SYBASE iAnywhere

Ultra Light
.NET プログラミング

改訂 2007 年 3 月

版権と商標

Copyright (c) 2007 iAnywhere Solutions, Inc. Portions copyright (c) 2007 Sybase, Inc. All rights reserved.

iAnywhere Solutions, Inc. は Sybase, Inc. の関連会社です。

iAnywhere は、(1) すべてのコピーにこの情報またはマニュアル内のその他の版権と商標の表示を含める、(2) マニュアルの偽装表示をしない、(3) マニュアルに変更を加えないことが遵守されるかぎり、このマニュアルをご自身の情報収集、教育、その他の非営利の目的で使用することを許可します。このマニュアルまたはその一部を、iAnywhere の書面による事前の許可なく発行または配布することは禁じられています。

このマニュアルは、iAnywhere が何らかの行動を行う、または行わない責任を表明するものではありません。このマニュアルは、iAnywhere の判断で予告なく内容が変更される場合があります。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 iAnywhere は一切の責任を負いません。

iAnywhere (R)、Sybase (R)、http://www.ianywhere.com/trademarks に示す商標は Sybase, Inc. またはその関連会社の商標です。(R) は米国での登録商標を示します。

Java および Java 関連のすべての商標は、米国またはその他の国での Sun Microsystems, Inc. の商標または登録商標です。

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

はじめに ix

SQL Anywhere のマニュアル x
表記の規則 xiii
詳細情報の検索／フィードバックの提供 xvii

Ultra Light.NET の概要 1

Ultra Light と .NET 2
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 4
Ultra Light.NET アーキテクチャ 5

Ultra Light 用 SQL Anywhere Explorer 7

SQL Anywhere Explorer の概要 8
Ultra Light プロジェクトでの SQL Anywhere Explorer の使用 9

Ultra Light.NET 開発の概要 13

Visual Studio .NET での SQL Anywhere ツールの使用 14
データベースへの接続 15
暗号化と難読化 17
SQL を使用したデータへのアクセスと操作 18
テーブル API によるデータ・アクセスと操作 22
トランザクションの管理 28
スキーマ情報へのアクセス 29
エラー処理 30
ユーザの認証 31
Ultra Light アプリケーションの同期 32

チュートリアル：Ultra Light.NET アプリケーションの構築 35

Ultra Light.NET 開発のチュートリアル 36
レッスン 1：Visual Studio プロジェクトの作成 37

Copyright © 2007, iAnywhere Solutions, Inc.

レッスン 2：Ultra Light データベースの作成 40
レッスン 3：データベースへの接続 41
レッスン 4：データの挿入、更新、削除 43
レッスン 5：アプリケーションのビルドと配置 48

Ultra Light .NET 1.0 API リファレンス 51

ULActiveSyncListener インタフェース 53
ULAuthStatusCode 列挙 56
ULCommand クラス 57
ULCommandBuilder クラス 82
ULConnection クラス 90
ULConnectionParms クラス 123
ULConnectionParms.UnusedEventHandler 委任 136
ULCreateParms クラス 137
ULCursorSchema クラス 147
ULDataAdapter クラス 156
ULDatabaseManager クラス 182
ULDatabaseSchema クラス 190
ULDataReader クラス 201
ULDateOrder enumeration 244
ULDbType 列挙 245
ULException クラス 249
ULFileTransfer クラス 253
ULFileTransferProgressData クラス 268
ULFileTransferProgressListener インタフェース 271
ULIndexSchema クラス 273
ULInfoMessageEventArgs クラス 282
ULInfoMessageEventHandler 委任 285
ULParameter クラス 286
ULParameterCollection クラス 303
ULPublicationSchema クラス 321
ULResultSet クラス 325
ULResultSetSchema クラス 351
ULRowUpdatedEventArgs クラス 353

 Copyright © 2007, iAnywhere Solutions, Inc.

