitung
 - UltraLite.NET, Tabellen-API,10
- DateOrder-Eigenschaft
 - ULCreateParms-Klasse [UltraLite.NET-API],199
- DbType-Eigenschaft
 - ULParameter-Klasse [UltraLite.NET-API],327
- DeclareEvent-Methode
 - ULConnection-Klasse [UltraLite.NET-API],137
- Delete-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],360
- DeleteAllRows-Methode
 - ULTable-Klasse [UltraLite.NET-API],422
- DeleteCommand-Eigenschaft
 - ULDataAdapter-Klasse [UltraLite.NET-API],218
- Deployment
 - UltraLite.NET-Anwendungen,20,22
- Depth-Eigenschaft
 - ULDataReader-Klasse [UltraLite.NET-API],269
- DesignTimeVisible-Eigenschaft
 - ULCommand-Klasse [UltraLite.NET-API],106
- DestinationColumn-Eigenschaft
 - ULBulkCopyColumnMapping-Klasse [UltraLite.NET-API],61
- DestinationOrdinal-Eigenschaft
 - ULBulkCopyColumnMapping-Klasse [UltraLite.NET-API],61
- DestinationTableName-Eigenschaft
 - ULBulkCopy-Klasse [UltraLite.NET-API],55
- DestroyNotificationQueue-Methode
 - ULConnection-Klasse [UltraLite.NET-API],138
- Direction-Eigenschaft
 - ULParameter-Klasse [UltraLite.NET-API],327
- Dispose-Methode
 - ULBulkCopy-Klasse [UltraLite.NET-API],51
- DML
 - UltraLite.NET,5
- DownloadFile-Methode
 - ULFileTransfer-Klasse [UltraLite.NET-API],284
- DownloadOnly-Eigenschaft
 - ULSyncParms-Klasse [UltraLite.NET-API],391
- DropDatabase-Methode
 - ULDatabaseManager-Klasse [UltraLite.NET-API],225
- Dynamic SQL
 - praktische Einführung in UltraLite.NET,25

E

- Einfügemodus
 - UltraLite.NET,11
- Einfügen
 - UltraLite.NET, Tabellenzeilen,11
- EncryptionKey-Eigenschaft
 - ULConnectionParms-Klasse [UltraLite.NET-API],177
- EndExecuteNonQuery-Methode
 - ULCommand-Klasse [UltraLite.NET-API],86
- EndExecuteReader-Methode

- ULCommand-Klasse [UltraLite.NET-API],89
- EndSynchronize-Methode
 - ULConnection-Klasse [UltraLite.NET-API],138
- Entwicklung
 - UltraLite.NET,3
- Entwicklungsplattformen
 - UltraLite.NET,1
- EquivalentTo-Methode
 - ULConnectionStringBuilder-Klasse [UltraLite.NET-API],184
- Ergebnismengen
 - UltraLite.NET, Navigation,9
- Ergebnismengenschemata
 - UltraLite.NET,9
- ExecuteNonQuery-Methode
 - ULCommand-Klasse [UltraLite.NET-API],93
- ExecuteReader-Methode
 - ULCommand-Klasse [UltraLite.NET-API],94
- ExecuteResultSet-Methode
 - ULCommand-Klasse [UltraLite.NET-API],97
- ExecuteScalar-Methode
 - ULCommand-Klasse [UltraLite.NET-API],99
- ExecuteTable-Methode
 - ULCommand-Klasse [UltraLite.NET-API],100
 - ULConnection-Klasse [UltraLite.NET-API],139

F

- Fehler
 - UltraLite.NET-API-Behandlung,18
- Fehlerbehandlung
 - UltraLite.NET,18
 - UltraLite.NET, Deployment-Checkliste,25
 - UltraLite.NET, Fehler beheben,18
- Festschreiben
 - UltraLite.NET-Transaktionen,16
- FieldCount-Eigenschaft
 - ULDataReader-Klasse [UltraLite.NET-API],270
- FileAuthCode-Eigenschaft
 - ULFileTransfer-Klasse [UltraLite.NET-API],291
- FileName-Eigenschaft
 - ULFileTransfer-Klasse [UltraLite.NET-API],291
- FileSize-Eigenschaft
 - ULFileTransferProgressData-Klasse [UltraLite.NET-API],298
- FileTransferProgressed-Methode
 - ULFileTransferProgressListener-Schnittstelle [UltraLite.NET-API],300

- FindBegin-Methode
 - ULTable-Klasse [UltraLite.NET-API],422
- FindFirst-Methode
 - ULTable-Klasse [UltraLite.NET-API],423
- FindLast-Methode
 - ULTable-Klasse [UltraLite.NET-API],425
- FindNext-Methode
 - ULTable-Klasse [UltraLite.NET-API],427
- FindPrevious-Methode
 - ULTable-Klasse [UltraLite.NET-API],429
- FIPS-Eigenschaft
 - ULCreateParms-Klasse [UltraLite.NET-API],199
- FLAG_IS_BLOCKING-Feld
 - ULFileTransferProgressData-Klasse [UltraLite.NET-API],299
 - ULSyncProgressData-Klasse [UltraLite.NET-API],409
- Flags-Eigenschaft
 - ULFileTransferProgressData-Klasse [UltraLite.NET-API],299
 - ULSyncProgressData-Klasse [UltraLite.NET-API],401
- ForeignKeys-Eigenschaft
 - ULMetaDataCollectionNames-Klasse [UltraLite.NET-API],313

G

- GetBoolean-Methode
 - ULDataReader-Klasse [UltraLite.NET-API],243
- GetByte-Methode
 - ULDataReader-Klasse [UltraLite.NET-API],244
- GetBytes-Methode
 - ULDataReader-Klasse [UltraLite.NET-API],244
- GetChar-Methode
 - ULDataReader-Klasse [UltraLite.NET-API],247
- GetChars-Methode
 - ULDataReader-Klasse [UltraLite.NET-API],247
- GetColumnDefaultValue-Methode
 - ULTableSchema-Klasse [UltraLite.NET-API],439
- GetColumnID-Methode
 - ULCursorSchema-Klasse [UltraLite.NET-API],206
- GetColumnName-Methode
 - ULCursorSchema-Klasse [UltraLite.NET-API],207
 - ULIndexSchema-Klasse [UltraLite.NET-API],302
- GetColumnPartitionSize-Methode
 - ULTableSchema-Klasse [UltraLite.NET-API],440
- GetColumnPrecision-Methode

ULCursorSchema-Klasse [UltraLite.NET-API],207	GetLastDownloadTime-Methode
GetColumnScale-Methode	ULConnection-Klasse [UltraLite.NET-API],144
ULCursorSchema-Klasse [UltraLite.NET-API],208	GetName-Methode
GetColumnSize-Methode	ULDataReader-Klasse [UltraLite.NET-API],256
ULCursorSchema-Klasse [UltraLite.NET-API],208	GetNewUUID-Methode
GetColumnSQLName-Methode	ULConnection-Klasse [UltraLite.NET-API],144
ULCursorSchema-Klasse [UltraLite.NET-API],209	GetNotification-Methode
GetColumnULDbType-Methode	ULConnection-Klasse [UltraLite.NET-API],145
ULCursorSchema-Klasse [UltraLite.NET-API],210	GetNotificationParameter-Methode
GetDatabaseProperty-Methode	ULConnection-Klasse [UltraLite.NET-API],146
ULDatabaseSchema-Klasse [UltraLite.NET-API],230	GetObjectData-Methode
GetDataTypeName-Methode	ULException-Klasse [UltraLite.NET-API],276
ULDataReader-Klasse [UltraLite.NET-API],249	GetOptimalIndex-Methode
GetDateTime-Methode	ULTableSchema-Klasse [UltraLite.NET-API],442
ULDataReader-Klasse [UltraLite.NET-API],250	GetOrdinal-Methode
GetDecimal-Methode	ULDataReader-Klasse [UltraLite.NET-API],256
ULDataReader-Klasse [UltraLite.NET-API],250	GetPublicationName-Methode
GetDeleteCommand-Methode	ULDatabaseSchema-Klasse [UltraLite.NET-API],233
ULCommandBuilder-Klasse [UltraLite.NET-API],114	GetPublicationPredicate-Methode
GetDouble-Methode	ULTableSchema-Klasse [UltraLite.NET-API],442
ULDataReader-Klasse [UltraLite.NET-API],251	GetRowCount-Methode
GetEnumerator-Methode	ULDataReader-Klasse [UltraLite.NET-API],257
ULDataReader-Klasse [UltraLite.NET-API],251	GetSchema-Methode
ULParameterCollection-Klasse [UltraLite.NET-API],344	ULConnection-Klasse [UltraLite.NET-API],147
GetFieldType-Methode	GetSchemaTable-Methode
ULDataReader-Klasse [UltraLite.NET-API],252	ULCursorSchema-Klasse [UltraLite.NET-API],210
GetFillParameters-Methode	ULDataReader-Klasse [UltraLite.NET-API],258
ULDataAdapter-Klasse [UltraLite.NET-API],218	GetShortName-Methode
GetFloat-Methode	ULConnectionStringBuilder-Klasse [UltraLite.NET-API],185
ULDataReader-Klasse [UltraLite.NET-API],252	GetString-Methode
GetGuid-Methode	ULDataReader-Klasse [UltraLite.NET-API],260
ULDataReader-Klasse [UltraLite.NET-API],253	GetTableName-Methode
GetIndex-Methode	ULDatabaseSchema-Klasse [UltraLite.NET-API],233
ULTableSchema-Klasse [UltraLite.NET-API],441	GetTimeSpan-Methode
GetIndexName-Methode	ULDataReader-Klasse [UltraLite.NET-API],261
ULTableSchema-Klasse [UltraLite.NET-API],441	GetUInt16-Methode
GetInsertCommand-Methode	ULDataReader-Klasse [UltraLite.NET-API],261
ULCommandBuilder-Klasse [UltraLite.NET-API],116	GetUInt32-Methode
GetInt16-Methode	ULDataReader-Klasse [UltraLite.NET-API],262
ULDataReader-Klasse [UltraLite.NET-API],254	GetUInt64-Methode
GetInt32-Methode	ULDataReader-Klasse [UltraLite.NET-API],262
ULDataReader-Klasse [UltraLite.NET-API],254	GetUpdateCommand-Methode
GetInt64-Methode	ULCommandBuilder-Klasse [UltraLite.NET-API],118
ULDataReader-Klasse [UltraLite.NET-API],255	GetValue-Methode

