SYBASE iAnywhere

Ultra Light .NET プログラミング

バージョン 12.0.1

2012 年 1 月

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

バージョン 12.0.1
2012 年 1 月

Copyright © 2012 iAnywhere Solutions, Inc. Portions copyright © 2012 Sybase, Inc. All rights reserved.

iAnywhere との間に書面による合意がないかぎり、このマニュアルは現状のまま提供されるものであり、その使用または記載内容の誤りに対して一切の責任を負いません。

次の条件に従うかぎり、このマニュアルの一部または全体を使用、印刷、複製、配布することができます。1) マニュアルの一部または全体にかかわらず、ここに示したものとそれ以外のすべての版権と商標の表示をすべてのコピーに含めること。2) マニュアルに変更を加えないこと。3) iAnywhere 以外の人間がマニュアルの著者または情報源であるかのように示す一切の行為をしないこと。

iAnywhere®、Sybase®、http://www.sybase.com/detail?id=1011207 に示す商標は Sybase, Inc. またはその関連会社の商標です。® は米国での登録商標を示します。

このマニュアルに記載されているその他の会社名と製品名は各社の商標である場合があります。

目次

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

はじめに

このマニュアルでは、Ultra Light.NET について説明します。Ultra Light.NET を使用すると、PC、ハンドヘルドデバイス、モバイルデバイス、または埋め込みデバイスのデータベースアプリケーションを開発し、これらのデバイスに配備できます。

Ultra Light .NET

Ultra Light.NET アプリケーションは、Windows Mobile および Windows に配備できます。Windows Mobile に配備する場合、Ultra Light.NET は .NET Compact Framework を必要とします。Windows に配備する場合は、.NET Framework を必要とします。Ultra Light.NET では、ActiveSync 同期もサポートしています。

.NET Compact Framework は、Windows Mobile 用の Microsoft .NET ランタイムコンポーネントです。複数のプログラミング言語をサポートします。Visual Basic.NET または C# を使用して、Ultra Light.NET を使用したアプリケーションを構築できます。

Ultra Light.NET には、次のネームスペースが用意されています。

- **iAnywhere.Data.UltraLite** このネームスペースには、Ultra Light への ADO.NET インターフェイスが用意されています。業界標準モデルに基づいて構築でき、非常によく似た SQL Anywhere ADO.NET インターフェイスへのマイグレーションパスが提供されるという利点があります。

システムの稼働条件とサポートされるプラットフォーム

開発プラットフォーム

Ultra Light.NET を使用してアプリケーションを開発するための要件は次のとおりです。

- サポートされているデスクトップバージョンの Microsoft Windows
- Microsoft Visual Studio 2005 または Visual Studio 2008
- Windows Mobile デバイスを使用する場合は、.NET Compact Framework バージョン 2 以降

ターゲットプラットフォーム

Ultra Light.NET でサポートしているターゲットプラットフォームは次のとおりです。

- Windows 上で動作する Microsoft .NET Compact Framework バージョン 2.0 以降と .NET Framework 2.0 以降
- Windows Mobile デバイスを使用する場合は、Microsoft .NET Compact Framework バージョン 2 以降

Ultra Light がサポートされているプラットフォームの詳細については、http://www.ianywhere.jp/sas/os.html を参照してください。

Ultra Light.NET API アーキテクチャー

Ultra Light.NET ネームスペースの名前は、iAnywhere.Data.UltraLite (ADO.NET インターフェイス) です。

次のリストは、iAnywhere.Data.UltraLite ADO.NET ネームスペースでよく使用される高レベルクラスの一部を説明しています。

- **ULConnection** 各 ULConnection オブジェクトは、Ultra Light データベースとの接続を表します。ULConnection オブジェクトは 1 つまたは複数作成できます。
- **ULTable** 各 ULTable オブジェクトを使用して、単一テーブル内のデータにアクセスできます。
- **ULCommand オブジェクト** 各 ULCommand オブジェクトは、データベースに対して実行される SQL 文を格納します。
- **ULDataReader オブジェクト** 各 ULDataReader オブジェクトは、単一のクエリの結果セットを格納します。
- **ULSyncParms** ULSyncParms オブジェクトを使用して、Ultra Light データベースを Mobile Link サーバーと同期させます。

参照

- 「Ultra Light.NET API リファレンス」37 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Ultra Light.NET アプリケーション開発

Visual Studio の SQL Anywhere ツール

Visual Studio 統合は Ultra Light.NET に対してサポートされています。Visual Studio 2005 以降では、Visual Studio サーバーエクスプローラーから SQL Anywhere 統合ツールにアクセスできます。

Ultra Light データベースへの接続

データベースのデータを操作するには、Ultra Light アプリケーションをデータベースに接続する必要があります。この項では、Ultra Light データベースに接続する方法について説明します。

注意
この章のサンプルコードは Microsoft C# で記述されています。サポートされている他の開発ツールを使用している場合は、このマニュアルの記述を適宜読み替えてください。

♦ Ultra Light データベースへの接続

1. ULConnection オブジェクトを宣言します。

   ほとんどのアプリケーションは、Ultra Light データベースへの単一接続を使用し、接続を開いたままにしておきます。複数の接続が必要になるのは、マルチスレッドデータアクセスの場合だけです。そのため、ULConnection オブジェクトは、アプリケーションに対してグローバルに宣言するのが最も効果的です。

   ```
   ULConnection conn;
   ```

2. 既存のデータベースへの接続を開きます。

   Ultra Light アプリケーションには、初期データベースファイルが配備されているか、データベースファイルを作成するコードが含まれている必要があります。初期データベースファイルは、Sybase Central、または Ultra Light に付属のコマンドラインユーティリティを使用して作成できます。

   接続パラメーターは、接続文字列として指定するか、ULConnectionParms オブジェクトを使用して指定します。次の例では、ULConnectionParms オブジェクトを使用して、*mydata.udb* という名前の Ultra Light データベースに接続します。

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

ULConnection オブジェクトの使用

ULConnection オブジェクトの次のプロパティは、アプリケーションのグローバルな動作を管理します。

- **コミット動作**　デフォルトでは、Ultra Light.NET アプリケーションは AutoCommit モードに設定されています。Insert 文、Update 文、Delete 文はすべて、すぐにデータベースにコミットされます。アプリケーションでのトランザクションの開始を定義するには、ULConnection.BeginTransaction を使用します。

- **ユーザー認証**　接続許可の付与と取り消しを行うメソッドを使用すると、アプリケーションのユーザー ID とパスワードをデフォルト値である DBA と sql から別の値に変更できます。各 Ultra Light データベースには、4 つまでのユーザー ID を定義できます。

- **同期**　Connection オブジェクトから、同期を管理するオブジェクトのセットにアクセスできます。

- **テーブル**　Ultra Light テーブルには、Connection オブジェクトのメソッドを使用してアクセスします。

- **コマンド**　動的 SQL 文の実行を処理し、結果セットをナビゲーションするために、オブジェクトのセットが提供されています。

マルチスレッドアプリケーション

ULConnection オブジェクトと、このオブジェクトから作成されるすべてのオブジェクトは、それぞれ単一のスレッドで使用してください。アプリケーションが Ultra Light データベースにアクセスするのに複数のスレッドを必要とする場合は、スレッドごとに個別の接続が必要です。たとえば、独立したスレッドから同期を実行するアプリケーションを設計する場合、同期用に別の接続を使用し、その接続を独立したスレッドから開く必要があります。

参照

- 「トランザクション管理」14 ページ
- 「Mobile Link データ同期」16 ページ
- 「ULTable クラスを使用したデータの作成と修正」8 ページ
- 「SQL 文を使用したデータの作成と修正」4 ページ
- ULConnection クラス [Ultra Light.NET]116 ページ

SQL 文を使用したデータの作成と修正

Ultra Light アプリケーションは、SQL 文またはテーブル API を使用してテーブルデータにアクセスできます。この項では、SQL 文を使用したデータアクセスについて説明します。

この項では、SQL を使用して次の操作を行う方法を説明します。

- ローの挿入、削除、更新

- クエリの実行と結果セットのローの取得

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

●結果セットのローのスクロール

この項では、SQL 言語そのものについては説明しません。

参照

- 「ULTable クラスを使用したデータの作成と修正」8 ページ
- 「SQL 文」『SQL Anywhere サーバー SQL リファレンス』

INSERT、UPDATE、DELETE を使用したデータ修正

Ultra Light では、SQL データ操作言語の操作を実行できます。この操作は、ULCommand.ExecuteNonQuery メソッドを使用して実行します。

♦ ローの挿入

SQL 文のパラメーターのプレースホルダーは、? 文字を使用して指定します。INSERT、UPDATE、DELETE では必ず、コマンドのパラメーターでの順序位置に従ってそれぞれの ? が参照されます。たとえば、最初の ? は 0、2 番目の ? は 1 のようになります。

1. ULCommand を宣言します。

   ```
   ULCommand cmd;
   ```

2. SQL 文を ULCommand オブジェクトに割り当てます。

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

3. 文の入力パラメーター値を割り当てます。

   次のコードは、文字列パラメーターを示します。

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

4. 文を実行します。

   戻り値は、文に影響されたローの数を示します。

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

5. 明示的なトランザクションを使用している場合は、変更をコミットします。

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

♦ ローの更新

SQL 文のパラメーターのプレースホルダーは、? 文字を使用して指定します。INSERT、UPDATE、DELETE では必ず、コマンドのパラメーターでの順序位置に従ってそれぞれの ? が参照されます。たとえば、最初の ? は 0、2 番目の ? は 1 のようになります。

1. ULCommand を宣言します。

```
ULCommand cmd;
```

2. ULCommand オブジェクトに文を割り当てます。

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

3. 文の入力パラメーター値を割り当てます。

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

4. 文を実行します。

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

5. 明示的なトランザクションを使用している場合は、変更をコミットします。

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

♦ ローの削除

SQL 文のパラメーターのプレースホルダーは、? 文字を使用して指定します。INSERT、UPDATE、DELETE では必ず、コマンドのパラメーターでの順序位置に従ってそれぞれの ? が参照されます。たとえば、最初の ? は 0、2 番目の ? は 1 のようになります。

1. ULCommand を宣言します。

```
ULCommand cmd;
```

2. ULCommand オブジェクトに文を割り当てます。

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

3. 文の入力パラメーター値を割り当てます。

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

4. 文を実行します。

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

5. 明示的なトランザクションを使用している場合は、変更をコミットします。

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

参照

- ULCommand クラス [Ultra Light.NET]66 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

SELECT を使用したデータの検索

♦ SELECT 文の実行

SELECT 文を使用すると、データベースから情報を取り出すことができます。この項では、SELECT 文の実行方法と返される結果セットの処理方法について説明します。

1. クエリを保持する ULCommand オブジェクトを宣言します。

```
ULCommand cmd;
```

2. このオブジェクトに文を割り当てます。

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

3. 文を実行します。

   クエリの結果は、数種類のオブジェクトの 1 つとして返されます。この例では、ULDataReader オブジェクトが使用されています。次のコードでは、SELECT 文の結果に文字列が含まれています。これはコマンドプロンプトへ出力されます。

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

結果セットスキーマの説明

ULDataReader.GetSchemaTable メソッドと ULDataReader.Schema プロパティを使用すると、カラム名、カラムの総数、カラムスケール、カラムサイズ、カラムの SQL 型など、結果セットに関する情報を取得できます。

例

次のサンプルコードは、ULDataReader.Schema プロパティと ResultSet.Schema プロパティを使用して、スキーマ情報をコマンドプロンプトに表示する方法を示しています。

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

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

SQL 結果セットのナビゲーション

ULDataReader オブジェクトに関連したメソッドを使用して、結果セット内をナビゲーションできます。

結果セットオブジェクトは、結果セットをナビゲーションする次のメソッドを提供します。

- **MoveAfterLast**　最後のローの後に移動します。
- **MoveBeforeFirst**　最初のローの前に移動します。
- **MoveFirst**　最初のローに移動します。
- **MoveLast**　最後のローに移動します。
- **MoveNext**　次のローに移動します。
- **MovePrevious**　前のローに移動します。
- **MoveRelative(offset)**　現在のローを基準にして、オフセットが指定した数だけローを移動します。オフセット値を正の値で指定すると、現在の結果セットのカーソル位置から前方に移動します。負の値で指定すると後方に移動します。オフセット値が 0 の場合、カーソルは移動しませんが、ローバッファーが再配置されます。

ULTable クラスを使用したデータの作成と修正

Ultra Light アプリケーションは、SQL 文または ULTable クラスを使用してテーブルデータにアクセスできます。この項では、ULTable クラスを使用したデータアクセスについて説明します。

この項では、テーブル API を使用して次の操作を行う方法について説明します。

●テーブルのローのスクロール

●現在のローの値へのアクセス

●find メソッドと lookup メソッドを使用したテーブルのローの検索

●ローの挿入、削除、更新

参照

●「SQL 文を使用したデータの作成と修正」4 ページ

ローナビゲーション

Ultra Light.NET は、幅広いナビゲーション作業を行うために、テーブルをナビゲーションする複数のメソッドを提供します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

テーブルオブジェクトは、テーブルをナビゲーションする次のメソッドを提供します。

- **MoveAfterLast**　最後のローの後に移動します。
- **MoveBeforeFirst**　最初のローの前に移動します。
- **MoveFirst**　最初のローに移動します。
- **MoveLast**　最後のローに移動します。
- **MoveNext**　次のローに移動します。
- **MovePrevious**　前のローに移動します。
- **MoveRelative(offset)**　現在のローを基準にして、オフセットが指定した数だけローを移動します。オフセット値を正の値で指定すると、現在のテーブルのカーソル位置から前方に移動します。負の値で指定すると後方に移動します。オフセット値が 0 の場合、カーソルは移動しませんが、ローバッファーが再配置されます。

参照

- ULTable クラス [Ultra Light.NET]415 ページ
- ULTableSchema クラス [Ultra Light.NET]437 ページ

例

次のサンプルコードは、MyTable テーブルを開き、各ローの MyColumn カラムの値を表示します。

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

テーブルオブジェクトを開くと、テーブルのローがアプリケーションに公開されます。デフォルトでは、ローはプライマリキー値の順に並んでいますが、テーブルを開くときにインデックスを指定すると特定の順序でローにアクセスできます。

例

次のコードは、ix_col インデックスで順序付けられた MyTable テーブルの最初のローに移動します。

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

Ultra Light のモード

Ultra Light モードは、バッファー内の値の使用目的を指定します。Ultra Light には、デフォルトモードに加えて、次の 4 つの操作モードがあります。

- **挿入モード**　Insert メソッドを呼び出すと、バッファー内のデータが新しいローとしてテーブルに追加されます。
- **更新モード**　Update メソッドを呼び出すと、現在のローがバッファー内のデータに置き換えられます。
- **検索モード**　find メソッドの 1 つが呼び出されたときに、値がバッファー内のデータに正確に一致するローの検索に使用されます。
- **ルックアップモード**　いずれかの lookup メソッドが呼び出されたときに、バッファー内のデータと一致するか、それより大きい値のローを検索します。

ローの挿入

ローの挿入手順は、ローの更新手順とほぼ同じです。ただし、挿入操作の場合は、テーブル内のローをあらかじめ指定する必要はありません。テーブルへのローの挿入順序に意味はありません。

例

次のコードでは、新しいローが挿入されます。

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

カラムの値を設定しない場合、そのカラムにデフォルト値があるときはデフォルト値が使用されます。カラムにデフォルトがない場合は、次のエントリが使用されます。

●NULL 入力可のカラムの場合は NULL

●NULL 入力不可の数値カラムの場合は 0

●NULL 入力不可の文字カラムの場合は空の文字列

●明示的に値を NULL に設定するには、SetDBNull メソッドを使用します。

更新操作の場合は、コミットを実行したときに、永続的な記憶領域のデータベースに挿入が適用されます。AutoCommit モードでは、Insert メソッドの一部として Commit が実行されます。

ローの更新

ローの更新手順を次に示します。

警告
ローのプライマリキー値は更新できません。代わりに、ローを削除して新しいローを追加してください。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

♦ **ローの更新**

1. 更新するローに移動します。

   ローに移動するには、テーブルをスクロールするか、find メソッドまたは lookup メソッドを使用してテーブルを検索します。

2. 更新モードを開始します。

   たとえば、次の指示は、テーブル t 上で更新モードを開始します。

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

3. 更新するローの新しい値を設定します。

   たとえば、次の指示は、バッファー内の id カラムを 3 に設定します。

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

4. Update を実行します。

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

更新操作が終了すると、更新したローが現在のローになります。Table オブジェクトを開いたときに指定したインデックスのカラム値を変更した場合は、現在のローの定義が解除されます。

デフォルトでは、Ultra Light.NET は AutoCommit モードで動作するため、更新は永続的な記憶領域のローに即時適用されます。AutoCommit モードを無効にした場合は、コミット操作を実行するまで、更新は適用されません。

参照

● 「トランザクション管理」14 ページ

find と lookup を使用したローの検索

Ultra Light には、データを操作するための操作モードがいくつかあります。これらのモードのうちの 2 つ (検索モードとルックアップモード) は、検索に使用されます。Table オブジェクトには、テーブル内の特定のローを検索するために、これらのモードに対応するメソッドがあります。

> **注意**
> find メソッドや lookup メソッドを使用して検索されるカラムは、テーブルを開くのに使用されたインデックスにあることが必要です。

● **find メソッド**　Table オブジェクトを開いたときに指定したソート順に基づいて、指定された検索値と正確に一致する最初のローに移動します。検索値が見つからない場合は、最初のローの前、または最後のローの後ろに位置設定されます。

- **lookup メソッド**　Table オブジェクトを開いたときに指定したソート順に基づいて、指定された検索値と一致するか、それより大きい値の最初のローに移動します。

♦ ローの検索

1. 検索モードまたはルックアップモードを開始します。

   モードを開始するには、テーブルオブジェクトのメソッドを呼び出します。たとえば、次のコードは検索モードを開始します。

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

2. 検索値を設定します。

   検索値は、現在のローの値を設定することで設定します。これらの値の設定は、データベースではなく、現在のローを保持しているバッファーにのみ影響します。たとえば、次のコードは、バッファーの値を Kaminski に設定します。

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

3. ローを検索します。

   適切なメソッドを使用して検索を実行します。たとえば、次の指示は、現在のインデックスで指定された値と正確に一致する最初のローを検索します。

   マルチカラムインデックスの場合、最初のカラムの値が常に使用され、他のカラムは省略できます。

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

4. ローの次のインスタンスを検索します。

   適切なメソッドを使用して検索を実行します。検索操作の場合は、FindNext でインデックス内のパラメーターの次のインスタンスを検索します。ルックアップ操作では、MoveNext で次のインスタンスを検索します。

参照

- ULTable クラス [Ultra Light.NET]415 ページ

現在のローの値へのアクセス

Table オブジェクトは、次のいずれかの位置に常に置かれています。

●テーブルの最初のローの前

●テーブルのいずれかのローの上

●テーブルの最後のローの後ろ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Table オブジェクトがローの上に置かれている場合は、そのデータ型に適したメソッドセットを使用して、各カラムの値を取得したり、変更したりできます。

カラム値の取得

Table オブジェクトは、カラム値を取得するメソッドセットを提供します。これらのメソッドは、カラム ID を引数として取ります。

例

次のコードは、lname カラムの値を取得します。このカラムの値は文字列です。

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

次のコードは、cust_id カラムの値を取得します。このカラムの値は整数です。

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

カラム値の変更

値を取り出すメソッド以外に、値を設定するメソッドもあります。値を設定するメソッドは、カラム ID と値を引数として取ります。

例

たとえば、次のコードは、lname カラムの値を Kaminski に設定します。

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

これらのプロパティへの値の割り当てによって、データベース内のデータの値が変更されることはありません。現在の位置がテーブルの最初のローの前でも、テーブルの最後のローの後ろであっても、プロパティに値を割り当てることができます。ただし、これらの位置に現在のローが置かれている場合に、たとえば、変数へのプロパティの割り当てなどによって、データにアクセスしようとするとエラーになります。

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

値のキャスト

選択するメソッドは、割り当てるデータ型に一致させてください。データ型に互換性がある場合は、Ultra Light が自動的にデータベースのデータ型をキャストするため、getString メソッドを使用して整数値を文字列変数にフェッチしたりできます。

参照

- 「データ型の明示的な変換」『Ultra Light データベース管理とリファレンス』

ローの削除

ローの削除手順は、ローの挿入や更新よりも簡単です。挿入モードや更新モードに対応する削除モードはありません。

次の手順は、ローを削除します。

♦ ローの削除

1. 削除するローに移動します。
2. Table.Delete メソッドを実行します。

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

トランザクション管理

Ultra Light のトランザクション処理は、データベース内のデータの整合性を保証します。トランザクションは、作業の論理単位です。トランザクション全体が実行されるか、トランザクション内の文がどれも実行されないかのいずれかです。

デフォルトでは、Ultra Light.NET は AutoCommit モードで動作するため、挿入、更新、削除はそれぞれ独立したトランザクションとして実行されます。操作が完了すると、データベースに変更が加えられます。

複数文のトランザクションを使用するには、ULConnection.BeginTransaction を呼び出して ULTransaction クラスのオブジェクトを作成する必要があります。たとえば、2 つの口座間で資金を移動するアプリケーションでは、振込元口座からの引き落としと振込先口座への振り込みとを合わせて 1 つの操作として完了する必要があります。完了できない場合は、両方とも完了しないでください。

接続が有効なトランザクションを処理した場合、ULTransaction.Commit 文を実行して、トランザクションを完了し、データベースへの変更をコミットする必要があります。一連の更新を中止する場合は、ULTransaction.Rollback 文を実行して、トランザクションのすべての操作を取り消してロールバックする必要があります。トランザクションがコミットまたはロールバックされると、次に ULConnection.BeginTransaction を呼び出すまで、接続は AutoCommit モードに戻ります。

たとえば、次のコードフラグメントは、(デフォルトのオートコミット動作を回避しながら) 複数の操作を伴うトランザクションを設定する方法を示しています。

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

注意
Ultra Light は、IsolationLevel 列挙体の IsolationLevel.ReadUncommitted メンバーのみをサポートします。

一部の SQL 文 (特にデータベースの構造を変更する文) は、保留中のトランザクションをすべてコミットします。処理中のトランザクションを自動的にコミットする SQL 文には、CREATE TABLE や ALTER TABLE などがあります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULConnection クラス [Ultra Light.NET]116 ページ
- ULTransaction クラス [Ultra Light.NET]451 ページ

スキーマ情報へのアクセス

テーブル API のオブジェクトは、テーブル、カラム、インデックス、同期の各パブリケーションを表します。各オブジェクトには、そのオブジェクトの構造情報へアクセスするための Schema プロパティがあります。

API によるスキーマの変更はできません。スキーマに関する情報の取得のみが可能です。

次のスキーマオブジェクトと情報にアクセスできます。

- **ULDatabaseSchema** データベース内のテーブルの数と名前、日付と時刻のフォーマットなどのグローバルプロパティを公開します。

  ULDatabaseSchema オブジェクトを取得するには、ULConnection.Schema を呼び出します。

- **ULTableSchema** このテーブル内のカラムとインデックスの数と名前。

  ULTableSchema オブジェクトを取得するには、ULTable.Schema を呼び出します。

- **ULIndexSchema** インデックス内のカラムに関する情報。インデックスには直接に対応するデータがないため、個別の Index クラスはなく、ULIndexSchema クラスのみが存在します。

  ULIndexSchema オブジェクトを取得するには、ULTableSchema.GetIndex、ULTableSchema.GetOptimalIndex、ULTableSchema.GetPrimaryKey のいずれかのメソッドを呼び出します。

参照

- ULTable.Schema プロパティ [Ultra Light.NET]436 ページ
- ULDatabaseSchema クラス [Ultra Light.NET]227 ページ
- ULTableSchema クラス [Ultra Light.NET]437 ページ
- ULIndexSchema クラス [Ultra Light.NET]301 ページ

エラー処理

.NET の標準エラー処理機能を使用して、エラーを処理できます。ほとんどの Ultra Light メソッドは、ULException エラーをスローします。ULException.NativeError を使用して、エラーに割り当てられた ULSQLCode 値を取得できます。ULException には Message プロパティがあり、エラーの説明文の取得に使用できます。ULSQLCode エラーは、エラータイプを示す負の番号です。

同期後は、接続の SyncResult プロパティを使用して詳細なエラー情報を取得できます。たとえば、次の例は、同期時に発生したエラーをレポートする方法を示しています。

```
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);
    }
}
```

参照

- エラーメッセージ
- ULSyncProgressListener インターフェイス [Ultra Light.NET]409 ページ
- ULSyncResult クラス [Ultra Light.NET]410 ページ

Mobile Link データ同期

Ultra Light データベースを中央の統合データベースと同期します。同期には、SQL Anywhere に付属の Mobile Link 同期ソフトウェアが必要です。

この項では、同期の概要について簡単に説明し、Ultra Light.Net コンポーネントを使用するうえで重要となるいくつかの機能について説明します。

また、CustDB サンプルアプリケーションには、同期の実例もあります。詳細については、SQL Anywhere 12 インストール環境の *Samples¥UltraLite.NET¥CustDB* サブフォルダーを参照してください。

Ultra Light.NET は、TCP/IP、HTTP、HTTPS、TLS (トランスポートレイヤーセキュリティ) の各同期をサポートしています。同期は、Ultra Light アプリケーションによって開始されます。常に、SyncParms オブジェクトのプロパティを使用して同期を制御します。

注意
別途ライセンスが必要な必須コンポーネント

ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。

「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- 「Ultra Light クライアント」『Ultra Light データベース管理とリファレンス』

C# アプリケーションでの同期の開始

次のコードは、C# で記述されているアプリケーションで同期を開始する方法を示します。

```
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);
        }
}
```

アプリケーションへの ActiveSync 同期の追加

この項では、ActiveSync を Ultra Light.NET アプリケーションに追加する方法について説明します。また、エンドユーザーのコンピューター上の ActiveSync で使用するアプリケーションの登録方法についても説明します。

ActiveSync 同期を開始できるのは、ActiveSync 自体だけです。デバイスがクレードルにある場合や、[ActiveSync] ウィンドウで **[同期]** が選択された場合に、ActiveSync は同期を開始します。

ActiveSync が同期を開始すると、Ultra Light アプリケーションがまだ実行されていない場合は ActiveSync 用 Mobile Link プロバイダーが Ultra Light アプリケーションを起動し、メッセージを送信します。Mobile Link プロバイダーからのメッセージを受信および処理するには、ULActiveSyncListener オブジェクトをアプリケーションに実装します。アプリケーションは、SetActiveSyncListener メソッドを使用してリスナーオブジェクトを指定する必要があります。ここで、**MyAppClassName** は、アプリケーションのユニークな Windows クラス名です。

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

Ultra Light は、ActiveSync メッセージを受信すると、指定されたリスナーの ActiveSyncInvoked メソッドを別のスレッド上で呼び出します。マルチスレッドの問題を回避するには、ActiveSyncInvoked メソッドでユーザーインターフェイスにイベントを通知してください。

マルチスレッドアプリケーションを使用する場合は、スレッドごとに別々の接続を使用し、C# の **lock** キーワードまたは Visual Basic .NET の **SyncLock** キーワードを使用して、アプリケーショ

ンの共有オブジェクトにアクセスします。ActiveSyncInvoked メソッドは、使用する接続の SyncParms.Stream に ULStreamType.ACTIVE_SYNC を指定してから ULConnection.Synchronize を呼び出します。

アプリケーションを登録するときは、次のパラメーターを設定します。

- **クラス名**　Connection.SetActiveSyncListener メソッドでアプリケーションが使用したクラス名と同じクラス名

参照

- ULActiveSyncListener インターフェイス [Ultra Light.NET]37 ページ

Ultra Light.NET アプリケーションの配備

次のファイルを、Windows Mobile デバイスまたは Windows デスクトップに配備する必要があります。すべてのファイルパスは、インストールした SQL Anywhere の UltraLite ディレクトリに対する相対パスです。

- ***iAnywhere.Data.UltraLite.dll***　iAnywhere.Data.UltraLite ADO.NET ネームスペースを含むアセンブリ。このファイルは、*UltraLite.NET¥Assembly¥V2* ディレクトリと *UltraLite.NET¥CE¥Assembly¥V2* ディレクトリにあります。

- ***iAnywhere.Data.UltraLite.resources.dll***　iAnywhere.Data.UltraLite ADO.NET ネームスペースで必要とされるリソース。このファイルは、*UltraLite.NET¥Assembly¥V2ja* ディレクトリと *UltraLite.NET¥CE¥Assembly¥V2ja* ディレクトリにあります。

- ***ulnet12.dll***　このファイルには、1 つのクライアントが使用する Ultra Light ランタイムが含まれてます。このランタイムの異なるバージョンが、ターゲットプラットフォームごとに用意されています。このファイルは、*UltraLite.NET¥win32* ディレクトリ、*UltraLite.NET¥win32* ディレクトリ、および *UltraLite.NET¥CE¥Arm.50* ディレクトリにあります。

- ***ulnetclient12.dll***　このファイルには、Ultra Light エンジンへのインターフェイスが含まれており、複数のクライアントがエンジンにアクセスするのに使用されます。このファイルは、*UltraLite.NET¥win32* ディレクトリ、*UltraLite.NET¥win32* ディレクトリ、および *UltraLite.NET¥CE¥Arm.50* ディレクトリにあります。

参照

- 「Ultra Light アプリケーションのビルドと配備の仕様」『Ultra Light データベース管理とリファレンス』
- 「Ultra Light の ActiveSync プロバイダーの配備」『Ultra Light データベース管理とリファレンス』

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

チュートリアル：Ultra Light.NET を使用した Windows Mobile アプリケーションの構築

このチュートリアルでは、Microsoft Visual Studio を使用して Windows Mobile 用の Ultra Light アプリケーションをビルドする手順について説明します。また、このチュートリアルは iAnywhere.Data.UltraLite ネームスペースに用意されている ADO.NET インターフェイスを使用し、.NET 3.5 Compact Framework 上で動作します。

このチュートリアルには、Visual Basic アプリケーションと Visual C# アプリケーションのコードが含まれています。

前提知識と経験

このチュートリアルは、次のことを前提にしています。

- C# プログラミング言語または Visual Basic プログラミング言語に精通している。
- Sybase Central の Ultra Light プラグインを使用して Ultra Light データベースを作成する方法を知っている。
- Microsoft Visual Studio がコンピューターにインストール済みである。また、Visual Studio に精通している。このチュートリアルは Visual Studio 2008 を使用してテストされており、Visual Studio のその他のバージョンとは多少異なる Visual Studio アクションやプロシージャーに基づいている場合があります。
- Microsoft から Windows Mobile 5.0 SDK 以降をインストール済みである。
- 使用中のモバイルデバイスに .NET 3.5 Compact Framework をインストール済みである。

目的

このチュートリアルの目的は、Visual Studio 環境での Ultra Light アプリケーションの開発プロセスについて、知識と経験を得ることです。

インストールの注意

Ultra Light ソフトウェアをインストールした Windows コンピューターに Visual Studio がインストール済みの場合、Ultra Light のインストールプロセスは Visual Studio の存在を検出し、必要な統合手順を実行します。Visual Studio を Ultra Light の後にインストールする場合、または Visual Studio の新しいバージョンをインストールする場合、次の手順に従って、コマンドプロンプトから手動で Ultra Light を Visual Studio に統合する必要があります。

- Visual Studio が実行されていないことを確認します。
- Visual Studio 2005 以降の場合は、*%SQLANY12%¥UltraLite¥UltraLite.NET¥Assembly¥v2¥* フォルダーから *installULNet.exe* を実行します。この作業には管理者権限が必要です。

参照

- 「データベース作成ウィザードでの Ultra Light データベースの作成」『Ultra Light データベース管理とリファレンス』

レッスン 1：Visual Studio プロジェクトの作成

次の手順では、新しい Visual Studio アプリケーションを作成および設定します。プログラミング言語として Visual Basic と C# のどちらも使用できます。

このチュートリアルでは、C# アプリケーションを設計している場合はファイルがディレクトリ *C:¥tutorial¥uldotnet¥CSApp* にあり、Visual Basic アプリケーションを設計している場合はファイルがディレクトリ *C:¥tutorial¥uldotnet¥VBApp* にあることを前提とします。別の名前のディレクトリを使用する場合は、チュートリアルを通じてそのディレクトリを使用してください。

♦ Visual Studio プロジェクトの作成

1. Visual Studio プロジェクトを作成します。

   ●Visual Studio で、**[ファイル]** » **[新規作成]** » **[プロジェクト]** をクリックします。

   ●**[新しいプロジェクト]** ウィンドウが表示されます。左ウィンドウ枠で、**[Visual Basic]** フォルダーまたは **[Visual C#]** フォルダーを選択します。プロジェクトタイプとして **[スマートデバイス]** をクリックします。

   右ウィンドウ枠で、**[スマート デバイス プロジェクト]** をクリックし、Visual Basic と C# のどちらをプログラミング言語として使用するかに応じて、プロジェクトに **VBApp** または **CSApp** という名前を付けます。

   ●*C:¥tutorial¥uldotnet* の場所を入力し、**[OK]** をクリックします。

   ●ターゲットプラットフォームとして **[Windows Mobile 5.0 Pocket PC SDK]**、ターゲット .NET Compact Framework バージョンとして **[.NET Compact Framework バージョン 3.5]** をクリックします。**[OK]** をクリックします。

2. プロジェクトに参照を追加します。

   ●iAnywhere.Data.UltraLite アセンブリと、関連するリソースをプロジェクトに追加します。

   a. **[プロジェクト]** » **[参照の追加]** をクリックします。

   b. 利用可能な参照のリストで **[iAnywhere.Data.UltraLite]** と **[iAnywhere.Data.UltraLite EN]** (英語の場合) をクリックします。**[OK]** をクリックして、選択済みコンポーネントのリストに追加します。

   使用する言語が英語でない場合は、**[参照]** をクリックして、SQL Anywhere インストール環境の *UltraLite¥UltraLite.NET¥ce¥Assembly¥v2¥xx* サブフォルダーで **iAnywhere.Data.UltraLite xx** を見つけます。この場合、xx は、言語を表す 2 文字の略語 (英語の場合は **en**、など) です。*iAnywhere.Data.UltraLite.resources.dll* をクリックして **[開く]** をクリックします。

   ●Ultra Light コンポーネントをプロジェクトにリンクします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

ここで、リンクをコンポーネントに追加したことと、コンポーネントを開いていないことを確認します。

a. **[プロジェクト]** » **[既存項目の追加]** をクリックし、SQL Anywhere インストール環境の *UltraLite¥UltraLite.NET¥ce* サブフォルダーを参照します。

b. **[Objects of Type]** リストで **[実行ファイル]** をクリックします。

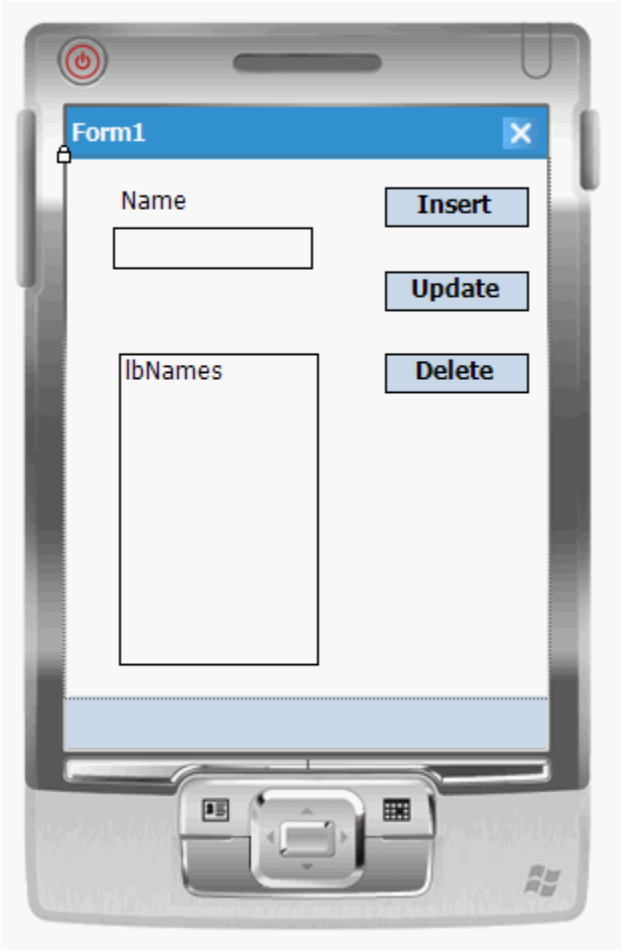
c. 使用している Windows Mobile デバイスのプロセッサーに対応するフォルダーを開きます。Visual Studio 2005 以降の場合は、*Arm.50* フォルダーを開きます。*ulnet12.dll* をクリックし、**[追加]** ボタンの矢印をクリックして、**[リンクとして追加]** をクリックします。

3. アプリケーションのフォームを作成します。

Visual Studio のツールボックスパネルが表示されていない場合は、メインメニューで **[表示]** » **[ツールボックス]** をクリックします。オブジェクトを選択して目的のロケーションのフォームにドラッグすることで、次のビジュアルコンポーネントをフォームに追加します。

型	設計 - 名前	外観 - テキスト
Button	btnInsert	Insert
Button	btnUpdate	Update
Button	btnDelete	Delete
TextBox	txtName	(テキストなし)
ListBox	lbNames	(テキストなし)
Label	laName	Name

フォームは、次の図のようになります。



4. ソリューションをビルドおよび配置します。

   ソリューションをビルドおよび配置し、Visual Studio プロジェクトが正しく設定されたことを確認します。

   a. **[ビルド]** メニューで **[ソリューションのビルド]** をクリックします。プロジェクトが正常にビルドされたことを確認します。Visual Basic アプリケーションをビルドしている場合、次の警告は無視してかまいません。

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

   b. **[デバッグ]** » **[デバッグの開始]** をクリックします。

      これによって、アプリケーションがモバイルデバイスまたはエミュレータに配置され、起動されます。アプリケーションは、エミュレーターまたはプロジェクト名に応じたデバイスのロケーション (*¥Program Files¥VBApp* または *¥Program Files¥CSApp*) に配置されます。

      配置には時間がかかる場合があります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

c. アプリケーションがエミュレーターまたはターゲットデバイスに配置され、設計したフォーム (**Form1**) が正しく表示されることを確認します。

d. エミュレーターまたはターゲットデバイスのアプリケーションを終了します。

レッスン 2：Ultra Light データベースの作成

(デスクトップ PC で実行する) 次の手順では、Sybase Central を使用して Ultra Light データベースを作成します。

♦ データベースの作成

1. **[スタート] » [プログラム] » [SQL Anywhere 12] » [管理ツール] » [Sybase Central]** をクリックします。

2. Sybase Central の Ultra Light プラグインを使用して、アプリケーションと同じディレクトリにデータベースを作成します。

   [ツール] » [Ultra Light 12] » [データベースの作成] をクリックします。

   通常、Sybase Central によって設定されるデフォルトのデータベース特性のままで問題ありません。特性は次のとおりです。

   - **データベースファイル名**　*c:¥tutorial¥uldotnet¥VBApp¥VBApp.udb* または *c:¥tutorial¥uldotnet¥CSApp¥CSApp.udb* (アプリケーションタイプによって異なります)。
   - **照合順**　デフォルトの照合を使用します。
   - **文字列の比較で大文字と小文字を区別する**　このオプションはオフにしてください。

   [完了] をクリックして、Ultra Light データベースに接続します。

3. Sybase Central ツリービューの **[テーブル]** フォルダーアイコンを強調表示してから、**[ファイル] » [新規] » [テーブル]** をクリックして、新しい Ultra Light テーブルを作成します。特性は次のとおりです。

   - **テーブル名**　**Names** と入力します。
   - **カラム**　次の属性を持つカラムを **Names** テーブルに作成します。

カラム名	データ型 (サイズ)	NULL	ユニーク	デフォルト値
ID	integer	いいえ	はい (プライマリキー)	グローバルオートインクリメント
Name	varchar(30)	いいえ	いいえ	なし

   - **プライマリキー**　ID カラムをプライマリキーとして指定します。

4. Sybase Central を終了し、データベースファイルが指定のディレクトリに作成されていることを確認します。

5. 初期化済みの (空の) データベースファイルを Visual Studio プロジェクトにリンクし、データベースファイルがアプリケーションコードとともにデバイスに配備されるようにします。

   ●**[Visual Studio]** » **[プロジェクト]** » **[既存項目の追加]** をクリックします。

   ●**[Objects of Type]** が **[すべてのファイル]** に設定されていることを確認します。データベースファイルを作成したディレクトリを参照し、アプリケーションタイプに応じてファイル *VBApp.udb* または *CSApp.udb* をクリックします。

   ●**[追加]** ボタンの矢印をクリックし、**[リンクとして追加]** をクリックします。

   ●Solution Explorer フレームで、プロジェクトに追加したデータベースファイル名を右クリックし、**[プロパティ]** をクリックします。

   プロパティのパネルで、**[ビルド アクション]** プロパティを **[コンテンツ]** に設定し、**[出力ディレクトリにコピー]** プロパティを **[常にコピーする]** に設定します。

参照

- 「データベース作成ウィザードでの Ultra Light データベースの作成」『Ultra Light データベース管理とリファレンス』

レッスン 3：データベースへの接続

次の手順では、Ultra Light データベースへの接続を確立する Ultra Light.NET アプリケーションにコントロールを追加します。

♦ アプリケーションへの Ultra Light 接続の追加

1. フォームをダブルクリックして、ソースファイル (*Form1.cs* または *Form1.vb*) を開きます。

2. コードを追加して、iAnywhere.Data.UltraLite ネームスペースをインポートします。

   ファイルの先頭行に次の文を追加します。

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

   'Visual Basic
   Imports iAnywhere.Data.UltraLite
   ```

3. グローバル変数をフォーム宣言に追加します。

   Visual C# の場合は、フォームコンポーネントを記述したコードと、最初のメソッド宣言までの間に、次のコードを追加します。

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

   Visual Basic の場合は、Form1 クラスの先頭に次のコードを追加します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

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

これらの変数は次のように使用します。

- **ULConnection**　Connection オブジェクトは、データベースへの接続時に実行されるすべてのアクションのルートオブジェクトです。
- **ids**　ids 配列を使用して、クエリの実行後に返される ID カラム値を格納します。

  ListBox コントロール自体によって順序番号にアクセスできますが、ローが削除されると、これらの番号は ID カラムの値とは異なります。このため、ID カラム値を個別に格納してください。

4. フォームの空白領域をダブルクリックして、Form1_Load メソッドを作成します。

   このメソッドでは次のタスクを実行します。

   - ulConnectionParms1 コントロールで設定されている接続パラメーターを使用して、データベースへの接続を開きます。
   - RefreshListBox メソッドを呼び出します (このチュートリアルの後半で定義)。
   - エラーが発生したら、エラーメッセージを印刷 (表示) します。SQL Anywhere エラーの場合は、エラーコードも出力されます。エラーメッセージを参照してください。

   C# の場合は、次のコードを Form1_Load メソッドに追加します。

```
//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);
}
```

   Visual Basic の場合は、次のコードを Form1_Load メソッドに追加します。

```
'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. プロジェクトをビルドします。

   [ビルド] メニューで **[ソリューションのビルド]** をクリックします。この段階で、1 つのエラーがレポートされます。たとえば C# の場合は、RefreshListBox が宣言されていないことを

示す「error CS0103: The name 'RefreshListBox' does not exist in the current context.」というエラーがレポートされます。次のレッスンではこの関数を追加します。

他のエラーが表示された場合は、先へ進む前にエラーを修正します。C# における大文字と小文字の違いなどの一般的なエラーをチェックします。たとえば、**UltraLite** および **ULConnection** は大文字と小文字が正確に一致している必要があります。Visual Basic では、レッスン 3 で説明した **Imports iAnywhere.Data.UltraLite** 文を含めることが重要です。

レッスン 4：データの挿入、更新、削除

このレッスンでは、コードをアプリケーションに追加し、データベースのデータを変更します。次の手順では、動的 SQL を使用します。テーブル API を使用して同じテクニックを実行することもできます。

次の手順では、リストボックスを管理するサポートメソッドを作成します。このメソッドは、残りの手順で作成されるデータ操作メソッドに必要です。

♦ リストボックスを管理するコードの追加

1. フォームを右クリックし、**[コードの表示]** をクリックします。
2. Form1 クラスのメソッドを追加し、リストボックスを更新してデータを移植します。このメソッドでは、次のタスクを実行します。

   ●リストボックスをクリアします。

   ●ULCommand オブジェクトをインスタンス化し、データベース内の Names テーブルのデータを返す SELECT クエリに割り当てます。

   ●クエリを実行し、ULDataReader として結果セットを返します。

   ●結果セットの行数と同じ長さの整数配列をインスタンス化します。

   ●ULDataReader で返される名前をリストボックスに移植し、ULDataReader で返される ID を整数配列に移植します。

   ●ULDataReader を閉じます。

   ●エラーが発生した場合は、エラーメッセージが出力されます。SQL エラーの場合は、エラーコードも出力されます。

   C# の場合は、Form1 クラスのメソッドとして次のコードをアプリケーションに追加します。

```
//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();
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
                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 );
    }
}
```

Visual Basic の場合は、Form1 クラスのメソッドとして次のコードをアプリケーションに追加します。

```
'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. プロジェクトをビルドします。

   プロジェクトのビルドによってエラーが発生しないようにします。

♦ INSERT、UPDATE、DELETE の実装

1. [フォーム デザイン] タブで **[Insert]** をダブルクリックして、btnInsert_Click メソッドを作成します。このメソッドでは、次のタスクを実行します。

   ●ULCommand オブジェクトをインスタンス化し、テキストボックス内の値をデータベースに挿入する INSERT 文に割り当てます。

   ●文を実行します。

   ●ULCommand オブジェクトを破棄します。

   ●リストボックスをリフレッシュします。

●エラーが発生した場合は、エラーメッセージが出力されます。SQL エラーの場合は、エラーコードも出力されます。

C# の場合は、次のコードを btnInsert_Click メソッドに追加します。

```
//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 );
}
```

Visual Basic の場合は、次のコードを btnInsert_Click メソッドに追加します。

```
'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
```

2. [フォーム デザイン] タブで **[Update]** をダブルクリックして、btnUpdate_Click メソッドを作成します。このメソッドでは、次のタスクを実行します。

●ULCommand オブジェクトをインスタンス化し、関連 ID に基づいて、テキストボックス内の値をデータベースに挿入する UPDATE 文に割り当てます。

●文を実行します。

●ULCommand オブジェクトを破棄します。

●リストボックスをリフレッシュします。

●エラーが発生した場合は、エラーメッセージが出力されます。SQL エラーの場合は、エラーコードも出力されます。

C# の場合は、次のコードを btnUpdate_Click メソッドに追加します。

```
//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);
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
            RowsUpdated = cmd.ExecuteNonQuery();
        }
        RefreshListBox();
    }
    catch( Exception err ) {
        MessageBox.Show(
            "Exception: " + err.Message);
    }
```

Visual Basic の場合は、次のコードを btnUpdate_Click メソッドに追加します。

```
'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
```

3. [フォーム デザイン] タブで **[Delete]** をダブルクリックして、btnDelete_Click メソッドを作成します。コードを追加し、次のタスクを実行します。

   ●ULCommand オブジェクトをインスタンス化し、DELETE 文に割り当てます。DELETE 文は、整数配列 ids からの関連 ID に基づいて、選択した行をデータベースから削除します。

   ●文を実行します。

   ●ULCommand オブジェクトを破棄します。

   ●リストボックスをリフレッシュします。

   ●エラーが発生した場合は、エラーメッセージが表示されます。SQL エラーの場合は、エラーコードも表示されます。

   C# の場合は、次のコードを btnDelete_Click メソッドに追加します。

```
//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 );
}
```

   Visual Basic の場合は、次のコードを btnDelete_Click メソッドに追加します。

```
'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
```

4. アプリケーションをビルドし、正常にコンパイルされることを確認します。

参照

- 「ULTable クラスを使用したデータの作成と修正」8 ページ
- エラーメッセージ

レッスン 5：アプリケーションのビルドと配置

次の手順では、アプリケーションをビルドし、リモートデバイスまたはエミュレーターに配置します。

♦ アプリケーションの配置

1. ソリューションをビルドします。

   アプリケーションがエラーなくビルドされることを確認します。

2. 配置ターゲットを選択します。

   配置ターゲットは、アプリケーションに含める *ulnet12.dll* のバージョンと一致している必要があります。

3. **[デバッグ]** » **[開始]** をクリックします。

   これにより、アプリケーションが格納された実行可能ファイルがビルドされ、エミュレーターに配置されます。このプロセスには時間がかかることがあります。アプリケーションを実行する前に .NET Compact Framework を配置する必要がある場合は、特に時間がかかります。

配置のトラブルシューティングのチェックリスト

エラーがレポートされた場合は、次のチェックリストを使用して、配置が正常に完了したかどうかをチェックします。

- アプリケーションが *¥Program Files¥appname* に配置されていることを確認します。*appname* は、レッスン 1 でアプリケーションに付けた名前 (CSApp または VBApp) です。

- アプリケーションコード内のデータベースファイルへのパスが正しいことを確認します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- データベースファイルをプロジェクトに追加するときに [リンク ファイル] を選択し、[ビルド アクション] を [コンテンツ] に設定し、[出力ディレクトリにコピー] を [常にコピーする] に設定したことを確認します。これらのオプションを適切に設定しなかった場合は、ファイルがデバイスに配置されません。

- ターゲットプラットフォームに対して適切なバージョンの *ulnet12.dll* への参照を追加したこと、または Windows Mobile インストーラーを実行したことを確認します。Windows Mobile 5.0 より前のバージョンの Windows Mobile で、エミュレーターと実際のデバイスの切り替えを行う場合は、使用するライブラリのバージョンを変更する必要があります。

- エミュレーターの状態を保存しないでエミュレーターを終了できます。アプリケーションの再配置を行うと、必要なすべてのファイルがエミュレーターにコピーされ、バージョン上の問題がないことが確認されます。

♦ アプリケーションのテスト

1. データベースにデータを挿入します。

   テキストボックスに名前を入力し、**[Insert]** をクリックします。名前がリストボックスに表示されます。

2. データベース内のデータを更新します。

   リストボックスで名前をクリックします。テキストボックスに新しい名前を入力します。**[Update]** をクリックします。新しい名前が、リストボックスの古い名前の代わりに表示されます。

3. データベースからデータを削除します。

   リストで名前をクリックします。**[Delete]** をクリックします。名前はリストに表示されなくなります。

このチュートリアルは、これで終了です。

C# チュートリアルのコードリスト

この章で説明したチュートリアルプログラムの完全なコードを次に示します。

```
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 =
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
                    "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);
        }
    }
  }
}
```

Visual Basic チュートリアルのコードリスト

```
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)
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
            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
```

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Ultra Light.NET API リファレンス

この章では、.NET Framework 2.0 と .NET Compact Framework 2.0 用 Ultra Light.NET データプロバイダーの API について説明します。

SQL Anywhere の ADO.NET 用データプロバイダーで利用できない Ultra Light.NET 拡張については、この API リファレンスでは **UL 拡張：** と示します。

Ultra Light.NET の Ultra Light エンジンのランタイムを使用するには、RuntimeType プロパティを適切な値に設定してから、他の Ultra Light.NET API を使用します。
ULDatabaseManager.RuntimeType プロパティ [Ultra Light.NET]227 ページを参照してください。

データベースで操作を実行するには、アプリケーションは接続を開く必要があります。接続を開くには、ULConnection クラス [Ultra Light.NET]116 ページを使用します。

iAnywhere.Data.UltraLite アセンブリは、**iAnywhere.Data.UltraLite.resources** というサテライトリソースアセンブリを使用します。メインアセンブリは、このリソースアセンブリを次の順序で国別に検索します。

- CultureInfo.CurrentUICulture：http://msdn.microsoft.com/ja-jp/library/System.Globalization.CultureInfo.CurrentUICulture.aspx
- CultureInfo.CurrentCulture：http://msdn.microsoft.com/ja-jp/library/System.Globalization.CultureInfo.CurrentCulture.aspx
- **EN**

この章の多くのプロパティとメソッドは、.NET Framework Data Provider for OLE DB (System.Data.OleDb) とよく似ています。詳細と例については、Microsoft .NET Framework のマニュアルを参照してください。

ネームスペース

```
iAnywhere.Data.UltraLite
```

ULActiveSyncListener インターフェイス

UL 拡張： ActiveSync イベントを受信するリスナーインターフェイスです。

Visual Basic 構文

```
Public Interface ULActiveSyncListener
```

C# 構文

```
public interface ULActiveSyncListener
```

メンバー

継承されたメンバーを含む ULActiveSyncListener インターフェイスのすべてのメンバー。

名前	説明
ActiveSyncInvoked メソッド	ActiveSync 用の Mobile Link プロバイダーがアプリケーションを呼び出して同期を実行するときに起動されます。

ActiveSyncInvoked メソッド

ActiveSync 用の Mobile Link プロバイダーがアプリケーションを呼び出して同期を実行するときに起動されます。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **launchedByProvider** ActiveSync 同期を実行するために、Mobile Link プロバイダーがアプリケーションを起動した場合は true です。同期の完了後、アプリケーションは自動的に停止しなければなりません。ActiveSync 用 Mobile Link プロバイダーが呼び出したときに、すでにアプリケーションが実行されていた場合は false です。

備考

このメソッドは、別のスレッドによって呼び出されます。マルチスレッドの問題を回避するには、このメソッドがユーザーインターフェイスにイベントを通知する必要があります。マルチスレッドを使用する場合は、スレッドごとに別々の接続を使用し、lock キーワードを使用して、アプリケーションの共有オブジェクトにアクセスすることをおすすめします。

同期が完了したら、アプリケーションは ULDatabaseManager.SignalSyncIsComplete メソッドを呼び出して、ActiveSync 用の Mobile Link プロバイダーに対して通知する必要があります。

参照

- ULDatabaseManager.SignalSyncIsComplete メソッド [Ultra Light.NET]226 ページ

例

```
' 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()
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

' 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

対応する C# 言語のコードを次に示します。

```
 // 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.
      //
      ULConnection.DatabaseManager.SetActiveSyncListener( _
         "myCompany.myapp", this _
      );

      // Create connection
      ...
   }

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

   protected override void OnClosing( _
      System.ComponentModel.CancelEventArgs e )
   {
      ULConnection.DatabaseManager.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();
        ULConnection.DatabaseManager.SignalSyncIsComplete();
    }
}
```

ULBulkCopy クラス

別のソースのデータを使用して、Ultra Light テーブルを効率的にバルクロードします。

Visual Basic 構文

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

C# 構文

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

基本クラス

- System.IDisposable

メンバー

継承されたメンバーを含む ULBulkCopy クラスのすべてのメンバー。

名前	説明
ULBulkCopy コンストラクター	ULBulkCopy オブジェクトを、指定された ULConnection オブジェクトで初期化します。
Close メソッド	ULBulkCopy オブジェクトを閉じます。
Dispose メソッド	ULBulkCopy オブジェクトを破棄します。
WriteToServer メソッド	指定された System.Data.DataRow オブジェクトの配列内のすべてのローを、ULBulkCopy オブジェクトの DestinationTableName フィールドで指定される送信先テーブルにコピーします。
BatchSize プロパティ	各バッチの中のローの数を取得または設定します。
BulkCopyTimeOut プロパティ	タイムアウトするまでに完了するオペレーションの秒数を取得または設定します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
ColumnMappings プロパティ	ULBulkCopyColumnMapping 項目のコレクションを返します。
DestinationTableName プロパティ	サーバー上の送信先テーブルの名前を取得または設定します。
NotifyAfter プロパティ	通知イベントが生成されるまでに処理されるローの数を指定します。
ULRowsCopied イベント	NotifyAfter プロパティで指定される数のローが処理されるたびに、このイベントが発生します。

備考

ULBulkCopy クラスは、.NET Compact Framework 2.0 では使用できません。

ULBulkCopy コンストラクター

ULBulkCopy オブジェクトを、指定された ULConnection オブジェクトで初期化します。

オーバーロードリスト

名前	説明
ULBulkCopy(string) コンストラクター	ULBulkCopy オブジェクトを、指定された接続文字列で初期化します。
ULBulkCopy(string, ULBulkCopyOptions) コンストラクター	ULBulkCopy オブジェクトを、指定された接続文字列とコピーオプションで初期化します。
ULBulkCopy(ULConnection) コンストラクター	ULBulkCopy オブジェクトを、指定された ULConnection オブジェクトで初期化します。
ULBulkCopy(ULConnection, ULBulkCopyOptions, ULTransaction) コンストラクター	ULBulkCopy オブジェクトを、指定された ULConnection オブジェクト、コピーオプション、ULTransaction オブジェクトで初期化します。

ULBulkCopy(string) コンストラクター

ULBulkCopy オブジェクトを、指定された接続文字列で初期化します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **connectionString**　ULBulkCopy オブジェクトによって使用されるために開かれる接続を定義する文字列。接続文字列は keyword=value のペアがセミコロンで区切られたリストです。

備考

ULBulkCopy クラスは、.NET Compact Framework 2.0 では使用できません。

この構文で、connectionString 値による WriteToServer メソッド呼び出し中に接続が開かれます。WriteToServer 呼び出しが終了すると、接続が閉じます。

接続文字列は、ULConnectionParms オブジェクトを使用して指定できます。

参照

- ULConnection.ConnectionString プロパティ [Ultra Light.NET]155 ページ
- ULConnectionParms クラス [Ultra Light.NET]164 ページ
- System.IDisposable

ULBulkCopy(string, ULBulkCopyOptions) コンストラクター

ULBulkCopy オブジェクトを、指定された接続文字列とコピーオプションで初期化します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **connectionString**　ULBulkCopy オブジェクトによって使用されるために開かれる接続を定義する文字列。接続文字列は keyword=value のペアがセミコロンで区切られたリストです。

- **copyOptions**　目的のテーブルにデータソースローをコピーする方法を決定する、ULBulkCopyOptions 列挙体の値の組み合わせ。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

ULBulkCopy クラスは、.NET Compact Framework 2.0 では使用できません。

この構文で、connectionString 値による WriteToServer メソッド呼び出し中に接続が開かれます。WriteToServer 呼び出しが終了すると、接続が閉じます。

参照

- ULConnection.ConnectionString プロパティ [Ultra Light.NET]155 ページ
- ULBulkCopyOptions 列挙体 [Ultra Light.NET]458 ページ

ULBulkCopy(ULConnection) コンストラクター

ULBulkCopy オブジェクトを、指定された ULConnection オブジェクトで初期化します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **接続**　バルクコピーオペレーションの実行に使用する、すでに開いている ULConnection オブジェクト。接続が開いていない場合は、WriteToServer メソッド呼び出し中に例外がスローされます。

備考

ULBulkCopy クラスは、.NET Compact Framework 2.0 では使用できません。

参照

- ULConnection クラス [Ultra Light.NET]116 ページ

ULBulkCopy(ULConnection, ULBulkCopyOptions, ULTransaction) コンストラクター

ULBulkCopy オブジェクトを、指定された ULConnection オブジェクト、コピーオプション、ULTransaction オブジェクトで初期化します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **connection** バルクコピーオペレーションの実行に使用する、すでに開いている ULConnection オブジェクト。接続が開いていない場合は、WriteToServer メソッド呼び出し中に例外がスローされます。
- **copyOptions** 目的のテーブルにデータソースローをコピーする方法を決定する、ULBulkCopyOptions 列挙体の値の組み合わせ。
- **externalTransaction** バルクコピーが発生する、既存の ULTransaction オブジェクト。この値が NULL 参照 (Visual Basic の Nothing) でない場合は、バルクコピー操作が内部で実行されます。外部トランザクションと ULBulkCopyOptions.UseInternalTransaction オプションの両方を指定すると、エラーになります。

備考

ULBulkCopy クラスは、.NET Compact Framework 2.0 では使用できません。

参照

- ULConnection クラス [Ultra Light.NET]116 ページ
- ULTransaction クラス [Ultra Light.NET]451 ページ

Close メソッド

ULBulkCopy オブジェクトを閉じます。

Visual Basic 構文

```
Public Sub Close()
```

C# 構文

```
public void Close()
```

Dispose メソッド

ULBulkCopy オブジェクトを破棄します。

Visual Basic 構文

```
Public Sub Dispose()
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

C# 構文

```
public void Dispose()
```

WriteToServer メソッド

指定された System.Data.DataRow オブジェクトの配列内のすべてのローを、ULBulkCopy オブジェクトの DestinationTableName フィールドで指定される送信先テーブルにコピーします。

オーバーロードリスト

名前	説明
WriteToServer(DataRow[]) メソッド	指定された System.Data.DataRow オブジェクトの配列内のすべてのローを、ULBulkCopy オブジェクトの DestinationTableName フィールドで指定される送信先テーブルにコピーします。
WriteToServer(DataTable) メソッド	指定された System.Data.DataTable のすべてのローを、ULBulkCopy.DestinationTableName プロパティで指定される送信先テーブルにコピーします。
WriteToServer(DataTable, DataRowState) メソッド	指定されたローの状態を持つ指定された System.Data.DataTable のすべてのローを、ULBulkCopy.DestinationTableName プロパティで指定される送信先テーブルにコピーします。
WriteToServer(IDataReader) メソッド	指定された System.Data.DataReader のすべてのローを、ULBulkCopy.DestinationTableName プロパティで指定される送信先テーブルにコピーします。

WriteToServer(DataRow[]) メソッド

指定された System.Data.DataRow オブジェクトの配列内のすべてのローを、ULBulkCopy オブジェクトの DestinationTableName フィールドで指定される送信先テーブルにコピーします。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **rows** 送信先テーブルにコピーされる System.Data.DataRow オブジェクトの配列。

参照

- ULBulkCopy.DestinationTableName プロパティ [Ultra Light.NET]49 ページ
- System.Data.DataRow

WriteToServer(DataTable) メソッド

指定された System.Data.DataTable のすべてのローを、ULBulkCopy.DestinationTableName プロパティで指定される送信先テーブルにコピーします。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **table** ローが送信先テーブルにコピーされる System.Data.DataTable。

参照

- ULBulkCopy.DestinationTableName プロパティ [Ultra Light.NET]49 ページ
- System.Data.DataTable

WriteToServer(DataTable, DataRowState) メソッド

指定されたローの状態を持つ指定された System.Data.DataTable のすべてのローを、ULBulkCopy.DestinationTableName プロパティで指定される送信先テーブルにコピーします。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **table** ローが送信先テーブルにコピーされる System.Data.DataTable。
- **rowState** System.Data.DataRowState 列挙体の値。ローステータスに一致するローのみ、送信先にコピーされます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

rowState パラメーターが指定されている場合、ローステータスが同じローだけがコピーされます。

参照

- ULBulkCopy.DestinationTableName プロパティ [Ultra Light.NET]49 ページ
- System.Data.DataTable
- System.Data.DataRowState

WriteToServer(IDataReader) メソッド

指定された System.Data.DataReader のすべてのローを、ULBulkCopy.DestinationTableName プロパティで指定される送信先テーブルにコピーします。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **reader** ローが送信先テーブルにコピーされる System.Data.IDataReader。

参照

- ULBulkCopy.DestinationTableName プロパティ [Ultra Light.NET]49 ページ
- System.Data.IDataReader

BatchSize プロパティ

各バッチの中のローの数を取得または設定します。

Visual Basic 構文

```
Public Property BatchSize As Integer
```

C# 構文

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

備考

各バッチが終了すると、バッチ内のローがサーバーに送信されます。

各バッチの中のロー数。デフォルトは 0 です。

0 に設定すると、すべてのローが 1 つのバッチで送信されます。

0 未満の値を設定すると、エラーになります。

バッチの進行中にこの値が変更された場合、現在のバッチはそのまま完了し、それ以降のバッチが新しい値を使用します。

BulkCopyTimeOut プロパティ

タイムアウトするまでに完了するオペレーションの秒数を取得または設定します。

Visual Basic 構文

```
Public Property BulkCopyTimeout As Integer
```

C# 構文

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

備考

デフォルト値は 30 秒です。

ゼロという値の設定は無制限を意味し、待機時間が無制限になることがあるため、避けてください。

オペレーションがタイムアウトすると、現在のトランザクション内のすべてのローがロールバックされ、SAException エラーがスローされます。

ゼロ未満の値を設定すると、エラーがスローされます。

ColumnMappings プロパティ

ULBulkCopyColumnMapping 項目のコレクションを返します。

Visual Basic 構文

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

C# 構文

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

備考

カラムマッピングは、データソース内のカラムと、送信先のカラムの間の関係を定義します。

デフォルトでは、空のコレクションです。

WriteToServer メソッド呼び出しの実行中は、プロパティを変更できません。

WriteToServer メソッドの実行時に ColumnMappings オブジェクトが空の場合は、ソース内の先頭のカラムが送信先の先頭のカラムにマッピングされ、2 番目は 2 番目にマッピングされます。以

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

降についても同様です。この処理は、カラムの型が変換可能な場合、ソースカラム以上の送信先カラムがある場合、余分な送信先カラムが NULL 入力可のカラムである場合に行われます。

参照

- ULBulkCopyColumnMapping クラス [Ultra Light.NET]50 ページ

DestinationTableName プロパティ

サーバー上の送信先テーブルの名前を取得または設定します。

Visual Basic 構文

```
Public Property DestinationTableName As String
```

C# 構文

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

備考

デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

WriteToServer 呼び出しの実行時に値が変更されても、変更は反映されません。

WriteToServer メソッドへの呼び出しの前に値が設定されていない場合、InvalidOperationException エラーがスローされます。

値を NULL (Visual Basic の場合は Nothing) または空の文字列に設定すると、エラーになります。

NotifyAfter プロパティ

通知イベントが生成されるまでに処理されるローの数を指定します。

Visual Basic 構文

```
Public Property NotifyAfter As Integer
```

C# 構文

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

備考

通知イベントが生成されるまでに処理されるローの数を表す整数。プロパティが設定されていない場合は 0。

WriteToServer メソッドの実行時にこのプロパティに加えられる変更は、次の通知まで反映されません。

この値をゼロ未満の値に設定すると、エラーがスローされます。

NotifyAfter と BulkCopyTimeout プロパティの値は相互に排他的であるため、データベースにローが送信されない場合や、コミットされない場合も、イベントは起動します。

参照

- ULBulkCopy.BulkCopyTimeout プロパティ [Ultra Light.NET]48 ページ

ULRowsCopied イベント

NotifyAfter プロパティで指定される数のローが処理されるたびに、このイベントが発生します。

Visual Basic 構文

```
Public Event ULRowsCopied As ULRowsCopiedEventHandler
```

C# 構文

```
public event ULRowsCopiedEventHandler ULRowsCopied;
```

備考

ULRowsCopied イベントの受信は、ローがコミットされたことを意味するわけではありません。このイベントから Close メソッドを呼び出すことはできません。

参照

- ULBulkCopy.NotifyAfter プロパティ [Ultra Light.NET]49 ページ

ULBulkCopyColumnMapping クラス

ULBulkCopy インスタンスのデータソース内のカラムと、インスタンスの送信先テーブル内のカラムの間のマッピングを定義します。

Visual Basic 構文

```
Public NotInheritable Class ULBulkCopyColumnMapping
```

C# 構文

```
public sealed class ULBulkCopyColumnMapping
```

メンバー

継承されたメンバーを含む ULBulkCopyColumnMapping クラスのすべてのメンバー。

名前	説明
ULBulkCopyColumnMapping コンストラクター	新しいカラムマッピングを作成します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
DestinationColumn プロパティ	マッピングされる送信先データベーステーブルのカラムの名前を指定します。
DestinationOrdinal プロパティ	マッピングされる送信先データベーステーブルにおけるカラムの順序を指定します。
SourceColumn プロパティ	データソースにマッピングされるカラムの名前を指定します。
SourceOrdinal プロパティ	データソース内のソースカラムの序数位置を指定します。

備考

ULBulkCopyColumnMapping クラスは、.NET Compact Framework 2.0 では使用できません。

参照

- ULBulkCopy クラス [Ultra Light.NET]40 ページ

ULBulkCopyColumnMapping コンストラクター

新しいカラムマッピングを作成します。

オーバーロードリスト

名前	説明
ULBulkCopyColumnMapping() コンストラクター	新しいカラムマッピングを作成します。
ULBulkCopyColumnMapping(int, int) コンストラクター	カラムの順序または名前を使用してソースカラムと送信先カラムを参照する、新しいカラムマッピングを作成します。
ULBulkCopyColumnMapping(int, string) コンストラクター	ソースカラムを参照するカラムの順序と、送信先カラムを参照するカラム名を使用して、新しいカラムマッピングを作成します。
ULBulkCopyColumnMapping(string, int) コンストラクター	ソースカラムを参照するカラム名と、送信先カラムを参照するカラムの順序を使用して、新しいカラムマッピングを作成します。
ULBulkCopyColumnMapping(string, string) コンストラクター	カラムの名前を使用してソースカラムと送信先カラムを参照する、新しいカラムマッピングを作成します。

ULBulkCopyColumnMapping() コンストラクター

新しいカラムマッピングを作成します。

Visual Basic 構文

```
Public Sub New()
```

C# 構文

```
public ULBulkCopyColumnMapping()
```

備考

ULBulkCopyColumnMapping クラスは、.NET Compact Framework 2.0 では使用できません。

ULBulkCopyColumnMapping(int, int) コンストラクター

カラムの順序または名前を使用してソースカラムと送信先カラムを参照する、新しいカラムマッピングを作成します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **sourceColumnOrdinal**　データソース内のソースカラムの序数位置。データソースの最初のカラムの序数位置は 0 です。
- **destinationColumnOrdinal**　送信先テーブル内の送信先カラムの序数位置。テーブルの最初のカラムの序数位置は 0 です。

備考

ULBulkCopyColumnMapping クラスは、.NET Compact Framework 2.0 では使用できません。

ULBulkCopyColumnMapping(int, string) コンストラクター

ソースカラムを参照するカラムの順序と、送信先カラムを参照するカラム名を使用して、新しいカラムマッピングを作成します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

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

C# 構文

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

パラメーター

- **sourceColumnOrdinal** データソース内のソースカラムの序数位置。データソースの最初のカラムの序数位置は 0 です。
- **destinationColumn** 送信先テーブル内の送信先カラムの名前。

備考

ULBulkCopyColumnMapping クラスは、.NET Compact Framework 2.0 では使用できません。

ULBulkCopyColumnMapping(string, int) コンストラクター

ソースカラムを参照するカラム名と、送信先カラムを参照するカラムの順序を使用して、新しいカラムマッピングを作成します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **sourceColumn** データソース内のソースカラムの名前。
- **destinationColumnOrdinal** 送信先テーブル内の送信先カラムの序数位置。テーブルの最初のカラムの序数位置は 0 です。

備考

ULBulkCopyColumnMapping クラスは、.NET Compact Framework 2.0 では使用できません。

ULBulkCopyColumnMapping(string, string) コンストラクター

カラムの名前を使用してソースカラムと送信先カラムを参照する、新しいカラムマッピングを作成します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **sourceColumn**　データソース内のソースカラムの名前。
- **destinationColumn**　送信先テーブル内の送信先カラムの名前。

備考

ULBulkCopyColumnMapping クラスは、.NET Compact Framework 2.0 では使用できません。

DestinationColumn プロパティ

マッピングされる送信先データベーステーブルのカラムの名前を指定します。

Visual Basic 構文

```
Public Property DestinationColumn As String
```

C# 構文

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

備考

送信先テーブルのカラムの名前を指定する文字列。DestinationOrdinal プロパティに優先度がない場合は NULL 参照 (Visual Basic の Nothing)。

DestinationColumn プロパティと DestinationOrdinal プロパティは、相互に排他的です。直前に設定された値が優先されます。

DestinationColumn プロパティを設定すると、DestinationOrdinal プロパティは -1 に設定されます。DestinationOrdinal プロパティを設定すると、DestinationColumn プロパティは NULL 参照 (Visual Basic の Nothing) に設定されます。

DestinationColumn プロパティを NULL または空の文字列に設定すると、エラーになります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULBulkCopy.ColumnMappings.DestinationOrdinal プロパティ [Ultra Light.NET]55 ページ

DestinationOrdinal プロパティ

マッピングされる送信先データベーステーブルにおけるカラムの順序を指定します。

Visual Basic 構文

```
Public Property DestinationOrdinal As Integer
```

C# 構文

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

備考

マッピングされるカラムの送信先テーブルにおける順序を指定する整数。プロパティが設定されていない場合は -1。

DestinationColumn プロパティと DestinationOrdinal プロパティは、相互に排他的です。直前に設定された値が優先されます。

DestinationColumn プロパティを設定すると、DestinationOrdinal プロパティは -1 に設定されます。DestinationOrdinal プロパティを設定すると、DestinationColumn プロパティは NULL 参照 (Visual Basic の Nothing) に設定されます。

参照

- ULBulkCopy.ColumnMappings.DestinationColumn プロパティ [Ultra Light.NET]54 ページ

SourceColumn プロパティ

データソースにマッピングされるカラムの名前を指定します。

Visual Basic 構文

```
Public Property SourceColumn As String
```

C# 構文

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

備考

データソースのカラムの名前を指定する文字列。SourceOrdinal に優先度がない場合は NULL 参照 (Visual Basic の Nothing)。

SourceColumn プロパティと SourceOrdinal プロパティは、相互に排他的です。直前に設定された値が優先されます。

SourceColumn プロパティを設定すると、SourceOrdinal プロパティは -1 に設定されます。SourceOrdinal プロパティを設定すると、SourceColumn プロパティは NULL 参照 (Visual Basic の Nothing) に設定されます。

SourceColumn プロパティを NULL または空の文字列に設定すると、エラーになります。

参照

- ULBulkCopy.ColumnMappings.SourceOrdinal プロパティ [Ultra Light.NET]56 ページ

SourceOrdinal プロパティ

データソース内のソースカラムの序数位置を指定します。

Visual Basic 構文