ULRowUpdatedEventHandler 委任 **357**
ULRowUpdatingEventArgs クラス **358**
ULRowUpdatingEventHandler 委任 **361**
ULRuntimeType 列挙 **362**
ULServerSyncListener インタフェース **363**
ULSQLCode 列挙 **366**
ULStreamErrorCode 列挙 **376**
ULStreamErrorContext 列挙 **392**
ULStreamErrorID 列挙 **393**
ULStreamType 列挙 **395**
ULSyncParms クラス **396**
ULSyncProgressData クラス **411**
ULSyncProgressListener インタフェース **421**
ULSyncProgressState 列挙 **423**
ULSyncResult クラス **425**
ULTable クラス **431**
ULTableSchema クラス **453**
ULTransaction クラス **468**

Ultra Light .NET 2.0 API リファレンス 473

ULActiveSyncListener インタフェース **475**
ULAuthStatusCode 列挙 **479**
ULBulkCopy クラス **480**
ULBulkCopyColumnMapping クラス **492**
ULBulkCopyColumnMappingCollection クラス **499**
ULBulkCopyOptions 列挙 **508**
ULCommand クラス **509**
ULCommandBuilder クラス **545**
ULConnection クラス **555**
ULConnectionParms クラス **592**
ULConnectionParms.UnusedEventHandler 委任 **605**
ULConnectionStringBuilder クラス **606**
ULCreateParms クラス **624**
ULCursorSchema クラス **636**

Copyright © 2007, iAnywhere Solutions, Inc.

ULDataAdapter クラス **645**
ULDatabaseManager クラス **657**
ULDatabaseSchema クラス **665**
ULDataReader クラス **676**
ULDateOrder enumeration **714**
ULDbType 列挙 **715**
ULException クラス **719**
ULFactory クラス **723**
ULFileTransfer クラス **729**
ULFileTransferProgressData クラス **745**
ULFileTransferProgressListener インタフェース **749**
ULIndexSchema クラス **751**
ULInfoMessageEventArgs クラス **760**
ULInfoMessageEventHandler 委任 **763**
ULMetaDataCollectionNames クラス **764**
ULParameter クラス **773**
ULParameterCollection クラス **790**
ULPublicationSchema クラス **810**
ULResultSet クラス **814**
ULResultSetSchema クラス **839**
ULRowsCopiedEventArgs クラス **842**
ULRowsCopiedEventHandler 委任 **845**
ULRowUpdatedEventArgs クラス **846**
ULRowUpdatedEventHandler 委任 **850**
ULRowUpdatingEventArgs クラス **851**
ULRowUpdatingEventHandler 委任 **854**
ULRuntimeType 列挙 **855**
ULServerSyncListener インタフェース **856**
ULSQLCode 列挙 **859**
ULStreamErrorCode 列挙 **871**
ULStreamErrorContext 列挙 **892**
ULStreamErrorID 列挙 **893**
ULStreamType 列挙 **895**
ULSyncParms クラス **896**
ULSyncProgressData クラス **912**

 Copyright © 2007, iAnywhere Solutions, Inc.

ULSyncProgressListener インタフェース 923
ULSyncProgressState 列挙 925
ULSyncResult クラス 927
ULTable クラス 933
ULTableSchema クラス 954
ULTransaction クラス 969
索引 973

Copyright © 2007, iAnywhere Solutions, Inc.

はじめに

このマニュアルの内容

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、モバイル、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。

対象読者

このマニュアルは、Ultra Light リレーショナル・データベースのパフォーマンス、リソース効率、堅牢性、セキュリティを利用してデータを格納、同期することを目的とする .NET アプリケーション開発者を対象にしています。

Copyright © 2007, iAnywhere Solutions, Inc.

SQL Anywhere のマニュアル

このマニュアルは、SQL Anywhere のマニュアル・セットの一部です。この項では、マニュアル・セットに含まれる各マニュアルと使用法について説明します。

SQL Anywhere のマニュアル

SQL Anywhere の完全なマニュアルは、各マニュアルをまとめたオンライン形式とマニュアル別の PDF ファイルで提供されます。いずれの形式のマニュアルも、同じ情報が含まれ、次のマニュアルから構成されます。