- ULDataReader-Klasse [UltraLite.NET-API],263
- GetValues-Methode
 - ULDataReader-Klasse [UltraLite.NET-API],264
- GlobalAutoIncrementUsage-Eigenschaft
 - ULConnection-Klasse [UltraLite.NET-API],163
- GrantConnectTo-Methode
 - ULConnection-Klasse [UltraLite.NET-API],150

H

- HasRows-Eigenschaft
 - ULDataReader-Klasse [UltraLite.NET-API],270

I

- iAnywhere.Data.UltraLite-Namespace
 - UltraLite.NET-API,43
- IgnoredDeletes-Eigenschaft
 - ULSyncProgressData-Klasse [UltraLite.NET-API],402
- IgnoredRows-Eigenschaft
 - ULSyncResult-Klasse [UltraLite.NET-API],412
- IgnoredUpdates-Eigenschaft
 - ULSyncProgressData-Klasse [UltraLite.NET-API],402
- IndexColumns-Eigenschaft
 - ULMetaDataCollectionNames-Klasse [UltraLite.NET-API],314
- IndexCount-Eigenschaft
 - ULTableSchema-Klasse [UltraLite.NET-API],448
- Indexes-Eigenschaft
 - ULMetaDataCollectionNames-Klasse [UltraLite.NET-API],314
- IndexName-Eigenschaft
 - ULCommand-Klasse [UltraLite.NET-API],107
- IndexOf-Methode
 - ULBulkCopyColumnMappingCollection-Klasse [UltraLite.NET-API],70
 - ULParameterCollection-Klasse [UltraLite.NET-API],344
- Indizes
 - UltraLite.NET, Schemainformationen ,17
- InfoMessage-Ereignis
 - ULConnection-Klasse [UltraLite.NET-API],166
- Insert-Methode
 - ULParameterCollection-Klasse [UltraLite.NET-API],345
 - ULTable-Klasse [UltraLite.NET-API],431
- InsertBegin-Methode

- ULTable-Klasse [UltraLite.NET-API],431
- InsertCommand-Eigenschaft
 - ULDataAdapter-Klasse [UltraLite.NET-API],219
- Instance-Feld
 - ULFactory-Klasse [UltraLite.NET-API],281
- INVALID_DATABASE_ID-Feld
 - ULConnection-Klasse [UltraLite.NET-API],168
- IsBOF-Eigenschaft
 - ULDataReader-Klasse [UltraLite.NET-API],270
- IsCaseSensitive-Eigenschaft
 - ULDatabaseSchema-Klasse [UltraLite.NET-API],235
- IsClosed-Eigenschaft
 - ULDataReader-Klasse [UltraLite.NET-API],271
- IsColumnAutoIncrement-Methode
 - ULTableSchema-Klasse [UltraLite.NET-API],443
- IsColumnCurrentDate-Methode
 - ULTableSchema-Klasse [UltraLite.NET-API],443
- IsColumnCurrentTime-Methode
 - ULTableSchema-Klasse [UltraLite.NET-API],444
- IsColumnCurrentTimestamp-Methode
 - ULTableSchema-Klasse [UltraLite.NET-API],445
- IsColumnCurrentUTCTimestamp-Methode
 - ULTableSchema-Klasse [UltraLite.NET-API],445
- IsColumnDescending-Methode
 - ULIndexSchema-Klasse [UltraLite.NET-API],303
- IsColumnGlobalAutoIncrement-Methode
 - ULTableSchema-Klasse [UltraLite.NET-API],446
- IsColumnNewUUID-Methode
 - ULTableSchema-Klasse [UltraLite.NET-API],446
- IsColumnNullable-Methode
 - ULTableSchema-Klasse [UltraLite.NET-API],447
- IsDBNull-Methode
 - ULDataReader-Klasse [UltraLite.NET-API],264
- IsEOF-Eigenschaft
 - ULDataReader-Klasse [UltraLite.NET-API],271
- IsFinalSyncProgress-Eigenschaft
 - ULSyncProgressData-Klasse [UltraLite.NET-API],402
- IsFixedSize-Eigenschaft
 - ULParameterCollection-Klasse [UltraLite.NET-API],348
- IsForeignKey-Eigenschaft
 - ULIndexSchema-Klasse [UltraLite.NET-API],304
- IsForeignKeyCheckOnCommit-Eigenschaft
 - ULIndexSchema-Klasse [UltraLite.NET-API],304
- IsForeignKeyNullable-Eigenschaft
 - ULIndexSchema-Klasse [UltraLite.NET-API],305

IsInPublication-Methode
 ULTableSchema-Klasse [UltraLite.NET-API],447
IsNeverSynchronized-Eigenschaft
 ULTableSchema-Klasse [UltraLite.NET-API],448
IsNullable-Eigenschaft
 ULParameter-Klasse [UltraLite.NET-API],328
IsolationLevel-Eigenschaft
 ULTransaction-Klasse [UltraLite.NET-API],453
IsOpen-Eigenschaft
 ULCursorSchema-Klasse [UltraLite.NET-API],211
 ULDatabaseSchema-Klasse [UltraLite.NET-API],236
 ULIndexSchema-Klasse [UltraLite.NET-API],305
IsPrimaryKey-Eigenschaft
 ULIndexSchema-Klasse [UltraLite.NET-API],306
IsReadOnly-Eigenschaft
 ULParameterCollection-Klasse [UltraLite.NET-API],348
IsSynchronized-Eigenschaft
 ULParameterCollection-Klasse [UltraLite.NET-API],349
IsUniqueIndex-Eigenschaft
 ULIndexSchema-Klasse [UltraLite.NET-API],306
IsUniqueKey-Eigenschaft
 ULIndexSchema-Klasse [UltraLite.NET-API],306

K

KeepPartialDownload-Eigenschaft
 ULSyncParms-Klasse [UltraLite.NET-API],392

L

LastIdentity-Eigenschaft
 ULConnection-Klasse [UltraLite.NET-API],163
LocalFileName-Eigenschaft
 ULFileTransfer-Klasse [UltraLite.NET-API],291
LocalPath-Eigenschaft
 ULFileTransfer-Klasse [UltraLite.NET-API],292
LookupBackward-Methode
 ULTable-Klasse [UltraLite.NET-API],432
LookupBegin-Methode
 ULTable-Klasse [UltraLite.NET-API],434
LookupForward-Methode
 ULTable-Klasse [UltraLite.NET-API],434
Löschen
 UltraLite.NET, Tabellenzeilen,16