```
Public Property SourceOrdinal As Integer
```

C# 構文

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

備考

データソースのカラムの順序を指定する整数。プロパティが設定されていない場合は -1。

SourceColumn プロパティと SourceOrdinal プロパティは、相互に排他的です。直前に設定された値が優先されます。

SourceColumn プロパティを設定すると、SourceOrdinal プロパティは -1 に設定されます。SourceOrdinal プロパティを設定すると、SourceColumn プロパティは NULL 参照 (Visual Basic の Nothing) に設定されます。

参照

- ULBulkCopy.ColumnMappings.SourceColumn プロパティ [Ultra Light.NET]55 ページ

ULBulkCopyColumnMappingCollection クラス

System.Collections.CollectionBase から継承した ULBulkCopyColumnMapping オブジェクトのコレクションです。

Visual Basic 構文

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

C# 構文

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

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

基本クラス

- System.Collections.CollectionBase

メンバー

継承されたメンバーを含む ULBulkCopyColumnMappingCollection クラスのすべてのメンバー。

名前	説明
Add メソッド	指定された ULBulkCopyColumnMapping オブジェクトをコレクションに追加します。
Clear メソッド (System.Collections.CollectionBase から継承)	System.Collections.CollectionBase インスタンスからすべてのオブジェクトを削除します。
Contains メソッド	コレクション内に指定された ULBulkCopyColumnMapping オブジェクトが存在するかどうかを返します。
CopyTo メソッド	ULBulkCopyColumnMappingCollection オブジェクトの要素を、特定のインデックスを先頭に、ULBulkCopyColumnMapping オブジェクトの配列にコピーします。
GetEnumerator メソッド (System.Collections.CollectionBase から継承)	System.Collections.CollectionBase インスタンスで反復処理する列挙子を返します。
IndexOf メソッド	コレクション内で、指定された ULBulkCopyColumnMapping オブジェクトのインデックスを返します。
OnClear メソッド (System.Collections.CollectionBase から継承)	System.Collections.CollectionBase インスタンスの内容をクリアするときに、追加のカスタム処理を実行します。
OnClearComplete メソッド (System.Collections.CollectionBase から継承)	System.Collections.CollectionBase インスタンスの内容をクリアした後に、追加のカスタム処理を実行します。
OnInsert メソッド (System.Collections.CollectionBase から継承)	System.Collections.CollectionBase インスタンスに新しい要素を挿入する前に、追加のカスタム処理を実行します。
OnInsertComplete メソッド (System.Collections.CollectionBase から継承)	System.Collections.CollectionBase インスタンスに新しい要素を挿入した後に、追加のカスタム処理を実行します。

名前	説明
OnRemove メソッド (System.Collections.CollectionBase から継承)	System.Collections.CollectionBase インスタンスから要素を削除するときに、追加のカスタム処理を実行します。
OnRemoveComplete メソッド (System.Collections.CollectionBase から継承)	System.Collections.CollectionBase インスタンスから要素を削除した後に、追加のカスタム処理を実行します。
OnSet メソッド (System.Collections.CollectionBase から継承)	System.Collections.CollectionBase インスタンスに値を設定する前に、追加のカスタム処理を実行します。
OnSetComplete メソッド (System.Collections.CollectionBase から継承)	System.Collections.CollectionBase インスタンスに値を設定した後に、追加のカスタム処理を実行します。
OnValidate メソッド (System.Collections.CollectionBase から継承)	値を検証するときに、追加のカスタム処理を実行します。
Remove メソッド	指定された ULBulkCopyColumnMapping オブジェクトを ULBulkCopyColumnMappingCollection オブジェクトから削除します。
RemoveAt メソッド	指定されたインデックス位置のマッピングをコレクションから削除します。
Capacity property (System.Collections.CollectionBase から継承)	System.Collections.CollectionBase に含めることのできる要素数を取得または設定します。
Count property (System.Collections.CollectionBase から継承)	System.Collections.CollectionBase インスタンスに含まれている要素数を取得します。
InnerList property (System.Collections.CollectionBase から継承)	System.Collections.CollectionBase インスタンスの要素リストに含まれている System.Collections.ArrayList を取得します。
List property (System.Collections.CollectionBase から継承)	System.Collections.CollectionBase インスタンスの要素リストに含まれている System.Collections.IList を取得します。
this プロパティ	指定されたインデックス位置の ULBulkCopyColumnMapping オブジェクトを取得します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

ULBulkCopyColumnMappingCollection クラスは、.NET Compact Framework 2.0 では使用できません。

参照

- ULBulkCopyColumnMapping クラス [Ultra Light.NET]50 ページ

Add メソッド

指定された ULBulkCopyColumnMapping オブジェクトをコレクションに追加します。

オーバーロードリスト

名前	説明
Add(int, int) メソッド	ソースカラムと送信先カラムの両方を指定する順序を使用して、新しい ULBulkCopyColumnMapping オブジェクトを作成し、このマッピングをコレクションに追加します。
Add(int, string) メソッド	カラムの順序を使用してソースカラムを参照し、カラム名を使用して送信先カラムを参照する、新しい ULBulkCopyColumnMapping オブジェクトを作成し、このマッピングをコレクションに追加します。
Add(string, int) メソッド	カラム名を使用してソースカラムを参照し、カラムの順序を使用して送信先カラムを参照する、新しい ULBulkCopyColumnMapping オブジェクトを作成し、このマッピングをコレクションに追加します。
Add(string, string) メソッド	ソースカラムと送信先カラムの両方を指定するカラム名を使用して、新しい ULBulkCopyColumnMapping オブジェクトを作成し、このマッピングをコレクションに追加します。
Add(ULBulkCopyColumnMapping) メソッド	指定された ULBulkCopyColumnMapping オブジェクトをコレクションに追加します。

Add(int, int) メソッド

ソースカラムと送信先カラムの両方を指定する順序を使用して、新しい ULBulkCopyColumnMapping オブジェクトを作成し、このマッピングをコレクションに追加します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **sourceColumnOrdinal** データソース内のソースカラムの序数位置。データソースの最初のカラムの序数位置は 0 です。
- **destinationColumnOrdinal** 送信先テーブル内の送信先カラムの序数位置。テーブルの最初のカラムの序数位置は 0 です。

備考

ULBulkCopyColumnMappingCollection クラスは、.NET Compact Framework 2.0 では使用できません。

参照

- ULBulkCopyColumnMapping クラス [Ultra Light.NET]50 ページ

Add(int, string) メソッド

カラムの順序を使用してソースカラムを参照し、カラム名を使用して送信先カラムを参照する、新しい ULBulkCopyColumnMapping オブジェクトを作成し、このマッピングをコレクションに追加します。

Visual Basic 構文

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

C# 構文

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

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
    string destinationColumn
)
```

パラメーター

- **sourceColumnOrdinal**　データソース内のソースカラムの序数位置。データソースの最初のカラムの序数位置は 0 です。
- **destinationColumn**　送信先テーブル内の送信先カラムの名前。

備考

ULBulkCopyColumnMappingCollection クラスは、.NET Compact Framework 2.0 では使用できません。

参照

- ULBulkCopyColumnMapping クラス [Ultra Light.NET]50 ページ

Add(string, int) メソッド

カラム名を使用してソースカラムを参照し、カラムの順序を使用して送信先カラムを参照する、新しい ULBulkCopyColumnMapping オブジェクトを作成し、このマッピングをコレクションに追加します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **sourceColumn**　データソース内のソースカラムの名前。
- **destinationColumnOrdinal**　送信先テーブル内の送信先カラムの序数位置。テーブルの最初のカラムの序数位置は 0 です。

備考

カラムの順序または名前を使用してソースカラムと送信先カラムを参照する、新しいカラムマッピングを作成します。

ULBulkCopyColumnMappingCollection クラスは、.NET Compact Framework 2.0 では使用できません。

参照

- ULBulkCopyColumnMapping クラス [Ultra Light.NET]50 ページ

Add(string, string) メソッド

ソースカラムと送信先カラムの両方を指定するカラム名を使用して、新しい ULBulkCopyColumnMapping オブジェクトを作成し、このマッピングをコレクションに追加します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **sourceColumn** データソース内のソースカラムの名前。
- **destinationColumn** 送信先テーブル内の送信先カラムの名前。

備考

ULBulkCopyColumnMappingCollection クラスは、.NET Compact Framework 2.0 では使用できません。

参照

- ULBulkCopyColumnMapping クラス [Ultra Light.NET]50 ページ

Add(ULBulkCopyColumnMapping) メソッド

指定された ULBulkCopyColumnMapping オブジェクトをコレクションに追加します。

Visual Basic 構文

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

C# 構文

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

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

パラメーター

- **bulkCopyColumnMapping**　コレクションに追加される、マッピングを記述する ULBulkCopyColumnMapping オブジェクト。

備考

ULBulkCopyColumnMappingCollection クラスは、.NET Compact Framework 2.0 では使用できません。

参照

- ULBulkCopyColumnMapping クラス [Ultra Light.NET]50 ページ

Contains メソッド

コレクション内に指定された ULBulkCopyColumnMapping オブジェクトが存在するかどうかを返します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **value**　有効な ULBulkCopyColumnMapping オブジェクト。

戻り値

指定のマッピングがコレクション内にある場合は true、ない場合は false。

参照

- ULBulkCopyColumnMapping クラス [Ultra Light.NET]50 ページ

CopyTo メソッド

ULBulkCopyColumnMappingCollection オブジェクトの要素を、特定のインデックスを先頭に、ULBulkCopyColumnMapping オブジェクトの配列にコピーします。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **array** ULBulkCopyColumnMappingCollection オブジェクトの要素のコピー先である、1 次元の ULBulkCopyColumnMapping 配列。配列のインデックスは 0 から始まります。
- **index** コピーが開始される、配列内の 0 から始まるインデックス。

参照

- ULBulkCopyColumnMapping クラス [Ultra Light.NET]50 ページ

IndexOf メソッド

コレクション内で、指定された ULBulkCopyColumnMapping オブジェクトのインデックスを返します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **value** 検索対象の ULBulkCopyColumnMapping オブジェクト。

戻り値

カラムマッピングの 0 から始まるインデックス、またはコレクション内にカラムマッピングが見つからない場合は -1 が返されます。

参照

- ULBulkCopyColumnMapping クラス [Ultra Light.NET]50 ページ

Remove メソッド

指定された ULBulkCopyColumnMapping オブジェクトを ULBulkCopyColumnMappingCollection オブジェクトから削除します。

Visual Basic 構文

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

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

C# 構文

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

パラメーター

- **value** コレクションから削除する ULBulkCopyColumnMapping オブジェクト。

参照

- ULBulkCopyColumnMapping クラス [Ultra Light.NET]50 ページ

RemoveAt メソッド

指定されたインデックス位置のマッピングをコレクションから削除します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **index** コレクションから削除する ULBulkCopyColumnMapping オブジェクトの、0 から始まるインデックス。

this プロパティ

指定されたインデックス位置の ULBulkCopyColumnMapping オブジェクトを取得します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **index** 検索する ULBulkCopyColumnMapping オブジェクトの、0 から始まるインデックス。

備考

ULBulkCopyColumnMapping オブジェクトが返されます。

参照

- ULBulkCopyColumnMapping クラス [Ultra Light.NET]50 ページ

ULCommand クラス

IN パラメーターあり、または IN パラメーターなしで事前にコンパイルされた SQL 文またはクエリを表します。

Visual Basic 構文

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

C# 構文

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

基本クラス

- System.Data.Common.DbCommand
- System.ICloneable

メンバー

継承されたメンバーを含む ULCommand クラスのすべてのメンバー。

名前	説明
ULCommand コンストラクター	ULCommand オブジェクトを初期化します。
BeginExecuteNonQuery メソッド	この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始します。
BeginExecuteReader メソッド	この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始し、結果セットを取得します。
Cancel メソッド	このメソッドは Ultra Light.NET ではサポートされていません。
CreateParameter メソッド	ULCommand オブジェクトにパラメーターを指定するために ULParameter オブジェクトを提供します。
EndExecuteNonQuery メソッド	SQL 文の非同期実行を終了します。
EndExecuteReader メソッド	SQL 文の非同期実行を終了し、要求された ULDataReader を返します。
ExecuteDbDataReader method (System.Data.Common.DbCommand から継承)	接続に対してコマンドテキストを実行します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
ExecuteNonQuery メソッド	SQL INSERT 文、DELETE 文、UPDATE 文のように、結果セットを返さない文を実行します。
ExecuteReader メソッド	SQL SELECT 文を実行し、結果セットを返します。
ExecuteResultSet メソッド	**UL 拡張：** SQL SELECT 文を実行し、ULResultSet オブジェクトとして結果セットを返します。
ExecuteScalar メソッド	SQL SELECT 文を実行し、単一の値を返します。
ExecuteTable メソッド	**UL 拡張：** 直接の操作用に、ULTable オブジェクトのデータベーステーブルを取り出します。
Prepare メソッド	このコマンドの SQL 文を事前にコンパイルして格納します。
CommandText プロパティ	ULCommand.CommandType プロパティが System.Data.CommandType.TableDirect である場合の、SQL 文のテキストまたはテーブルの名前を指定します。
CommandTimeout プロパティ	この機能は Ultra Light.NET ではサポートされていません。
CommandType プロパティ	実行されるコマンドのタイプを指定します。
Connection プロパティ	ULCommand オブジェクトを実行する接続オブジェクトです。
DesignTimeVisible プロパティ	カスタマイズされた Windows Form Designer 制御で ULCommand オブジェクトを参照できるようにするかどうかを指定します。
IndexName プロパティ	**UL 拡張：** ULCommand.CommandType プロパティが System.Data.CommandType.TableDirect であるときに、テーブルを開く (ソートする) インデックスの名前を指定します。
Parameters プロパティ	現在の文のパラメーターを指定します。

名前	説明
Plan プロパティ	**UL 拡張：** Ultra Light.NET がクエリの実行に使用するアクセスプランを返します。
Transaction プロパティ	ULCommand オブジェクトが実行される ULTransaction オブジェクトを指定します。
UpdatedRowSource プロパティ	ULDataAdapterUpdate メソッドによって使用されるときにコマンドの結果が DataRow に適用される方法を指定します。

備考

このオブジェクトを使用すると、文またはクエリを効率的に何回も実行できます。

ULCommand オブジェクトは、直接作成したり、ULConnection.CreateCommand メソッドを使用して作成したりできます。このメソッドによって、コマンドは特定の接続で文を実行するための適切なトランザクションを使用できます。

ULCommand.Transaction メソッドは、現在のトランザクションがコミットまたはロールバックされた後でリセットする必要があります。

ULCommand クラスには、Ultra Light.NET データベースでコマンドを実行するための次のメソッドがあります。

メソッド	説明
ULCommand.ExecuteNonQuery	SQL INSERT 文、DELETE 文、UPDATE 文のように、結果セットを返さない文を実行します。
ULCommand.ExecuteReader()	SQL SELECT 文を実行し、ULDataReader オブジェクトで結果セットを返します。読み込み専用の結果セットの作成には、このメソッドを使用します。
ULCommand.ExecuteResultSet()	**UL 拡張：** SQL SELECT 文を実行し、ULResultSet オブジェクトで結果セットを返します。可変の結果セットの作成には、このメソッドを使用します。
ULCommand.ExecuteScalar	SQL SELECT 文を実行し、単一の値を返します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

メソッド	説明
ULCommand.ExecuteTable()	**UL 拡張：** 直接の操作用に、ULTable オブジェクトのデータベーステーブルを取り出します。ULCommand.CommandText プロパティはテーブルの名前として解釈され、ULCommand.IndexName プロパティはテーブルのソート順の指定に使用できます。ULCommand.CommandType プロパティは System.Data.CommandType.TableDirect であることが必要です。

ULCommand.CommandText プロパティを含め、ほとんどのプロパティをリセットできるほか、ULCommand オブジェクトを再利用できます。

リソース管理の理由により、コマンドを使用し終わったら、そのコマンドを明示的に破棄することをおすすめします。C# では、使用中の文を使って自動的に System.ComponentModel.Component.Dispose メソッドを呼び出すことができます。または、明示的に System.ComponentModel.Component.Dispose メソッドを呼び出すことができます。Visual Basic では、常に明示的に System.ComponentModel.Component.Dispose メソッドを呼び出します。

参照

- ULConnection.CreateCommand メソッド [Ultra Light.NET]130 ページ
- ULCommand.Transaction プロパティ [Ultra Light.NET]102 ページ
- ULCommand.ExecuteNonQuery メソッド [Ultra Light.NET]87 ページ
- ULCommand.ExecuteReader メソッド [Ultra Light.NET]87 ページ
- ULCommand.ExecuteResultSet メソッド [Ultra Light.NET]90 ページ
- ULCommand.ExecuteScalar メソッド [Ultra Light.NET]93 ページ
- ULCommand.ExecuteTable メソッド [Ultra Light.NET]94 ページ
- ULTable クラス [Ultra Light.NET]415 ページ
- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand.CommandType プロパティ [Ultra Light.NET]99 ページ
- ULCommand.IndexName プロパティ [Ultra Light.NET]101 ページ
- ULResultSet クラス [Ultra Light.NET]352 ページ
- ULDataReader クラス [Ultra Light.NET]236 ページ
- System.Data.Common.DbCommand
- System.Data.IDbCommand
- System.IDisposable
- System.Data.CommandType
- System.ComponentModel.Component.Dispose

ULCommand コンストラクター

ULCommand オブジェクトを初期化します。

オーバーロードリスト

名前	説明
ULCommand() コンストラクター	ULCommand オブジェクトを初期化します。
ULCommand(string) コンストラクター	ULCommand オブジェクトを、指定されたコマンドテキストで初期化します。
ULCommand(string, ULConnection) コンストラクター	ULCommand オブジェクトを、指定されたコマンドテキストと接続で初期化します。
ULCommand(string, ULConnection, ULTransaction) コンストラクター	ULCommand オブジェクトを、指定されたコマンドテキスト、接続、トランザクションで初期化します。

ULCommand() コンストラクター

ULCommand オブジェクトを初期化します。

Visual Basic 構文

```
Public Sub New()
```

C# 構文

```
public ULCommand()
```

備考

ULCommand オブジェクトに ULCommand.CommandText、ULCommand.Connection、ULCommand.Transaction のプロパティが設定されていないと、文を実行できません。

参照

- ULConnection.CreateCommand メソッド [Ultra Light.NET]130 ページ
- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand.Connection プロパティ [Ultra Light.NET]100 ページ
- ULCommand.Transaction プロパティ [Ultra Light.NET]102 ページ

ULCommand(string) コンストラクター

ULCommand オブジェクトを、指定されたコマンドテキストで初期化します。

Visual Basic 構文

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

C# 構文

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


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

パラメーター

- **cmdText** ULCommand.CommandType プロパティが System.Data.CommandType.TableDirect である場合の、SQL 文のテキストまたはテーブルの名前。パラメーター化された文の場合、疑問符 (?) プレースホルダを使用してパラメーターを渡します。

備考

ULCommand オブジェクトに ULCommand.Connection プロパティと ULCommand.Transaction プロパティが設定されていないと、文を実行できません。

参照

- ULConnection.CreateCommand メソッド [Ultra Light.NET]130 ページ
- ULCommand.ULCommand コンストラクター [Ultra Light.NET]69 ページ
- ULCommand.Connection プロパティ [Ultra Light.NET]100 ページ
- ULCommand.Transaction プロパティ [Ultra Light.NET]102 ページ
- ULCommand.CommandType プロパティ [Ultra Light.NET]99 ページ
- System.Data.CommandType

ULCommand(string, ULConnection) コンストラクター

ULCommand オブジェクトを、指定されたコマンドテキストと接続で初期化します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **cmdText** ULCommand.CommandType プロパティが System.Data.CommandType.TableDirect である場合の、SQL 文のテキストまたはテーブルの名前。パラメーター化された文の場合、疑問符 (?) プレースホルダを使用してパラメーターを渡します。
- **connection** 現在の接続を表す ULConnection オブジェクト。

備考

ULCommand オブジェクトに ULCommand.Transaction プロパティが設定されていないと、文を実行できないことがあります。

参照

- ULConnection クラス [Ultra Light.NET]116 ページ
- ULConnection.CreateCommand メソッド [Ultra Light.NET]130 ページ
- ULCommand.ULCommand コンストラクター [Ultra Light.NET]69 ページ
- ULCommand.Transaction プロパティ [Ultra Light.NET]102 ページ
- ULCommand.CommandType プロパティ [Ultra Light.NET]99 ページ
- System.Data.CommandType

ULCommand(string, ULConnection, ULTransaction) コンストラクター

ULCommand オブジェクトを、指定されたコマンドテキスト、接続、トランザクションで初期化します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **cmdText**　ULCommand.CommandType プロパティが System.Data.CommandType.TableDirect である場合の、SQL 文のテキストまたはテーブルの名前。パラメーター化された文の場合、疑問符 (?) プレースホルダを使用してパラメーターを渡します。
- **接続**　現在の接続を表す ULConnection オブジェクト。
- **transaction**　ULCommand オブジェクトが実行される ULTransaction オブジェクト。

参照

- ULConnection.CreateCommand メソッド [Ultra Light.NET]130 ページ
- ULCommand.ULCommand コンストラクター [Ultra Light.NET]69 ページ
- ULCommand.CommandType プロパティ [Ultra Light.NET]99 ページ
- ULTransaction クラス [Ultra Light.NET]451 ページ
- System.Data.CommandType

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

BeginExecuteNonQuery メソッド

この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始します。

オーバーロードリスト

名前	説明
BeginExecuteNonQuery() メソッド	この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始します。
BeginExecuteNonQuery(AsyncCallback, object) メソッド	コールバックプロシージャーと状態情報を指定して、この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始します。

BeginExecuteNonQuery() メソッド

この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始します。

Visual Basic 構文

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

C# 構文

```
public IAsyncResult BeginExecuteNonQuery()
```

戻り値

ポーリング、結果の待機、または両方に使用できる System.IAsyncResult が返されます。この値は、影響を受けたローの数を返す EndExecuteNonQuery(IAsyncResult) メソッドを起動する場合にも必要です。

例外

- **ULException クラス**　コマンドテキストの実行中に発生したエラー。

参照

- ULCommand.EndExecuteNonQuery メソッド [Ultra Light.NET]80 ページ
- System.IAsyncResult

BeginExecuteNonQuery(AsyncCallback, object) メソッド

コールバックプロシージャーと状態情報を指定して、この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始します。

Visual Basic 構文

```
Public Function BeginExecuteNonQuery(
     ByVal callback As AsyncCallback,
```

```
    ByVal stateObject As Object
) As IAsyncResult
```

C# 構文

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

パラメーター

- **callback**　コマンドの実行が終了すると起動される System.AsyncCallback デリゲート。コールバックが必要ないことを示すには、NULL (Microsoft Visual Basic の場合は Nothing) を渡します。
- **stateObject**　コールバックプロシージャーに渡される、ユーザー定義のステータスオブジェクト。コールバックプロシージャーからこのオブジェクトを取得するには、System.IAsyncResult.AsyncState プロパティを使用します。

戻り値

ポーリング、結果の待機、または両方に使用できる System.IAsyncResult が返されます。この値は、影響を受けたローの数を返す EndExecuteNonQuery(IAsyncResult) メソッドを起動する場合にも必要です。

例外

- **ULException クラス**　コマンドテキストの実行中に発生したエラー。

参照

- ULCommand.EndExecuteNonQuery メソッド [Ultra Light.NET]80 ページ
- System.IAsyncResult.AsyncState
- System.AsyncCallback
- System.IAsyncResult

BeginExecuteReader メソッド

この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始し、結果セットを取得します。

オーバーロードリスト

名前	説明
BeginExecuteReader() メソッド	この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始し、結果セットを取得します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
BeginExecuteReader(AsyncCallback, object) メソッド	コールバックプロシージャーと状態情報を指定して、この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始し、結果セットを取得します。
BeginExecuteReader(AsyncCallback, object, CommandBehavior) メソッド	コールバックプロシージャーと状態情報を指定して、CommandBehavior 値の 1 つを使用してこの ULCommand オブジェクトで記述される SQL 文の非同期実行を開始し、結果セットを取得します。
BeginExecuteReader(CommandBehavior) メソッド	CommandBehavior 値の 1 つを使用してこの ULCommand オブジェクトで記述される SQL 文の非同期実行を開始し、結果セットを取得します。

BeginExecuteReader() メソッド

この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始し、結果セットを取得します。

Visual Basic 構文

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

C# 構文

```
public IAsyncResult BeginExecuteReader()
```

戻り値

ポーリング、結果の待機、または両方に使用できる System.IAsyncResult が返されます。この値は、返されたローを取得するために使用する ULDataReader オブジェクトを返す、EndExecuteReader(IAsyncResult) メソッドを起動する場合にも必要です。

例外

- **ULException クラス**　コマンドテキストの実行中に発生したエラー。

参照

- ULCommand.EndExecuteReader メソッド [Ultra Light.NET]83 ページ
- ULDataReader クラス [Ultra Light.NET]236 ページ
- System.IAsyncResult

BeginExecuteReader(AsyncCallback, object) メソッド

コールバックプロシージャーと状態情報を指定して、この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始し、結果セットを取得します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **callback**　コマンドの実行が終了すると起動される System.AsyncCallback デリゲート。コールバックが必要ないことを示すには、NULL (Microsoft Visual Basic の場合は Nothing) を渡します。
- **stateObject**　コールバックプロシージャーに渡される、ユーザー定義のステータスオブジェクト。コールバックプロシージャーからこのオブジェクトを取得するには、System.IAsyncResult.AsyncState プロパティを使用します。

戻り値

ポーリング、結果の待機、または両方に使用できる System.IAsyncResult が返されます。この値は、返されたローを取得するために使用する ULDataReader オブジェクトを返す、EndExecuteReader(IAsyncResult) メソッドを起動する場合にも必要です。

例外

- **ULException クラス**　コマンドテキストの実行中に発生したエラー。

参照

- ULDataReader クラス [Ultra Light.NET]236 ページ
- ULCommand.EndExecuteReader メソッド [Ultra Light.NET]83 ページ
- System.AsyncCallback
- System.IAsyncResult.AsyncState
- System.IAsyncResult.AsyncState
- System.IAsyncResult

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

BeginExecuteReader(AsyncCallback, object, CommandBehavior) メソッド

コールバックプロシージャーと状態情報を指定して、CommandBehavior 値の 1 つを使用してこの ULCommand オブジェクトで記述される SQL 文の非同期実行を開始し、結果セットを取得します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **callback**　コマンドの実行が終了すると起動される System.AsyncCallback デリゲート。コールバックが必要ないことを示すには、NULL (Microsoft Visual Basic の場合は Nothing) を渡します。
- **stateObject**　コールバックプロシージャーに渡される、ユーザー定義のステータスオブジェクト。コールバックプロシージャーからこのオブジェクトを取得するには、System.IAsyncResult.AsyncState プロパティを使用します。
- **cmdBehavior**　クエリの結果の記述と、接続への影響の記述の System.Data.CommandBehavior フラグのビット単位の組み合わせ。Ultra Light.NET で有効なのは、System.Data.CommandBehavior.Default フラグ、System.Data.CommandBehavior.CloseConnection フラグ、System.Data.CommandBehavior.SchemaOnly フラグだけです。

戻り値

ポーリング、結果の待機、または両方に使用できる System.IAsyncResult が返されます。この値は、返されたローを取得するために使用する ULDataReader オブジェクトを返す、EndExecuteReader(IAsyncResult) メソッドを起動する場合にも必要です。

例外

- **ULException クラス**　コマンドテキストの実行中に発生したエラー。

参照

- ULCommand.EndExecuteReader メソッド [Ultra Light.NET]83 ページ
- ULDataReader クラス [Ultra Light.NET]236 ページ
- System.AsyncCallback
- System.IAsyncResult
- System.IAsyncResult.AsyncState
- System.Data.CommandBehavior
- System.Data.CommandBehavior.Default
- System.Data.CommandBehavior.CloseConnection
- System.Data.CommandBehavior.SchemaOnly

BeginExecuteReader(CommandBehavior) メソッド

CommandBehavior 値の 1 つを使用してこの ULCommand オブジェクトで記述される SQL 文の非同期実行を開始し、結果セットを取得します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **cmdBehavior**　クエリの結果の記述と、接続への影響の記述の System.Data.CommandBehavior フラグのビット単位の組み合わせ。Ultra Light.NET で有効なのは、System.Data.CommandBehavior.Default フラグ、System.Data.CommandBehavior.CloseConnection フラグ、System.Data.CommandBehavior.SchemaOnly フラグだけです。

戻り値

ポーリング、結果の待機、または両方に使用できる System.IAsyncResult が返されます。この値は、返されたローを取得するために使用する ULDataReader オブジェクトを返す、EndExecuteReader(IAsyncResult) メソッドを起動する場合にも必要です。

例外

- **ULException クラス**　コマンドテキストの実行中に発生したエラー。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULCommand.EndExecuteReader メソッド [Ultra Light.NET]83 ページ
- ULDataReader クラス [Ultra Light.NET]236 ページ
- System.Data.CommandBehavior
- System.Data.CommandBehavior.Default
- System.Data.CommandBehavior.CloseConnection
- System.Data.CommandBehavior.SchemaOnly
- System.IAsyncResult

Cancel メソッド

このメソッドは Ultra Light.NET ではサポートされていません。

Visual Basic 構文

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

C# 構文

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

備考

このメソッドによって行われる処理はありません。Ultra Light.NET コマンドは、実行中に中断できません。

CreateParameter メソッド

ULCommand オブジェクトにパラメーターを指定するために ULParameter オブジェクトを提供します。

Visual Basic 構文

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

C# 構文

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

戻り値

ULParameter オブジェクトとして返される新しいパラメーター。

備考

一部の SQL 文では、疑問符 (?) によって文のテキストに示されているパラメーターを使用できます。CreateParameter メソッドは、ULParameter オブジェクトを提供します。ULParameter オブジェクトにプロパティを設定して、そのパラメーターの値を指定できます。

これは、System.Data.IDbCommand.CreateParameter と System.Data.Common.DbCommand.CreateParameter が厳密に型指定されたものです。

参照

- ULParameter クラス [Ultra Light.NET]319 ページ
- System.Data.IDbCommand.CreateParameter
- System.Data.Common.DbCommand.CreateParameter

EndExecuteNonQuery メソッド

SQL 文の非同期実行を終了します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **asyncResult** BeginExecuteNonQuery メソッドへの呼び出しによって返される System.IAsyncResult。

戻り値

影響を受けるローの数。これは、ExecuteNonQuery メソッドと同じ動作です。

例外

- **ArgumentException** asyncResult パラメーターが null (Microsoft Visual Basic の Nothing) です。
- **InvalidOperationException** 1 回のコマンド実行に対して EndExecuteNonQuery(IAsyncResult) メソッドが複数回呼び出されたか、このメソッドがその実行メソッドと一致しませんでした。

備考

BeginExecuteNonQuery を呼び出すたびに、EndExecuteNonQuery メソッドを呼び出す必要があります。呼び出しは、BeginExecuteNonQuery 呼び出しが返された後に行う必要があります。ADO.NET はスレッドに対応していないため、BeginExecuteNonQuery 呼び出しが返されたことを確認する必要があります。EndExecuteNonQuery メソッドに渡される System.IAsyncResult は、完了した BeginExecuteNonQuery 呼び出しから返されるものと同じでなければなりません。EndExecuteNonQuery メソッドを呼び出して、BeginExecuteReader メソッドへの呼び出しを終了すると、エラーになります。その逆も同様です。

コマンドの実行中にエラーが発生すると、EndExecuteNonQuery メソッドが呼び出されるときに例外がスローされます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

実行の完了を待機するには、4 通りの方法があります。

「EndExecuteNonQuery を呼び出す。」EndExecuteNonQuery を呼び出すと、コマンドが完了するまでブロックします。次に例を示します。

```
' 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 )
```

対応する C# 言語のコードを次に示します。

```
// 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 );
```

「IAsyncResult の IsCompleted プロパティをポーリングする。」IAsyncResult の IsCompleted プロパティをポーリングできます。次に例を示します。

```
' 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 )
```

対応する C# 言語のコードを次に示します。

```
// 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 );
```

「IAsyncResult.AsyncWaitHandle プロパティを使用して同期オブジェクトを取得する。」
IAsyncResult.AsyncWaitHandle プロパティを使用して同期オブジェクトを取得し、その状態で待機できます。次に例を示します。

```
' 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 )
```

対応する C# 言語のコードを次に示します。

```
// 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 );
```

「BeginExecuteNonQuery メソッドの呼び出し時にコールバック関数を指定する。」
BeginExecuteNonQuery メソッドの呼び出し時にコールバック関数を指定できます。次に例を示します。

```
' Visual Basic
Private Sub callbackFunction(ByVal ar As IAsyncResult)
    Dim cmd As ULCommand = _
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
        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
```

対応する C# 言語のコードを次に示します。

```
// 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.
}
```

コールバック関数は別のスレッドで実行するため、スレッド化されたプログラム内でのユーザーインターフェイスの更新に関する通常の注意が適用されます。

参照

- ULCommand.BeginExecuteNonQuery メソッド [Ultra Light.NET]73 ページ
- System.IAsyncResult

EndExecuteReader メソッド

SQL 文の非同期実行を終了し、要求された ULDataReader を返します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **asyncResult** BeginExecuteReader 呼び出しによって返される System.IAsyncResult。

戻り値

要求されたローの取り出しに使用する ULDataReader オブジェクト (ExecuteReader メソッドと同じ動作)。

例外

- **ArgumentException** asyncResult パラメーターが null (Microsoft Visual Basic の Nothing) です。
- **InvalidOperationException** 1 回のコマンド実行に対して EndExecuteReader メソッドが複数回呼び出されたか、このメソッドがその実行メソッドに一致しませんでした。

備考

BeginExecuteReader メソッドへのそれぞれの呼び出しに対して、EndExecuteReader メソッドを 1 度呼び出す必要があります。この呼び出しは、BeginExecuteReader 呼び出しが返された後に行う必要があります。ADO.NET はスレッドに対応していないため、各自で BeginExecuteReader メソッドが返されたことを確認する必要があります。EndExecuteReader メソッドに渡される System.IAsyncResult は、完了した BeginExecuteReader 呼び出しから返されるものと同じでなければなりません。EndExecuteReader メソッドを呼び出して、BeginExecuteNonQuery への呼び出しを終了すると、エラーになります。逆についても同様です。

コマンドの実行中にエラーが発生すると、EndExecuteReader メソッドが呼び出されるときに例外がスローされます。

実行の完了を待機するには、4 通りの方法があります。

Call the EndExecuteReader method EndExecuteReader メソッドを呼び出すと、コマンドが完了するまでブロックします。次に例を示します。

```
' 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 )
```

対応する C# 言語のコードを次に示します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
// 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 );
```

「IAsyncResult の IsCompleted プロパティをポーリングする。」IAsyncResult の IsCompleted プロパティをポーリングできます。次に例を示します。

```
' 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 );
```

「IAsyncResult.AsyncWaitHandle プロパティを使用して同期オブジェクトを取得する。」IAsyncResult.AsyncWaitHandle プロパティを使用して同期オブジェクトを取得し、その状態で待機できます。次に例を示します。

```
' 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 )
```

対応する C# 言語のコードを次に示します。

```
// 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 BeginExecuteReader メソッドの呼び出し時にコールバック関数を指定できます。次に例を示します。

```
' 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
```

対応する C# 言語のコードを次に示します。

```
// 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.
}
```

コールバック関数は別のスレッドで実行するため、スレッド化されたプログラム内でのユーザーインターフェイスの更新に関する通常の注意が適用されます。

参照

- ULCommand.BeginExecuteReader メソッド [Ultra Light.NET]74 ページ
- ULDataReader クラス [Ultra Light.NET]236 ページ
- System.IAsyncResult

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

ExecuteNonQuery メソッド

SQL INSERT 文、DELETE 文、UPDATE 文のように、結果セットを返さない文を実行します。

Visual Basic 構文

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

C# 構文

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

戻り値

影響を受けたローの数。

例外

- **ULException クラス**　SQL エラーが発生しました。
- **InvalidOperationException**　コマンドのステータスは有効ではありません。ULCommand.Connection オブジェクトが見つからないか閉じている、ULCommand.Transaction 値が接続の現在のトランザクションのステータスに一致しない、または ULCommand.CommandText 値が無効です。

備考

この文は、必要に応じて ULCommand.CommandText 値と ULCommand.Parameters 値が指定された、現在の ULCommand オブジェクトです。

UPDATE、INSERT、DELETE 文の場合、戻り値はコマンドの影響を受けるローの数です。その他すべての文のタイプとロールバックの場合、戻り値は -1 です。

ULCommand.CommandType プロパティを System.Data.CommandType.TableDirect にすることはできません。

参照

- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand.CommandType プロパティ [Ultra Light.NET]99 ページ
- ULCommand.Connection プロパティ [Ultra Light.NET]100 ページ
- ULCommand.Parameters プロパティ [Ultra Light.NET]101 ページ
- ULCommand.Transaction プロパティ [Ultra Light.NET]102 ページ
- System.Data.CommandType

ExecuteReader メソッド

SQL SELECT 文を実行し、結果セットを返します。

オーバーロードリスト

名前	説明
ExecuteReader() メソッド	SQL SELECT 文を実行し、結果セットを返します。
ExecuteReader(CommandBehavior) メソッド	コマンドの動作を指定して SQL SELECT 文を実行し、結果セットを返します。

ExecuteReader() メソッド

SQL SELECT 文を実行し、結果セットを返します。

Visual Basic 構文

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

C# 構文

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

戻り値

ULDataReader オブジェクトとして返される結果セット。

例外

- **ULException クラス**　SQL エラーが発生しました。
- **InvalidOperationException**　コマンドのステータスは有効ではありません。ULCommand.Connection 値が見つからないか閉じている、ULCommand.Transaction 値が接続の現在のトランザクションのステータスに一致しない、または ULCommand.CommandText 値が無効です。

備考

この文は、必要に応じて ULCommand.CommandText 値と ULCommand.Parameters 値が指定された、現在の ULCommand オブジェクトです。ULDataReader オブジェクトは、読み込み専用の結果セットです。編集可能な結果セットには、ULCommand.ExecuteResultSet メソッド、ULCommand.ExecuteTable メソッド、または ULDataAdapter を使用します。

ULCommand.CommandType 値が System.Data.CommandType.TableDirect の場合は、ExecuteReader メソッドが ULCommand.ExecuteTable 呼び出しを時実行して、ULTable オブジェクト ダウンキャストを ULDataReader オブジェクトとして返します。

SELECT 文は、パフォーマンス上の理由により、デフォルトで読み込み専用とマーク付けされます。クエリを使用して更新を行う場合、SELECT 文は "FOR UPDATE" で終わる必要があります。

これは、System.Data.IDbCommand.ExecuteReader メソッドと System.Data.Common.DbCommand.ExecuteReader メソッドが厳密に型指定されたものです。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand.CommandType プロパティ [Ultra Light.NET]99 ページ
- ULCommand.Connection プロパティ [Ultra Light.NET]100 ページ
- ULCommand.Parameters プロパティ [Ultra Light.NET]101 ページ
- ULCommand.ExecuteResultSet メソッド [Ultra Light.NET]90 ページ
- ULCommand.ExecuteTable メソッド [Ultra Light.NET]94 ページ
- ULCommand.Transaction プロパティ [Ultra Light.NET]102 ページ
- ULDataAdapter クラス [Ultra Light.NET]209 ページ
- ULDataReader クラス [Ultra Light.NET]236 ページ
- ULTable クラス [Ultra Light.NET]415 ページ
- System.Data.CommandType
- System.Data.IDbCommand.ExecuteReader
- System.Data.Common.DbCommand.ExecuteReader

ExecuteReader(CommandBehavior) メソッド

コマンドの動作を指定して SQL SELECT 文を実行し、結果セットを返します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **cmdBehavior**　クエリの結果の記述と、接続への影響の記述の System.Data.CommandBehavior フラグのビット単位の組み合わせ。Ultra Light.NET で有効なのは、System.Data.CommandBehavior.Default フラグ、System.Data.CommandBehavior.CloseConnection フラグ、System.Data.CommandBehavior.SchemaOnly フラグだけです。

戻り値

ULDataReader オブジェクトとして返される結果セット。

例外

- **ULException クラス**　SQL エラーが発生しました。
- **InvalidOperationException**　コマンドのステータスは有効ではありません。ULCommand.Connection 値が見つからないか閉じている、ULCommand.Transaction 値が接続の現在のトランザクションのステータスに一致しない、または ULCommand.CommandText 値が無効です。

備考

この文は、必要に応じて ULCommand.CommandText 値と ULCommand.Parameters 値が指定された、現在の ULCommand オブジェクトです。ULDataReader オブジェクトは、読み込み専用の結果セットです。編集可能な結果セットの場合は、ULCommand.ExecuteResultSet (CommandBehavior) メソッド、ULCommand.ExecuteTable (CommandBehavior) メソッド、または ULDataAdapter オブジェクトを使用します。

ULCommand.CommandType 値が System.Data.CommandType.TableDirect の場合は、ExecuteReader メソッドが ULCommand.ExecuteTable (CommandBehavior) 呼び出しを時実行して、ULTable オブジェクト ダウンキャストを ULDataReader オブジェクトとして返します。

SELECT 文は、パフォーマンス上の理由により、デフォルトで読み込み専用とマーク付けされます。クエリを使用して更新を行う場合、SELECT 文は "FOR UPDATE" で終わる必要があります。

これは、System.Data.IDbCommand.ExecuteReader(System.Data.CommandBehavior) と System.Data.Common.DbCommand.ExecuteReader(System.Data.CommandBehavior) が厳密に型指定されたものです。

参照

- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand.Connection プロパティ [Ultra Light.NET]100 ページ
- ULCommand.ExecuteReader メソッド [Ultra Light.NET]87 ページ
- ULCommand.ExecuteResultSet メソッド [Ultra Light.NET]90 ページ
- ULCommand.ExecuteTable メソッド [Ultra Light.NET]94 ページ
- ULCommand.Parameters プロパティ [Ultra Light.NET]101 ページ
- ULCommand.Transaction プロパティ [Ultra Light.NET]102 ページ
- ULDataAdapter クラス [Ultra Light.NET]209 ページ
- ULDataReader クラス [Ultra Light.NET]236 ページ
- ULTable クラス [Ultra Light.NET]415 ページ
- ULDataReader クラス [Ultra Light.NET]236 ページ
- System.Data.IDbCommand.ExecuteReader
- System.Data.Common.DbCommand.ExecuteReader
- System.Data.CommandType
- System.Data.CommandBehavior
- System.Data.CommandBehavior.Default
- System.Data.CommandBehavior.CloseConnection
- System.Data.CommandBehavior.SchemaOnly

ExecuteResultSet メソッド

UL 拡張： SQL SELECT 文を実行し、ULResultSet オブジェクトとして結果セットを返します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

オーバーロードリスト

名前	説明
ExecuteResultSet() メソッド	**UL 拡張：** SQL SELECT 文を実行し、ULResultSet オブジェクトとして結果セットを返します。
ExecuteResultSet(CommandBehavior) メソッド	**UL 拡張：** コマンドの動作を指定して SQL SELECT 文を実行し、ULResultSet オブジェクトとして結果セットを返します。

ExecuteResultSet() メソッド

UL 拡張： SQL SELECT 文を実行し、ULResultSet オブジェクトとして結果セットを返します。

Visual Basic 構文

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

C# 構文

```
public ULResultSet ExecuteResultSet()
```

戻り値

ULResultSet オブジェクトとして返される結果セット。

例外

- **ULException クラス**　SQL エラーが発生しました。
- **InvalidOperationException**　コマンドのステータスは有効ではありません。ULCommand.Connection 値が見つからないか閉じている、ULCommand.Transaction 値が接続の現在のトランザクションのステータスに一致しない、または ULCommand.CommandText 値が無効です。

備考

この文は、必要に応じて ULCommand.CommandText 値と ULCommand.Parameters 値が指定された、現在の ULCommand オブジェクトです。ULResultSet オブジェクトは、位置付け更新や削除の実行対象となる編集可能な結果を表します。編集可能な結果セットには、ULCommand.ExecuteTable メソッドまたは ULDataAdapter オブジェクトを使用します。

ULCommand.CommandType 値が System.Data.CommandType.TableDirect の場合は、ExecuteReader メソッドが ULCommand.ExecuteTable 呼び出しを時実行して、ULTable オブジェクト ダウンキャストを ULResultSet オブジェクトとして返します。

このメソッドは、Dynamic SQL による位置付け更新と削除をサポートします。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand.Parameters プロパティ [Ultra Light.NET]101 ページ
- ULCommand.CommandType プロパティ [Ultra Light.NET]99 ページ
- ULResultSet クラス [Ultra Light.NET]352 ページ
- ULTable クラス [Ultra Light.NET]415 ページ
- System.Data.CommandType

例

```
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) メソッド

UL 拡張： コマンドの動作を指定して SQL SELECT 文を実行し、ULResultSet オブジェクトとして結果セットを返します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **cmdBehavior**　クエリの結果の記述と、接続への影響の記述の System.Data.CommandBehavior フラグのビット単位の組み合わせ。Ultra Light.NET で有効なのは、System.Data.CommandBehavior.Default フラグ、System.Data.CommandBehavior.CloseConnection フラグ、System.Data.CommandBehavior.SchemaOnly フラグだけです。

戻り値

ULResultSet オブジェクトとして返される結果セット。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

例外

- **ULException クラス**　SQL エラーが発生しました。
- **InvalidOperationException**　コマンドのステータスは有効ではありません。ULCommand.Connection 値が見つからないか閉じている、ULCommand.Transaction 値が接続の現在のトランザクションのステータスに一致しない、または ULCommand.CommandText 値が無効です。

備考

この文は、必要に応じて ULCommand.CommandText 値と任意の ULCommand.Parameters 値が指定された、現在の ULCommand オブジェクトです。ULResultSet オブジェクトは、位置付け更新や削除の実行対象となる編集可能な結果を表します。編集可能な結果セットには、ULCommand.ExecuteTable(CommandBehavior) メソッドまたは ULDataAdapter オブジェクトを使用します。

ULCommand.CommandType 値が System.Data.CommandType.TableDirect の場合は、ExecuteReader メソッドが ULCommand.ExecuteTable(CommandBehavior) 呼び出しを時実行して、ULTable オブジェクト ダウンキャストを ULResultSet オブジェクトとして返します。

このメソッドは、Dynamic SQL による位置付け更新と削除をサポートします。

参照

- ULResultSet クラス [Ultra Light.NET]352 ページ
- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand.CommandType プロパティ [Ultra Light.NET]99 ページ
- ULCommand.Connection プロパティ [Ultra Light.NET]100 ページ
- ULCommand.ExecuteReader メソッド [Ultra Light.NET]87 ページ
- ULCommand.ExecuteTable メソッド [Ultra Light.NET]94 ページ
- ULCommand.Parameters プロパティ [Ultra Light.NET]101 ページ
- ULCommand.Transaction プロパティ [Ultra Light.NET]102 ページ
- ULDataAdapter クラス [Ultra Light.NET]209 ページ
- ULTable クラス [Ultra Light.NET]415 ページ
- System.Data.CommandType
- System.Data.CommandBehavior
- System.Data.CommandBehavior.Default
- System.Data.CommandBehavior.CloseConnection
- System.Data.CommandBehavior.SchemaOnly

ExecuteScalar メソッド

SQL SELECT 文を実行し、単一の値を返します。

Visual Basic 構文

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

C# 構文

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

戻り値

結果セットの最初のローの最初のカラム。結果セットが空の場合は NULL 参照 (Visual Basic の Nothing)。

例外

- **ULException クラス**　SQL エラーが発生しました。
- **InvalidOperationException**　コマンドのステータスは有効ではありません。ULCommand.Connection 値が見つからないか閉じている、ULCommand.Transaction 値が接続の現在のトランザクションのステータスに一致しない、または ULCommand.CommandText 値が無効です。

備考

この文は、必要に応じて ULCommand.CommandText 値と任意の ULCommand.Parameters 値が指定された、現在の ULCommand オブジェクトです。

複数のローとカラムを返すクエリでこのメソッドが呼び出されると、最初のローの最初のカラムのみが返されます。

ULCommand.CommandType 値が System.Data.CommandType.TableDirect である場合、ExecuteScalar メソッドは ULCommand.ExecuteTable 呼び出しを実行し、最初のローの最初のカラムを返します。

SELECT 文は、パフォーマンス上の理由により、デフォルトで読み込み専用とマーク付けされます。クエリを使用して更新を行う場合、SELECT 文は "FOR UPDATE" で終わる必要があります。

参照

- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand.CommandType プロパティ [Ultra Light.NET]99 ページ
- ULCommand.Connection プロパティ [Ultra Light.NET]100 ページ
- ULCommand.IndexName プロパティ [Ultra Light.NET]101 ページ
- ULCommand.Parameters プロパティ [Ultra Light.NET]101 ページ
- ULCommand.Transaction プロパティ [Ultra Light.NET]102 ページ
- System.Data.CommandType
- System.Data.CommandBehavior
- System.Data.CommandBehavior.Default
- System.Data.CommandBehavior.CloseConnection
- System.Data.CommandBehavior.SchemaOnly

ExecuteTable メソッド

UL 拡張： 直接操作するために、ULTable オブジェクトのデータベーステーブルを取り出します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

オーバーロードリスト

名前	説明
ExecuteTable() メソッド	**UL 拡張：** 直接の操作用に、ULTable オブジェクトのデータベーステーブルを取り出します。
ExecuteTable(CommandBehavior) メソッド	**UL 拡張：** 直接の操作用に、コマンド動作を指定してデータベーステーブルを取り出します。

ExecuteTable() メソッド

UL 拡張： 直接の操作用に、ULTable オブジェクトのデータベーステーブルを取り出します。

Visual Basic 構文

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

C# 構文

```
public ULTable ExecuteTable()
```

戻り値

ULTable オブジェクトとしてのテーブル。

例外

- **ULException クラス**　SQL エラーが発生しました。
- **InvalidOperationException**　コマンドのステータスは有効ではありません。ULCommand.Connection 値が見つからないか閉じている、ULCommand.Transaction 値が接続の現在のトランザクションのステータスに一致しない、または ULCommand.CommandText 値が無効です。

備考

ULCommand.CommandText 値はテーブルの名前として解釈され、ULCommand.IndexName 値はテーブルのソート順の指定に使用できます。

ULCommand.CommandType 値は System.Data.CommandType.TableDirect に設定する必要があります。

ULCommand.IndexName 値が NULL 参照 (Visual Basic の Nothing) である場合は、プライマリキーを使用してテーブルが開かれます。そうでない場合は、ソート基準となるインデックスの名前として ULCommand.IndexName 値を使用してテーブルが開かれます。

参照

- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand.CommandType プロパティ [Ultra Light.NET]99 ページ
- ULCommand.Connection プロパティ [Ultra Light.NET]100 ページ
- ULCommand.IndexName プロパティ [Ultra Light.NET]101 ページ
- ULCommand.Transaction プロパティ [Ultra Light.NET]102 ページ
- ULTable クラス [Ultra Light.NET]415 ページ
- System.Data.CommandBehavior
- System.Data.CommandBehavior.Default
- System.Data.CommandBehavior.CloseConnection
- System.Data.CommandBehavior.SchemaOnly
- System.Data.CommandType

ExecuteTable(CommandBehavior) メソッド

UL 拡張： 直接の操作用に、コマンド動作を指定してデータベーステーブルを取り出します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **cmdBehavior** クエリの結果の記述と、接続への影響の記述の System.Data.CommandBehavior フラグのビット単位の組み合わせ。Ultra Light.NET で有効なのは、System.Data.CommandBehavior.Default フラグ、System.Data.CommandBehavior.CloseConnection フラグ、System.Data.CommandBehavior.SchemaOnly フラグだけです。

戻り値

ULTable オブジェクトとしてのテーブル。

例外

- **ULException クラス** SQL エラーが発生しました。
- **InvalidOperationException** コマンドのステータスは有効ではありません。ULCommand.Connection 値が見つからないか閉じている、ULCommand.Transaction 値が接続の現在のトランザクションのステータスに一致しない、または ULCommand.CommandText 値が無効です。

備考

ULCommand.CommandText 値はテーブルの名前として解釈され、ULCommand.IndexName 値はテーブルのソート順の指定に使用できます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

ULCommand.CommandType 値は System.Data.CommandType.TableDirect に設定する必要があります。

ULCommand.IndexName 値が NULL 参照 (Visual Basic の Nothing) である場合は、プライマリキーを使用してテーブルが開かれます。そうでない場合は、ソート基準となるインデックスの名前として ULCommand.IndexName 値を使用してテーブルが開かれます。

参照

- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand.CommandType プロパティ [Ultra Light.NET]99 ページ
- ULCommand.Connection プロパティ [Ultra Light.NET]100 ページ
- ULCommand.ExecuteTable メソッド [Ultra Light.NET]94 ページ
- ULCommand.IndexName プロパティ [Ultra Light.NET]101 ページ
- ULCommand.Transaction プロパティ [Ultra Light.NET]102 ページ
- System.Data.CommandBehavior
- System.Data.CommandBehavior.Default
- System.Data.CommandBehavior.CloseConnection
- System.Data.CommandBehavior.SchemaOnly
- System.Data.CommandType

Prepare メソッド

このコマンドの SQL 文を事前にコンパイルして格納します。

Visual Basic 構文

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

C# 構文

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

例外

- **ULException クラス**　SQL エラーが発生しました。
- **InvalidOperationException**　コマンドのステータスは有効ではありません。ULCommand.Connection 値が見つからないか閉じている、ULCommand.Transaction 値が接続の現在のトランザクションのステータスに一致しない、または ULCommand.CommandText 値が無効です。

備考

文を事前にコンパイルすると、パラメーター値のみが変更されているときに文を効率的に再使用できます。このコマンドの他のプロパティを変更すると、文の準備は解除されます。

準備が解除されたコマンドはいずれも、さまざまな Execute メソッドへの呼び出し時に準備が行われるため、Ultra Light.NET では明示的に文を準備する必要はありません。

参照

- ULCommand.Connection プロパティ [Ultra Light.NET]100 ページ
- ULCommand.Transaction プロパティ [Ultra Light.NET]102 ページ
- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ

CommandText プロパティ

ULCommand.CommandType プロパティが System.Data.CommandType.TableDirect である場合の、SQL 文のテキストまたはテーブルの名前を指定します。

Visual Basic 構文

```
Public Overrides Property CommandText As String
```

C# 構文

```
public override string CommandText {get;set;}
```

備考

パラメーター化された文の場合、疑問符 (?) プレースホルダーを使用してパラメーターを渡します。

SQL 文のテキストまたはテーブルの名前を指定する文字列。デフォルトは、空の文字列 (無効なコマンド) です。

SELECT 文は、パフォーマンス上の理由により、デフォルトで読み込み専用とマーク付けされます。クエリを使用して更新を行う場合、SELECT 文は "FOR UPDATE" で終わる必要があります。

参照

- ULCommand.ExecuteNonQuery メソッド [Ultra Light.NET]87 ページ
- ULCommand.ExecuteReader メソッド [Ultra Light.NET]87 ページ
- ULCommand.ExecuteResultSet メソッド [Ultra Light.NET]90 ページ
- ULCommand.ExecuteScalar メソッド [Ultra Light.NET]93 ページ
- ULCommand.ExecuteTable メソッド [Ultra Light.NET]94 ページ
- ULCommand.CommandType プロパティ [Ultra Light.NET]99 ページ
- System.Data.CommandType

例

' Visual Basic myCmd.CommandText = "SELECT * FROM Customers WHERE CustomerID = ?"

対応する C# 言語のコードを次に示します。

```
// C#
myCmd.CommandText = "SELECT * FROM Customers WHERE CustomerID = ?";
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

CommandTimeout プロパティ

この機能は Ultra Light.NET ではサポートされていません。

Visual Basic 構文

```
Public Overrides Property CommandTimeout As Integer
```

C# 構文

```
public override int CommandTimeout {get;set;}
```

例外

- **ULException クラス**　値を設定することは、Ultra Light.NET ではサポートされていません。

備考

値は常に 0 です。

CommandType プロパティ

実行されるコマンドのタイプを指定します。

Visual Basic 構文

```
Public Overrides Property CommandType As CommandType
```

C# 構文

```
public override CommandType CommandType {get;set;}
```

例外

- **ArgumentException**　CommandType.StoredProcedure は Ultra Light.NET ではサポートされていません。

備考

System.Data.CommandType 値の 1 つ。デフォルト値は System.Data.CommandType.Text です。

サポートされているコマンドタイプは、次のとおりです。

- System.Data.CommandType.TableDirect - **UL 拡張：** この CommandType プロパティを指定するときは、ULCommand.CommandText プロパティがデータベーステーブルの名前である必要があります。また、ULCommand.IndexName プロパティを使用して、テーブルを開く (ソートする) インデックスを指定することもできます。テーブルにアクセスするには、ULCommand.ExecuteTable メソッドまたは ULCommand.ExecuteReader メソッドを使用します。

- System.Data.CommandType.Text - この CommandType プロパティを指定するときは、ULCommand.CommandText プロパティが SQL 文またはクエリである必要があります。クエリ以外の SQL 文を実行するには、ULCommand.ExecuteNonQuery メソッドを使用します。また、

クエリを実行するには、ULCommand.ExecuteReader または ULCommand.ExecuteScalar メソッドを使用します。

参照

- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand.IndexName プロパティ [Ultra Light.NET]101 ページ
- ULCommand.ExecuteTable メソッド [Ultra Light.NET]94 ページ
- ULCommand.ExecuteReader メソッド [Ultra Light.NET]87 ページ
- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand.ExecuteNonQuery メソッド [Ultra Light.NET]87 ページ
- ULCommand.ExecuteReader メソッド [Ultra Light.NET]87 ページ
- ULCommand.ExecuteScalar メソッド [Ultra Light.NET]93 ページ
- System.Data.CommandType
- System.Data.CommandType.Text
- System.Data.CommandType

Connection プロパティ

ULCommand オブジェクトを実行する接続オブジェクトです。

Visual Basic 構文

```
Public Shadows Property Connection As ULConnection
```

C# 構文

```
public new ULConnection Connection {get;set;}
```

備考

コマンドを実行する ULConnection オブジェクト。

ULCommand オブジェクトは、開かれた接続がないと、実行できません。

デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

これは、System.Data.IDbCommand.Connection と System.Data.Common.DbCommand.Connection が厳密に型指定されたものです。

参照

- ULConnection クラス [Ultra Light.NET]116 ページ
- System.Data.IDbCommand.Connection
- System.Data.Common.DbCommand.Connection

DesignTimeVisible プロパティ

カスタマイズされた Windows Form Designer 制御で ULCommand オブジェクトを参照できるようにするかどうかを指定します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public Overrides Property DesignTimeVisible As Boolean
```

C# 構文

```
public override bool DesignTimeVisible {get;set;}
```

備考

ULCommand オブジェクトを参照できるようにする場合は true、このオブジェクトを参照できないようにする場合は false。デフォルトは false です。

IndexName プロパティ

UL 拡張： ULCommand.CommandType プロパティが System.Data.CommandType.TableDirect であるときに、テーブルを開く (ソートする) インデックスの名前を指定します。

Visual Basic 構文

```
Public Property IndexName As String
```

C# 構文

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

備考

インデックスの名前を指定する文字列。デフォルトは NULL 参照 (Visual Basic の Nothing) です。この場合、テーブルはプライマリキーを使用して開かれます。

参照

- ULCommand.ExecuteTable メソッド [Ultra Light.NET]94 ページ
- ULCommand.ExecuteReader メソッド [Ultra Light.NET]87 ページ
- ULCommand.CommandType プロパティ [Ultra Light.NET]99 ページ
- System.Data.CommandType

Parameters プロパティ

現在の文のパラメーターを指定します。

Visual Basic 構文

```
Public ReadOnly Shadows Property Parameters As ULParameterCollection
```

C# 構文

```
public new ULParameterCollection Parameters {get;}
```

備考

SQL 文のパラメーターを保持する ULParameterCollection オブジェクト。デフォルト値は空のコレクションです。

ULCommand.CommandText プロパティ値で疑問符を使用してパラメーターを示します。コレクション内のパラメーターは、疑問符のプレースホルダーと同じ順序で指定されます。たとえば、コレクション内の最初のパラメーターは、SQL 文の最初の疑問符に対応し、コレクション内の 2 番目のパラメーターは、SQL 文の 2 番目の疑問符に対応します。ULCommand.CommandText プロパティ内の疑問符の数は、少なくともコレクション内のパラメーターの数と同じでなければなりません。

これは、System.Data.IDbCommand.Parameters と System.Data.Common.DbCommand.Parameters が厳密に型指定されたものです。

参照

- ULParameterCollection クラス [Ultra Light.NET]334 ページ
- ULParameter クラス [Ultra Light.NET]319 ページ
- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- System.Data.IDbCommand.Connection
- System.Data.Common.DbCommand.Connection

Plan プロパティ

UL 拡張： Ultra Light.NET がクエリの実行に使用するアクセスプランを返します。

Visual Basic 構文

```
Public ReadOnly Property Plan As String
```

C# 構文

```
public string Plan {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

このプロパティは、主に開発中の使用を目的とします。

クエリ実行プランのテキストベースの記述が含まれる文字列。

Transaction プロパティ

ULCommand オブジェクトが実行される ULTransaction オブジェクトを指定します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public Shadows Property Transaction As ULTransaction
```

C# 構文

```
public new ULTransaction Transaction {get;set;}
```

備考

ULCommand オブジェクトが実行される ULTransaction オブジェクト。これは、ULCommand.Connection オブジェクトによって指定された接続の現在のトランザクションでなければなりません。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

トランザクションがコミットまたはロールバックされた後でコマンドを再使用する場合は、このプロパティをリセットする必要があります。

これは、System.Data.IDbCommand.Transaction と System.Data.Common.DbCommand.Transaction が厳密に型指定されたものです。

参照

- ULConnection.BeginTransaction メソッド [Ultra Light.NET]124 ページ
- ULTransaction クラス [Ultra Light.NET]451 ページ
- ULCommand.Connection プロパティ [Ultra Light.NET]100 ページ
- System.Data.IDbCommand.Transaction
- System.Data.Common.DbCommand.Transaction

UpdatedRowSource プロパティ

ULDataAdapterUpdate メソッドによって使用されるときにコマンドの結果が DataRow に適用される方法を指定します。

Visual Basic 構文

```
Public Overrides Property UpdatedRowSource As UpdateRowSource
```

C# 構文

```
public override UpdateRowSource UpdatedRowSource {get;set;}
```

備考

System.Data.UpdateRowSource 値の 1 つ。デフォルト値は System.Data.UpdateRowSource.Both です。

参照

- System.Data.UpdateRowSource

ULCommandBuilder クラス

System.Data.DataSet の変更内容を、関連するデータベース内のデータに一致させる単一テーブルのコマンドを、自動的に生成します。

Visual Basic 構文

```
Public Class ULCommandBuilder
    Inherits System.Data.Common.DbCommandBuilder
```

C# 構文

```
public class ULCommandBuilder : System.Data.Common.DbCommandBuilder
```

基本クラス

- System.Data.Common.DbCommandBuilder

メンバー

継承されたメンバーを含む ULCommandBuilder クラスのすべてのメンバー。

名前	説明
ULCommandBuilder コンストラクター	ULCommandBuilder オブジェクトを初期化します。
ApplyParameterInfo メソッド (System.Data.Common.DbCommandBuilder から継承)	System.Data.Common.DbCommandBuilder クラスのプロバイダー実装が追加のパラメーター プロパティを処理することを許可します。
Dispose メソッド (System.Data.Common.DbCommandBuilder から継承)	System.Data.Common.DbCommandBuilder によって使用されるアンマネージリソースを解放するほか、必要に応じてマネージリソースを解放します。
GetDeleteCommand メソッド	データベースで削除処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。
GetInsertCommand メソッド	データベースで挿入処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。
GetParameterName メソッド (System.Data.Common.DbCommandBuilder から継承)	指定されたパラメーターの名前を @p# のフォーマットで返します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
GetParameterPlaceholder メソッド (System.Data.Common.DbCommandBuilder から継承)	関連する SQL 文のパラメーターのプレースホルダーを返します。
GetSchemaTable メソッド (System.Data.Common.DbCommandBuilder から継承)	System.Data.Common.DbCommandBuilder のスキーマテーブルを返します。
GetUpdateCommand メソッド	データベースで更新処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。
InitializeCommand メソッド (System.Data.Common.DbCommandBuilder から継承)	System.Data.Common.DbCommand の System.Data.Common.DbCommand.CommandTimeout、System.Data.Common.DbCommand.Transaction、System.Data.Common.DbCommand.CommandType、および System.Data.UpdateRowSource のプロパティをリセットします。
QuoteIdentifier メソッド (System.Data.Common.DbCommandBuilder から継承)	カタログの大文字／小文字が正しい引用符なしの識別子が指定されると、識別子に埋め込まれる引用符を適切にエスケープして、正しい形式の引用符付き識別子を返します。
RefreshSchema メソッド (System.Data.Common.DbCommandBuilder から継承)	この System.Data.Common.DbCommandBuilder に関連付けられたコマンドをクリアします。
RowUpdatingHandler メソッド (System.Data.Common.DbCommandBuilder から継承)	System.Data.OleDb.OleDbDataAdapter.RowUpdating のイベントのイベントハンドラーを追加します。
SetRowUpdatingHandler メソッド (System.Data.Common.DbCommandBuilder から継承)	System.Data.Common.DbDataAdapter の System.Data.OleDb.OleDbDataAdapter.RowUpdating イベントを処理するために、System.Data.Common.DbCommandBuilder を登録します。
UnquoteIdentifier メソッド (System.Data.Common.DbCommandBuilder から継承)	引用符付き識別子が指定されると、識別子に埋め込まれた引用符を適切にアンエスケープして、正しい形式の引用符なしの識別子を返します。

名前	説明
CatalogLocation プロパティ (System.Data.Common.DbCommandBuilder から継承)	System.Data.Common.DbCommandBuilder クラスのインスタンス用に System.Data.Common.CatalogLocation を設定または取得します。
CatalogSeparator プロパティ (System.Data.Common.DbCommandBuilder から継承)	System.Data.Common.DbCommandBuilder クラスのインスタンスのカタログセパレーターとして使用する文字列を設定または取得します。
ConflictOption プロパティ (System.Data.Common.DbCommandBuilder から継承)	System.Data.Common.DbCommandBuilder によって使用される System.Data.ConflictOption を指定します。
DataAdapter プロパティ	SQL 文が自動的に生成される ULDataAdapter オブジェクトを取得または設定します。
QuotePrefix プロパティ (System.Data.Common.DbCommandBuilder から継承)	スペースや予約語などの文字列が名前に含まれているデータベースオブジェクト (テーブルやカラムなど) を指定するときに使用する先頭の文字または文字列を取得または設定します。
QuoteSuffix プロパティ (System.Data.Common.DbCommandBuilder から継承)	スペースや予約語などの文字列が名前に含まれているデータベースオブジェクト (テーブルやカラムなど) を指定するときに使用する先頭の文字または文字列を取得または設定します。
SchemaSeparator プロパティ (System.Data.Common.DbCommandBuilder から継承)	スキーマ識別子とその他の識別子とを区切るセパレーターに使用する文字を取得または設定します。
SetAllValues プロパティ (System.Data.Common.DbCommandBuilder から継承)	UPDATE 文のすべてのカラム値が含まれるか、変更された値のみが含まれるかを指定します。

備考

ULDataAdapter オブジェクトは、System.Data.DataSet の変更内容を関連するデータソース内のデータに一致させるために必要な SQL 文を、自動的には生成しません。ただし、ULDataAdapter オブジェクトの SelectCommand プロパティを設定すると単一テーブルを更新する SQL 文を自動的に生成するような、ULCommandBuilder オブジェクトを作成することはできます。これにより、設定していない追加の SQL 文が ULCommandBuilder オブジェクトによって生成されます。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- ULConnection クラス [Ultra Light.NET]116 ページ
- ULDataAdapter クラス [Ultra Light.NET]209 ページ
- System.Data.DataSet
- System.ComponentModel.Component
- System.IDisposable

例

次の例では、ULCommand オブジェクトを ULDataAdapter オブジェクトと ULConnection オブジェクトとともに使用して、データソースからローを選択します。この例では、接続文字列、クエリ文字列 (SQL SELECT 文)、データベーステーブル名を表す文字列を受け取り、次に ULCommandBuilder オブジェクトを作成します。

```
' 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
```

対応する C# 言語のコードを次に示します。

```
// 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 コンストラクター

ULCommandBuilder オブジェクトを初期化します。

オーバーロードリスト

名前	説明
ULCommandBuilder() コンストラクター	ULCommandBuilder オブジェクトを初期化します。
ULCommandBuilder(ULDataAdapter) コンストラクター	ULCommandBuilder オブジェクトを、指定された ULDataAdapter オブジェクトで初期化します。

ULCommandBuilder() コンストラクター

ULCommandBuilder オブジェクトを初期化します。

Visual Basic 構文

```
Public Sub New()
```

C# 構文

```
public ULCommandBuilder()
```

参照

- ULCommandBuilder.ULCommandBuilder コンストラクター [Ultra Light.NET]108 ページ

ULCommandBuilder(ULDataAdapter) コンストラクター

ULCommandBuilder オブジェクトを、指定された ULDataAdapter オブジェクトで初期化します。

Visual Basic 構文

```
Public Sub New(ByVal adapter As ULDataAdapter)
```

C# 構文

```
public ULCommandBuilder(ULDataAdapter adapter)
```

パラメーター

- **adapter** ULDataAdapter オブジェクト。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULDataAdapter クラス [Ultra Light.NET]209 ページ

GetDeleteCommand メソッド

データベースで削除処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

オーバーロードリスト

名前	説明
GetDeleteCommand() メソッド	データベースで削除処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。
GetDeleteCommand(bool) メソッド	データベースで削除処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

GetDeleteCommand() メソッド

データベースで削除処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

Visual Basic 構文

```
Public Shadows Function GetDeleteCommand() As ULCommand
```

C# 構文

```
public new ULCommand GetDeleteCommand()
```

戻り値

削除の実行に必要な、自動的に生成された ULCommand オブジェクト。

例外

- **InvalidOperationException** DbCommandBuilder.DataAdapter プロパティが初期化されていません。DataAdapter.SelectCommand プロパティが初期化されていません。DataAdapter.SelectCommand.Connection プロパティが初期化されていません。動的 SQL の生成は、複数のベーステーブルに対してはサポートされていません。動的 SQL の生成は、重複したカラムを含む SelectCommand 値に対してはサポートされていません。DeleteCommand プロパティに対する動的 SQL の生成は、キーカラム情報を返さない SelectCommand 値に対してはサポートされていません。

備考

SQL 文が最初に生成された後で、アプリケーションが ULDataAdapter.SelectCommand 値を変更する場合は、DbCommandBuilder.RefreshSchema メソッドを明示的に呼び出す必要があります。この処理を行わないと、GetDeleteCommand メソッドは、正しくない可能性がある古い文の情報を使用し続けます。SQL 文が最初に生成されるのは、アプリケーションが DbDataAdapter.Update(System.Data.DataSet) または GetDeleteCommand メソッドを呼び出したときです。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- ULDataAdapter.SelectCommand プロパティ [Ultra Light.NET]217 ページ
- System.Data.Common.DbCommandBuilder.DataAdapter
- System.Data.Common.DbCommandBuilder.RefreshSchema

GetDeleteCommand(bool) メソッド

データベースで削除処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

Visual Basic 構文

```
Public Shadows Function GetDeleteCommand(
     ByVal useColumnsForParameterNames As Boolean
) As ULCommand
```

C# 構文

```
public new ULCommand GetDeleteCommand(bool useColumnsForParameterNames)
```

パラメーター

- **useColumnsForParameterNames** true の場合、可能であれば、カラム名に一致するパラメーター名を生成します。false の場合、@p1、@p2 などを生成します。

戻り値

削除の実行に必要な、自動的に生成された ULCommand オブジェクト。

例外

- **InvalidOperationException** DbCommandBuilder.DataAdapter プロパティが初期化されていません。DataAdapter.SelectCommand プロパティが初期化されていません。DataAdapter.SelectCommand.Connection プロパティが初期化されていません。動的 SQL の生成は、複数のベーステーブルに対してはサポートされていません。動的 SQL の生成は、重複したカラムを含む SelectCommand 値に対してはサポートされていません。DeleteCommand プロパティに対する動的 SQL の生成は、キーカラム情報を返さない SelectCommand 値に対してはサポートされていません。

備考

SQL 文が最初に生成された後で、アプリケーションが ULDataAdapter.SelectCommand 値を変更する場合は、DbCommandBuilder.RefreshSchema メソッドを明示的に呼び出す必要があります。こ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

の処理を行わないと、GetDeleteCommand メソッドは、正しくない可能性がある古い文の情報を使用し続けます。SQL 文が最初に生成されるのは、アプリケーションが DbDataAdapter.Update(System.Data.DataSet) または GetDeleteCommand メソッドを呼び出したときです。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- ULDataAdapter.SelectCommand プロパティ [Ultra Light.NET]217 ページ
- System.Data.Common.DbCommandBuilder.DataAdapter
- System.Data.Common.DbCommandBuilder.RefreshSchema

GetInsertCommand メソッド

データベースで挿入処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

オーバーロードリスト

名前	説明
GetInsertCommand() メソッド	データベースで挿入処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。
GetInsertCommand(bool) メソッド	データベースで挿入処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

GetInsertCommand() メソッド

データベースで挿入処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

Visual Basic 構文

```
Public Shadows Function GetInsertCommand() As ULCommand
```

C# 構文

```
public new ULCommand GetInsertCommand()
```

戻り値

挿入の実行に必要な、自動的に生成された ULCommand オブジェクト。

例外

- **InvalidOperationException** DbCommandBuilder.DataAdapter プロパティが初期化されていません。DataAdapter.SelectCommand プロパティが初期化されていません。