- **『SQL Anywhere 10 - 紹介』** このマニュアルでは、データの管理および交換機能を提供する包括的なパッケージである SQL Anywhere 10 について説明します。SQL Anywhere を使用すると、サーバ環境、デスクトップ環境、モバイル環境、リモート・オフィス環境に適したデータベース・ベースのアプリケーションを迅速に開発できるようになります。
- **『SQL Anywhere 10 - 変更点とアップグレード』** このマニュアルでは、SQL Anywhere 10 とそれ以前のバージョンに含まれる新機能について説明します。
- **『SQL Anywhere サーバ - データベース管理』** このマニュアルでは、SQL Anywhere データベースの実行、管理、設定について説明します。管理ユーティリティとオプションのほか、データベース接続、データベース・サーバ、データベース・ファイル、バックアップ・プロシージャ、セキュリティ、高可用性、Replication Server を使用したレプリケーションについて説明します。
- **『SQL Anywhere サーバ - SQL の使用法』** このマニュアルでは、データベースの設計と作成の方法、データのインポート・エクスポート・変更の方法、データの検索方法、ストアド・プロシージャとトリガの構築方法について説明します。
- **『SQL Anywhere サーバ - SQL リファレンス』** このマニュアルは、SQL Anywhere で使用する SQL 言語の完全なリファレンスです。また、SQL Anywhere のシステム・ビューとシステム・プロシージャについても説明しています。
- **『SQL Anywhere サーバ - プログラミング』** このマニュアルでは、C、C++、Java プログラミング言語、Visual Studio .NET を使用してデータベース・アプリケーションを構築、配備する方法について説明します。Visual Basic や PowerBuilder などのツールのユーザは、それらのツールのプログラミング・インタフェースを使用できます。
- **『SQL Anywhere 10 - エラー・メッセージ』** このマニュアルでは、SQL Anywhere エラー・メッセージの完全なリストを、その診断情報とともに説明します。
- **『Mobile Link - クイック・スタート』** このマニュアルでは、セッションベースのリレーショナル・データベース同期システムである Mobile Link について説明します。Mobile Link テクノロジは、双方向レプリケーションを可能にし、モバイル・コンピューティング環境に非常に適しています。
- **『Mobile Link - サーバ管理』** このマニュアルでは、Mobile Link アプリケーションを設定して管理する方法について説明します。

Copyright © 2007, iAnywhere Solutions, Inc.

- **『Mobile Link - クライアント管理』** このマニュアルでは、Mobile Link クライアントを設定、構成、同期する方法について説明します。Mobile Link クライアントには、SQL Anywhere または Ultra Light のいずれかのデータベースを使用できます。
- **『Mobile Link - サーバ起動同期』** このマニュアルでは、Mobile Link のサーバによって開始される同期について説明します。サーバによって開始される同期とは、統合データベースから同期またはその他のリモート・アクションの開始を可能にする Mobile Link の機能です。
- **『QAnywhere』** このマニュアルでは QAnywhere について説明します。QAnywhere は、従来のデスクトップ・クライアントやラップトップ・クライアント用のメッセージング・プラットフォームであるほか、モバイル・クライアントや無線クライアント用のメッセージング・プラットフォームでもあります。
- **『SQL Remote』** このマニュアルでは、モバイル・コンピューティング用の SQL Remote データ・レプリケーション・システムについて説明します。このシステムによって、SQL Anywhere の統合データベースと複数の SQL Anywhere リモート・データベースの間で、電子メールやファイル転送などの間接的リンクを使用したデータ共有が可能になります。
- **『SQL Anywhere 10 - コンテキスト別ヘルプ』** このマニュアルには、[接続] ダイアログ、クエリ・エディタ、Mobile Link モニタ、SQL Anywhere コンソール・ユーティリティ、インデックス・コンサルタント、Interactive SQL のコンテキスト別のヘルプが収録されています。
- **『Ultra Light - データベース管理とリファレンス』** このマニュアルでは、小型デバイス用 Ultra Light データベース・システムの概要を説明します。
- **『Ultra Light - AppForge プログラミング』** このマニュアルでは、Ultra Light for AppForge について説明します。Ultra Light for AppForge を使用すると、Palm OS、Symbian OS、または Windows CE を搭載しているハンドヘルド、モバイル、または埋め込みデバイスに対してデータベース・アプリケーションを開発、配備できます。
- **『Ultra Light - .NET プログラミング』** 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、モバイル、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 - M-Business Anywhere プログラミング』** このマニュアルは、Ultra Light for M-Business Anywhere について説明します。Ultra Light for M-Business Anywhere を使用すると、Palm OS、Windows CE、または Windows XP を搭載しているハンドヘルド、モバイル、または埋め込みデバイスに対して Web ベースのデータベース・アプリケーションを開発、配備できます。
- **『Ultra Light - C/C++ プログラミング』** このマニュアルでは、Ultra Light C および Ultra Light C++ のプログラミング・インタフェースについて説明します。Ultra Light を使用すると、ハンドヘルド、モバイル、埋め込みデバイスに対してデータベース・アプリケーションを開発、配備できます。