M

MaxHashSize-Eigenschaft
 ULCreateParms-Klasse [UltraLite.NET-API],200
Message-Eigenschaft
 ULInfoMessageEventArgs-Klasse [UltraLite.NET-API],309
MetaDataCollections-Eigenschaft
 ULMetaDataCollectionNames-Klasse [UltraLite.NET-API],315
Modi
 UltraLite.NET,11
MoveAfterLast-Methode
 ULDataReader-Klasse [UltraLite.NET-API],265
MoveBeforeFirst-Methode
 ULDataReader-Klasse [UltraLite.NET-API],265
moveFirst method (Tabellenklasse)
 UltraLite.NET-Entwicklung,10
MoveFirst, Methode
 UltraLite.NET, Datenabfragebeispiel,8
MoveFirst-Methode
 ULDataReader-Klasse [UltraLite.NET-API],266
MoveLast-Methode
 ULDataReader-Klasse [UltraLite.NET-API],266
MoveNext, Methode
 UltraLite.NET, Datenabfragebeispiel,8
moveNext, Methode (Table-Klasse)
 UltraLite.NET-Entwicklung,10
MoveNext-Methode
 ULDataReader-Klasse [UltraLite.NET-API],266
MovePrevious-Methode
 ULDataReader-Klasse [UltraLite.NET-API],267
MoveRelative-Methode
 ULDataReader-Klasse [UltraLite.NET-API],267
Multi-Thread-Anwendungen
 UltraLite.NET,4

N

Nachschlagemethoden
 UltraLite.NET,13
Nachschlagemodus
 UltraLite.NET,11
Name-Eigenschaft
 ULCursorSchema-Klasse [UltraLite.NET-API],211
 ULIndexSchema-Klasse [UltraLite.NET-API],307
 ULResultSetSchema-Klasse [UltraLite.NET-API],376
 ULTableSchema-Klasse [UltraLite.NET-API],449

- NativeError-Eigenschaft
 - ULException-Klasse [UltraLite.NET-API],276
 - ULInfoMessageEventArgs-Klasse [UltraLite.NET-API],310
- Navigation durch SQL-Ergebnismengen
 - UltraLite.NET,9
- Navigieren
 - UltraLite.NET, Tabellen-API,10
- NearestCentury-Eigenschaft
 - ULCreateParms-Klasse [UltraLite.NET-API],200
- NewPassword-Eigenschaft
 - ULSyncParms-Klasse [UltraLite.NET-API],393
- NextResult-Methode
 - ULDataReader-Klasse [UltraLite.NET-API],268
- NotifyAfter-Eigenschaft
 - ULBulkCopy-Klasse [UltraLite.NET-API],56

O

- Obfuscate-Eigenschaft
 - ULCreateParms-Klasse [UltraLite.NET-API],201
- Offset-Eigenschaft
 - ULParameter-Klasse [UltraLite.NET-API],328
- Offsets
 - UltraLite.NET, relative,10
- Open-Methode
 - ULConnection-Klasse [UltraLite.NET-API],150
- OrderedTableScans-Eigenschaft
 - ULConnectionStringBuilder-Klasse [UltraLite.NET-API],190

P

- PageSize-Eigenschaft
 - ULCreateParms-Klasse [UltraLite.NET-API],201
- ParameterName-Eigenschaft
 - ULParameter-Klasse [UltraLite.NET-API],329
- Parameters-Eigenschaft
 - ULCommand-Klasse [UltraLite.NET-API],107
- PartialDownloadRetained-Eigenschaft
 - ULSyncResult-Klasse [UltraLite.NET-API],412
- Password-Eigenschaft
 - ULConnectionParms-Klasse [UltraLite.NET-API],177
 - ULConnectionStringBuilder-Klasse [UltraLite.NET-API],191
 - ULFileTransfer-Klasse [UltraLite.NET-API],292
 - ULSyncParms-Klasse [UltraLite.NET-API],393
- PingOnly-Eigenschaft

- ULSyncParms-Klasse [UltraLite.NET-API],394
- Plan-Eigenschaft
 - ULCommand-Klasse [UltraLite.NET-API],108
- Plattformen
 - in UltraLite.NET unterstützt,1
- Praktische Einführungen
 - Windows Mobile-Anwendung erstellen,25
- Precision-Eigenschaft
 - ULCreateParms-Klasse [UltraLite.NET-API],202
 - ULParameter-Klasse [UltraLite.NET-API],329
- Prepare-Methode
 - ULCommand-Klasse [UltraLite.NET-API],103
- PrimaryKey-Eigenschaft
 - ULTableSchema-Klasse [UltraLite.NET-API],449
- Programmcode
 - UltraLite.NET C#, praktische Einführung,38
 - UltraLite.NET Visual Basic, praktische Einführung,41
- PublicationCount-Eigenschaft
 - ULDatabaseSchema-Klasse [UltraLite.NET-API],236
- Publications-Eigenschaft
 - ULMetaDataCollectionNames-Klasse [UltraLite.NET-API],316
 - ULSyncParms-Klasse [UltraLite.NET-API],394

R

- Read-Methode
 - ULDataReader-Klasse [UltraLite.NET-API],269
- ReceivedBytes-Eigenschaft
 - ULSyncProgressData-Klasse [UltraLite.NET-API],403
- ReceivedDeletes-Eigenschaft
 - ULSyncProgressData-Klasse [UltraLite.NET-API],403
- ReceivedInserts-Eigenschaft
 - ULSyncProgressData-Klasse [UltraLite.NET-API],404
- ReceivedUpdates-Eigenschaft
 - ULSyncProgressData-Klasse [UltraLite.NET-API],404
- RecordsAffected-Eigenschaft
 - ULDataReader-Klasse [UltraLite.NET-API],271
 - ULRowUpdatedEventArgs-Klasse [UltraLite.NET-API],381
- ReferencedIndexName-Eigenschaft
 - ULIndexSchema-Klasse [UltraLite.NET-API],307

ReferencedTableName-Eigenschaft
 ULIndexSchema-Klasse [UltraLite.NET-API],308
 RegisterForEvent-Methode
 ULConnection-Klasse [UltraLite.NET-API],151
 Relativer Offset
 UltraLite.NET, Tabellen-API,10
 RemoteKey-Eigenschaft
 ULFileTransfer-Klasse [UltraLite.NET-API],293
 Remove-Methode
 ULBulkCopyColumnMappingCollection-Klasse [UltraLite.NET-API],71
 ULConnectionStringBuilder-Klasse [UltraLite.NET-API],185
 ULParameterCollection-Klasse [UltraLite.NET-API],346
 RemoveAt-Methode
 ULBulkCopyColumnMappingCollection-Klasse [UltraLite.NET-API],71
 ULParameterCollection-Klasse [UltraLite.NET-API],347
 ReservedWords-Eigenschaft
 ULMetaDataCollectionNames-Klasse [UltraLite.NET-API],316
 ReserveSize-Eigenschaft
 ULConnectionStringBuilder-Klasse [UltraLite.NET-API],191
 ResetDbType-Methode
 ULParameter-Klasse [UltraLite.NET-API],326
 ResetLastDownloadTime-Methode
 ULConnection-Klasse [UltraLite.NET-API],152
 Restrictions-Eigenschaft
 ULMetaDataCollectionNames-Klasse [UltraLite.NET-API],317
 ResumedAtSize-Eigenschaft
 ULFileTransferProgressData-Klasse [UltraLite.NET-API],299
 ResumePartialDownload-Eigenschaft
 ULFileTransfer-Klasse [UltraLite.NET-API],293
 ULSyncParms-Klasse [UltraLite.NET-API],395
 RevokeConnectFrom-Methode
 ULConnection-Klasse [UltraLite.NET-API],152
 Rollback-Methode
 UltraLite.NET-Transaktionen,16
 ULTransaction-Klasse [UltraLite.NET-API],452
 RollbackPartialDownload-Methode
 ULConnection-Klasse [UltraLite.NET-API],153
 RowCount-Eigenschaft
 ULDataReader-Klasse [UltraLite.NET-API],272
 RowsCopied-Eigenschaft
 ULRowsCopiedEventArgs-Klasse [UltraLite.NET-API],378
 RowUpdated-Ereignis
 ULDataAdapter-Klasse [UltraLite.NET-API],222
 RowUpdating-Ereignis
 ULDataAdapter-Klasse [UltraLite.NET-API],222
 RuntimeType-Eigenschaft
 ULDatabaseManager-Klasse [UltraLite.NET-API],229