DataAdapter.SelectCommand.Connection プロパティが初期化されていません。InsertCommand プロパティに対する動的 SQL の生成は、修正可能なカラムを返さない SelectCommand 値に対してはサポートされていません。動的 SQL の生成は、複数のベーステーブルに対してはサポートされていません。動的 SQL の生成は、重複したカラムを含む SelectCommand 値に対してはサポートされていません。

備考

SQL 文が最初に生成された後で、アプリケーションが ULDataAdapter.SelectCommand 値を変更する場合は、DbCommandBuilder.RefreshSchema メソッドを明示的に呼び出す必要があります。この処理を行わないと、GetInsertCommand メソッドは、正しくない可能性がある古い文の情報を使用し続けます。SQL 文が最初に生成されるのは、アプリケーションが DbDataAdapter.Update(System.Data.DataSet) メソッドまたは GetInsertCommand メソッドを呼び出したときです。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- ULDataAdapter.SelectCommand プロパティ [Ultra Light.NET]217 ページ
- System.Data.Common.DbCommandBuilder.RefreshSchema
- System.Data.Common.DbCommandBuilder.DataAdapter

GetInsertCommand(bool) メソッド

データベースで挿入処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

Visual Basic 構文

```
Public Shadows Function GetInsertCommand(
     ByVal useColumnsForParameterNames As Boolean
) As ULCommand
```

C# 構文

```
public new ULCommand GetInsertCommand(bool useColumnsForParameterNames)
```

パラメーター

- **useColumnsForParameterNames** true の場合、可能であれば、カラム名に一致するパラメーター名を生成します。false の場合、@p1、@p2 などを生成します。

戻り値

挿入の実行に必要な、自動的に生成された ULCommand オブジェクト。

例外

- **InvalidOperationException** DbCommandBuilder.DataAdapter プロパティが初期化されていません。DataAdapter.SelectCommand プロパティが初期化されていません。DataAdapter.SelectCommand.Connection プロパティが初期化されていません。InsertCommand プロパティに対する動的 SQL の生成は、修正可能なカラムを返さない SelectCommand 値に対してはサポートされていません。動的 SQL の生成は、複数のベーステーブルに対してはサ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

ポートされていません。動的 SQL の生成は、重複したカラムを含む SelectCommand 値に対してはサポートされていません。

備考

SQL 文が最初に生成された後で、アプリケーションが ULDataAdapter.SelectCommand 値を変更する場合は、DbCommandBuilder.RefreshSchema メソッドを明示的に呼び出す必要があります。この処理を行わないと、GetInsertCommand メソッドは、正しくない可能性がある古い文の情報を使用し続けます。SQL 文が最初に生成されるのは、アプリケーションが DbDataAdapter.Update(System.Data.DataSet) メソッドまたは GetInsertCommand メソッドを呼び出したときです。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- ULDataAdapter.SelectCommand プロパティ [Ultra Light.NET]217 ページ
- System.Data.Common.DbCommandBuilder.DataAdapter
- System.Data.Common.DbCommandBuilder.RefreshSchema

GetUpdateCommand メソッド

データベースで更新処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

オーバーロードリスト

名前	説明
GetUpdateCommand() メソッド	データベースで更新処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。
GetUpdateCommand(bool) メソッド	データベースで更新処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

GetUpdateCommand() メソッド

データベースで更新処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

Visual Basic 構文

```
Public Shadows Function GetUpdateCommand() As ULCommand
```

C# 構文

```
public new ULCommand GetUpdateCommand()
```

戻り値

更新の実行に必要な、自動的に生成された ULCommand オブジェクト。

例外

- **InvalidOperationException** DbCommandBuilder.DataAdapter プロパティが初期化されていません。DataAdapter.SelectCommand プロパティが初期化されていません。DataAdapter.SelectCommand.Connection プロパティが初期化されていません。UpdateCommand プロパティに対する動的 SQL の生成は、修正可能なカラムを返さない SelectCommand 値に対してはサポートされていません。動的 SQL の生成は、複数のベーステーブルに対してはサポートされていません。動的 SQL の生成は、重複したカラムを含む SelectCommand 値に対してはサポートされていません。UpdateCommand プロパティに対する動的 SQL の生成は、キーカラム情報を返さない SelectCommand 値に対してはサポートされていません。

備考

SQL 文が最初に生成された後で、アプリケーションが ULDataAdapter.SelectCommand 値を変更する場合は、DbCommandBuilder.RefreshSchema メソッドを明示的に呼び出す必要があります。この処理を行わないと、GetUpdateCommand メソッドは、正しくない可能性がある古い文の情報を使用し続けます。SQL 文が最初に生成されるのは、アプリケーションが DbDataAdapter.Update(System.Data.DataSet) または GetUpdateCommand メソッドを呼び出したときです。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- ULDataAdapter.SelectCommand プロパティ [Ultra Light.NET]217 ページ
- System.Data.Common.DbCommandBuilder.DataAdapter
- System.Data.Common.DbCommandBuilder.RefreshSchema

GetUpdateCommand(bool) メソッド

データベースで更新処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

Visual Basic 構文

```
Public Shadows Function GetUpdateCommand(
     ByVal useColumnsForParameterNames As Boolean
) As ULCommand
```

C# 構文

```
public new ULCommand GetUpdateCommand(bool useColumnsForParameterNames)
```

パラメーター

- **useColumnsForParameterNames** true の場合、可能であれば、カラム名に一致するパラメーター名を生成します。false の場合、@p1、@p2 などを生成します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

戻り値

更新の実行に必要な、自動的に生成された ULCommand オブジェクト。

例外

- **InvalidOperationException** DbCommandBuilder.DataAdapter プロパティが初期化されていません。DataAdapter.SelectCommand プロパティが初期化されていません。DataAdapter.SelectCommand.Connection プロパティが初期化されていません。UpdateCommand プロパティに対する動的 SQL の生成は、修正可能なカラムを返さない SelectCommand 値に対してはサポートされていません。動的 SQL の生成は、複数のベーステーブルに対してはサポートされていません。動的 SQL の生成は、重複したカラムを含む SelectCommand 値に対してはサポートされていません。UpdateCommand プロパティに対する動的 SQL の生成は、キーカラム情報を返さない SelectCommand 値に対してはサポートされていません。

備考

SQL 文が最初に生成された後で、アプリケーションが ULDataAdapter.SelectCommand を変更する場合、DbCommandBuilder.RefreshSchema を明示的に呼び出す必要があります。この処理を行わないと、GetUpdateCommand メソッドは、正しくない可能性がある古い文の情報を使用し続けます。SQL 文が最初に生成されるのは、アプリケーションが DbDataAdapter.Update(System.Data.DataSet) または GetUpdateCommand メソッドを呼び出したときです。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- ULDataAdapter.SelectCommand プロパティ [Ultra Light.NET]217 ページ
- System.Data.Common.DbCommandBuilder.DataAdapter
- System.Data.Common.DbCommandBuilder.RefreshSchema

DataAdapter プロパティ

SQL 文が自動的に生成される ULDataAdapter オブジェクトを取得または設定します。

Visual Basic 構文

```
Public Shadows Property DataAdapter As ULDataAdapter
```

C# 構文

```
public new ULDataAdapter DataAdapter {get;set;}
```

備考

ULDataAdapter オブジェクト。

参照

- ULDataAdapter クラス [Ultra Light.NET]209 ページ

ULConnection クラス

Ultra Light.NET データベースへの接続を表します。

Visual Basic 構文

```
Public NotInheritable Class ULConnection
    Inherits System.Data.Common.DbConnection
```

C# 構文

```
public sealed class ULConnection : System.Data.Common.DbConnection
```

基本クラス

- System.Data.Common.DbConnection

メンバー

継承されたメンバーを含む ULConnection クラスのすべてのメンバー。

名前	説明
ULConnection コンストラクター	ULConnection オブジェクトを初期化します。
BeginDbTransaction method (System.Data.Common.DbConnection から継承)	データベーストランザクションを開始します。
BeginSynchronize メソッド	**UL 拡張：** 現在の SyncParms オブジェクトを使用して同期を非同期に起動します。
BeginTransaction メソッド	トランザクションオブジェクトを返します。
CancelGetNotification メソッド	**UL 拡張：** 指定された名前に一致する、すべてのキューに登録されている保留中の取得通知コールをキャンセルします。
CancelSynchronize メソッド	**UL 拡張：** 実行中の同期を次回キャンセルさせます。
ChangeDatabase メソッド	開いている接続の現在のデータベースを変更します。
ChangeEncryptionKey メソッド	**UL 拡張：** データベースの暗号化キーを、指定された新しいキーに変更します。
ChangePassword メソッド	接続文字列に示されるユーザーのパスワードを、指定される新しいパスワードに変更します。
Close メソッド	データベース接続を閉じます。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
CountUploadRows メソッド	**UL 拡張：** 次回の同期でアップロードする必要のあるロー数を返します。
CreateCommand メソッド	この接続と現在のトランザクションに関連付けられる ULCommand オブジェクトを作成して初期化します。
CreateNotificationQueue メソッド	**UL 拡張：** イベントキューを作成します。
DeclareEvent メソッド	**UL 拡張：** 指定したイベントを宣言します。
DestroyNotificationQueue メソッド	**UL 拡張：** イベントキューを破棄します。
EndSynchronize メソッド	**UL 拡張：** 非同期で起動された同期が終了するまでブロックします。
EnlistTransaction method (System.Data.Common.DbConnection から継承)	指定されたトランザクションにエンリストします。
ExecuteTable メソッド	**UL 拡張：** 直接の操作用に、ULTable オブジェクトのデータベーステーブルを取り出します。
GetLastDownloadTime メソッド	**UL 拡張：** 指定されたパブリケーションの最後のダウンロードの時刻を返します。
GetNewUUID メソッド	**UL 拡張：** 新しい UUID を生成します (System.Guid)。
GetNotification メソッド	**UL 拡張：** 通知またはタイムアウトをブロックします。
GetNotificationParameter メソッド	**UL 拡張：** GetNotification メソッドが読み取ったばかりのイベントのパラメーターの値を取得します。
GetSchema メソッド	サポートされているスキーマコレクションのリストを返します。
GrantConnectTo メソッド	**UL 拡張：** 指定されたパスワードを持つユーザー ID に、Ultra Light データベースへのアクセスを許可します。
OnStateChange method (System.Data.Common.DbConnection から継承)	System.Data.Common.DbConnection.StateChange イベントを発生させます。

名前	説明
Open メソッド	以前に指定された接続文字列を使用してデータベースへの接続を開きます。
RegisterForEvent メソッド	**UL 拡張:** オブジェクトからイベントを取得するためのキューを登録します。
ResetLastDownloadTime メソッド	**UL 拡張：** 最後のダウンロードの時刻をリセットします。
RevokeConnectFrom メソッド	**UL 拡張：** 指定されたユーザー ID から、Ultra Light データベースへのアクセス権を取り消します。
RollbackPartialDownload メソッド	**UL 拡張：** 部分的なダウンロードから、未処理の変更をデータベースにロールバックします。
SendNotification メソッド	**UL 拡張：** 一致するキューに通知を送信します。
SetSyncListener メソッド	同期メッセージの処理に使用するリスナーオブジェクトを指定します。
StartSynchronizationDelete メソッド	**UL 拡張：** 同期用に、この接続によって行われる以後のすべての削除にマークを付けます。
StopSynchronizationDelete メソッド	**UL 拡張：** 削除操作が同期されないようにします。
Synchronize メソッド	**UL 拡張：** 現在の ULConnection.SyncParms を使用してデータベースを同期します。
TriggerEvent メソッド	**UL 拡張：** イベントをトリガーします。
ValidateDatabase メソッド	**UL 拡張:** 現在のデータベースの検証を実行します。
ConnectionString プロパティ	Ultra Light.NET データベースへの接続を開くためのパラメーターを指定します。
ConnectionTimeout プロパティ	この機能は Ultra Light.NET ではサポートされていません。
Database プロパティ	接続を開くデータベースの名前を返します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
DatabaseID プロパティ	**UL 拡張：** グローバルオートインクリメントのカラムに使用するデータベース ID の値を指定します。
DataSource プロパティ	この機能は Ultra Light.NET ではサポートされていません。
DbProviderFactory property (System.Data.Common.DbConnection から継承)	この System.Data.Common.DbConnection の System.Data.Common.DbProviderFactory を取得します。
GlobalAutoIncrementUsage プロパティ	**UL 拡張：** 利用可能なグローバルオートインクリメントの値の使用済み比率 (%) を返します。
LastIdentity プロパティ	**UL 拡張:** 直前に使用した identity の値を返します。
Schema プロパティ	**UL 拡張：** この接続に関連付けられている現在のデータベースのスキーマへのアクセスに使用します。
ServerVersion プロパティ	この機能は Ultra Light.NET ではサポートされていません。
State プロパティ	接続の現在のステータスを返します。
SyncParms プロパティ	**UL 拡張:** この接続の同期設定を指定します。
SyncResult プロパティ	**UL 拡張：** この接続の最後の同期結果を返します。
InfoMessage イベント	Ultra Light.NET が、この接続の警告または情報メッセージを送信するときに発生します。
StateChange イベント	この接続のステータスが変更されると発生します。
INVALID_DATABASE_ID フィールド	**UL 拡張：** ULConnection.DatabaseID プロパティが設定されていないことを示すデータベース ID の定数です。
SYNC_ALL_DB フィールド	データベース全体を表す空のパブリケーションリストです。
SYNC_ALL_PUBS フィールド	すべてのパブリケーションを表すパブリケーション名 "*" です。

備考

Ultra Light.NET の Ultra Light エンジンのランタイムを使用するには、ULDatabaseManager.RuntimeType プロパティを適切な値に設定してから、他の Ultra Light.NET API を使用します。

既存のデータベースへの接続は、ULConnection.Open メソッドを使用して開かれます。

接続は、他の操作の実行前に開きます。また、その接続での操作がすべて終了したら、接続を閉じてからアプリケーションを終了します。また、接続で開いた結果セットとテーブルをすべて閉じてから、その接続を閉じます。

データベースのスキーマには、開いている接続の ULConnection.Schema 値を使用してアクセスできます。

参照

- ULConnection.Open メソッド [Ultra Light.NET]145 ページ
- ULConnection.Schema プロパティ [Ultra Light.NET]159 ページ
- System.Data.IDbConnection
- System.IDisposable

ULConnection コンストラクター

ULConnection オブジェクトを初期化します。

オーバーロードリスト

名前	説明
ULConnection() コンストラクター	ULConnection オブジェクトを初期化します。
ULConnection(string) コンストラクター	ULConnection オブジェクトを、指定された接続文字列で初期化します。

ULConnection() コンストラクター

ULConnection オブジェクトを初期化します。

Visual Basic 構文

```
Public Sub New()
```

C# 構文

```
public ULConnection()
```

備考

接続を開いてから、データベース操作を実行してください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Ultra Light.NET の Ultra Light エンジンのランタイムを使用するには、ULDatabaseManager.RuntimeType プロパティを適切な値に設定してから、他の Ultra Light.NET API を使用します。

ULConnection オブジェクトは、ULConnection.ConnectionString プロパティが設定されていないと、開くことができません。

参照

- ULConnection.Open メソッド [Ultra Light.NET]145 ページ
- ULConnection.ConnectionString プロパティ [Ultra Light.NET]155 ページ

ULConnection(string) コンストラクター

ULConnection オブジェクトを、指定された接続文字列で初期化します。

Visual Basic 構文

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

C# 構文

```
public ULConnection(string connectionString)
```

パラメーター

- **connectionString**　Ultra Light.NET の接続文字列。接続文字列は、キーワード値の組がセミコロンで区切られたリストです。

例外

- **ArgumentException**　指定された接続文字列が無効です。

備考

接続を開いてから、データベース操作を実行してください。

Ultra Light.NET の Ultra Light エンジンのランタイムを使用するには、ULDatabaseManager.RuntimeType プロパティを適切な値に設定してから、他の Ultra Light.NET API を使用します。

接続文字列は、ULConnectionParms オブジェクトを使用して指定できます。

参照

- ULConnection.Open メソッド [Ultra Light.NET]145 ページ
- ULConnection.ULConnection コンストラクター [Ultra Light.NET]120 ページ
- ULConnectionParms クラス [Ultra Light.NET]164 ページ
- ULConnection.ConnectionString プロパティ [Ultra Light.NET]155 ページ

例

次のコードでは、Windows Mobile デバイス上のデータベース ¥UltraLite¥MyDatabase.udb への接続を作成して開きます。

```
' Visual Basic
Dim openParms As ULConnectionParms = New ULConnectionParms
openParms.DatabaseOnDevice = "¥UltraLite¥MyDatabase.udb"
Dim conn As ULConnection = _
    New ULConnection( openParms.ToString() )
conn.Open()
```

対応する C# 言語のコードを次に示します。

```
// C#
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnDevice = ".udb";
ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```

BeginSynchronize メソッド

UL 拡張： 現在の SyncParms オブジェクトを使用して同期を非同期に起動します。

オーバーロードリスト

名前	説明
BeginSynchronize() メソッド	**UL 拡張：** 現在の SyncParms オブジェクトを使用して同期を非同期に起動します。
BeginSynchronize(Control, ULSyncProgressedDlg, object) メソッド	**UL 拡張：** 現在の SyncParms を使用して同期を非同期に起動します。

BeginSynchronize() メソッド

UL 拡張： 現在の SyncParms オブジェクトを使用して同期を非同期に起動します。

Visual Basic 構文

```
Public Function BeginSynchronize() As IAsyncResult
```

C# 構文

```
public IAsyncResult BeginSynchronize()
```

戻り値

同期が完了したか、同期が終了するまでブロックされるかを判断するために使用できる IAsyncResult オブジェクト。

例外

- **ULException クラス** SQL エラーが発生しました。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

このメソッドは、同期化を行う新しいスレッドを作成して、すぐに返します。EndSynchronize メソッドを呼び出して、同期が完了するまでブロックしてください。

参照

- ULConnection.BeginSynchronize メソッド [Ultra Light.NET]122 ページ
- ULConnection.EndSynchronize メソッド [Ultra Light.NET]133 ページ
- ULConnection.CancelSynchronize メソッド [Ultra Light.NET]127 ページ
- ULSyncProgressedDlg デリゲート [Ultra Light.NET]456 ページ

BeginSynchronize(Control, ULSyncProgressedDlg, object) メソッド

UL 拡張： 現在の SyncParms を使用して同期を非同期に起動します。

Visual Basic 構文

```
Public Function BeginSynchronize(
     ByVal control As Control,
    ByVal dlg As ULSyncProgressedDlg,
    ByVal state As Object
) As IAsyncResult
```

C# 構文

```
public IAsyncResult BeginSynchronize(
     Control control,
    ULSyncProgressedDlg dlg,
    object state
)
```

パラメーター

- **control**　同期スレッドが ULSyncProgressedDlg 呼び出しを起動するために使用する System.Windows.Forms.Control オブジェクト。
- **dlg**　定期的に起動されて同期の進行状況を更新する ULSyncProgressedDlg メソッド。
- **state**　このユーザーコンテキストは、IAsyncResult.AsyncState を使用する ULSyncProgressedDlg メソッドでアクセスできます。

戻り値

同期が完了したか、同期が終了するまでブロックされるかを判断するために使用できる IAsyncResult オブジェクト。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

このメソッドは、同期を行った後すぐに戻る新しいメソッドを作成し、同期の進行状況を更新するために提供された ULSyncProgressedDlg メソッドを定期的に起動します。同期が完了するまでブロックするように EndSynchronize を呼び出してください。

参照

- ULConnection.BeginSynchronize メソッド [Ultra Light.NET]122 ページ
- ULConnection.EndSynchronize メソッド [Ultra Light.NET]133 ページ
- ULConnection.CancelSynchronize メソッド [Ultra Light.NET]127 ページ
- ULSyncProgressedDlg デリゲート [Ultra Light.NET]456 ページ

BeginTransaction メソッド

トランザクションオブジェクトを返します。

オーバーロードリスト

名前	説明
BeginTransaction() メソッド	トランザクションオブジェクトを返します。
BeginTransaction(IsolationLevel) メソッド	指定された独立性レベルのトランザクションオブジェクトを返します。

BeginTransaction() メソッド

トランザクションオブジェクトを返します。

Visual Basic 構文

```
Public Shadows Function BeginTransaction() As ULTransaction
```

C# 構文

```
public new ULTransaction BeginTransaction()
```

戻り値

新しいトランザクションを表す ULTransaction オブジェクト。

例外

- **ULException クラス**　接続が閉じています。
- **InvalidOperationException**　ULConnection クラスは並列トランザクションをサポートしていません。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

トランザクションオブジェクトに関連付けられているコマンドは、単一のトランザクションとして実行されます。トランザクションは、ULTransaction.Commit メソッドまたは ULTransaction.Rollback メソッドで終了します。

トランザクションは IsolationLevel.ReadCommitted 値で作成されます。

コマンドをトランザクションオブジェクトに関連付けるには、ULCommand.Transaction プロパティを使用します。現在のトランザクションは、ULConnection.CreateCommand メソッドによって作成されたコマンドに自動的に関連付けられます。

デフォルトでは、接続はトランザクションを使用しません。また、すべてのコマンドは、実行されるときに自動的にコミットされます。現在のトランザクションがコミットまたはロールバックされると、次に BeginTransaction メソッドを呼び出すまで、接続はオートコミットモードおよび以前の独立性レベルに戻ります。

Ultra Light での独立性レベルの定義は、ADO.NET のマニュアルの IsolationLevel の説明とは若干異なります。

これは、System.Data.IDbConnection.BeginTransaction メソッドと System.Data.Common.DbConnection.BeginTransaction メソッドが厳密に型指定されたものです。

参照

- ULTransaction.Commit メソッド [Ultra Light.NET]452 ページ
- ULTransaction.Rollback メソッド [Ultra Light.NET]452 ページ
- ULCommand.Transaction プロパティ [Ultra Light.NET]102 ページ
- ULConnection.CreateCommand メソッド [Ultra Light.NET]130 ページ
- System.Data.IDbConnection.BeginTransaction
- System.Data.Common.DbConnection.BeginTransaction
- 「独立性レベルの変更」『Ultra Light データベース管理とリファレンス』

BeginTransaction(IsolationLevel) メソッド

指定された独立性レベルのトランザクションオブジェクトを返します。

Visual Basic 構文

```
Public Shadows Function BeginTransaction(
     ByVal isolationLevel As IsolationLevel
) As ULTransaction
```

C# 構文

```
public new ULTransaction BeginTransaction(IsolationLevel isolationLevel)
```

パラメーター

- **isolationLevel**　要求されたトランザクションの独立性レベル。Ultra Light.NET では、System.Data.IsolationLevel.ReadUncommitted および ReadCommitted 値のみがサポートされています。

戻り値

新しいトランザクションを表す ULTransaction オブジェクト。

例外

- **ULException クラス**　接続が閉じているか、サポートされていない独立性レベルが指定されました。
- **InvalidOperationException**　ULConnection クラスは並列トランザクションをサポートしていません。

備考

トランザクションオブジェクトに関連付けられているコマンドは、単一のトランザクションとして実行されます。トランザクションは、ULTransaction.Commit メソッドまたは ULTransaction.Rollback メソッドで終了します。

コマンドをトランザクションオブジェクトに関連付けるには、ULCommand.Transaction プロパティを使用します。現在のトランザクションは、ULConnection.CreateCommand メソッドによって作成されたコマンドに自動的に関連付けられます。

デフォルトでは、接続はトランザクションを使用しません。また、すべてのコマンドは、実行されるときに自動的にコミットされます。現在のトランザクションがコミットまたはロールバックされると、次に BeginTransaction メソッドを呼び出すまで、接続はオートコミットモードおよび以前の独立性レベルに戻ります。

Ultra Light での独立性レベルの定義は、ADO.NET のマニュアルの IsolationLevel の説明とは若干異なります。

これは、System.Data.IDbConnection.BeginTransaction(System.Data.IsolationLevel) メソッドと System.Data.Common.DbConnection.BeginTransaction(System.Data.IsolationLevel) メソッドが厳密に型指定されたものです。

参照

- ULTransaction クラス [Ultra Light.NET]451 ページ
- ULConnection.BeginTransaction メソッド [Ultra Light.NET]124 ページ
- ULTransaction.Commit メソッド [Ultra Light.NET]452 ページ
- ULTransaction.Rollback メソッド [Ultra Light.NET]452 ページ
- ULCommand.Transaction プロパティ [Ultra Light.NET]102 ページ
- ULConnection.CreateCommand メソッド [Ultra Light.NET]130 ページ
- System.Data.IDbConnection.BeginTransaction
- System.Data.Common.DbConnection.BeginTransaction
- System.Data.IsolationLevel
- 「独立性レベルの変更」『Ultra Light データベース管理とリファレンス』

CancelGetNotification メソッド

UL 拡張： 指定された名前に一致する、すべてのキューに登録されている保留中の取得通知コールをキャンセルします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public Function CancelGetNotification(
     ByVal queueName As String
) As Integer
```

C# 構文

```
public int CancelGetNotification(string queueName)
```

パラメーター

- **queueName**　キュー名と一致する式。

戻り値

影響を受けるキューの数 (ブロックされた読み込み数とは異なります)。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

このメソッドは、指定された名前に一致する、すべてのキューに登録されている保留中の取得通知コールをキャンセルします。

参照

- ULConnection.GetNotification メソッド [Ultra Light.NET]139 ページ
- ULException クラス [Ultra Light.NET]274 ページ

CancelSynchronize メソッド

UL 拡張： 実行中の同期を次回キャンセルさせます。

Visual Basic 構文

```
Public Sub CancelSynchronize(ByVal asyncResult As IAsyncResult)
```

C# 構文

```
public void CancelSynchronize(IAsyncResult asyncResult)
```

パラメーター

- **asyncResult**　BeginSynchronize メソッドから返された IAsyncResult。

備考

このメソッドは、同期化スレッドに終了を伝えて、すぐに戻ります。EndSynchronize メソッドを呼び出して、同期が正常に終了するまでブロックしてください。

参照

- ULConnection.BeginSynchronize メソッド [Ultra Light.NET]122 ページ
- ULConnection.EndSynchronize メソッド [Ultra Light.NET]133 ページ

ChangeDatabase メソッド

開いている接続の現在のデータベースを変更します。

Visual Basic 構文

```
Public Overrides Sub ChangeDatabase(ByVal connectionString As String)
```

C# 構文

```
public override void ChangeDatabase(string connectionString)
```

パラメーター

- **connectionString**　新しいデータベースへの接続を開く完全な接続文字列。

備考

パラメーターのエラーがあった場合でも、現在のデータベースへの接続は閉じられます。

UL 拡張： *connectionString* は、dbn 値でも dbf 値でもなく、完全な接続文字列です。

参照

- ULConnection.ConnectionString プロパティ [Ultra Light.NET]155 ページ

ChangeEncryptionKey メソッド

UL 拡張： データベースの暗号化キーを、指定された新しいキーに変更します。

Visual Basic 構文

```
Public Sub ChangeEncryptionKey(ByVal newKey As String)
```

C# 構文

```
public void ChangeEncryptionKey(string newKey)
```

パラメーター

- **newkey**　データベースの新しい暗号化キー。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

暗号化キーが失われた場合は、データベースを開くことができません。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULConnectionParms.EncryptionKey プロパティ [Ultra Light.NET]173 ページ

ChangePassword メソッド

接続文字列に示されるユーザーのパスワードを、指定される新しいパスワードに変更します。

Visual Basic 構文

```
Public Shared Sub ChangePassword(
     ByVal connectionString As String,
    ByVal newPassword As String
)
```

C# 構文

```
public static void ChangePassword(
     string connectionString,
    string newPassword
)
```

パラメーター

- **connectionString** 目的のデータベースに接続できるだけの情報が含まれる接続文字列。接続文字列には、ユーザー ID と現在のパスワードが含まれる可能性があります。
- **newPassword** 設定する新しいパスワード。

例外

- **ArgumentNullException** connectionString または newPassword parameter のいずれかが null です。
- **ArgumentException** 接続文字列に、統合化セキュリティを使用するオプションが含まれています。
- **ULException クラス** データベースを開くときに SQL エラーが発生しました。

Close メソッド

データベース接続を閉じます。

Visual Basic 構文

```
Public Overrides Sub Close()
```

C# 構文

```
public override void Close()
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

Close メソッドは、保留中のトランザクションをロールバックし、その接続を閉じます。アプリケーションは、このメソッドを複数回呼び出すことができます。

CountUploadRows メソッド

UL 拡張： 次回の同期でアップロードする必要のあるロー数を返します。

Visual Basic 構文

```
Public Function CountUploadRows(
     ByVal pubs As String,
    ByVal threshold As Long
) As Long
```

C# 構文

```
public long CountUploadRows(string pubs, long threshold)
```

パラメーター

- **pubs**　ローをチェックするパブリケーションのカンマ区切りのリスト。
- **threshold**　カウントするローの最大数。CountUploadRows の所要時間を制限します。値 0 は制限が最大であることを示します。値 1 は、同期の必要なローがあるかどうかを判別する場合に使用します。

戻り値

指定されたパブリケーションからアップロードする必要のあるロー数。

例外

- **ULException クラス**　SQL エラーが発生しました。

CreateCommand メソッド

この接続と現在のトランザクションに関連付けられる ULCommand オブジェクトを作成して初期化します。

Visual Basic 構文

```
Public Shadows Function CreateCommand() As ULCommand
```

C# 構文

```
public new ULCommand CreateCommand()
```


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

戻り値

新しい ULCommand オブジェクト。

備考

ULCommand オブジェクトのプロパティを使用して、その動作を制御できます。

ULCommand.CommandText プロパティを設定してから、コマンドを実行する必要があります。

これは、System.Data.IDbConnection.CreateCommand メソッドと System.Data.Common.DbConnection.CreateCommand メソッドが厳密に型指定されたものです。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- System.Data.IDbConnection.CreateCommand
- System.Data.Common.DbConnection.CreateCommand

CreateNotificationQueue メソッド

UL 拡張： イベントキューを作成します。

Visual Basic 構文

```
Public Sub CreateNotificationQueue(
     ByVal queueName As String,
    ByVal parameters As String
)
```

C# 構文

```
public void CreateNotificationQueue(string queueName, string parameters)
```

パラメーター

- **queueName** 新しいキューの名前。
- **parameters** 作成パラメーター。現在は使われず、NULL に設定されています。

例外

- **ULException クラス** SQL エラーが発生しました。

備考

このメソッドは、この接続のイベント通知キューを作成します。キュー名は、接続ごとにスコープされるため、別々の接続で同じ名前を持つキューを作成できます。イベント通知が送信されると、データベース内で一致する名前を持つすべてのキューが、個別のインスタンスの通知を受け取ります。名前では、大文字と小文字が区別されません。RegisterForEvent イベントを呼び出したときに、キューが指定されていない場合は、接続ごとにデフォルトのキューが作成されます。

参照

- ULConnection.DestroyNotificationQueue メソッド [Ultra Light.NET]132 ページ
- ULException クラス [Ultra Light.NET]274 ページ

DeclareEvent メソッド

UL 拡張： 指定したイベントを宣言します。

Visual Basic 構文

```
Public Sub DeclareEvent(ByVal eventName As String)
```

C# 構文

```
public void DeclareEvent(string eventName)
```

パラメーター

- **eventName** イベント名。

例外

- **ULException クラス** SQL エラーが発生しました。

備考

登録およびトリガーされるイベントを宣言します。Ultra Light では、データベースまたは環境での操作によってトリガーされるシステムイベントの一部が事前に定義されています。イベント名は、ユニークにする必要があります。名前では、大文字と小文字が区別されません。このメソッドは、名前がすでに使用されているか無効な場合は、エラーをスローします。

参照

- ULConnection.CreateNotificationQueue メソッド [Ultra Light.NET]131 ページ
- ULException クラス [Ultra Light.NET]274 ページ

DestroyNotificationQueue メソッド

UL 拡張： イベントキューを破棄します。

Visual Basic 構文

```
Public Sub DestroyNotificationQueue(ByVal queueName As String)
```

C# 構文

```
public void DestroyNotificationQueue(string queueName)
```

パラメーター

- **queueName** キューの名前。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

指定されたイベント通知キューを破棄します。キュー内に未読の通知が残っている場合は、警告が通知されます。未読の通知は破棄されます。接続のデフォルトのイベントキューが作成されている場合、接続が閉じると破棄されます。

参照

- ULConnection.CreateNotificationQueue メソッド [Ultra Light.NET]131 ページ
- ULException クラス [Ultra Light.NET]274 ページ

EndSynchronize メソッド

UL 拡張： 非同期で起動された同期が終了するまでブロックします。

Visual Basic 構文

```
Public Sub EndSynchronize(ByVal asyncResult As IAsyncResult)
```

C# 構文

```
public void EndSynchronize(IAsyncResult asyncResult)
```

パラメーター

- **asyncResult**　BeginSynchronize メソッドから返された IAsyncResult。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

同期中にエラーが発生した場合、ULException 例外がスローされます。

参照

- ULConnection.BeginSynchronize メソッド [Ultra Light.NET]122 ページ
- ULConnection.CancelSynchronize メソッド [Ultra Light.NET]127 ページ

ExecuteTable メソッド

UL 拡張： 直接操作するために、ULTable オブジェクトのデータベーステーブルを取り出します。

オーバーロードリスト

名前	説明
ExecuteTable(string) メソッド	**UL 拡張：** 直接の操作用に、ULTable オブジェクトのデータベーステーブルを取り出します。
ExecuteTable(string, string) メソッド	**UL 拡張：** 直接の操作用に、ULTable オブジェクトのデータベーステーブルを取り出します。
ExecuteTable(string, string, CommandBehavior) メソッド	**UL 拡張：** 直接の操作用に、コマンド動作を指定してデータベーステーブルを取り出します。

ExecuteTable(string) メソッド

UL 拡張： 直接の操作用に、ULTable オブジェクトのデータベーステーブルを取り出します。

Visual Basic 構文

```
Public Function ExecuteTable(ByVal tableName As String) As ULTable
```

C# 構文

```
public ULTable ExecuteTable(string tableName)
```

パラメーター

- **tableName** 開くテーブルの名前。

戻り値

ULTable オブジェクトとしてのテーブル。

例外

- **ULException クラス** SQL エラーが発生しました。
- **InvalidOperationException** *tableName* は無効です。

備考

テーブルのプライマリキーを使用して、テーブルが開かれます (ソートされます)。

このメソッドは、ULCommand オブジェクトを必要としない ULCommand.ExecuteTable メソッドのショートカットです。また、以前のバージョンの Ultra Light.NET から移行できるようにするために用意されています (iAnywhere.UltraLite.Connection.GetTable メソッドと iAnywhere.UltraLite.Table.Open メソッドが置き換えられます)。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULConnection.ExecuteTable メソッド [Ultra Light.NET]133 ページ
- ULTable クラス [Ultra Light.NET]415 ページ
- ULCommand.ExecuteTable メソッド [Ultra Light.NET]94 ページ
- ULCommand クラス [Ultra Light.NET]66 ページ

例

次のコードでは、テーブルのプライマリキーを使用して MyTable というテーブルが開かれます。また、conn という開いた ULConnection インスタンスが想定されています。

```
' Visual Basic
Dim t As ULTable = 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 ULTable = cmd.ExecuteTable()
' cmd.Dispose()
```

対応する C# 言語のコードを次に示します。

```
// C#
ULTable t = conn.ExecuteTable("MyTable");

// The line above is equivalent to
// ULTable t;
// using(ULCommand cmd = conn.CreateCommand())
// {
//     cmd.CommandText = "MyTable";
//     cmd.CommandType = CommandType.TableDirect;
//     t = cmd.ExecuteTable();
// }
```

ExecuteTable(string, string) メソッド

UL 拡張： 直接の操作用に、ULTable オブジェクトのデータベーステーブルを取り出します。

Visual Basic 構文

```
Public Function ExecuteTable(
     ByVal tableName As String,
    ByVal indexName As String
) As ULTable
```

C# 構文

```
public ULTable ExecuteTable(string tableName, string indexName)
```

パラメーター

- **tableName**　開くテーブルの名前。
- **indexName**　テーブルを開く (ソートする) インデックスの名前。

戻り値

ULTable オブジェクトとしてのテーブル。

例外

- **ULException クラス**　SQL エラーが発生しました。
- **InvalidOperationException**　*tableName* は無効です。

備考

テーブルは、指定されたインデックスを使用して開かれます (ソートされます)。

このメソッドは、ULCommand オブジェクトを必要としない ULCommand.ExecuteTable メソッドのショートカットです。また、以前のバージョンの Ultra Light.NET から移行できるようにするために用意されています (iAnywhere.UltraLite.Connection.GetTable メソッドと iAnywhere.UltraLite.Table.Open メソッドが置き換えられます)。

参照

- ULConnection.ExecuteTable メソッド [Ultra Light.NET]133 ページ
- ULTable クラス [Ultra Light.NET]415 ページ
- ULCommand.ExecuteTable メソッド [Ultra Light.NET]94 ページ
- ULCommand クラス [Ultra Light.NET]66 ページ

例

次のコードでは、MyIndex というインデックスを使用して MyTable というテーブルが開かれます。また、conn という開いた ULConnection オブジェクトが想定されています。

```
' 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()
```

対応する C# 言語のコードを次に示します。

```
// 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();
// }
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

ExecuteTable(string, string, CommandBehavior) メソッド

UL 拡張： 直接の操作用に、コマンド動作を指定してデータベーステーブルを取り出します。

Visual Basic 構文

```
Public Function ExecuteTable(
     ByVal tableName As String,
    ByVal indexName As String,
    ByVal cmdBehavior As CommandBehavior
) As ULTable
```

C# 構文

```
public ULTable ExecuteTable(
     string tableName,
    string indexName,
    CommandBehavior cmdBehavior
)
```

パラメーター

- **tableName**　開くテーブルの名前。
- **indexName**　テーブルを開く (ソートする) インデックスの名前。
- **cmdBehavior**　クエリの結果の記述と、接続への影響の記述の System.Data.CommandBehavior フラグのビット単位の組み合わせ。Ultra Light.NET で有効なのは、System.Data.CommandBehavior.Default フラグ、System.Data.CommandBehavior.CloseConnection フラグ、System.Data.CommandBehavior.SchemaOnly フラグだけです。

戻り値

ULTable オブジェクトとしてのテーブル。

例外

- **ULException クラス**　SQL エラーが発生しました。
- **InvalidOperationException**　*tableName* は無効です。

備考

テーブルは、指定されたインデックスを使用して開かれます (ソートされます)。

このメソッドは、ULCommand オブジェクトを必要としない ULCommand.ExecuteTable(System.Data.CommandBehavior) メソッドのショートカットです。また、以前のバージョンの Ultra Light.NET から移行できるようにするために用意されています (iAnywhere.UltraLite.Connection.GetTable メソッドと iAnywhere.UltraLite.Table.Open メソッドが置き換えられます)。

参照

- ULConnection.ExecuteTable メソッド [Ultra Light.NET]133 ページ
- ULTable クラス [Ultra Light.NET]415 ページ
- ULCommand クラス [Ultra Light.NET]66 ページ
- System.Data.CommandBehavior
- System.Data.CommandBehavior.Default
- System.Data.CommandBehavior.CloseConnection
- System.Data.CommandBehavior.SchemaOnly

例

次のコードでは、MyIndex というインデックスを使用して MyTable というテーブルが開かれます。また、conn という、開いた ULConnection オブジェクトが想定されています。

```
' 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()
```

対応する C# 言語のコードを次に示します。

```
// 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 メソッド

UL 拡張： 指定されたパブリケーションの最後のダウンロードの時刻を返します。

Visual Basic 構文

```
Public Function GetLastDownloadTime(ByVal publication As String) As Date
```

C# 構文

```
public DateTime GetLastDownloadTime(string publication)
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

パラメーター

- **publication**　チェックするパブリケーション。

戻り値

最後のダウンロードのタイムスタンプ。パブリケーションに SYNC_ALL_DB 定数を使用した場合は、データベース全体を最後にダウンロードした時刻を返します。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

publication パラメーターは、チェックするパブリケーションの名前です。

参照

- ULConnection.ResetLastDownloadTime メソッド [Ultra Light.NET]147 ページ
- ULConnection.SYNC_ALL_DB フィールド [Ultra Light.NET]163 ページ

GetNewUUID メソッド

UL 拡張： 新しい UUID を生成します (System.Guid)。

Visual Basic 構文

```
Public Function GetNewUUID() As Guid
```

C# 構文

```
public Guid GetNewUUID()
```

戻り値

System.Guid として返される新しい UUID。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

このメソッドは、.NET Compact Framework に含まれていないため、ここで提供されています。

参照

- System.Guid

GetNotification メソッド

UL 拡張： 通知またはタイムアウトをブロックします。

Visual Basic 構文

```
Public Function GetNotification(
     ByVal queueName As String,
    ByVal wait_ms As Integer
) As String
```

C# 構文

```
public string GetNotification(string queueName, int wait_ms)
```

パラメーター

- **queueName**　待機するキューの名前。
- **wait_ms**　待機時間 (ミリ秒)。待機時間を無限に設定するには、System.Threading.Timeout.Infinite (-1) を使用します。

戻り値

待機時間の期限が切れるか、待機がキャンセルされた場合は NULL。それ以外の場合はイベント名。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

このメソッドは、イベント通知を読み込みます。この呼び出しは、通知が受信されるまで、または指定された待機時間が経過するまでブロックします。無期限に待機する場合は、wait_ms パラメーターに System.Threading.Timeout.Infinite を渡します。待機をキャンセルするには、指定したキューに別の通知を送信するか、CancelGetNotification メソッドを使用します。通知を読み込んだら、ReadNotificationParameter メソッドを使用して追加のパラメーターを取得します。

参照

- ULConnection.SendNotification メソッド [Ultra Light.NET]148 ページ
- ULConnection.GetNotificationParameter メソッド [Ultra Light.NET]140 ページ
- ULConnection.CancelGetNotification メソッド [Ultra Light.NET]126 ページ
- ULException クラス [Ultra Light.NET]274 ページ
- System.Threading.Timeout

GetNotificationParameter メソッド

UL 拡張： GetNotification メソッドが読み取ったばかりのイベントのパラメーターの値を取得します。

Visual Basic 構文

```
Public Function GetNotificationParameter(
     ByVal queueName As String,
    ByVal parameterName As String
) As String
```


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

C# 構文

```
public string GetNotificationParameter(
     string queueName,
    string parameterName
)
```

パラメーター

- **queueName**　待機するキューの名前。
- **parameterName**　値を返すパラメーターの名前。

戻り値

パラメーターが見つかった場合はパラメーター値、それ以外の場合は NULL。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

このメソッドは、ULGetNotification メソッドによって読み込まれたイベント通知のパラメーター値を取得します。指定されたキューでの最後に読み込まれた通知のパラメーターのみ取得できます。

参照

- ULConnection.GetNotification メソッド [Ultra Light.NET]139 ページ
- ULException クラス [Ultra Light.NET]274 ページ

GetSchema メソッド

サポートされているスキーマコレクションのリストを返します。

オーバーロードリスト

名前	説明
GetSchema() メソッド	サポートされているスキーマコレクションのリストを返します。
GetSchema(string) メソッド	この ULConnection オブジェクトについて指定されたメタデータコレクションに関する情報を返します。

名前	説明
GetSchema(string, string[]) メソッド	この ULConnection オブジェクトのデータソースのスキーマ情報を返します。このとき、文字列が指定されている場合はスキーマ名として使用し、文字列配列が指定されている場合は制限値として使用します。

GetSchema() メソッド

サポートされているスキーマコレクションのリストを返します。

Visual Basic 構文

```
Public Overrides Function GetSchema() As DataTable
```

C# 構文

```
public override DataTable GetSchema()
```

GetSchema(string) メソッド

この ULConnection オブジェクトについて指定されたメタデータコレクションに関する情報を返します。

Visual Basic 構文

```
Public Overrides Function GetSchema(
     ByVal collection As String
) As DataTable
```

C# 構文

```
public override DataTable GetSchema(string collection)
```

パラメーター

- **collection**　メタデータコレクションの名前。名前が提供されない場合は、MetaDataCollections 値が使用されます。

参照

- ULConnection.GetSchema メソッド [Ultra Light.NET]141 ページ
- ULConnection クラス [Ultra Light.NET]116 ページ

GetSchema(string, string[]) メソッド

この ULConnection オブジェクトのデータソースのスキーマ情報を返します。このとき、文字列が指定されている場合はスキーマ名として使用し、文字列配列が指定されている場合は制限値として使用します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public Overrides Function GetSchema(
     ByVal collection As String,
    ByVal restrictions As String()
) As DataTable
```

C# 構文

```
public override DataTable GetSchema(
     string collection,
    string[] restrictions
)
```

パラメーター

- **collection**　メタデータコレクションの名前。名前が提供されない場合は、MetaDataCollections が使用されます。
- **restrictions**　要求されるスキーマの制限値のセット。

戻り値

スキーマ情報が格納されている DataTable オブジェクト。

備考

このメソッドを使用すると、データベースに各種のメタデータを問い合わせることができます。メタデータの各型にはコレクション名が指定されており、そのデータを受け取るにはコレクション名を渡す必要があります。デフォルトのコレクション名は MetaDataCollections です。

引数を指定せずに、または **MetaDataCollections** というスキーマコレクション名を指定して GetSchema メソッドを呼び出すことによって、.NET データプロバイダーに問い合わせをして、サポートされているスキーマコレクションのリストを判断できます。これによって、サポートされているスキーマコレクション (CollectionName)、それぞれがサポートする制限の数 (NumberOfRestrictions)、使用する識別子部分の数のリストから成る DataTable が返されます。

コレクション	メタデータ
Columns	データベース内のすべてのカラムに関する情報を返します。
DataSourceInformation	データベースプロバイダーに関する情報を返します。
DataType	サポートされているデータ型のリストを返します。
ForeignKeys	データベース内のすべての外部キーに関する情報を返します。
IndexColumns	データベース内のすべてのインデックスカラムに関する情報を返します。

コレクション	メタデータ
Indexes	データベース内のすべてのインデックスに関する情報を返します。
MetaDataCollections	すべてのコレクション名のリストを返します。
Publications	データベース内のすべてのパブリケーションに関する情報を返します。
ReservedWords	Ultra Light が使用する予約語のリストを返します。
Restrictions	GetSchema で使用される制限に関する情報を返します。
Tables	データベース内のすべてのテーブルに関する情報を返します。

これらのコレクション名は、ULMetaDataCollectionNames クラスの読み込み専用プロパティとしても使用できます。

返される結果は、GetSchema メソッドへの呼び出しの中で制限の配列を指定することによってフィルターできます。

各コレクションで有効な制限は、次の文を呼び出すことで問い合わせることができます。

```
GetSchema( "Restrictions" )
```

コレクションが 4 つの制限を必要とする場合、制限パラメーターは、NULL、または 4 つの値から成る文字列です。

特定の制限をフィルターするには、フィルターする文字列を配列内の所定の位置に指定し、使用しない位置には NULL を指定します。たとえば、Tables コレクションには、Table、TableType、SyncType という 3 つの制限があります。

Table コレクションをフィルターするには、次の手順に従います。

GetSchema("Tables", new string[ ] { "my_table", NULL, NULL }): my_table という名前のすべてのテーブルに関する情報を返します。

GetSchema("Tables", new string[ ] { NULL, "User", NULL }): すべてのユーザーテーブルに関する情報を返します。

参照

- ULConnection クラス [Ultra Light.NET]116 ページ
- ULMetaDataCollectionNames クラス [Ultra Light.NET]310 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

GrantConnectTo メソッド

UL 拡張： 指定されたパスワードを持つユーザー ID に、Ultra Light データベースへのアクセスを許可します。

Visual Basic 構文

```
Public Sub GrantConnectTo(ByVal uid As String, ByVal pwd As String)
```

C# 構文

```
public void GrantConnectTo(string uid, string pwd)
```

パラメーター

- **uid** データベースへのアクセス権を受け取るユーザー ID。
- **pwd** ユーザー ID に関連付けられるパスワード。

備考

既存のユーザー ID が指定されていれば、この関数を使用してそのユーザーのパスワードを更新します。Ultra Light では、最大で 4 人のユーザーがサポートされます。このメソッドが有効になるのは、接続が開かれたときにユーザー認証が有効になっていた場合だけです。

参照

- ULConnectionParms.UserID プロパティ [Ultra Light.NET]174 ページ
- ULConnectionParms.Password プロパティ [Ultra Light.NET]174 ページ
- ULConnection.ConnectionString プロパティ [Ultra Light.NET]155 ページ

Open メソッド

以前に指定された接続文字列を使用してデータベースへの接続を開きます。

Visual Basic 構文

```
Public Overrides Sub Open()
```

C# 構文

```
public override void Open()
```

例外

- **InvalidOperationException** 接続がすでに開かれているか、接続文字列が ULConnection.ConnectionString プロパティで指定されていません。
- **ULException クラス** データベースを開くときに SQL エラーが発生しました。

備考

接続は、使用し終わったら明示的に閉じるか、破棄する必要があります。

参照

- ULConnection.ConnectionString プロパティ [Ultra Light.NET]155 ページ
- ULConnection.State プロパティ [Ultra Light.NET]160 ページ

RegisterForEvent メソッド

UL 拡張：オブジェクトからイベントを取得するためのキューを登録します。

Visual Basic 構文

```
Public Sub RegisterForEvent(
     ByVal eventName As String,
    ByVal objectName As String,
    ByVal queueName As String,
    ByVal registerNotUnReg As Boolean
)
```

C# 構文

```
public void RegisterForEvent(
     string eventName,
    string objectName,
    string queueName,
    bool registerNotUnReg
)
```

パラメーター

- **eventName**　イベント名。
- **objectName**　イベントを適用するオブジェクトの名前。たとえば、テーブル名です。
- **queueName**　使用するイベントキューの名前。
- **registerNotUnReg**　登録する場合は true、登録解除する場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

このメソッドは、イベントの通知を受信するキューを登録します。キュー名が指定されていない場合は、デフォルトの接続キューが暗黙で指定され、作成されます。特定のシステムイベントでは、そのイベントが適用されるオブジェクト名を指定できます。たとえば、TableModified イベントではテーブル名を指定できます。SendNotification メソッドとは異なり、登録された特定のキューのみイベントの通知を受信します。別の接続における同じ名前のキューでは、明示的に登録されていないかぎり、通知は受信されません。キューまたはイベントが存在しない場合は、エラーをスローします。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULConnection.DeclareEvent メソッド [Ultra Light.NET]132 ページ
- ULConnection.CreateNotificationQueue メソッド [Ultra Light.NET]131 ページ
- ULException クラス [Ultra Light.NET]274 ページ

ResetLastDownloadTime メソッド

UL 拡張： 最後のダウンロードの時刻をリセットします。

Visual Basic 構文

```
Public Sub ResetLastDownloadTime(ByVal pubs As String)
```

C# 構文

```
public void ResetLastDownloadTime(string pubs)
```

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULConnection.GetLastDownloadTime メソッド [Ultra Light.NET]138 ページ

RevokeConnectFrom メソッド

UL 拡張： 指定されたユーザー ID から、Ultra Light データベースへのアクセス権を取り消します。

Visual Basic 構文

```
Public Sub RevokeConnectFrom(ByVal uid As String)
```

C# 構文

```
public void RevokeConnectFrom(string uid)
```

パラメーター

- **uid**　データベースへのアクセス権を取り消すユーザー ID。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULConnection.GrantConnectTo メソッド [Ultra Light.NET]145 ページ

RollbackPartialDownload メソッド

UL 拡張： 部分的なダウンロードから、未処理の変更をデータベースにロールバックします。

Visual Basic 構文

```
Public Sub RollbackPartialDownload()
```

C# 構文

```
public void RollbackPartialDownload()
```

例外

- **ULException クラス** SQL エラーが発生しました。

参照

- ULSyncParms.KeepPartialDownload プロパティ [Ultra Light.NET]394 ページ
- ULSyncParms.ResumePartialDownload プロパティ [Ultra Light.NET]397 ページ

SendNotification メソッド

UL 拡張： 一致するキューに通知を送信します。

Visual Basic 構文

```
Public Function SendNotification(
     ByVal queueName As String,
    ByVal eventName As String,
    ByVal parameters As String
) As Integer
```

C# 構文

```
public int SendNotification(
     string queueName,
    string eventName,
    string parameters
)
```

パラメーター

- **queueName** 使用するイベントキューの名前。
- **eventName** イベント名。
- **parameters** 受け渡すパラメーター。

戻り値

送信済みの通知の数 (一致するキューの数)。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

一致するキューの数を返します。

このメソッドは、指定された名前に一致するすべてのキュー (現在の接続におけるキューを含む) に通知を送信します。この呼び出しはブロックしません。特別なキュー名の "*" を使用すると、すべてのキューに送信します。

参照

- ULConnection.DeclareEvent メソッド [Ultra Light.NET]132 ページ
- ULConnection.RegisterForEvent メソッド [Ultra Light.NET]146 ページ
- ULException クラス [Ultra Light.NET]274 ページ

SetSyncListener メソッド

同期メッセージの処理に使用するリスナーオブジェクトを指定します。

Visual Basic 構文

```
Public Sub SetSyncListener(ByVal listener As ULSyncProgressListener)
```

C# 構文

```
public void SetSyncListener(ULSyncProgressListener listener)
```

パラメーター

- **listener**　SyncProgressed メソッドを実装する ULSyncProgressListener オブジェクト。この接続で、同期メッセージに対して呼び出されます。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

SQL 文 SYNCHRONIZE profileName が実行されると、その進行状況のメッセージが NULL (Microsoft Visual Basic の場合は Nothing) でない場合は syncListener オブジェクトにルーティングされます。

リスナーを削除するには、SetSyncListener メソッドへの呼び出しで NULL 参照を渡します。

参照

- ULSyncProgressListener インターフェイス [Ultra Light.NET]409 ページ

StartSynchronizationDelete メソッド

UL 拡張： 同期用に、この接続によって行われる以後のすべての削除にマークを付けます。

Visual Basic 構文

```
Public Sub StartSynchronizationDelete()
```

C# 構文

```
public void StartSynchronizationDelete()
```

例外

- **ULException クラス** SQL エラーが発生しました。

備考

このメソッドが呼び出されると、すべての削除操作が再度同期され、Ultra Light データベースから削除されたローが統合データベースからも削除されます。

参照

- ULConnection.StopSynchronizationDelete メソッド [Ultra Light.NET]150 ページ
- ULTable.Truncate メソッド [Ultra Light.NET]436 ページ

StopSynchronizationDelete メソッド

UL 拡張： 削除操作が同期されないようにします。

Visual Basic 構文

```
Public Sub StopSynchronizationDelete()
```

C# 構文

```
public void StopSynchronizationDelete()
```

例外

- **ULException クラス** SQL エラーが発生しました。

備考

領域を節約するために Ultra Light データベースの古い情報を削除して、統合データベースにはそれを残しておく場合に、このメソッドを使用すると便利です。

参照

- ULConnection.StartSynchronizationDelete メソッド [Ultra Light.NET]150 ページ

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Synchronize メソッド

UL 拡張： 現在の ULConnection.SyncParms を使用してデータベースを同期します。

オーバーロードリスト

名前	説明
Synchronize() メソッド	**UL 拡張：** 現在の ULConnection.SyncParms を使用してデータベースを同期します。
Synchronize(ULSyncProgressListener) メソッド	**UL 拡張：** 指定されたリスナーに送信されるプログレスイベントとともに、現在の ULConnection.SyncParms オブジェクトを使用してデータベースを同期します。

Synchronize() メソッド

UL 拡張： 現在の ULConnection.SyncParms を使用してデータベースを同期します。

Visual Basic 構文

```
Public Sub Synchronize()
```

C# 構文

```
public void Synchronize()
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

結果の詳細なステータスは、この接続の ULConnection.SyncResult プロパティでレポートされます。

参照

- ULConnection.Synchronize メソッド [Ultra Light.NET]151 ページ
- ULConnection.SyncParms プロパティ [Ultra Light.NET]160 ページ
- ULConnection.SyncResult プロパティ [Ultra Light.NET]161 ページ

Synchronize(ULSyncProgressListener) メソッド

UL 拡張： 指定されたリスナーに送信されるプログレスイベントとともに、現在の ULConnection.SyncParms オブジェクトを使用してデータベースを同期します。

Visual Basic 構文

```
Public Sub Synchronize(ByVal listener As ULSyncProgressListener)
```

C# 構文

```
public void Synchronize(ULSyncProgressListener listener)
```

パラメーター

- **listener**　同期プログレスイベントを受信するオブジェクト。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

同期中のエラーは、ULSyncProgressState.STATE_ERROR イベントとして送信され、ULExceptions としてスローされます。

結果の詳細なステータスは、この接続の ULConnection.SyncResult プロパティでレポートされます。

参照

- ULSyncProgressListener インターフェイス [Ultra Light.NET]409 ページ
- ULConnection.Synchronize メソッド [Ultra Light.NET]151 ページ
- ULConnection.SyncParms プロパティ [Ultra Light.NET]160 ページ
- ULException クラス [Ultra Light.NET]274 ページ
- ULConnection.SyncResult プロパティ [Ultra Light.NET]161 ページ

TriggerEvent メソッド

UL 拡張： イベントをトリガーします。

Visual Basic 構文

```
Public Function TriggerEvent(
     ByVal eventName As String,
    ByVal parameters As String
) As Integer
```

C# 構文

```
public int TriggerEvent(string eventName, string parameters)
```

パラメーター

- **eventName**　トリガーされるイベントの名前。
- **parameters**　受け渡すパラメーター。

戻り値

送信済みのイベント通知の数。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

送信済みの通知の数を返します。

このメソッドは、イベントをトリガーして、登録されたすべてのキューに通知を送信します。

参照

- ULConnection.DeclareEvent メソッド [Ultra Light.NET]132 ページ
- ULConnection.RegisterForEvent メソッド [Ultra Light.NET]146 ページ
- ULException クラス [Ultra Light.NET]274 ページ

ValidateDatabase メソッド

UL 拡張：現在のデータベースの検証を実行します。

オーバーロードリスト

名前	説明
ValidateDatabase(ULDBValid) メソッド	**UL 拡張：**現在のデータベースの検証を実行します。
ValidateDatabase(ULDBValid, string) メソッド	**UL 拡張：**現在のデータベースの検証を実行します。

ValidateDatabase(ULDBValid) メソッド

UL 拡張：現在のデータベースの検証を実行します。

Visual Basic 構文

```
Public Sub ValidateDatabase(ByVal how As ULDBValid)
```

C# 構文

```
public void ValidateDatabase(ULDBValid how)
```

パラメーター

- **how**　データベースを検証する方法を記述します。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDatabaseManager.ValidateDatabase メソッド [Ultra Light.NET]226 ページ
- ULDBValid 列挙体 [Ultra Light.NET]459 ページ

例

次のコードでは、現在のデータベースを検証します。

```
' Visual Basic
conn.ValidateDatabase( iAnywhere.Data.UltraLite.ULVF_INDEX )
```

対応する C# 言語のコードを次に示します。

```
// C#
conn.ValidateDatabase( iAnywhere.Data.UltraLite.ULVF_INDEX )
```

ValidateDatabase(ULDBValid, string) メソッド

UL 拡張：現在のデータベースの検証を実行します。

Visual Basic 構文

```
Public Sub ValidateDatabase(
     ByVal how As ULDBValid,
    ByVal tableName As String
)
```

C# 構文

```
public void ValidateDatabase(ULDBValid how, string tableName)
```

パラメーター

- **how**　データベースを検証する方法を記述します。
- **tableName**　NULL (Visual Basic の場合は Nothing) の場合はデータベース全体を検証し、それ以外の場合は指定されたテーブルだけを検証します。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDatabaseManager.ValidateDatabase メソッド [Ultra Light.NET]226 ページ
- ULDBValid 列挙体 [Ultra Light.NET]459 ページ

例

次のコードでは、現在のデータベースを検証します。

```
' Visual Basic
conn.ValidateDatabase( iAnywhere.Data.UltraLite.ULVF_INDEX, Nothing )
```

対応する C# 言語のコードを次に示します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
// C#
conn.ValidateDatabase( iAnywhere.Data.UltraLite.ULVF_INDEX, null )
```

ConnectionString プロパティ

Ultra Light.NET データベースへの接続を開くためのパラメーターを指定します。

Visual Basic 構文

```
Public Overrides Property ConnectionString As String
```

C# 構文

```
public override string ConnectionString {get;set;}
```

例外

- **InvalidOperationException**　接続が開いている間は、値を設定できません。
- **ArgumentException**　指定された接続文字列が無効です。

備考

接続文字列は、ULConnectionParms オブジェクトを使用して指定できます。

この接続を開くためのパラメーターは、キーワードと値の組み合わせがセミコロンで区切られたリストにする必要があります。デフォルトは、空の文字列 (無効な接続文字列) です。

UL 拡張： Ultra Light.NET によって使用されるパラメーターは Ultra Light データベースに固有であるため、その接続文字列は SQL Anywhere の接続文字列との互換性がありません。

パラメーター値は、単一引用符または二重引用符で囲むことができますが、引用符で囲まれた部分に同じ種類の引用符が含まれないことが条件になります。値にセミコロンが含まれるか、値が引用符で始まるか、値の前後にスペースが必要な場合は、値を引用符で囲む必要があります。

パラメーター値を引用符で囲まない場合は、値にセミコロンが含まれていないこと、また値が単一引用符と二重引用符のいずれかで始まることを確認してください。値の前後のスペースは無視されます。

デフォルトでは、UID=DBA と PWD=sql で接続が確立されます。データベースをより強力に保護するには、ユーザー名 DBA のパスワードを変更するか、新しいユーザーを作成して (GrantConnectTo メソッドを使用)、DBA ユーザーを削除します (RevokeConnectFrom を使用)。

参照

- ULConnection.Open メソッド [Ultra Light.NET]145 ページ
- ULConnectionParms クラス [Ultra Light.NET]164 ページ
- ULConnection.GrantConnectTo メソッド [Ultra Light.NET]145 ページ
- 「Ultra Light 接続パラメーター」『Ultra Light データベース管理とリファレンス』

例

次のコードでは、Windows Mobile デバイス上のデータベース ¥UltraLite¥MyDatabase.udb への接続を作成して開きます。

```
' 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()
```

対応する C# 言語のコードを次に示します。

```
// C#
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnDevice = ".udb";
ULConnection conn = new ULConnection();
conn.ConnectionString = openParms.ToString();
conn.Open();
```

ConnectionTimeout プロパティ

この機能は Ultra Light.NET ではサポートされていません。

Visual Basic 構文

```
Public ReadOnly Overrides Property ConnectionTimeout As Integer
```

C# 構文

```
public override int ConnectionTimeout {get;}
```

例外

- **ULException クラス**　値を設定することは、Ultra Light.NET ではサポートされていません。

備考

値は常に 0 です。

Database プロパティ

接続を開くデータベースの名前を返します。

Visual Basic 構文

```
Public ReadOnly Overrides Property Database As String
```

C# 構文

```
public override string Database {get;}
```


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

データベースの名前が含まれる文字列。

Windows Mobile デバイスでは、ULConnection オブジェクトは dbn、ce_file の順に接続文字列を参照します。

デスクトップマシンでは、ULConnection オブジェクトは dbn、nt_file の順に接続文字列を参照します。

DatabaseID プロパティ

UL 拡張： グローバルオートインクリメントのカラムに使用するデータベース ID の値を指定します。

Visual Basic 構文

```
Public Property DatabaseID As Long
```

C# 構文

```
public long DatabaseID {get;set;}
```

例外

- **ULException クラス**　指定された新しいデータベース ID が無効です。

備考

現在のデータベースのデータベース ID の値。

データベース ID の値は、[0,System.UInt32.MaxValue] の範囲内であることが必要です。ULConnection.INVALID_DATABASE_ID の値は、現在のデータベースにデータベース ID が設定されていないことを示すのに使用されます。

参照

- ULDatabaseSchema.GetDatabaseProperty メソッド [Ultra Light.NET]228 ページ
- ULDatabaseSchema.SetDatabaseOption メソッド [Ultra Light.NET]232 ページ
- ULConnection.INVALID_DATABASE_ID フィールド [Ultra Light.NET]163 ページ
- System.UInt32.MaxValue

DataSource プロパティ

この機能は Ultra Light.NET ではサポートされていません。

Visual Basic 構文

```
Public ReadOnly Overrides Property DataSource As String
```

C# 構文

```
public override string DataSource {get;}
```

備考

値は常に空の文字列です。

GlobalAutoIncrementUsage プロパティ

UL 拡張： 利用可能なグローバルオートインクリメントの値の使用済み比率 (%) を返します。

Visual Basic 構文

```
Public ReadOnly Property GlobalAutoIncrementUsage As Short
```

C# 構文

```
public short GlobalAutoIncrementUsage {get;}
```

例外

- **ULException クラス** SQL エラーが発生しました。

備考

使用可能なグローバルオートインクリメントの値の使用済み比率 (%)。これは、0 ～ 100 の範囲内の整数です。

比率が 100% に近づいたら、アプリケーションが ULConnection.DatabaseID 値を使用して、新しいグローバルデータベース ID を設定します。

参照

- ULDatabaseManager クラス [Ultra Light.NET]220 ページ
- ULConnection.DatabaseID プロパティ [Ultra Light.NET]157 ページ

LastIdentity プロパティ

UL 拡張： 直前に使用した identity の値を返します。

Visual Basic 構文

```
Public ReadOnly Property LastIdentity As ULong
```

C# 構文

```
public ulong LastIdentity {get;}
```

例外

- **ULException クラス** SQL エラーが発生しました。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

直前に使用した identity の値 (符号なし long)。

直前に使用した identity の値です。このプロパティは、次の SQL Anywhere 文と同義です。

```
SELECT @identity
```

LastIdentity プロパティは、グローバルオートインクリメントカラムで使うと特に便利です。

このプロパティでは最後に割り当てられたデフォルト値がわかるだけなので、間違った結果を取らないために INSERT 文を実行した直後にこの値を取り出してください。

ときには、1 つの INSERT 文にグローバルオートインクリメント型のカラムが複数含まれていることがあります。この場合、LastIdentity プロパティは、生成されたデフォルト値の 1 つですが、その値の生成元のカラムを判別する信頼できる方法はありません。このため、このような状況を回避するようなデータベースの設計と INSERT 文の記述を行ってください。

Schema プロパティ

UL 拡張： この接続に関連付けられている現在のデータベースのスキーマへのアクセスに使用します。

Visual Basic 構文

```
Public ReadOnly Property Schema As ULDatabaseSchema
```

C# 構文

```
public ULDatabaseSchema Schema {get;}
```

備考

この接続が開かれているデータベースのスキーマを表す ULDatabaseSchema オブジェクトへの参照。

このプロパティが有効なのは、接続が開かれている間だけです。

参照

- ULDatabaseSchema クラス [Ultra Light.NET]227 ページ

ServerVersion プロパティ

この機能は Ultra Light.NET ではサポートされていません。

Visual Basic 構文

```
Public ReadOnly Overrides Property ServerVersion As String
```

C# 構文

```
public override string ServerVersion {get;}
```

備考

値は常に空の文字列です。

State プロパティ

接続の現在のステータスを返します。

Visual Basic 構文

```
Public ReadOnly Overrides Property State As ConnectionState
```

C# 構文

```
public override ConnectionState State {get;}
```

備考

接続が開いている場合は System.Data.ConnectionState.Open、閉じている場合は System.Data.ConnectionState.Closed を返します。

参照

- ULConnection.StateChange event [Ultra Light.NET]162 ページ
- System.Data.ConnectionState

SyncParms プロパティ

UL 拡張： この接続の同期設定を指定します。

Visual Basic 構文

```
Public ReadOnly Property SyncParms As ULSyncParms
```

C# 構文

```
public ULSyncParms SyncParms {get;}
```

備考

この接続によって同期に使用されるパラメーターを表す ULSyncParms オブジェクトへの参照。パラメーターを修正すると、この接続で行われる次の同期に反映されます。

参照

- ULConnection.Synchronize メソッド [Ultra Light.NET]151 ページ
- ULConnection.SyncResult プロパティ [Ultra Light.NET]161 ページ
- ULSyncParms クラス [Ultra Light.NET]389 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

SyncResult プロパティ

UL 拡張： この接続の最後の同期結果を返します。

Visual Basic 構文

```
Public ReadOnly Property SyncResult As ULSyncResult
```

C# 構文

```
public ULSyncResult SyncResult {get;}
```

備考

この接続の最後の同期結果を表す ULSyncResult オブジェクトへの参照。

参照

- ULConnection.Synchronize メソッド [Ultra Light.NET]151 ページ
- ULConnection.SyncParms プロパティ [Ultra Light.NET]160 ページ
- ULConnection.SyncResult プロパティ [Ultra Light.NET]161 ページ

InfoMessage イベント

Ultra Light.NET が、この接続の警告または情報メッセージを送信するときに発生します。

Visual Basic 構文

```
Public Event InfoMessage As ULInfoMessageEventHandler
```

C# 構文

```
public event ULInfoMessageEventHandler InfoMessage;
```

備考

Ultra Light.NET の警告または情報メッセージを処理するには、ULInfoMessageEventHandler デリゲートを作成し、このイベントにアタッチする必要があります。

参照

- ULInfoMessageEventHandler デリゲート [Ultra Light.NET]454 ページ

例

次のコードでは、情報メッセージのイベントハンドラーが定義されます。

```
' Visual Basic
Private Sub MyInfoMessageHandler( _
    obj As Object, args As ULInfoMessageEventArgs _
)
    System.Console.WriteLine( _
        "InfoMessageHandler: " + args.NativeError + ", " _
        + args.Message _
    )
End Sub
```

対応する C# 言語のコードを次に示します。

```
// C#
private void MyInfoMessageHandler(
    object obj, ULInfoMessageEventArgs args
)
{
    System.Console.WriteLine(
        "InfoMessageHandler: " + args.NativeError + ", "
        + args.Message
    );
}
```

次のコードでは、MyInfoMessageHandler メソッドが conn という接続に追加されます。

```
' Visual Basic
AddHandler conn.InfoMessage, AddressOf MyInfoMessageHandler
```

対応する C# 言語のコードを次に示します。

```
// C#
conn.InfoMessage +=
    new ULInfoMessageEventHandler(MyInfoMessageHandler);
```

StateChange イベント

この接続のステータスが変更されると発生します。

Visual Basic 構文

```
Public Event StateChange As StateChangeEventHandler
```

C# 構文

```
public event override StateChangeEventHandler StateChange;
```

備考

ステータス変更メッセージを処理するには、System.Data.StateChangeEventHandler デリゲートを作成し、このイベントにアタッチする必要があります。

参照

- System.Data.StateChangeEventHandler

例

次のコードでは、ステータス変更のイベントハンドラーが定義されます。

```
' Visual Basic
Private Sub MyStateHandler( _
    obj As Object, args As StateChangeEventArgs _
)
    System.Console.WriteLine( _
        "StateHandler: " + args.OriginalState + " to " _
        + args.CurrentState _
    )
End Sub
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

対応する C# 言語のコードを次に示します。

```
// C#
private void MyStateHandler(
    object obj, StateChangeEventArgs args
)
{
    System.Console.WriteLine(
        "StateHandler: " + args.OriginalState + " to "
        + args.CurrentState
    );
}
```

次のコードでは、MyStateHandler が conn という接続に追加されます。

```
' Visual Basic
AddHandler conn.StateChange, AddressOf MyStateHandler
```

対応する C# 言語のコードを次に示します。

```
// C#
conn.StateChange += new StateChangeEventHandler(MyStateHandler);
```

INVALID_DATABASE_ID フィールド

UL 拡張： ULConnection.DatabaseID プロパティが設定されていないことを示すデータベース ID の定数です。

Visual Basic 構文

```
Public Const INVALID_DATABASE_ID As Long
```

C# 構文

```
public const long INVALID_DATABASE_ID;
```

参照

- ULConnection.DatabaseID プロパティ [Ultra Light.NET]157 ページ

SYNC_ALL_DB フィールド

データベース全体を表す空のパブリケーションリストです。

Visual Basic 構文

```
Public Const SYNC_ALL_DB As String
```

C# 構文

```
public const String SYNC_ALL_DB;
```

SYNC_ALL_PUBS フィールド

パブリケーション名 "*" はすべてのパブリケーションを表します。

Visual Basic 構文

```
Public Const SYNC_ALL_PUBS As String
```

C# 構文

```
public const String SYNC_ALL_PUBS;
```

ULConnectionParms クラス

UL 拡張： Ultra Light データベースへの接続を開く接続文字列を作成します。

Visual Basic 構文

```
Public Class ULConnectionParms Inherits System.ComponentModel.Component
```

C# 構文

```
public class ULConnectionParms : System.ComponentModel.Component
```

基本クラス

- System.ComponentModel.Component

メンバー

継承されたメンバーを含む ULConnectionParms クラスのすべてのメンバー。

名前	説明
ULConnectionParms コンストラクター	ULConnectionParms インスタンスを、そのデフォルト値で初期化します。
Dispose method (System.ComponentModel.Component から継承)	System.ComponentModel.Component で使用されるすべてのリソースを解放します。
Finalize method (System.ComponentModel.Component から継承)	System.ComponentModel.Component がガベージコレクションによって回収される前に、アンマネージリソースを解放し、その他のクリーンアップ操作を行います。
GetService method (System.ComponentModel.Component から継承)	System.ComponentModel.Component またはその System.ComponentModel.Container により提供されるサービスを表すオブジェクトを返します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
ToString メソッド	このインスタンスの文字列表現を返します。
AdditionalParms プロパティ	「名前=値」のペアをセミコロンで区切ったリストで、追加のパラメーターを指定します。
CacheSize プロパティ	キャッシュサイズを指定します。
CanRaiseEvents property (System.ComponentModel.Component から継承)	コンポーネントでイベントを発生させることができるかどうかを示す値を取得します。
ConnectionName プロパティ	接続の名前を指定します。
Container property (System.ComponentModel.Component から継承)	System.ComponentModel.Component を含む System.ComponentModel.IContainer を取得します。
DatabaseOnDesktop プロパティ	Windows デスクトッププラットフォーム上の Ultra Light データベースのパスとファイル名を指定します。
DatabaseOnDevice プロパティ	Windows Mobile 上の Ultra Light データベースのパスとファイル名を指定します。
DesignMode property (System.ComponentModel.Component から継承)	System.ComponentModel.Component が現在設計モードになっているかどうかを示す値を取得します。
EncryptionKey プロパティ	データベースを暗号化するためのキーを指定します。
Disposed (System.ComponentModel.Component から継承) この System.ComponentModel.Component にアタッチされているイベントハンドラーのリストを取得します。 Events property (System.ComponentModel.Component から継承)	System.ComponentModel.Component.Dispose メソッドの呼び出しによってコンポーネントが破棄されたときに発生します。
Password プロパティ	認証済みユーザーのパスワードを指定します。