マニュアルの形式

SQL Anywhere のマニュアルは、次の形式で提供されています。

- **オンライン・マニュアル** オンライン・マニュアルには、SQL Anywhere の完全なマニュアルがあり、SQL Anywhere ツールに関する印刷マニュアルとコンテキスト別のヘルプの両方が含

Copyright © 2007, iAnywhere Solutions, Inc.

まれています。オンライン・マニュアルは、製品のメンテナンス・リリースごとに更新されます。これは、最新の情報を含む最も完全なマニュアルです。

Windows オペレーティング・システムでオンライン・マニュアルにアクセスするには、[スタート] − [プログラム] − [SQL Anywhere 10] − [オンライン・マニュアル] を選択します。オンライン・マニュアルをナビゲートするには、左ウィンドウ枠で HTML ヘルプの目次、索引、検索機能を使用し、右ウィンドウ枠でリンク情報とメニューを使用します。

UNIX オペレーティング・システムでオンライン・マニュアルにアクセスするには、SQL Anywhere のインストール・ディレクトリまたはインストール CD に保存されている HTML マニュアルを参照してください。

- **PDF ファイル** SQL Anywhere の完全なマニュアル・セットは、Adobe Reader で表示できる Adobe Portable Document Format (pdf) 形式のファイルとして提供されています。

  Windows では、PDF 形式のマニュアルはオンライン・マニュアルの各ページ上部にある PDF のリンクから、または Windows の [スタート] メニュー ([スタート] − [プログラム] − [SQL Anywhere 10] − [オンライン・マニュアル - PDF フォーマット]) からアクセスできます。

  UNIX では、PDF 形式のマニュアルはインストール CD にあります。

 Copyright © 2007, iAnywhere Solutions, Inc.

表記の規則

この項では、このマニュアルで使用されている書体およびグラフィック表現の規則について説明します。

SQL 構文の表記規則

SQL 構文の表記には、次の規則が適用されます。

- **キーワード**　SQL キーワードはすべて次の例に示す ALTER TABLE のように大文字で表記します。

  ALTER TABLE [*owner.*]*table-name*

- **プレースホルダ**　適切な識別子または式で置き換えられる項目は、次の例に示す *owner* や *table-name* のように表記します。

  ALTER TABLE [*owner.*]*table-name*

- **繰り返し項目**　繰り返し項目のリストは、次の例に示す *column-constraint* のように、リストの要素の後ろに省略記号 (ピリオド 3 つ …) を付けて表します。

  ADD *column-definition* [*column-constraint*, …]

  複数の要素を指定できます。複数の要素を指定する場合は、各要素間をカンマで区切る必要があります。

- **オプション部分**　文のオプション部分は角カッコで囲みます。

  RELEASE SAVEPOINT [*savepoint-name*]

  この例では、角カッコで囲まれた *savepoint-name* がオプション部分です。角カッコは入力しないでください。

- **オプション**　項目リストから 1 つだけ選択する場合や、何も選択しなくてもよい場合は、項目間を縦線で区切り、リスト全体を角カッコで囲みます。

  [**ASC** | **DESC**]

  この例では、ASC と DESC のどちらか 1 つを選択しても、選択しなくてもかまいません。角カッコは入力しないでください。

- **選択肢**　オプションの中の 1 つを必ず選択しなければならない場合は、選択肢を中カッコで囲み、縦棒で区切ります。

  [**QUOTES** { **ON** | **OFF** }]

  QUOTES オプションを使用する場合は、ON または OFF のどちらかを選択する必要があります。角カッコと中カッコは入力しないでください。

Copyright © 2007, iAnywhere Solutions, Inc.