S

Scale-Eigenschaft
 ULCreateParms-Klasse [UltraLite.NET-API],202
 ULParameter-Klasse [UltraLite.NET-API],330
 Schema-Eigenschaft
 ULConnection-Klasse [UltraLite.NET-API],164
 ULDataReader-Klasse [UltraLite.NET-API],272
 ULTable-Klasse [UltraLite.NET-API],436
 Schemata
 UltraLite.NET, Zugriff,17
 UltraLite.NET-API-Schemadaten,17
 ScriptCount-Eigenschaft
 ULSqlProgressData-Klasse [UltraLite.NET-API],387
 SELECT-Anweisung
 UltraLite.NET-Datenabrufbeispiel,8
 SelectCommand-Eigenschaft
 ULDataAdapter-Klasse [UltraLite.NET-API],219
 SendDownloadAck-Eigenschaft
 ULSyncParms-Klasse [UltraLite.NET-API],396
 SendNotification-Methode
 ULConnection-Klasse [UltraLite.NET-API],153
 SentBytes-Eigenschaft
 ULSyncProgressData-Klasse [UltraLite.NET-API],404
 SentDeletes-Eigenschaft
 ULSyncProgressData-Klasse [UltraLite.NET-API],405
 SentInserts-Eigenschaft
 ULSyncProgressData-Klasse [UltraLite.NET-API],405
 SentUpdates-Eigenschaft
 ULSyncProgressData-Klasse [UltraLite.NET-API],405
 ServerSyncInvoked-Methode

- ULServerSyncListener-Schnittstelle
[UltraLite.NET-API],384
- ServerVersion-Eigenschaft
 - ULConnection-Klasse [UltraLite.NET-API],164
- SetActiveSyncListener-Methode
 - ULDatabaseManager-Klasse [UltraLite.NET-API],225
- SetBoolean-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],360
- SetByte-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],361
- SetBytes-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],362
- SetDatabaseOption-Methode
 - ULDatabaseSchema-Klasse [UltraLite.NET-API],234
- SetDateTime-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],363
- SetDBNull-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],363
- SetDecimal-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],364
- SetDouble-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],365
- SetFloat-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],366
- SetGuid-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],366
- SetInt16-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],367
- SetInt32-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],368
- SetInt64-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],368
- SetServerSyncListener-Methode
 - ULDatabaseManager-Klasse [UltraLite.NET-API],226
- SetString-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],369
- SetSyncListener-Methode
 - ULConnection-Klasse [UltraLite.NET-API],154
- SetTimeSpan-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],370
- SetToDefault-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],371
- SetUInt16-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],371
- SetUInt32-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],372
- SetUInt64-Methode
 - ULResultSet-Klasse [UltraLite.NET-API],373
- SignalSyncIsComplete-Methode
 - ULDatabaseManager-Klasse [UltraLite.NET-API],228
- Size-Eigenschaft
 - ULParameter-Klasse [UltraLite.NET-API],330
- Source-Eigenschaft
 - ULException-Klasse [UltraLite.NET-API],276
 - ULInfoMessageEventArgs-Klasse [UltraLite.NET-API],310
- SourceColumn-Eigenschaft
 - ULBulkCopyColumnMapping-Klasse [UltraLite.NET-API],62
 - ULParameter-Klasse [UltraLite.NET-API],330
- SourceColumnNullMapping-Eigenschaft
 - ULParameter-Klasse [UltraLite.NET-API],331
- SourceOrdinal-Eigenschaft
 - ULBulkCopyColumnMapping-Klasse [UltraLite.NET-API],62
- SourceVersion-Eigenschaft
 - ULParameter-Klasse [UltraLite.NET-API],331
- Spalten
 - UltraLite.NET Werte ändern,15
- Sperren
 - UltraLite.NET, Schlüsselwörter,20
- SQLCODE
 - UltraLite.NET-Fehlerbehandlung,18
- StartLine-Eigenschaft
 - ULConnectionStringBuilder-Klasse [UltraLite.NET-API],192
- StartSynchronizationDelete-Methode
 - ULConnection-Klasse [UltraLite.NET-API],155
- State-Eigenschaft
 - ULConnection-Klasse [UltraLite.NET-API],165
 - ULSqlProgressData-Klasse [UltraLite.NET-API],387
 - ULSyncProgressData-Klasse [UltraLite.NET-API],406
- StateChange-Ereignis
 - ULConnection-Klasse [UltraLite.NET-API],167
- StopSynchronizationDelete-Methode
 - ULConnection-Klasse [UltraLite.NET-API],155
- Stream-Eigenschaft
 - ULFileTransfer-Klasse [UltraLite.NET-API],294
 - ULSyncParms-Klasse [UltraLite.NET-API],396
- StreamErrorCode-Eigenschaft

- ULFileTransfer-Klasse [UltraLite.NET-API],295
- ULSyncResult-Klasse [UltraLite.NET-API],413
- StreamErrorParameters-Eigenschaft
 - ULSyncResult-Klasse [UltraLite.NET-API],413
- StreamErrorSystem-Eigenschaft
 - ULFileTransfer-Klasse [UltraLite.NET-API],295
 - ULSyncResult-Klasse [UltraLite.NET-API],413
- StreamParms-Eigenschaft
 - ULFileTransfer-Klasse [UltraLite.NET-API],295
 - ULSyncParms-Klasse [UltraLite.NET-API],397
- Suchen
 - Zeilen mit UltraLite.NET,13
- Suchmethoden
 - UltraLite.NET,13
- Suchmodus
 - UltraLite.NET,11
- SYNC_ALL_DB-Feld
 - ULConnection-Klasse [UltraLite.NET-API],168
- SYNC_ALL_PUBS-Feld
 - ULConnection-Klasse [UltraLite.NET-API],169
- Synchronisation
 - ActiveSync in UltraLite.NET,19
 - UltraLite.NET,18
 - UltraLite.NET, C#-Beispiel,19
- Synchronize-Methode
 - ULConnection-Klasse [UltraLite.NET-API],156
- SyncParms-Eigenschaft
 - ULConnection-Klasse [UltraLite.NET-API],165
- SyncProgressed-Methode
 - ULSyncProgressListener-Schnittstelle [UltraLite.NET-API],409
- SyncResult-Eigenschaft
 - ULConnection-Klasse [UltraLite.NET-API],166
- SyncRoot-Eigenschaft
 - ULParameterCollection-Klasse [UltraLite.NET-API],349
- SyncTableCount-Eigenschaft
 - ULSyncProgressData-Klasse [UltraLite.NET-API],406
- SyncTableIndex-Eigenschaft
 - ULSyncProgressData-Klasse [UltraLite.NET-API],407

T

- Tabellen
 - UltraLite.NET, Schemainformationen ,17
- Tabellen-API

- UltraLite.NET Einführung,10
- TableCount-Eigenschaft
 - ULDatabaseSchema-Klasse [UltraLite.NET-API],237
- TableID-Eigenschaft
 - ULSyncProgressData-Klasse [UltraLite.NET-API],407
- TableMappings-Eigenschaft
 - ULDataAdapter-Klasse [UltraLite.NET-API],220
- TableName-Eigenschaft
 - ULSyncProgressData-Klasse [UltraLite.NET-API],407
- Tables-Eigenschaft
 - ULMetaDataCollectionNames-Klasse [UltraLite.NET-API],317
- this-Eigenschaft
 - ULBulkCopyColumnMappingCollection-Klasse [UltraLite.NET-API],71
 - ULConnectionStringBuilder-Klasse [UltraLite.NET-API],193
 - ULDataReader-Klasse [UltraLite.NET-API],273
 - ULParameterCollection-Klasse [UltraLite.NET-API],349
- Threads
 - UltraLite.NET-API-Anwendungen mit mehreren Threads,4
- TimeFormat-Eigenschaft
 - ULCreateParms-Klasse [UltraLite.NET-API],203
- Timestamp-Eigenschaft
 - ULSyncResult-Klasse [UltraLite.NET-API],414
- TimestampFormat-Eigenschaft
 - ULCreateParms-Klasse [UltraLite.NET-API],203
- TimestampIncrement-Eigenschaft
 - ULCreateParms-Klasse [UltraLite.NET-API],204
- ToString-Methode
 - ULConnectionParms-Klasse [UltraLite.NET-API],172
 - ULCreateParms-Klasse [UltraLite.NET-API],197
 - ULInfoMessageEventArgs-Klasse [UltraLite.NET-API],309
 - ULParameter-Klasse [UltraLite.NET-API],326
 - ULSyncParms-Klasse [UltraLite.NET-API],389
- TotalDownloadRowCount-Eigenschaft
 - ULSyncProgressData-Klasse [UltraLite.NET-API],408
- Transaction-Eigenschaft
 - ULCommand-Klasse [UltraLite.NET-API],108
- Transaktionen