名前	説明
Site property (System.ComponentModel.Component から継承)	System.ComponentModel.Component の System.ComponentModel.ISite を取得または設定します。
UserID プロパティ	データベースで認証されるユーザーを指定します。

備考

頻繁に使用する接続パラメーターは、ULConnectionParms オブジェクトの個々のプロパティです。

ULConnectionParms オブジェクトは、接続を開く (ULConnection.Open メソッドで)、またはデータベースを削除する (ULDatabaseManager.DropDatabase メソッドで) パラメーターを指定するために使用します。

前後のスペースは、すべての値で無視されます。値には、前後のスペースやセミコロンを含めることはできません。また、一重引用符または二重引用符で始めることはできません。

接続文字列を作成するときは、データベースを識別し、オプションの接続設定を指定する必要があります。ULConnectionParms オブジェクトで適切なプロパティを設定して、接続パラメーターをすべて指定したら、ULConnectionParms.ToString メソッドを使用して接続文字列を作成します。作成した文字列は、ULConnection(String) コンストラクターで新しい ULConnection オブジェクトを作成したり、既存の ULConnection オブジェクトの ULConnection.ConnectionString プロパティを設定したりするのに使用されます。

「データベースの識別」

各インスタンスには、データベースへのプラットフォーム固有のパスがあります。実行されているプラットフォームに対応する値だけが使用されます。たとえば、次のコードでは、パス ¥UltraLite¥mydb1.udb は Windows Mobile で使用され、mydb2.udb はその他のプラットフォームで使用されます。

```
' Visual Basic
Dim dbName As ULConnectionParms = new ULConnectionParms
dbName.DatabaseOnDevice = "¥UltraLite¥mydb1.udb"
dbName.DatabaseOnDesktop = "somedir¥mydb2.udb"
```

対応する C# 言語のコードを次に示します。

```
// C#
ULConnectionParms dbName = new ULConnectionParms();
dbName.DatabaseOnDevice = "¥¥UltraLite¥¥mydb1.udb";
dbName.DatabaseOnDesktop = "somedir¥mydb2.udb";
```

Ultra Light データベースファイルに推奨されるファイル拡張子は .udb です。Windows Mobile デバイスでは、デフォルトのデータベースは ¥UltraLiteDB¥ulstore.udb です。その他の Windows プラットフォームでは、デフォルトのデータベースは ulstore.udb です。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

複数のデータベースを使用している場合は、各データベースのデータベース名を指定する必要があります。

「オプションの接続設定」

アプリケーションでの必要性やデータベースの作成方法によっては、デフォルト以外の ULConnectionParms.UserID 値、デフォルト以外の ULConnectionParms.Password 値、データベースの ULConnectionParms.EncryptionKey 値、接続の ULConnectionParms.CacheSize 値を指定することが必要になる場合があります。複数の接続を使用するアプリケーションでは、各接続に対してユニークな ULConnectionParms.ConnectionName 値を指定しなければなりません。

データベースは、1 人の認証済みユーザー DBA で作成されます。このユーザーの最初のパスワードは sql です。デフォルトでは、ユーザー ID DBA とパスワード sql を使用して、接続が開かれます。デフォルトのユーザーを無効にするには、ULConnection.RevokeConnectFrom メソッドを使用します。ユーザーを追加したりユーザーのパスワードを変更するには、ULConnection.GrantConnectTo メソッドを使用します。

データベースを作成したときに暗号化キーを指定した場合は、そのデータベースへの以後の接続では、すべて同じ暗号化キーを使用する必要があります。データベースの暗号化キーを変更するには、ULConnection.ChangeEncryptionKey メソッドを使用します。

参照

- ULConnection.Open メソッド [Ultra Light.NET]145 ページ
- ULDatabaseManager.DropDatabase メソッド [Ultra Light.NET]223 ページ
- ULConnectionParms.ToString メソッド [Ultra Light.NET]168 ページ
- ULConnection クラス [Ultra Light.NET]116 ページ
- ULConnection.ConnectionString プロパティ [Ultra Light.NET]155 ページ
- ULConnectionParms.UserID プロパティ [Ultra Light.NET]174 ページ
- ULConnectionParms.Password プロパティ [Ultra Light.NET]174 ページ
- ULConnectionParms.EncryptionKey プロパティ [Ultra Light.NET]173 ページ
- ULConnectionParms.CacheSize プロパティ [Ultra Light.NET]171 ページ
- ULConnectionParms.ConnectionName プロパティ [Ultra Light.NET]172 ページ
- ULConnectionParms.AdditionalParms プロパティ [Ultra Light.NET]168 ページ
- ULConnection.RevokeConnectFrom メソッド [Ultra Light.NET]147 ページ
- ULConnection.GrantConnectTo メソッド [Ultra Light.NET]145 ページ
- ULConnection.ChangeEncryptionKey メソッド [Ultra Light.NET]128 ページ
- 「Ultra Light 接続パラメーター」『Ultra Light データベース管理とリファレンス』

ULConnectionParms コンストラクター

ULConnectionParms インスタンスを、そのデフォルト値で初期化します。

Visual Basic 構文

```
Public Sub New()
```

C# 構文

```
public ULConnectionParms()
```

ToString メソッド

このインスタンスの文字列表記を返します。

Visual Basic 構文

```
Public Overrides Function ToString() As String
```

C# 構文

```
public override string ToString()
```

戻り値

「キーワード=値」の組み合わせがセミコロンで区切られたリスト形式の、このインスタンスの文字列表現。

AdditionalParms プロパティ

「名前=値」のペアをセミコロンで区切ったリストで、追加のパラメーターを指定します。

Visual Basic 構文

```
Public Property AdditionalParms As String
```

C# 構文

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

例外

- **ArgumentException**　値に無効な接続文字列が含まれていました。

備考

ここで指定されるのは、使用頻度の低いパラメーターです。

「キーワード=値」をセミコロンで区切ったリストによる追加のパラメーター。「キーワード=値」リストの値は、ULConnection.ConnectionString のルールに従う必要があります。デフォルトは NULL 参照 (Visual Basic の Nothing) です。

ページサイズや予約サイズのパラメーターの値は、バイト単位で指定します。キロバイトの単位を示すにはサフィックス k または K を使用し、メガバイトの単位を示すにはサフィックス m または M を使用します。

追加のパラメーターを次に示します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

キーワード	説明
dbn	接続する必要のある、ロードしたデータベースを識別します。データベースが起動されると、データベース名が割り当てられます。データベース名は、dbn パラメーターを使用して明示的に割り当てられるか、または拡張子とパスが削除されたベースファイル名を使用して Ultra Light によって割り当てられます。接続が開かれると、Ultra Light は dbn 値が一致する実行中のデータベースをまず検索します。一致するデータベースが見つからない場合、Ultra Light は適切なデータベースのファイル名のパラメーター (DatabaseOnDevice プロパティまたは DatabaseOnDesktop プロパティを持つ) を使用して、新しいデータベースを起動します。このパラメーターが必要なのは、同じベースファイル名を持つ 2 つの異なるデータベースにアプリケーション (または Ultra Light エンジン) がアクセスする場合です。このパラメーターが使用されるのは、ULConnection.Open メソッドを使用して接続を開いたときだけです。

キーワード	説明
reserve_size	Ultra Light の永続的データの保管に使用する、ファイルシステム領域を予約します。reserve_size パラメーターを使用すると、データを挿入することなく、Ultra Light データベースが必要とするファイルシステム領域を事前に割り付けることができます。ファイルシステムの領域を予約すると、パフォーマンスが多少向上し、メモリが不足するという障害を防ぐことができます。デフォルトでは、永続ストアファイルのサイズは、アプリケーションがデータベースを更新して、サイズを大きくする必要が生じた場合にだけ大きくなります。reserve_size パラメーターで予約されるファイルシステム領域には、未加工データだけでなく、永続ストアファイルのメタデータも含まれることに注意してください。データベースのデータ量から、必要なファイルシステム領域を計算する場合は、メタデータのオーバーヘッドとデータの圧縮を考慮してください。reserve_size パラメーターは、起動時に永続ストアファイルを設定された予約サイズまで大きくすることにより、領域を予約します。これは、そのファイルが以前に存在していたかどうかに関係なく行われます。このファイルはトランケートされません。パラメーター文字列 createParms.AdditionalParms = "reserve_size=2m" で、起動時に継続保管ファイルが少なくとも 2 MB であることが保証されます。このパラメーターは、ULConnection.Open メソッドで接続開くときにのみ使用されます。
start	ロケーションを指定して、Ultra Light エンジンを開始します。現在実行中でないエンジンに接続する場合は、StartLine (START) 接続パラメーターだけを指定します。ロケーションの指定は、Ultra Light エンジンがシステム パスに登録されていない場合にのみ必要です。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULDatabaseManager.RuntimeType プロパティ [Ultra Light.NET]227 ページ
- ULConnection.ConnectionString プロパティ [Ultra Light.NET]155 ページ
- ULConnectionParms.DatabaseOnDevice プロパティ [Ultra Light.NET]172 ページ
- ULConnectionParms.DatabaseOnDesktop プロパティ [Ultra Light.NET]172 ページ
- ULConnection.Open メソッド [Ultra Light.NET]145 ページ
- 「Ultra Light 接続パラメーター」『Ultra Light データベース管理とリファレンス』

CacheSize プロパティ

キャッシュサイズを指定します。

Visual Basic 構文

```
Public Property CacheSize As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

キャッシュサイズを指定する文字列。デフォルトは NULL 参照 (Visual Basic の Nothing) です。この場合、デフォルトの 16 ページが使用されます。

キャッシュサイズの値は、バイト単位で指定します。キロバイトの単位を示すにはサフィックス k または K を使用し、メガバイトの単位を示すにはサフィックス m または M を使用します。

たとえば、次の例ではキャッシュサイズが 128 KB に設定されます。

```
connParms.CacheSize = "128k"
```

デフォルトのキャッシュサイズは 16 ページです。デフォルトのページサイズは 4 KB なので、デフォルトのキャッシュサイズは 64 KB です。キャッシュの最小サイズは、プラットフォームによって異なります。

デフォルトのキャッシュサイズは小さめの値です。テスト結果により、パフォーマンスの向上が必要と判断した場合は、キャッシュサイズを大きくしてください。

キャッシュサイズをデータベース自体のサイズよりも大きくすると、パフォーマンスは向上しません。また、キャッシュサイズが大きいと、その他の使用可能なアプリケーションの数が少なくなることがあります。

キャッシュサイズが未指定であるか、正しく指定されていない場合は、デフォルトのサイズが使用されます。

ConnectionName プロパティ

接続の名前を指定します。

Visual Basic 構文

```
Public Property ConnectionName As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

接続名が必要となるのは、データベースとの接続を複数作成する場合だけです。

接続の名前を指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

DatabaseOnDesktop プロパティ

Windows デスクトッププラットフォーム上の Ultra Light データベースのパスとファイル名を指定します。

Visual Basic 構文

```
Public Property DatabaseOnDesktop As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

データベースへの絶対パスまたは相対パスを指定する文字列。値が NULL 参照 (Visual Basic の Nothing) の場合、データベース ulstore.udb が使用されます。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

DatabaseOnDevice プロパティ

Windows Mobile 上の Ultra Light データベースのパスとファイル名を指定します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public Property DatabaseOnDevice As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

データベースへのフルパスを指定する文字列。値が NULL 参照 (Visual Basic の Nothing) の場合、データベース ¥UltraLiteDB¥ulstore.udb が使用されます。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

EncryptionKey プロパティ

データベースを暗号化するためのキーを指定します。

Visual Basic 構文

```
Public Property EncryptionKey As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

暗号化キーを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) で、暗号化は行われません。

すべての接続では、データベースが作成されたときに指定されたキーと同じキーを使用する必要があります。キーを忘れた場合はデータベースにまったくアクセスできなくなります。

すべてのパスワードに共通することですが、最善の方法は、簡単には推測できないキー値を選択することです。キーの長さは任意ですが、短いと推測されやすいため、一般的には長い方が適しています。数字、文字、特殊文字を組み合わせると、キーは推測されにくくなります。

参照

- ULConnection.ChangeEncryptionKey メソッド [Ultra Light.NET]128 ページ

Password プロパティ

認証済みユーザーのパスワードを指定します。

Visual Basic 構文

```
Public Property Password As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

データベースのユーザー ID を指定する文字列。デフォルトは NULL 参照 (Visual Basic の Nothing) です。

パスワードでは大文字と小文字が区別されます。

データベースが作成されると、ユーザー ID DBA のパスワードは SQL に設定されます。

参照

- ULConnectionParms.UserID プロパティ [Ultra Light.NET]174 ページ

UserID プロパティ

データベースで認証されるユーザーを指定します。

Visual Basic 構文

```
Public Property UserID As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

データベースのユーザー ID を指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

ユーザー ID の大文字と小文字は区別されません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

データベースは最初、DBA という名前の 1 人の認証済みユーザーで作成されます。

ユーザー ID とパスワードを両方とも入力しないと、ユーザー名 DBA とパスワード SQL が使用されます。データベースをより強力に保護するには、ユーザー名 DBA のパスワードを変更するか、新しいユーザーを作成して (ULConnection.GrantConnectTo メソッドで)、DBA ユーザーを削除します (ULConnection.RevokeConnectFrom メソッドで)。

参照

- ULConnectionParms.Password プロパティ [Ultra Light.NET]174 ページ
- ULConnection.GrantConnectTo メソッド [Ultra Light.NET]145 ページ
- ULConnection.RevokeConnectFrom メソッド [Ultra Light.NET]147 ページ

ULConnectionStringBuilder クラス

Ultra Light データベースへの接続を開く接続文字列を作成します。

Visual Basic 構文

```
Public NotInheritable Class ULConnectionStringBuilder
    Inherits System.Data.Common.DbConnectionStringBuilder
```

C# 構文

```
public sealed class ULConnectionStringBuilder :
    System.Data.Common.DbConnectionStringBuilder
```

基本クラス

- System.Data.Common.DbConnectionStringBuilder

メンバー

継承されたメンバーを含む ULConnectionStringBuilder クラスのすべてのメンバー。

名前	説明
ULConnectionStringBuilder コンストラクター	ULConnectionStringBuilder オブジェクトを、そのデフォルト値で初期化します。
Add method (System.Data.Common.DbConnectionStringBuilder から継承)	指定されたキーと値を持つエントリを System.Data.Common.DbConnectionStringBuilder に追加します。
AppendKeyValuePair method (System.Data.Common.DbConnectionStringBuilder から継承)	キーと値を既存の System.Text.StringBuilder オブジェクトに追加する十分で安全な方法を提供します。

名前	説明
Clear method (System.Data.Common.DbConnectionStringBuilder から継承)	System.Data.Common.DbConnectionStringBuilder インスタンスの内容をクリアします。
ClearPropertyDescriptors method (System.Data.Common.DbConnectionStringBuilder から継承)	関連する System.Data.Common.DbConnectionStringBuilder の System.ComponentModel.PropertyDescriptor オブジェクトのコレクションをクリアします。
ContainsKey メソッド	ULConnectionStringBuilder オブジェクトに特定のキーワードが含まれているかどうかを判断します。
EquivalentTo メソッド	この ULConnectionStringBuilder オブジェクト内の接続情報と指定された DbConnectionStringBuilder オブジェクト内の接続情報を比較します。
GetProperties method (System.Data.Common.DbConnectionStringBuilder から継承)	提供される System.Collections.Hashtable にこの System.Data.Common.DbConnectionStringBuilder のすべてのプロパティに関する情報を格納します。
GetShortName メソッド	指定されたキーワードの短縮バージョンを取得します。
Remove メソッド	指定されたキーが設定されたエントリを ULConnectionStringBuilder オブジェクトから削除します。
ShouldSerialize method (System.Data.Common.DbConnectionStringBuilder から継承)	指定されたキーが、この System.Data.Common.DbConnectionStringBuilder インスタンスに存在するかどうかを示します。
ToString method (System.Data.Common.DbConnectionStringBuilder から継承)	この System.Data.Common.DbConnectionStringBuilder に関連付けられた接続文字列を返します。
TryGetValue メソッド	入力されたキーに対応する値を、この ULConnectionStringBuilder オブジェクトから取り出します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
BrowsableConnectionString property (System.Data.Common.DbConnectionStringBuilder から継承)	Visual Studio デザイナーで System.Data.Common.DbConnectionStringBuilder.ConnectionString プロパティを表示するかどうかを示す値を取得または設定します。
CacheSize プロパティ	**UL 拡張：** キャッシュのサイズを指定します。
ConnectionName プロパティ	接続の名前を指定します。
ConnectionString property (System.Data.Common.DbConnectionStringBuilder から継承)	System.Data.Common.DbConnectionStringBuilder に関連付けられた接続文字列を取得または設定します。
Count property (System.Data.Common.DbConnectionStringBuilder から継承)	System.Data.Common.DbConnectionStringBuilder.ConnectionString プロパティに含まれているキーの現在の数を取得します。
DatabaseKey プロパティ	データベースを暗号化するためのキーを指定します。
DatabaseName プロパティ	データベース名、または接続する必要のあるロードしたデータベースの名前を指定します。
DatabaseOnDesktop プロパティ	**UL 拡張：** Windows デスクトッププラットフォーム上の Ultra Light データベースのパスとファイル名を指定します。
DatabaseOnDevice プロパティ	**UL 拡張：** Windows Mobile 上の Ultra Light データベースのパスとファイル名を指定します。
IsFixedSize property (System.Data.Common.DbConnectionStringBuilder から継承)	System.Data.Common.DbConnectionStringBuilder のサイズが固定かどうかを示す値を取得します。
IsReadOnly property (System.Data.Common.DbConnectionStringBuilder から継承)	System.Data.Common.DbConnectionStringBuilder が読み込み専用かどうかを示す値を取得します。
Keys property (System.Data.Common.DbConnectionStringBuilder から継承)	System.Data.Common.DbConnectionStringBuilder にキーが含まれている System.Collections.ICollection を取得します。

名前	説明
OrderedTableScans プロパティ	ORDER BY 句を指定しない SQL クエリで、デフォルトで順序付けされたテーブルスキャンを実行するかどうかを指定します。
Password プロパティ	認証済みユーザーのパスワードを指定します。
ReserveSize プロパティ	**UL 拡張：** Ultra Light の永続的データの保管に使用する予約ファイルシステム領域を指定します。
StartLine プロパティ	ロケーションを指定して、Ultra Light エンジンを開始します。
this プロパティ	指定された接続キーワードの値を指定します。
UserID プロパティ	データベースで認証されるユーザーを指定します。
Values property (System.Data.Common.DbConnectionStringBuilder から継承)	System.Data.Common.DbConnectionStringBuilder に値が含まれている System.Collections.ICollection を取得します。

備考

頻繁に使用する接続パラメーターは、ULConnectionStringBuilder オブジェクトの個々のプロパティです。

ULConnectionStringBuilder クラスは、.NET Compact Framework 2.0 では使用できません。

ULConnectionStringBuilder オブジェクトは、接続を開く (ULConnection.Open メソッドで)、またはデータベースを削除する (ULDatabaseManager.DropDatabase メソッドで) パラメーターを指定するために使用します。

前後のスペースは、すべての値で無視されます。値には、前後のスペースやセミコロンを含めることはできません。また、一重引用符または二重引用符で始めることはできません。

接続文字列を作成するときは、データベースを識別し、オプションの接続設定を指定する必要があります。ULConnectionStringBuilder オブジェクトで適切なプロパティを設定して、接続パラメーターをすべて指定したら、System.Data.Common.DbConnectionStringBuilder.ConnectionString を使用して接続文字列を作成します。作成した文字列は、ULConnection(String) コンストラクターで新しい ULConnection オブジェクトを作成したり、既存の ULConnection オブジェクトの ULConnection.ConnectionString プロパティを設定したりするのに使用されます。

「データベースの識別」

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

各インスタンスには、データベースへのプラットフォーム固有のパスがあります。実行されているプラットフォームに対応する値だけが使用されます。たとえば、次のコードでは、パス ¥UltraLite¥mydb1.udb は Windows Mobile で使用され、mydb2.udb はその他のプラットフォームで使用されます。

```
' Visual Basic
Dim dbName As ULConnectionStringBuilder = _
    new ULConnectionStringBuilder
dbName.DatabaseOnDevice = "¥UltraLite¥mydb1.udb"
dbName.DatabaseOnDesktop = "somedir¥mydb2.udb"
```

対応する C# 言語のコードを次に示します。

```
// C#
ULConnectionStringBuilder dbName = new ULConnectionStringBuilder();
dbName.DatabaseOnDevice = "¥¥UltraLite¥¥mydb1.udb";
 dbName.DatabaseOnDesktop = @"somedir¥¥mydb2.udb";
```

Ultra Light データベースファイルに推奨されるファイル拡張子は .udb です。Windows Mobile デバイスでは、デフォルトのデータベースは ¥UltraLiteDB¥ulstore.udb です。その他の Windows プラットフォームでは、デフォルトのデータベースは ulstore.udb です。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。

複数のデータベースを使用している場合は、各データベースのデータベース名を指定する必要があります。

「オプションの接続設定」

アプリケーションでの必要性やデータベースの作成方法によっては、デフォルト以外の ULConnectionStringBuilder.UserID 値、デフォルト以外の ULConnectionStringBuilder.Password 値、データベースの ULConnectionStringBuilder.EncryptionKey 値、接続の ULConnectionStringBuilder.CacheSize 値を指定することが必要になる場合があります。複数の接続を使用するアプリケーションでは、各接続に対してユニークな ULConnectionStringBuilder.ConnectionName 値を指定しなければなりません。

データベースは、1 人の認証済みユーザー DBA で作成されます。このユーザーの最初のパスワードは sql です。デフォルトでは、ユーザー ID DBA とパスワード sql を使用して、接続が開かれます。デフォルトのユーザーを無効にするには、ULConnection.RevokeConnectFrom メソッドを呼び出します。ユーザーを追加したりユーザーのパスワードを変更するには、ULConnection.GrantConnectTo メソッドを呼び出します。

データベースを作成したときに暗号化キーを指定した場合は、そのデータベースへの以後の接続では、すべて同じ暗号化キーを使用する必要があります。データベースの暗号化キーを変更するには、ULConnection.ChangeEncryptionKey メソッドを使用します。

参照

- ULConnection.Open メソッド [Ultra Light.NET]145 ページ
- ULDatabaseManager.DropDatabase メソッド [Ultra Light.NET]223 ページ
- ULConnection クラス [Ultra Light.NET]116 ページ
- ULConnection.ConnectionString プロパティ [Ultra Light.NET]155 ページ
- ULConnectionStringBuilder.DatabaseName プロパティ [Ultra Light.NET]185 ページ
- ULConnectionStringBuilder.UserID プロパティ [Ultra Light.NET]191 ページ
- ULConnectionStringBuilder.Password プロパティ [Ultra Light.NET]188 ページ
- ULConnectionStringBuilder.DatabaseKey プロパティ [Ultra Light.NET]185 ページ
- ULConnectionStringBuilder.CacheSize プロパティ [Ultra Light.NET]183 ページ
- ULConnectionStringBuilder.ConnectionName プロパティ [Ultra Light.NET]184 ページ
- ULConnection.RevokeConnectFrom メソッド [Ultra Light.NET]147 ページ
- ULConnection.GrantConnectTo メソッド [Ultra Light.NET]145 ページ
- ULConnection.ChangeEncryptionKey メソッド [Ultra Light.NET]128 ページ
- System.Data.Common.DbConnectionStringBuilder.ConnectionString
- 「Ultra Light 接続パラメーター」『Ultra Light データベース管理とリファレンス』

ULConnectionStringBuilder コンストラクター

ULConnectionStringBuilder オブジェクトを、そのデフォルト値で初期化します。

オーバーロードリスト

名前	説明
ULConnectionStringBuilder() コンストラクター	ULConnectionStringBuilder オブジェクトを、そのデフォルト値で初期化します。
ULConnectionStringBuilder(string) コンストラクター	ULConnectionStringBuilder オブジェクトを、指定された接続文字列で初期化します。

ULConnectionStringBuilder() コンストラクター

ULConnectionStringBuilder オブジェクトを、そのデフォルト値で初期化します。

Visual Basic 構文

```
Public Sub New()
```

C# 構文

```
public ULConnectionStringBuilder()
```

ULConnectionStringBuilder(string) コンストラクター

ULConnectionStringBuilder オブジェクトを、指定された接続文字列で初期化します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

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

C# 構文

```
public ULConnectionStringBuilder(string connectionString)
```

パラメーター

- **connectionString** Ultra Light.NET の接続文字列。接続文字列は、キーワード値の組がセミコロンで区切られたリストです。

ContainsKey メソッド

ULConnectionStringBuilder オブジェクトに特定のキーワードが含まれているかどうかを判断します。

Visual Basic 構文

```
Public Overrides Function ContainsKey(
     ByVal keyword As String
) As Boolean
```

C# 構文

```
public override bool ContainsKey(string keyword)
```

パラメーター

- **keyword** 接続キーワードの名前。

戻り値

指定されたキーワードの値がこの接続文字列ビルダに含まれる場合は True、それ以外の場合は false を返します。

EquivalentTo メソッド

この ULConnectionStringBuilder オブジェクト内の接続情報と指定された DbConnectionStringBuilder オブジェクト内の接続情報を比較します。

Visual Basic 構文

```
Public Overrides Function EquivalentTo(
     ByVal connectionStringBuilder As DbConnectionStringBuilder
) As Boolean
```

C# 構文

```
public override bool EquivalentTo(
     DbConnectionStringBuilder connectionStringBuilder
)
```

パラメーター

- **connectionStringBuilder** この ULConnectionStringBuilder オブジェクトと比較する別の DbConnectionStringBuilder オブジェクト。

戻り値

このオブジェクトが指定された DbConnectionStringBuilder オブジェクトと同じ場合は True。それ以外の場合は false。

参照

- System.Data.Common.DbConnectionStringBuilder

GetShortName メソッド

指定されたキーワードの短縮バージョンを取得します。

Visual Basic 構文

```
Public Shared Function GetShortName(ByVal keyword As String) As String
```

C# 構文

```
public static string GetShortName(string keyword)
```

パラメーター

- **keyword** 取り出す項目のキー。

戻り値

キーワードが認識された場合は、指定されたキーワードの短縮バージョン。その他の場合は NULL。

Remove メソッド

指定されたキーが設定されたエントリを ULConnectionStringBuilder オブジェクトから削除します。

Visual Basic 構文

```
Public Overrides Function Remove(ByVal keyword As String) As Boolean
```

C# 構文

```
public override bool Remove(string keyword)
```

パラメーター

- **keyword** 接続キーワードの名前。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

戻り値

接続文字列内のキーが削除された場合は true、キーが存在しなかった場合は false。

TryGetValue メソッド

入力されたキーに対応する値を、この ULConnectionStringBuilder オブジェクトから取り出します。

Visual Basic 構文

```
Public Overrides Function TryGetValue(
     ByVal keyword As String,
    ByVal value As Object
) As Boolean
```

C# 構文

```
public override bool TryGetValue(string keyword, out Object value)
```

パラメーター

- **keyword** 取り出す項目のキー。
- **value** キーに対応する値。

戻り値

キーワードが接続文字列内にある場合は true、それ以外は false。

備考

TryGetValue メソッドを使用すると、開発者は、安全に ULConnectionStringBuilder から値を取得できます。最初に ContainsKey メソッドを呼び出す必要はありません。TryGetValue メソッドは、存在しないキーを指定してこのメソッドを呼び出しても例外を発行しないため、値を取得する前にキーを検索する必要はありません。存在しないキーを指定して TryGetValue を呼び出すと、値パラメーターに NULL 値 (Visual Basic の Nothing) が設定されます。

CacheSize プロパティ

UL 拡張： キャッシュのサイズを指定します。

Visual Basic 構文

```
Public Property CacheSize As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

キャッシュサイズを指定する文字列。デフォルトは NULL 参照 (Visual Basic の Nothing) です。この場合、デフォルトの 16 ページが使用されます。

キャッシュサイズの値は、バイト単位で指定します。キロバイトの単位を示すにはサフィックス k または K を使用し、メガバイトの単位を示すにはサフィックス m または M を使用します。

たとえば、次の例ではキャッシュサイズが 128 KB に設定されます。

```
connParms.CacheSize = "128k"
```

デフォルトのキャッシュサイズは 16 ページです。デフォルトのページサイズは 4 KB なので、デフォルトのキャッシュサイズは 64 KB です。キャッシュの最小サイズは、プラットフォームによって異なります。

デフォルトのキャッシュサイズは小さめの値です。テスト結果により、パフォーマンスの向上が必要と判断した場合は、キャッシュサイズを大きくしてください。

キャッシュサイズをデータベース自体のサイズよりも大きくすると、パフォーマンスは向上しません。また、キャッシュサイズが大きいと、その他の使用可能なアプリケーションの数が少なくなることがあります。

キャッシュサイズが未指定であるか、正しく指定されていない場合は、デフォルトのサイズが使用されます。

ConnectionName プロパティ

接続の名前を指定します。

Visual Basic 構文

```
Public Property ConnectionName As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

接続名が必要となるのは、データベースとの接続を複数作成する場合だけです。

接続の名前を指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

DatabaseKey プロパティ

データベースを暗号化するためのキーを指定します。

Visual Basic 構文

```
Public Property DatabaseKey As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

暗号化キーを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) で、暗号化は行われません。

すべての接続では、データベースが作成されたときに指定されたキーと同じキーを使用する必要があります。キーを忘れた場合はデータベースにまったくアクセスできなくなります。

すべてのパスワードに共通することですが、最善の方法は、簡単には推測できないキー値を選択することです。キーの長さは任意ですが、短いと推測されやすいため、一般的には長い方が適しています。数字、文字、特殊文字を組み合わせると、キーは推測されにくくなります。

参照

- ULConnection.ChangeEncryptionKey メソッド [Ultra Light.NET]128 ページ

DatabaseName プロパティ

データベース名、または接続する必要のあるロードしたデータベースの名前を指定します。

Visual Basic 構文

```
Public Property DatabaseName As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

データベースの名前を指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

データベースが起動されると、データベース名が割り当てられます。データベース名は、dbn パラメーターを使用して明示的に割り当てられるか、または拡張子とパスが削除されたベースファイル名を使用して Ultra Light によって割り当てられます。

接続が開かれると、Ultra Light は dbn パラメーターが一致する実行中のデータベースをまず検索します。一致するデータベースが見つからない場合、Ultra Light は適切なデータベースのファイル名のパラメーター (DatabaseOnDevice プロパティまたは DatabaseOnDesktop プロパティ) を使用して、新しいデータベースを起動します。

このパラメーターが必要なのは、同じベースファイル名を持つ 2 つの異なるデータベースにアプリケーション (または Ultra Light エンジン) がアクセスする場合です。

参照

- ULConnectionStringBuilder.DatabaseOnDevice プロパティ [Ultra Light.NET]186 ページ
- ULConnectionStringBuilder.DatabaseOnDesktop プロパティ [Ultra Light.NET]186 ページ

DatabaseOnDesktop プロパティ

UL 拡張： Windows デスクトッププラットフォーム上の Ultra Light データベースのパスとファイル名を指定します。

Visual Basic 構文

```
Public Property DatabaseOnDesktop As String
```

C# 構文

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

例外

- **ArgumentException** 値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

データベースへの絶対パスまたは相対パスを指定する文字列。値が NULL 参照 (Visual Basic の Nothing) の場合、データベース ulstore.udb が使用されます。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

DatabaseOnDevice プロパティ

UL 拡張： Windows Mobile 上の Ultra Light データベースのパスとファイル名を指定します。

Visual Basic 構文

```
Public Property DatabaseOnDevice As String
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

データベースへのフルパスを指定する文字列。値が NULL 参照 (Visual Basic の Nothing) の場合、データベース ¥UltraLiteDB¥ulstore.udb が使用されます。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

OrderedTableScans プロパティ

ORDER BY 句を指定しない SQL クエリで、デフォルトで順序付けされたテーブルスキャンを実行するかどうかを指定します。

Visual Basic 構文

```
Public Property OrderedTableScans As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

順序付けされたテーブルスキャンを実行するかどうかを指定する boolean 文字列。たとえば、true と false、yes と no、1 と 0 などです。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

Ultra Light で動的 SQL を使用するとき、クエリ実行の順序は重要ではありません。Ultra Light は、プライマリキーインデックスを使用するのではなく、データベースページからローに直接アクセスします。これにより、ローをフェッチするパフォーマンスが改善されました。この最適化を使用するには、クエリが読み込み専用であり、すべてのローをスキャンする必要があります。

ローが特定の順序に並べることを期待する場合は、SQL クエリに ORDER BY 文を指定します。ただし、一部のアプリケーションでは、プライマリキーの順にローを返すようなデフォルトの動作に依存する可能性があります。このような場合は、OrderedTableScans パラメーターを 1 (true、yes、on) に設定し、テーブル上で反復するときに以前の動作に戻すようにすることが必要です。

OrderedTableScans 値を 1 (true、yes、on) に設定しても、ORDER BY 句を指定しなかったり、クエリがインデックスを利用できなかったりすると、Ultra Light はプライマリキーを使用してデフォルトの動作を実行します。

Password プロパティ

認証済みユーザーのパスワードを指定します。

Visual Basic 構文

```
Public Property Password As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

データベースのユーザー ID を指定する文字列。デフォルトは NULL 参照 (Visual Basic の Nothing) です。

パスワードでは大文字と小文字が区別されます。

データベースが作成されると、ユーザー ID DBA のパスワードは SQL に設定されます。

参照

- ULConnectionStringBuilder.UserID プロパティ [Ultra Light.NET]191 ページ

ReserveSize プロパティ

UL 拡張： Ultra Light の永続的データの保管に使用する予約ファイルシステム領域を指定します。

Visual Basic 構文

```
Public Property ReserveSize As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

予約サイズを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

予約サイズパラメーターの値は、バイト単位で指定します。キロバイトの単位を示すにはサフィックス k または K を使用し、メガバイトの単位を示すにはサフィックス m または M を使用します。

reserve_size パラメーターを使用すると、データを挿入することなく、Ultra Light データベースが必要とするファイルシステム領域を事前に割り付けることができます。ファイルシステムの領域を予約すると、パフォーマンスが多少向上し、メモリが不足するという障害を防ぐことができます。デフォルトでは、永続ストアファイルのサイズは、アプリケーションがデータベースを更新して、サイズを大きくする必要が生じた場合にだけ大きくなります。

reserve_size で予約されるファイルシステム領域には、未加工データだけでなく、永続ストアファイルのメタデータも含まれることに注意してください。データベースのデータ量から、必要なファイルシステム領域を計算する場合は、メタデータのオーバーヘッドとデータの圧縮を考慮してください。

reserve_size パラメーターは、起動時に永続ストアファイルを設定された予約サイズまで大きくすることにより、領域を予約します。これは、そのファイルが以前に存在していたかどうかに関係なく行われます。このファイルはトランケートされません。

次のパラメーター文字列では、起動時に永続ストアファイルのサイズが最低 2 MB 確保されます。

```
connParms.ReserveSize = "2m"
```

StartLine プロパティ

ロケーションを指定して、Ultra Light エンジンを開始します。

Visual Basic 構文

```
Public Property StartLine As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

Ultra Light エンジンの実行可能ファイルのロケーションを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

現在実行中でないエンジンに接続する場合は、StartLine (START) 接続パラメーターだけを指定します。

参照

- ULDatabaseManager.RuntimeType プロパティ [Ultra Light.NET]227 ページ

this プロパティ

指定された接続キーワードの値を指定します。

Visual Basic 構文

```
Public Overrides Property Item(ByVal keyword As String) As Object
```

C# 構文

```
public override object this[string keyword] {get;set;}
```

パラメーター

- **keyword**　接続キーワードの名前。

備考

指定された接続キーワードの値を表すオブジェクト。

接続キーワードと、ULConnectionStringBuilder クラスの対応するプロパティを次の表に示します。

キーワード	対応するプロパティ
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

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULConnectionStringBuilder.CacheSize プロパティ [Ultra Light.NET]183 ページ
- ULConnectionStringBuilder.DatabaseOnDevice プロパティ [Ultra Light.NET]186 ページ
- ULConnectionStringBuilder.ConnectionName プロパティ [Ultra Light.NET]184 ページ
- ULConnectionStringBuilder.DatabaseKey プロパティ [Ultra Light.NET]185 ページ
- ULConnectionStringBuilder.DatabaseName プロパティ [Ultra Light.NET]185 ページ
- ULConnectionStringBuilder.DatabaseOnDesktop プロパティ [Ultra Light.NET]186 ページ
- ULConnectionStringBuilder.Password プロパティ [Ultra Light.NET]188 ページ
- ULConnectionStringBuilder.ReserveSize プロパティ [Ultra Light.NET]188 ページ
- ULConnectionStringBuilder.StartLine プロパティ [Ultra Light.NET]189 ページ
- ULConnectionStringBuilder.UserID プロパティ [Ultra Light.NET]191 ページ

UserID プロパティ

データベースで認証されるユーザーを指定します。

Visual Basic 構文

```
Public Property UserID As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

データベースのユーザー ID を指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

ユーザー ID では大文字と小文字が区別されません。

データベースは最初、DBA という名前の 1 人の認証済みユーザーで作成されます。

ユーザー ID とパスワードを両方とも入力しないと、ユーザー名 DBA とパスワード SQL が使用されます。データベースをより強力に保護するには、ユーザー名 DBA のパスワードを変更するか、新しいユーザーを作成して (ULConnection.GrantConnectTo メソッドで)、DBA ユーザーを削除します (ULConnection.RevokeConnectFrom メソッドで)。

参照

- ULConnectionStringBuilder.Password プロパティ [Ultra Light.NET]188 ページ
- ULConnection.GrantConnectTo メソッド [Ultra Light.NET]145 ページ
- ULConnection.RevokeConnectFrom メソッド [Ultra Light.NET]147 ページ

ULCreateParms クラス

UL 拡張： Ultra Light データベースを作成するときの作成時オプションの文字列を作成します。

Visual Basic 構文

```
Public Class ULCreateParms
```

C# 構文

```
public class ULCreateParms
```

メンバー

継承されたメンバーを含む ULCreateParms クラスのすべてのメンバー。

名前	説明
ULCreateParms コンストラクター	ULCreateParms オブジェクトを、そのデフォルト値で初期化します。
ToString メソッド	このインスタンスの文字列表記を返します。
CaseSensitive プロパティ	文字列を比較するときに、新しいデータベースで大文字と小文字を区別するかどうかを指定します。
ChecksumLevel プロパティ	新しいデータベースで有効にするデータベースページのチェックサムのレベルを指定します。
DateFormat プロパティ	新しいデータベースで文字列変換に使用される日付フォーマットを指定します。
DateOrder プロパティ	新しいデータベースで文字列変換に使用される日付順を指定します。
FIPS プロパティ	新しいデータベースで使用する暗号化 (AES_FIPS または AES) を指定します。
MaxHashSize プロパティ	新しいデータベースでインデックスのハッシュに使用するデフォルトの最大バイト数を指定します。
NearestCentury プロパティ	新しいデータベースで文字列変換に使用される最も近い世紀を指定します。
Obfuscate プロパティ	新しいデータベースが難読化を使用してデータベースを暗号化するかどうかを指定します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
PageSize プロパティ	新しいデータベースのページサイズ (バイトまたはキロバイト単位) を指定します。
Precision プロパティ	データベースで文字列変換に使用される浮動小数点の精度を指定します。
Scale プロパティ	新しいデータベースによる文字列変換中に、計算結果が最大の精度でトランケートされるときの小数点以下の最大桁数を指定します。
TimeFormat プロパティ	データベースで文字列変換に使用される時刻フォーマットを指定します。
TimestampFormat プロパティ	データベースで文字列変換に使用されるタイムスタンプのフォーマットを指定します。
TimestampIncrement プロパティ	2 つのユニークなタイムスタンプ間のマイクロ秒 (1,000,000 分の 1 秒) 単位の最小差を指定します。
UTF8Encoding プロパティ	新しいデータベースで使用する文字セット (UTF8 文字セットまたは照合に関連付けられた文字セット) を指定します。

備考

ULCreateParms オブジェクトを使用して、ULDatabaseManager.CreateDatabase メソッドでデータベースを作成するときのパラメーターを指定します。

前後のスペースは、すべての値で無視されます。値には、前後のスペースやセミコロンを含めることはできません。また、一重引用符または二重引用符で始めることはできません。

ULCreateParms オブジェクトで適切なプロパティを設定して、作成パラメーターをすべて指定したら、ULCreateParms.ToString メソッドを使用して作成パラメーター文字列を作成します。結果として返される文字列は ULDatabaseManager.CreateDatabase メソッドの createParms パラメーターとして使用できます。

参照

- ULDatabaseSchema.GetDatabaseProperty メソッド [Ultra Light.NET]228 ページ
- ULDatabaseManager.CreateDatabase メソッド [Ultra Light.NET]221 ページ
- ULCreateParms.ToString メソッド [Ultra Light.NET]194 ページ
- 「Ultra Light 接続パラメーター」『Ultra Light データベース管理とリファレンス』

例

次のコードでは、Windows Mobile デバイス上にデータベース ¥UltraLite¥MyDatabase.udb を作成します。このデータベースは、大文字と小文字を区別し、UTF8 文字セットを使用して作成されます。

```
' 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()
```

対応する C# 言語のコードを次に示します。

```
// 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 コンストラクター

ULCreateParms オブジェクトを、そのデフォルト値で初期化します。

Visual Basic 構文

```
Public Sub New()
```

C# 構文

```
public ULCreateParms()
```

ToString メソッド

このインスタンスの文字列表記を返します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public Overrides Function ToString() As String
```

C# 構文

```
public override string ToString()
```

戻り値

「キーワード=値」の組み合わせがセミコロンで区切られたリスト形式の、このインスタンスの文字列表現。

CaseSensitive プロパティ

文字列を比較するときに、新しいデータベースで大文字と小文字を区別するかどうかを指定します。

Visual Basic 構文

```
Public Property CaseSensitive As Boolean
```

C# 構文

```
public bool CaseSensitive {get;set;}
```

備考

データベースで大文字と小文字が区別される場合は true、区別されない場合は false。デフォルトは false です。

このメソッドは、文字列データの比較とソートの方法にのみ影響します。テーブル名、カラム名、インデックス名、接続ユーザー ID などのデータベース識別子については、どのような場合であっても、大文字と小文字は区別されません。接続用のパスワードとデータベース暗号化キーでは、常に、大文字と小文字が区別されます。

ChecksumLevel プロパティ

新しいデータベースで有効にするデータベースページのチェックサムのレベルを指定します。

Visual Basic 構文

```
Public Property ChecksumLevel As Integer
```

C# 構文

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

例外

- **ArgumentException**　値は無効です。

備考

チェックサムのレベルを指定する整数。有効な値は 0、1、2 です。デフォルトは 0 です。

DateFormat プロパティ

新しいデータベースで文字列変換に使用される日付フォーマットを指定します。

Visual Basic 構文

```
Public Property DateFormat As String
```

C# 構文

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

例外

- **ArgumentException** 値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

日付フォーマットを指定する文字列。値が NULL 参照 (Visual Basic の Nothing) の場合、データベースでは "YYYY-MM-DD" が使用されます。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

DateOrder プロパティ

新しいデータベースで文字列変換に使用される日付順を指定します。

Visual Basic 構文

```
Public Property DateOrder As ULDateOrder
```

C# 構文

```
public ULDateOrder DateOrder {get;set;}
```

備考

文字列変換に使用される日付順を指定する ULDateOrder 値。デフォルト値は YMD です。

参照

- ULDateOrder 列挙体 [Ultra Light.NET]459 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

FIPS プロパティ

新しいデータベースで使用する暗号化 (AES_FIPS または AES) を指定します。

Visual Basic 構文

```
Public Property FIPS As Boolean
```

C# 構文

```
public bool FIPS {get;set;}
```

備考

データベースで AES_FIPS を使用して暗号化される場合は true、AES を使用して暗号化される場合は false。デフォルトは false です。

新しいデータベースが作成されたときは、EncryptionKey 接続パラメーターの値を指定して暗号化を有効にする必要があります。FIPS が true で、暗号化キーを指定していない場合、ULDatabaseManager.CreateDatabase メソッドは、暗号化キーが見つからないために失敗します。

参照

- ULConnectionParms.EncryptionKey プロパティ [Ultra Light.NET]173 ページ
- ULDatabaseManager.CreateDatabase メソッド [Ultra Light.NET]221 ページ

MaxHashSize プロパティ

新しいデータベースでインデックスのハッシュに使用するデフォルトの最大バイト数を指定します。

Visual Basic 構文

```
Public Property MaxHashSize As Integer
```

C# 構文

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

例外

- **ArgumentException**　値は無効です。

備考

最大ハッシュサイズを指定する整数。値は、[0.32] の範囲内であることが必要です。デフォルトは 8 です。

NearestCentury プロパティ

新しいデータベースで文字列変換に使用される最も近い世紀を指定します。

Visual Basic 構文

```
Public Property NearestCentury As Integer
```

C# 構文

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

例外

- **ArgumentException**　値は無効です。

備考

最も近い世紀を指定する整数。値は、[0.100] の範囲内であることが必要です。デフォルトは 50 です。

Obfuscate プロパティ

新しいデータベースが難読化を使用してデータベースを暗号化するかどうかを指定します。

Visual Basic 構文

```
Public Property Obfuscate As Boolean
```

C# 構文

```
public bool Obfuscate {get;set;}
```

備考

データベースで難読化を使用して暗号化される場合は true、難読化されない場合は false。デフォルトは false です。

FIPS 暗号化が ULCreateParms.FIPS プロパティで有効になっている場合は、このオプションは無視されます。難読化が有効になっている場合に、新しいデータベースの作成時に EncryptionKey 接続パラメーターに値を指定すると、暗号化キーは無視されます。

参照

- ULCreateParms.FIPS プロパティ [Ultra Light.NET]197 ページ

PageSize プロパティ

新しいデータベースのページサイズ (バイトまたはキロバイト単位) を指定します。

Visual Basic 構文

```
Public Property PageSize As Integer
```

C# 構文

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

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

例外

- **ArgumentException**　値は無効です。

備考

ページサイズ (バイト単位) を指定する整数。有効な値は、1024 (1K)、2048 (2K)、4096 (4K)、8192 (8K)、16384 (16K) です。デフォルトは 4096 です。

Precision プロパティ

データベースで文字列変換に使用される浮動小数点の精度を指定します。

Visual Basic 構文

```
Public Property Precision As Integer
```

C# 構文

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

例外

- **ArgumentException**　値は無効です。

備考

精度を指定する整数。値は、[1,127] の範囲内であることが必要です。デフォルトは 30 です。

参照

- ULCreateParms.Scale プロパティ [Ultra Light.NET]199 ページ

Scale プロパティ

新しいデータベースによる文字列変換中に、計算結果が最大の精度でトランケートされるときの小数点以下の最大桁数を指定します。

Visual Basic 構文

```
Public Property Scale As Integer
```

C# 構文

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

例外

- **ArgumentException**　値は無効です。

備考

位取りを指定する整数。値は、[0.127] の範囲内であることが必要です。デフォルトは 6 です。

Scale 値は Precision 値以下である必要があります。それ以外の場合は、データベースの作成時にエラーが発生します。

TimeFormat プロパティ

データベースで文字列変換に使用される時刻フォーマットを指定します。

Visual Basic 構文

```
Public Property TimeFormat As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

時刻フォーマットを指定する文字列。値が NULL 参照 (Visual Basic の Nothing) の場合、データベースでは "HH:NN:SS.SSS" が使用されます。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

TimestampFormat プロパティ

データベースで文字列変換に使用されるタイムスタンプのフォーマットを指定します。

Visual Basic 構文

```
Public Property TimestampFormat As String
```

C# 構文

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

例外

- **ArgumentException**　値には、セミコロンが含まれています。または、一重引用符または二重引用符で始まります。

備考

タイムスタンプのフォーマットを指定する文字列。値が NULL 参照 (Visual Basic の Nothing) の場合、データベースでは "YYYY-MM-DD HH:NN:SS.SSS" が使用されます。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

TimestampIncrement プロパティ

2 つのユニークなタイムスタンプ間のマイクロ秒 (1,000,000 分の 1 秒) 単位の最小差を指定します。

Visual Basic 構文

```
Public Property TimestampIncrement As Integer
```

C# 構文

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

例外

- **ArgumentException**　値は無効です。

備考

タイムスタンプのインクリメントを指定する整数。値は、[1,60000000] の範囲内であることが必要です。デフォルトは 1 です。

UTF8Encoding プロパティ

新しいデータベースで使用する文字セット (UTF8 文字セットまたは照合に関連付けられた文字セット) を指定します。

Visual Basic 構文

```
Public Property UTF8Encoding As Boolean
```

C# 構文

```
public bool UTF8Encoding {get;set;}
```

備考

データベースで UTF8 文字セットが使用される場合は true、照合に関連付けられた文字セットが使用される場合は false。デフォルトは false です。

照合に関連付けられた文字セット以外の文字セットに文字を格納する場合は、UTF8 文字セットを選択します。たとえば、US ソートを使用したいけれども、現地の綴りで国際住所を格納したいために UTF8Encoding を true に設定している場合は、1252LATIN1 照合を使用してデータベースを作成します。

ULCursorSchema クラス

UL 拡張： Ultra Light.NET カーソルのスキーマを示します。

Visual Basic 構文

```
Public MustInherit Class ULCursorSchema
```

C# 構文

```
public abstract class ULCursorSchema
```

派生クラス

- ULResultSetSchema クラス [Ultra Light.NET]376 ページ
- ULTableSchema クラス [Ultra Light.NET]437 ページ

メンバー

継承されたメンバーを含む ULCursorSchema クラスのすべてのメンバー。

名前	説明
GetColumnID メソッド	指定されたカラムのカラム ID を返します。
GetColumnName メソッド	指定されたカラム ID で識別されたカラムの名前を返します。
GetColumnPrecision メソッド	カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの精度を返します。
GetColumnScale メソッド	カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの位取りを返します。
GetColumnSize メソッド	カラムがサイズ指定されたカラム (BINARY SQL 型または CHAR SQL 型) の場合は、指定されたカラム ID によって識別されたカラムのサイズを返します。
GetColumnSQLName メソッド	指定されたカラム ID で識別されたカラムの名前を返します。
GetColumnULDbType メソッド	指定されたカラム ID によって識別されたカラムの Ultra Light.NET データ型を返します。
GetSchemaTable メソッド	ULDataReader オブジェクトのカラムのスキーマが記述された System.Data.DataTable を返します。
ColumnCount プロパティ	このカーソル内のカラム数を返します。
IsOpen プロパティ	カーソルのスキーマが現在開いているかどうかを確認します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
Name プロパティ	カーソルの名前を返します。

備考

このクラスは、ULTableSchema クラスと ULResultSetSchema クラスの抽象基本クラスです。

注意
iAnywhere.UltraLite ネームスペースから移行するユーザーの場合、カラム ID は、iAnywhere.UltraLite ネームスペース内にあるため、1 ではなく 0 から始まります。

参照

- ULTableSchema クラス [Ultra Light.NET]437 ページ
- ULResultSetSchema クラス [Ultra Light.NET]376 ページ

GetColumnID メソッド

指定されたカラムのカラム ID を返します。

Visual Basic 構文

```
Public Function GetColumnID(ByVal name As String) As Short
```

C# 構文

```
public short GetColumnID(string name)
```

パラメーター

- **name**　カラム名。

戻り値

指定されたカラムのカラム ID。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

カラム ID の範囲は、0 ～ ColumnCount-1 です。

結果セットでは、すべてのカラムに名前があるとは限らないほか、すべてのカラム名がユニークであるとは限りません。エイリアスを使用していない場合は、非計算カラムの名前には、そのカラムの元となるテーブルの名前がプレフィクスとして付けられます。たとえば、MyTable.ID は、クエリ "SELECT ID FROM MyTable" の結果セットに含まれる唯一のカラムの名前です。

カラムの ID とカウントは、スキーマのアップグレード中に変更されることがあります。カラムを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後にリフレッシュします。

参照

- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ

GetColumnName メソッド

指定されたカラム ID で識別されたカラムの名前を返します。

Visual Basic 構文

```
Public Function GetColumnName(ByVal columnID As Integer) As String
```

C# 構文

```
public string GetColumnName(int columnID)
```

パラメーター

- **columnID**　カラムの ID。値は、[0,ColumnCount-1] の範囲内であることが必要です。

戻り値

カラムの名前。カラムに名前がない場合は NULL 参照 (Visual Basic の Nothing)。SQL クエリでカラムのエイリアスが使用されている場合は、そのエイリアスが返されます。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

結果セットでは、すべてのカラムに名前があるとは限らないほか、すべてのカラム名がユニークであるとは限りません。エイリアスを使用していない場合は、非計算カラムの名前には、そのカラムの元となるテーブルの名前がプレフィクスとして付けられます。たとえば、MyTable.ID は、クエリ "SELECT ID FROM MyTable" の結果セットに含まれる唯一のカラムの名前です。

カラムの ID とカウントは、スキーマのアップグレード中に変更されることがあります。カラムを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後にリフレッシュします。

参照

- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

GetColumnPrecision メソッド

カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの精度を返します。

Visual Basic 構文

```
Public Function GetColumnPrecision(ByVal columnID As Integer) As Integer
```

C# 構文

```
public int GetColumnPrecision(int columnID)
```

パラメーター

- **columnID**　カラムの ID。値は、[0,ColumnCount-1] の範囲内であることが必要です。

戻り値

指定された数値カラムの精度。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULCursorSchema.GetColumnULDbType メソッド [Ultra Light.NET]207 ページ
- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ

GetColumnScale メソッド

カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの位取りを返します。

Visual Basic 構文

```
Public Function GetColumnScale(ByVal columnID As Integer) As Integer
```

C# 構文

```
public int GetColumnScale(int columnID)
```

パラメーター

- **columnID**　カラムの ID。値は、[0,ColumnCount-1] の範囲内であることが必要です。

戻り値

指定された数値カラムの位取り。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULCursorSchema.GetColumnULDbType メソッド [Ultra Light.NET]207 ページ
- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ

GetColumnSize メソッド

カラムがサイズ指定されたカラム (BINARY SQL 型または CHAR SQL 型) の場合は、指定されたカラム ID によって識別されたカラムのサイズを返します。

Visual Basic 構文

```
Public Function GetColumnSize(ByVal columnID As Integer) As Integer
```

C# 構文

```
public int GetColumnSize(int columnID)
```

パラメーター

- **columnID**　カラムの ID。値は、[0,ColumnCount-1] の範囲内であることが必要です。

戻り値

指定された、サイズ指定されたカラムのサイズ。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULCursorSchema.GetColumnULDbType メソッド [Ultra Light.NET]207 ページ
- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ

GetColumnSQLName メソッド

指定されたカラム ID で識別されたカラムの名前を返します。

Visual Basic 構文

```
Public Function GetColumnSQLName(ByVal columnID As Integer) As String
```

C# 構文

```
public string GetColumnSQLName(int columnID)
```

パラメーター

- **columnID**　カラムの ID。値は、[0,ColumnCount-1] の範囲内であることが必要です。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

戻り値

カラムの名前。カラムに名前がない場合は NULL 参照 (Visual Basic の Nothing)。SQL クエリでカラムのエイリアスが使用されている場合は、そのエイリアスが返されます。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

結果セットでは、すべてのカラムに名前があるとは限らないほか、すべてのカラム名がユニークであるとは限りません。エイリアスを使用している場合は、カラムの名前がエイリアスになります。

GetColumnSQLName メソッドは、 GetColumnName メソッドとは異なります。これは、GetColumnSQLName メソッドは常に、非エイリアスカラム、非計算カラムの名前 (プレフィクスとしてテーブル名は付きません) のみを返すためです。この動作は、他の ADO.NET プロバイダーの動作によく似ていますが、ユニークでない名前を生成する可能性が高くなります。

カラムの ID とカウントは、スキーマのアップグレード中に変更されることがあります。カラムを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後にリフレッシュします。

参照

- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ
- ULCursorSchema.GetColumnName メソッド [Ultra Light.NET]204 ページ
- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ

GetColumnULDbType メソッド

指定されたカラム ID によって識別されたカラムの Ultra Light.NET データ型を返します。

Visual Basic 構文

```
Public Function GetColumnULDbType(ByVal columnID As Integer) As ULDbType
```

C# 構文

```
public ULDbType GetColumnULDbType(int columnID)
```

パラメーター

- **columnID**　カラムの ID。値は、[0,ColumnCount-1] の範囲内であることが必要です。

戻り値

ULDbType 列挙整数。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ
- ULDbType 列挙体 [Ultra Light.NET]460 ページ

GetSchemaTable メソッド

ULDataReader オブジェクトのカラムのスキーマが記述された System.Data.DataTable を返します。

Visual Basic 構文

```
Public Function GetSchemaTable() As DataTable
```

C# 構文

```
public DataTable GetSchemaTable()
```

戻り値

カラムのスキーマが記述された System.Data.DataTable。

参照

- ULDataReader.GetSchemaTable メソッド [Ultra Light.NET]256 ページ
- ULDataReader クラス [Ultra Light.NET]236 ページ
- System.Data.DataTable

ColumnCount プロパティ

このカーソル内のカラム数を返します。

Visual Basic 構文

```
Public ReadOnly Property ColumnCount As Short
```

C# 構文

```
public short ColumnCount {get;}
```

備考

カーソル内のカラム数。カーソルのスキーマが閉じている場合は 0。

カラム ID の範囲は、0 ～ ColumnCount-1 です。

カラムの ID とカウントは、スキーマのアップグレード中に変更されることがあります。カラムを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後にリフレッシュします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

IsOpen プロパティ

カーソルのスキーマが現在開いているかどうかを確認します。

Visual Basic 構文

```
Public ReadOnly Property IsOpen As Boolean
```

C# 構文

```
public bool IsOpen {get;}
```

備考

カーソルのスキーマが現在開いている場合は true、閉じている場合は false。

Name プロパティ

カーソルの名前を返します。

Visual Basic 構文

```
Public ReadOnly Property Name As String
```

C# 構文

```
public abstract string Name {get;}
```

備考

文字列として返されるカーソル名。

ULDataAdapter クラス

System.Data.DataSet に入力したりデータベースを更新したりするために使用する一連のコマンドとデータベース接続を表します。

Visual Basic 構文

```
Public NotInheritable Class ULDataAdapter
    Inherits System.Data.Common.DbDataAdapter
    Implements System.ICloneable
```

C# 構文

```
public sealed class ULDataAdapter :
    System.Data.Common.DbDataAdapter,
    System.ICloneable
```

基本クラス

- System.Data.Common.DbDataAdapter
- System.ICloneable

メンバー

継承されたメンバーを含む ULDataAdapter クラスのすべてのメンバー。

名前	説明
ULDataAdapter コンストラクター	ULDataAdapter オブジェクトを初期化します。
AddToBatch method (System.Data.Common.DbDataAdapter から継承)	System.Data.IDbCommand を現在のバッチに追加します。
ClearBatch method (System.Data.Common.DbDataAdapter から継承)	バッチからすべての System.Data.IDbCommand オブジェクトを削除します。
CreateRowUpdatedEvent method (System.Data.Common.DbDataAdapter から継承)	System.Data.Common.RowUpdatedEventArgs クラスの新しいインスタンスを初期化します。
CreateRowUpdatingEvent method (System.Data.Common.DbDataAdapter から継承)	System.Data.Common.RowUpdatingEventArgs クラスの新しいインスタンスを初期化します。
Dispose method (System.Data.Common.DbDataAdapter から継承)	System.Data.Common.DbDataAdapter によって使用されるアンマネージリソースを解放するほか、必要に応じてマネージリソースを解放します。
ExecuteBatch method (System.Data.Common.DbDataAdapter から継承)	現在のバッチを実行します。
Fill method (System.Data.Common.DbDataAdapter から継承)	System.Data.DataSet で、ローを追加または再表示します。
FillSchema method (System.Data.Common.DbDataAdapter から継承)	"Table" という名前の System.Data.DataTable を指定された System.Data.DataSet に追加し、指定された System.Data.SchemaType に基づいてスキーマがデータソースのスキーマと一致するように設定します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
GetBatchedParameter method (System.Data.Common.DbDataAdapter から継承)	現在のバッチにあるコマンドの 1 つから System.Data.IDataParameter を返します。
GetBatchedRecordsAffected method (System.Data.Common.DbDataAdapter から継承)	大規模なバッチ更新内の個別の更新試行に関する情報を返します。
GetFillParameters メソッド	SELECT 文の実行時にユーザーによって設定されるパラメーターを返します。
InitializeBatching method (System.Data.Common.DbDataAdapter から継承)	System.Data.Common.DbDataAdapter のためのバッチを初期化します。
FillError (System.Data.Common.DbDataAdapter から継承) OnFillError method (System.Data.Common.DbDataAdapter から継承)	
OnRowUpdated method (System.Data.Common.DbDataAdapter から継承)	.NET Framework データプロバイダーの RowUpdated イベントを発します。
OnRowUpdating method (System.Data.Common.DbDataAdapter から継承)	.NET Framework データプロバイダーの RowUpdating イベントを発します。
TerminateBatching method (System.Data.Common.DbDataAdapter から継承)	System.Data.Common.DbDataAdapter のためのバッチを終了します。
Update method (System.Data.Common.DbDataAdapter から継承)	System.Data.DataRow オブジェクトの指定された配列で挿入、更新、削除されたローに対して、それぞれ INSERT、UPDATE、DELETE 文を呼び出します。
DeleteCommand プロパティ	System.Data.DataSet で削除されたローに該当するデータベース内のローを削除するために、DbDataAdapter.Update メソッドが呼び出されたときにデータベースに対して実行される ULCommand オブジェクトを指定します。

名前	説明
FillCommandBehavior property (System.Data.Common.DbDataAdapter から継承)	データアダプターの入力に使用されるコマンドの動作を取得または設定します。
InsertCommand プロパティ	System.Data.DataSet で挿入されたローに該当するローをデータベースに挿入するために、DbDataAdapter.Update メソッドが呼び出されたときにデータベースに対して実行される ULCommand オブジェクトを指定します。
SelectCommand プロパティ	System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet) メソッド呼び出しまたは System.Data.Common.DbDataAdapter.FillSchema(System.Data.DataSet,System.Data.SchemaType) メソッド呼び出しの実行時に、System.Data.DataSet にコピーするための結果セットをデータベースから取得するために使用される ULCommand を指定します。
TableMappings プロパティ	ソーステーブルと System.Data.DataTable 間のマスタマッピングを提供するコレクションを返します。
UpdateBatchSize property (System.Data.Common.DbDataAdapter から継承)	バッチプロセスサポートを有効または無効にする値を取得または設定し、バッチで実行可能なコマンド数を指定します。
UpdateCommand プロパティ	System.Data.DataSet で更新されたローに該当するデータベース内のローを更新するために、System.Data.Common.DbDataAdapter.Update メソッドが呼び出されたときにデータベースに対して実行される ULCommand オブジェクトを指定します。
RowUpdated イベント	データソースに対してコマンドが実行された後の更新時に発生します。
RowUpdating イベント	データソースに対してコマンドが実行される前の更新時に発生します。
DefaultSourceTableName フィールド (System.Data.Common.DbDataAdapter から継承)	テーブルマッピングのために System.Data.Common.DataAdapter オブジェクトによって使用されるデフォルト名。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

System.Data.DataSet は、Ultra Light データベースとは別に、データをオフラインで処理するための方法を提供します。ULDataAdapter クラスは、System.Data.DataSet を一連の SQL 文に関連付けるためのメソッドを提供します。

Ultra Light はローカルのデータベースであり、Mobile Link には競合解決があるため、ULDataAdapter の使用は制限されています。ほとんどの場合、ULDataReader クラスまたは ULTable クラスを使用すると、より効率的にデータにアクセスできます。

参照

- ULDataReader クラス [Ultra Light.NET]236 ページ
- ULTable クラス [Ultra Light.NET]415 ページ
- System.Data.DataSet
- System.Data.Common.DbDataAdapter
- System.Data.IDbDataAdapter
- System.Data.IDataAdapter
- System.IDisposable

ULDataAdapter コンストラクター

ULDataAdapter オブジェクトを初期化します。

オーバーロードリスト

名前	説明
ULDataAdapter() コンストラクター	ULDataAdapter オブジェクトを初期化します。
ULDataAdapter(string, string) コンストラクター	ULDataAdapter オブジェクトを、指定された SELECT 文と接続文字列で初期化します。
ULDataAdapter(string, ULConnection) コンストラクター	ULDataAdapter オブジェクトを、指定された SELECT 文と接続で初期化します。
ULDataAdapter(ULCommand) コンストラクター	ULDataAdapter オブジェクトを、指定された SELECT 文で初期化します。

ULDataAdapter() コンストラクター

ULDataAdapter オブジェクトを初期化します。

Visual Basic 構文

```
Public Sub New()
```

C# 構文

```
public ULDataAdapter()
```

参照

- ULDataAdapter.ULDataAdapter コンストラクター [Ultra Light.NET]213 ページ

ULDataAdapter(string, string) コンストラクター

ULDataAdapter オブジェクトを、指定された SELECT 文と接続文字列で初期化します。

Visual Basic 構文

```
Public Sub New(
     ByVal selectCommandText As String,
    ByVal selectConnectionString As String
)
```

C# 構文

```
public ULDataAdapter(
     string selectCommandText,
    string selectConnectionString
)
```

パラメーター

- **selectCommandText** ULDataAdapter.SelectCommand メソッドによって使用される SELECT 文。
- **selectConnectionString** Ultra Light.NET データベースの接続文字列。

参照

- ULDataAdapter.ULDataAdapter コンストラクター [Ultra Light.NET]213 ページ
- ULDataAdapter.SelectCommand プロパティ [Ultra Light.NET]217 ページ

ULDataAdapter(string, ULConnection) コンストラクター

ULDataAdapter オブジェクトを、指定された SELECT 文と接続で初期化します。

Visual Basic 構文

```
Public Sub New(
     ByVal selectCommandText As String,
    ByVal selectConnection As ULConnection
)
```

C# 構文

```
public ULDataAdapter(
     string selectCommandText,
    ULConnection selectConnection
)
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

パラメーター

- **selectCommandText**　ULDataAdapter オブジェクトの ULDataAdapter.SelectCommand メソッドによって使用される SELECT 文。
- **selectConnection**　データベースへの接続を定義する ULConnection オブジェクト。

参照

- ULDataAdapter.ULDataAdapter コンストラクター [Ultra Light.NET]213 ページ
- ULDataAdapter.SelectCommand プロパティ [Ultra Light.NET]217 ページ
- ULConnection クラス [Ultra Light.NET]116 ページ

ULDataAdapter(ULCommand) コンストラクター

ULDataAdapter オブジェクトを、指定された SELECT 文で初期化します。

Visual Basic 構文

```
Public Sub New(ByVal selectCommand As ULCommand)
```

C# 構文

```
public ULDataAdapter(ULCommand selectCommand)
```

パラメーター

- **selectCommand**　System.Data.DataSet に配置するためのレコードをデータソースから選択するために System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet) の実行時に使用される ULCommand オブジェクト。

参照

- ULDataAdapter.ULDataAdapter コンストラクター [Ultra Light.NET]213 ページ
- ULCommand クラス [Ultra Light.NET]66 ページ
- System.Data.DataSet

GetFillParameters メソッド

SELECT 文の実行時にユーザーによって設定されるパラメーターを返します。

Visual Basic 構文

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

C# 構文

```
public new ULParameter[] GetFillParameters()
```

戻り値

ユーザーによって設定されたパラメーターが含まれる ULParameter オブジェクトの配列。

備考

これは、System.Data.Common.DbDataAdapter.GetFillParameters メソッドが厳密に型指定されたものです。

参照

- ULParameter クラス [Ultra Light.NET]319 ページ
- System.Data.Common.DbDataAdapter.GetFillParameters

DeleteCommand プロパティ

System.Data.DataSet で削除されたローに該当するデータベース内のローを削除するために、DbDataAdapter.Update メソッドが呼び出されたときにデータベースに対して実行される ULCommand オブジェクトを指定します。

Visual Basic 構文

```
Public Shadows Property DeleteCommand As ULCommand
```

C# 構文

```
public new ULCommand DeleteCommand {get;set;}
```

備考

System.Data.DataSet で削除されたローに該当するデータベース内のローを削除するために実行される ULCommand オブジェクト。

DeleteCommand プロパティが既存の ULCommand オブジェクトに割り当てられる場合、ULCommand オブジェクトのクローンは作成されません。DeleteCommand プロパティは、既存の ULCommand オブジェクトへの参照を保持します。

これは、System.Data.IDbDataAdapter.DeleteCommand プロパティと System.Data.Common.DbDataAdapter.DeleteCommand プロパティが厳密に型指定されたものです。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- System.Data.DataSet
- System.Data.IDbDataAdapter.DeleteCommand
- System.Data.Common.DbDataAdapter.DeleteCommand

InsertCommand プロパティ

System.Data.DataSet で挿入されたローに該当するローをデータベースに挿入するために、DbDataAdapter.Update メソッドが呼び出されたときにデータベースに対して実行される ULCommand オブジェクトを指定します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public Shadows Property InsertCommand As ULCommand
```

C# 構文

```
public new ULCommand InsertCommand {get;set;}
```

備考

System.Data.DataSet で挿入されたローに該当するデータベース内のローを挿入するために実行される ULCommand オブジェクト。

InsertCommand プロパティが既存の ULCommand オブジェクトに割り当てられる場合、ULCommand オブジェクトのクローンは作成されません。InsertCommand プロパティは、既存の ULCommand オブジェクトへの参照を保持します。

これは、System.Data.IDbDataAdapter.InsertCommand プロパティと System.Data.Common.DbDataAdapter.InsertCommand プロパティが厳密に型指定されたものです。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- System.Data.DataSet
- System.Data.IDbDataAdapter.InsertCommand
- System.Data.Common.DbDataAdapter.InsertCommand

SelectCommand プロパティ

System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet) メソッド呼び出しまたは System.Data.Common.DbDataAdapter.FillSchema(System.Data.DataSet,System.Data.SchemaType) メソッド呼び出しの実行時に、System.Data.DataSet にコピーするための結果セットをデータベースから取得するために使用される ULCommand を指定します。

Visual Basic 構文

```
Public Shadows Property SelectCommand As ULCommand
```

C# 構文

```
public new ULCommand SelectCommand {get;set;}
```

備考

System.Data.DataSet を設定するために実行される ULCommand オブジェクト。

SelectCommand プロパティが既存の ULCommand オブジェクトに割り当てられる場合、ULCommand オブジェクトのクローンは作成されません。SelectCommand プロパティは、既存の ULCommand オブジェクトへの参照を保持します。

SelectCommand プロパティがローを返さない場合は、System.Data.DataSet にテーブルが追加されず、例外も発生しません。SELECT 文は、ULDataAdapter(ULCommand)、

ULDataAdapter(String,ULConnection)、ULDataAdapter(String,String) のコンストラクターにも指定できます。

これは、System.Data.IDbDataAdapter.SelectCommand プロパティと System.Data.Common.DbDataAdapter.SelectCommand プロパティが厳密に型指定されたものです。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- ULDataAdapter.ULDataAdapter コンストラクター [Ultra Light.NET]213 ページ
- System.Data.DataSet
- System.Data.IDbDataAdapter.SelectCommand
- System.Data.Common.DbDataAdapter.SelectCommand

TableMappings プロパティ

ソーステーブルと System.Data.DataTable 間のマスタマッピングを提供するコレクションを返します。

Visual Basic 構文

```
Public ReadOnly Shadows Property TableMappings As
DataTableMappingCollection
```

C# 構文

```
public new DataTableMappingCollection TableMappings {get;}
```

備考

ソーステーブルと System.Data.DataTables 間のマスタマッピングを提供する System.Data.Common.DataTableMapping オブジェクトのコレクション。デフォルト値は空のコレクションです。

変更を調整する場合、ULDataAdapter オブジェクトは System.Data.Common.DataTableMappingCollection コレクションを使用して、データソースによって使用されるカラム名を、System.Data.DataSet によって使用されるカラム名に関連付けます。

これは、System.Data.IDataAdapter.TableMappings プロパティが厳密に型指定されたものです。

参照

- System.Data.DataTable
- System.Data.Common.DataTableMapping
- System.Data.DataTable
- System.Data.Common.DataTableMappingCollection
- System.Data.DataSet
- System.Data.IDataAdapter.TableMappings

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

UpdateCommand プロパティ

System.Data.DataSet で更新されたローに該当するデータベース内のローを更新するために、System.Data.Common.DbDataAdapter.Update メソッドが呼び出されたときにデータベースに対して実行される ULCommand オブジェクトを指定します。

Visual Basic 構文