オペレーティング・システムの表記規則

- **Windows**　デスクトップおよびラップトップ・コンピュータ用の Microsoft Windows オペレーティング・システムのファミリのことです。Windows ファミリには Windows Vista や Windows XP も含まれます。
- **Windows CE**　Microsoft Windows CE モジュラ・オペレーティング・システムに基づいて構築されたプラットフォームです。Windows Mobile や Windows Embedded CE などのプラットフォームが含まれます。

  Windows Mobile は Windows CE 上に構築されています。これにより、Windows のユーザ・インタフェースや、Word や Excel といったアプリケーションの小規模バージョンなどの追加機能が実現されています。Windows Mobile は、モバイル・デバイスで最も広く使用されています。

  SQL Anywhere の制限事項や相違点は、基盤となっているオペレーティング・システム (Windows CE) に由来しており、使用しているプラットフォーム (Windows Mobile など) に依存していることはほとんどありません。
- **UNIX**　特に記述がないかぎり、UNIX は Linux プラットフォームと UNIX プラットフォームの両方のことです。

ファイルの命名規則

マニュアルでは、パス名やファイル名などのオペレーティング・システムに依存するタスクと機能を表すときは、通常 Windows の表記規則が使用されます。ほとんどの場合、他のオペレーティング・システムで使用される構文に簡単に変換できます。

- **ディレクトリ名とパス名**　マニュアルでは、ドライブを示すコロンや、ディレクトリの区切り文字として使用する円記号など、Windows の表記規則を使用して、ディレクトリ・パスのリストを示します。次に例を示します。

  ```
  MobiLink¥redirector
  ```

  UNIX、Linux、Mac OS X では、代わりにスラッシュを使用してください。次に例を示します。

  ```
  MobiLink/redirector
  ```

  SQL Anywhere がマルチプラットフォーム環境で使用されている場合、プラットフォーム間でのパス名の違いに注意する必要があります。
- **実行ファイル**　マニュアルでは、実行ファイルの名前は、Windows の表記規則が使用され、拡張子 *.exe* が付きます。UNIX、Linux、Mac OS X では、実行ファイルの名前には拡張子は付きません。NetWare では、実行ファイルの名前には、拡張子 *.nlm* が付きます。

  たとえば、Windows では、ネットワーク・データベース・サーバは *dbsrv10.exe* です。UNIX、Linux、Mac OS X では、*dbsrv10* になります。NetWare では、*dbsrv10.nlm* になります。
- **install-dir**　インストール・プロセスでは、SQL Anywhere をインストールするロケーションを選択できます。マニュアルでは、このロケーションは *install-dir* という表記で示されます。

Copyright © 2007, iAnywhere Solutions, Inc.

インストールが完了すると、環境変数 SQLANY10 によって SQL Anywhere コンポーネントがあるインストール・ディレクトリのロケーション (*install-dir*) が指定されます。SQLANYSH10 は、SQL Anywhere が他の Sybase アプリケーションと共有しているコンポーネントがあるディレクトリのロケーションを指定します。

オペレーティング・システム別の *install-dir* のデフォルト・ロケーションの詳細については、「SQLANY10 環境変数」『SQL Anywhere サーバ - データベース管理』を参照してください。

- **samples-dir** インストール・プロセスでは、SQL Anywhere に含まれるサンプルをインストールするロケーションを選択できます。マニュアルでは、このロケーションは *samples-dir* という表記で示されます。

  インストールが完了すると、環境変数 SQLANYSAMP10 によってサンプルがあるディレクトリのロケーション (*samples-dir*) が指定されます。Windows の [スタート] メニューから、[プログラム] – [SQL Anywhere 10] – [サンプル・アプリケーションおよびプロジェクト] を選択すると、このディレクトリで [Windows エクスプローラ] ウィンドウが表示されます。

  オペレーティング・システム別の *samples-dir* のデフォルト・ロケーションの詳細については、「サンプル・ディレクトリ」『SQL Anywhere サーバ - データベース管理』を参照してください。

- **環境変数** マニュアルでは、環境変数設定が引用されます。Windows では、環境変数を参照するのに、構文 *%envvar%* が使用されます。UNIX、Linux、Mac OS X では、環境変数を参照するのに、構文 *$envvar* または *${envvar}* が使用されます。

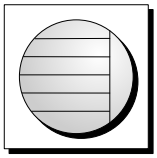
  UNIX、Linux、Mac OS X 環境変数は、*.cshrc* や *.tcshrc* などのシェルとログイン・スタートアップ・ファイルに格納されます。

グラフィック・アイコン

このマニュアルでは、次のアイコンを使用します。

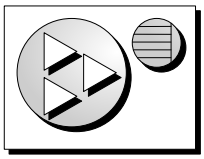
- クライアント・アプリケーション

- SQL Anywhere などのデータベース・サーバ

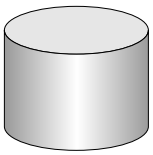


- Ultra Light アプリケーション

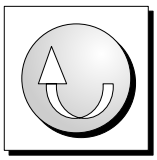
Copyright © 2007, iAnywhere Solutions, Inc.