- UltraLite.NET-Verwaltung, 16
- Transaktionsverarbeitung
 - UltraLite.NET-Verwaltung, 16
- TransferredFile-Eigenschaft
 - ULFileTransfer-Klasse [UltraLite.NET-API], 296
- TriggerEvent-Methode
 - ULConnection-Klasse [UltraLite.NET-API], 157
- Truncate-Methode
 - ULTable-Klasse [UltraLite.NET-API], 436
- TruncateDeletes-Eigenschaft
 - ULSyncProgressData-Klasse [UltraLite.NET-API], 408
- TryGetValue-Methode
 - ULConnectionStringBuilder-Klasse [UltraLite.NET-API], 186

U

- ULActiveSyncListener-Schnittstelle [UltraLite.NET-API]
 - ActiveSyncInvoked-Methode, 44
 - Beschreibung, 44
- ULAuthStatusCode-Enumeration [UltraLite.NET-API]
 - Beschreibung, 456
- ULBulkCopy-Klasse [UltraLite.NET-API]
 - BatchSize-Eigenschaft, 54
 - Beschreibung, 46
 - BulkCopyTimeout-Eigenschaft, 54
 - Close-Methode, 51
 - ColumnMappings-Eigenschaft, 55
 - DestinationTableName-Eigenschaft, 55
 - Dispose-Methode, 51
 - NotifyAfter-Eigenschaft, 56
 - ULBulkCopy-Konstruktor, 48
 - ULRowsCopied-Ereignis, 56
 - WriteToServer-Methode, 51
- ULBulkCopy-Konstruktor
 - ULBulkCopy-Klasse [UltraLite.NET-API], 48
- ULBulkCopyColumnMapping-Klasse [UltraLite.NET-API]
 - Beschreibung, 57
 - DestinationColumn-Eigenschaft, 61
 - DestinationOrdinal-Eigenschaft, 61
 - SourceColumn-Eigenschaft, 62
 - SourceOrdinal-Eigenschaft, 62
 - ULBulkCopyColumnMapping-Konstruktor, 58
- ULBulkCopyColumnMapping-Konstruktor

- ULBulkCopyColumnMapping-Klasse [UltraLite.NET-API], 58
- ULBulkCopyColumnMappingCollection-Klasse [UltraLite.NET-API]
 - Add-Methode, 65
 - Beschreibung, 63
 - Contains-Methode, 69
 - CopyTo-Methode, 70
 - IndexOf-Methode, 70
 - Remove-Methode, 71
 - RemoveAt-Methode, 71
 - this-Eigenschaft, 71
- ULBulkCopyOptions-Enumeration [UltraLite.NET-API]
 - Beschreibung, 457
- ULCommand, Klasse
 - UltraLite.NET, Datenabfragebeispiel, 8
 - UltraLite.NET, Datenbearbeitungsbeispiel, 5
- ULCommand-Klasse [UltraLite.NET-API]
 - BeginExecuteNonQuery-Methode, 79
 - BeginExecuteReader-Methode, 81
 - Beschreibung, 72
 - Cancel-Methode, 85
 - CommandText-Eigenschaft, 104
 - CommandTimeout-Eigenschaft, 104
 - CommandType-Eigenschaft, 105
 - Connection-Eigenschaft, 106
 - CreateParameter-Methode, 85
 - DesignTimeVisible-Eigenschaft, 106
 - EndExecuteNonQuery-Methode, 86
 - EndExecuteReader-Methode, 89
 - ExecuteNonQuery-Methode, 93
 - ExecuteReader-Methode, 94
 - ExecuteResultSet-Methode, 97
 - ExecuteScalar-Methode, 99
 - ExecuteTable-Methode, 100
 - IndexName-Eigenschaft, 107
 - Parameters-Eigenschaft, 107
 - Plan-Eigenschaft, 108
 - Prepare-Methode, 103
 - Transaction-Eigenschaft, 108
 - ULCommand-Konstruktor, 76
 - UpdatedRowSource-Eigenschaft, 109
- ULCommand-Konstruktor
 - ULCommand-Klasse [UltraLite.NET-API], 76
- ULCommandBuilder-Klasse [UltraLite.NET-API]
 - Beschreibung, 109
 - DataAdapter-Eigenschaft, 121

- GetDeleteCommand-Methode, 114
- GetInsertCommand-Methode, 116
- GetUpdateCommand-Methode, 118
- ULCommandBuilder-Konstruktor, 113
- ULCommandBuilder-Konstruktor
 - ULCommandBuilder-Klasse [UltraLite.NET-API], 113
- ULConnection-Klasse [UltraLite.NET-API]
 - BeginSynchronize-Methode, 127
 - BeginTransaction-Methode, 129
 - Beschreibung, 121
 - CancelGetNotification-Methode, 132
 - CancelSynchronize-Methode, 133
 - ChangeDatabase-Methode, 133
 - ChangeEncryptionKey-Methode, 134
 - ChangePassword-Methode, 134
 - Close-Methode, 135
 - ConnectionString-Eigenschaft, 160
 - ConnectionTimeout-Eigenschaft, 161
 - CountUploadRows-Methode, 135
 - CreateCommand-Methode, 136
 - CreateNotificationQueue-Methode, 137
 - Database-Eigenschaft, 161
 - DatabaseID-Eigenschaft, 162
 - DataSource-Eigenschaft, 162
 - DeclareEvent-Methode, 137
 - DestroyNotificationQueue-Methode, 138
 - EndSynchronize-Methode, 138
 - ExecuteTable-Methode, 139
 - GetLastDownloadTime-Methode, 144
 - GetNewUUID-Methode, 144
 - GetNotification-Methode, 145
 - GetNotificationParameter-Methode, 146
 - GetSchema-Methode, 147
 - GlobalAutoIncrementUsage-Eigenschaft, 163
 - GrantConnectTo-Methode, 150
 - InfoMessage-Ereignis, 166
 - INVALID_DATABASE_ID-Feld, 168
 - LastIdentity-Eigenschaft, 163
 - Open-Methode, 150
 - RegisterForEvent-Methode, 151
 - ResetLastDownloadTime-Methode, 152
 - RevokeConnectFrom-Methode, 152
 - RollbackPartialDownload-Methode, 153
 - Schema-Eigenschaft, 164
 - SendNotification-Methode, 153
 - ServerVersion-Eigenschaft, 164
 - SetSyncListener-Methode, 154
 - StartSynchronizationDelete-Methode, 155
 - State-Eigenschaft, 165
 - StateChange-Ereignis, 167
 - StopSynchronizationDelete-Methode, 155
 - SYNC_ALL_DB-Feld, 168
 - SYNC_ALL_PUBS-Feld, 169
 - Synchronize-Methode, 156
 - SyncParms-Eigenschaft, 165
 - SyncResult-Eigenschaft, 166
 - TriggerEvent-Methode, 157
 - ULConnection-Konstruktor, 125
 - ValidateDatabase-Methode, 158
- ULConnection-Konstruktor
 - ULConnection-Klasse [UltraLite.NET-API], 125
- ULConnectionParms-Klasse [UltraLite.NET-API]
 - AdditionalParms-Eigenschaft, 173
 - Beschreibung, 169
 - CacheSize-Eigenschaft, 175
 - ConnectionName-Eigenschaft, 175
 - DatabaseOnDesktop-Eigenschaft, 176
 - DatabaseOnDevice-Eigenschaft, 176
 - EncryptionKey-Eigenschaft, 177
 - Password-Eigenschaft, 177
 - ToString-Methode, 172
 - ULConnectionParms-Konstruktor, 172
 - UserID-Eigenschaft, 178
- ULConnectionParms-Konstruktor
 - ULConnectionParms-Klasse [UltraLite.NET-API], 172
- ULConnectionStringBuilder-Klasse [UltraLite.NET-API]
 - Beschreibung, 179
 - CacheSize-Eigenschaft, 186
 - ConnectionName-Eigenschaft, 187
 - ContainsKey-Methode, 184
 - DatabaseKey-Eigenschaft, 188
 - DatabaseName-Eigenschaft, 188
 - DatabaseOnDesktop-Eigenschaft, 189
 - DatabaseOnDevice-Eigenschaft, 190
 - EquivalentTo-Methode, 184
 - GetShortName-Methode, 185
 - OrderedTableScans-Eigenschaft, 190
 - Password-Eigenschaft, 191
 - Remove-Methode, 185
 - ReserveSize-Eigenschaft, 191
 - StartLine-Eigenschaft, 192
 - this-Eigenschaft, 193
 - TryGetValue-Methode, 186