```
Public Shadows Property UpdateCommand As ULCommand
```

C# 構文

```
public new ULCommand UpdateCommand {get;set;}
```

備考

System.Data.DataSet で更新されたローに該当するデータベース内のローを更新するために実行される ULCommand オブジェクト。

UpdateCommand が既存の ULCommand オブジェクトに割り当てられる場合、ULCommand オブジェクトのクローンは作成されません。UpdateCommand プロパティは、既存の ULCommand オブジェクトへの参照を保持します。

このコマンドを実行するとローが返される場合、ULCommand オブジェクトの ULCommand.UpdatedRowSource プロパティの設定方法によっては、これらのローが System.Data.DataSet にマージされることがあります。

これは、System.Data.IDbDataAdapter.UpdateCommand プロパティと System.Data.Common.DbDataAdapter.DeleteCommand プロパティが厳密に型指定されたものです。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- ULCommand.UpdatedRowSource プロパティ [Ultra Light.NET]103 ページ
- System.Data.DataSet
- System.Data.IDbDataAdapter.UpdateCommand
- System.Data.Common.DbDataAdapter.DeleteCommand

RowUpdated イベント

データソースに対してコマンドが実行された後の更新時に発生します。

Visual Basic 構文

```
Public Event RowUpdated As ULRowUpdatedEventHandler
```

C# 構文

```
public event ULRowUpdatedEventHandler RowUpdated;
```

備考

更新が試みられると、イベントが発生します。

ロー更新イベントを処理するには、ULRowUpdatedEventHandler デリゲートを作成し、このイベントにアタッチする必要があります。

参照

- ULRowUpdatedEventHandler デリゲート [Ultra Light.NET]454 ページ

RowUpdating イベント

データソースに対してコマンドが実行される前の更新時に発生します。

Visual Basic 構文

```
Public Event RowUpdating As ULRowUpdatingEventHandler
```

C# 構文

```
public event ULRowUpdatingEventHandler RowUpdating;
```

備考

更新が試みられると、イベントが発生します。

ロー更新イベントを処理するには、ULRowUpdatingEventHandler デリゲートを作成し、このイベントにアタッチする必要があります。

参照

- ULRowUpdatedEventHandler デリゲート [Ultra Light.NET]454 ページ

ULDatabaseManager クラス

UL 拡張： データベースの作成、削除、検証を行う静的メソッドを提供します。

Visual Basic 構文

```
Public NotInheritable Class ULDatabaseManager
```

C# 構文

```
public sealed class ULDatabaseManager
```

メンバー

継承されたメンバーを含む ULDatabaseManager クラスのすべてのメンバー。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
CreateDatabase メソッド	新しい Ultra Light データベースを作成します。
DropDatabase メソッド	指定されたデータベースを削除します。
SetActiveSyncListener メソッド	ActiveSync 用 Mobile Link プロバイダーからの ActiveSync の呼び出しを処理するリスナーオブジェクトを指定します。
SetServerSyncListener メソッド	指定されたサーバー同期メッセージの処理に使用するリスナーオブジェクトを指定します。
SignalSyncIsComplete メソッド	ActiveSync 用の Mobile Link プロバイダーに対して、アプリケーションが同期を完了したことを通知します。
ValidateDatabase メソッド	データベースで低レベルのインデックス検証を実行します。
RuntimeType プロパティ	Ultra Light.NET ランタイムタイプを指定します。

備考

Ultra Light.NET の Ultra Light エンジンのランタイムを使用するには、ULDatabaseManager.RuntimeType プロパティを適切な値に設定してから、他の Ultra Light.NET API を使用します。

例

次の例では、Ultra Light エンジンのランタイムが選択され、接続が作成されます。

```
' Visual Basic
ULDatabaseManager.RuntimeType = ULRuntimeType.UL_ENGINE_CLIENT
Dim conn As ULConnection = new ULConnection
' The RuntimeType is now locked
```

対応する C# 言語のコードを次に示します。

```
// C#
ULDatabaseManager.RuntimeType = ULRuntimeType.UL_ENGINE_CLIENT;
ULConnection conn = new ULConnection();
// The RuntimeType is now locked
```

CreateDatabase メソッド

新しい Ultra Light データベースを作成します。

Visual Basic 構文

```
Public Shared Sub CreateDatabase(
     ByVal connString As String,
    ByVal createParms As String
)
```

C# 構文

```
public static void CreateDatabase(string connString, string createParms)
```

パラメーター

- **connString**　キーワードと値の組み合わせがセミコロンで区切られたリスト形式の、データベースを指定するためのパラメーター。
- **createParms**　キーワードと値の組み合わせがセミコロンで区切られたリスト形式の、新しいデータベースを設定するためのパラメーター。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULConnection.Open メソッド [Ultra Light.NET]145 ページ
- ULConnectionParms クラス [Ultra Light.NET]164 ページ
- ULCreateParms クラス [Ultra Light.NET]192 ページ

例

次のコードでは、Windows Mobile デバイス上にデータベース ¥UltraLite¥MyDatabase.udb を作成し、接続を開きます。

```
' 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()
```

対応する C# 言語のコードを次に示します。

```
// C#
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnDevice = ".udb";

ULConnection.DatabaseManager.CreateDatabase(
   openParms.ToString(),
   ""
);
ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

DropDatabase メソッド

指定されたデータベースを削除します。

Visual Basic 構文

```
Public Shared Sub DropDatabase(ByVal connString As String)
```

C# 構文

```
public static void DropDatabase(string connString)
```

パラメーター

- **connString**　キーワードと値の組み合わせがセミコロンで区切られたリスト形式の、データベースを指定するためのパラメーター。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

接続が開かれているデータベースを削除することはできません。

参照

- ULConnection.Open メソッド [Ultra Light.NET]145 ページ
- ULConnectionParms クラス [Ultra Light.NET]164 ページ

例

次のコードでは、Windows Mobile デバイス上にデータベース ¥UltraLite¥MyDatabase.udb を作成し、接続を開きます。

```
' Visual Basic
Dim connParms As ULConnectionParms = New ULConnectionParms
connParms.DatabaseOnDevice = "¥UltraLite¥MyDatabase.udb"
ULConnection.DatabaseManager.DropDatabase( _
   connParms.ToString() _
)
```

対応する C# 言語のコードを次に示します。

```
// C#
ULConnectionParms connParms = new ULConnectionParms();
connParms.DatabaseOnDevice = ".udb";
ULConnection.DatabaseManager.DropDatabase(
   connParms.ToString()
);
ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```

SetActiveSyncListener メソッド

ActiveSync 用 Mobile Link プロバイダーからの ActiveSync の呼び出しを処理するリスナーオブジェクトを指定します。

Visual Basic 構文

```
Public Shared Sub SetActiveSyncListener(
     ByVal appClassName As String,
    ByVal listener As ULActiveSyncListener
)
```

C# 構文

```
public static void SetActiveSyncListener(
     string appClassName,
    ULActiveSyncListener listener
)
```

パラメーター

- **appClassName** アプリケーションのユニークなクラス名。これは、アプリケーションが ActiveSync で使用されるように登録されているときに使用されるクラス名です。
- **listener** ULActiveSyncListener オブジェクト。前のリスナーを削除するには、NULL (Visual Basic の Nothing) を使用します。

例外

- **ULException クラス** SQL エラーが発生しました。

備考

appClassName パラメーターは、アプリケーションの識別に使用されるユニークな識別子です。アプリケーションは、一度に 1 つの *appClassName* 値のみを使用できます。リスナーが、特定の *appClassName* 値に登録されているときは、別の *appClassName* 値で SetServerSyncListener または SetActiveSyncListener メソッドを呼び出すと失敗します。

ActiveSync リスナーを削除するには、*listener* パラメーターとして null 参照 (Visual Basic の Nothing) を使用して SetActiveSyncListener メソッドを呼び出します。

すべてのリスナーを削除するには、すべてのパラメーターに NULL 参照 (Visual Basic の Nothing) を使用して SetServerSyncListener メソッドを呼び出します。

アプリケーションは、すべてのリスナーを削除してから終了する必要があります。

参照

- ULDatabaseManager.SetServerSyncListener メソッド [Ultra Light.NET]225 ページ
- ULDatabaseManager.SetActiveSyncListener メソッド [Ultra Light.NET]224 ページ
- ULActiveSyncListener インターフェイス [Ultra Light.NET]37 ページ
- ULActiveSyncListener.ActiveSyncInvoked メソッド [Ultra Light.NET]38 ページ

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

SetServerSyncListener メソッド

指定されたサーバー同期メッセージの処理に使用するリスナーオブジェクトを指定します。

Visual Basic 構文

```
Public Shared Sub SetServerSyncListener(
     ByVal messageName As String,
    ByVal appClassName As String,
    ByVal listener As ULServerSyncListener
)
```

C# 構文

```
public static void SetServerSyncListener(
     string messageName,
    string appClassName,
    ULServerSyncListener listener
)
```

パラメーター

- **messageName** メッセージの名前。
- **appClassName** アプリケーションのユニークなクラス名。これは、アプリケーションの識別に使用されるユニークな識別子です。
- **listener** ULServerSyncListener オブジェクト。前のリスナーを削除するには、NULL (Visual Basic の Nothing) を使用します。

例外

- **ULException クラス** SQL エラーが発生しました。

備考

appClassName パラメーターは、アプリケーションの識別に使用されるユニークな識別子です。アプリケーションは、一度に 1 つの *appClassName* 値のみを使用できます。リスナーが、特定の *appClassName* 値に登録されているときは、別の *appClassName* 値で SetServerSyncListener または SetActiveSyncListener メソッドを呼び出すと失敗します。

特定のメッセージのリスナーを削除するには、*listener* パラメーターとして NULL 参照 (Visual Basic の Nothing) を使用して SetServerSyncListener メソッドを呼び出します。

すべてのリスナーを削除するには、すべてのパラメーターに NULL 参照 (Visual Basic の Nothing) を使用して SetServerSyncListener メソッドを呼び出します。

アプリケーションは、すべてのリスナーを削除してから終了する必要があります。

参照

- ULDatabaseManager.SetServerSyncListener メソッド [Ultra Light.NET]225 ページ
- ULDatabaseManager.SetActiveSyncListener メソッド [Ultra Light.NET]224 ページ
- ULServerSyncListener.ServerSyncInvoked メソッド [Ultra Light.NET]386 ページ

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

SignalSyncIsComplete メソッド

ActiveSync 用の Mobile Link プロバイダーに対して、アプリケーションが同期を完了したことを通知します。

Visual Basic 構文

```
Public Shared Sub SignalSyncIsComplete()
```

C# 構文

```
public static void SignalSyncIsComplete()
```

参照

- ULDatabaseManager.SignalSyncIsComplete メソッド [Ultra Light.NET]226 ページ
- ULActiveSyncListener.ActiveSyncInvoked メソッド [Ultra Light.NET]38 ページ

ValidateDatabase メソッド

データベースで低レベルのインデックス検証を実行します。

Visual Basic 構文

```
Public Shared Sub ValidateDatabase(
     ByVal start_parms As String,
    ByVal how As ULDBValid
)
```

C# 構文

```
public static void ValidateDatabase(string start_parms, ULDBValid how)
```

パラメーター

- **start_parms**　キーワードと値の組み合わせがセミコロンで区切られたリスト形式の、データベースを指定するためのパラメーター。
- **how**　データベースを検証する方法を記述します。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULConnection.ValidateDatabase メソッド [Ultra Light.NET]153 ページ
- ULDBValid 列挙体 [Ultra Light.NET]459 ページ
- ULConnectionParms クラス [Ultra Light.NET]164 ページ

例

次のコードでは、Windows Mobile のデータベース ¥UltraLite¥MyDatabase.udb のインデックスを検証します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
' Visual Basic
Dim openParms As ULConnectionParms = New ULConnectionParms
openParms.DatabaseOnDevice = "¥UltraLite¥MyDatabase.udb"
ULConnection.DatabaseManager.ValidateDatabase(_
   openParms.ToString(), iAnywhere.Data.UltraLite.ULVF_INDEX )
```

対応する C# 言語のコードを次に示します。

```
// C#
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnDevice = ".udb";
ULConnection.DatabaseManager.ValidateDatabase(
   openParms.ToString(), iAnywhere.Data.UltraLite.ULVF_INDEX );
```

RuntimeType プロパティ

Ultra Light.NET ランタイムタイプを指定します。

Visual Basic 構文

```
Public Shared Property RuntimeType As ULRuntimeType
```

C# 構文

```
public ULRuntimeType RuntimeType {get;set;}
```

備考

ランタイムタイプを選択してから、その他の Ultra Light.NET API を使用してください。

アンマネージ Ultra Light .NET ランタイムのタイプを示す ULRuntimeType 値。

参照

- ULRuntimeType 列挙体 [Ultra Light.NET]464 ページ

例

次の例では、Ultra Light エンジンのランタイムが選択され、接続が作成されます。

```
' Visual Basic
ULDatabaseManager.RuntimeType = ULRuntimeType.UL_ENGINE_CLIENT
Dim conn As ULConnection = new ULConnection
' The RuntimeType is now locked
```

対応する C# 言語のコードを次に示します。

```
// C#
ULDatabaseManager.RuntimeType = ULRuntimeType.UL_ENGINE_CLIENT;
ULConnection conn = new ULConnection();
// The RuntimeType is now locked
```

ULDatabaseSchema クラス

UL 拡張： Ultra Light.NET データベースのスキーマを示します。

Visual Basic 構文

```
Public NotInheritable Class ULDatabaseSchema
```

C# 構文

```
public sealed class ULDatabaseSchema
```

メンバー

継承されたメンバーを含む ULDatabaseSchema クラスのすべてのメンバー。

名前	説明
GetDatabaseProperty メソッド	指定されたデータベースのプロパティの値を返します。
GetPublicationName メソッド	指定されたパブリケーション ID で識別されたパブリケーションの名前を返します。
GetTableName メソッド	指定されたテーブル ID で識別されたテーブルの名前を返します。
SetDatabaseOption メソッド	指定されたデータベースオプションの値を設定します。
IsCaseSensitive プロパティ	データベースで大文字と小文字が区別されるかどうかをチェックします。
IsOpen プロパティ	データベーススキーマが開いているかどうかを示します。
PublicationCount プロパティ	データベース内のパブリケーションの数です。
TableCount プロパティ	データベース内のテーブルの数です。

備考

このクラスにはコンストラクターがありません。ULDatabaseSchema オブジェクトは、ULConnection.Schema として接続にアタッチされ、接続が開いている間のみ有効です。

参照

- ULDatabaseSchema クラス [Ultra Light.NET]227 ページ
- ULConnection.Schema プロパティ [Ultra Light.NET]159 ページ

GetDatabaseProperty メソッド

指定されたデータベースのプロパティの値を返します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public Function GetDatabaseProperty(ByVal name As String) As String
```

C# 構文

```
public string GetDatabaseProperty(string name)
```

パラメーター

- **name**　値を取得するデータベースプロパティの名前。プロパティ名では大文字と小文字が区別されません。

戻り値

文字列として返されるプロパティの値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

識別されるプロパティは次のとおりです。

プロパティ	説明
CaseSensitive	大文字と小文字の区別のステータス。データベースで大文字と小文字が区別される場合は、ON を返します。それ以外の場合は、OFF を返します。データベースで大文字と小文字が区別されるかどうかは、テーブルのインデックスと結果セットのソート方法に影響します。大文字と小文字の区別は、接続の ULConnectionParms.UserID と ULConnectionParms.Password の検証方法に影響します。ユーザー ID は常に大文字と小文字が区別されません。パスワードは常に大文字と小文字が区別されます。
CharSet	データベースの文字セット。
ChecksumLevel	データベースで有効にするデータベースページのチェックサムのレベル。
Collation	データベースの照合順の名前。
ConnCount	データベースとの接続の数。
date_format	データベースによる文字列変換に使用される日付フォーマット。このフォーマットは、System.DateTime によって使用されるフォーマットと同じであるとは限りません。

プロパティ	説明
date_order	データベースによる文字列変換に使用される日付順。
Encryption	データベースに適用されている暗号化のタイプ。None、Simple、AES、または AES_FIPS を返します。
File	データベースのファイル名。
global_database_id	グローバルオートインクリメントカラムに使用される global_database_id オプションの値。
isolation_level	あるトランザクションの操作が同時に実行される他のトランザクションの操作でも認識される程度を制御するのに使用される isolation_level オプションの値。この値は、接続単位で設定します。
MaxHashSize	インデックスのハッシュに使用する、デフォルトの最大バイト数。このプロパティは、インデックス単位で指定できます。
ml_remote_id	同期中にデータベースを識別するために使用する、ml_remote_id オプションの値。
Name	データベースの名前 (DBN)。
nearest_century	データベースによる文字列変換に使用される最も近い世紀。
PageSize	データベースのページサイズ (バイト)。
precision	データベースによる文字列変換に使用される浮動小数点の精度。
scale	データベースによる文字列変換中に、計算結果が最大精度にトランケートされるときの小数点以下の最大桁数。
time_format	データベースによる文字列変換に使用される時刻フォーマット。このフォーマットは、System.TimeSpan によって使用されるフォーマットと同じであるとは限りません。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

プロパティ	説明
timestamp_format	データベースによる文字列変換に使用されるタイムスタンプのフォーマット。このフォーマットは、System.DateTime によって使用されるフォーマットと同じであるとは限りません。
timestamp_increment	2 つのユニークなタイムスタンプ間のマイクロ秒 (1,000,000 分の 1 秒) 単位の最小差。

参照

- ULDatabaseSchema.SetDatabaseOption メソッド [Ultra Light.NET]232 ページ
- ULConnectionParms.UserID プロパティ [Ultra Light.NET]174 ページ
- ULConnectionParms.Password プロパティ [Ultra Light.NET]174 ページ
- System.TimeSpan
- System.DateTime

GetPublicationName メソッド

指定されたパブリケーション ID で識別されたパブリケーションの名前を返します。

Visual Basic 構文

```
Public Function GetPublicationName(ByVal pubID As Integer) As String
```

C# 構文

```
public string GetPublicationName(int pubID)
```

パラメーター

- **pubID**　パブリケーションの ID。値は、[1,PublicationCount] の範囲内であることが必要です。

戻り値

文字列として返されるパブリケーション名。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

注意
パブリケーション ID とカウントは、スキーマのアップグレード中に変更されることがあります。パブリケーションを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後にリフレッシュします。

参照

- ULDatabaseSchema.PublicationCount プロパティ [Ultra Light.NET]235 ページ

GetTableName メソッド

指定されたテーブル ID で識別されたテーブルの名前を返します。

Visual Basic 構文

```
Public Function GetTableName(ByVal tableID As Integer) As String
```

C# 構文

```
public string GetTableName(int tableID)
```

パラメーター

- **tableID**　テーブルの ID。値は、[1,TableCount] の範囲内であることが必要です。

戻り値

文字列としてのテーブル名。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

テーブル ID は、スキーマのアップグレード中に変更されることがあります。テーブルを正しく識別するには、名前でアクセスするか、キャッシュされている ID をスキーマのアップグレード後にリフレッシュします。

参照

- ULDatabaseSchema.TableCount プロパティ [Ultra Light.NET]236 ページ

SetDatabaseOption メソッド

指定されたデータベースオプションの値を設定します。

Visual Basic 構文

```
Public Sub SetDatabaseOption(
     ByVal name As String,
    ByVal value As String
)
```

C# 構文

```
public void SetDatabaseOption(string name, string value)
```


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

パラメーター

- **name**　データベースオプションの名前。オプション名では大文字と小文字が区別されません。
- **value**　オプションの新しい値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

データベースオプションを指定すると、コミットが実行されます。

トランザクションがアクティブなときにこのメソッドを使用することは、予期しない結果を引き起こす場合があるため、推奨できません。呼び出すと、接続の独立性レベルは変更されますが、ULTransaction.IsolationLevel 値は更新されません。

識別されるオプションは次のとおりです。

オプション	説明
global_database_id	グローバルオートインクリメントカラムに使用する値。値は、[0,System.UInt32.MaxValue] の範囲内であることが必要です。デフォルトは ULConnection.INVALID_DATABASE_ID (現在のデータベースにデータベース ID が設定されていないことを示すのに使用される) です。

オプション	説明
isolation_level	あるトランザクションの操作が、同時に実行される他のトランザクションの操作で認識される程度を制御するのに使用される値。値は "read_uncommitted" または "read_committed" のいずれかです。デフォルトは "read_committed" です。接続の isolation_level を "read_uncommitted" に設定することは、BeginTransaction(System.Data.IsolationLevel.ReadUncommitted) および Commit() 呼び出してその接続におけるすべての操作をラップすることと同じです。同様に、"read_committed" は System.Data.IsolationLevel.ReadCommitted と同じです。現在のトランザクションの独立性レベルを設定する場合は、SetDatabaseOption() ではなく、BeginTransaction(IsolationLevel) を使用してください。Ultra Light での独立性レベルの定義は、ADO.NET のマニュアルの IsolationLevel の説明とは若干異なります。この値は、接続単位で設定します。
ml_remote_id	同期中にデータベースを識別するために使用する値。値として NULL 参照 (Visual Basic の Nothing) を使用して、データベースから ml_remote_id オプションを削除します。

参照

- ULDatabaseSchema.GetDatabaseProperty メソッド [Ultra Light.NET]228 ページ
- ULConnection.INVALID_DATABASE_ID フィールド [Ultra Light.NET]163 ページ
- System.UInt32.MaxValue
- 「独立性レベルの変更」『Ultra Light データベース管理とリファレンス』

IsCaseSensitive プロパティ

データベースで大文字と小文字が区別されるかどうかをチェックします。

Visual Basic 構文

```
Public ReadOnly Property IsCaseSensitive As Boolean
```

C# 構文

```
public bool IsCaseSensitive {get;}
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

データベースで大文字と小文字が区別される場合は true、区別されない場合は false。

データベースで大文字と小文字が区別されるかどうかは、テーブルのインデックスと結果セットのソート方法に影響します。また、大文字と小文字の区別は、ULConnectionParms.UserID と ULConnectionParms.Password の検証方法に影響します。

参照

- ULDatabaseSchema.GetDatabaseProperty メソッド [Ultra Light.NET]228 ページ
- ULConnectionParms.UserID プロパティ [Ultra Light.NET]174 ページ
- ULConnectionParms.Password プロパティ [Ultra Light.NET]174 ページ

IsOpen プロパティ

データベーススキーマが開いているかどうかを示します。

Visual Basic 構文

```
Public ReadOnly Property IsOpen As Boolean
```

C# 構文

```
public bool IsOpen {get;}
```

備考

データベースのスキーマが現在開いている場合は true、現在閉じている場合は false。

ULDatabaseSchema オブジェクトが開いているのは、それがアタッチされている接続が開いている場合だけです。

PublicationCount プロパティ

データベース内のパブリケーションの数です。

Visual Basic 構文

```
Public ReadOnly Property PublicationCount As Integer
```

C# 構文

```
public int PublicationCount {get;}
```

備考

データベース内のパブリケーションの数です。

パブリケーション ID の範囲は、1 ～ PublicationCount です。

注意
パブリケーション ID とカウントは、スキーマのアップグレード中に変更されることがあります。パブリケーションを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後にリフレッシュします。

参照

- ULDatabaseSchema.GetPublicationName メソッド [Ultra Light.NET]231 ページ

TableCount プロパティ

データベース内のテーブルの数です。

Visual Basic 構文

```
Public ReadOnly Property TableCount As Integer
```

C# 構文

```
public int TableCount {get;}
```

備考

データベース内のテーブルの数です。

テーブル ID の範囲は、1 ～ TableCount です。

注意
テーブルの ID とカウントは、スキーマのアップグレード中に変更されることがあります。テーブルを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後にリフレッシュします。

ULDataReader クラス

Ultra Light データベースの読み込み専用の双方向カーソルを表します。

Visual Basic 構文

```
Public Class ULDataReader
    Inherits System.Data.Common.DbDataReader
    Implements System.ComponentModel.IListSource
```

C# 構文

```
public class ULDataReader :
    System.Data.Common.DbDataReader,
    System.ComponentModel.IListSource
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

基本クラス

- System.Data.Common.DbDataReader
- System.ComponentModel.IListSource

派生クラス

- ULResultSet クラス [Ultra Light.NET]352 ページ

メンバー

継承されたメンバーを含む ULDataReader クラスのすべてのメンバー。

名前	説明
Close メソッド	カーソルを閉じます。
Dispose method (System.Data.Common.DbDataReader から継承)	System.Data.Common.DbDataReader クラスの現在のインスタンスで使用しているすべてのリソースを解放します。
GetBoolean メソッド	指定されたカラムの値を System.Boolean として返します。
GetByte メソッド	指定されたカラムの値を符号なし 8 ビット値 (System.Byte) として返します。
GetBytes メソッド	**UL 拡張：** 指定されたカラムの値を System.Bytes 配列として返します。
GetChar メソッド	このメソッドは Ultra Light.NET ではサポートされていません。
GetChars メソッド	指定されたオフセットで始まる、指定された ULDbType.LongVarchar カラムの値のサブセットを、コピー先の System.Char 配列の指定されたオフセットにコピーします。
GetData method (System.Data.Common.DbDataReader から継承)	要求されたカラム序数の System.Data.Common.DbDataReader オブジェクトを返します。
GetDataTypeName メソッド	指定されたカラムのプロバイダーのデータ型の名前を返します。
GetDateTime メソッド	指定されたカラムの値を、ミリ秒の精度の System.DateTime として返します。

名前	説明
GetDbDataReader method (System.Data.Common.DbDataReader から継承)	プロバイダー固有の実装によって無効になる可能性のある、要求されたカラム序数の System.Data.Common.DbDataReader オブジェクトを返します。
GetDecimal メソッド	指定されたカラムの値を System.Decimal として返します。
GetDouble メソッド	指定されたカラムの値を System.Double として返します。
GetEnumerator メソッド	ULDataReader の反復処理を実行する System.Collections.IEnumerator を返します。
GetFieldType メソッド	指定されたカラムに最適な System.Type を返します。
GetFloat メソッド	指定されたカラムの値を System.Single として返します。
GetGuid メソッド	指定されたカラムの値を UUID (System.Guid) として返します。
GetInt16 メソッド	指定されたカラムの値を System.Int16 として返します。
GetInt32 メソッド	指定されたカラムの値を Int32 として返します。
GetInt64 メソッド	指定されたカラムの値を Int64 として返します。
GetName メソッド	指定されたカラムの名前を返します。
GetOrdinal メソッド	指定されたカラムのカラム ID を返します。
GetProviderSpecificFieldType method (System.Data.Common.DbDataReader から継承)	指定されたカラムのプロバイダー固有のフィールドタイプを返します。
GetProviderSpecificValue method (System.Data.Common.DbDataReader から継承)	指定されたカラムの値を System.Object のインスタンスとして取得します。
GetProviderSpecificValues method (System.Data.Common.DbDataReader から継承)	現在のローのコレクション内のプロバイダー固有のすべての属性カラムを取得します。
GetRowCount メソッド	**UL 拡張：** カーソル内のローの数を、スレッショルド以内で返します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
GetSchemaTable メソッド	ULDataReader のカラムのメタデータが記述された System.Data.DataTable を返します。
GetString メソッド	指定されたカラムの値を System.String として返します。
GetTimeSpan メソッド	指定されたカラムの値を、ミリ秒の精度の System.TimeSpan として返します。
GetUInt16 メソッド	指定されたカラムの値を System.UInt16 として返します。
GetUInt32 メソッド	指定されたカラムの値を UInt32 として返します。
GetUInt64 メソッド	指定されたカラムの値を System.UInt64 として返します。
GetValue メソッド	指定されたカラムの値をネイティブフォーマットで返します。
GetValues メソッド	現在のローのすべてのカラム値を返します。
IsDBNull メソッド	指定されたカラムの値が NULL かどうかをチェックします。
MoveAfterLast メソッド	**UL 拡張：** カーソルの最後のローの後に、カーソルを配置します。
MoveBeforeFirst メソッド	**UL 拡張：** カーソルの最初のローの前に、カーソルを配置します。
MoveFirst メソッド	**UL 拡張：** カーソルの最初のローに、カーソルを配置します。
MoveLast メソッド	**UL 拡張：** カーソルの最後のローに、カーソルを配置します。
MoveNext メソッド	**UL 拡張：** カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。
MovePrevious メソッド	**UL 拡張：** カーソルを前のローに配置するか、最初のローの前に配置します。
MoveRelative メソッド	**UL 拡張：** 現在のローを基準としてカーソルを配置します。

名前	説明
NextResult メソッド	バッチ SQL 文の結果を読み込むときに ULDataReader を次の結果に進めます。
Read メソッド	カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。
Depth プロパティ	現在のローのネストの深さを返します。
FieldCount プロパティ	このカーソル内のカラム数を返します。
HasRows プロパティ	ULDataReader に 1 つまたは複数のローがあるかどうかをチェックします。
IsBOF プロパティ	**UL 拡張：** 現在のローの位置が最初のローの前であるかどうかを確認します。
IsClosed プロパティ	カーソルが現在開いているかどうかを確認します。
IsEOF プロパティ	**UL 拡張：** 現在のローの位置が最後のローの後であるかどうかを確認します。
RecordsAffected プロパティ	SQL 文の実行によって変更、挿入、または削除されたローの数を返します。
RowCount プロパティ	**UL 拡張：** カーソル内のロー数を返します。
Schema プロパティ	**UL 拡張：** このカーソルのスキーマを保持します。
this プロパティ	指定されたカラムの値をネイティブフォーマットで返します。
VisibleFieldCount property (System.Data.Common.DbDataReader から継承)	System.Data.Common.DbDataReader の非表示でないフィールドの数を取得します。

備考

カーソルとは、テーブルまたはクエリからの結果セットの一連のローです。

ULDataReader にはコンストラクターがありません。ULDataReader オブジェクトを取得するには、次のように ULCommand を実行します。

```
' Visual Basic
Dim cmd As ULCommand = new ULCommand( _
    "SELECT emp_id FROM employee", conn _
)
Dim reader As ULDataReader = cmd.ExecuteReader()
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

対応する C# 言語のコードを次に示します。

```
// C#
ULCommand cmd = new ULCommand(
   "SELECT emp_id FROM employee", conn
);
ULDataReader reader = cmd.ExecuteReader();
```

UL 拡張： ADO.NET 標準では、結果セットで前方専用の動作だけが必要ですが、ULDataReader は双方向です。ULDataReader の Move メソッドによって、結果セットを移動するときに最大限の柔軟性が得られます。

ULDataReader は、読み込み専用の結果セットです。より柔軟なオブジェクトで結果を操作する必要がある場合は、ULCommand.ExecuteResultSet()、ULCommand.ExecuteTable()、または ULDataAdapter を使用します。ULDataReader は必要に応じてローを取得しますが、ULDataAdapter の場合、結果セットのすべてのローを取得しないと、オブジェクトに対してアクションを実行できません。結果セットのサイズが大きい場合、この違いのために ULDataReader の方が応答時間が速くなります。

UL 拡張： ULDataReader のすべてのカラムは、GetString を使用して取得できます。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- ULCommand.ExecuteResultSet メソッド [Ultra Light.NET]90 ページ
- ULCommand.ExecuteTable メソッド [Ultra Light.NET]94 ページ
- ULDataAdapter クラス [Ultra Light.NET]209 ページ
- ULDataReader.GetString メソッド [Ultra Light.NET]259 ページ
- System.Data.Common.DbDataReader
- System.Data.IDataReader
- System.Data.IDataRecord
- System.IDisposable

Close メソッド

カーソルを閉じます。

Visual Basic 構文

```
Public Overrides Sub Close()
```

C# 構文

```
public override void Close()
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

すでに閉じられているカーソルを閉じるのはエラーではありません。

GetBoolean メソッド

指定されたカラムの値を System.Boolean として返します。

Visual Basic 構文

```
Public Overrides Function GetBoolean(ByVal colID As Integer) As Boolean
```

C# 構文

```
public override bool GetBoolean(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Boolean として返されるカラム値。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Boolean

GetByte メソッド

指定されたカラムの値を符号なし 8 ビット値 (System.Byte) として返します。

Visual Basic 構文

```
Public Overrides Function GetByte(ByVal colID As Integer) As Byte
```

C# 構文

```
public override byte GetByte(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Byte として返されるカラム値。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Byte

GetBytes メソッド

UL 拡張： 指定されたカラムの値を System.Bytes 配列として返します。

オーバーロードリスト

名前	説明
GetBytes(int) メソッド	**UL 拡張：** 指定されたカラムの値を System.Bytes 配列として返します。
GetBytes(int, long, byte[], int, int) メソッド	指定されたオフセットで始まる、指定された ULDbType.LongBinary カラムの値のサブセットを、コピー先の System.Byte 配列の指定されたオフセットにコピーします。

GetBytes(int) メソッド

UL 拡張： 指定されたカラムの値を System.Bytes 配列として返します。

Visual Basic 構文

```
Public Function GetBytes(ByVal colID As Integer) As Byte()
```

C# 構文

```
public byte[] GetBytes(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Bytes 配列として返されるカラム値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

ULDbType.Binary 型、ULDbType.LongBinary 型、ULDbType.UniqueIdentifier 型のカラムの場合にのみ有効です。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.GetBytes メソッド [Ultra Light.NET]243 ページ
- System.Byte

GetBytes(int, long, byte[], int, int) メソッド

指定されたオフセットで始まる、指定された ULDbType.LongBinary カラムの値のサブセットを、コピー先の System.Byte 配列の指定されたオフセットにコピーします。

Visual Basic 構文

```
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# 構文

```
public override long GetBytes(
     int colID,
    long srcOffset,
    byte[] dst,
    int dstOffset,
    int count
)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **srcOffset**　カラム値の開始位置。最初の値は 0 です。
- **dst**　コピー先の配列。
- **dstOffset**　コピー先の配列の開始位置。
- **count**　コピーされるバイト数。

戻り値

実際にコピーされたバイト数。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

NULL 参照 (Visual Basic の Nothing) である *dst* バッファーを渡すと、GetBytes はフィールドの長さをバイト数で返します。

値の *srcOffset* から *srcOffset*+*count*-1 までの位置のバイトが、コピー先の配列の *dstOffset* から *dstOffset*+*count*-1 までの位置に、それぞれコピーされます。*count* のバイト数がコピーされる前に、値の末尾が検出された場合は、コピー先の配列の残りは変更されないままになります。

次のいずれかに該当する場合、コード ULSQLCode.SQLE_INVALID_PARAMETER とともに ULException がスローされ、追加先は修正されません。

●*srcOffset* が負である。

●*dstOffset* が負である。

●*count* が負である。

●*dstOffset* +*count* は *dst* の長さより長い。

その他のエラーの場合は、それに応じたエラーコードとともに ULException がスローされます。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.GetBytes メソッド [Ultra Light.NET]243 ページ
- ULException クラス [Ultra Light.NET]274 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Byte

GetChar メソッド

このメソッドは Ultra Light.NET ではサポートされていません。

Visual Basic 構文

```
Public Overrides Function GetChar(ByVal colID As Integer) As Char
```

C# 構文

```
public override char GetChar(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

このメソッドは Ultra Light.NET ではサポートされていません。

例外

- **ULException クラス**　このメソッドは Ultra Light.NET ではサポートされていません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.GetString メソッド [Ultra Light.NET]259 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ

GetChars メソッド

指定されたオフセットで始まる、指定された ULDbType.LongVarchar カラムの値のサブセットを、コピー先の System.Char 配列の指定されたオフセットにコピーします。

Visual Basic 構文

```
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# 構文

```
public override long GetChars(
     int colID,
    long srcOffset,
    char[] dst,
    int dstOffset,
    int count
)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **srcOffset**　カラム値の開始位置。最初の値は 0 です。
- **dst**　コピー先の配列。
- **dstOffset**　コピー先の配列の開始位置。
- **count**　コピーされる文字数。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

戻り値

実際にコピーされた文字数。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

NULL 参照 (Visual Basic の Nothing) である *dst* バッファーを渡すと、GetChars はフィールドの長さを文字数として返します。

値の *srcOffset* から *srcOffset*+*count*-1 までの位置の文字が、コピー先の配列の *dstOffset* から *dstOffset*+*count*-1 までの位置に、それぞれコピーされます。*count* の文字数がコピーされる前に、値の末尾が検出された場合は、コピー先の配列の残りは変更されないままになります。

次のいずれかに該当する場合、コード ULSQLCode.SQLE_INVALID_PARAMETER とともに ULException がスローされ、追加先は修正されません。

●*srcOffset* が負である。

●*dstOffset* が負である。

●*count* が負である。

●*dstOffset* +*count* は *dst* の長さより長い。

その他のエラーの場合は、それに応じたエラーコードとともに ULException がスローされます。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULException クラス [Ultra Light.NET]274 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Char

GetDataTypeName メソッド

指定されたカラムのプロバイダーのデータ型の名前を返します。

Visual Basic 構文

```
Public Overrides Function GetDataTypeName(
     ByVal colID As Integer
) As String
```

C# 構文

```
public override string GetDataTypeName(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

カラムの ULDbType に対応する文字列。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULCursorSchema.GetColumnULDbType メソッド [Ultra Light.NET]207 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- ULDbType 列挙体 [Ultra Light.NET]460 ページ

GetDateTime メソッド

指定されたカラムの値を、ミリ秒の精度の System.DateTime として返します。

Visual Basic 構文

```
Public Overrides Function GetDateTime(ByVal colID As Integer) As Date
```

C# 構文

```
public override DateTime GetDateTime(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.DateTime として返されるカラム値。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.DateTime

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

GetDecimal メソッド

指定されたカラムの値を System.Decimal として返します。

Visual Basic 構文

```
Public Overrides Function GetDecimal(ByVal colID As Integer) As Decimal
```

C# 構文

```
public override decimal GetDecimal(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Decimal として返されるカラム値。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Decimal

GetDouble メソッド

指定されたカラムの値を System.Double として返します。

Visual Basic 構文

```
Public Overrides Function GetDouble(ByVal colID As Integer) As Double
```

C# 構文

```
public override double GetDouble(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Double として返されるカラム値。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Double

GetEnumerator メソッド

ULDataReader の反復処理を実行する System.Collections.IEnumerator を返します。

Visual Basic 構文

```
Public Overrides Function GetEnumerator()
    As System.Collections.IEnumerator
```

C# 構文

```
public override IEnumerator GetEnumerator()
```

戻り値

ULDataReader の System.Collections.IEnumerator。

参照

- System.Collections.IEnumerator

GetFieldType メソッド

指定されたカラムに最適な System.Type を返します。

Visual Basic 構文

```
Public Overrides Function GetFieldType(ByVal colID As Integer) As Type
```

C# 構文

```
public override Type GetFieldType(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

カラムの System.Type 値。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetDataTypeName メソッド [Ultra Light.NET]247 ページ
- ULCursorSchema.GetColumnULDbType メソッド [Ultra Light.NET]207 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Type

GetFloat メソッド

指定されたカラムの値を System.Single として返します。

Visual Basic 構文

```
Public Overrides Function GetFloat(ByVal colID As Integer) As Single
```

C# 構文

```
public override float GetFloat(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Single として返されるカラム値。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Single

GetGuid メソッド

指定されたカラムの値を UUID (System.Guid) として返します。

Visual Basic 構文

```
Public Overrides Function GetGuid(ByVal colID As Integer) As Guid
```

C# 構文

```
public override Guid GetGuid(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

Guid として返されるカラム値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

このメソッドが有効なのは、ULDbType.UniqueIdentifier 型のカラム、または長さが 16 の ULDbType.Binary 型のカラムの場合だけです。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULCursorSchema.GetColumnULDbType メソッド [Ultra Light.NET]207 ページ
- ULCursorSchema.GetColumnSize メソッド [Ultra Light.NET]206 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Guid

GetInt16 メソッド

指定されたカラムの値を System.Int16 として返します。

Visual Basic 構文

```
Public Overrides Function GetInt16(ByVal colID As Integer) As Short
```

C# 構文

```
public override short GetInt16(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Int16 として返されるカラム値。

例外

- **ULException クラス**　SQL エラーが発生しました。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Int16

GetInt32 メソッド

指定されたカラムの値を Int32 として返します。

Visual Basic 構文

```
Public Overrides Function GetInt32(ByVal colID As Integer) As Integer
```

C# 構文

```
public override int GetInt32(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

Int32 として返されるカラム値。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Int32

GetInt64 メソッド

指定されたカラムの値を Int64 として返します。

Visual Basic 構文

```
Public Overrides Function GetInt64(ByVal colID As Integer) As Long
```

C# 構文

```
public override long GetInt64(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

Int64 として返されるカラム値。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Int64

GetName メソッド

指定されたカラムの名前を返します。

Visual Basic 構文

```
Public Overrides Function GetName(ByVal colID As Integer) As String
```

C# 構文

```
public override string GetName(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

カラムの名前。カラムに名前がない場合は NULL 参照 (Visual Basic の Nothing)。SQL クエリでカラムのエイリアスが使用されている場合は、そのエイリアスが返されます。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

結果セットでは、すべてのカラムに名前があるとは限らないほか、すべてのカラム名がユニークであるとは限りません。エイリアスを使用していない場合は、非計算カラムの名前には、そのカラムの元となるテーブルの名前がプレフィクスとして付けられます。たとえば、MyTable.ID は、クエリ "SELECT ID FROM MyTable" の結果セットに含まれる唯一のカラムの名前です。

このメソッドは、ULCursorSchema.GetColumnName メソッドと同じです。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- ULDataReader.GetSchemaTable メソッド [Ultra Light.NET]256 ページ
- ULCursorSchema.GetColumnName メソッド [Ultra Light.NET]204 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ

GetOrdinal メソッド

指定されたカラムのカラム ID を返します。

Visual Basic 構文

```
Public Overrides Function GetOrdinal(
     ByVal columnName As String
) As Integer
```

C# 構文

```
public override int GetOrdinal(string columnName)
```

パラメーター

- **columnName**　カラム名。

戻り値

指定されたカラムのカラム ID。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

カラム ID の範囲は、0 ～ ULDataReader.FieldCount-1 です。

結果セットでは、すべてのカラムに名前があるとは限らないほか、すべてのカラム名がユニークであるとは限りません。エイリアスを使用していない場合は、非計算カラムの名前には、そのカラムの元となるテーブルの名前がプレフィクスとして付けられます。たとえば、MyTable.ID は、クエリ "SELECT ID FROM MyTable" の結果セットに含まれる唯一のカラムの名前です。

カラムの ID とカウントは、スキーマのアップグレード中に変更されることがあります。カラムを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後にリフレッシュします。

このメソッドは、ULCursorSchema.GetColumnID メソッドと同じです。

参照

- ULDataReader.GetSchemaTable メソッド [Ultra Light.NET]256 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- ULCursorSchema.GetColumnID メソッド [Ultra Light.NET]203 ページ

GetRowCount メソッド

UL 拡張： カーソル内のローの数を、スレッショルド以内で返します。

Visual Basic 構文

```
Public Function GetRowCount(ByVal threshold As Integer) As Integer
```

C# 構文

```
public int GetRowCount(int threshold)
```

パラメーター

- **threshold** ローカウントのスレッショルド制限。

例外

- **ULException クラス** SQL エラーが発生しました。

備考

カーソル内のロー数。

RowCount プロパティではカーソルのローを通過する必要があるため、クエリが複雑で大きな負荷がかかります。GetRowCount(threshold) を使用することにより、少なくともスレッショルドのローが存在するかどうかを呼び出し元で判断できます。ロー数がスレッショルドより低い場合はロー数が返され、そうでない場合はスレッショルドが返されます。高いスレッショルドを使用して、このメソッドを再度呼び出すことができます。

スレッショルドが 0 の場合は、RowCount プロパティが返されます。

参照

- ULDataReader.RowCount プロパティ [Ultra Light.NET]271 ページ

GetSchemaTable メソッド

ULDataReader のカラムのメタデータが記述された System.Data.DataTable を返します。

Visual Basic 構文

```
Public Overrides Function GetSchemaTable() As DataTable
```

C# 構文

```
public override DataTable GetSchemaTable()
```

戻り値

ULDataReader の各カラムのスキーマが記述された System.Data.DataTable。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

GetSchemaTable メソッドは、各カラムに関するメタデータを次の順で返します。

DataTable カラム	説明
ColumnName	カラムの名前。カラムに名前がない場合は NULL 参照 (Visual Basic の Nothing)。SQL クエリでカラムのエイリアスが使用されている場合は、そのエイリアスが返されます。結果セットでは、すべてのカラムに名前があるとは限らないほか、すべてのカラム名がユニークであるとは限りません。
ColumnOrdinal	カラムの ID。値の範囲は、[0, FieldCount -1] です。
ColumnSize	サイズ指定されたカラムの場合、カラムの値の最大長。その他のカラムの場合、これは、そのデータ型のバイト単位のサイズです。
NumericPrecision	数値カラム (ProviderType ULDbType.Decimal または ULDbType.Numeric) の精度。カラムが数値型ではない場合は DBNull。
NumericScale	数値カラム (ProviderType ULDbType.Decimal または ULDbType.Numeric) の位取り。カラムが数値型ではない場合は DBNull。
IsUnique	カラムが取得元のテーブル (BaseTableName) でユニークな非計算カラムである場合は true。
IsKey	カラムが、結果セットのユニークキーからともに取得された結果セットの一連のカラムのいずれかである場合は true。IsKey が true に設定されたカラムのセットは、結果セット内のローをユニークに識別する最小限のセットである必要はありません。
BaseCatalogName	カラムが含まれているデータベース内のカタログの名前。Ultra Light.NET の場合、この値は常に DBNull です。
BaseColumnName	データベースのテーブル BaseTableName にあるカラムの元の名前。カラムが計算される場合と、この情報を特定できない場合は、DBNull です。

DataTable カラム	説明
BaseSchemaName	カラムが含まれているデータベース内のスキーマの名前。Ultra Light.NET の場合、この値は常に DBNull です。
BaseTableName	カラムが含まれているデータベースのテーブルの名前。カラムが計算される場合と、この情報を特定できない場合は、DBNull です。
DataType	この型のカラムに最適な .NET データ型。
AllowDBNull	カラムが NULL 入力可である場合は true、NULL 入力不可である場合またはこの情報を特定できない場合は false。
ProviderType	カラムの ULDbType。
IsIdentity	カラムが identity カラムである場合は true、identity カラムでない場合は false。Ultra Light.NET の場合、この値は常に false です。
IsAutoIncrement	カラムがオートインクリメントカラムまたはグローバルオートインクリメントカラムである場合は true、それ以外のカラムである場合 (または、この情報を特定できない場合) は false。
IsRowVersion	書き込みできない永続的なロー識別子がカラムに含まれており、ローを識別する以外は意味を持たない値がカラムにある場合は true。Ultra Light.NET の場合、この値は常に false です。
IsLong	カラムが ULDbType.LongVarchar カラムまたは ULDbType.LongBinary カラムである場合は true、それ以外のカラムの場合は false。
IsReadOnly	カラムが読み込み専用である場合は true、カラムが修正可能であるか、カラムのアクセス権を特定できない場合は false。
IsAliased	カラム名がエイリアスの場合は true、エイリアスでない場合は false。
IsExpression	カラムが式の場合は true、カラム値の場合は false。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- ULDbType 列挙体 [Ultra Light.NET]460 ページ
- System.Data.DataTable

GetString メソッド

指定されたカラムの値を System.String として返します。

Visual Basic 構文

```
Public Overrides Function GetString(ByVal colID As Integer) As String
```

C# 構文

```
public override String GetString(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.String として返されるカラム値。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.String

GetTimeSpan メソッド

指定されたカラムの値を、ミリ秒の精度の System.TimeSpan として返します。

Visual Basic 構文

```
Public Function GetTimeSpan(ByVal colID As Integer) As TimeSpan
```

C# 構文

```
public TimeSpan GetTimeSpan(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.TimeSpan として返されるカラム値。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.TimeSpan

GetUInt16 メソッド

指定されたカラムの値を System.UInt16 として返します。

Visual Basic 構文

```
Public Function GetUInt16(ByVal colID As Integer) As UShort
```

C# 構文

```
public ushort GetUInt16(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.UInt16 として返されるカラム値。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.UInt16

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

GetUInt32 メソッド

指定されたカラムの値を UInt32 として返します。

Visual Basic 構文

```
Public Function GetUInt32(ByVal colID As Integer) As UInteger
```

C# 構文

```
public uint GetUInt32(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

UInt32 として返されるカラム値。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.UInt32

GetUInt64 メソッド

指定されたカラムの値を System.UInt64 として返します。

Visual Basic 構文

```
Public Function GetUInt64(ByVal colID As Integer) As ULong
```

C# 構文

```
public ulong GetUInt64(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.UInt64 として返されるカラム値。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.UInt64

GetValue メソッド

指定されたカラムの値をネイティブフォーマットで返します。

Visual Basic 構文

```
Public Overrides Function GetValue(ByVal colID As Integer) As Object
```

C# 構文

```
public override object GetValue(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

そのカラムに最適な .NET 型としてのカラム値。カラムが NULL の場合は DBNull。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

このメソッドは、機能的には ULDataReader.this[int] と同じです。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ

GetValues メソッド

現在のローのすべてのカラム値を返します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public Overrides Function GetValues(ByVal values As Object()) As Integer
```

C# 構文

```
public override int GetValues(object[] values)
```

パラメーター

- **values**　ロー全体を保持する System.Objects 配列。

戻り値

取り出されるカラム値の数。配列の長さがカラムの数 (ULDataReader.FieldCount) よりも大きい場合は、FieldCount の項目のみが取り出され、配列の残りは変更されないままになります。

例外

- **ArgumentNullException**　*values* 配列が NULL であるか、長さがゼロです。
- **ULException クラス**　SQL エラーが発生しました。

備考

ほとんどのアプリケーションについて、GetValues メソッドは、各カラムを個々に取り出すのではなく、すべてのカラムを取り出す効率的な方法を提供します。

結果のローに含まれるカラムの数より少ないカラムが含まれる System.Object 配列を渡すことができます。System.Object 配列が保持するデータ量のみが配列にコピーされます。また、結果のローに含まれるカラムの数より長い System.Object 配列を渡すこともできます。

このメソッドは、NULL データベースカラムに対して DBNull を返します。その他のカラムの場合は、カラムの値をネイティブフォーマットで返します。

参照

- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.GetValue メソッド [Ultra Light.NET]262 ページ
- System.Object

IsDBNull メソッド

指定されたカラムの値が NULL かどうかをチェックします。

Visual Basic 構文

```
Public Overrides Function IsDBNull(ByVal colID As Integer) As Boolean
```

C# 構文

```
public override bool IsDBNull(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

戻り値

値が NULL の場合は true、NULL でない場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ

MoveAfterLast メソッド

UL 拡張： カーソルの最後のローの後に、カーソルを配置します。

Visual Basic 構文

```
Public Sub MoveAfterLast()
```

C# 構文

```
public void MoveAfterLast()
```

例外

- **ULException クラス**　SQL エラーが発生しました。

MoveBeforeFirst メソッド

UL 拡張： カーソルの最初のローの前に、カーソルを配置します。

Visual Basic 構文

```
Public Sub MoveBeforeFirst()
```

C# 構文

```
public void MoveBeforeFirst()
```

例外

- **ULException クラス**　SQL エラーが発生しました。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

MoveFirst メソッド

UL 拡張： カーソルの最初のローに、カーソルを配置します。

Visual Basic 構文

```
Public Function MoveFirst() As Boolean
```

C# 構文

```
public bool MoveFirst()
```

戻り値

成功した場合は true、失敗した場合は false。たとえば、ローがない場合、メソッドは失敗します。

例外

- **ULException クラス** SQL エラーが発生しました。

MoveLast メソッド

UL 拡張： カーソルの最後のローに、カーソルを配置します。

Visual Basic 構文

```
Public Function MoveLast() As Boolean
```

C# 構文

```
public bool MoveLast()
```

戻り値

成功した場合は true、失敗した場合は false。たとえば、ローがない場合、メソッドは失敗します。

例外

- **ULException クラス** SQL エラーが発生しました。

MoveNext メソッド

UL 拡張： カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。

Visual Basic 構文

```
Public Function MoveNext() As Boolean
```

C# 構文

```
public bool MoveNext()
```

戻り値

成功した場合は true、失敗した場合は false。たとえば、それ以上ローがない場合、メソッドは失敗します。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

このメソッドは、ULDataReader.Read メソッドと同じです。

参照

- ULDataReader.Read メソッド [Ultra Light.NET]268 ページ

MovePrevious メソッド

UL 拡張： カーソルを前のローに配置するか、最初のローの前に配置します。

Visual Basic 構文

```
Public Function MovePrevious() As Boolean
```

C# 構文

```
public bool MovePrevious()
```

戻り値

成功した場合は true、失敗した場合は false。たとえば、それ以上ローがない場合、メソッドは失敗します。

例外

- **ULException クラス**　SQL エラーが発生しました。

MoveRelative メソッド

UL 拡張： 現在のローを基準としてカーソルを配置します。

Visual Basic 構文

```
Public Function MoveRelative(ByVal offset As Integer) As Boolean
```

C# 構文

```
public bool MoveRelative(int offset)
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

パラメーター

- **offset**　移動するローの数。負の値を指定すると、後方に移動します。

戻り値

成功した場合は true、失敗した場合は false。たとえば、最初または最後のローを超えて移動する場合、メソッドは失敗します。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

指定された移動先にローがない場合には、false が返されます。その場合のカーソル位置は、*offset* が正であるときには最後のローの後 (ULDataReader.IsEOF) になり、*offset* が負であるときには最初のローの前 (ULDataReader.IsBOF) になります。

参照

- ULDataReader.IsEOF プロパティ [Ultra Light.NET]270 ページ
- ULDataReader.IsBOF プロパティ [Ultra Light.NET]269 ページ

NextResult メソッド

バッチ SQL 文の結果を読み込むときに ULDataReader を次の結果に進めます。

Visual Basic 構文

```
Public Overrides Function NextResult() As Boolean
```

C# 構文

```
public override bool NextResult()
```

戻り値

さらに結果セットがある場合は true、そうでない場合は false。Ultra Light.NET では、常に false が返されます。

例外

- **ULException クラス**　ULDataReader が開かれていません。

備考

UL 拡張： Ultra Light.NET では SQL 文のバッチはサポートされていないため、ULDataReader は最初 (唯一) の結果セットに常に配置されます。NextResult は、呼び出しても機能しません。

Read メソッド

カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。

Visual Basic 構文

```
Public Overrides Function Read() As Boolean
```

C# 構文

```
public override bool Read()
```

戻り値

成功した場合は true、失敗した場合は false。たとえば、それ以上ローがない場合、メソッドは失敗します。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

このメソッドは、ULDataReader.MoveNext メソッドと同じです。

参照

- ULDataReader.MoveNext メソッド [Ultra Light.NET]265 ページ

Depth プロパティ

現在のローのネストの深さを返します。

Visual Basic 構文

```
Public ReadOnly Overrides Property Depth As Integer
```

C# 構文

```
public override int Depth {get;}
```

例外

- **ULException クラス**　ULDataReader が開かれていません。

備考

最も外側のテーブルの深さは 0 です。

Ultra Light.NET のすべての結果セットの深さは 0 です。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

FieldCount プロパティ

このカーソル内のカラム数を返します。

Visual Basic 構文

```
Public ReadOnly Overrides Property FieldCount As Integer
```

C# 構文

```
public override int FieldCount {get;}
```

戻り値

整数としてのカーソル内のカラム数。カーソルが閉じている場合は 0 を返します。

備考

このメソッドは、ULCursorSchema.ColumnCount メソッドと同じです。

参照

- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ

HasRows プロパティ

ULDataReader に 1 つまたは複数のローがあるかどうかをチェックします。

Visual Basic 構文

```
Public ReadOnly Overrides Property HasRows As Boolean
```

C# 構文

```
public override bool HasRows {get;}
```

備考

結果セットに少なくとも 1 つのローがある場合は True、ローがない場合は false。

IsBOF プロパティ

UL 拡張： 現在のローの位置が最初のローの前であるかどうかを確認します。

Visual Basic 構文

```
Public ReadOnly Property IsBOF As Boolean
```

C# 構文

```
public bool IsBOF {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

現在のローの位置が最初のローの前にくる場合は true、それ以外の場合は false。

IsClosed プロパティ

カーソルが現在開いているかどうかを確認します。

Visual Basic 構文

```
Public ReadOnly Overrides Property IsClosed As Boolean
```

C# 構文

```
public override bool IsClosed {get;}
```

備考

カーソルが現在開いている場合は true、閉じている場合は false。

IsEOF プロパティ

UL 拡張： 現在のローの位置が最後のローの後であるかどうかを確認します。

Visual Basic 構文

```
Public ReadOnly Property IsEOF As Boolean
```

C# 構文

```
public bool IsEOF {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

現在のローの位置が最後のローの後にくる場合は true、それ以外の場合は false。

RecordsAffected プロパティ

SQL 文の実行によって変更、挿入、または削除されたローの数を返します。

Visual Basic 構文

```
Public ReadOnly Overrides Property RecordsAffected As Integer
```


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

C# 構文

```
public override int RecordsAffected {get;}
```

備考

SELECT 文または CommandType.TableDirect テーブルの場合、この値は -1 です。

SQL 文の実行によって変更、挿入、または削除されたローの数。

参照

- System.Data.CommandType

RowCount プロパティ

UL 拡張： カーソル内のロー数を返します。

Visual Basic 構文

```
Public ReadOnly Property RowCount As Integer
```

C# 構文

```
public int RowCount {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

カーソル内のロー数。

RowCount は、古いローを削除して領域を節約するタイミングを決定するときに使用します。ULConnection.StopSynchronizationDelete メソッドを使用すると、統合データベースからは削除しないで Ultra Light データベースから古いローを削除できます。

参照

- ULConnection.StartSynchronizationDelete メソッド [Ultra Light.NET]150 ページ
- ULConnection.StopSynchronizationDelete メソッド [Ultra Light.NET]150 ページ

Schema プロパティ

UL 拡張： このカーソルのスキーマを保持します。

Visual Basic 構文

```
Public ReadOnly Property Schema As ULCursorSchema
```

C# 構文

```
public ULCursorSchema Schema {get;}
```

備考

結果セットの場合、結果セットのスキーマを表す ULResultSetSchema オブジェクト。テーブルの場合、テーブルのスキーマを表す ULTableSchema オブジェクト。

このプロパティは、ULDataReader.GetSchemaTable からの結果に示されない Ultra Light.NET の詳細情報を含め、カーソルの完全なスキーマを表します。

参照

- ULTableSchema クラス [Ultra Light.NET]437 ページ
- ULDataReader.GetSchemaTable メソッド [Ultra Light.NET]256 ページ
- ULResultSetSchema クラス [Ultra Light.NET]376 ページ

this プロパティ

指定されたカラムの値をネイティブフォーマットで返します。

オーバーロードリスト

名前	説明
this[int] プロパティ	指定されたカラムの値をネイティブフォーマットで返します。
this[string] プロパティ	指定された名前のカラムの値をネイティブフォーマットで返します。

this[int] プロパティ

指定されたカラムの値をネイティブフォーマットで返します。

Visual Basic 構文

```
Public ReadOnly Overrides Property Item(
     ByVal colID As Integer
) As Object
```

C# 構文

```
public override object this[int colID] {get;}
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

戻り値

そのカラムに最適な .NET 型としてのカラム値。カラムが NULL の場合は DBNull。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

C# では、このプロパティは ULDataReader クラスのインデクサです。

このメソッドは、機能的には ULDataReader.GetValue(int) メソッドと同じです。

参照

- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.GetValue メソッド [Ultra Light.NET]262 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ

this[string] プロパティ

指定された名前のカラムの値をネイティブフォーマットで返します。

Visual Basic 構文

```
Public ReadOnly Overrides Property Item(ByVal name As String) As Object
```

C# 構文

```
public override object this[string name] {get;}
```

パラメーター

- **名前**　カラム名。

戻り値

そのカラムに最適な .NET 型としてのカラム値。カラムが NULL の場合は DBNull。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

C# では、このプロパティは ULDataReader クラスのインデクサです。

結果セットでは、すべてのカラムに名前があるとは限らないほか、すべてのカラム名がユニークであるとは限りません。エイリアスを使用していない場合は、非計算カラムの名前には、そのカラムの元となるテーブルの名前がプレフィクスとして付けられます。たとえば、MyTable.ID は、クエリ "SELECT ID FROM MyTable" の結果セットに含まれる唯一のカラムの名前です。

カラムに何回もアクセスするときは、名前ではなく、カラム ID でアクセスすると効率が良くなります。

このメソッドは次と同じです。

```
dataReader.GetValue( dataReader.GetOrdinal( name ) )
```

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetValue メソッド [Ultra Light.NET]262 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ

ULException クラス

Ultra Light.NET データベースによって返される SQL エラーを表します。

Visual Basic 構文

```
Public NotInheritable Class ULException
    Inherits System.ApplicationException
```

C# 構文

```
public sealed class ULException : System.ApplicationException
```

基本クラス

- System.ApplicationException

メンバー

継承されたメンバーを含む ULException クラスのすべてのメンバー。

名前	説明
GetObjectData メソッド	この ULException のシリアル化に必要なデータを SerializationInfo に移植します。
NativeError プロパティ	データベースによって返される SQLCODE を返します。
Source プロパティ	エラーを生成したプロバイダーの名前を返します。

備考

NativeError にエラーが返されたことを示す SQLCODE。

.NET Compact Framework では、このクラスはシリアル化可能ではありません。

参照

- ULException.NativeError プロパティ [Ultra Light.NET]275 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

GetObjectData メソッド

この ULException のシリアル化に必要なデータを SerializationInfo に移植します。

Visual Basic 構文

```
Public Overrides Sub GetObjectData(
     ByVal info As SerializationInfo,
    ByVal context As StreamingContext
)
```

C# 構文

```
public override void GetObjectData(
     SerializationInfo info,
    StreamingContext context
)
```

パラメーター

- **info**　データを移植する SerializationInfo。
- **context**　このシリアル化の宛先。

備考

.NET Compact Framework では、このメソッドはサポートされていません。

NativeError プロパティ

データベースによって返される SQLCODE を返します。

Visual Basic 構文

```
Public ReadOnly Property NativeError As ULSQLCode
```

C# 構文

```
public ULSQLCode NativeError {get;}
```

備考

データベースによって返される ULSQLCode 値。

Source プロパティ

エラーを生成したプロバイダーの名前を返します。

Visual Basic 構文

```
Public ReadOnly Overrides Property Source As String
```

C# 構文

```
public override string Source {get;}
```

備考

プロバイダーが Ultra Light.NET であることを示す文字列値。

ULFactory クラス

データソースクラスの iAnywhere.Data.UltraLite プロバイダーの実装のインスタンスを作成する、メソッドのセットを表します。

Visual Basic 構文

```
Public NotInheritable Class ULFactory
    Inherits System.Data.Common.DbProviderFactory
```

C# 構文

```
public sealed class ULFactory : System.Data.Common.DbProviderFactory
```

基本クラス

- System.Data.Common.DbProviderFactory

メンバー

継承されたメンバーを含む ULFactory クラスのすべてのメンバー。

名前	説明
CreateCommand メソッド	厳密に型指定された System.Data.Common.DbCommand インスタンスを返します。
CreateCommandBuilder メソッド	厳密に型指定された System.Data.Common.DbCommandBuilder インスタンスを返します。
CreateConnection メソッド	厳密に型指定された System.Data.Common.DbConnection インスタンスを返します。
CreateConnectionStringBuilder メソッド	厳密に型指定された System.Data.Common.DbConnectionStringBuilder インスタンスを返します。
CreateDataAdapter メソッド	厳密に型指定された System.Data.Common.DbDataAdapter インスタンスを返します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
CreateDataSourceEnumerator method (System.Data.Common.DbProviderFactory から継承)	System.Data.Common.DbDataSourceEnumerator クラスを実装するプロバイダーのクラスの新しいインスタンスを返します。
CreateParameter メソッド	厳密に型指定された System.Data.Common.DbParameter インスタンスを返します。
CreatePermission method (System.Data.Common.DbProviderFactory から継承)	System.Security.CodeAccessPermission クラスのプロバイダーバージョンを実装するプロバイダーのクラスの新しいインスタンスを返します。
CanCreateDataSourceEnumerator プロパティ	Ultra Light.NET で DbDataSourceEnumerator がサポートされないことを示す false を返します。
Instance フィールド	ULFactory クラスのシングルトンインスタンスを表します。

備考

ULFactory クラスは、.NET Compact Framework 2.0 では使用できません。

ADO.NET 2.0 には System.Data.Common.DbProviderFactories と System.Data.Common.DbProviderFactory という 2 つクラスが新しく追加され、プロバイダーに依存しないコードを簡単に作成できるようになりました。これらを Ultra Light.NET で使用するには、GetFactory に渡されるプロバイダーのインバリアント名として iAnywhere.Data.UltraLite を指定します。次に例を示します。

```
' Visual Basic
Dim factory As DbProviderFactory =
    DbProviderFactories.GetFactory( "iAnywhere.Data.UltraLite" )
Dim conn As DbConnection =
    factory.CreateConnection()
```

対応する C# 言語のコードを次に示します。

```
// C#
DbProviderFactory factory =
    DbProviderFactories.GetFactory( "iAnywhere.Data.UltraLite" );
DbConnection conn = factory.CreateConnection();
```

この例の中の conn は、ULConnection オブジェクトとして作成されます。

ADO.NET 2.0 におけるプロバイダーファクトリと汎用プログラミングについては、http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnvs05/html/vsgenerics.asp を参照してください。

Ultra Light.NET では、CreateCommandBuilder()、CreateDataSourceEnumerator()、CreatePermission() をサポートしていません。

参照

- System.Data.Common.DbProviderFactories
- System.Data.Common.DbProviderFactory

CreateCommand メソッド

厳密に型指定された System.Data.Common.DbCommand インスタンスを返します。

Visual Basic 構文

```
Public Overrides Function CreateCommand() As DbCommand
```

C# 構文

```
public override DbCommand CreateCommand()
```

戻り値

DbCommand として型指定された新しい ULCommand インスタンス。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- System.Data.Common.DbCommand

CreateCommandBuilder メソッド

厳密に型指定された System.Data.Common.DbCommandBuilder インスタンスを返します。

Visual Basic 構文

```
Public Overrides Function CreateCommandBuilder() As DbCommandBuilder
```

C# 構文

```
public override DbCommandBuilder CreateCommandBuilder()
```

戻り値

DbCommandBuilder として型指定された新しい ULCommandBuilder インスタンス。

参照

- ULCommandBuilder クラス [Ultra Light.NET]104 ページ
- System.Data.Common.DbCommandBuilder

CreateConnection メソッド

厳密に型指定された System.Data.Common.DbConnection インスタンスを返します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public Overrides Function CreateConnection() As DbConnection
```

C# 構文

```
public override DbConnection CreateConnection()
```

戻り値

DbConnection として型指定された新しい ULConnection インスタンス。

参照

- ULConnection クラス [Ultra Light.NET]116 ページ
- System.Data.Common.DbConnection

CreateConnectionStringBuilder メソッド

厳密に型指定された System.Data.Common.DbConnectionStringBuilder インスタンスを返します。

Visual Basic 構文

```
Public Overrides Function CreateConnectionStringBuilder()
    As DbConnectionStringBuilder
```

C# 構文

```
public override DbConnectionStringBuilder CreateConnectionStringBuilder()
```

戻り値

DbConnectionStringBuilder として型指定された新しい ULConnectionStringBuilder インスタンス。

参照

- ULConnectionStringBuilder クラス [Ultra Light.NET]175 ページ
- System.Data.Common.DbConnectionStringBuilder

CreateDataAdapter メソッド

厳密に型指定された System.Data.Common.DbDataAdapter インスタンスを返します。

Visual Basic 構文

```
Public Overrides Function CreateDataAdapter() As DbDataAdapter
```

C# 構文

```
public override DbDataAdapter CreateDataAdapter()
```

戻り値

DbDataAdapter として型指定された新しい ULDataAdapter インスタンス。

参照

- ULDataAdapter クラス [Ultra Light.NET]209 ページ
- System.Data.Common.DbDataAdapter

CreateParameter メソッド

厳密に型指定された System.Data.Common.DbParameter インスタンスを返します。

Visual Basic 構文

```
Public Overrides Function CreateParameter() As DbParameter
```

C# 構文

```
public override DbParameter CreateParameter()
```

戻り値

DbParameter として型指定された新しい ULParameter インスタンス。

参照

- ULParameter クラス [Ultra Light.NET]319 ページ
- System.Data.Common.DbParameter

CanCreateDataSourceEnumerator プロパティ

Ultra Light.NET で DbDataSourceEnumerator がサポートされないことを示す false を返します。

Visual Basic 構文

```
Public ReadOnly Overrides Property CanCreateDataSourceEnumerator As
Boolean
```

C# 構文

```
public override bool CanCreateDataSourceEnumerator {get;}
```

備考

ULFactory で CreateDataSourceEnumerator() メソッドを実装していないことを示す false。

Instance フィールド

ULFactory クラスのシングルトンインスタンスを表します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public Shared ReadOnly Instance As ULFactory
```

C# 構文

```
public static readonly ULFactory Instance;
```

備考

ULFactory クラスは、.NET Compact Framework 2.0 では使用できません。

ULFactory はシングルトンクラスであり、このクラスにはこのインスタンスだけが存在できることを意味します。

通常、このフィールドは使用されません。代わりに、ULFactory のこのインスタンスへの参照を取得するには、System.Data.Common.DbProviderFactories.GetFactory(String) を使用します。

参照

- ULFactory クラス [Ultra Light.NET]276 ページ

ULFileTransfer クラス

UL 拡張： Mobile Link サーバーを使用して、リモートデータベースからファイルを転送します。

Visual Basic 構文

```
Public NotInheritable Class ULFileTransfer
```

C# 構文

```
public sealed class ULFileTransfer
```

メンバー

継承されたメンバーを含む ULFileTransfer クラスのすべてのメンバー。

名前	説明
ULFileTransfer コンストラクター	ULFileTransfer オブジェクトを初期化します。
DownloadFile メソッド	このオブジェクトのプロパティで指定されたファイルをダウンロードします。
UploadFile メソッド	このオブジェクトのプロパティで指定されたファイルをアップロードします。
AuthenticationParms プロパティ	カスタムユーザー認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメーターを指定します。

名前	説明
AuthStatus プロパティ	前回行われたファイル転送の認証ステータスコードを返します。
AuthValue プロパティ	カスタムユーザー認証同期スクリプトからの戻り値を返します。
FileAuthCode プロパティ	前回行われたファイル転送の authenticate_file_transfer スクリプトからの戻り値を返します。
FileName プロパティ	ダウンロードするファイルの名前を指定します。
LocalFileName プロパティ	ダウンロードファイルのローカル名を指定します。
LocalPath プロパティ	ファイルをダウンロードする場所を指定します。
Password プロパティ	UserName で指定されたユーザーの Mobile Link パスワードです。
RemoteKey プロパティ	Mobile Link サーバーが Mobile Link クライアントをユニークに識別するキーです。
ResumePartialDownload プロパティ	前の部分的なダウンロードを再開するか、破棄するかを指定します。
Stream プロパティ	ファイル転送に使用する Mobile Link 同期ストリームを指定します。
StreamErrorCode プロパティ	前回行われたファイル転送のストリーム自体によってレポートされるエラーを返します。
StreamErrorSystem プロパティ	ストリームエラーシステム固有のコードを返します。
StreamParms プロパティ	同期ストリームの設定パラメーターを指定します。
TransferedFile プロパティ	前回行われたファイル転送時にファイルが実際にダウンロードされたかどうかを確認します。
UserName プロパティ	Mobile Link サーバーが Mobile Link クライアントを識別するユーザー名です。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
Version プロパティ	使用する同期スクリプトを指定します。

備考

ファイルを転送するためにデータベースに接続する必要はありません。ただし、Ultra Light ランタイムを使用した Ultra Light データベースをアプリケーションで使用する場合は、この API や Ultra Light.NET API を使用する前に、ULDatabaseManager.RuntimeType に適切な値を設定する必要があります。

ファイルを転送するには、ULFileTransfer.FileName、ULFileTransfer.Stream、ULFileTransfer.UserName、ULFileTransfer.Version を設定する必要があります。

参照

- ULFileTransfer.FileName プロパティ [Ultra Light.NET]291 ページ
- ULFileTransfer.Stream プロパティ [Ultra Light.NET]294 ページ
- ULFileTransfer.UserName プロパティ [Ultra Light.NET]296 ページ
- ULFileTransfer.Version プロパティ [Ultra Light.NET]297 ページ

ULFileTransfer コンストラクター

ULFileTransfer オブジェクトを初期化します。

Visual Basic 構文

```
Public Sub New()
```

C# 構文

```
public ULFileTransfer()
```

備考

接続を開いてから、データベース操作を実行してください。

ファイルを転送するためにデータベースに接続する必要はありません。ただし、Ultra Light ランタイムを使用した Ultra Light データベースをアプリケーションで使用する場合は、この API や Ultra Light.NET API を使用する前に、ULDatabaseManager.RuntimeType に適切な値を設定する必要があります。

ULFileTransfer オブジェクトに ULFileTransfer.FileName、ULFileTransfer.Stream、ULFileTransfer.UserName、ULFileTransfer.Version が設定されていないと、ファイルを転送できません。

参照

- ULFileTransfer.DownloadFile メソッド [Ultra Light.NET]284 ページ
- ULFileTransfer.UploadFile メソッド [Ultra Light.NET]286 ページ
- ULFileTransfer.FileName プロパティ [Ultra Light.NET]291 ページ
- ULFileTransfer.Stream プロパティ [Ultra Light.NET]294 ページ
- ULFileTransfer.UserName プロパティ [Ultra Light.NET]296 ページ
- ULFileTransfer.Version プロパティ [Ultra Light.NET]297 ページ

DownloadFile メソッド

このオブジェクトのプロパティで指定されたファイルをダウンロードします。

オーバーロードリスト

名前	説明
DownloadFile() メソッド	このオブジェクトのプロパティで指定されたファイルをダウンロードします。
DownloadFile(ULFileTransferProgressListener) メソッド	指定されたリスナーに送信されるプログレスイベントとともに、このオブジェクトのプロパティで指定されたファイルをダウンロードします。

DownloadFile() メソッド

このオブジェクトのプロパティで指定されたファイルをダウンロードします。

Visual Basic 構文