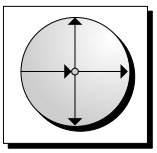
- データベース。高度な図では、データベースとデータベースを管理するデータ・サーバの両方をこのアイコンで表します。



- レプリケーションまたは同期のミドルウェア。ソフトウェアのこれらの部分は、データベース間のデータ共有を支援します。たとえば、Mobile Link サーバ、SQL Remote Message Agent などが挙げられます。



- Sybase Replication Server



- プログラミング・インタフェース

Copyright © 2007, iAnywhere Solutions, Inc.

詳細情報の検索／フィードバックの提供

詳細情報の検索

詳しい情報やリソース (コード交換など) については、iAnywhere Developer Network (http://www.ianywhere.com/developer/) を参照してください。

ご質問がある場合や支援が必要な場合は、次に示す Sybase iAnywhere ニュースグループのいずれかにメッセージをお寄せください。

ニュースグループにメッセージをお送りいただく際には、ご使用の SQL Anywhere バージョンのビルド番号を明記し、現在発生している問題について詳しくお知らせくださいますようお願いいたします。バージョン情報は、コマンド・プロンプトで **dbeng10 -v** と入力して確認できます。

ニュースグループは、ニュース・サーバ *forums.sybase.com* にあります (ニュースグループにおけるサービスは英語でのみの提供となります)。以下のニュースグループがあります。

- sybase.public.sqlanywhere.general
- sybase.public.sqlanywhere.linux
- sybase.public.sqlanywhere.mobilink
- sybase.public.sqlanywhere.product_futures_discussion
- sybase.public.sqlanywhere.replication
- sybase.public.sqlanywhere.ultralite
- ianywhere.public.sqlanywhere.qanywhere

ニュースグループに関するお断り
iAnywhere Solutions は、ニュースグループ上に解決策、情報、または意見を提供する義務を負うものではありません。また、システム・オペレータ以外のスタッフにこのサービスを監視させて、操作状況や可用性を保証する義務もありません。
iAnywhere のテクニカル・アドバイザとその他のスタッフは、時間のある場合にかぎりニュースグループでの支援を行います。こうした支援は基本的にボランティアで行われるため、解決策や情報を定期的に提供できるとはかぎりません。支援できるかどうかは、スタッフの仕事量に左右されます。

フィードバック

このマニュアルに関するご意見、ご提案、フィードバックをお寄せください。

マニュアルに関するご意見、ご提案は、SQL Anywhere ドキュメンテーション・チームの iasdoc@ianywhere.com 宛てに電子メールでお寄せください。このアドレスに送信された電子メールに返信はいたしませんが、お寄せいただいたご意見、ご提案は必ず読ませていただきます。

マニュアルまたはソフトウェアについてのフィードバックは、上記のニュースグループを通してお寄せいただいてもかまいません。

Copyright © 2007, iAnywhere Solutions, Inc.

第 1 章

Ultra Light.NET の概要

目次

Ultra Light と .NET 2
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 4
Ultra Light.NET アーキテクチャ 5

Copyright © 2007, iAnywhere Solutions, Inc.

Ultra Light と .NET

.NET Compact Framework は、Windows CE 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 アプリケーションは、Windows CE および Windows に配備できます。Windows CE 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。

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 インタフェー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 インタフェー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。

iAnywhere.Data.UltraLite ネームスペース (.NET 1.0)

iAnywhere.Data.UltraLite ネームスペースには、ADO.NET 用 Ultra Light.NET データ・プロバイダのクラス、インタフェース、列挙が含まれます。Ultra Light.NET を使用すると、C# または Visual Basic .NET コードを記述して、ADO.NET 標準を使用した Ultra Light データベース・アプリケーションを開発できます。そして、SQL Anywhere へのマイグレーション・パスを提供できます。