- ULConnectionStringBuilder-Konstruktor, 183
- UserID-Eigenschaft, 194
- ULConnectionStringBuilder-Konstruktor
 - ULConnectionStringBuilder-Klasse [UltraLite.NET-API], 183
- ULCreateParms-Klasse [UltraLite.NET-API]
 - Beschreibung, 195
 - CaseSensitive-Eigenschaft, 198
 - ChecksumLevel-Eigenschaft, 198
 - DateFormat-Eigenschaft, 198
 - DateOrder-Eigenschaft, 199
 - FIPS-Eigenschaft, 199
 - MaxHashSize-Eigenschaft, 200
 - NearestCentury-Eigenschaft, 200
 - Obfuscate-Eigenschaft, 201
 - PageSize-Eigenschaft, 201
 - Precision-Eigenschaft, 202
 - Scale-Eigenschaft, 202
 - TimeFormat-Eigenschaft, 203
 - TimestampFormat-Eigenschaft, 203
 - TimestampIncrement-Eigenschaft, 204
 - ToString-Methode, 197
 - ULCreateParms-Konstruktor, 197
 - UTF8Encoding-Eigenschaft, 204
- ULCreateParms-Konstruktor
 - ULCreateParms-Klasse [UltraLite.NET-API], 197
- ULCursorSchema-Klasse [UltraLite.NET-API]
 - Beschreibung, 204
 - ColumnCount-Eigenschaft, 211
 - GetColumnID-Methode, 206
 - GetColumnName-Methode, 207
 - GetColumnPrecision-Methode, 207
 - GetColumnScale-Methode, 208
 - GetColumnSize-Methode, 208
 - GetColumnSQLName-Methode, 209
 - GetColumnULDbType-Methode, 210
 - GetSchemaTable-Methode, 210
 - IsOpen-Eigenschaft, 211
 - Name-Eigenschaft, 211
- ULDataAdapter-Klasse [UltraLite.NET-API]
 - Beschreibung, 212
 - DeleteCommand-Eigenschaft, 218
 - GetFillParameters-Methode, 218
 - InsertCommand-Eigenschaft, 219
 - RowUpdated-Ereignis, 222
 - RowUpdating-Ereignis, 222
 - SelectCommand-Eigenschaft, 219
 - TableMappings-Eigenschaft, 220
 - ULDataAdapter-Konstruktor, 215
 - UpdateCommand-Eigenschaft, 221
- ULDataAdapter-Konstruktor
 - ULDataAdapter-Klasse [UltraLite.NET-API], 215
- ULDatabaseManager-Klasse [UltraLite.NET-API]
 - Beschreibung, 222
 - CreateDatabase-Methode, 223
 - DropDatabase-Methode, 225
 - RuntimeType-Eigenschaft, 229
 - SetActiveSyncListener-Methode, 225
 - SetServerSyncListener-Methode, 226
 - SignalSyncIsComplete-Methode, 228
 - ValidateDatabase-Methode, 228
- ULDatabaseSchema-Klasse
 - UltraLite.NET-Entwicklung, 17
- ULDatabaseSchema-Klasse [UltraLite.NET-API]
 - Beschreibung, 229
 - GetDatabaseProperty-Methode, 230
 - GetPublicationName-Methode, 233
 - GetTableName-Methode, 233
 - IsCaseSensitive-Eigenschaft, 235
 - IsOpen-Eigenschaft, 236
 - PublicationCount-Eigenschaft, 236
 - SetDatabaseOption-Methode, 234
 - TableCount-Eigenschaft, 237
- ULDataReader, Klasse
 - UltraLite.NET, Datenabfragebeispiel, 8
- ULDataReader-Klasse [UltraLite.NET-API]
 - Beschreibung, 237
 - Close-Methode, 243
 - Depth-Eigenschaft, 269
 - FieldCount-Eigenschaft, 270
 - GetBoolean-Methode, 243
 - GetByte-Methode, 244
 - GetBytes-Methode, 244
 - GetChar-Methode, 247
 - GetChars-Methode, 247
 - GetDataTypeName-Methode, 249
 - GetDateTime-Methode, 250
 - GetDecimal-Methode, 250
 - GetDouble-Methode, 251
 - GetEnumerator-Methode, 251
 - GetFieldType-Methode, 252
 - GetFloat-Methode, 252
 - GetGuid-Methode, 253
 - GetInt16-Methode, 254
 - GetInt32-Methode, 254
 - GetInt64-Methode, 255

- GetName-Methode,256
- GetOrdinal-Methode,256
- GetRowCount-Methode,257
- GetSchemaTable-Methode,258
- GetString-Methode,260
- GetTimeSpan-Methode,261
- GetUInt16-Methode,261
- GetUInt32-Methode,262
- GetUInt64-Methode,262
- GetValue-Methode,263
- GetValues-Methode,264
- HasRows-Eigenschaft,270
- IsBOF-Eigenschaft,270
- IsClosed-Eigenschaft,271
- IsDBNull-Methode,264
- IsEOF-Eigenschaft,271
- MoveAfterLast-Methode,265
- MoveBeforeFirst-Methode,265
- MoveFirst-Methode,266
- MoveLast-Methode,266
- MoveNext-Methode,266
- MovePrevious-Methode,267
- MoveRelative-Methode,267
- NextResult-Methode,268
- Read-Methode,269
- RecordsAffected-Eigenschaft,271
- RowCount-Eigenschaft,272
- Schema-Eigenschaft,272
- this-Eigenschaft,273
- ULDateOrder-Enumeration [UltraLite.NET-API]
 - Beschreibung,458
- ULDbType-Eigenschaft
 - ULParameter-Klasse [UltraLite.NET-API],332
- ULDbType-Enumeration [UltraLite.NET-API]
 - Beschreibung,459
- ULDBValid-Enumeration [UltraLite.NET-API]
 - Beschreibung,458
- ULException-Klasse [UltraLite.NET-API]
 - Beschreibung,275
 - GetObjectData-Methode,276
 - NativeError-Eigenschaft,276
 - Source-Eigenschaft,276
- ULFactory-Klasse [UltraLite.NET-API]
 - Beschreibung,277
 - CanCreateDataSourceEnumerator-Eigenschaft,281
 - CreateCommand-Methode,278
 - CreateCommandBuilder-Methode,279
 - CreateConnection-Methode,279
 - CreateConnectionStringBuilder-Methode,280
 - CreateDataAdapter-Methode,280
 - CreateParameter-Methode,280
 - Instance-Feld,281
- ULFileTransfer-Klasse [UltraLite.NET-API]
 - AuthenticationParms-Eigenschaft,289
 - AuthStatus-Eigenschaft,290
 - AuthValue-Eigenschaft,290
 - Beschreibung,282
 - DownloadFile-Methode,284
 - FileAuthCode-Eigenschaft,291
 - FileName-Eigenschaft,291
 - LocalFileName-Eigenschaft,291
 - LocalPath-Eigenschaft,292
 - Password-Eigenschaft,292
 - RemoteKey-Eigenschaft,293
 - ResumePartialDownload-Eigenschaft,293
 - Stream-Eigenschaft,294
 - StreamErrorCode-Eigenschaft,295
 - StreamErrorSystem-Eigenschaft,295
 - StreamParms-Eigenschaft,295
 - TransferredFile-Eigenschaft,296
 - ULFileTransfer-Konstruktor,283
 - UploadFile-Methode,287
 - UserName-Eigenschaft,296
 - Version-Eigenschaft,297
- ULFileTransfer-Konstruktor
 - ULFileTransfer-Klasse [UltraLite.NET-API],283
- ULFileTransferProgressData-Klasse [UltraLite.NET-API]
 - Beschreibung,297
 - BytesReceived-Eigenschaft,298
 - FileSize-Eigenschaft,298
 - FLAG_IS_BLOCKING-Feld,299
 - Flags-Eigenschaft,299
 - ResumedAtSize-Eigenschaft,299
- ULFileTransferProgressListener-Schnittstelle [UltraLite.NET-API]
 - Beschreibung,300
 - FileTransferProgressed-Methode,300
- ULIndexSchema-Klasse
 - UltraLite.NET-Entwicklung,17
- ULIndexSchema-Klasse [UltraLite.NET-API]
 - Beschreibung,301
 - ColumnCount-Eigenschaft,303
 - GetColumnName-Methode,302
 - IsColumnDescending-Methode,303
 - IsForeignKey-Eigenschaft,304