```
Public Function DownloadFile() As Boolean
```

C# 構文

```
public bool DownloadFile()
```

戻り値

成功であった場合は true、それ以外の場合は false (理由については ULFileTransfer.StreamErrorCode とその他のステータスに関するプロパティを確認)。

備考

ULFileTransfer.FileName で指定されたファイルが、Mobile Link サーバーによって ULFileTransfer.LocalPath にダウンロードされます。このとき、ULFileTransfer.Stream、ULFileTransfer.UserName、ULFileTransfer.Password、および ULFileTransfer.Version が使用されます。ダウンロードに影響を及ぼすその他のプロパティは、ULFileTransfer.LocalFileName、ULFileTransfer.AuthenticationParms、および ULFileTransfer.ResumePartialDownload です。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

ファイルが破損することを防ぐため、Ultra Light.NET はテンポラリファイルにダウンロードし、ダウンロードが完了したときのみローカルファイルを置換します。

結果の詳細なステータスは、このオブジェクトの ULFileTransfer.AuthStatus、ULFileTransfer.AuthStatus、ULFileTransfer.FileAuthCode、ULFileTransfer.TransferedFile、ULFileTransfer.StreamErrorCode、および ULFileTransfer.StreamErrorSystem でレポートされます。

参照

- ULFileTransfer.DownloadFile メソッド [Ultra Light.NET]284 ページ
- ULFileTransfer.FileName プロパティ [Ultra Light.NET]291 ページ
- ULFileTransfer.LocalPath プロパティ [Ultra Light.NET]292 ページ
- ULFileTransfer.Stream プロパティ [Ultra Light.NET]294 ページ
- ULFileTransfer.UserName プロパティ [Ultra Light.NET]296 ページ
- ULFileTransfer.Password プロパティ [Ultra Light.NET]292 ページ
- ULFileTransfer.Version プロパティ [Ultra Light.NET]297 ページ
- ULFileTransfer.LocalFileName プロパティ [Ultra Light.NET]291 ページ
- ULFileTransfer.AuthenticationParms プロパティ [Ultra Light.NET]289 ページ
- ULFileTransfer.ResumePartialDownload プロパティ [Ultra Light.NET]293 ページ
- ULFileTransfer.AuthStatus プロパティ [Ultra Light.NET]290 ページ
- ULFileTransfer.AuthValue プロパティ [Ultra Light.NET]290 ページ
- ULFileTransfer.FileAuthCode プロパティ [Ultra Light.NET]290 ページ
- ULFileTransfer.TransferedFile プロパティ [Ultra Light.NET]296 ページ
- ULFileTransfer.StreamErrorCode プロパティ [Ultra Light.NET]294 ページ
- ULFileTransfer.StreamErrorSystem プロパティ [Ultra Light.NET]295 ページ
- ULFileTransfer.StreamErrorCode プロパティ [Ultra Light.NET]294 ページ
- ULFileTransfer.UploadFile メソッド [Ultra Light.NET]286 ページ

DownloadFile(ULFileTransferProgressListener) メソッド

指定されたリスナーに送信されるプログレスイベントとともに、このオブジェクトのプロパティで指定されたファイルをダウンロードします。

Visual Basic 構文

```
Public Function DownloadFile(
     ByVal listener As ULFileTransferProgressListener
) As Boolean
```

C# 構文

```
public bool DownloadFile(ULFileTransferProgressListener listener)
```

パラメーター

- **listener**　ファイル転送プログレスイベントを受信するオブジェクト。

戻り値

成功であった場合は true、それ以外の場合は false (理由については ULFileTransfer.StreamErrorCode とその他のステータスに関するプロパティを確認)。

備考

ULFileTransfer.FileName で指定されたファイルが、Mobile Link サーバーによって ULFileTransfer.LocalPath にダウンロードされます。このとき、ULFileTransfer.Stream、ULFileTransfer.UserName、ULFileTransfer.Password、および ULFileTransfer.Version が使用されます。ダウンロードに影響を及ぼすその他のプロパティは、ULFileTransfer.LocalFileName、ULFileTransfer.AuthenticationParms、および ULFileTransfer.ResumePartialDownload です。

ファイルが破損することを防ぐため、Ultra Light.NET はテンポラリファイルにダウンロードし、ダウンロードが完了したときのみローカルファイルを置換します。

結果の詳細なステータスは、このオブジェクトの ULFileTransfer.AuthStatus、ULFileTransfer.AuthStatus、ULFileTransfer.FileAuthCode、ULFileTransfer.TransferedFile、ULFileTransfer.StreamErrorCode、および ULFileTransfer.StreamErrorSystem でレポートされます。

エラーが発生すると、リスナーにはデータが送信されないことがあります。

参照

- ULFileTransfer.FileName プロパティ [Ultra Light.NET]291 ページ
- ULFileTransfer.LocalPath プロパティ [Ultra Light.NET]292 ページ
- ULFileTransfer.Stream プロパティ [Ultra Light.NET]294 ページ
- ULFileTransfer.UserName プロパティ [Ultra Light.NET]296 ページ
- ULFileTransfer.Password プロパティ [Ultra Light.NET]292 ページ
- ULFileTransfer.Version プロパティ [Ultra Light.NET]297 ページ
- ULFileTransfer.LocalFileName プロパティ [Ultra Light.NET]291 ページ
- ULFileTransfer.AuthenticationParms プロパティ [Ultra Light.NET]289 ページ
- ULFileTransfer.ResumePartialDownload プロパティ [Ultra Light.NET]293 ページ
- ULFileTransfer.AuthStatus プロパティ [Ultra Light.NET]290 ページ
- ULFileTransfer.AuthValue プロパティ [Ultra Light.NET]290 ページ
- ULFileTransfer.FileAuthCode プロパティ [Ultra Light.NET]290 ページ
- ULFileTransfer.TransferedFile プロパティ [Ultra Light.NET]296 ページ
- ULFileTransfer.StreamErrorCode プロパティ [Ultra Light.NET]294 ページ
- ULFileTransfer.StreamErrorSystem プロパティ [Ultra Light.NET]295 ページ
- ULFileTransfer.StreamErrorCode プロパティ [Ultra Light.NET]294 ページ
- ULFileTransfer.UploadFile メソッド [Ultra Light.NET]286 ページ

UploadFile メソッド

このオブジェクトのプロパティで指定されたファイルをアップロードします。

オーバーロードリスト

名前	説明
UploadFile() メソッド	このオブジェクトのプロパティで指定されたファイルをアップロードします。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
UploadFile(ULFileTransferProgressListener) メソッド	指定されたリスナーに送信されるプログレスイベントとともに、このオブジェクトのプロパティで指定されたファイルをアップロードします。

UploadFile() メソッド

このオブジェクトのプロパティで指定されたファイルをアップロードします。

Visual Basic 構文

```
Public Function UploadFile() As Boolean
```

C# 構文

```
public bool UploadFile()
```

戻り値

成功であった場合は true、それ以外の場合は false (理由については ULFileTransfer.StreamErrorCode とその他のステータスに関するプロパティを確認)。

備考

ULFileTransfer.FileName で指定されたファイルが、Mobile Link サーバーに ULFileTransfer.LocalPath からアップロードされます。このとき、ULFileTransfer.Stream、ULFileTransfer.UserName、ULFileTransfer.Password、および ULFileTransfer.Version が使用されます。

結果の詳細なステータスは、このオブジェクトの ULFileTransfer.AuthStatus、ULFileTransfer.AuthStatus、ULFileTransfer.FileAuthCode、ULFileTransfer.TransferedFile、ULFileTransfer.StreamErrorCode、および ULFileTransfer.StreamErrorSystem でレポートされます。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULFileTransfer.UploadFile メソッド [Ultra Light.NET]286 ページ
- ULFileTransfer.FileName プロパティ [Ultra Light.NET]291 ページ
- ULFileTransfer.LocalPath プロパティ [Ultra Light.NET]292 ページ
- ULFileTransfer.Stream プロパティ [Ultra Light.NET]294 ページ
- ULFileTransfer.UserName プロパティ [Ultra Light.NET]296 ページ
- ULFileTransfer.Password プロパティ [Ultra Light.NET]292 ページ
- ULFileTransfer.Version プロパティ [Ultra Light.NET]297 ページ
- ULFileTransfer.LocalFileName プロパティ [Ultra Light.NET]291 ページ
- ULFileTransfer.AuthenticationParms プロパティ [Ultra Light.NET]289 ページ
- ULFileTransfer.ResumePartialDownload プロパティ [Ultra Light.NET]293 ページ
- ULFileTransfer.AuthStatus プロパティ [Ultra Light.NET]290 ページ
- ULFileTransfer.AuthValue プロパティ [Ultra Light.NET]290 ページ
- ULFileTransfer.FileAuthCode プロパティ [Ultra Light.NET]290 ページ
- ULFileTransfer.TransferedFile プロパティ [Ultra Light.NET]296 ページ
- ULFileTransfer.StreamErrorCode プロパティ [Ultra Light.NET]294 ページ
- ULFileTransfer.StreamErrorSystem プロパティ [Ultra Light.NET]295 ページ
- ULFileTransfer.StreamErrorCode プロパティ [Ultra Light.NET]294 ページ

UploadFile(ULFileTransferProgressListener) メソッド

指定されたリスナーに送信されるプログレスイベントとともに、このオブジェクトのプロパティで指定されたファイルをアップロードします。

Visual Basic 構文

```
Public Function UploadFile(
     ByVal listener As ULFileTransferProgressListener
) As Boolean
```

C# 構文

```
public bool UploadFile(ULFileTransferProgressListener listener)
```

パラメーター

- **listener**　ファイル転送プログレスイベントを受信するオブジェクト。

戻り値

成功であった場合は true、それ以外の場合は false (理由については ULFileTransfer.StreamErrorCode とその他のステータスに関するプロパティを確認)。

備考

ULFileTransfer.FileName で指定されたファイルが、Mobile Link サーバーに ULFileTransfer.LocalPath からアップロードされます。このとき、ULFileTransfer.Stream、ULFileTransfer.UserName、ULFileTransfer.Password、および ULFileTransfer.Version が使用されます。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

結果の詳細なステータスは、このオブジェクトの ULFileTransfer.AuthStatus、ULFileTransfer.AuthStatus、ULFileTransfer.FileAuthCode、ULFileTransfer.TransferedFile、ULFileTransfer.StreamErrorCode、および ULFileTransfer.StreamErrorSystem でレポートされます。

エラーが発生すると、リスナーにはデータが送信されないことがあります。

参照

- ULFileTransfer.UploadFile メソッド [Ultra Light.NET]286 ページ
- ULFileTransfer.FileName プロパティ [Ultra Light.NET]291 ページ
- ULFileTransfer.LocalPath プロパティ [Ultra Light.NET]292 ページ
- ULFileTransfer.Stream プロパティ [Ultra Light.NET]294 ページ
- ULFileTransfer.UserName プロパティ [Ultra Light.NET]296 ページ
- ULFileTransfer.Password プロパティ [Ultra Light.NET]292 ページ
- ULFileTransfer.Version プロパティ [Ultra Light.NET]297 ページ
- ULFileTransfer.LocalFileName プロパティ [Ultra Light.NET]291 ページ
- ULFileTransfer.AuthenticationParms プロパティ [Ultra Light.NET]289 ページ
- ULFileTransfer.ResumePartialDownload プロパティ [Ultra Light.NET]293 ページ
- ULFileTransfer.AuthStatus プロパティ [Ultra Light.NET]290 ページ
- ULFileTransfer.AuthValue プロパティ [Ultra Light.NET]290 ページ
- ULFileTransfer.FileAuthCode プロパティ [Ultra Light.NET]290 ページ
- ULFileTransfer.TransferedFile プロパティ [Ultra Light.NET]296 ページ
- ULFileTransfer.StreamErrorCode プロパティ [Ultra Light.NET]294 ページ
- ULFileTransfer.StreamErrorSystem プロパティ [Ultra Light.NET]295 ページ
- ULFileTransfer.StreamErrorCode プロパティ [Ultra Light.NET]294 ページ

AuthenticationParms プロパティ

カスタムユーザー認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメーターを指定します。

Visual Basic 構文

```
Public Property AuthenticationParms As String()
```

C# 構文

```
public String[] AuthenticationParms {get;set;}
```

備考

それぞれに認証パラメーターが格納された、文字列の配列 (配列のエントリが NULL であると、同期エラーになります)。デフォルト値は NULL 参照 (Visual Basic の Nothing) で、認証パラメーターは指定されません。

最初の 255 文字列のみが使用されます。それぞれの文字列は認証パラメーターの Mobile Link サーバーの制限 (現在は 4000 UTF8 バイト) よりも長くはなりません。

AuthStatus プロパティ

前回行われたファイル転送の認証ステータスコードを返します。

Visual Basic 構文

```
Public ReadOnly Property AuthStatus As ULAuthStatusCode
```

C# 構文

```
public ULAuthStatusCode AuthStatus {get;}
```

備考

前回行われたファイル転送の認証ステータスを示す ULAuthStatusCode 値の 1 つ。

参照

- ULAuthStatusCode 列挙体 [Ultra Light.NET]457 ページ

AuthValue プロパティ

カスタムユーザー認証同期スクリプトからの戻り値を返します。

Visual Basic 構文

```
Public ReadOnly Property AuthValue As Long
```

C# 構文

```
public long AuthValue {get;}
```

備考

カスタムユーザー認証同期スクリプトから返された long integer。

FileAuthCode プロパティ

前回行われたファイル転送の authenticate_file_transfer スクリプトからの戻り値を返します。

Visual Basic 構文

```
Public ReadOnly Property FileAuthCode As UShort
```

C# 構文

```
public ushort FileAuthCode {get;}
```

備考

前回行われたファイル転送の authenticate_file_transfer スクリプトから戻された unsigned short の整数。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

FileName プロパティ

ダウンロードするファイルの名前を指定します。

Visual Basic 構文

```
Public Property FileName As String
```

C# 構文

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

備考

Mobile Link サーバーによって認識されたファイルの名前を指定する文字列。このプロパティにはデフォルト値がないので、明示的に設定してください。

FileName は、Mobile Link を実行するサーバー上にあるファイルの名前です。指定されたファイルは、まず UserName サブフォルダーで、次にルートディレクトリで検索されます (ルートディレクトリは、Mobile Link サーバーの -ftr オプションで指定)。Mobile Link でファイルを検索できるようにするため、FileName にはドライブまたはパスの情報を含めないでください。たとえば、"myfile.txt" は有効ですが、"somedir¥myfile.txt"、"..¥myfile.txt"、"c:¥myfile.txt" は無効です。

参照

- ULFileTransfer.LocalFileName プロパティ [Ultra Light.NET]291 ページ
- ULFileTransfer.DownloadFile メソッド [Ultra Light.NET]284 ページ
- ULFileTransfer.UploadFile メソッド [Ultra Light.NET]286 ページ

LocalFileName プロパティ

ダウンロードファイルのローカル名を指定します。

Visual Basic 構文

```
Public Property LocalFileName As String
```

C# 構文

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

備考

ダウンロードファイルのローカル名を指定する文字列。値が NULL 参照 (Visual Basic の Nothing) の場合、FileName が使用されます。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

参照

- ULFileTransfer.FileName プロパティ [Ultra Light.NET]291 ページ

LocalPath プロパティ

ファイルをダウンロードする場所を指定します。

Visual Basic 構文

```
Public Property LocalPath As String
```

C# 構文

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

備考

ファイルのローカルディレクトリを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

デフォルトのローカルディレクトリは、デバイスのオペレーティングシステムによって異なります。

- Windows モバイルデバイスの場合は、LocalPath は NULL 参照 (Visual Basic の Nothing) で、ファイルはルート (¥) ディレクトリに格納されます。
- デスクトップアプリケーションの場合は、LocalPath は NULL 参照 (Visual Basic の Nothing) で、ファイルは現在のディレクトリに格納されます。

参照

- ULFileTransfer.DownloadFile メソッド [Ultra Light.NET]284 ページ
- ULFileTransfer.UploadFile メソッド [Ultra Light.NET]286 ページ

Password プロパティ

UserName で指定されたユーザーの Mobile Link パスワードです。

Visual Basic 構文

```
Public Property Password As String
```

C# 構文

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

備考

Mobile Link パスワードを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) で、パスワードは指定されません。

Mobile Link ユーザー名とパスワードは他のデータベースユーザー ID やパスワードとは別のもので、アプリケーションを Mobile Link サーバーに対して識別し、認証するために使用されます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULFileTransfer.UserName プロパティ [Ultra Light.NET]296 ページ
- ULFileTransfer.DownloadFile メソッド [Ultra Light.NET]284 ページ
- ULFileTransfer.UploadFile メソッド [Ultra Light.NET]286 ページ

RemoteKey プロパティ

Mobile Link サーバーが Mobile Link クライアントをユニークに識別するキーです。

Visual Basic 構文

```
Public Property RemoteKey As String
```

C# 構文

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

備考

リモートキーを指定する文字列。このプロパティにはデフォルト値がないので、明示的に設定してください。

Mobile Link サーバーはこの値をさまざまなスクリプトに渡して、このクライアントをユニークに識別します。

参照

- ULFileTransfer.DownloadFile メソッド [Ultra Light.NET]284 ページ
- ULFileTransfer.UploadFile メソッド [Ultra Light.NET]286 ページ

ResumePartialDownload プロパティ

前の部分的なダウンロードを再開するか、破棄するかを指定します。

Visual Basic 構文

```
Public Property ResumePartialDownload As Boolean
```

C# 構文

```
public bool ResumePartialDownload {get;set;}
```

備考

前の部分的なダウンロードを再開する場合は true、破棄する場合は false。デフォルトは false です。

Ultra Light.NET では、通信エラーや、ユーザーによる ULFileTransferListener からのアボートが原因で失敗したダウンロードを再起動できます。Ultra Light.NET は、ダウンロードを受信しながら処理します。ダウンロードが中断した場合は、部分的なダウンロードファイルが保持されるため、次の転送中に再開できます。

ファイルがサーバー上で更新されている場合は、部分的なダウンロードは破棄され、新しくダウンロードが開始されます。

参照

- ULFileTransfer.DownloadFile メソッド [Ultra Light.NET]284 ページ
- ULFileTransfer.UploadFile メソッド [Ultra Light.NET]286 ページ

Stream プロパティ

ファイル転送に使用する Mobile Link 同期ストリームを指定します。

Visual Basic 構文

```
Public Property Stream As ULStreamType
```

C# 構文

```
public ULStreamType Stream {get;set;}
```

備考

使用する同期ストリームのタイプを指定する ULStreamType 値の 1 つ。デフォルトは ULStreamType.TCPIP です。

ほとんどの同期ストリームでは、Mobile Link サーバーのアドレスを識別したり、その他の動作を制御したりするパラメーターが必要です。これらのパラメーターは、ULFileTransfer.StreamParms で指定します。

ストリームタイプが、プラットフォームに適さない無効な値に設定されていると、ストリームタイプは ULStreamType.TCPIP に設定されます。

参照

- ULStreamType 列挙体 [Ultra Light.NET]466 ページ
- ULFileTransfer.StreamParms プロパティ [Ultra Light.NET]295 ページ
- ULFileTransfer.DownloadFile メソッド [Ultra Light.NET]284 ページ
- ULFileTransfer.UploadFile メソッド [Ultra Light.NET]286 ページ

StreamErrorCode プロパティ

前回行われたファイル転送のストリーム自体によってレポートされるエラーを返します。

Visual Basic 構文

```
Public ReadOnly Property StreamErrorCode As ULStreamErrorCode
```

C# 構文

```
public ULStreamErrorCode StreamErrorCode {get;}
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

ストリーム自体によってレポートされるエラーを示す ULStreamErrorCode 値の 1 つ。ただし、エラーが発生しなかった場合は ULStreamErrorCode.NONE。

StreamErrorSystem プロパティ

ストリームエラーシステム固有のコードを返します。

Visual Basic 構文

```
Public ReadOnly Property StreamErrorSystem As Integer
```

C# 構文

```
public int StreamErrorSystem {get;}
```

備考

ストリームエラーシステム固有のコードを示す整数。

StreamParms プロパティ

同期ストリームの設定パラメーターを指定します。

Visual Basic 構文

```
Public Property StreamParms As String
```

C# 構文

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

備考

キーワードと値の組み合わせがセミコロンで区切られたリスト形式の、ストリームのパラメーターを指定する文字列。デフォルトは NULL 参照です。(Visual Basic の Nothing)

StreamParms は、同期ストリームに使用されるすべてのパラメーターが含まれている文字列です。パラメーターは、「名前=値」のペアをセミコロンで区切ったリスト ("param1=value1;param2=value2") で指定します。

参照

- ULFileTransfer.Stream プロパティ [Ultra Light.NET]294 ページ
- ULFileTransfer.DownloadFile メソッド [Ultra Light.NET]284 ページ
- ULFileTransfer.UploadFile メソッド [Ultra Light.NET]286 ページ
- ULStreamType 列挙体 [Ultra Light.NET]466 ページ
- 「Ultra Light 同期ストリームのネットワークプロトコルのオプション」『Ultra Light データベース管理とリファレンス』

TransferedFile プロパティ

前回行われたファイル転送時にファイルが実際にダウンロードされたかどうかを確認します。

Visual Basic 構文

```
Public ReadOnly Property TransferedFile As Boolean
```

C# 構文

```
public bool TransferedFile {get;}
```

備考

ファイルがダウンロードされている場合は true、それ以外の場合は false。

DownloadFile() または UploadFile() が呼び出された時点でファイルがすでに最新だった場合、結果は true になりますが、TransferedFile は false に設定されます。エラーが発生して DownloadFile() または UploadFile() が false を返した場合は、TransferedFile は false に設定されます。

参照

- ULFileTransfer.DownloadFile メソッド [Ultra Light.NET]284 ページ
- ULFileTransfer.UploadFile メソッド [Ultra Light.NET]286 ページ

UserName プロパティ

Mobile Link サーバーが Mobile Link クライアントを識別するユーザー名です。

Visual Basic 構文

```
Public Property UserName As String
```

C# 構文

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

備考

ユーザー名を指定する文字列。このプロパティにはデフォルト値がないので、明示的に設定してください。

Mobile Link サーバーは、この値を使用して、ダウンロードするファイルを特定します。Mobile Link ユーザー名とパスワードは他のデータベースユーザー ID やパスワードとは別のもので、アプリケーションを Mobile Link サーバーに対して識別し、認証するために使用されます。

参照

- ULFileTransfer.Password プロパティ [Ultra Light.NET]292 ページ
- ULFileTransfer.DownloadFile メソッド [Ultra Light.NET]284 ページ
- ULFileTransfer.UploadFile メソッド [Ultra Light.NET]286 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Version プロパティ

使用する同期スクリプトを指定します。

Visual Basic 構文

```
Public Property Version As String
```

C# 構文

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

備考

使用する同期スクリプトのバージョンを指定する文字列。このプロパティにはデフォルト値がないので、明示的に設定してください。

統合データベースの同期スクリプトは、それぞれバージョン文字列でマーク付けされます。Ultra Light アプリケーションは、バージョン文字列により、同期スクリプトのセットから選択できます。

参照

- ULFileTransfer.DownloadFile メソッド [Ultra Light.NET]284 ページ
- ULFileTransfer.UploadFile メソッド [Ultra Light.NET]286 ページ

ULFileTransferProgressData クラス

UL 拡張： ファイル転送の進行状況のモニタリングデータを返します。

Visual Basic 構文

```
Public Class ULFileTransferProgressData
```

C# 構文

```
public class ULFileTransferProgressData
```

メンバー

継承されたメンバーを含む ULFileTransferProgressData クラスのすべてのメンバー。

名前	説明
BytesReceived プロパティ	現在までに受信したバイト数を返します。
FileSize プロパティ	転送されるファイルのサイズを返します。
Flags プロパティ	現在の状態に関連する追加情報を示す、現在のファイル転送フラグを返します。

名前	説明
ResumedAtSize プロパティ	転送が再開されるファイル内のポイントを返します。
FLAG_IS_BLOCKING フィールド	Mobile Link サーバーからの応答の待機中、ファイル転送はブロックされていることを示すフラグです。

参照

- ULFileTransferProgressListener インターフェイス [Ultra Light.NET]299 ページ

BytesReceived プロパティ

現在までに受信したバイト数を返します。

Visual Basic 構文

```
Public ReadOnly Property BytesReceived As ULong
```

C# 構文

```
public ulong BytesReceived {get;}
```

備考

現在までに受信したバイト数。

FileSize プロパティ

転送されるファイルのサイズを返します。

Visual Basic 構文

```
Public ReadOnly Property FileSize As ULong
```

C# 構文

```
public ulong FileSize {get;}
```

備考

ファイルのサイズ (バイト単位)。

Flags プロパティ

現在の状態に関連する追加情報を示す、現在のファイル転送フラグを返します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public ReadOnly Property Flags As Integer
```

C# 構文

```
public int Flags {get;}
```

備考

フラグの組み合わせを保持する整数。

参照

- ULFileTransferProgressData.FLAG_IS_BLOCKING フィールド [Ultra Light.NET]299 ページ

ResumedAtSize プロパティ

転送が再開されるファイル内のポイントを返します。

Visual Basic 構文

```
Public ReadOnly Property ResumedAtSize As ULong
```

C# 構文

```
public ulong ResumedAtSize {get;}
```

備考

以前に転送されたバイト数。

FLAG_IS_BLOCKING フィールド

Mobile Link サーバーからの応答の待機中、ファイル転送はブロックされていることを示すフラグです。

Visual Basic 構文

```
Public Const FLAG_IS_BLOCKING As Integer
```

C# 構文

```
public const int FLAG_IS_BLOCKING;
```

ULFileTransferProgressListener インターフェイス

UL 拡張： ファイル転送プログレスイベントを受信するリスナーインターフェイスです。

Visual Basic 構文

```
Public Interface ULFileTransferProgressListener
```

C# 構文

```
public interface ULFileTransferProgressListener
```

メンバー

継承されたメンバーを含む ULFileTransferProgressListener インターフェイスのすべてのメンバー。

名前	説明
FileTransferProgressed メソッド	ユーザーに進行状況を通知するために、ファイル転送中に起動されます。

参照

- ULFileTransfer.DownloadFile メソッド [Ultra Light.NET]284 ページ

FileTransferProgressed メソッド

ユーザーに進行状況を通知するために、ファイル転送中に起動されます。

Visual Basic 構文

```
Public Function FileTransferProgressed(
     ByVal data As ULFileTransferProgressData
) As Boolean
```

C# 構文

```
public bool FileTransferProgressed(ULFileTransferProgressData data)
```

パラメーター

- **data** 最新の同期のプログレスデータを保持している ULFileTransferProgressData オブジェクト。

戻り値

このメソッドは、転送をキャンセルする場合は true を、続行する場合は false を返します。

備考

このメソッドは、転送をキャンセルする場合は true を、続行する場合は false を返します。

FileTransferProgressed の呼び出し中に、Ultra Light.NET API のメソッドを呼び出さないでください。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULFileTransferProgressData クラス [Ultra Light.NET]297 ページ

ULIndexSchema クラス

UL 拡張： Ultra Light テーブルのインデックスのスキーマを示します。

Visual Basic 構文

```
Public NotInheritable Class ULIndexSchema
```

C# 構文

```
public sealed class ULIndexSchema
```

メンバー

継承されたメンバーを含む ULIndexSchema クラスのすべてのメンバー。

名前	説明
GetColumnName メソッド	このインデックス内の *colOrdinalInIndex* 番目のカラム名を返します。
IsColumnDescending メソッド	指定されたカラムが、インデックスによって降順で使用されるかどうかをチェックします。
ColumnCount プロパティ	インデックス内のカラム数を返します。
IsForeignKey プロパティ	インデックスが外部キーであるかどうかをチェックします。
IsForeignKeyCheckOnCommit プロパティ	外部キーの参照整合性のチェックが、コミット時に行われるか、挿入時と更新時に行われるかを確認します。
IsForeignKeyNullable プロパティ	外部キーが NULL 入力可であるかどうかをチェックします。
IsOpen プロパティ	インデックススキーマが開いているか、閉じているかを調べます。
IsPrimaryKey プロパティ	インデックスがプライマリキーであるかどうかをチェックします。
IsUniqueIndex プロパティ	インデックスがユニークであるかどうかをチェックします。

名前	説明
IsUniqueKey プロパティ	インデックスがユニークキーであるかどうかをチェックします。
Name プロパティ	インデックスの名前を返します。
ReferencedIndexName プロパティ	インデックスが外部キーである場合、参照されるプライマリインデックスの名前です。
ReferencedTableName プロパティ	インデックスが外部キーである場合、参照されるプライマリテーブルの名前です。

備考

このクラスにはコンストラクターがありません。インデックスのスキーマは、ULTableSchema の ULTableSchema.PrimaryKey、ULTableSchema.GetIndex(string)、および ULTableSchema.GetOptimalIndex(int) を使用して作成されます。

参照

- ULTableSchema.PrimaryKey プロパティ [Ultra Light.NET]450 ページ
- ULTableSchema.GetIndex メソッド [Ultra Light.NET]441 ページ
- ULTableSchema.GetOptimalIndex メソッド [Ultra Light.NET]442 ページ
- ULTableSchema クラス [Ultra Light.NET]437 ページ

GetColumnName メソッド

このインデックス内の *colOrdinalInIndex* 番目のカラム名を返します。

Visual Basic 構文

```
Public Function GetColumnName(
     ByVal colOrdinalInIndex As Short
) As String
```

C# 構文

```
public string GetColumnName(short colOrdinalInIndex)
```

パラメーター

- **colOrdinalInIndex**　インデックス内の対象のカラムの順序。値は、[1,ColumnCount] の範囲内であることが必要です。

戻り値

カラム名。

例外

- **ULException クラス**　SQL エラーが発生しました。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

カラムの順序とカウントは、スキーマのアップグレード中に変更されることがあります。インデックスからのカラムの順序は、特定のテーブルの同じ物理的カラムを参照する場合であっても、テーブルまたは別のインデックスのカラム ID とは異なります。

参照

- ULIndexSchema.ColumnCount プロパティ [Ultra Light.NET]303 ページ

IsColumnDescending メソッド

指定されたカラムが、インデックスによって降順で使用されるかどうかをチェックします。

Visual Basic 構文

```
Public Function IsColumnDescending(ByVal name As String) As Boolean
```

C# 構文

```
public bool IsColumnDescending(string name)
```

パラメーター

- **name**　カラム名。

戻り値

カラムが降順で使用される場合は true、昇順で使用される場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULIndexSchema.GetColumnName メソッド [Ultra Light.NET]302 ページ
- ULIndexSchema.ColumnCount プロパティ [Ultra Light.NET]303 ページ

ColumnCount プロパティ

インデックス内のカラム数を返します。

Visual Basic 構文

```
Public ReadOnly Property ColumnCount As Short
```

C# 構文

```
public short ColumnCount {get;}
```

備考

インデックス内のカラム数。

インデックス内のカラムの順序の範囲は、1 ～ ColumnCount です。

カラムの順序とカウントは、スキーマのアップグレード中に変更されることがあります。インデックスからのカラムの順序は、特定のテーブルの同じ物理的カラムを参照する場合であっても、テーブルまたは別のインデックスのカラム ID とは異なります。

IsForeignKey プロパティ

インデックスが外部キーであるかどうかをチェックします。

Visual Basic 構文

```
Public ReadOnly Property IsForeignKey As Boolean
```

C# 構文

```
public bool IsForeignKey {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

インデックスが外部キーである場合は true、外部キーでない場合は false。

外部キー内のカラムは、別のテーブルの NULL 以外のユニークインデックスを参照することができます。

IsForeignKeyCheckOnCommit プロパティ

外部キーの参照整合性のチェックが、コミット時に行われるか、挿入時と更新時に行われるかを確認します。

Visual Basic 構文

```
Public ReadOnly Property IsForeignKeyCheckOnCommit As Boolean
```

C# 構文

```
public bool IsForeignKeyCheckOnCommit {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました (インデックスが外部キーではない場合を含む)。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

参照整合性がコミット時にチェックされる場合は true、挿入時と更新時にチェックされる場合は false。

参照

- ULIndexSchema.IsForeignKey プロパティ [Ultra Light.NET]304 ページ

IsForeignKeyNullable プロパティ

外部キーが NULL 入力可であるかどうかをチェックします。

Visual Basic 構文

```
Public ReadOnly Property IsForeignKeyNullable As Boolean
```

C# 構文

```
public bool IsForeignKeyNullable {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました (インデックスが外部キーではない場合を含む)。

備考

外部キーが NULL 入力可の場合は true、NULL 入力不可の場合は false。

参照

- ULIndexSchema.IsForeignKey プロパティ [Ultra Light.NET]304 ページ

IsOpen プロパティ

インデックススキーマが開いているか、閉じているかを調べます。

Visual Basic 構文

```
Public ReadOnly Property IsOpen As Boolean
```

C# 構文

```
public bool IsOpen {get;}
```

備考

インデックススキーマが開いている場合は true、閉じている場合は false。

IsPrimaryKey プロパティ

インデックスがプライマリキーであるかどうかをチェックします。

Visual Basic 構文

```
Public ReadOnly Property IsPrimaryKey As Boolean
```

C# 構文

```
public bool IsPrimaryKey {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

インデックスがプライマリキーである場合は true、プライマリキーでない場合は false。

プライマリキー内のカラムでは NULL は許可されません。

IsUniqueIndex プロパティ

インデックスがユニークであるかどうかをチェックします。

Visual Basic 構文

```
Public ReadOnly Property IsUniqueIndex As Boolean
```

C# 構文

```
public bool IsUniqueIndex {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

インデックスがユニークである場合は true、ユニークでない場合は false。

ユニークなインデックス内のカラムは NULL であることがあります。

IsUniqueKey プロパティ

インデックスがユニークキーであるかどうかをチェックします。

Visual Basic 構文

```
Public ReadOnly Property IsUniqueKey As Boolean
```


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

C# 構文

```
public bool IsUniqueKey {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

インデックスがユニークキーである場合は true、ユニークキーでない場合は false。

ユニークキー内のカラムでは NULL は許可されません。

Name プロパティ

インデックスの名前を返します。

Visual Basic 構文

```
Public ReadOnly Property Name As String
```

C# 構文

```
public string Name {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

インデックスの名前を指定する文字列。

ReferencedIndexName プロパティ

インデックスが外部キーである場合、参照されるプライマリインデックスの名前です。

Visual Basic 構文

```
Public ReadOnly Property ReferencedIndexName As String
```

C# 構文

```
public string ReferencedIndexName {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました (インデックスが外部キーではない場合を含む)。

備考

参照されるプライマリインデックスの名前を指定する文字列。

参照

- ULIndexSchema.IsForeignKey プロパティ [Ultra Light.NET]304 ページ

ReferencedTableName プロパティ

インデックスが外部キーである場合、参照されるプライマリテーブルの名前です。

Visual Basic 構文

```
Public ReadOnly Property ReferencedTableName As String
```

C# 構文

```
public string ReferencedTableName {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました (インデックスが外部キーではない場合を含む)。

備考

参照されるプライマリテーブルの名前を指定する文字列。

参照

- ULIndexSchema.IsForeignKey プロパティ [Ultra Light.NET]304 ページ

ULInfoMessageEventArgs クラス

ULConnection.InfoMessage イベントのデータを提供します。

Visual Basic 構文

```
Public NotInheritable Class ULInfoMessageEventArgs
    Inherits System.EventArgs
```

C# 構文

```
public sealed class ULInfoMessageEventArgs : System.EventArgs
```

基本クラス

- System.EventArgs

メンバー

継承されたメンバーを含む ULInfoMessageEventArgs クラスのすべてのメンバー。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
ToString メソッド	ULConnection.InfoMessage イベントの文字列表現。
Message プロパティ	データベースによって返される情報メッセージまたは警告メッセージの文字列です。
NativeError プロパティ	データベースによって返される情報メッセージまたは警告に対応する SQLCODE です。
Source プロパティ	メッセージを返す ADO.NET データプロバイダーの名前です。
Empty field (System.EventArgs から継承)	イベントデータのないイベントを表します。

参照

- ULConnection.InfoMessage event [Ultra Light.NET]161 ページ

ToString メソッド

ULConnection.InfoMessage イベントの文字列表現。

Visual Basic 構文

```
Public Overrides Function ToString() As String
```

C# 構文

```
public override string ToString()
```

戻り値

情報メッセージまたは警告メッセージの文字列。

備考

ULConnection.InfoMessage イベントの文字列表現。

参照

- ULConnection.InfoMessage event [Ultra Light.NET]161 ページ

Message プロパティ

データベースによって返される情報メッセージまたは警告メッセージの文字列です。

Visual Basic 構文

```
Public ReadOnly Property Message As String
```

C# 構文

```
public string Message {get;}
```

備考

情報メッセージまたは警告メッセージが含まれる文字列。

NativeError プロパティ

データベースによって返される情報メッセージまたは警告に対応する SQLCODE です。

Visual Basic 構文

```
Public ReadOnly Property NativeError As ULSQLCode
```

C# 構文

```
public ULSQLCode NativeError {get;}
```

備考

情報メッセージまたは警告の ULSQLCode 値。

Source プロパティ

メッセージを返す ADO.NET データプロバイダーの名前です。

Visual Basic 構文

```
Public ReadOnly Property Source As String
```

C# 構文

```
public string Source {get;}
```

備考

文字列 "Ultra Light.NET Data Provider"。

ULMetaDataCollectionNames クラス

メタデータコレクションを取得する ULConnection.GetSchema(String,String[]) で使用する定数のリストを提供します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public NotInheritable Class ULMetaDataCollectionNames
```

C# 構文

```
public sealed class ULMetaDataCollectionNames
```

メンバー

継承されたメンバーを含む ULMetaDataCollectionNames クラスのすべてのメンバー。

名前	説明
Columns プロパティ	Columns コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。
DataSourceInformation プロパティ	DataSourceInformation コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。
DataTypes プロパティ	DataTypes コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。
ForeignKeys プロパティ	ForeignKeys コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。
IndexColumns プロパティ	IndexColumns コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。
Indexes プロパティ	Indexes コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。
MetaDataCollections プロパティ	MetaDataCollections コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。
Publications プロパティ	Publications コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。
ReservedWords プロパティ	ReservedWords コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。

名前	説明
Restrictions プロパティ	Restrictions コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。
Tables プロパティ	Tables コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。

Columns プロパティ

Columns コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。

Visual Basic 構文

```
Public Shared ReadOnly Property Columns As String
```

C# 構文

```
public string Columns {get;}
```

備考

Columns コレクションの名前を表す文字列。

参照

- ULConnection.GetSchema メソッド [Ultra Light.NET]141 ページ

例

次のコードは、Columns コレクションを使用して DataTable を設定します。

```
' Visual Basic
Dim schema As DataTable =
   conn.GetSchema( ULMetaDataCollectionNames.Columns )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
   conn.GetSchema( ULMetaDataCollectionNames.Columns );
```

DataSourceInformation プロパティ

DataSourceInformation コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。

Visual Basic 構文

```
Public Shared ReadOnly Property DataSourceInformation As String
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

C# 構文

```
public string DataSourceInformation {get;}
```

備考

DataSourceInformation コレクションの名前を表す文字列。

参照

- ULConnection.GetSchema メソッド [Ultra Light.NET]141 ページ

例

次のコードは、DataSourceInformation コレクションを使用して DataTable を設定します。

```
' Visual Basic
Dim schema As DataTable =
   conn.GetSchema( ULMetaDataCollectionNames.DataSourceInformation )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
   conn.GetSchema( ULMetaDataCollectionNames.DataSourceInformation );
```

DataTypes プロパティ

DataTypes コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。

Visual Basic 構文

```
Public Shared ReadOnly Property DataTypes As String
```

C# 構文

```
public string DataTypes {get;}
```

備考

DataTypes コレクションの名前を表す文字列。

参照

- ULConnection.GetSchema メソッド [Ultra Light.NET]141 ページ

例

次のコードは、DataTypes コレクションを使用して DataTable を設定します。

```
' Visual Basic
Dim schema As DataTable =
   conn.GetSchema( ULMetaDataCollectionNames.DataTypes )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
   conn.GetSchema( ULMetaDataCollectionNames.DataTypes );
```

ForeignKeys プロパティ

ForeignKeys コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。

Visual Basic 構文

```
Public Shared ReadOnly Property ForeignKeys As String
```

C# 構文

```
public string ForeignKeys {get;}
```

備考

ForeignKeys コレクションの名前を表す文字列。

参照

- ULConnection.GetSchema メソッド [Ultra Light.NET]141 ページ

例

次のコードは、ForeignKeys コレクションを使用して DataTable を設定します。

```
' Visual Basic
Dim schema As DataTable =
   conn.GetSchema( ULMetaDataCollectionNames.ForeignKeys )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
   conn.GetSchema( ULMetaDataCollectionNames.ForeignKeys );
```

IndexColumns プロパティ

IndexColumns コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。

Visual Basic 構文

```
Public Shared ReadOnly Property IndexColumns As String
```

C# 構文

```
public string IndexColumns {get;}
```

備考

IndexColumns コレクションの名前を表す文字列。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULConnection.GetSchema メソッド [Ultra Light.NET]141 ページ

例

次のコードは、IndexColumns コレクションを使用して DataTable を設定します。

```
' Visual Basic
Dim schema As DataTable =
    conn.GetSchema( ULMetaDataCollectionNames.IndexColumns )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.IndexColumns );
```

Indexes プロパティ

Indexes コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。

Visual Basic 構文

```
Public Shared ReadOnly Property Indexes As String
```

C# 構文

```
public string Indexes {get;}
```

備考

Indexes コレクションの名前を表す文字列。

参照

- ULConnection.GetSchema メソッド [Ultra Light.NET]141 ページ

例

次のコードは、Indexes コレクションを使用して DataTable を設定します。

```
' Visual Basic
Dim schema As DataTable =
    conn.GetSchema( ULMetaDataCollectionNames.Indexes )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.Indexes );
```

MetaDataCollections プロパティ

MetaDataCollections コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。

Visual Basic 構文

```
Public Shared ReadOnly Property MetaDataCollections As String
```

C# 構文

```
public string MetaDataCollections {get;}
```

備考

MetaDataCollections コレクションの名前を表す文字列。

参照

- ULConnection.GetSchema メソッド [Ultra Light.NET]141 ページ

例

次のコードは、MetaDataCollections コレクションを使用して DataTable を設定します。

```
' Visual Basic
Dim schema As DataTable =
   conn.GetSchema( ULMetaDataCollectionNames.MetaDataCollections )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
   conn.GetSchema( ULMetaDataCollectionNames.MetaDataCollections );
```

Publications プロパティ

Publications コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。

Visual Basic 構文

```
Public Shared ReadOnly Property Publications As String
```

C# 構文

```
public string Publications {get;}
```

備考

Publications コレクションの名前を示す文字列。

参照

- ULConnection.GetSchema メソッド [Ultra Light.NET]141 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

例

次のコードは、Publications コレクションを使用して DataTable を設定します。

```
' Visual Basic
Dim schema As DataTable =
    conn.GetSchema( ULMetaDataCollectionNames.Publications )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.Publications );
```

ReservedWords プロパティ

ReservedWords コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。

Visual Basic 構文

```
Public Shared ReadOnly Property ReservedWords As String
```

C# 構文

```
public string ReservedWords {get;}
```

備考

ReservedWords コレクションの名前を表す文字列。

参照

- ULConnection.GetSchema メソッド [Ultra Light.NET]141 ページ

例

次のコードは、ReservedWords コレクションを使用して DataTable を設定します。

```
' Visual Basic
Dim schema As DataTable =
    conn.GetSchema( ULMetaDataCollectionNames.ReservedWords )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.ReservedWords );
```

Restrictions プロパティ

Restrictions コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。

Visual Basic 構文

```
Public Shared ReadOnly Property Restrictions As String
```

C# 構文

```
public string Restrictions {get;}
```

備考

Restrictions コレクションの名前を表す文字列。

参照

- ULConnection.GetSchema メソッド [Ultra Light.NET]141 ページ

例

次のコードは、Restrictions コレクションを使用して DataTable を設定します。

```
' Visual Basic
Dim schema As DataTable =
   conn.GetSchema( ULMetaDataCollectionNames.Restrictions )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
   conn.GetSchema( ULMetaDataCollectionNames.Restrictions );
```

Tables プロパティ

Tables コレクションを表す ULConnection.GetSchema(String) で使用する定数を提供します。

Visual Basic 構文

```
Public Shared ReadOnly Property Tables As String
```

C# 構文

```
public string Tables {get;}
```

備考

Tables コレクションの名前を表す文字列。

参照

- ULConnection.GetSchema メソッド [Ultra Light.NET]141 ページ

例

次のコードは、Tables コレクションを使用して DataTable を設定します。

```
' Visual Basic
Dim schema As DataTable =
   conn.GetSchema( ULMetaDataCollectionNames.Tables )
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
// C#
DataTable schema =
   conn.GetSchema( ULMetaDataCollectionNames.Tables );
```

ULParameter クラス

ULCommand のパラメーターを表します。

Visual Basic 構文

```
Public NotInheritable Class ULParameter
    Inherits System.Data.Common.DbParameter
    Implements System.ICloneable
```

C# 構文

```
public sealed class ULParameter :
    System.Data.Common.DbParameter,
    System.ICloneable
```

基本クラス

- System.Data.Common.DbParameter
- System.ICloneable

メンバー

継承されたメンバーを含む ULParameter クラスのすべてのメンバー。

名前	説明
ULParameter コンストラクター	値として NULL (Visual Basic の Nothing) を使用して、ULParameter オブジェクトを初期化します。
ResetDbType メソッド	このメソッドは Ultra Light.NET ではサポートされていません。
ToString メソッド	このインスタンスの文字列表記を返します。
DbType プロパティ	パラメーターの System.Data.DbType を指定します。
Direction プロパティ	パラメーターが入力専用、出力専用、双方向性、またはストアドプロシージャー戻り値パラメーターのいずれであるかを示す値です。
IsNullable プロパティ	パラメーターが NULL 値を受け入れるかどうかを指定します。

名前	説明
Offset プロパティ	ULParameter.Value のオフセットを指定します。
ParameterName プロパティ	パラメーターの名前を指定します。
Precision プロパティ	ULParameter.Value を表すために使用される最大桁数を指定します。
Scale プロパティ	ULParameter.Value が解決される小数点の桁の数を指定します。
Size プロパティ	カラム内のデータの最大サイズを指定します。
SourceColumn プロパティ	DataSet にマッピングされ、値をロードしたり返したりするときに使用するソースカラムの名前を指定します。
SourceColumnNullMapping プロパティ	ソースカラムが NULL 入力可であるかどうかを指定します。
SourceVersion プロパティ	ULParameter.Value をロードするときに使用する System.Data.DataRowVersion を指定します。
ULDbType プロパティ	パラメーターの iAnywhere.Data.UltraLite.ULDbType を指定します。
Value プロパティ	パラメーターの値を指定します。

備考

ULParameter オブジェクトは、その多数のコンストラクターのいずれかを使用したり、ULCommand.CreateParameter メソッドを使用したりして直接作成できます。定数 0 および 0.0 の特別な処理と、オーバロードされたメソッドの解決方法のため、ULParameter(string,object) コンストラクターを使用するときは、定数値を型オブジェクトに明示的にキャストすることを強くおすすめします。次に例を示します。

```
' Visual Basic
Dim p As ULParameter = New ULParameter( "", CType( 0, Object ) )
```

対応する C# 言語のコードを次に示します。

```
// C#
ULParameter p = new ULParameter( "", (object)0 );
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

パラメーター (ULCommand.CreateParameter によって作成されたものを含む) は、使用される ULCommand.Parameters コレクションに追加する必要があります。すべてのパラメーターは、位置パラメーターとして扱われ、コマンドによって追加された順序で使用されます。

Ultra Light.NET では、パラメーターは IN パラメーターとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value だけです。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- ULCommand.CreateParameter メソッド [Ultra Light.NET]79 ページ
- ULParameter.ULParameter コンストラクター [Ultra Light.NET]321 ページ
- ULCommand.Parameters プロパティ [Ultra Light.NET]101 ページ
- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ
- System.Data.Common.DbParameter
- System.Data.IDbDataParameter
- System.Data.IDataParameter

ULParameter コンストラクター

値として NULL (Visual Basic の Nothing) を使用して、ULParameter オブジェクトを初期化します。

オーバーロードリスト

名前	説明
ULParameter() コンストラクター	値として NULL (Visual Basic の Nothing) を使用して、ULParameter オブジェクトを初期化します。
ULParameter(string, object) コンストラクター	ULParameter オブジェクトを、指定されたパラメーター名と値で初期化します。
ULParameter(string, ULDbType) コンストラクター	ULParameter オブジェクトを、指定されたパラメーター名とデータ型で初期化します。
ULParameter(string, ULDbType, int) コンストラクター	ULParameter オブジェクトを、指定されたパラメーター名とデータ型で初期化します。
ULParameter(string, ULDbType, int, ParameterDirection, bool, byte, byte, string, DataRowVersion, object) コンストラクター	指定されたパラメーター名、データ型、長さ、方向、NULL 入力属性、数値精度、数値の位取り、ソースカラム、ソースのバージョン、値で、ULParameter オブジェクトを初期化します。
ULParameter(string, ULDbType, int, string) コンストラクター	ULParameter オブジェクトを、指定されたパラメーター名、データ型、長さで初期化します。

ULParameter() コンストラクター

値として NULL (Visual Basic の Nothing) を使用して、ULParameter オブジェクトを初期化します。

Visual Basic 構文

```
Public Sub New()
```

C# 構文

```
public ULParameter()
```

参照

- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ
- ULParameter.ULParameter コンストラクター [Ultra Light.NET]321 ページ
- ULCommand クラス [Ultra Light.NET]66 ページ

例

次のコードでは、値が 3 である ULParameter パラメーターが作成され、cmd という ULCommand が追加されます。

```
' Visual Basic
Dim p As ULParameter = New ULParameter
p.Value = 3
cmd.Parameters.Add( p )
```

対応する C# 言語のコードを次に示します。

```
// C#
ULParameter p = new ULParameter();
p.Value = 3;
cmd.Parameters.Add( p );
```

ULParameter(string, object) コンストラクター

ULParameter オブジェクトを、指定されたパラメーター名と値で初期化します。

Visual Basic 構文

```
Public Sub New(ByVal parameterName As String, ByVal value As Object)
```

C# 構文

```
public ULParameter(string parameterName, object value)
```

パラメーター

- **parameterName**　パラメーターの名前。名前のないパラメーターの場合、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand はパラメーターの名前を使用しません。
- **value**　パラメーターの値である System.Object。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

定数 0 と 0.0 の特別な処理と、オーバロードされたメソッドの解決方法のため、このコンストラクターを使用するときは、定数値を型オブジェクトに明示的にキャストすることを強くおすすめします。

参照

- ULParameter.ULParameter コンストラクター [Ultra Light.NET]321 ページ
- ULCommand クラス [Ultra Light.NET]66 ページ
- System.Object

例

次のコードでは、値が 0 である ULParameter パラメーターが作成され、cmd という ULCommand が追加されます。

```
' Visual Basic
cmd.Parameters.Add( New ULParameter( "", CType( 0, Object ) ) )
```

対応する C# 言語のコードを次に示します。

```
// C#
cmd.Parameters.Add( new ULParameter( "", (object)0 ) );
```

ULParameter(string, ULDbType) コンストラクター

ULParameter オブジェクトを、指定されたパラメーター名とデータ型で初期化します。

Visual Basic 構文

```
Public Sub New(ByVal parameterName As String, ByVal dbType As ULDbType)
```

C# 構文

```
public ULParameter(string parameterName, ULDbType dbType)
```

パラメーター

- **parameterName**　パラメーターの名前。名前のないパラメーターの場合、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand はパラメーターの名前を使用しません。

- **dbType**　iAnywhere.Data.UltraLite.ULDbType 値の 1 つ。

備考

このコンストラクターの使用はおすすめしません。これは、他のデータプロバイダーとの互換性のために用意されています。

Ultra Light.NET では、パラメーターは IN パラメーターとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value だけです。

参照

- ULParameter.ULParameter コンストラクター [Ultra Light.NET]321 ページ
- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ
- ULCommand クラス [Ultra Light.NET]66 ページ

ULParameter(string, ULDbType, int) コンストラクター

ULParameter オブジェクトを、指定されたパラメーター名とデータ型で初期化します。

Visual Basic 構文

```
Public Sub New(
     ByVal parameterName As String,
    ByVal dbType As ULDbType,
    ByVal size As Integer
)
```

C# 構文

```
public ULParameter(string parameterName, ULDbType dbType, int size)
```

パラメーター

- **parameterName** パラメーターの名前。名前のないパラメーターの場合、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand はパラメーターの名前を使用しません。
- **dbType** iAnywhere.Data.UltraLite.ULDbType 値の 1 つ。
- **size** パラメーターの長さ。

備考

このコンストラクターの使用はおすすめしません。これは、他のデータプロバイダーとの互換性のために用意されています。

Ultra Light.NET では、パラメーターは IN パラメーターとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value だけです。

参照

- ULParameter.ULParameter コンストラクター [Ultra Light.NET]321 ページ
- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ

ULParameter(string, ULDbType, int, ParameterDirection, bool, byte, byte, string, DataRowVersion, object) コンストラクター

指定されたパラメーター名、データ型、長さ、方向、NULL 入力属性、数値精度、数値の位取り、ソースカラム、ソースのバージョン、値で、ULParameter オブジェクトを初期化します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
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# 構文

```
public ULParameter(
     string parameterName,
    ULDbType dbType,
    int size,
    ParameterDirection direction,
    bool isNullable,
    byte precision,
    byte scale,
    string sourceColumn,
    DataRowVersion sourceVersion,
    object value
)
```

パラメーター

- **parameterName** パラメーターの名前。名前のないパラメーターの場合、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand はパラメーターの名前を使用しません。
- **dbType** iAnywhere.Data.UltraLite.ULDbType 値の 1 つ。
- **size** パラメーターの長さ。
- **direction** System.Data.ParameterDirection 値の 1 つ。
- **isNullable** フィールドの値を NULL にできる場合は true、できない場合は false。
- **precision** Value が解決される小数点の左右の桁の合計数。
- **scale** Value が解決される小数点までの桁の合計数。
- **sourceColumn** マッピングするソースカラムの名前。
- **sourceVersion** System.Data.DataRowVersion 値の 1 つ。
- **value** パラメーターの値である System.Object。

例外

- **ULException クラス**　Ultra Light.NET では、System.Data.ParameterDirection.Input 方向しかサポートされていません。

備考

このコンストラクターの使用はおすすめしません。これは、他のデータプロバイダーとの互換性のために用意されています。

参照

- ULParameter.ULParameter コンストラクター [Ultra Light.NET]321 ページ
- ULCommand クラス [Ultra Light.NET]66 ページ
- System.Data.ParameterDirection
- System.Data.DataRowVersion
- System.Object
- System.Data.ParameterDirection.Input

ULParameter(string, ULDbType, int, string) コンストラクター

ULParameter オブジェクトを、指定されたパラメーター名、データ型、長さで初期化します。

Visual Basic 構文

```
Public Sub New(
     ByVal parameterName As String,
    ByVal dbType As ULDbType,
    ByVal size As Integer,
    ByVal sourceColumn As String
)
```

C# 構文

```
public ULParameter(
     string parameterName,
    ULDbType dbType,
    int size,
    string sourceColumn
)
```

パラメーター

- **parameterName**　パラメーターの名前。名前のないパラメーターの場合、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand はパラメーターの名前を使用しません。
- **dbType**　iAnywhere.Data.UltraLite.ULDbType 値の 1 つ。
- **size**　パラメーターの長さ。
- **sourceColumn**　マッピングするソースカラムの名前。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

このコンストラクターの使用はおすすめしません。これは、他のデータプロバイダーとの互換性のために用意されています。

Ultra Light.NET では、パラメーターは IN パラメーターとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value だけです。

参照

- ULParameter.ULParameter コンストラクター [Ultra Light.NET]321 ページ
- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ
- ULCommand クラス [Ultra Light.NET]66 ページ

ResetDbType メソッド

このメソッドは Ultra Light.NET ではサポートされていません。

Visual Basic 構文

```
Public Overrides Sub ResetDbType()
```

C# 構文

```
public override void ResetDbType()
```

備考

Ultra Light.NET では、パラメーターは IN パラメーターとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value だけです。

参照

- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ

ToString メソッド

このインスタンスの文字列表記を返します。

Visual Basic 構文

```
Public Overrides Function ToString() As String
```

C# 構文

```
public override string ToString()
```

戻り値

パラメーターの名前。

DbType プロパティ

パラメーターの System.Data.DbType を指定します。

Visual Basic 構文

```
Public Overrides Property DbType As DbType
```

C# 構文

```
public override DbType DbType {get;set;}
```

例外

- **ArgumentException**　指定された値から iAnywhere.Data.UltraLite.ULDbType へのマッピングは存在しないため、指定された値はサポートされません。

備考

System.Data.DbType 値の 1 つ。

ULParameter.ULDbType と DbType プロパティはリンクされます。このため、DbType を設定すると、ULParameter.ULDbType がサポートされている iAnywhere.Data.UltraLite.ULDbType に変更されます。

参照

- ULParameter.ULDbType プロパティ [Ultra Light.NET]333 ページ
- System.Data.DbType

Direction プロパティ

パラメーターが入力専用、出力専用、双方向性、またはストアドプロシージャー戻り値パラメーターのいずれであるかを示す値です。

Visual Basic 構文

```
Public Overrides Property Direction As ParameterDirection
```

C# 構文

```
public override ParameterDirection Direction {get;set;}
```

例外

- **ULException クラス**　Ultra Light.NET では、System.Data.ParameterDirection.Input 方向しかサポートされていません。

備考

System.Data.ParameterDirection 値の 1 つ。

Ultra Light.NET では、パラメーターは IN パラメーターとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value だけです。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ
- System.Data.ParameterDirection
- System.Data.ParameterDirection.Input

IsNullable プロパティ

パラメーターが NULL 値を受け入れるかどうかを指定します。

Visual Basic 構文

```
Public Overrides Property IsNullable As Boolean
```

C# 構文

```
public override bool IsNullable {get;set;}
```

備考

NULL 値を受け入れる場合は true、受け入れない場合は false。デフォルトは false です。NULL 値は DBNull クラスを使用して処理されます。

Ultra Light.NET では、パラメーターは IN パラメーターとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value だけです。

参照

- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ

Offset プロパティ

ULParameter.Value のオフセットを指定します。

Visual Basic 構文

```
Public Property Offset As Integer
```

C# 構文

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

備考

値のオフセット。デフォルトは 0 です。

Ultra Light.NET では、パラメーターは IN パラメーターとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value だけです。

参照

- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ

ParameterName プロパティ

パラメーターの名前を指定します。

Visual Basic 構文

```
Public Overrides Property ParameterName As String
```

C# 構文

```
public override string ParameterName {get;set;}
```

備考

パラメーターの名前を表す文字列。名前のないパラメーターの場合は空の文字列 ("")。NULL 参照 (Visual Basic の Nothing) を指定すると、空の文字列が使用されます。

Ultra Light.NET では、ULCommand はパラメーターの名前を使用しません。すべてのパラメーターは、位置パラメーターとして扱われ、コマンドによって追加された順序で使用されます。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ

Precision プロパティ

ULParameter.Value を表すために使用される最大桁数を指定します。

Visual Basic 構文

```
Public Property Precision As Byte
```

C# 構文

```
public byte Precision {get;set;}
```

例外

- **ArgumentException**　値が 38 を超えています。

備考

ULParameter.Value を表すために使用される最大桁数。デフォルト値は 0 です。これは、データプロバイダーが ULParameter.Value の精度を設定することを示します。

Ultra Light.NET では、パラメーターは IN パラメーターとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value だけです。

参照

- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Scale プロパティ

ULParameter.Value が解決される小数点の桁の数を指定します。

Visual Basic 構文

```
Public Property Scale As Byte
```

C# 構文

```
public byte Scale {get;set;}
```

備考

ULParameter.Value が解決される小数点の桁の数。デフォルトは 0 です。

Ultra Light.NET では、パラメーターは IN パラメーターとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value だけです。

参照

- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ

Size プロパティ

カラム内のデータの最大サイズを指定します。

Visual Basic 構文

```
Public Overrides Property Size As Integer
```

C# 構文

```
public override int Size {get;set;}
```

備考

カラム内のデータの最大サイズ。デフォルト値はパラメーター値から推測されます。Size プロパティは、バイナリと文字列型に対して使用されます。

Ultra Light.NET では、パラメーターは IN パラメーターとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value だけです。

参照

- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ

SourceColumn プロパティ

DataSet にマッピングされ、値をロードしたり返したりするときに使用するソースカラムの名前を指定します。

Visual Basic 構文

```
Public Overrides Property SourceColumn As String
```

C# 構文

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

備考

DataSet にマッピングされ、値をロードしたり返したりするときに使用するソースカラムの名前を指定する文字列。

Ultra Light.NET では、パラメーターは IN パラメーターとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value だけです。

参照

- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ

SourceColumnNullMapping プロパティ

ソースカラムが NULL 入力可であるかどうかを指定します。

Visual Basic 構文

```
Public Overrides Property SourceColumnNullMapping As Boolean
```

C# 構文

```
public override bool SourceColumnNullMapping {get;set;}
```

備考

ソースカラムが NULL 入力可の場合は true、そうでない場合は false。

Ultra Light.NET では、パラメーターは IN パラメーターとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value だけです。

参照

- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ

SourceVersion プロパティ

ULParameter.Value をロードするときに使用する System.Data.DataRowVersion を指定します。

Visual Basic 構文

```
Public Overrides Property SourceVersion As DataRowVersion
```

C# 構文

```
public override DataRowVersion SourceVersion {get;set;}
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ
- System.Data.DataRowVersion

ULDbType プロパティ

パラメーターの iAnywhere.Data.UltraLite.ULDbType を指定します。

Visual Basic 構文

```
Public Property ULDbType As ULDbType
```

C# 構文

```
public ULDbType ULDbType {get;set;}
```

備考

iAnywhere.Data.UltraLite.ULDbType 値の 1 つ。

ULDbType と ULParameter.DbType はリンクされます。このため、ULDbType を設定すると、ULParameter.DbType がサポートされている System.Data.DbType を変更します。

Ultra Light.NET では、パラメーターは IN パラメーターとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value だけです。

参照

- ULParameter.Value プロパティ [Ultra Light.NET]333 ページ
- ULParameter.DbType プロパティ [Ultra Light.NET]328 ページ
- System.Data.DbType

Value プロパティ

パラメーターの値を指定します。

Visual Basic 構文

```
Public Overrides Property Value As Object
```

C# 構文

```
public override object Value {get;set;}
```

備考

パラメーターの値を指定する System.Object。

値は、型変換やマッピングは行われずに、データプロバイダーにそのまま送信されます。コマンドが実行されると、コマンドは必要な型に値を変換しますが、値を変換できない場合は ULSQLCode.SQLE_CONVERSION_ERROR とともに ULException を通知します。

参照

- ULException クラス [Ultra Light.NET]274 ページ
- System.Object

ULParameterCollection クラス

ULCommand のすべてのパラメーターを表します。

Visual Basic 構文

```
Public NotInheritable Class ULParameterCollection
    Inherits System.Data.Common.DbParameterCollection
```

C# 構文

```
public sealed class ULParameterCollection :
    System.Data.Common.DbParameterCollection
```

基本クラス

- System.Data.Common.DbParameterCollection

メンバー

継承されたメンバーを含む ULParameterCollection クラスのすべてのメンバー。

名前	説明
Add メソッド	ULParameter をコレクションに追加します。
AddRange メソッド	ULParameterCollection の末尾に値の配列を追加します。
Clear メソッド	すべてのパラメーターをコレクションから削除します。
Contains メソッド	ULParameter がコレクション内に存在するかどうかをチェックします。
CopyTo メソッド	ULParameter オブジェクトを ULParameterCollection から指定された配列にコピーします。
GetEnumerator メソッド	コレクションの列挙子を返します。
GetParameter method (System.Data.Common.DbParameterCollection から継承)	コレクションで指定されたインデックスに System.Data.Common.DbParameter オブジェクトを返します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
IndexOf メソッド	コレクション内の ULParameter のロケーションを返します。
Insert メソッド	コレクション内の指定されたインデックス位置に ULParameter を挿入します。
Remove メソッド	ULParameter をコレクションから削除します。
RemoveAt メソッド	コレクションの指定されたインデックス位置のパラメーターを削除します。
SetParameter method (System.Data.Common.DbParameterCollection から継承)	指定されたインデックスにある System.Data.Common.DbParameter オブジェクトに新しい値を設定します。
Count プロパティ	コレクション内の ULParameter オブジェクトの数を返します。
IsFixedSize プロパティ	ULParameterCollection のサイズが固定かどうかを示します。
IsReadOnly プロパティ	ULParameterCollection が読み取り専用であるかどうかを示します。
IsSynchronized プロパティ	ULParameterCollection が同期されるかどうかを示します。
SyncRoot プロパティ	SAParameterCollection へのアクセスを同期するために使用するオブジェクトを返します。
this プロパティ	指定されたインデックスの ULParameter を返します。

備考

コレクション内のすべてのパラメーターは、位置パラメーターとして扱われ、ULCommand.CommandText の疑問符のプレースホルダーと同じ順序で指定されます。たとえば、コレクション内の最初のパラメーターは、SQL 文の最初の疑問符に対応し、コレクション内の 2 番目のパラメーターは、SQL 文の 2 番目の疑問符に対応します。ULCommand.CommandText 内の疑問符の数は、少なくともコレクション内のパラメーターの数と同じでなければなりません。パラメーターが不足している場合は、NULL が代用されます。

ULParameterCollection にはコンストラクターがありません。ULParameterCollection は、ULCommand.Parameters から取得します。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand.Parameters プロパティ [Ultra Light.NET]101 ページ
- System.Data.Common.DbParameterCollection
- System.Data.IDataParameterCollection

Add メソッド

ULParameter をコレクションに追加します。

オーバーロードリスト

名前	説明
Add(object) メソッド	ULParameter をコレクションに追加します。
Add(string, object) メソッド	指定されたパラメーター名と値を使用して作成された新しい ULParameter をコレクションに追加します。
Add(string, ULDbType) メソッド	指定されたパラメーター名とデータ型を使用して作成された新しい ULParameter をコレクションに追加します。
Add(string, ULDbType, int) メソッド	指定されたパラメーター名、データ型、長さを使用して作成された新しい ULParameter をコレクションに追加します。
Add(string, ULDbType, int, string) メソッド	指定されたパラメーター名、データ型、長さ、ソースカラム名を使用して作成された新しい ULParameter をコレクションに追加します。
Add(ULParameter) メソッド	ULParameter をコレクションに追加します。

Add(object) メソッド

ULParameter をコレクションに追加します。

Visual Basic 構文

```
Public Overrides Function Add(ByVal value As Object) As Integer
```

C# 構文

```
public override int Add(object value)
```


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

パラメーター

- **value** コレクションに追加される ULParameter オブジェクト。

戻り値

新しい ULParameter オブジェクトのインデックス。

例外

- **ArgumentNullException** 値を null (Visual Basic の Nothing) にすることはできません。
- **InvalidCastException** 指定される値は ULParameter である必要があります。
- **ArgumentException** ULParameter オブジェクトは、一度しかコレクションに追加できません。

備考

コレクション内のすべてのパラメーターは、位置パラメーターとして扱われ、ULCommand.CommandText の疑問符のプレースホルダーと同じ順序で指定する必要があります。たとえば、コレクション内の最初のパラメーターは、SQL 文の最初の疑問符に対応し、コレクション内の 2 番目のパラメーターは、SQL 文の 2 番目の疑問符に対応します。ULCommand.CommandText 内の疑問符の数は、少なくともコレクション内のパラメーターの数と同じでなければなりません。パラメーターが不足している場合は、NULL が代用されます。

参照

- ULParameterCollection.Add メソッド [Ultra Light.NET]336 ページ
- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULParameter クラス [Ultra Light.NET]319 ページ

Add(string, object) メソッド

指定されたパラメーター名と値を使用して作成された新しい ULParameter をコレクションに追加します。

Visual Basic 構文

```
Public Function Add(
     ByVal parameterName As String,
    ByVal value As Object
) As ULParameter
```

C# 構文

```
public ULParameter Add(string parameterName, object value)
```

パラメーター

- **parameterName** パラメーターの名前。名前のないパラメーターの場合、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand はパラメーターの名前を使用しません。
- **value** パラメーターの値である System.Object。

戻り値

新しい ULParameter オブジェクト。

備考

コレクション内のすべてのパラメーターは、位置パラメーターとして扱われ、ULCommand.CommandText の疑問符のプレースホルダーと同じ順序で指定する必要があります。たとえば、コレクション内の最初のパラメーターは、SQL 文の最初の疑問符に対応し、コレクション内の 2 番目のパラメーターは、SQL 文の 2 番目の疑問符に対応します。ULCommand.CommandText 内の疑問符の数は、少なくともコレクション内のパラメーターの数と同じでなければなりません。パラメーターが不足している場合は、NULL が代用されます。

定数 0 と 0.0 の特別な処理と、オーバロードされたメソッドの解決方法のため、このメソッドを使用するときは、定数値を型オブジェクトに明示的にキャストすることを強くおすすめします。

参照

- ULParameterCollection.Add メソッド [Ultra Light.NET]336 ページ
- ULParameter クラス [Ultra Light.NET]319 ページ
- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand クラス [Ultra Light.NET]66 ページ
- System.Object

例

次のコードでは、値が 0 である ULParameter パラメーターが cmd という ULCommand に追加されます。

```
' Visual Basic
cmd.Parameters.Add( "", CType( 0, Object ) )
```

対応する C# 言語のコードを次に示します。

```
// C#
cmd.Parameters.Add( "", (object)0 );
```

Add(string, ULDbType) メソッド

指定されたパラメーター名とデータ型を使用して作成された新しい ULParameter をコレクションに追加します。

Visual Basic 構文

```
Public Function Add(
     ByVal parameterName As String,
    ByVal ulDbType As ULDbType
) As ULParameter
```

C# 構文

```
public ULParameter Add(string parameterName, ULDbType ulDbType)
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

パラメーター

- **parameterName** パラメーターの名前。名前のないパラメーターの場合、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand はパラメーターの名前を使用しません。
- **ulDbType** iAnywhere.Data.UltraLite.ULDbType 値の 1 つ。

戻り値

新しい ULParameter オブジェクト。

備考

コレクション内のすべてのパラメーターは、位置パラメーターとして扱われ、ULCommand.CommandText の疑問符のプレースホルダーと同じ順序で指定する必要があります。たとえば、コレクション内の最初のパラメーターは、SQL 文の最初の疑問符に対応し、コレクション内の 2 番目のパラメーターは、SQL 文の 2 番目の疑問符に対応します。ULCommand.CommandText 内の疑問符の数は、少なくともコレクション内のパラメーターの数と同じでなければなりません。パラメーターが不足している場合は、NULL が代用されます。

参照

- ULParameterCollection.Add メソッド [Ultra Light.NET]336 ページ
- ULParameter クラス [Ultra Light.NET]319 ページ
- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand クラス [Ultra Light.NET]66 ページ

Add(string, ULDbType, int) メソッド

指定されたパラメーター名、データ型、長さを使用して作成された新しい ULParameter をコレクションに追加します。

Visual Basic 構文

```
Public Function Add(
     ByVal parameterName As String,
    ByVal ulDbType As ULDbType,
    ByVal size As Integer
) As ULParameter
```

C# 構文

```
public ULParameter Add(
     string parameterName,
    ULDbType ulDbType,
    int size
)
```

パラメーター

- **parameterName** パラメーターの名前。名前のないパラメーターの場合、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand はパラメーターの名前を使用しません。

- **ulDbType** iAnywhere.Data.UltraLite.ULDbType 値の 1 つ。
- **size** パラメーターの長さ。

戻り値

新しい ULParameter オブジェクト。

備考

コレクション内のすべてのパラメーターは、位置パラメーターとして扱われ、ULCommand.CommandText の疑問符のプレースホルダーと同じ順序で指定する必要があります。たとえば、コレクション内の最初のパラメーターは、SQL 文の最初の疑問符に対応し、コレクション内の 2 番目のパラメーターは、SQL 文の 2 番目の疑問符に対応します。ULCommand.CommandText 内の疑問符の数は、少なくともコレクション内のパラメーターの数と同じでなければなりません。パラメーターが不足している場合は、NULL が代用されます。

参照

- ULParameterCollection.Add メソッド [Ultra Light.NET]336 ページ
- ULParameter クラス [Ultra Light.NET]319 ページ
- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand クラス [Ultra Light.NET]66 ページ

Add(string, ULDbType, int, string) メソッド

指定されたパラメーター名、データ型、長さ、ソースカラム名を使用して作成された新しい ULParameter をコレクションに追加します。

Visual Basic 構文

```
Public Function Add(
     ByVal parameterName As String,
    ByVal ulDbType As ULDbType,
    ByVal size As Integer,
    ByVal sourceColumn As String
) As ULParameter
```

C# 構文

```
public ULParameter Add(
     string parameterName,
    ULDbType ulDbType,
    int size,
    string sourceColumn
)
```

パラメーター

- **parameterName** パラメーターの名前。名前のないパラメーターの場合、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand はパラメーターの名前を使用しません。
- **ulDbType** iAnywhere.Data.UltraLite.ULDbType 値の 1 つ。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- **size**　パラメーターの長さ。
- **sourceColumn**　マッピングするソースカラムの名前。

戻り値

新しい ULParameter オブジェクト。

備考

コレクション内のすべてのパラメーターは、位置パラメーターとして扱われ、ULCommand.CommandText の疑問符のプレースホルダーと同じ順序で指定する必要があります。たとえば、コレクション内の最初のパラメーターは、SQL 文の最初の疑問符に対応し、コレクション内の 2 番目のパラメーターは、SQL 文の 2 番目の疑問符に対応します。ULCommand.CommandText 内の疑問符の数は、少なくともコレクション内のパラメーターの数と同じでなければなりません。パラメーターが不足している場合は、NULL が代用されます。

参照

- ULParameterCollection.Add メソッド [Ultra Light.NET]336 ページ
- ULParameter クラス [Ultra Light.NET]319 ページ
- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ
- ULCommand クラス [Ultra Light.NET]66 ページ

Add(ULParameter) メソッド

ULParameter をコレクションに追加します。

Visual Basic 構文