- IsForeignKeyCheckOnCommit-Eigenschaft,304
- IsForeignKeyNullable-Eigenschaft,305
- IsOpen-Eigenschaft,305
- IsPrimaryKey-Eigenschaft,306
- IsUniqueIndex-Eigenschaft,306
- IsUniqueKey-Eigenschaft,306
- Name-Eigenschaft,307
- ReferencedIndexName-Eigenschaft,307
- ReferencedTableName-Eigenschaft,308
- ULInfoMessageEventArgs-Klasse [UltraLite.NET-API]
 - Beschreibung,308
 - Message-Eigenschaft,309
 - NativeError-Eigenschaft,310
 - Source-Eigenschaft,310
 - ToString-Methode,309
- ULInfoMessageEventHandler-Delegat [UltraLite.NET-API]
 - Beschreibung,453
- ULMetaDataCollectionNames-Klasse [UltraLite.NET-API]
 - Beschreibung,310
 - Columns-Eigenschaft,311
 - DataSourceInformation-Eigenschaft,312
 - DataTypes-Eigenschaft,313
 - ForeignKeys-Eigenschaft,313
 - IndexColumns-Eigenschaft,314
 - Indexes-Eigenschaft,314
 - MetaDataCollections-Eigenschaft,315
 - Publications-Eigenschaft,316
 - ReservedWords-Eigenschaft,316
 - Restrictions-Eigenschaft,317
 - Tables-Eigenschaft,317
- ULParameter-Klasse [UltraLite.NET-API]
 - Beschreibung,318
 - DbType-Eigenschaft,327
 - Direction-Eigenschaft,327
 - IsNullable-Eigenschaft,328
 - Offset-Eigenschaft,328
 - ParameterName-Eigenschaft,329
 - Precision-Eigenschaft,329
 - ResetDbType-Methode,326
 - Scale-Eigenschaft,330
 - Size-Eigenschaft,330
 - SourceColumn-Eigenschaft,330
 - SourceColumnNullMapping-Eigenschaft,331
 - SourceVersion-Eigenschaft,331
 - ToString-Methode,326
 - ULDbType-Eigenschaft,332
 - ULParameter-Konstruktor,320
 - Value-Eigenschaft,332
- ULParameter-Konstruktor
 - ULParameter-Klasse [UltraLite.NET-API],320
- ULParameterCollection-Klasse [UltraLite.NET-API]
 - Add-Methode,335
 - AddRange-Methode,341
 - Beschreibung,333
 - Clear-Methode,342
 - Contains-Methode,342
 - CopyTo-Methode,343
 - Count-Eigenschaft,348
 - GetEnumerator-Methode,344
 - IndexOf-Methode,344
 - Insert-Methode,345
 - IsFixedSize-Eigenschaft,348
 - IsReadOnly-Eigenschaft,348
 - IsSynchronized-Eigenschaft,349
 - Remove-Methode,346
 - RemoveAt-Methode,347
 - SyncRoot-Eigenschaft,349
 - this-Eigenschaft,349
- ULResultSet-Klasse [UltraLite.NET-API]
 - AppendBytes-Methode,357
 - AppendChars-Methode,359
 - Beschreibung,351
 - Delete-Methode,360
 - SetBoolean-Methode,360
 - SetByte-Methode,361
 - SetBytes-Methode,362
 - SetDateTime-Methode,363
 - SetDBNull-Methode,363
 - SetDecimal-Methode,364
 - SetDouble-Methode,365
 - SetFloat-Methode,366
 - SetGuid-Methode,366
 - SetInt16-Methode,367
 - SetInt32-Methode,368
 - SetInt64-Methode,368
 - SetString-Methode,369
 - SetTimeSpan-Methode,370
 - SetToDefault-Methode,371
 - SetUInt16-Methode,371
 - SetUInt32-Methode,372
 - SetUInt64-Methode,373
 - Update-Methode,374
 - UpdateBegin-Methode,374

- ULResultSetSchema-Klasse [UltraLite.NET-API]
 - Beschreibung,375
 - Name-Eigenschaft,376
- ULRowsCopied-Ereignis
 - ULBulkCopy-Klasse [UltraLite.NET-API],56
- ULRowsCopiedEventArgs-Klasse [UltraLite.NET-API]
 - Abort-Eigenschaft,377
 - Beschreibung,376
 - RowsCopied-Eigenschaft,378
 - ULRowsCopiedEventArgs-Konstruktor,377
- ULRowsCopiedEventArgs-Konstruktor
 - ULRowsCopiedEventArgs-Klasse [UltraLite.NET-API],377
- ULRowsCopiedEventHandler-Delegat [UltraLite.NET-API]
 - Beschreibung,455
- ULRowUpdatedEventArgs-Klasse [UltraLite.NET-API]
 - Beschreibung,378
 - Command-Eigenschaft,380
 - RecordsAffected-Eigenschaft,381
 - ULRowUpdatedEventArgs-Konstruktor,379
- ULRowUpdatedEventArgs-Konstruktor
 - ULRowUpdatedEventArgs-Klasse [UltraLite.NET-API],379
- ULRowUpdatedEventHandler-Delegat [UltraLite.NET-API]
 - Beschreibung,454
- ULRowUpdatingEventArgs-Klasse [UltraLite.NET-API]
 - Beschreibung,381
 - Command-Eigenschaft,383
 - ULRowUpdatingEventArgs-Konstruktor,382
- ULRowUpdatingEventArgs-Konstruktor
 - ULRowUpdatingEventArgs-Klasse [UltraLite.NET-API],382
- ULRowUpdatingEventHandler-Delegat [UltraLite.NET-API]
 - Beschreibung,454
- ULRuntimeType-Enumeration [UltraLite.NET-API]
 - Beschreibung,463
- ULServerSyncListener-Schnittstelle [UltraLite.NET-API]
 - Beschreibung,384
 - ServerSyncInvoked-Methode,384
- ULSqlProgressData-Klasse [UltraLite.NET-API]
 - Beschreibung,386
- CurrentScript-Eigenschaft,386
- ScriptCount-Eigenschaft,387
- State-Eigenschaft,387
- ULSqlProgressState-Enumeration [UltraLite.NET-API]
 - Beschreibung,464
- ULStreamType-Enumeration [UltraLite.NET-API]
 - Beschreibung,464
- ULSyncParms-Klasse [UltraLite.NET-API]
 - AdditionalParms-Eigenschaft,390
 - AuthenticationParms-Eigenschaft,390
 - Beschreibung,387
 - CopyFrom-Methode,389
 - DownloadOnly-Eigenschaft,391
 - KeepPartialDownload-Eigenschaft,392
 - NewPassword-Eigenschaft,393
 - Password-Eigenschaft,393
 - PingOnly-Eigenschaft,394
 - Publications-Eigenschaft,394
 - ResumePartialDownload-Eigenschaft,395
 - SendDownloadAck-Eigenschaft,396
 - Stream-Eigenschaft,396
 - StreamParms-Eigenschaft,397
 - ToString-Methode,389
 - UploadOnly-Eigenschaft,397
 - UserName-Eigenschaft,398
 - Version-Eigenschaft,399
- ULSyncProgressData-Klasse [UltraLite.NET-API]
 - Beschreibung,399
 - CurrentDownloadRowCount-Eigenschaft,401
 - FLAG_IS_BLOCKING-Feld,409
 - Flags-Eigenschaft,401
 - IgnoredDeletes-Eigenschaft,402
 - IsFinalSyncProgress-Eigenschaft,402
 - ReceivedBytes-Eigenschaft,403
 - ReceivedDeletes-Eigenschaft,403
 - ReceivedInserts-Eigenschaft,404
 - ReceivedUpdates-Eigenschaft,404
 - SentBytes-Eigenschaft,404
 - SentDeletes-Eigenschaft,405
 - SentInserts-Eigenschaft,405
 - SentUpdates-Eigenschaft,405
 - State-Eigenschaft,406
 - SyncTableCount-Eigenschaft,406
 - SyncTableIndex-Eigenschaft,407
 - TableID-Eigenschaft,407
 - TableName-Eigenschaft,407
 - TotalDownloadRowCount-Eigenschaft,408