```
Public Function Add(ByVal value As ULParameter) As ULParameter
```

C# 構文

```
public ULParameter Add(ULParameter value)
```

パラメーター

- **value**　コレクションに追加される ULParameter オブジェクト。

戻り値

新しい ULParameter オブジェクト。

例外

- **ArgumentNullException**　値を null (Visual Basic の Nothing) にすることはできません。
- **ArgumentException**　ULParameter オブジェクトは、一度しかコレクションに追加できません。

備考

コレクション内のすべてのパラメーターは、位置パラメーターとして扱われ、ULCommand.CommandText の疑問符のプレースホルダーと同じ順序で指定する必要があります。

たとえば、コレクション内の最初のパラメーターは、SQL 文の最初の疑問符に対応し、コレクション内の 2 番目のパラメーターは、SQL 文の 2 番目の疑問符に対応します。
ULCommand.CommandText 内の疑問符の数は、少なくともコレクション内のパラメーターの数と同じでなければなりません。パラメーターが不足している場合は、NULL が代用されます。

参照

- ULParameterCollection.Add メソッド [Ultra Light.NET]336 ページ
- ULParameter クラス [Ultra Light.NET]319 ページ
- ULCommand.CommandText プロパティ [Ultra Light.NET]98 ページ

AddRange メソッド

ULParameterCollection の末尾に値の配列を追加します。

オーバーロードリスト

名前	説明
AddRange(Array) メソッド	ULParameterCollection の末尾に値の配列を追加します。
AddRange(ULParameter[]) メソッド	ULParameterCollection の末尾に値の配列を追加します。

AddRange(Array) メソッド

ULParameterCollection の末尾に値の配列を追加します。

Visual Basic 構文

```
Public Overrides Sub AddRange(ByVal values As Array)
```

C# 構文

```
public override void AddRange(Array values)
```

パラメーター

- **values** このコレクションの末尾に追加する ULParameter オブジェクトの配列。

参照

- ULParameter クラス [Ultra Light.NET]319 ページ

AddRange(ULParameter[]) メソッド

ULParameterCollection の末尾に値の配列を追加します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public Sub AddRange(ByVal values As ULParameter())
```

C# 構文

```
public void AddRange(ULParameter[] values)
```

パラメーター

- **values** このコレクションの末尾に追加する ULParameter オブジェクトの配列。

備考

これは、DbParameterCollection.AddRange(Array) が厳密に型指定されたものです。

参照

- ULParameter クラス [Ultra Light.NET]319 ページ
- ULParameterCollection クラス [Ultra Light.NET]334 ページ

Clear メソッド

すべてのパラメーターをコレクションから削除します。

Visual Basic 構文

```
Public Overrides Sub Clear()
```

C# 構文

```
public override void Clear()
```

Contains メソッド

ULParameter がコレクション内に存在するかどうかをチェックします。

オーバーロードリスト

名前	説明
Contains(object) メソッド	ULParameter がコレクション内に存在するかどうかをチェックします。
Contains(string) メソッド	指定された名前の ULParameter がコレクション内に存在するかどうかをチェックします。

Contains(object) メソッド

ULParameter がコレクション内に存在するかどうかをチェックします。

Visual Basic 構文

```
Public Overrides Function Contains(ByVal value As Object) As Boolean
```

C# 構文

```
public override bool Contains(object value)
```

パラメーター

- **value** チェック対象の ULParameter オブジェクト。

戻り値

コレクションに、その ULParameter がある場合は true、ない場合は false。

参照

- ULParameterCollection.Contains メソッド [Ultra Light.NET]343 ページ
- ULParameter クラス [Ultra Light.NET]319 ページ

Contains(string) メソッド

指定された名前の ULParameter がコレクション内に存在するかどうかをチェックします。

Visual Basic 構文

```
Public Overrides Function Contains(ByVal value As String) As Boolean
```

C# 構文

```
public override bool Contains(string value)
```

パラメーター

- **value** 検索対象のパラメーターの名前。

戻り値

コレクションに、その ULParameter がある場合は true、ない場合は false。

参照

- ULParameterCollection.Contains メソッド [Ultra Light.NET]343 ページ
- ULParameter クラス [Ultra Light.NET]319 ページ

CopyTo メソッド

ULParameter オブジェクトを ULParameterCollection から指定された配列にコピーします。

Visual Basic 構文

```
Public Overrides Sub CopyTo(
     ByVal array As Array,
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
    ByVal index As Integer
)
```

C# 構文

```
public override void CopyTo(Array array, int index)
```

パラメーター

- **array** ULParameter オブジェクトのコピー先の配列。
- **index** 配列の開始インデックス。

参照

- ULParameter クラス [Ultra Light.NET]319 ページ

GetEnumerator メソッド

コレクションの列挙子を返します。

Visual Basic 構文

```
Public Overrides Function GetEnumerator()
    As System.Collections.IEnumerator
```

C# 構文

```
public override IEnumerator GetEnumerator()
```

戻り値

コレクション内のパラメーターを列挙する ArrayList 列挙子。

IndexOf メソッド

コレクション内の ULParameter のロケーションを返します。

オーバーロードリスト

名前	説明
IndexOf(object) メソッド	コレクション内の ULParameter のロケーションを返します。
IndexOf(string) メソッド	指定された名前を持つ ULParameter のコレクション内のロケーションを返します。

IndexOf(object) メソッド

コレクション内の ULParameter のロケーションを返します。

Visual Basic 構文

```
Public Overrides Function IndexOf(ByVal value As Object) As Integer
```

C# 構文

```
public override int IndexOf(object value)
```

パラメーター

- **value** 検索する ULParameter オブジェクト。

戻り値

コレクション内の ULParameter の 0 から始まるインデックス。パラメーターが見つからない場合は -1。

例外

- **InvalidCastException** 指定される値は ULParameter である必要があります。

参照

- ULParameterCollection.IndexOf メソッド [Ultra Light.NET]345 ページ
- ULParameter クラス [Ultra Light.NET]319 ページ

IndexOf(string) メソッド

指定された名前を持つ ULParameter のコレクション内のロケーションを返します。

Visual Basic 構文

```
Public Overrides Function IndexOf(
     ByVal parameterName As String
) As Integer
```

C# 構文

```
public override int IndexOf(string parameterName)
```

パラメーター

- **parameterName** 検索するパラメーターの名前。

戻り値

コレクション内の ULParameter の 0 から始まるインデックス。パラメーターが見つからない場合は -1。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULParameterCollection.IndexOf メソッド [Ultra Light.NET]345 ページ
- ULParameter クラス [Ultra Light.NET]319 ページ

Insert メソッド

コレクション内の指定されたインデックス位置に ULParameter を挿入します。

Visual Basic 構文

```
Public Overrides Sub Insert(
     ByVal index As Integer,
    ByVal value As Object
)
```

C# 構文

```
public override void Insert(int index, object value)
```

パラメーター

- **index**　コレクション内にパラメーターを挿入するロケーションの 0 から始まるインデックス。
- **value**　挿入する ULParameter オブジェクト。

例外

- **IndexOutOfRangeException**　インデックスは無効です。
- **ArgumentNullException**　null 参照 (Visual Basic の Nothing) を使用してパラメーターを設定することはできません。
- **InvalidCastException**　指定される値は ULParameter である必要があります。

参照

- ULParameter クラス [Ultra Light.NET]319 ページ

Remove メソッド

ULParameter をコレクションから削除します。

Visual Basic 構文

```
Public Overrides Sub Remove(ByVal value As Object)
```

C# 構文

```
public override void Remove(object value)
```

パラメーター

- **value**　削除する ULParameter オブジェクト。

例外

- **ArgumentNullException**　null 参照 (Visual Basic の Nothing) を使用してパラメーターを設定することはできません。
- **InvalidCastException**　指定される値は ULParameter である必要があります。
- **ArgumentException**　コレクション内に、指定されたパラメーターがありません。

参照

- ULParameter クラス [Ultra Light.NET]319 ページ

RemoveAt メソッド

コレクションの指定されたインデックス位置のパラメーターを削除します。

オーバーロードリスト

名前	説明
RemoveAt(int) メソッド	コレクションの指定されたインデックス位置のパラメーターを削除します。
RemoveAt(string) メソッド	指定された名前のパラメーターをコレクションから削除します。

RemoveAt(int) メソッド

コレクションの指定されたインデックス位置のパラメーターを削除します。

Visual Basic 構文

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

C# 構文

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

パラメーター

- **index**　削除するパラメーターの 0 から始まるインデックス。値は、[0,ULParameterCollection.Count-1] の範囲内であることが必要です。コレクションの最初のパラメーターのインデックス値は 0 です。

例外

- **IndexOutOfRangeException**　インデックスは無効です。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULParameterCollection.RemoveAt メソッド [Ultra Light.NET]348 ページ
- ULParameterCollection.Count プロパティ [Ultra Light.NET]349 ページ

RemoveAt(string) メソッド

指定された名前のパラメーターをコレクションから削除します。

Visual Basic 構文

```
Public Overrides Sub RemoveAt(ByVal parameterName As String)
```

C# 構文

```
public override void RemoveAt(string parameterName)
```

パラメーター

- **parameterName**　取り出すパラメーターの名前。

例外

- **IndexOutOfRangeException**　指定した名前のパラメーターがありません。

参照

- ULParameterCollection.RemoveAt メソッド [Ultra Light.NET]348 ページ

Count プロパティ

コレクション内の ULParameter オブジェクトの数を返します。

Visual Basic 構文

```
Public ReadOnly Overrides Property Count As Integer
```

C# 構文

```
public override int Count {get;}
```

備考

コレクション内の ULParameter オブジェクトの数。

IsFixedSize プロパティ

ULParameterCollection のサイズが固定かどうかを示します。

Visual Basic 構文

```
Public ReadOnly Overrides Property IsFixedSize As Boolean
```

C# 構文

```
public override bool IsFixedSize {get;}
```

備考

コレクションのサイズが固定の場合は true、そうでない場合は false。

IsReadOnly プロパティ

ULParameterCollection が読み取り専用であるかどうかを示します。

Visual Basic 構文

```
Public ReadOnly Overrides Property IsReadOnly As Boolean
```

C# 構文

```
public override bool IsReadOnly {get;}
```

備考

コレクションが読み込み専用の場合は true、そうでない場合は false。

IsSynchronized プロパティ

ULParameterCollection が同期されるかどうかを示します。

Visual Basic 構文

```
Public ReadOnly Overrides Property IsSynchronized As Boolean
```

C# 構文

```
public override bool IsSynchronized {get;}
```

備考

コレクションが同期している場合は true、そうでない場合は false。

SyncRoot プロパティ

SAParameterCollection へのアクセスを同期するために使用するオブジェクトを返します。

Visual Basic 構文

```
Public ReadOnly Overrides Property SyncRoot As Object
```

C# 構文

```
public override object SyncRoot {get;}
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

このコレクションへのアクセスの同期に使用されるオブジェクト。

this プロパティ

指定されたインデックスの ULParameter を返します。

オーバーロードリスト

名前	説明
this[int] プロパティ	指定されたインデックスの ULParameter を返します。
this[string] プロパティ	指定された名前の ULParameter を返します。

this[int] プロパティ

指定されたインデックスの ULParameter を返します。

Visual Basic 構文

```
Public Shadows Property Item(ByVal index As Integer) As ULParameter
```

C# 構文

```
public new ULParameter this[int index] {get;set;}
```

パラメーター

- **index**　取り出すパラメーターの 0 から始まるインデックス。値は、[0,ULParameterCollection.Count-1] の範囲内であることが必要です。コレクションの最初のパラメーターのインデックス値は 0 です。

戻り値

指定されたインデックス位置の ULParameter。

例外

- **IndexOutOfRangeException**　インデックスは無効です。

備考

C# では、このプロパティは ULParameterCollection クラスのインデクサです。

これは、DbParameterCollection.this[int] が厳密に型指定されたものです。

参照

- ULParameter クラス [Ultra Light.NET]319 ページ

this[string] プロパティ

指定された名前の ULParameter を返します。

Visual Basic 構文

```
Public Shadows Property Item(
     ByVal parameterName As String
) As ULParameter
```

C# 構文

```
public new ULParameter this[string parameterName] {get;set;}
```

パラメーター

- **parameterName** 取り出すパラメーターの名前。

戻り値

指定された名前の ULParameter。

例外

- **IndexOutOfRangeException** 指定した名前のパラメーターがありません。
- **ArgumentNullException** NULL (Visual Basic の Nothing) パラメーター名を使用してパラメーターを設定することはできません。

備考

C# では、このプロパティは ULParameterCollection クラスのインデクサです。

これは、DbParameterCollection.this[string] が厳密に型指定されたものです。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetValue メソッド [Ultra Light.NET]262 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULParameter クラス [Ultra Light.NET]319 ページ

ULResultSet クラス

UL 拡張： Ultra Light データベース内の編集可能な結果セットを示します。

Visual Basic 構文

```
Public Class ULResultSet Inherits ULDataReader
```

C# 構文

```
public class ULResultSet : ULDataReader
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

基本クラス

- ULDataReader クラス [Ultra Light.NET]236 ページ

派生クラス

- ULTable クラス [Ultra Light.NET]415 ページ

メンバー

継承されたメンバーを含む ULResultSet クラスのすべてのメンバー。

名前	説明
AppendBytes メソッド	指定された System.Bytes 配列の指定されたサブセットを、指定された ULDbType.LongBinary カラムの新しい値に追加します。
AppendChars メソッド	指定された System.Chars 配列の指定されたサブセットを、指定された ULDbType.LongVarchar カラムの新しい値に追加します。
Close メソッド	カーソルを閉じます。
Delete メソッド	現在の行を削除します。
Dispose method (System.Data.Common.DbDataReader から継承)	System.Data.Common.DbDataReader クラスの現在のインスタンスで使用しているすべてのリソースを解放します。
GetBoolean メソッド	指定されたカラムの値を System.Boolean として返します。
GetByte メソッド	指定されたカラムの値を符号なし 8 ビット値 (System.Byte) として返します。
GetBytes メソッド	**UL 拡張：** 指定されたカラムの値を System.Bytes 配列として返します。
GetChar メソッド	このメソッドは Ultra Light.NET ではサポートされていません。
GetChars メソッド	指定されたオフセットで始まる、指定された ULDbType.LongVarchar カラムの値のサブセットを、コピー先の System.Char 配列の指定されたオフセットにコピーします。

名前	説明
GetData method (System.Data.Common.DbDataReader から継承)	要求されたカラム序数の System.Data.Common.DbDataReader オブジェクトを返します。
GetDataTypeName メソッド	指定されたカラムのプロバイダーのデータ型の名前を返します。
GetDateTime メソッド	指定されたカラムの値を、ミリ秒の精度の System.DateTime として返します。
GetDbDataReader method (System.Data.Common.DbDataReader から継承)	プロバイダー固有の実装によって無効になる可能性のある、要求されたカラム序数の System.Data.Common.DbDataReader オブジェクトを返します。
GetDecimal メソッド	指定されたカラムの値を System.Decimal として返します。
GetDouble メソッド	指定されたカラムの値を System.Double として返します。
GetEnumerator メソッド	ULDataReader の反復処理を実行する System.Collections.IEnumerator を返します。
GetFieldType メソッド	指定されたカラムに最適な System.Type を返します。
GetFloat メソッド	指定されたカラムの値を System.Single として返します。
GetGuid メソッド	指定されたカラムの値を UUID (System.Guid) として返します。
GetInt16 メソッド	指定されたカラムの値を System.Int16 として返します。
GetInt32 メソッド	指定されたカラムの値を Int32 として返します。
GetInt64 メソッド	指定されたカラムの値を Int64 として返します。
GetName メソッド	指定されたカラムの名前を返します。
GetOrdinal メソッド	指定されたカラムのカラム ID を返します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
GetProviderSpecificFieldType method (System.Data.Common.DbDataReader から継承)	指定されたカラムのプロバイダー固有のフィールドタイプを返します。
GetProviderSpecificValue method (System.Data.Common.DbDataReader から継承)	指定されたカラムの値を System.Object のインスタンスとして取得します。
GetProviderSpecificValues method (System.Data.Common.DbDataReader から継承)	現在のローのコレクション内のプロバイダー固有のすべての属性カラムを取得します。
GetRowCount メソッド	**UL 拡張：** カーソル内のローの数を、スレッショルド以内で返します。
GetSchemaTable メソッド	ULDataReader のカラムのメタデータが記述された System.Data.DataTable を返します。
GetString メソッド	指定されたカラムの値を System.String として返します。
GetTimeSpan メソッド	指定されたカラムの値を、ミリ秒の精度の System.TimeSpan として返します。
GetUInt16 メソッド	指定されたカラムの値を System.UInt16 として返します。
GetUInt32 メソッド	指定されたカラムの値を UInt32 として返します。
GetUInt64 メソッド	指定されたカラムの値を System.UInt64 として返します。
GetValue メソッド	指定されたカラムの値をネイティブフォーマットで返します。
GetValues メソッド	現在のローのすべてのカラム値を返します。
IsDBNull メソッド	指定されたカラムの値が NULL かどうかをチェックします。
MoveAfterLast メソッド	**UL 拡張：** カーソルの最後のローの後に、カーソルを配置します。
MoveBeforeFirst メソッド	**UL 拡張：** カーソルの最初のローの前に、カーソルを配置します。
MoveFirst メソッド	**UL 拡張：** カーソルの最初のローに、カーソルを配置します。

名前	説明
MoveLast メソッド	**UL 拡張：** カーソルの最後のローに、カーソルを配置します。
MoveNext メソッド	**UL 拡張：** カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。
MovePrevious メソッド	**UL 拡張：** カーソルを前のローに配置するか、最初のローの前に配置します。
MoveRelative メソッド	**UL 拡張：** 現在のローを基準としてカーソルを配置します。
NextResult メソッド	バッチ SQL 文の結果を読み込むときに ULDataReader を次の結果に進めます。
Read メソッド	カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。
SetBoolean メソッド	指定されたカラムの値を、System.Boolean を使用して設定します。
SetByte メソッド	指定されたカラムの値を、System.Byte (符号なし 8 ビット整数) を使用して設定します。
SetBytes メソッド	指定されたカラムの値を、System.Bytes を使用して設定します。
SetDateTime メソッド	指定されたカラムの値を、System.DateTime を使用して設定します。
SetDBNull メソッド	カラムを NULL に設定します。
SetDecimal メソッド	指定されたカラムの値を、System.Decimal を使用して設定します。
SetDouble メソッド	指定されたカラムの値を、System.Double を使用して設定します。
SetFloat メソッド	指定されたカラムの値を、System.Single を使用して設定します。
SetGuid メソッド	指定されたカラムの値を、System.Guid を使用して設定します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
SetInt16 メソッド	指定されたカラムの値を、System.Int16 を使用して設定します。
SetInt32 メソッド	指定されたカラムの値を、System.Int32 を使用して設定します。
SetInt64 メソッド	指定されたカラムの値を、Int64 を使用して設定します。
SetString メソッド	指定されたカラムの値を、System.String を使用して設定します。
SetTimeSpan メソッド	指定されたカラムの値を、System.TimeSpan を使用して設定します。
SetToDefault メソッド	指定されたカラムの値を、そのデフォルト値に設定します。
SetUInt16 メソッド	指定されたカラムの値を、System.UInt16 を使用して設定します。
SetUInt32 メソッド	指定されたカラムの値を、System.UInt32 を使用して設定します。
SetUInt64 メソッド	指定されたカラムの値を、System.UInt64 を使用して設定します。
Update メソッド	現在のカラム値 (set メソッドを使用して指定されます) で新しいローを更新します。
UpdateBegin メソッド	現在のローを更新する準備を行います。
Depth プロパティ	現在のローのネストの深さを返します。
FieldCount プロパティ	このカーソル内のカラム数を返します。
HasRows プロパティ	ULDataReader に 1 つまたは複数のローがあるかどうかをチェックします。
IsBOF プロパティ	**UL 拡張：** 現在のローの位置が最初のローの前であるかどうかを確認します。
IsClosed プロパティ	カーソルが現在開いているかどうかを確認します。
IsEOF プロパティ	**UL 拡張：** 現在のローの位置が最後のローの後であるかどうかを確認します。

名前	説明
RecordsAffected プロパティ	SQL 文の実行によって変更、挿入、または削除されたローの数を返します。
RowCount プロパティ	**UL 拡張：** カーソル内のロー数を返します。
Schema プロパティ	**UL 拡張：** このカーソルのスキーマを保持します。
this プロパティ	指定されたカラムの値をネイティブフォーマットで返します。
VisibleFieldCount property (System.Data.Common.DbDataReader から継承)	System.Data.Common.DbDataReader の非表示でないフィールドの数を取得します。

備考

このクラスにはコンストラクターがありません。結果セットは、ULCommand クラスの ULCommand.ExecuteResultSet() メソッドを使用して作成されます。

```
' Visual Basic
Dim cmd As ULCommand = new ULCommand( _
    "SELECT emp_id FROM employee", conn _
)
Dim resultSet As ULResultSet = cmd.ExecuteResultSet()
```

対応する C# 言語のコードを次に示します。

```
// C#
ULCommand cmd = new ULCommand(
    "SELECT emp_id FROM employee", conn
);
ULResultSet resultSet = cmd.ExecuteResultSet();
```

ULResultSet オブジェクトは、位置付け更新や削除の実行対象となる編集可能な結果を表します。編集可能な結果セットには、ULCommand.ExecuteTable() または ULDataAdapter を使用します。

参照

- ULCommand.ExecuteResultSet メソッド [Ultra Light.NET]90 ページ
- ULCommand クラス [Ultra Light.NET]66 ページ
- ULResultSet クラス [Ultra Light.NET]352 ページ
- ULCommand.ExecuteTable メソッド [Ultra Light.NET]94 ページ
- ULDataAdapter クラス [Ultra Light.NET]209 ページ
- ULDataReader クラス [Ultra Light.NET]236 ページ
- System.Data.IDataReader
- System.Data.IDataRecord
- System.IDisposable

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

AppendBytes メソッド

指定された System.Bytes 配列の指定されたサブセットを、指定された ULDbType.LongBinary カラムの新しい値に追加します。

Visual Basic 構文

```
Public Sub AppendBytes(
     ByVal colID As Integer,
    ByVal val As Byte(),
    ByVal srcOffset As Integer,
    ByVal count As Integer
)
```

C# 構文

```
public void AppendBytes(int colID, byte[] val, int srcOffset, int count)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの現在の新しい値に追加する値。
- **srcOffset**　ソース配列の開始位置。
- **count**　コピーされるバイト数。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

srcOffset (0 から始まります) から、配列 *val* の *srcOffset*+*count*-1 までの位置のバイトが、指定されたカラムの値に追加されます。

挿入時には、ULTable.InsertBegin は新しい値をカラムのデフォルト値に初期化します。ローのデータは、ULTable.Insert が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

更新時の、カラムに対する最初の追加では、現在のカラム値がクリアされてから新しい値が追加されます。

次のいずれかに該当する場合、コード ULSQLCode.SQLE_INVALID_PARAMETER とともに ULException がスローされ、追加先は修正されません。

- *val* が NULL である。
- *srcOffset* が負である。
- *count* が負である。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- *srcOffset*+*count* が *val* の長さより長い。

その他のエラーの場合は、それに応じたエラーコードとともに ULException がスローされます。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.InsertBegin メソッド [Ultra Light.NET]431 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULException クラス [Ultra Light.NET]274 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Byte

AppendChars メソッド

指定された System.Chars 配列の指定されたサブセットを、指定された ULDbType.LongVarchar カラムの新しい値に追加します。

Visual Basic 構文

```
Public Sub AppendChars(
     ByVal colID As Integer,
    ByVal val As Char(),
    ByVal srcOffset As Integer,
    ByVal count As Integer
)
```

C# 構文

```
public void AppendChars(int colID, char[] val, int srcOffset, int count)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの現在の新しい値に追加する値。
- **srcOffset**　ソース配列の開始位置。
- **count**　コピーされるバイト数。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

srcOffset (0 から始まります) から、配列 *val* の *srcOffset*+*count*-1 までの位置の文字が、指定されたカラムの値に追加されます。挿入時には、ULTable.InsertBegin は新しい値をカラムのデフォルト

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

値に初期化します。ローのデータは、ULTable.Insert が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

更新時の、カラムに対する最初の追加では、現在のカラム値がクリアされてから新しい値が追加されます。

次のいずれかに該当する場合、コード ULSQLCode.SQLE_INVALID_PARAMETER とともに ULException がスローされ、追加先は修正されません。

●*val* が NULL である。

●*srcOffset* が負である。

●*count* が負である。

●*srcOffset+count* が *value* の長さより長い。

その他のエラーの場合は、それに応じたエラーコードとともに ULException がスローされます。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.InsertBegin メソッド [Ultra Light.NET]431 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULException クラス [Ultra Light.NET]274 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Char

Delete メソッド

現在の行を削除します。

Visual Basic 構文

```
Public Sub Delete()
```

C# 構文

```
public void Delete()
```

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULConnection.StartSynchronizationDelete メソッド [Ultra Light.NET]150 ページ
- ULConnection.StopSynchronizationDelete メソッド [Ultra Light.NET]150 ページ

SetBoolean メソッド

指定されたカラムの値を、System.Boolean を使用して設定します。

Visual Basic 構文

```
Public Sub SetBoolean(ByVal colID As Integer, ByVal val As Boolean)
```

C# 構文

```
public void SetBoolean(int colID, bool val)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの新しい値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Boolean

SetByte メソッド

指定されたカラムの値を、System.Byte (符号なし 8 ビット整数) を使用して設定します。

Visual Basic 構文

```
Public Sub SetByte(ByVal colID As Integer, ByVal val As Byte)
```

C# 構文

```
public void SetByte(int colID, byte val)
```


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの新しい値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Byte

SetBytes メソッド

指定されたカラムの値を、System.Bytes を使用して設定します。

Visual Basic 構文

```
Public Sub SetBytes(ByVal colID As Integer, ByVal val As Byte())
```

C# 構文

```
public void SetBytes(int colID, byte[] val)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの新しい値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

ULDbType.Binary 型または ULDbType.LongBinary 型のカラム、あるいは val の長さが 16 の場合の ULDbType.UniqueIdentifier 型のカラムにのみ適しています。ローのデータは、ULTable.Insert

または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Byte

SetDateTime メソッド

指定されたカラムの値を、System.DateTime を使用して設定します。

Visual Basic 構文

```
Public Sub SetDateTime(ByVal colID As Integer, ByVal val As Date)
```

C# 構文

```
public void SetDateTime(int colID, DateTime val)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの新しい値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

設定された値の精度はミリ秒です。ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.DateTime

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

SetDBNull メソッド

カラムを NULL に設定します。

Visual Basic 構文

```
Public Sub SetDBNull(ByVal colID As Integer)
```

C# 構文

```
public void SetDBNull(int colID)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

データは、ULTable.Insert または Update を実行するまでは、実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULTableSchema.IsColumnNullable メソッド [Ultra Light.NET]447 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ

SetDecimal メソッド

指定されたカラムの値を、System.Decimal を使用して設定します。

Visual Basic 構文

```
Public Sub SetDecimal(ByVal colID As Integer, ByVal val As Decimal)
```

C# 構文

```
public void SetDecimal(int colID, decimal val)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの新しい値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Decimal

SetDouble メソッド

指定されたカラムの値を、System.Double を使用して設定します。

Visual Basic 構文

```
Public Sub SetDouble(ByVal colID As Integer, ByVal val As Double)
```

C# 構文

```
public void SetDouble(int colID, double val)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの新しい値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Double

SetFloat メソッド

指定されたカラムの値を、System.Single を使用して設定します。

Visual Basic 構文

```
Public Sub SetFloat(ByVal colID As Integer, ByVal val As Single)
```

C# 構文

```
public void SetFloat(int colID, float val)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの新しい値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Single

SetGuid メソッド

指定されたカラムの値を、System.Guid を使用して設定します。

Visual Basic 構文

```
Public Sub SetGuid(ByVal colID As Integer, ByVal val As Guid)
```

C# 構文

```
public void SetGuid(int colID, Guid val)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの新しい値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。ULDbType.UniqueIdentifier 型のカラム、または長さが 16 の ULDbType.Binary 型のカラムの場合にのみ有効です。

参照

- ULConnection.GetNewUUID メソッド [Ultra Light.NET]139 ページ
- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULCursorSchema.GetColumnSize メソッド [Ultra Light.NET]206 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Guid

SetInt16 メソッド

指定されたカラムの値を、System.Int16 を使用して設定します。

Visual Basic 構文

```
Public Sub SetInt16(ByVal colID As Integer, ByVal val As Short)
```

C# 構文

```
public void SetInt16(int colID, short val)
```


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの新しい値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Int16

SetInt32 メソッド

指定されたカラムの値を、System.Int32 を使用して設定します。

Visual Basic 構文

```
Public Sub SetInt32(ByVal colID As Integer, ByVal val As Integer)
```

C# 構文

```
public void SetInt32(int colID, int val)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの新しい値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Int32

SetInt64 メソッド

指定されたカラムの値を、Int64 を使用して設定します。

Visual Basic 構文

```
Public Sub SetInt64(ByVal colID As Integer, ByVal val As Long)
```

C# 構文

```
public void SetInt64(int colID, long val)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの新しい値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.Int64

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

SetString メソッド

指定されたカラムの値を、System.String を使用して設定します。

Visual Basic 構文

```
Public Sub SetString(ByVal colID As Integer, ByVal val As String)
```

C# 構文

```
public void SetString(int colID, string val)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの新しい値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ

SetTimeSpan メソッド

指定されたカラムの値を、System.TimeSpan を使用して設定します。

Visual Basic 構文

```
Public Sub SetTimeSpan(ByVal colID As Integer, ByVal val As TimeSpan)
```

C# 構文

```
public void SetTimeSpan(int colID, TimeSpan val)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

- **val** カラムの新しい値。

例外

- **ULException クラス** SQL エラーが発生しました。

備考

設定された値は、ミリ秒の精度であり、0 ～ 24 時までの間の負でない値に正規化されます。ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.TimeSpan

SetToDefault メソッド

指定されたカラムの値を、そのデフォルト値に設定します。

Visual Basic 構文

```
Public Sub SetToDefault(ByVal colID As Integer)
```

C# 構文

```
public void SetToDefault(int colID)
```

パラメーター

- **colID** カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

例外

- **ULException クラス** SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTableSchema.GetColumnDefaultValue メソッド [Ultra Light.NET]439 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ

SetUInt16 メソッド

指定されたカラムの値を、System.UInt16 を使用して設定します。

Visual Basic 構文

```
Public Sub SetUInt16(ByVal colID As Integer, ByVal val As UShort)
```

C# 構文

```
public void SetUInt16(int colID, ushort val)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの新しい値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.UInt16

SetUInt32 メソッド

指定されたカラムの値を、System.UInt32 を使用して設定します。

Visual Basic 構文

```
Public Sub SetUInt32(ByVal colID As Integer, ByVal val As UInteger)
```

C# 構文

```
public void SetUInt32(int colID, uint val)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの新しい値。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.UInt32

SetUInt64 メソッド

指定されたカラムの値を、System.UInt64 を使用して設定します。

Visual Basic 構文

```
Public Sub SetUInt64(ByVal colID As Integer, ByVal val As ULong)
```

C# 構文

```
public void SetUInt64(int colID, ulong val)
```

パラメーター

- **colID**　カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。
- **val**　カラムの新しい値。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

参照

- ULDataReader.GetOrdinal メソッド [Ultra Light.NET]255 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULTable.Insert メソッド [Ultra Light.NET]430 ページ
- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ
- ULDataReader.FieldCount プロパティ [Ultra Light.NET]269 ページ
- System.UInt64

Update メソッド

現在のカラム値 (set メソッドを使用して指定されます) で新しいローを更新します。

Visual Basic 構文

```
Public Sub Update()
```

C# 構文

```
public void Update()
```

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULResultSet.UpdateBegin メソッド [Ultra Light.NET]375 ページ

UpdateBegin メソッド

現在のローを更新する準備を行います。

Visual Basic 構文

```
Public Sub UpdateBegin()
```

C# 構文

```
public void UpdateBegin()
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

カラム値は、適切な setType メソッドまたは AppendType メソッドを呼び出すことによって修正します。カラムに対する最初の追加では、現在のカラム値がクリアされてから新しい値が追加されます。

ローのデータは、Update() を呼び出すまで実際には変更されません。また、その変更がコミットされないかぎり、永続化されません。

テーブルを開くのに使用されるインデックス内のカラムを修正すると、アクティブな検索処理に予期しない影響を及ぼします。テーブルのプライマリキー内のカラムは更新できません。

参照

- ULResultSet.Update メソッド [Ultra Light.NET]375 ページ

ULResultSetSchema クラス

UL 拡張： Ultra Light の結果セットのスキーマを示します。

Visual Basic 構文

```
Public NotInheritable Class ULResultSetSchema Inherits ULCursorSchema
```

C# 構文

```
public sealed class ULResultSetSchema : ULCursorSchema
```

基本クラス

- ULCursorSchema クラス [Ultra Light.NET]201 ページ

メンバー

継承されたメンバーを含む ULResultSetSchema クラスのすべてのメンバー。

名前	説明
GetColumnID メソッド	指定されたカラムのカラム ID を返します。
GetColumnName メソッド	指定されたカラム ID で識別されたカラムの名前を返します。
GetColumnPrecision メソッド	カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの精度を返します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
GetColumnScale メソッド	カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの位取りを返します。
GetColumnSize メソッド	カラムがサイズ指定されたカラム (BINARY SQL 型または CHAR SQL 型) の場合は、指定されたカラム ID によって識別されたカラムのサイズを返します。
GetColumnSQLName メソッド	指定されたカラム ID で識別されたカラムの名前を返します。
GetColumnULDbType メソッド	指定されたカラム ID によって識別されたカラムの Ultra Light.NET データ型を返します。
GetSchemaTable メソッド	ULDataReader オブジェクトのカラムのスキーマが記述された System.Data.DataTable を返します。
ColumnCount プロパティ	このカーソル内のカラム数を返します。
IsOpen プロパティ	カーソルのスキーマが現在開いているかどうかを確認します。
Name プロパティ	カーソルの名前を返します。

備考

このクラスにはコンストラクターがありません。ULResultSetSchema オブジェクトは、その ULDataReader.Schema として結果セットにアタッチされます。

結果セットのスキーマが有効なのは、データリーダーが開かれている間だけです。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- ULDataReader クラス [Ultra Light.NET]236 ページ
- ULResultSetSchema クラス [Ultra Light.NET]376 ページ
- ULDataReader.Schema プロパティ [Ultra Light.NET]271 ページ
- ULCursorSchema クラス [Ultra Light.NET]201 ページ

Name プロパティ

カーソルの名前を返します。

Visual Basic 構文

```
Public ReadOnly Overrides Property Name As String
```

C# 構文

```
public override string Name {get;}
```

備考

ULResultSetSchema を生成した SQL 文。

ULRowsCopiedEventArgs クラス

ULRowsCopiedEventHandler オブジェクトに渡される引数のセットを表します。

Visual Basic 構文

```
Public NotInheritable Class ULRowsCopiedEventArgs
```

C# 構文

```
public sealed class ULRowsCopiedEventArgs
```

メンバー

継承されたメンバーを含む ULRowsCopiedEventArgs クラスのすべてのメンバー。

名前	説明
ULRowsCopiedEventArgs コンストラクター	ULRowsCopiedEventArgs オブジェクトの新しいインスタンスを作成します。
Abort プロパティ	バルクコピーオペレーションをアボートするかどうかを示す値を取得または設定します。
RowsCopied プロパティ	現在のバルクコピーオペレーションでコピーされるローの数を返します。

備考

ULRowsCopiedEventArgs クラスは、.NET Compact Framework 2.0 では使用できません。

参照

- ULRowsCopiedEventHandler デリゲート [Ultra Light.NET]455 ページ

ULRowsCopiedEventArgs コンストラクター

ULRowsCopiedEventArgs オブジェクトの新しいインスタンスを作成します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public Sub New(ByVal rowsCopied As Long)
```

C# 構文

```
public ULRowsCopiedEventArgs(long rowsCopied)
```

パラメーター

- **rowsCopied** 現在のバルクコピーオペレーションでコピーされるローの数を示す 64 ビット整数値。

備考

ULRowsCopiedEventArgs クラスは、.NET Compact Framework 2.0 では使用できません。

Abort プロパティ

バルクコピーオペレーションをアボートするかどうかを示す値を取得または設定します。

Visual Basic 構文

```
Public Property Abort As Boolean
```

C# 構文

```
public bool Abort {get;set;}
```

備考

ULRowsCopiedEventArgs クラスは、.NET Compact Framework 2.0 では使用できません。

RowsCopied プロパティ

現在のバルクコピーオペレーションでコピーされるローの数を返します。

Visual Basic 構文

```
Public ReadOnly Property RowsCopied As Long
```

C# 構文

```
public long RowsCopied {get;}
```

備考

コピーされたロー数を表す long integer。

ULRowsCopiedEventArgs クラスは、.NET Compact Framework 2.0 では使用できません。

ULRowUpdatedEventArgs クラス

ULDataAdapter.RowUpdated イベントのデータを提供します。

Visual Basic 構文

```
Public NotInheritable Class ULRowUpdatedEventArgs
    Inherits System.Data.Common.RowUpdatedEventArgs
```

C# 構文

```
public sealed class ULRowUpdatedEventArgs :
    System.Data.Common.RowUpdatedEventArgs
```

基本クラス

- System.Data.Common.RowUpdatedEventArgs

メンバー

継承されたメンバーを含む ULRowUpdatedEventArgs クラスのすべてのメンバー。

名前	説明
ULRowUpdatedEventArgs コンストラクター	ULRowUpdatedEventArgs クラスの新しいインスタンスを初期化します。
CopyToRows method (System.Data.Common.RowUpdatedEventArgs から継承)	変更されたローへの参照を指定された配列にコピーします。
Command プロパティ	DbDataAdapter.Update メソッドが呼び出されると実行される ULCommand オブジェクトを返します。
Errors property (System.Data.Common.RowUpdatedEventArgs から継承)	System.Data.Common.RowUpdatedEventArgs.Command が実行されたときに .NET Framework データプロバイダーによって生成されたエラーがあれば取得します。
RecordsAffected プロパティ	SQL 文の実行によって変更、挿入、または削除されたローの数を返します。
Row property (System.Data.Common.RowUpdatedEventArgs から継承)	System.Data.Common.DbDataAdapter.Update(System.Data.DataSet) 経由で送信された System.Data.DataRow を取得します。
RowCount property (System.Data.Common.RowUpdatedEventArgs から継承)	更新済みレコードのバッチで処理されたローの数を取得します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
StatementType property (System.Data.Common.RowUpdatedEventArgs から継承)	実行された SQL 文のタイプを取得します。
Status property (System.Data.Common.RowUpdatedEventArgs から継承)	System.Data.Common.RowUpdatedEventArgs.Command プロパティの System.Data.UpdateStatus を取得します。
TableMapping property (System.Data.Common.RowUpdatedEventArgs から継承)	System.Data.Common.DbDataAdapter.Update(System.Data.DataSet) 経由で送信された System.Data.Common.DataTableMapping を取得します。

参照

- ULDataAdapter.RowUpdated event [Ultra Light.NET]219 ページ
- System.Data.Common.RowUpdatedEventArgs

ULRowUpdatedEventArgs コンストラクター

ULRowUpdatedEventArgs クラスの新しいインスタンスを初期化します。

Visual Basic 構文

```
Public Sub New(
     ByVal row As DataRow,
    ByVal command As IDbCommand,
    ByVal statementType As StatementType,
    ByVal tableMapping As DataTableMapping
)
```

C# 構文

```
public ULRowUpdatedEventArgs(
     DataRow row,
    IDbCommand command,
    StatementType statementType,
    DataTableMapping tableMapping
)
```

パラメーター

- **row** DbDataAdapter.Update 呼び出しにより送信される System.Data.DataRow。
- **command** DbDataAdapter.Update メソッドが呼び出されると実行される System.Data.IDbCommand。
- **statementType** 実行されたクエリのタイプを指定する System.Data.StatementType 値の 1 つ。

- **tableMapping**　DbDataAdapter.Update 呼び出しにより送信される System.Data.Common.DataTableMapping。

参照

- System.Data.DataRow
- System.Data.IDbCommand
- System.Data.StatementType
- System.Data.Common.DataTableMapping

Command プロパティ

DbDataAdapter.Update メソッドが呼び出されると実行される ULCommand オブジェクトを返します。

Visual Basic 構文

```
Public ReadOnly Shadows Property Command As ULCommand
```

C# 構文

```
public new ULCommand Command {get;}
```

備考

更新で実行された ULCommand オブジェクト。

これは、System.Data.Common.RowUpdatedEventArgs.Command が厳密に型指定されたものです。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- System.Data.Common.RowUpdatedEventArgs.Command

RecordsAffected プロパティ

SQL 文の実行によって変更、挿入、または削除されたローの数を返します。

Visual Basic 構文

```
Public ReadOnly Shadows Property RecordsAffected As Integer
```

C# 構文

```
public new int RecordsAffected {get;}
```

備考

SELECT 文の場合、この値は -1 です。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

変更、挿入、または削除されたローの数。文が失敗したときにローが影響されなかった場合は 0、SELECT 文の場合は -1。

ULRowUpdatingEventArgs クラス

ULDataAdapter.RowUpdating イベントのデータを提供します。

Visual Basic 構文

```
Public NotInheritable Class ULRowUpdatingEventArgs
    Inherits System.Data.Common.RowUpdatingEventArgs
```

C# 構文

```
public sealed class ULRowUpdatingEventArgs :
    System.Data.Common.RowUpdatingEventArgs
```

基本クラス

- System.Data.Common.RowUpdatingEventArgs

メンバー

継承されたメンバーを含む ULRowUpdatingEventArgs クラスのすべてのメンバー。

名前	説明
ULRowUpdatingEventArgs コンストラクター	ULRowUpdatingEventArgs クラスの新しいインスタンスを初期化します。
BaseCommand property (System.Data.Common.RowUpdatingEventArgs から継承)	このクラスのインスタンス用に System.Data.IDbCommand オブジェクトを取得または設定します。
Command プロパティ	DbDataAdapter.Update メソッドの実行時に実行する ULCommand オブジェクトを指定します。
Errors property (System.Data.Common.RowUpdatingEventArgs から継承)	System.Data.Common.RowUpdatedEventArgs.Command が実行するときに、.NET Framework データプロバイダーによって生成されるエラーがあれば取得します。
Row property (System.Data.Common.RowUpdatingEventArgs から継承)	挿入、更新、または削除操作の一部としてサーバーに送信される System.Data.DataRow を取得します。

名前	説明
StatementType property (System.Data.Common.RowUpdatingEventArgs から継承)	実行する SQL 文のタイプを取得します。
Status property (System.Data.Common.RowUpdatingEventArgs から継承)	System.Data.Common.RowUpdatedEventArgs.Command プロパティの System.Data.UpdateStatus を取得または設定します。
TableMapping property (System.Data.Common.RowUpdatingEventArgs から継承)	System.Data.Common.DbDataAdapter.Update(System.Data.DataSet) 経由で送信されるように System.Data.Common.DataTableMapping を取得します。

参照

- ULDataAdapter.RowUpdating event [Ultra Light.NET]220 ページ
- System.Data.Common.RowUpdatingEventArgs

ULRowUpdatingEventArgs コンストラクター

ULRowUpdatingEventArgs クラスの新しいインスタンスを初期化します。

Visual Basic 構文

```
Public Sub New(
     ByVal row As DataRow,
    ByVal command As IDbCommand,
    ByVal statementType As StatementType,
    ByVal tableMapping As DataTableMapping
)
```

C# 構文

```
public ULRowUpdatingEventArgs(
     DataRow row,
    IDbCommand command,
    StatementType statementType,
    DataTableMapping tableMapping
)
```

パラメーター

- **row**　更新する System.Data.DataRow。
- **command**　更新時に実行される System.Data.IDbCommand。
- **statementType**　実行されたクエリのタイプを指定する System.Data.StatementType 値の 1 つ。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

- **tableMapping**　DbDataAdapter.Update 呼び出しにより送信される System.Data.Common.DataTableMapping 値。

参照

- System.Data.DataRow
- System.Data.IDbCommand
- System.Data.StatementType
- System.Data.Common.DataTableMapping

Command プロパティ

DbDataAdapter.Update メソッドの実行時に実行する ULCommand オブジェクトを指定します。

Visual Basic 構文

```
Public Shadows Property Command As ULCommand
```

C# 構文

```
public new ULCommand Command {get;set;}
```

備考

更新時に実行される ULCommand オブジェクト。

これは、System.Data.Common.RowUpdatingEventArgs.Command 値が厳密に型指定されたものです。

参照

- ULCommand クラス [Ultra Light.NET]66 ページ
- System.Data.Common.RowUpdatingEventArgs.Command

ULServerSyncListener インターフェイス

UL 拡張： サーバーの同期メッセージを受信するリスナーインターフェイスです。

Visual Basic 構文

```
Public Interface ULServerSyncListener
```

C# 構文

```
public interface ULServerSyncListener
```

メンバー

継承されたメンバーを含む ULServerSyncListener インターフェイスのすべてのメンバー。

名前	説明
ServerSyncInvoked メソッド	サーバー起動同期用の Mobile Link Listener がアプリケーションを呼び出して同期を実行するときに起動されます。

ServerSyncInvoked メソッド

サーバー起動同期用の Mobile Link Listener がアプリケーションを呼び出して同期を実行するときに起動されます。

Visual Basic 構文

```
Public Sub ServerSyncInvoked(ByVal messageName As String)
```

C# 構文

```
public void ServerSyncInvoked(string messageName)
```

パラメーター

- **messageName** アプリケーションに送信されるメッセージの名前。

備考

このメソッドは、別のスレッドによって呼び出されます。マルチスレッドの問題を回避するには、このメソッドがユーザーインターフェイスにイベントを通知する必要があります。マルチスレッドを使用する場合は、スレッドごとに別々の接続を使用し、lock キーワードを使用して、アプリケーションの共有オブジェクトにアクセスすることをおすすめします。

例

次の Visual Basic コードフラグメントは、サーバー同期要求の受信方法と UI スレッドでの同期の実行方法を示してします。

```
Imports 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
        ...
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

```
    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
```

次の C# コードフラグメントは、サーバー同期要求の受信方法と UI スレッドでの同期の実行方法を示してします。

```
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 クラス

UL 拡張： SQL パススルースクリプトの進行状況のモニタリングデータを返します。

Visual Basic 構文

```
Public Class ULSqlProgressData
```

C# 構文

```
public class ULSqlProgressData
```

メンバー

継承されたメンバーを含む ULSqlProgressData クラスのすべてのメンバー。

名前	説明
CurrentScript プロパティ	これまでに実行されたスクリプトのインデックス。
ScriptCount プロパティ	実行中のスクリプトの数を返します。
State プロパティ	現在の進行状況のステータスを返します。

CurrentScript プロパティ

これまでに実行されたスクリプトのインデックス。

Visual Basic 構文

```
Public ReadOnly Property CurrentScript As Long
```

C# 構文

```
public long CurrentScript {get;}
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

実行中のスクリプトの現在のインデックス。

ScriptCount プロパティ

実行中のスクリプトの数を返します。

Visual Basic 構文

```
Public ReadOnly Property ScriptCount As Long
```

C# 構文

```
public long ScriptCount {get;}
```

備考

実行中のスクリプトの数。

State プロパティ

現在の進行状況のステータスを返します。

Visual Basic 構文

```
Public ReadOnly Property State As ULSqlProgressState
```

C# 構文

```
public ULSqlProgressState State {get;}
```

備考

現在の SQL パススルーのコールバックステータスを指定する ULSqlProgressState 値の 1 つ。

参照

- ULSyncProgressState 列挙体 [Ultra Light.NET]467 ページ

ULSyncParms クラス

UL 拡張： Ultra Light データベースの同期方法を定義する同期パラメーターを表します。

Visual Basic 構文

```
Public NotInheritable Class ULSyncParms
```

C# 構文

```
public sealed class ULSyncParms
```

メンバー

継承されたメンバーを含む ULSyncParms クラスのすべてのメンバー。

名前	説明
CopyFrom メソッド	指定された ULSyncParms オブジェクトのプロパティを、この ULSyncParms オブジェクトにコピーします。
ToString メソッド	このインスタンスの文字列表記を返します。
AdditionalParms プロパティ	「名前=値」のペアをセミコロンで区切ったリストで、追加の同期パラメーターを指定します。
AuthenticationParms プロパティ	カスタムユーザー認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメーターを指定します。
DownloadOnly プロパティ	同期時のアップロードを無効にするか、有効にするかを指定します。
KeepPartialDownload プロパティ	同期時の部分的なダウンロードを無効にするか、有効にするかを指定します。
NewPassword プロパティ	UserName で指定されたユーザーの新しい Mobile Link パスワードを指定します。
Password プロパティ	UserName で指定されたユーザーの Mobile Link パスワードです。
PingOnly プロパティ	実際に同期を行う代わりに、クライアントが Mobile Link サーバーに ping のみを行うかどうかを指定します。
Publications プロパティ	同期させるパブリケーションを指定します。
ResumePartialDownload プロパティ	前の部分的なダウンロードを再開するか、破棄するかを指定します。
SendColumnNames プロパティ	同期中に、クライアントが Mobile Link サーバーにカラム名を送信するかどうかを指定します。
SendDownloadAck プロパティ	同期中に、クライアントが Mobile Link サーバーにダウンロード確認を送信するかどうかを指定します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
Stream プロパティ	同期に使用する Mobile Link 同期ストリームを指定します。
StreamParms プロパティ	同期ストリームの設定パラメーターを指定します。
UploadOnly プロパティ	同期時のダウンロードを無効にするか、有効にするかを指定します。
UserName プロパティ	Mobile Link サーバーが Mobile Link クライアントをユニークに識別するユーザー名です。
Version プロパティ	使用する同期スクリプトを指定します。

備考

このクラスにはコンストラクターがありません。各接続には、ULConnection.SyncParms としてアタッチされた、固有の ULSyncParms インスタンスがあります。

同期コマンド (ULSyncParms.DownloadOnly、ULSyncParms.PingOnly、ULSyncParms.ResumePartialDownload、ULSyncParms.UploadOnly) は、一度に 1 つしか指定できません。これらの複数のパラメーターが true に設定されていると、ULSQLCode.SQLE_SYNC_INFO_INVALID SQLException が ULConnection.Synchronize() によってスローされます。

その他の ULSQLCode.SQLE_SYNC_INFO_INVALID エラーの原因には、ULSyncParms.Stream 値または ULSyncParms.Version 値を指定していないことが含まれます。

参照

- ULConnection クラス [Ultra Light.NET]116 ページ
- ULConnection.SyncParms プロパティ [Ultra Light.NET]160 ページ
- ULConnection.Synchronize メソッド [Ultra Light.NET]151 ページ
- ULSyncParms.DownloadOnly プロパティ [Ultra Light.NET]393 ページ
- ULSyncParms.PingOnly プロパティ [Ultra Light.NET]396 ページ
- ULSyncParms.ResumePartialDownload プロパティ [Ultra Light.NET]397 ページ
- ULSyncParms.UploadOnly プロパティ [Ultra Light.NET]400 ページ
- ULConnection.Synchronize メソッド [Ultra Light.NET]151 ページ
- ULSyncParms.Stream プロパティ [Ultra Light.NET]399 ページ
- ULSyncParms.Version プロパティ [Ultra Light.NET]401 ページ

CopyFrom メソッド

指定された ULSyncParms オブジェクトのプロパティを、この ULSyncParms オブジェクトにコピーします。

Visual Basic 構文

```
Public Sub CopyFrom(ByVal src As ULSyncParms)
```

C# 構文

```
public void CopyFrom(ULSyncParms src)
```

パラメーター

- **src** コピー元のオブジェクト。

参照

- ULSyncParms クラス [Ultra Light.NET]389 ページ

ToString メソッド

このインスタンスの文字列表記を返します。

Visual Basic 構文

```
Public Overrides Function ToString() As String
```

C# 構文

```
public override string ToString()
```

戻り値

「キーワード=値」の組み合わせがセミコロンで区切られたリスト形式の、このインスタンスの文字列表現。

AdditionalParms プロパティ

「名前=値」のペアをセミコロンで区切ったリストで、追加の同期パラメーターを指定します。

Visual Basic 構文

```
Public Property AdditionalParms As String
```

C# 構文

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

備考

「名前=値」のペアをセミコロンで区切ったリスト形式の文字列。

このプロパティを使用すると、他の定義済みパラメーターでは簡単に指定できないいくつかの追加の同期パラメーターを指定できます。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- 「Additional Parameters 同期パラメーター」『Ultra Light データベース管理とリファレンス』

例

次の例は、ULSyncParms オブジェクトに AllowDownloadDupRows、CheckpointStore、DisableConcurrency、TableOrder パラメーターを設定する方法を示します。

```
private ULSyncParms info;
// ...
info.AdditionalParms =
   "AllowDownloadDupRows=1;
   CheckpointStore=1;
   DisableConcurrency=1;
   TableOrder=Customer,Sales"
```

AuthenticationParms プロパティ

カスタムユーザー認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメーターを指定します。

Visual Basic 構文

```
Public Property AuthenticationParms As String()
```

C# 構文

```
public string[] AuthenticationParms {get;set;}
```

備考

それぞれに認証パラメーターが格納された、文字列の配列 (配列のエントリが NULL であると、同期エラーになります)。デフォルト値は NULL 参照 (Visual Basic の Nothing) で、認証パラメーターは指定されません。

最初の 255 文字列のみが使用されます。それぞれの文字列は認証パラメーターの Mobile Link サーバーの制限 (現在は 4000 UTF8 バイト) よりも長くはなりません。

DownloadOnly プロパティ

同期時のアップロードを無効にするか、有効にするかを指定します。

Visual Basic 構文

```
Public Property DownloadOnly As Boolean
```

C# 構文

```
public bool DownloadOnly {get;set;}
```

備考

同期時のアップロードを無効にする場合は true、有効にする場合は false。デフォルトは false です。

同期コマンド (ULSyncParms.DownloadOnly、ULSyncParms.PingOnly、ULSyncParms.ResumePartialDownload、ULSyncParms.UploadOnly) は、一度に 1 つしか指定できません。これらの複数のパラメーターが true に設定されていると、ULSQLCode.SQLE_SYNC_INFO_INVALID SQLException が ULConnection.Synchronize() によってスローされます。

参照

- ULSyncParms.UploadOnly プロパティ [Ultra Light.NET]400 ページ
- ULSyncParms.DownloadOnly プロパティ [Ultra Light.NET]393 ページ
- ULSyncParms.PingOnly プロパティ [Ultra Light.NET]396 ページ
- ULSyncParms.ResumePartialDownload プロパティ [Ultra Light.NET]397 ページ
- ULSyncParms.UploadOnly プロパティ [Ultra Light.NET]400 ページ
- ULConnection.Synchronize メソッド [Ultra Light.NET]151 ページ

KeepPartialDownload プロパティ

同期時の部分的なダウンロードを無効にするか、有効にするかを指定します。

Visual Basic 構文

```
Public Property KeepPartialDownload As Boolean
```

C# 構文

```
public bool KeepPartialDownload {get;set;}
```

備考

同期時の部分的なダウンロードを有効にする場合は true、無効にする場合は false。デフォルトは false です。

Ultra Light.NET では、通信エラーや、ユーザーによる ULSyncProgressListener からのアボートが原因で失敗したダウンロードを再起動できます。Ultra Light.NET は、ダウンロードを受信しながら処理します。ダウンロードが中断した場合は、部分的なダウンロードトランザクションがデータベース内に残るため、次の同期中に再開できます。

Ultra Light.NET で部分的なダウンロードを保存する必要があることを示すには、connection.SyncParms.KeepPartialDownload=true と指定します。指定しないと、エラーが発生した場合にダウンロードがロールバックされます。

部分的なダウンロードが保持された場合、connection.Synchronize() の終了時に、出力フィールド connection.SyncResult.ULSyncResult.PartialDownloadRetained が true に設定されます。

PartialDownloadRetained が設定されている場合は、ダウンロードを再開できます。再開するには、true に設定された connection.SyncParms.ULSyncParms.ResumePartialDownload を使用して

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

connection.Synchronize() を呼び出します。別の通信エラーの発生に備えて、KeepPartialDownload も true に設定しておくことをおすすめします。ダウンロードが省略された場合は、アップロードは行われません。

再開したダウンロードで受信するダウンロードは、最初にダウンロードを開始したときと同じものです。最新のデータが必要な場合は、再開された特別なダウンロードが完了した直後に、もう一度ダウンロードを行うことができます。

ダウンロードを再開する場合、ULSyncParms フィールドの多くは関係ありません。たとえば、Publications フィールドは使用されません。受信するパブリケーションは、最初のダウンロード時に要求したものです。設定する必要があるフィールドは、ResumePartialDownload と UserName だけです。KeepPartialDownload フィールドを必要に応じて設定すると、正常に機能します。

部分的なダウンロードが存在するが、このダウンロードが必要ではなくなった場合は、ULConnection.RollbackPartialDownload() を呼び出して、失敗したダウンロードトランザクションをロールバックできます。また、同期をもう一度実行したときに ResumePartialDownload を指定しなかった場合は、部分的なダウンロードがロールバックされてから、次の同期が開始されます。

参照

- ULSyncResult.PartialDownloadRetained プロパティ [Ultra Light.NET]412 ページ
- ULSyncParms.ResumePartialDownload プロパティ [Ultra Light.NET]397 ページ
- ULConnection.RollbackPartialDownload メソッド [Ultra Light.NET]148 ページ
- ULSyncParms.ResumePartialDownload プロパティ [Ultra Light.NET]397 ページ
- ULSyncParms.UserName プロパティ [Ultra Light.NET]401 ページ
- ULConnection.RollbackPartialDownload メソッド [Ultra Light.NET]148 ページ
- 「ダウンロードの失敗の再開」『Mobile Link サーバー管理』

NewPassword プロパティ

UserName で指定されたユーザーの新しい Mobile Link パスワードを指定します。

Visual Basic 構文

```
Public Property NewPassword As String
```

C# 構文

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

備考

新しい Mobile Link パスワードを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) で、パスワードは変更されません。

新しいパスワードが有効になるのは、次の同期の後です。

参照

- ULSyncParms.UserName プロパティ [Ultra Light.NET]401 ページ

Password プロパティ

UserName で指定されたユーザーの Mobile Link パスワードです。

Visual Basic 構文

```
Public Property Password As String
```

C# 構文

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

備考

Mobile Link パスワードを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) で、パスワードは指定されません。

Mobile Link ユーザー名とパスワードは他のデータベースユーザー ID やパスワードとは別のもので、アプリケーションを Mobile Link サーバーに対して識別し、認証するために使用されます。

参照

- ULSyncParms.NewPassword プロパティ [Ultra Light.NET]395 ページ
- ULSyncParms.UserName プロパティ [Ultra Light.NET]401 ページ

PingOnly プロパティ

実際に同期を行う代わりに、クライアントが Mobile Link サーバーに ping のみを行うかどうかを指定します。

Visual Basic 構文

```
Public Property PingOnly As Boolean
```

C# 構文

```
public bool PingOnly {get;set;}
```

備考

クライアントが Mobile Link サーバーに ping のみを行うように指定する場合は true、クライアントが実際に同期を行うように指定する場合は false。デフォルトは false です。

同期コマンド (ULSyncParms.DownloadOnly、ULSyncParms.PingOnly、ULSyncParms.ResumePartialDownload、ULSyncParms.UploadOnly) は、一度に 1 つしか指定できません。これらの複数のパラメーターが true に設定されていると、ULSQLCode.SQLE_SYNC_INFO_INVALID SQLException が ULConnection.Synchronize() によってスローされます。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULSyncParms.DownloadOnly プロパティ [Ultra Light.NET]393 ページ
- ULSyncParms.PingOnly プロパティ [Ultra Light.NET]396 ページ
- ULSyncParms.ResumePartialDownload プロパティ [Ultra Light.NET]397 ページ
- ULSyncParms.UploadOnly プロパティ [Ultra Light.NET]400 ページ
- ULConnection.Synchronize メソッド [Ultra Light.NET]151 ページ

Publications プロパティ

同期させるパブリケーションを指定します。

Visual Basic 構文

```
Public Property Publications As String
```

C# 構文

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

備考

カンマ (,) 区切りパブリケーション名、特別値 ULConnection.SYNC_ALL_PUBS、または特別値 ULConnection.SYNC_ALL_DB が含まれる文字列。デフォルトは ULConnection.SYNC_ALL_DB です。

参照

- ULConnection.SYNC_ALL_PUBS フィールド [Ultra Light.NET]164 ページ
- ULConnection.SYNC_ALL_DB フィールド [Ultra Light.NET]163 ページ

ResumePartialDownload プロパティ

前の部分的なダウンロードを再開するか、破棄するかを指定します。

Visual Basic 構文

```
Public Property ResumePartialDownload As Boolean
```

C# 構文

```
public bool ResumePartialDownload {get;set;}
```

備考

前の部分的なダウンロードを再開する場合は true、破棄する場合は false。デフォルトは false です。

同期コマンド (ULSyncParms.DownloadOnly、ULSyncParms.PingOnly、ULSyncParms.ResumePartialDownload、ULSyncParms.UploadOnly) は、一度に 1 つしか指定できません。これらの複数のパラメーターが true に設定されていると、

ULSQLCode.SQLE_SYNC_INFO_INVALID SQLException が ULConnection.Synchronize() によってスローされます。

参照

- ULSyncParms.KeepPartialDownload プロパティ [Ultra Light.NET]394 ページ
- ULSyncParms.DownloadOnly プロパティ [Ultra Light.NET]393 ページ
- ULSyncParms.PingOnly プロパティ [Ultra Light.NET]396 ページ
- ULSyncParms.ResumePartialDownload プロパティ [Ultra Light.NET]397 ページ
- ULSyncParms.UploadOnly プロパティ [Ultra Light.NET]400 ページ
- ULConnection.Synchronize メソッド [Ultra Light.NET]151 ページ
- ULSyncResult.PartialDownloadRetained プロパティ [Ultra Light.NET]412 ページ

SendColumnNames プロパティ

同期中に、クライアントが Mobile Link サーバーにカラム名を送信するかどうかを指定します。

Visual Basic 構文

```
Public Property SendColumnNames As Boolean
```

C# 構文

```
public bool SendColumnNames {get;set;}
```

備考

クライアントがカラム名を Mobile Link サーバーに送信する必要があると指定する場合は true、カラム名を送信しないように指定する場合は false。デフォルトは false です。

カラム名は、Mobile Link サーバーによってダイレクトローハンドリングで使用されます。Mobile Link サーバーがローハンドリング API を使用して、インデックスではなく、名前でカラムを参照する場合は、このオプションを設定してください。このオプションで送信されるカラム名は、このような場合にのみ使用されます。

SendDownloadAck プロパティ

同期中に、クライアントが Mobile Link サーバーにダウンロード確認を送信するかどうかを指定します。

Visual Basic 構文

```
Public Property SendDownloadAck As Boolean
```

C# 構文

```
public bool SendDownloadAck {get;set;}
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

ダウンロード確認は、ダウンロードがリモートで完全に適用されてコミットされた後 (正の確認)、またはダウンロードに失敗した後 (負の確認) に送信されます。

True に設定すると、クライアントが Mobile Link サーバーにダウンロード確認を送信することを指定します。ダウンロード確認を送信しないように指定する場合は False。デフォルトは False です。

クライアントがダウンロード確認を送信する場合、Mobile Link サーバーのデータベースワーカースレッドは、クライアントがダウンロードを適用してコミットするまで待機します。クライアントがダウンロード確認を送信しない場合、Mobile Link サーバーは、次の同期のため、より早く解放されます。

Stream プロパティ

同期に使用する Mobile Link 同期ストリームを指定します。

Visual Basic 構文

```
Public Property Stream As ULStreamType
```

C# 構文

```
public ULStreamType Stream {get;set;}
```

備考

使用する同期ストリームのタイプを指定する ULStreamType 値の 1 つ。デフォルトは ULStreamType.TCPIP です。

ほとんどの同期ストリームでは、Mobile Link サーバーのアドレスを識別したり、その他の動作を制御したりするパラメーターが必要です。これらのパラメーターは、ULSyncParms.StreamParms で指定します。

ストリームタイプが、プラットフォームに適さない無効な値に設定されていると、ストリームタイプは ULStreamType.TCPIP に設定されます。

参照

- ULStreamType 列挙体 [Ultra Light.NET]466 ページ
- ULSyncParms.StreamParms プロパティ [Ultra Light.NET]399 ページ

StreamParms プロパティ

同期ストリームの設定パラメーターを指定します。

Visual Basic 構文

```
Public Property StreamParms As String
```

C# 構文

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

備考

キーワードと値の組み合わせがセミコロンで区切られたリスト形式の、ストリームのパラメーターを指定する文字列。デフォルトは NULL 参照です。(Visual Basic の Nothing)

StreamParms は、同期ストリームに使用されるすべてのパラメーターが含まれている文字列です。パラメーターは、「名前=値」のペアをセミコロンで区切ったリスト ("param1=value1;param2=value2") で指定します。

参照

- ULSyncParms.Stream プロパティ [Ultra Light.NET]399 ページ
- ULStreamType 列挙体 [Ultra Light.NET]466 ページ
- 「Ultra Light 同期ストリームのネットワークプロトコルのオプション」『Ultra Light データベース管理とリファレンス』

UploadOnly プロパティ

同期時のダウンロードを無効にするか、有効にするかを指定します。

Visual Basic 構文

```
Public Property UploadOnly As Boolean
```

C# 構文

```
public bool UploadOnly {get;set;}
```

備考

ダウンロードを無効にする場合は true、有効にする場合は false。デフォルトは false です。

同期コマンド (ULSyncParms.DownloadOnly、ULSyncParms.PingOnly、ULSyncParms.ResumePartialDownload、ULSyncParms.UploadOnly) は、一度に 1 つしか指定できません。これらの複数のパラメーターが true に設定されていると、ULSQLCode.SQLE_SYNC_INFO_INVALID SQLException が ULConnection.Synchronize() によってスローされます。

参照

- ULSyncParms.DownloadOnly プロパティ [Ultra Light.NET]393 ページ
- ULSyncParms.PingOnly プロパティ [Ultra Light.NET]396 ページ
- ULSyncParms.ResumePartialDownload プロパティ [Ultra Light.NET]397 ページ
- ULSyncParms.UploadOnly プロパティ [Ultra Light.NET]400 ページ
- ULConnection.Synchronize メソッド [Ultra Light.NET]151 ページ

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

UserName プロパティ

Mobile Link サーバーが Mobile Link クライアントをユニークに識別するユーザー名です。

Visual Basic 構文

```
Public Property UserName As String
```

C# 構文

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

備考

ユーザー名を指定する文字列。このパラメーターにはデフォルト値がないので、明示的に設定してください。

Mobile Link サーバーでは、この値を使用して、ダウンロードする内容の決定、同期ステータスの記録、同期中の割り込みからの復帰を行います。このユーザー名とパスワードは他のデータベースユーザー ID やパスワードとは別のもので、アプリケーションを Mobile Link サーバーに対して識別し、認証するために使用されます。

参照

- ULSyncParms.Password プロパティ [Ultra Light.NET]396 ページ

Version プロパティ

使用する同期スクリプトを指定します。

Visual Basic 構文

```
Public Property Version As String
```

C# 構文

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

備考

使用する同期スクリプトのバージョンを指定する文字列。このパラメーターにはデフォルト値がないので、明示的に設定してください。

統合データベースの同期スクリプトは、それぞれバージョン文字列でマーク付けされます。たとえば、異なる文字列バージョンによって特定される 2 種類の download_cursor スクリプトがあります。Ultra Light アプリケーションは、バージョン文字列により、同期スクリプトのセットから選択できます。

ULSyncProgressData クラス

UL 拡張： 同期の進行状況のモニタリングデータを返します。

Visual Basic 構文

```
Public Class ULSyncProgressData
```

C# 構文

```
public class ULSyncProgressData
```

メンバー

継承されたメンバーを含む ULSyncProgressData クラスのすべてのメンバー。

名前	説明
Flags プロパティ	現在の状態に関連する追加情報を示す、現在の同期フラグを返します。
IsFinalSyncProgress プロパティ	これが最終同期進捗メッセージの場合は、true を返します。
ReceivedBytes プロパティ	現在までに受信したバイト数を返します。
ReceivedDeletes プロパティ	現在までに受信した削除済みのローの数を返します。
ReceivedInserts プロパティ	現在までに受信した挿入済みのローの数を返します。
ReceivedUpdates プロパティ	現在までに受信した更新済みのローの数を返します。
SentBytes プロパティ	現在までに送信されたバイト数を返します。
SentDeletes プロパティ	現在までに送信された削除済みのローの数を返します。
SentInserts プロパティ	現在までに送信された挿入済みのローの数を返します。
SentUpdates プロパティ	現在までに送信された更新済みのローの数を返します。
State プロパティ	現在の同期のステータスを返します。
SyncTableCount プロパティ	同期中のテーブルの数を返します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
SyncTableIndex プロパティ	現在同期中のテーブルのインデックスを返します (テーブルには 1 ～ DatabaseSchema.TableCount までの番号が付けられています)。
TableID プロパティ	現在同期中のテーブルのデータベースインデックスを返します。
TableName プロパティ	アップロード中またはダウンロード中の現在のテーブルの名前を返します。
FLAG_IS_BLOCKING フィールド	Mobile Link サーバーからの応答の待機中、同期はブロックされていることを示すフラグです。

参照

- ULSyncProgressListener インターフェイス [Ultra Light.NET]409 ページ

Flags プロパティ

現在の状態に関連する追加情報を示す、現在の同期フラグを返します。

Visual Basic 構文

```
Public ReadOnly Property Flags As Integer
```

C# 構文

```
public int Flags {get;}
```

備考

フラグの組み合わせを保持する整数。

参照

- ULSyncProgressData.FLAG_IS_BLOCKING フィールド [Ultra Light.NET]409 ページ

IsFinalSyncProgress プロパティ

これが最終同期進捗メッセージの場合は、true を返します。

Visual Basic 構文

```
Public ReadOnly Property IsFinalSyncProgress As Boolean
```

C# 構文

```
public bool IsFinalSyncProgress {get;}
```

備考

これが最終同期進捗メッセージの場合は、true になります。

ReceivedBytes プロパティ

現在までに受信したバイト数を返します。

Visual Basic 構文

```
Public ReadOnly Property ReceivedBytes As Long
```

C# 構文

```
public long ReceivedBytes {get;}
```

備考

この情報は、すべてのステータスで更新されます。

現在までに受信したバイト数。

参照

- ULSyncProgressState 列挙体 [Ultra Light.NET]467 ページ

ReceivedDeletes プロパティ

現在までに受信した削除済みのローの数を返します。

Visual Basic 構文

```
Public ReadOnly Property ReceivedDeletes As Integer
```

C# 構文

```
public int ReceivedDeletes {get;}
```

備考

現在までに受信した削除済みのローの数。

参照

- ULSyncProgressState 列挙体 [Ultra Light.NET]467 ページ

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

ReceivedInserts プロパティ

現在までに受信した挿入済みのローの数を返します。

Visual Basic 構文

```
Public ReadOnly Property ReceivedInserts As Integer
```

C# 構文

```
public int ReceivedInserts {get;}
```

備考

現在までに受信した挿入済みのローの数。

参照

- ULSyncProgressState 列挙体 [Ultra Light.NET]467 ページ

ReceivedUpdates プロパティ

現在までに受信した更新済みのローの数を返します。

Visual Basic 構文

```
Public ReadOnly Property ReceivedUpdates As Integer
```

C# 構文

```
public int ReceivedUpdates {get;}
```

備考

現在までに受信した更新済みのローの数。

参照

- ULSyncProgressState 列挙体 [Ultra Light.NET]467 ページ

SentBytes プロパティ

現在までに送信されたバイト数を返します。

Visual Basic 構文

```
Public ReadOnly Property SentBytes As Long
```

C# 構文

```
public long SentBytes {get;}
```

備考

この情報は、すべてのステータスで更新されます。

現在までに送信されたバイト数。

参照

- ULSyncProgressState 列挙体 [Ultra Light.NET]467 ページ

SentDeletes プロパティ

現在までに送信された削除済みのローの数を返します。

Visual Basic 構文

```
Public ReadOnly Property SentDeletes As Integer
```

C# 構文

```
public int SentDeletes {get;}
```

備考

現在までに送信された削除済みのローの数。

参照

- ULSyncProgressState 列挙体 [Ultra Light.NET]467 ページ

SentInserts プロパティ

現在までに送信された挿入済みのローの数を返します。

Visual Basic 構文

```
Public ReadOnly Property SentInserts As Integer
```

C# 構文

```
public int SentInserts {get;}
```

備考

現在までに送信された挿入済みのローの数。

参照

- ULSyncProgressState 列挙体 [Ultra Light.NET]467 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

SentUpdates プロパティ

現在までに送信された更新済みのローの数を返します。

Visual Basic 構文

```
Public ReadOnly Property SentUpdates As Integer
```

C# 構文

```
public int SentUpdates {get;}
```

備考

現在までに送信された更新済みのローの数。

参照

- ULSyncProgressState 列挙体 [Ultra Light.NET]467 ページ

State プロパティ

現在の同期のステータスを返します。

Visual Basic 構文

```
Public ReadOnly Property State As ULSyncProgressState
```

C# 構文

```
public ULSyncProgressState State {get;}
```

備考

現在の同期のステータスを指定する ULSyncProgressState 値の 1 つ。

参照

- ULSyncProgressState 列挙体 [Ultra Light.NET]467 ページ

SyncTableCount プロパティ

同期中のテーブルの数を返します。

Visual Basic 構文

```
Public ReadOnly Property SyncTableCount As Integer
```

C# 構文

```
public int SyncTableCount {get;}
```

備考

同期中のテーブルの数。テーブルごとに送信と受信のフェーズがあります。したがって、この数は同期されるテーブルの数より多い場合があります。

参照

- ULSyncProgressState 列挙体 [Ultra Light.NET]467 ページ

SyncTableIndex プロパティ

現在同期中のテーブルのインデックスを返します (テーブルには 1 ～ DatabaseSchema.TableCount までの番号が付けられています)。

Visual Basic 構文

```
Public ReadOnly Property SyncTableIndex As Integer
```

C# 構文

```
public int SyncTableIndex {get;}
```

備考

現在同期中のテーブルのインデックス (1 ～ SyncTableCount の値)

参照

- ULSyncProgressState 列挙体 [Ultra Light.NET]467 ページ

TableID プロパティ

現在同期中のテーブルのデータベースインデックスを返します。

Visual Basic 構文

```
Public ReadOnly Property TableID As Integer
```

C# 構文

```
public int TableID {get;}
```

備考

データベースインデックス (1 ～ ULDatabaseSchema.TableCount の値)

参照

- ULSyncProgressState 列挙体 [Ultra Light.NET]467 ページ
- ULDatabaseSchema.TableCount プロパティ [Ultra Light.NET]236 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

TableName プロパティ

アップロード中またはダウンロード中の現在のテーブルの名前を返します。

Visual Basic 構文

```
Public ReadOnly Property TableName As String
```

C# 構文

```
public string TableName {get;}
```

備考

同期される現在のテーブルの名前、該当しない場合は NULL。

FLAG_IS_BLOCKING フィールド

Mobile Link サーバーからの応答の待機中、同期はブロックされていることを示すフラグです。

Visual Basic 構文

```
Public Const FLAG_IS_BLOCKING As Integer
```

C# 構文

```
public const int FLAG_IS_BLOCKING;
```

ULSyncProgressListener インターフェイス

UL 拡張： 同期プログレスイベントを受信するリスナーインターフェイスです。

Visual Basic 構文

```
Public Interface ULSyncProgressListener
```

C# 構文

```
public interface ULSyncProgressListener
```

メンバー

継承されたメンバーを含む ULSyncProgressListener インターフェイスのすべてのメンバー。

名前	説明
SyncProgressed メソッド	ユーザーに進行状況を通知するために、同期処理中に呼び出されます。

参照

- ULConnection.Synchronize メソッド [Ultra Light.NET]151 ページ

SyncProgressed メソッド

ユーザーに進行状況を通知するために、同期処理中に呼び出されます。

Visual Basic 構文

```
Public Function SyncProgressed(
     ByVal data As ULSyncProgressData
) As Boolean
```

C# 構文

```
public bool SyncProgressed(ULSyncProgressData data)
```

パラメーター

- **data** 最新の同期のプログレスデータを保持している ULSyncProgressData オブジェクト。

戻り値

このメソッドは、同期をキャンセルする場合は true を、続行する場合は false を返す必要があります。

備考

このメソッドは、同期をキャンセルする場合は true を、続行する場合は false を返す必要があります。

SyncProgressed の呼び出し中に、Ultra Light.NET API のメソッドを呼び出さないでください。

参照

- ULSyncProgressData クラス [Ultra Light.NET]402 ページ

ULSyncResult クラス

UL 拡張： 前回の同期のステータスを表します。

Visual Basic 構文

```
Public Class ULSyncResult
```

C# 構文

```
public class ULSyncResult
```

メンバー

継承されたメンバーを含む ULSyncResult クラスのすべてのメンバー。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
AuthStatus プロパティ	前回試行された同期の認証ステータスコードを返します。
AuthValue プロパティ	カスタムユーザー認証同期スクリプトからの戻り値を返します。
IgnoredRows プロパティ	前回行われた同期で、アップロードされたローが無視されたかどうかを確認します。
PartialDownloadRetained プロパティ	前回行われた同期で、部分的なダウンロードが保持されたかどうかを確認します。
StreamErrorCode プロパティ	ストリーム自体によってレポートされるエラーを返します。
StreamErrorParameters プロパティ	ストリームエラーパラメーターをカンマで区切ったリストを返します。
StreamErrorSystem プロパティ	ストリームエラーシステム固有のコードを返します。
Timestamp プロパティ	前回の同期のタイムスタンプを返します。
UploadOK プロパティ	前回のアップロード同期が成功したかどうかをチェックします。

備考

このクラスにはコンストラクターがありません。各接続には、ULConnection.SyncResult としてアタッチされた、固有の ULSyncResult インスタンスがあります。ULSyncResult インスタンスが有効なのは、接続が開かれている間だけです。

参照

- ULConnection.SyncResult プロパティ [Ultra Light.NET]161 ページ
- ULConnection.Synchronize メソッド [Ultra Light.NET]151 ページ

AuthStatus プロパティ

前回試行された同期の認証ステータスコードを返します。

Visual Basic 構文

```
Public ReadOnly Property AuthStatus As ULAuthStatusCode
```

C# 構文

```
public ULAuthStatusCode AuthStatus {get;}
```

備考

前回行われた同期の認証ステータスを示す ULAuthStatusCode 値の 1 つ。

参照

- ULAuthStatusCode 列挙体 [Ultra Light.NET]457 ページ

AuthValue プロパティ

カスタムユーザー認証同期スクリプトからの戻り値を返します。

Visual Basic 構文

```
Public ReadOnly Property AuthValue As Long
```

C# 構文

```
public long AuthValue {get;}
```

備考

カスタムユーザー認証同期スクリプトから返された long integer。

IgnoredRows プロパティ

前回行われた同期で、アップロードされたローが無視されたかどうかを確認します。

Visual Basic 構文

```
Public ReadOnly Property IgnoredRows As Boolean
```

C# 構文

```
public bool IgnoredRows {get;}
```

備考

前回の同期中にアップロードされたローが無視された場合は true、ローが無視されなかった場合は false。

参照

- ULSyncParms.DownloadOnly プロパティ [Ultra Light.NET]393 ページ

PartialDownloadRetained プロパティ

前回行われた同期で、部分的なダウンロードが保持されたかどうかを確認します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public ReadOnly Property PartialDownloadRetained As Boolean
```

C# 構文

```
public bool PartialDownloadRetained {get;}
```

備考

ダウンロードが中断され、部分的なダウンロードが保持された場合は true、ダウンロードが中断されなかった場合または部分的なダウンロードがロールバックされた場合は false。

参照

- ULSyncParms.KeepPartialDownload プロパティ [Ultra Light.NET]394 ページ

StreamErrorCode プロパティ

ストリーム自体によってレポートされるエラーを返します。

Visual Basic 構文

```
Public ReadOnly Property StreamErrorCode As ULStreamErrorCode
```

C# 構文

```
public ULStreamErrorCode StreamErrorCode {get;}
```

備考

ストリーム自体によってレポートされるエラーを示す ULStreamErrorCode 値の 1 つ。ただし、エラーが発生しなかった場合は ULStreamErrorCode.NONE。

StreamErrorParameters プロパティ

ストリームエラーパラメーターをカンマで区切ったリストを返します。

Visual Basic 構文

```
Public ReadOnly Property StreamErrorParameters As String
```

C# 構文

```
public string StreamErrorParameters {get;}
```

備考

StreamErrorCode プロパティでレポートされるストリームエラーコードのエラーパラメーターをカンマで区切ったリストが含まれます。エラーにパラメーターがない場合、またエラーが設定されていない場合は、空の文字列になります。

参照

- ULFileTransfer.StreamErrorCode プロパティ [Ultra Light.NET]294 ページ

StreamErrorSystem プロパティ

ストリームエラーシステム固有のコードを返します。

Visual Basic 構文

```
Public ReadOnly Property StreamErrorSystem As Integer
```

C# 構文

```
public int StreamErrorSystem {get;}
```

備考

ストリームエラーシステム固有のコードを示す整数。

Timestamp プロパティ

前回の同期のタイムスタンプを返します。

Visual Basic 構文

```
Public ReadOnly Property Timestamp As Date
```

C# 構文

```
public DateTime Timestamp {get;}
```

備考

前回の同期のタイムスタンプを指定する System.DateTime。

参照

- System.DateTime

UploadOK プロパティ

前回のアップロード同期が成功したかどうかをチェックします。

Visual Basic 構文

```
Public ReadOnly Property UploadOK As Boolean
```

C# 構文

```
public bool UploadOK {get;}
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

前回のアップロード同期が成功であった場合は true、不成功であった場合は false。

ULTable クラス

UL 拡張： Ultra Light データベース内のテーブルを示します。

Visual Basic 構文

```
Public Class ULTable Inherits ULResultSet
```

C# 構文

```
public class ULTable : ULResultSet
```

基本クラス

- ULResultSet クラス [Ultra Light.NET]352 ページ

メンバー

継承されたメンバーを含む ULTable クラスのすべてのメンバー。

名前	説明
AppendBytes メソッド	指定された System.Bytes 配列の指定されたサブセットを、指定された ULDbType.LongBinary カラムの新しい値に追加します。
AppendChars メソッド	指定された System.Chars 配列の指定されたサブセットを、指定された ULDbType.LongVarchar カラムの新しい値に追加します。
Close メソッド	カーソルを閉じます。
Delete メソッド	現在の行を削除します。
DeleteAllRows メソッド	テーブルのすべてのローを削除します。
Dispose method (System.Data.Common.DbDataReader から継承)	System.Data.Common.DbDataReader クラスの現在のインスタンスで使用しているすべてのリソースを解放します。
FindBegin メソッド	テーブルで新規に検索を実行する準備を行います。

名前	説明
FindFirst メソッド	テーブルを先頭から順方向に移動しながら、現在のインデックスの値かそのセット全体に完全に一致するローを検索します。
FindLast メソッド	テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に完全に一致するローを検索します。
FindNext メソッド	現在の位置からテーブルを順方向に移動しながら、次のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindFirst() 検索を続行します。
FindPrevious メソッド	現在の位置からテーブルを逆方向に移動しながら、前のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindLast() 検索を続行します。
GetBoolean メソッド	指定されたカラムの値を System.Boolean として返します。
GetByte メソッド	指定されたカラムの値を符号なし 8 ビット値 (System.Byte) として返します。
GetBytes メソッド	**UL 拡張：** 指定されたカラムの値を System.Bytes 配列として返します。
GetChar メソッド	このメソッドは Ultra Light.NET ではサポートされていません。
GetChars メソッド	指定されたオフセットで始まる、指定された ULDbType.LongVarchar カラムの値のサブセットを、コピー先の System.Char 配列の指定されたオフセットにコピーします。
GetData method (System.Data.Common.DbDataReader から継承)	要求されたカラム序数の System.Data.Common.DbDataReader オブジェクトを返します。
GetDataTypeName メソッド	指定されたカラムのプロバイダーのデータ型の名前を返します。
GetDateTime メソッド	指定されたカラムの値を、ミリ秒の精度の System.DateTime として返します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
GetDbDataReader method (System.Data.Common.DbDataReader から継承)	プロバイダー固有の実装によって無効になる可能性のある、要求されたカラム序数の System.Data.Common.DbDataReader オブジェクトを返します。
GetDecimal メソッド	指定されたカラムの値を System.Decimal として返します。
GetDouble メソッド	指定されたカラムの値を System.Double として返します。
GetEnumerator メソッド	ULDataReader の反復処理を実行する System.Collections.IEnumerator を返します。
GetFieldType メソッド	指定されたカラムに最適な System.Type を返します。
GetFloat メソッド	指定されたカラムの値を System.Single として返します。
GetGuid メソッド	指定されたカラムの値を UUID (System.Guid) として返します。
GetInt16 メソッド	指定されたカラムの値を System.Int16 として返します。
GetInt32 メソッド	指定されたカラムの値を Int32 として返します。
GetInt64 メソッド	指定されたカラムの値を Int64 として返します。
GetName メソッド	指定されたカラムの名前を返します。
GetOrdinal メソッド	指定されたカラムのカラム ID を返します。
GetProviderSpecificFieldType method (System.Data.Common.DbDataReader から継承)	指定されたカラムのプロバイダー固有のフィールドタイプを返します。
GetProviderSpecificValue method (System.Data.Common.DbDataReader から継承)	指定されたカラムの値を System.Object のインスタンスとして取得します。
GetProviderSpecificValues method (System.Data.Common.DbDataReader から継承)	現在のローのコレクション内のプロバイダー固有のすべての属性カラムを取得します。
GetRowCount メソッド	**UL 拡張：** カーソル内のローの数を、スレッショルド以内で返します。

名前	説明
GetSchemaTable メソッド	ULDataReader のカラムのメタデータが記述された System.Data.DataTable を返します。
GetString メソッド	指定されたカラムの値を System.String として返します。
GetTimeSpan メソッド	指定されたカラムの値を、ミリ秒の精度の System.TimeSpan として返します。
GetUInt16 メソッド	指定されたカラムの値を System.UInt16 として返します。
GetUInt32 メソッド	指定されたカラムの値を UInt32 として返します。
GetUInt64 メソッド	指定されたカラムの値を System.UInt64 として返します。
GetValue メソッド	指定されたカラムの値をネイティブフォーマットで返します。
GetValues メソッド	現在のローのすべてのカラム値を返します。
Insert メソッド	現在のカラム値 (set メソッドを使用して指定されます) で新しいローを挿入します。
InsertBegin メソッド	現在のすべてのカラムをデフォルト値に設定して、テーブルに新しいローを挿入する準備を行います。
IsDBNull メソッド	指定されたカラムの値が NULL かどうかをチェックします。
LookupBackward メソッド	テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より小さい値を持つローを検索します。
LookupBegin メソッド	テーブルで新規に検索を実行する準備を行います。
LookupForward メソッド	テーブルを最初から順方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より大きい値を持つローを検索します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
MoveAfterLast メソッド	**UL 拡張：** カーソルの最後のローの後に、カーソルを配置します。
MoveBeforeFirst メソッド	**UL 拡張：** カーソルの最初のローの前に、カーソルを配置します。
MoveFirst メソッド	**UL 拡張：** カーソルの最初のローに、カーソルを配置します。
MoveLast メソッド	**UL 拡張：** カーソルの最後のローに、カーソルを配置します。
MoveNext メソッド	**UL 拡張：** カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。
MovePrevious メソッド	**UL 拡張：** カーソルを前のローに配置するか、最初のローの前に配置します。
MoveRelative メソッド	**UL 拡張：** 現在のローを基準としてカーソルを配置します。
NextResult メソッド	バッチ SQL 文の結果を読み込むときに ULDataReader を次の結果に進めます。
Read メソッド	カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。
SetBoolean メソッド	指定されたカラムの値を、System.Boolean を使用して設定します。
SetByte メソッド	指定されたカラムの値を、System.Byte (符号なし 8 ビット整数) を使用して設定します。
SetBytes メソッド	指定されたカラムの値を、System.Bytes を使用して設定します。
SetDateTime メソッド	指定されたカラムの値を、System.DateTime を使用して設定します。
SetDBNull メソッド	カラムを NULL に設定します。
SetDecimal メソッド	指定されたカラムの値を、System.Decimal を使用して設定します。

名前	説明
SetDouble メソッド	指定されたカラムの値を、System.Double を使用して設定します。
SetFloat メソッド	指定されたカラムの値を、System.Single を使用して設定します。
SetGuid メソッド	指定されたカラムの値を、System.Guid を使用して設定します。
SetInt16 メソッド	指定されたカラムの値を、System.Int16 を使用して設定します。
SetInt32 メソッド	指定されたカラムの値を、System.Int32 を使用して設定します。
SetInt64 メソッド	指定されたカラムの値を、Int64 を使用して設定します。
SetString メソッド	指定されたカラムの値を、System.String を使用して設定します。
SetTimeSpan メソッド	指定されたカラムの値を、System.TimeSpan を使用して設定します。
SetToDefault メソッド	指定されたカラムの値を、そのデフォルト値に設定します。
SetUInt16 メソッド	指定されたカラムの値を、System.UInt16 を使用して設定します。
SetUInt32 メソッド	指定されたカラムの値を、System.UInt32 を使用して設定します。
SetUInt64 メソッド	指定されたカラムの値を、System.UInt64 を使用して設定します。
Truncate メソッド	テーブル内のすべてのローを削除し、STOP SYNCHRONIZATION DELETE を一時的にアクティブにします。
Update メソッド	現在のカラム値 (set メソッドを使用して指定されます) で新しいローを更新します。
UpdateBegin メソッド	現在のローを更新する準備を行います。
Depth プロパティ	現在のローのネストの深さを返します。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
FieldCount プロパティ	このカーソル内のカラム数を返します。
HasRows プロパティ	ULDataReader に 1 つまたは複数のローがあるかどうかをチェックします。
IsBOF プロパティ	**UL 拡張：** 現在のローの位置が最初のローの前であるかどうかを確認します。
IsClosed プロパティ	カーソルが現在開いているかどうかを確認します。
IsEOF プロパティ	**UL 拡張：** 現在のローの位置が最後のローの後であるかどうかを確認します。
RecordsAffected プロパティ	SQL 文の実行によって変更、挿入、または削除されたローの数を返します。
RowCount プロパティ	**UL 拡張：** カーソル内のロー数を返します。
Schema プロパティ	テーブルスキーマを保持します。
this プロパティ	指定されたカラムの値をネイティブフォーマットで返します。
VisibleFieldCount property (System.Data.Common.DbDataReader から継承)	System.Data.Common.DbDataReader の非表示でないフィールドの数を取得します。

備考

このクラスにはコンストラクターがありません。テーブルは、ULCommand の ULCommand.ExecuteTable() を使用して作成されます。

参照

- ULCommand.ExecuteTable メソッド [Ultra Light.NET]94 ページ
- ULCommand クラス [Ultra Light.NET]66 ページ
- ULResultSet クラス [Ultra Light.NET]352 ページ
- System.Data.IDataReader
- System.Data.IDataRecord
- System.IDisposable

DeleteAllRows メソッド

テーブルのすべてのローを削除します。

Visual Basic 構文

```
Public Sub DeleteAllRows()
```

C# 構文

```
public void DeleteAllRows()
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

アプリケーションによっては、テーブル内のローをすべて削除してから、新しいデータセットをテーブルにダウンロードする方が便利なことがあります。
ULConnection.StopSynchronizationDelete を使用すると、統合データベースからは削除しないで Ultra Light データベースからローを削除できます。

参照

- ULTable.Truncate メソッド [Ultra Light.NET]436 ページ
- ULConnection.StopSynchronizationDelete メソッド [Ultra Light.NET]150 ページ

FindBegin メソッド

テーブルで新規に検索を実行する準備を行います。

Visual Basic 構文

```
Public Sub FindBegin()
```

C# 構文

```
public void FindBegin()
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

検索する値は、テーブルを開いたインデックス内のカラムで適切な setType メソッドを呼び出して指定します。

参照

- ULTable.FindFirst メソッド [Ultra Light.NET]423 ページ
- ULTable.FindLast メソッド [Ultra Light.NET]425 ページ

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

FindFirst メソッド

テーブルを先頭から順方向に移動しながら、現在のインデックスの値かそのセット全体に完全に一致するローを検索します。

オーバーロードリスト

名前	説明
FindFirst() メソッド	テーブルを先頭から順方向に移動しながら、現在のインデックスの値かそのセット全体に完全に一致するローを検索します。
FindFirst(short) メソッド	テーブルを先頭から順方向に移動しながら、現在のインデックスの値かそのセットの一部に完全に一致するローを検索します。

FindFirst() メソッド

テーブルを先頭から順方向に移動しながら、現在のインデックスの値かそのセット全体に完全に一致するローを検索します。

Visual Basic 構文

```
Public Function FindFirst() As Boolean
```

C# 構文

```
public bool FindFirst()
```

戻り値

成功した場合は true、失敗した場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値と完全に一致した最初のローで停止します。失敗すると、カーソル位置は最後のローの後ろ (ULDataReader.IsEOF) になります。

検索を行う前に FindBegin を呼び出してください。

参照

- ULTable.FindBegin メソッド [Ultra Light.NET]422 ページ
- ULTable.FindNext メソッド [Ultra Light.NET]427 ページ
- ULTable.FindPrevious メソッド [Ultra Light.NET]428 ページ
- ULTable.FindFirst メソッド [Ultra Light.NET]423 ページ
- ULDataReader.IsEOF プロパティ [Ultra Light.NET]270 ページ
- ULTable.FindBegin メソッド [Ultra Light.NET]422 ページ

FindFirst(short) メソッド

テーブルを先頭から順方向に移動しながら、現在のインデックスの値かそのセットの一部に完全に一致するローを検索します。

Visual Basic 構文

```
Public Function FindFirst(ByVal numColumns As Short) As Boolean
```

C# 構文

```
public bool FindFirst(short numColumns)
```

パラメーター

- **numColumns**　複合インデックスのための、検索で使用するカラムの数。たとえば、3 つのカラムのインデックスがあり、最初のカラムにのみ基づいて一致する値を検索する場合は、最初のカラムに値を設定してから、値を 1 に指定します。

戻り値

成功した場合は true、失敗した場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値と完全に一致した最初のローで停止します。失敗すると、カーソル位置は最後のローの後ろ (ULDataReader.IsEOF) になります。

検索を行う前に FindBegin を呼び出してください。

参照

- ULTable.FindBegin メソッド [Ultra Light.NET]422 ページ
- ULTable.FindNext メソッド [Ultra Light.NET]427 ページ
- ULTable.FindPrevious メソッド [Ultra Light.NET]428 ページ
- ULTable.FindFirst メソッド [Ultra Light.NET]423 ページ
- ULDataReader.IsEOF プロパティ [Ultra Light.NET]270 ページ
- ULTable.FindBegin メソッド [Ultra Light.NET]422 ページ

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

FindLast メソッド

テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に完全に一致するローを検索します。

オーバーロードリスト

名前	説明
FindLast() メソッド	テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に完全に一致するローを検索します。
FindLast(short) メソッド	テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセットの一部に完全に一致するローを検索します。

FindLast() メソッド

テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に完全に一致するローを検索します。

Visual Basic 構文

```
Public Function FindLast() As Boolean
```

C# 構文

```
public bool FindLast()
```

戻り値

成功した場合は true、失敗した場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値と完全に一致した最初のローで停止します。失敗すると、カーソル位置は最初のローの前 (ULDataReader.IsBOF) になります。

検索を行う前に FindBegin を呼び出してください。

参照

- ULTable.FindBegin メソッド [Ultra Light.NET]422 ページ
- ULTable.FindNext メソッド [Ultra Light.NET]427 ページ
- ULTable.FindPrevious メソッド [Ultra Light.NET]428 ページ
- ULTable.FindLast メソッド [Ultra Light.NET]425 ページ
- ULDataReader.IsBOF プロパティ [Ultra Light.NET]269 ページ
- ULTable.FindBegin メソッド [Ultra Light.NET]422 ページ

FindLast(short) メソッド

テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセットの一部に完全に一致するローを検索します。

Visual Basic 構文

```
Public Function FindLast(ByVal numColumns As Short) As Boolean
```

C# 構文

```
public bool FindLast(short numColumns)
```

パラメーター

- **numColumns**　複合インデックスのための、検索で使用するカラムの数。たとえば、3 つのカラムのインデックスがあり、最初のカラムにのみ基づいて一致する値を検索する場合は、最初のカラムに値を設定してから、値を 1 に指定します。

戻り値

成功した場合は true、失敗した場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値と完全に一致した最初のローで停止します。失敗すると、カーソル位置は最初のローの前 (ULDataReader.IsBOF) になります。

検索を行う前に FindBegin を呼び出してください。

参照

- ULTable.FindBegin メソッド [Ultra Light.NET]422 ページ
- ULTable.FindNext メソッド [Ultra Light.NET]427 ページ
- ULTable.FindPrevious メソッド [Ultra Light.NET]428 ページ
- ULTable.FindLast メソッド [Ultra Light.NET]425 ページ
- ULDataReader.IsBOF プロパティ [Ultra Light.NET]269 ページ
- ULTable.FindBegin メソッド [Ultra Light.NET]422 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

FindNext メソッド

現在の位置からテーブルを順方向に移動しながら、次のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindFirst() 検索を続行します。

オーバーロードリスト

名前	説明
FindNext() メソッド	現在の位置からテーブルを順方向に移動しながら、次のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindFirst() 検索を続行します。
FindNext(short) メソッド	現在の位置からテーブルを順方向に移動しながら、次のローが現在のインデックスの値かそのセットの一部に完全に一致するかどうかを調べて、ULTable.FindFirst() 検索を続行します。

FindNext() メソッド

現在の位置からテーブルを順方向に移動しながら、次のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindFirst() 検索を続行します。

Visual Basic 構文

```
Public Function FindNext() As Boolean
```

C# 構文

```
public bool FindNext()
```

戻り値

成功した場合は true、失敗した場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

インデックスの値と完全に一致すると、カーソルは次のローで停止します。失敗すると、カーソル位置は最後のローの後ろ (ULDataReader.IsEOF) になります。

検索されるカラムの値がローの更新において修正された場合の FindNext の動作は不確定です。

参照

- ULTable.FindFirst メソッド [Ultra Light.NET]423 ページ
- ULTable.FindNext メソッド [Ultra Light.NET]427 ページ
- ULDataReader.IsEOF プロパティ [Ultra Light.NET]270 ページ

FindNext(short) メソッド

現在の位置からテーブルを順方向に移動しながら、次のローが現在のインデックスの値かそのセットの一部に完全に一致するかどうかを調べて、ULTable.FindFirst() 検索を続行します。

Visual Basic 構文

```
Public Function FindNext(ByVal numColumns As Short) As Boolean
```

C# 構文

```
public bool FindNext(short numColumns)
```

パラメーター

- **numColumns**　複合インデックスのための、検索で使用するカラムの数。たとえば、3 つのカラムのインデックスがあり、最初のカラムにのみ基づいて一致する値を検索する場合は、最初のカラムに値を設定してから、値を 1 に指定します。

戻り値

成功した場合は true、失敗した場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

インデックスの値と完全に一致すると、カーソルは次のローで停止します。失敗すると、カーソル位置は最後のローの後ろ (ULDataReader.IsEOF) になります。

検索されるカラムの値がローの更新において修正された場合の FindNext の動作は不確定です。

参照

- ULTable.FindFirst メソッド [Ultra Light.NET]423 ページ
- ULTable.FindNext メソッド [Ultra Light.NET]427 ページ
- ULTable.FindFirst メソッド [Ultra Light.NET]423 ページ
- ULDataReader.IsEOF プロパティ [Ultra Light.NET]270 ページ

FindPrevious メソッド

現在の位置からテーブルを逆方向に移動しながら、前のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindLast() 検索を続行します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

オーバーロードリスト

名前	説明
FindPrevious() メソッド	現在の位置からテーブルを逆方向に移動しながら、前のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindLast() 検索を続行します。
FindPrevious(short) メソッド	現在の位置からテーブルを逆方向に移動しながら、前のローが現在のインデックスの値かそのセットの一部に完全に一致するかどうかを調べて、ULTable.FindLast() 検索を続行します。

FindPrevious() メソッド

現在の位置からテーブルを逆方向に移動しながら、前のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindLast() 検索を続行します。

Visual Basic 構文

```
Public Function FindPrevious() As Boolean
```

C# 構文

```
public bool FindPrevious()
```

戻り値

成功した場合は true、失敗した場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

インデックスの値と完全に一致すると、カーソルは前のローで停止します。失敗すると、カーソル位置は最初のローの前 (ULDataReader.IsBOF) になります。

検索されるカラムの値がローの更新において修正された場合の FindPrevious の動作は不確定です。

参照

- ULTable.FindLast メソッド [Ultra Light.NET]425 ページ
- ULTable.FindPrevious メソッド [Ultra Light.NET]428 ページ
- ULDataReader.IsBOF プロパティ [Ultra Light.NET]269 ページ

FindPrevious(short) メソッド

現在の位置からテーブルを逆方向に移動しながら、前のローが現在のインデックスの値かそのセットの一部に完全に一致するかどうかを調べて、ULTable.FindLast() 検索を続行します。

Visual Basic 構文

```
Public Function FindPrevious(ByVal numColumns As Short) As Boolean
```

C# 構文

```
public bool FindPrevious(short numColumns)
```

パラメーター

- **numColumns**　複合インデックスのための、検索で使用するカラムの数。たとえば、3 つのカラムのインデックスがあり、最初のカラムにのみ基づいて一致する値を検索する場合は、最初のカラムに値を設定してから、値を 1 に指定します。

戻り値

成功した場合は true、失敗した場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

インデックスの値と完全に一致すると、カーソルは前のローで停止します。失敗すると、カーソル位置は最初のローの前 (ULDataReader.IsBOF) になります。

検索されるカラムの値がローの更新において修正された場合の FindPrevious の動作は不確定です。

参照

- ULTable.FindLast メソッド [Ultra Light.NET]425 ページ
- ULTable.FindPrevious メソッド [Ultra Light.NET]428 ページ
- ULDataReader.IsBOF プロパティ [Ultra Light.NET]269 ページ

Insert メソッド

現在のカラム値 (set メソッドを使用して指定されます) で新しいローを挿入します。

Visual Basic 構文

```
Public Sub Insert()
```

C# 構文

```
public void Insert()
```


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

挿入を行う前に ULTable.InsertBegin を呼び出してください。

参照

- ULTable.InsertBegin メソッド [Ultra Light.NET]431 ページ

InsertBegin メソッド

現在のすべてのカラムをデフォルト値に設定して、テーブルに新しいローを挿入する準備を行います。

Visual Basic 構文

```
Public Sub InsertBegin()
```

C# 構文

```
public void InsertBegin()
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

適切な SetType メソッドまたは AppendType メソッドを呼び出して、挿入するデフォルト以外の値を指定します。

insert メソッドが実行されないと、ローが実際に挿入されることも、ロー内のデータが実際に変更されることもありません。また、その変更がコミットされないかぎり、永続化されません。

参照

- ULTable.Insert メソッド [Ultra Light.NET]430 ページ

LookupBackward メソッド

テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より小さい値を持つローを検索します。

オーバーロードリスト

名前	説明
LookupBackward() メソッド	テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より小さい値を持つローを検索します。
LookupBackward(short) メソッド	テーブルを最後から逆方向に移動しながら、現在のインデックス内の特定の値または値の部分的セットに一致するか、それより小さい値を持つローを検索します。

LookupBackward() メソッド

テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より小さい値を持つローを検索します。

Visual Basic 構文

```
Public Function LookupBackward() As Boolean
```

C# 構文

```
public bool LookupBackward()
```

戻り値

成功した場合は true、失敗した場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値に一致する最初のローか、それより少ない値の最初のローで停止します。失敗した場合 (検索する値より小さい値のローがない場合)、カーソル位置は最初のローの前 (ULDataReader.IsBOF) になります。

検索を行う前に LookupBegin を呼び出してください。

参照

- ULTable.LookupBegin メソッド [Ultra Light.NET]433 ページ
- ULTable.LookupBackward メソッド [Ultra Light.NET]431 ページ
- ULDataReader.IsBOF プロパティ [Ultra Light.NET]269 ページ

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

LookupBackward(short) メソッド

テーブルを最後から逆方向に移動しながら、現在のインデックス内の特定の値または値の部分的セットに一致するか、それより小さい値を持つローを検索します。

Visual Basic 構文

```
Public Function LookupBackward(ByVal numColumns As Short) As Boolean
```

C# 構文

```
public bool LookupBackward(short numColumns)
```

パラメーター

- **numColumns**　複合インデックスのための、ルックアップで使用するカラムの数。たとえば、3 つのカラムのインデックスがあり、最初のカラムにのみ基づいて一致する値を検索する場合は、最初のカラムに値を設定してから、値を 1 に指定します。

戻り値

成功した場合は true、失敗した場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値に一致する最初のローか、それより少ない値の最初のローで停止します。失敗した場合 (検索する値より小さい値のローがない場合)、カーソル位置は最初のローの前 (ULDataReader.IsBOF) になります。

検索を行う前に LookupBegin を呼び出してください。

参照

- ULTable.LookupBegin メソッド [Ultra Light.NET]433 ページ
- ULTable.LookupBackward メソッド [Ultra Light.NET]431 ページ
- ULDataReader.IsBOF プロパティ [Ultra Light.NET]269 ページ

LookupBegin メソッド

テーブルで新規に検索を実行する準備を行います。

Visual Basic 構文

```
Public Sub LookupBegin()
```

C# 構文

```
public void LookupBegin()
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

検索する値は、テーブルを開いたインデックス内のカラムで適切な setType メソッドを呼び出して指定します。

参照

- ULTable.LookupForward メソッド [Ultra Light.NET]434 ページ
- ULTable.LookupBackward メソッド [Ultra Light.NET]431 ページ

LookupForward メソッド

テーブルを最初から順方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より大きい値を持つローを検索します。

オーバーロードリスト

名前	説明
LookupForward() メソッド	テーブルを最初から順方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より大きい値を持つローを検索します。
LookupForward(short) メソッド	テーブルを最初から順方向に移動しながら、現在のインデックスの値またはそのセットの一部に一致するか、その値より大きい値を持つローを検索します。

LookupForward() メソッド

テーブルを最初から順方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より大きい値を持つローを検索します。

Visual Basic 構文

```
Public Function LookupForward() As Boolean
```

C# 構文

```
public bool LookupForward()
```

戻り値

成功した場合は true、失敗した場合は false。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値に一致するか、それより大きい値の最初のローで停止します。失敗した場合 (検索する値より大きい値のローがない場合)、カーソル位置は最後のローの後ろ (ULDataReader.IsEOF) になります。

検索を行う前に ULTable.LookupBegin() を呼び出してください。

参照

- ULTable.LookupBegin メソッド [Ultra Light.NET]433 ページ
- ULTable.LookupForward メソッド [Ultra Light.NET]434 ページ
- ULDataReader.IsEOF プロパティ [Ultra Light.NET]270 ページ

LookupForward(short) メソッド

テーブルを最初から順方向に移動しながら、現在のインデックスの値またはそのセットの一部に一致するか、その値より大きい値を持つローを検索します。

Visual Basic 構文

```
Public Function LookupForward(ByVal numColumns As Short) As Boolean
```

C# 構文

```
public bool LookupForward(short numColumns)
```

パラメーター

- **numColumns**　複合インデックスのための、ルックアップで使用するカラムの数。たとえば、3 つのカラムのインデックスがあり、最初のカラムにのみ基づいて一致する値を検索する場合は、最初のカラムに値を設定してから、値を 1 に指定します。

戻り値

成功した場合は true、失敗した場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値に一致するか、それより大きい値の最初のローで停止します。失敗した場合 (検索する値より大きい値のローがない場合)、カーソル位置は最後のローの後ろ (ULDataReader.IsEOF) になります。

検索を行う前に LookupBegin を呼び出してください。

参照

- ULTable.LookupBegin メソッド [Ultra Light.NET]433 ページ
- ULTable.LookupForward メソッド [Ultra Light.NET]434 ページ
- ULDataReader.IsEOF プロパティ [Ultra Light.NET]270 ページ
- ULTable.LookupBegin メソッド [Ultra Light.NET]433 ページ

Truncate メソッド

テーブル内のすべてのローを削除し、STOP SYNCHRONIZATION DELETE を一時的にアクティブにします。

Visual Basic 構文

```
Public Sub Truncate()
```

C# 構文

```
public void Truncate()
```

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULTable.DeleteAllRows メソッド [Ultra Light.NET]421 ページ

Schema プロパティ

テーブルスキーマを保持します。

Visual Basic 構文

```
Public ReadOnly Shadows Property Schema As ULTableSchema
```

C# 構文

```
public new ULTableSchema Schema {get;}
```

備考

このプロパティが有効なのは、接続が開かれている間だけです。

テーブルスキーマを表す ULTableSchema オブジェクト。

このプロパティは、ULDataReader.GetSchemaTable からの結果に示されない Ultra Light.NET の詳細情報を含め、テーブルの完全なスキーマを表します。

参照

- ULTableSchema クラス [Ultra Light.NET]437 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

ULTableSchema クラス

UL 拡張： Ultra Light テーブルのスキーマを示します。

Visual Basic 構文

```
Public NotInheritable Class ULTableSchema Inherits ULCursorSchema
```

C# 構文

```
public sealed class ULTableSchema : ULCursorSchema
```

基本クラス

- ULCursorSchema クラス [Ultra Light.NET]201 ページ

メンバー

継承されたメンバーを含む ULTableSchema クラスのすべてのメンバー。

名前	説明
GetColumnDefaultValue メソッド	指定されたカラムのデフォルト値を返します。
GetColumnID メソッド	指定されたカラムのカラム ID を返します。
GetColumnName メソッド	指定されたカラム ID で識別されたカラムの名前を返します。
GetColumnPartitionSize メソッド	指定されたカラムに割り当てられているグローバルオートインクリメントの分割サイズを返します。
GetColumnPrecision メソッド	カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの精度を返します。
GetColumnScale メソッド	カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの位取りを返します。
GetColumnSize メソッド	カラムがサイズ指定されたカラム (BINARY SQL 型または CHAR SQL 型) の場合は、指定されたカラム ID によって識別されたカラムのサイズを返します。
GetColumnSQLName メソッド	指定されたカラム ID で識別されたカラムの名前を返します。

名前	説明
GetColumnULDbType メソッド	指定されたカラム ID によって識別されたカラムの Ultra Light.NET データ型を返します。
GetIndex メソッド	指定されたインデックスのインデックススキーマを返します。
GetIndexName メソッド	指定されたインデックス ID で識別されたインデックスの名前を返します。
GetOptimalIndex メソッド	指定されたカラムを使用してテーブルを検索するために最適なインデックスです。
GetPublicationPredicate メソッド	指定されたパブリケーションに含まれているテーブルのパブリケーション述部を返します。
GetSchemaTable メソッド	ULDataReader オブジェクトのカラムのスキーマが記述された System.Data.DataTable を返します。
IsColumnAutoIncrement メソッド	指定されたカラムのデフォルトがオートインクリメントに設定されているかどうかをチェックします。
IsColumnCurrentDate メソッド	指定されたカラムのデフォルトが、現在の日付 (ULDbType.Date) に設定されているかどうかをチェックします。
IsColumnCurrentTime メソッド	指定されたカラムのデフォルトが、現在の時刻 (ULDbType.Time) に設定されているかどうかをチェックします。
IsColumnCurrentTimestamp メソッド	指定されたカラムのデフォルトが、現在のタイムスタンプ (ULDbType.TimeStamp) に設定されているかどうかをチェックします。
IsColumnCurrentUTCTimestamp メソッド	指定されたカラムのデフォルトが、現在の UTC のタイムスタンプ (ULDbType.TimeStamp) に設定されているかどうかをチェックします。
IsColumnGlobalAutoIncrement メソッド	指定されたカラムのデフォルトがグローバルオートインクリメントに設定されているかどうかをチェックします。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

名前	説明
IsColumnNewUUID メソッド	指定されたカラムのデフォルトが新しい UUID (System.Guid) に設定されているかどうかをチェックします。
IsColumnNullable メソッド	指定されたカラムが NULL 入力可であるかどうかをチェックします。
IsInPublication メソッド	テーブルが、指定されたパブリケーションに含まれているかどうかをチェックします。
ColumnCount プロパティ	このカーソル内のカラム数を返します。
IndexCount プロパティ	テーブルのインデックス数を返します。
IsNeverSynchronized プロパティ	テーブルが、まったく同期されないようにマーク付けされているかどうかをチェックします。
IsOpen プロパティ	カーソルのスキーマが現在開いているかどうかを確認します。
Name プロパティ	テーブルの名前を返します。
PrimaryKey プロパティ	テーブルのプライマリキーのインデックススキーマを返します。
UploadUnchangedRows プロパティ	データベースが、変更されていないローをアップロードするかどうかをチェックします。

備考

このクラスにはコンストラクターがありません。ULTableSchema オブジェクトは、その ULTable.Schema としてテーブルにアタッチされます。

参照

- ULTableSchema クラス [Ultra Light.NET]437 ページ
- ULTable.Schema プロパティ [Ultra Light.NET]436 ページ
- ULCursorSchema クラス [Ultra Light.NET]201 ページ

GetColumnDefaultValue メソッド

指定されたカラムのデフォルト値を返します。

Visual Basic 構文

```
Public Function GetColumnDefaultValue(
     ByVal columnID As Integer
) As String
```

C# 構文

```
public string GetColumnDefaultValue(int columnID)
```

パラメーター

- **columnID**　カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内であることが必要です。テーブルの先頭カラムの ID 値は 0 です。

戻り値

指定されたカラムの文字列としてのデフォルト値。デフォルト値が NULL の場合は NULL 参照 (Visual Basic の Nothing)。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ

GetColumnPartitionSize メソッド

指定されたカラムに割り当てられているグローバルオートインクリメントの分割サイズを返します。

Visual Basic 構文

```
Public Function GetColumnPartitionSize(
     ByVal columnID As Integer
) As ULong
```

C# 構文

```
public ulong GetColumnPartitionSize(int columnID)
```

パラメーター

- **columnID**　カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内であることが必要です。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムのグローバルオートインクリメントの分割サイズ (System.UInt64)。

例外

- **ULException クラス**　SQL エラーが発生しました。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

テーブルのすべてのグローバルオートインクリメントカラムは、同じグローバルオートインクリメントの分割サイズを共有します。

参照

- ULTableSchema.IsColumnGlobalAutoIncrement メソッド [Ultra Light.NET]446 ページ
- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ
- System.UInt64

GetIndex メソッド

指定されたインデックスのインデックススキーマを返します。

Visual Basic 構文

```
Public Function GetIndex(ByVal name As String) As ULIndexSchema
```

C# 構文

```
public ULIndexSchema GetIndex(string name)
```

パラメーター

- **name**　インデックスの名前。

戻り値

指定されたインデックスを表す ULIndexSchema オブジェクト。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULIndexSchema クラス [Ultra Light.NET]301 ページ

GetIndexName メソッド

指定されたインデックス ID で識別されたインデックスの名前を返します。

Visual Basic 構文

```
Public Function GetIndexName(ByVal indexID As Integer) As String
```

C# 構文

```
public string GetIndexName(int indexID)
```

パラメーター

- **indexID**　インデックスの ID。値は、[1,IndexCount] の範囲内であることが必要です。

戻り値

文字列として返されるインデックス名。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

インデックスの ID とカウントは、スキーマのアップグレード中に変更されることがあります。インデックスを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後にリフレッシュします。

参照

- ULTableSchema.IndexCount プロパティ [Ultra Light.NET]448 ページ

GetOptimalIndex メソッド

指定されたカラムを使用してテーブルを検索するために最適なインデックスです。

Visual Basic 構文

```
Public Function GetOptimalIndex(ByVal columnID As Integer) As String
```

C# 構文

```
public string GetOptimalIndex(int columnID)
```

パラメーター

- **columnID**　カラムの ID 番号。テーブルの先頭カラムの ID 値は 0 です。

戻り値

指定されたカラムの最適なインデックスを表す ULIndexSchema オブジェクト。

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

指定されたカラムは、インデックス内の最初のカラムですが、インデックスには複数のカラムがある場合があります。

参照

- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ
- ULIndexSchema クラス [Ultra Light.NET]301 ページ

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

GetPublicationPredicate メソッド

指定されたパブリケーションに含まれているテーブルのパブリケーション述部を返します。

Visual Basic 構文

```
Public Function GetPublicationPredicate(
     ByVal pubName As String
) As String
```

C# 構文

```
public string GetPublicationPredicate(string pubName)
```

パラメーター

- **pubName**　パブリケーションの名前。

戻り値

文字列として返されるパブリケーション述部。

例外

- **ULException クラス**　SQL エラーが発生しました。

IsColumnAutoIncrement メソッド

指定されたカラムのデフォルトがオートインクリメントに設定されているかどうかをチェックします。

Visual Basic 構文

```
Public Function IsColumnAutoIncrement(
     ByVal columnID As Integer
) As Boolean
```

C# 構文

```
public bool IsColumnAutoIncrement(int columnID)
```

パラメーター

- **columnID**　カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内であることが必要です。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムがオートインクリメントされる場合は true、オートインクリメントされない場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ

IsColumnCurrentDate メソッド

指定されたカラムのデフォルトが、現在の日付 (ULDbType.Date) に設定されているかどうかをチェックします。

Visual Basic 構文

```
Public Function IsColumnCurrentDate(
     ByVal columnID As Integer
) As Boolean
```

C# 構文

```
public bool IsColumnCurrentDate(int columnID)
```

パラメーター

- **columnID**　カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内であることが必要です。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムのデフォルトが現在の日付に設定されている場合は true、現在の日付に設定されていない場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ

IsColumnCurrentTime メソッド

指定されたカラムのデフォルトが、現在の時刻 (ULDbType.Time) に設定されているかどうかをチェックします。

Visual Basic 構文

```
Public Function IsColumnCurrentTime(
     ByVal columnID As Integer
) As Boolean
```

C# 構文

```
public bool IsColumnCurrentTime(int columnID)
```


Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

パラメーター

- **columnID**　カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内であることが必要です。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムのデフォルトが現在の時刻に設定されている場合は true、現在の時刻に設定されていない場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ

IsColumnCurrentTimestamp メソッド

指定されたカラムのデフォルトが、現在のタイムスタンプ (ULDbType.TimeStamp) に設定されているかどうかをチェックします。

Visual Basic 構文

```
Public Function IsColumnCurrentTimestamp(
     ByVal columnID As Integer
) As Boolean
```

C# 構文

```
public bool IsColumnCurrentTimestamp(int columnID)
```

パラメーター

- **columnID**　カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内であることが必要です。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムのデフォルトが現在のタイムスタンプに設定されている場合は true、現在のタイムスタンプに設定されていない場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ

IsColumnCurrentUTCTimestamp メソッド

指定されたカラムのデフォルトが、現在の UTC のタイムスタンプ (ULDbType.TimeStamp) に設定されているかどうかをチェックします。

Visual Basic 構文

```
Public Function IsColumnCurrentUTCTimestamp(
     ByVal columnID As Integer
) As Boolean
```

C# 構文

```
public bool IsColumnCurrentUTCTimestamp(int columnID)
```

パラメーター

- **columnID**　カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内であることが必要です。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムのデフォルトが現在のタイムスタンプに設定されている場合は true、現在のタイムスタンプに設定されていない場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ

IsColumnGlobalAutoIncrement メソッド

指定されたカラムのデフォルトがグローバルオートインクリメントに設定されているかどうかをチェックします。

Visual Basic 構文

```
Public Function IsColumnGlobalAutoIncrement(
     ByVal columnID As Integer
) As Boolean
```

C# 構文

```
public bool IsColumnGlobalAutoIncrement(int columnID)
```

パラメーター

- **columnID**　カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内であることが必要です。テーブルの先頭カラムの ID 値は 0 です。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

戻り値

カラムがグローバルオートインクリメントされる場合は true、グローバルオートインクリメントされない場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULTableSchema.GetColumnPartitionSize メソッド [Ultra Light.NET]440 ページ
- ULConnection.DatabaseID プロパティ [Ultra Light.NET]157 ページ
- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ

IsColumnNewUUID メソッド

指定されたカラムのデフォルトが新しい UUID (System.Guid) に設定されているかどうかをチェックします。

Visual Basic 構文

```
Public Function IsColumnNewUUID(ByVal columnID As Integer) As Boolean
```

C# 構文

```
public bool IsColumnNewUUID(int columnID)
```

パラメーター

- **columnID**　カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内であることが必要です。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムのデフォルトが新しい UUID に設定されている場合は true、新しい UUID に設定されていない場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ
- System.Guid

IsColumnNullable メソッド

指定されたカラムが NULL 入力可であるかどうかをチェックします。

Visual Basic 構文

```
Public Function IsColumnNullable(ByVal columnID As Integer) As Boolean
```

C# 構文

```
public bool IsColumnNullable(int columnID)
```

パラメーター

- **columnID**　カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内であることが必要です。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムが NULL 入力可の場合は true、NULL 入力不可の場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

参照

- ULCursorSchema.ColumnCount プロパティ [Ultra Light.NET]208 ページ

IsInPublication メソッド

テーブルが、指定されたパブリケーションに含まれているかどうかをチェックします。

Visual Basic 構文

```
Public Function IsInPublication(ByVal pubName As String) As Boolean
```

C# 構文

```
public bool IsInPublication(string pubName)
```

パラメーター

- **pubName**　パブリケーションの名前。

戻り値

テーブルがパブリケーション内にある場合は true、パブリケーション内にない場合は false。

例外

- **ULException クラス**　SQL エラーが発生しました。

IndexCount プロパティ

テーブルのインデックス数を返します。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

Visual Basic 構文

```
Public ReadOnly Property IndexCount As Integer
```

C# 構文

```
public int IndexCount {get;}
```

備考

テーブルのインデックスの数。テーブルスキーマが閉じている場合は 0。

インデックス ID の範囲は、1 ～ IndexCount です。

注意
インデックスの ID とカウントは、スキーマのアップグレード中に変更されることがあります。インデックスを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後にリフレッシュします。

IsNeverSynchronized プロパティ

テーブルが、まったく同期されないようにマーク付けされているかどうかをチェックします。

Visual Basic 構文

```
Public ReadOnly Property IsNeverSynchronized As Boolean
```

C# 構文

```
public bool IsNeverSynchronized {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

テーブルが、まったく同期されないようにマーク付けされている場合は true、そうでない場合は false。

まったく同期されないようにマーク付けされているテーブルは、パブリケーションに含まれているものであっても、まったく同期されていません。このようなテーブルは、「非同期」テーブルと呼ばれることもあります。

Name プロパティ

テーブルの名前を返します。

Visual Basic 構文

```
Public ReadOnly Overrides Property Name As String
```

C# 構文

```
public override string Name {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

文字列としてのテーブル名。

PrimaryKey プロパティ

テーブルのプライマリキーのインデックススキーマを返します。

Visual Basic 構文

```
Public ReadOnly Property PrimaryKey As ULIndexSchema
```

C# 構文

```
public ULIndexSchema PrimaryKey {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました。

備考

テーブルのプライマリキーを表す ULIndexSchema オブジェクト。

参照

- ULIndexSchema クラス [Ultra Light.NET]301 ページ

UploadUnchangedRows プロパティ

データベースが、変更されていないローをアップロードするかどうかをチェックします。

Visual Basic 構文

```
Public ReadOnly Property UploadUnchangedRows As Boolean
```

C# 構文

```
public bool UploadUnchangedRows {get;}
```

例外

- **ULException クラス**　SQL エラーが発生しました。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

テーブルが、同期時に常にすべてのローをアップロードするようにマーク付けされている場合は true、変更されたローのみをアップロードするようにマーク付けされている場合は false。

すべてのローをアップロードするようにマーク付けされているテーブルでは、その同期時に、変更されたローと未変更のローがアップロードされます。このようなテーブルは、「完全同期」テーブルと呼ばれることもあります。

ULTransaction クラス

SQL トランザクションを表します。

Visual Basic 構文

```
Public NotInheritable Class ULTransaction
    Inherits System.Data.Common.DbTransaction
```

C# 構文

```
public sealed class ULTransaction : System.Data.Common.DbTransaction
```

基本クラス

- System.Data.Common.DbTransaction

メンバー

継承されたメンバーを含む ULTransaction クラスのすべてのメンバー。

名前	説明
Commit メソッド	データベーストランザクションをコミットします。
Dispose method (System.Data.Common.DbTransaction から継承)	System.Data.Common.DbTransaction により使用されるアンマネージリソースを解放します。
Rollback メソッド	データベースへのトランザクションの未処理の変更をロールバックします。
Connection プロパティ	トランザクションに対応する接続を返します。
IsolationLevel プロパティ	トランザクションの独立性レベルを返します。

備考

ULTransaction にはコンストラクターがありません。ULTransaction オブジェクトを取得するには、ULConnection.BeginTransaction() を使用します。コマンドをトランザクションに関連付けるには、ULCommand.Transaction を使用します。

トランザクションがコミットまたはロールバックされると、接続はすべての操作の実行時にそれらを自動的にコミットする状態に戻ります。さらに操作をグループ化するには、新しいトランザクションを作成する必要があります。

参照

- ULConnection.BeginTransaction メソッド [Ultra Light.NET]124 ページ
- ULCommand.Transaction プロパティ [Ultra Light.NET]102 ページ
- System.Data.Common.DbTransaction
- System.Data.IDbTransaction
- System.IDisposable

Commit メソッド

データベーストランザクションをコミットします。

Visual Basic 構文

```
Public Overrides Sub Commit()
```

C# 構文

```
public override void Commit()
```

備考

トランザクションがコミットまたはロールバックされると、接続はすべての操作の実行時にそれらを自動的にコミットする状態に戻ります。さらに操作をグループ化するには、新しいトランザクションを作成する必要があります。

データベースエラーが原因で Commit() が失敗すると (たとえば、参照整合性エラーなど)、トランザクションはアクティブなまま残ります。問題を訂正し、もう一度 Commit() メソッドを呼び出します。または、ULTransaction.Rollback() を呼び出してトランザクションを完了させます。

参照

- ULTransaction.Rollback メソッド [Ultra Light.NET]452 ページ

Rollback メソッド

データベースへのトランザクションの未処理の変更をロールバックします。

Visual Basic 構文

```
Public Overrides Sub Rollback()
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

C# 構文

```
public override void Rollback()
```

備考

トランザクションがコミットまたはロールバックされると、接続はすべての操作の実行時にそれらを自動的にコミットする状態に戻ります。さらに操作をグループ化するには、新しいトランザクションを作成する必要があります。

参照

- ULTransaction.Commit メソッド [Ultra Light.NET]452 ページ

Connection プロパティ

トランザクションに対応する接続を返します。

Visual Basic 構文

```
Public ReadOnly Shadows Property Connection As ULConnection
```

C# 構文

```
public new ULConnection Connection {get;}
```

備考

トランザクションに関連付けられている ULConnection オブジェクト。トランザクションが無効な場合は NULL 参照 (Visual Basic の Nothing)。

これは、System.Data.IDbTransaction.Connection と System.Data.Common.DbCommand.Connection が厳密に型指定されたものです。

参照

- ULConnection.BeginTransaction メソッド [Ultra Light.NET]124 ページ
- ULConnection クラス [Ultra Light.NET]116 ページ
- System.Data.IDbTransaction.Connection
- System.Data.Common.DbCommand.Connection

IsolationLevel プロパティ

トランザクションの独立性レベルを返します。

Visual Basic 構文

```
Public ReadOnly Overrides Property IsolationLevel As IsolationLevel
```

C# 構文

```
public override IsolationLevel IsolationLevel {get;}
```

備考

System.Data.IsolationLevel 値の 1 つ。Ultra Light.NET では、System.Data.IsolationLevel.ReadUncommitted のみがサポートされています。

参照

- ULConnection.BeginTransaction メソッド [Ultra Light.NET]124 ページ
- System.Data.IsolationLevel
- System.Data.IsolationLevel.ReadUncommitted

ULInfoMessageEventHandler デリゲート

ULConnection.InfoMessage イベントを処理するメソッドを示します。

Visual Basic 構文

```
Public Delegate Sub ULInfoMessageEventHandler(
     ByVal obj As Object,
    ByVal args As ULInfoMessageEventArgs
)
```

C# 構文

```
public delegate void ULInfoMessageEventHandler(
     object obj,
    ULInfoMessageEventArgs args
);
```

パラメーター

- **obj** イベントを送信する接続。
- **args** イベントデータが含まれる ULInfoMessageEventArgs オブジェクト。

参照

- ULConnection.InfoMessage event [Ultra Light.NET]161 ページ
- ULInfoMessageEventArgs クラス [Ultra Light.NET]308 ページ

ULRowUpdatedEventHandler デリゲート

ULDataAdapter.RowUpdated イベントを処理するメソッドを示します。

Visual Basic 構文

```
Public Delegate Sub ULRowUpdatedEventHandler(
     ByVal sender As Object,
    ByVal e As ULRowUpdatedEventArgs
)
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

C# 構文

```
public delegate void ULRowUpdatedEventHandler(
     object sender,
    ULRowUpdatedEventArgs e
);
```

パラメーター

- **sender** イベントを送信する接続。
- **e** イベントデータが含まれる ULRowUpdatedEventArgs オブジェクト。

参照

- ULDataAdapter.RowUpdated event [Ultra Light.NET]219 ページ
- ULRowUpdatedEventArgs クラス [Ultra Light.NET]380 ページ

ULRowUpdatingEventHandler デリゲート

ULDataAdapter.RowUpdating イベントを処理するメソッドを示します。

Visual Basic 構文

```
Public Delegate Sub ULRowUpdatingEventHandler(
     ByVal sender As Object,
    ByVal e As ULRowUpdatingEventArgs
)
```

C# 構文

```
public delegate void ULRowUpdatingEventHandler(
     object sender,
    ULRowUpdatingEventArgs e
);
```

パラメーター

- **sender** イベントを送信する接続。
- **e** イベントデータが含まれる ULRowUpdatingEventArgs オブジェクト。

参照

- ULDataAdapter.RowUpdating event [Ultra Light.NET]220 ページ
- ULRowUpdatingEventArgs クラス [Ultra Light.NET]383 ページ

ULRowsCopiedEventHandler デリゲート

ULBulkCopy.ULRowsCopied イベントを処理するメソッドを示します。

Visual Basic 構文

```
Public Delegate Sub ULRowsCopiedEventHandler(
     ByVal sender As Object,
    ByVal rowsCopiedEventArgs As ULRowsCopiedEventArgs
)
```

C# 構文

```
public delegate void ULRowsCopiedEventHandler(
     object sender,
    ULRowsCopiedEventArgs rowsCopiedEventArgs
);
```

備考

ULRowsCopiedEventHandler デリゲートは、.NET Compact Framework 2.0 では使用できません。

参照

- ULBulkCopy.ULRowsCopied event [Ultra Light.NET]50 ページ
- System.Object

ULSyncProgressedDlg デリゲート

同期進捗情報と同期中に起動されたメソッドを表します。

Visual Basic 構文

```
Public Delegate Sub ULSyncProgressedDlg(
     ByVal result As IAsyncResult,
    ByVal data As ULSyncProgressData
)
```

C# 構文

```
public delegate void ULSyncProgressedDlg(
     IAsyncResult result,
    ULSyncProgressData data
);
```

パラメーター

- **result**　BeginSynchronize メソッドから返された IAsyncResult オブジェクト。result.AsyncState を使用して BeginSynchronize メソッドに提供されるオブジェクトにアクセスします。
- **data**　最新の同期のプログレスデータを保持している ULSyncProgressData オブジェクト。

備考

GUI 作業を行うか、この方法の中で Ultra Light.NET API 呼び出しを作成することが安全です。同期は、このメソッドに対する呼び出し中には保持されません。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

参照

- ULConnection.BeginSynchronize メソッド [Ultra Light.NET]122 ページ
- ULSyncProgressData クラス [Ultra Light.NET]402 ページ

ULAuthStatusCode 列挙体

UL 拡張： Mobile Link ユーザー認証の実行中にレポートされる可能性のあるステータスコードを列挙します。

Visual Basic 構文

```
Public Enum ULAuthStatusCode
```

C# 構文

```
public enum ULAuthStatusCode
```

メンバー

メンバー名	説明	値
UNKNOWN	認証ステータスが不明です。接続がまだ同期を実行していない可能性があります (UNKNOWN = 0)。	0
VALID	ユーザー ID とパスワードは、同期時には有効でした (VALID = 1)。	1
VALID_BUT_EXPIRES_SOON	ユーザー ID とパスワードは、同期時には有効でしたが、まもなく有効期限が切れます (VALID_BUT_EXPIRES_SOON = 2)。	2
EXPIRED	ユーザー ID またはパスワードの有効期限が切れているため、認証に失敗しました (EXPIRED = 3)。	3
INVALID	ユーザー ID またはパスワードが正しくないため、認証に失敗しました (INVALID = 4)。	4
IN_USE	ユーザー ID がすでに使用されているため、認証に失敗しました (IN_USE = 5)。	5

参照

- ULSyncResult.AuthStatus プロパティ [Ultra Light.NET]411 ページ

ULBulkCopyOptions 列挙体

ULBulkCopy クラスのインスタンスで使用する、1 つ以上のオプションを指定するビット単位フラグです。

Visual Basic 構文

```
Public Enum ULBulkCopyOptions
```

C# 構文

```
public enum ULBulkCopyOptions
```

メンバー

メンバー名	説明	値
Default	これだけを指定すると、デフォルトの動作が使用されます。	0x0
KeepIdentity	これを指定すると、IDENTITY カラムにコピーされる元の値が保持されます。 デフォルトでは、送信先テーブルでは新しい ID 番号が生成されます。	0x1
UseInternalTransaction	これを指定すると、バルクコピーオペレーションの各バッチがトランザクションの中で実行されます。 これが指定されない場合、トランザクションは使用されません。このオプションを指定し、コンストラクターに ULTransaction オブジェクトも指定すると、System.ArgumentException が発生します。	0x2

備考

ULBulkCopyOptions クラスは、.NET Compact Framework 2.0 では使用できません。

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

ULBulkCopyOptions 列挙体は、ULBulkCopy インスタンスの構築時に使用され、WriteToServer メソッドの動作方法を指定します。

参照

- ULBulkCopy クラス [Ultra Light.NET]40 ページ

ULDBValid 列挙体

Ultra Light.NET データベース検証メソッドを列挙します。

Visual Basic 構文

```
Public Enum ULDBValid
```

C# 構文

```
public enum ULDBValid
```

メンバー

メンバー名	説明	値
EXPRESS_VALIDATE	完全度は低いが、より高速な検証を実行します。	0
FULL_VALIDATE	テーブル、インデックス、すべてのデータベースページを検証します。	1

参照

- ULDatabaseManager.ValidateDatabase メソッド [Ultra Light.NET]226 ページ
- ULConnection.ValidateDatabase メソッド [Ultra Light.NET]153 ページ

ULDateOrder 列挙体

UL 拡張： データベースがサポートできる日付順を列挙します。

Visual Basic 構文

```
Public Enum ULDateOrder
```

C# 構文

```
public enum ULDateOrder
```

メンバー

メンバー名	説明
YMD	年、月、日の順に示された日付。
MDY	月、日、年の順に示された日付。
DMY	日、月、年の順に示された日付。

ULDbType 列挙体

Ultra Light.NET データベースのデータ型を列挙します。

Visual Basic 構文

```
Public Enum ULDbType
```

C# 構文

```
public enum ULDbType
```

メンバー

メンバー名	説明	値
BigInt	符号付き 64 ビット整数値	5
Binary	指定された最大長のバイナリデータ。 列挙値 **Binary** と **VarBinary** は互いのエイリアスです。	15
Bit	1 ビットフラグ	8
Char	指定された長さの文字データ。 Ultra Light.NET では、この型は常に Unicode 文字をサポートします。**Char** 型と **VarChar** 型は、完全に互換性があります。	0
Date	日付情報	10

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

メンバー名	説明	値
DateTime	タイムスタンプ情報 (日付、時刻)。 列挙値 **DateTime** と **TimeStamp** は互いのエイリアスです。	9
Decimal	精度と桁数が指定された正確な数値データ。 列挙値 **Decimal** と **Numeric** は互いのエイリアスです。	14
Double	倍精度浮動小数点数 (8 バイト)	12
Float	単精度浮動小数点数 (4 バイト)。 列挙値 **Float** と **Real** は互いのエイリアスです。	13
Integer	符号なし 32 ビット整数値	1
LongBinary	可変長のバイナリデータ	18
LongVarchar	可変長の文字データ。 Ultra Light.NET では、この型は常に Unicode 文字をサポートします。	17
Numeric	精度と桁数が指定された正確な数値データ。 列挙値 **Decimal** と **Numeric** は互いのエイリアスです。	14
Real	単精度浮動小数点数 (4 バイト)。 列挙値 **Float** と **Real** は互いのエイリアスです。	13
SmallInt	符号付き 16 ビット整数値	3

メンバー名	説明	値
STGeometry	ST ジオメトリ情報	ULNET_TYPE_ST_GEOMETRY
Time	時刻情報	11
TimeStamp	タイムスタンプ情報 (日付、時刻)。 列挙値 **DateTime** と **TimeStamp** は互いのエイリアスです。	9
TimeStampWithTimeZone	タイムゾーンオフセット付きのタイムスタンプ情報 (日付、時刻)	ULNET_TYPE_TIMESTAMP_WITH_TIME_ZONE
TinyInt	符号なし 8 ビット整数値	7
UniqueIdentifier	ユニバーサルユニーク識別子 (UUID/GUID)	19
UnsignedBigInt	符号なし 64 ビット整数値	6
UnsignedInt	符号なし 32 ビット整数値	2
UnsignedSmallInt	符号なし 16 ビット整数値	4
VarBinary	指定された最大長のバイナリデータ。 列挙値 **Binary** と **VarBinary** は互いのエイリアスです。	15
VarChar	指定された最大長の文字データ。 Ultra Light.NET では、この型は常に Unicode 文字をサポートします。**Char** 型と **VarChar** 型は、完全に互換性があります。	16

備考

下の表には、各 ULDbType との互換性がある .NET 型がリストされています。整数型の場合、テーブルのカラムは、常により小さい整数型を使用して設定できるほか、実際の値がその型の範囲内にあるかぎり、より大きい型を使用して設定することも可能です。

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

ULDbType	互換性のある .NET 型	C# 組み込みタイプ	Visual Basic 組み込みタイプ
Binary, **VarBinary**	System.Byte[]、または System.Guid (サイズが 16 の場合)	byte[]	Byte()
Bit	System.Boolean	bool	Boolean
Char, **VarChar**	System.String	String	String
Date	System.DateTime	DateTime (組み込みタイプなし)	Date
Double	System.Double	double	Double
LongBinary	System.Byte[]	byte[]	Byte()
LongVarchar	System.String	String	String
Decimal, **Numeric**	System.String	decimal	Decimal
Float, **Real**	System.Single	float	Single
BigInt	System.Int64	long	Long
Integer	System.Int32	int	Integer
SmallInt	System.Int16	short	Short
STGeometry	System.String	String	String
Time	System.TimeSpan	TimeSpan (組み込みタイプなし)	TimeSpan (組み込みタイプなし)
DateTime, **TimeStamp**	System.DateTime	DateTime (組み込みタイプなし)	Date
TimeStampWithTimeZone	System.String	String	String
TinyInt	System.Byte	byte	Byte
UnsignedBigInt	System.UInt64	ulong	UInt64 (組み込みタイプなし)
UnsignedInt	System.UInt32	uint	UInt32 (組み込みタイプなし)

ULDbType	互換性のある .NET 型	C# 組み込みタイプ	Visual Basic 組み込みタイプ
UnsignedSmallInt	System.UInt16	ushort	UInt16 (組み込みタイプなし)
UniqueIdentifier	System.Guid	Guid (組み込みタイプなし)	Guid (組み込みタイプなし)

長さが 16 のバイナリカラムには、UniqueIdentifier 型との完全な互換性があります。

参照

- ULDataReader.GetFieldType メソッド [Ultra Light.NET]250 ページ
- ULDataReader.GetDataTypeName メソッド [Ultra Light.NET]247 ページ
- ULCursorSchema.GetColumnULDbType メソッド [Ultra Light.NET]207 ページ
- 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 列挙体

UL 拡張： Ultra Light.NET ランタイムのタイプを列挙します。

Visual Basic 構文

```
Public Enum ULRuntimeType
```

C# 構文

```
public enum ULRuntimeType
```

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

メンバー

メンバー名	説明	値
STANDALONE_UL	スタンドアロンの Ultra Light.NET ランタイムを選択します。 スタンドアロンのランタイムは、データベースに直接アクセスします。データベースへのアクセスは高速になりますが、データベースを共有することはできません。	0
UL_ENGINE_CLIENT	Ultra Light エンジンのランタイムを選択します。 Ultra Light.NET エンジンクライアントは、Ultra Light エンジンと通信してデータベースにアクセスします。つまり、別々のアプリケーションでデータベースを共有できます。	1

参照

- ULDatabaseManager.RuntimeType プロパティ [Ultra Light.NET]227 ページ

ULSqlProgressState 列挙体

UL 拡張： SQL パススルースクリプトの実行中に発生する可能性のあるすべてのステータスを列挙します。

Visual Basic 構文

```
Public Enum ULSqlProgressState
```

C# 構文

```
public enum ULSqlProgressState
```

メンバー

メンバー名	説明	値
STATE_STARTING	実行されたスクリプトはまだありません。	0

メンバー名	説明	値
STATE_RUNNING_SCRIPT	SQL パススルースクリプトを現在実行中です。	1
STATE_DONE	スクリプトは正常に完了しました。	2
STATE_ERROR	スクリプトは完了しましたが、エラーが発生しました。	3

参照

- ULSqlProgressData クラス [Ultra Light.NET]388 ページ

ULStreamType 列挙体

UL 拡張： 同期に使用する Mobile Link 同期ストリームのタイプを列挙します。

Visual Basic 構文

```
Public Enum ULStreamType
```

C# 構文

```
public enum ULStreamType
```

メンバー

メンバー名	説明
TCPIP	TCP/IP を介して同期します。
HTTP	HTTP を介して同期します。 HTTP ストリームは基本となるトランスポートとして TCP/IP を使用します。Ultra Light アプリケーションは Web ブラウザとして機能し、Mobile Link サーバーは Web サーバーとして機能します。Ultra Light アプリケーションは、サーバーへのデータ送信のために POST 要求を送り、サーバーからのデータの読み込みのために GET 要求を送ります。
HTTPS	HTTPS を介して同期します (トランスポートレイヤーセキュリティを適用した HTTP)。
TLS	TLS (TCP/IP を介して同期)

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

備考

注意
別途ライセンスが必要な必須コンポーネント

ECC 暗号化と FIPS 認定の暗号化には、別途ライセンスが必要です。強力な暗号化テクノロジはすべて、輸出規制対象品目です。

「別途ライセンスが必要なコンポーネント」『SQL Anywhere 12 紹介』を参照してください。

参照

- ULSyncParms.Stream プロパティ [Ultra Light.NET]399 ページ
- 「Ultra Light 同期ストリームのネットワークプロトコルのオプション」『Ultra Light データベース管理とリファレンス』

ULSyncProgressState 列挙体

UL 拡張： 同期中に発生する可能性のあるすべてのステータスを列挙します。

Visual Basic 構文

```
Public Enum ULSyncProgressState
```

C# 構文

```
public enum ULSyncProgressState
```

メンバー

メンバー名	説明	値
STATE_STARTING	同期処理はまだ行われていません。	0
STATE_CONNECTING	同期ストリームは構築されていますが、まだ開かれていません。	1
STATE_SENDING_HEADER	同期ストリームが開かれ、ヘッダーが送信されようとしています。	2

メンバー名	説明	値
STATE_SENDING_TABLE	テーブルが送信されています。 進行状況をモニターするには、ULSyncProgressData.SyncTableIndex と ULSyncProgressData.SyncTableCount を使用します。	3
STATE_SENDING_DATA	現在のテーブルのデータが送信されています。 ULSyncProgressData.SentBytes、ULSyncProgressData.SentInserts、ULSyncProgressData.SentUpdates、および ULSyncProgressData.SentDeletes は更新されました。	4
STATE_FINISHING_UPLOAD	アップロードの完了処理中です。 送信された最終的なロー数が、このイベントに含まれます。	5
STATE_RECEIVING_UPLOAD_ACK	アップロード完了の確認を受信しています。	6
STATE_RECEIVING_TABLE	テーブルを受信しています。 進行状況をモニターするには、ULSyncProgressData.SyncTableIndex と ULSyncProgressData.SyncTableCount を使用します。	7

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

メンバー名	説明	値
STATE_RECEIVING_DATA	現在のテーブルのデータを受信しています。 ULSyncProgressData.ReceivedBytes、ULSyncProgressData.ReceivedInserts、ULSyncProgressData.ReceivedUpdates、および ULSyncProgressData.ReceivedDeletes は更新されました。	8
STATE_COMMITTING_DOWNLOAD	ダウンロードをコミットしています。 受信された最終的なロー数が、このイベントに含まれます。	9
STATE_ROLLING_BACK_DOWNLOAD	ダウンロード中にエラーが発生したため、同期によってダウンロードがロールバックされています。 後続の STATE_ERROR 進行状況レポートにエラーがレポートされます。	10
STATE_SENDING_DOWNLOAD_ACK	ダウンロード完了の確認が送信されています。	11
STATE_DISCONNECTING	同期ストリームが閉じられようとしています。	12
STATE_DONE	同期は正常に完了しました。	13
STATE_ERROR	同期は完了しましたが、エラーが発生しました。	14
STATE_CANCELLED	同期がキャンセルされました。	15

参照

- ULSyncProgressData クラス [Ultra Light.NET]402 ページ
- ULSyncProgressData.ReceivedBytes プロパティ [Ultra Light.NET]404 ページ
- ULSyncProgressData.ReceivedInserts プロパティ [Ultra Light.NET]405 ページ
- ULSyncProgressData.ReceivedUpdates プロパティ [Ultra Light.NET]405 ページ
- ULSyncProgressData.ReceivedDeletes プロパティ [Ultra Light.NET]404 ページ
- ULSyncProgressData.SentBytes プロパティ [Ultra Light.NET]405 ページ
- ULSyncProgressData.SentInserts プロパティ [Ultra Light.NET]406 ページ
- ULSyncProgressData.SentUpdates プロパティ [Ultra Light.NET]407 ページ
- ULSyncProgressData.SentDeletes プロパティ [Ultra Light.NET]406 ページ
- ULSyncProgressData.SyncTableIndex プロパティ [Ultra Light.NET]408 ページ
- ULSyncProgressData.SyncTableCount プロパティ [Ultra Light.NET]407 ページ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

索引

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

D

E

F

G

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

H

I

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

K

L

M

N

O

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

P

R

S

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

T

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

U

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

V

W

あ

い

え

お

か

き

け

こ

 Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1

ま

も

る

ろ

Copyright © 2012, iAnywhere Solutions, Inc. - SQL Anywhere 12.0.1