- TruncateDeletes-Eigenschaft,408
- ULSyncProgressData-Klasse [UltraLite.NET-API]
 - IgnoredUpdates-Eigenschaft,402
- ULSyncProgressedDlg-Delegat [UltraLite.NET-API]
 - Beschreibung,455
- ULSyncProgressListener-Schnittstelle [UltraLite.NET-API]
 - Beschreibung,409
 - SyncProgressed-Methode,409
- ULSyncProgressState-Enumeration [UltraLite.NET-API]
 - Beschreibung,465
- ULSyncResult-Klasse [UltraLite.NET-API]
 - AuthStatus-Eigenschaft,411
 - AuthValue-Eigenschaft,411
 - Beschreibung,410
 - IgnoredRows-Eigenschaft,412
 - PartialDownloadRetained-Eigenschaft,412
 - StreamErrorCode-Eigenschaft,413
 - StreamErrorParameters-Eigenschaft,413
 - StreamErrorSystem-Eigenschaft,413
 - Timestamp-Eigenschaft,414
 - UploadOK-Eigenschaft,414
- ULTable-Klasse [UltraLite.NET-API]
 - Beschreibung,414
 - DeleteAllRows-Methode,422
 - FindBegin-Methode,422
 - FindFirst-Methode,423
 - FindLast-Methode,425
 - FindNext-Methode,427
 - FindPrevious-Methode,429
 - Insert-Methode,431
 - InsertBegin-Methode,431
 - LookupBackward-Methode,432
 - LookupBegin-Methode,434
 - LookupForward-Methode,434
 - Schema-Eigenschaft,436
 - Truncate-Methode,436
- ULTableSchema-Klasse
 - UltraLite.NET-Entwicklung,17
- ULTableSchema-Klasse [UltraLite.NET-API]
 - Beschreibung,437
 - GetColumnDefaultValue-Methode,439
 - GetColumnPartitionSize-Methode,440
 - GetIndex-Methode,441
 - GetIndexName-Methode,441
 - GetOptimalIndex-Methode,442
 - GetPublicationPredicate-Methode,442
 - IndexCount-Eigenschaft,448
 - IsColumnAutoIncrement-Methode,443
 - IsColumnCurrentDate-Methode,443
 - IsColumnCurrentTime-Methode,444
 - IsColumnCurrentTimestamp-Methode,445
 - IsColumnCurrentUTCTimestamp-Methode,445
 - IsColumnGlobalAutoIncrement-Methode,446
 - IsColumnNewUUID-Methode,446
 - IsColumnNullable-Methode,447
 - IsInPublication-Methode,447
 - IsNeverSynchronized-Eigenschaft,448
 - Name-Eigenschaft,449
 - PrimaryKey-Eigenschaft,449
 - UploadUnchangedRows-Eigenschaft,450
- UltraLite
 - Visual Studio-Integration,25
- UltraLite, Modi
 - UltraLite.NET,11
- UltraLite-Datenbanken
 - UltraLite.NET-API-Informationszugriff,17
 - verbinden in UltraLite.NET,3
- UltraLite.NET
 - ActiveSync-Synchronisation ,19
 - Daten synchronisieren,18
 - Datenabfrage,8
 - Datenänderung,5
 - Datenänderung mit SQL,5
 - Deployment,20,22
 - Info,3
 - praktische Einführung in Dynamic SQL,25
 - praktische Einführungen,25
 - Synchronisationsbeispiel ,19
 - Tabellen-API, Einführung,10
 - Transaktionsverarbeitung,16
 - unterstützte Plattformen,1
 - Zugriff auf Schemainformationen,17
- UltraLite.NET, praktische Einführung
 - Programmcode für C#,38
 - Programmcode für Visual Basic,41
- UltraLite.NET-API
 - Architektur,43
 - ULActiveSyncListener-Schnittstelle,44
 - ULAuthStatusCode-Enumeration,456
 - ULBulkCopy-Klasse,46
 - ULBulkCopyColumnMapping-Klasse,57
 - ULBulkCopyColumnMappingCollection-Klasse,63
 - ULBulkCopyOptions-Enumeration,457
 - ULCommand-Klasse,72

- ULCommandBuilder-Klasse, 109
- ULConnection-Klasse, 121
- ULConnectionParms-Klasse, 169
- ULConnectionStringBuilder-Klasse, 179
- ULCreateParms-Klasse, 195
- ULCursorSchema-Klasse, 204
- ULDataAdapter-Klasse, 212
- ULDatabaseManager-Klasse, 222
- ULDatabaseSchema-Klasse, 229
- ULDataReader-Klasse, 237
- ULDateOrder-Enumeration, 458
- ULDbType-Enumeration, 459
- ULDBValid-Enumeration, 458
- ULException-Klasse, 275
- ULFactory-Klasse, 277
- ULFileTransfer-Klasse, 282
- ULFileTransferProgressData-Klasse, 297
- ULFileTransferProgressListener-Schnittstelle, 300
- ULIndexSchema-Klasse, 301
- ULInfoMessageEventArgs-Klasse, 308
- ULInfoMessageEventHandler-Delegat, 453
- ULMetaDataCollectionNames-Klasse, 310
- ULParameter-Klasse, 318
- ULParameterCollection-Klasse, 333
- ULResultSet-Klasse, 351
- ULResultSetSchema-Klasse, 375
- ULRowsCopiedEventArgs-Klasse, 376
- ULRowsCopiedEventHandler-Delegat, 455
- ULRowUpdatedEventArgs-Klasse, 378
- ULRowUpdatedEventHandler-Delegat, 454
- ULRowUpdatingEventArgs-Klasse, 381
- ULRowUpdatingEventHandler-Delegat, 454
- ULRuntimeType-Enumeration, 463
- ULServerSyncListener-Schnittstelle, 384
- ULSqlProgressData-Klasse, 386
- ULSqlProgressState-Enumeration, 464
- ULStreamType-Enumeration, 464
- ULSyncParms-Klasse, 387
- ULSyncProgressData-Klasse, 399
- ULSyncProgressedDlg-Delegat, 455
- ULSyncProgressListener-Schnittstelle, 409
- ULSyncProgressState-Enumeration, 465
- ULSyncResult-Klasse, 410
- ULTable-Klasse, 414
- ULTableSchema-Klasse, 437
- ULTransaction-Klasse, 450
- ULTransaction-Klasse [UltraLite.NET-API]
Beschreibung, 450
- Commit-Methode, 451
- Connection-Eigenschaft, 452
- IsolationLevel-Eigenschaft, 453
- Rollback-Methode, 452
- Unterstützte Plattformen
UltraLite.NET, 1
- Update-Methode
ULResultSet-Klasse [UltraLite.NET-API], 374
- UpdateBegin-Methode
ULResultSet-Klasse [UltraLite.NET-API], 374
- UpdateCommand-Eigenschaft
ULDataAdapter-Klasse [UltraLite.NET-API], 221
- UpdatedRowSource-Eigenschaft
ULCommand-Klasse [UltraLite.NET-API], 109
- UploadFile-Methode
ULFileTransfer-Klasse [UltraLite.NET-API], 287
- UploadOK-Eigenschaft
ULSyncResult-Klasse [UltraLite.NET-API], 414
- UploadOnly-Eigenschaft
ULSyncParms-Klasse [UltraLite.NET-API], 397
- UploadUnchangedRows-Eigenschaft
ULTableSchema-Klasse [UltraLite.NET-API], 450
- UserID-Eigenschaft
ULConnectionParms-Klasse [UltraLite.NET-API], 178
- ULConnectionStringBuilder-Klasse
[UltraLite.NET-API], 194
- UserName-Eigenschaft
ULFileTransfer-Klasse [UltraLite.NET-API], 296
- ULSyncParms-Klasse [UltraLite.NET-API], 398
- UTF8Encoding-Eigenschaft
ULCreateParms-Klasse [UltraLite.NET-API], 204

V

- ValidateDatabase-Methode
ULConnection-Klasse [UltraLite.NET-API], 158
- ULDatabaseManager-Klasse [UltraLite.NET-API], 228
- Value-Eigenschaft
ULParameter-Klasse [UltraLite.NET-API], 332
- Verbinden
UltraLite-Datenbanken, 3
- Verbindung herstellen
praktische Einführung in UltraLite.NET, 25
- Version-Eigenschaft
ULFileTransfer-Klasse [UltraLite.NET-API], 297
- ULSyncParms-Klasse [UltraLite.NET-API], 399

Verwalten

UltraLite.NET, Transaktionen,16

Visual Studio

Integration mit UltraLite,25

UltraLite.NET mit UltraLite-Datenbanken

verbinden,3

W

Werte

UltraLite.NET, Zugriff in ,14

Windows Mobile

Anwendungen entwickeln,3

Datenbank erstellen,3

UltraLite.NET, Zielplattformen,1

verbinden mit einer Datenbank,3

WriteToServer-Methode

ULBulkCopy-Klasse [UltraLite.NET-API],51

Z

Zeilen

UltraLite, Aktualisieren in UltraLite.NET,12

UltraLite, Einfügen in UltraLite.NET,11

UltraLite, Löschen in UltraLite.NET,16

UltraLite.NET, Tabellennavigation,10

UltraLite.NET, Tabellenzugriff auf aktuelle,14

Zielplattformen

UltraLite.NET,1

Zugriff auf Daten

UltraLite.NET Tabellen-API,10

Zugriff auf Schemainformationen

UltraLite.NET-Info,17

Zurücksetzungen

UltraLite.NET-Transaktionen,16