Ultra Light と SQL Anywhere のどちらを使用するかについてまだ決定していない場合は、最初は iAnywhere.Data.UltraLite ネームスペースを使用して、より高度な SQL Anywhere 機能が必要だと判断したときに、SQL Anywhere (iAnywhere.Data.SQLAnywhere ネームスペース) に切り替えます。SQL Anywhere に移行する可能性がある場合は、Ultra Light のみに対応した iAnywhere.Data.UltraLite ネームスペース拡張機能の使用は、可能なかぎり避けてください。

SQL Anywhere の ADO.NET 用データ・プロバイダで利用できない 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。

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・アセンブリです。メイン・アセンブリは、このリソース・アセンブリを、CultureInfo.CurrentUICulture、CultureInfo.CurrentCulture、国 "EN" の順序で国別に検索します。

iAnywhere.Data.UltraLite ネームスペース (.NET 2.0)

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

Copyright © 2007, iAnywhere Solutions, Inc.

SQL Anywhere の ADO.NET 用データ・プロバイダで利用できない 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」 658 ページを適切な値に設定してから、他の Ultra Light.NET API を使用します。

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」 555 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・アセンブリです。メイン・アセンブリは、このリソース・アセンブリを、CultureInfo.CurrentUICulture、CultureInfo.CurrentCulture、国 "EN" の順序で国別に検索します。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

- Microsoft Windows
- Visual Studio.NET、Visual Studio.NET 2003、Visual Studio 2005 のいずれか
- Windows CE デバイスでは、.NET Compact Framework バージョン 1.0.5000 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。

- Microsoft .NET Compact Framework 2.0 と .NET Framework 2.0
- Windows CE 3.0 以降で動作する Microsoft .NET Compact Framework 1.0.3705 以降 (ARM プロセッサ上で動作する Pocket PC も含む)。Pocket PC 2002 エミュレータもサポートされています。
- Windows CE デバイスでは、.NET Compact Framework バージョン 1.0.5000 以降
- Windows 上で動作する Microsoft .NET Compact Framework バージョン 1.0.5000 以降。

アプリケーション・コードと .NET Compact Framework に加えて、次のファイルを Windows CE デバイスまたは Windows がインストールされているコンピュータに配備してください。

- **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、アプリケーションで iAnywhere.Data.UltraLite ネームスペースを使用する場合のみ必要です。
- **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、アプリケーションで iAnywhere.Data.UltraLite ネームスペースを使用する場合のみ必要です。
- **ulnet10.dll** このファイルには、Ultra Light ランタイムが含まれます。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。

Ultra Light がサポートしているプラットフォームについては、「SQL Anywhere 用 Ultra Light 展開オプション」を参照してください。

 Copyright © 2007, iAnywhere Solutions, Inc.

Ultra Light.NET アーキテクチャ

Ultra Light.NET ネームスペースの名前は、iAnywhere.Data.UltraLite (ADO.NET インタフェース) 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 サーバと同期させます。

ADO.NET インタフェースの詳細については、「Ultra Light .NET 1.0 API リファレンス」 51 ページを参照してください。

Copyright © 2007, iAnywhere Solutions, Inc.

第 2 章

Ultra Light 用 SQL Anywhere Explorer

目次

SQL Anywhere Explorer の概要 8
Ultra Light プロジェクトでの SQL Anywhere Explorer の使用 9

Copyright © 2007, iAnywhere Solutions, Inc.

SQL Anywhere Explorer の概要

SQL Anywhere Explorer は、Visual Studio .NET から SQL Anywhere と Ultra Light に接続できるコンポーネントです。

SQL Anywhere データベースでの SQL Anywhere Explorer の使用については、「SQL Anywhere Explorer」『SQL Anywhere サーバ - プログラミング』を参照してください。

Copyright © 2007, iAnywhere Solutions, Inc.

Ultra Light プロジェクトでの SQL Anywhere Explorer の使用

Visual Studio .NET 2003 と 2005 では、SQL Anywhere Explorer を使用して Ultra Light データベースへの接続を作成できます。データベースに接続すると、次の操作を実行できます。

- テーブルとテーブル・データを参照する。
- Ultra Light データベースへの接続を開くプログラムやデータの取り出しと操作を行うプログラムを設計する。
- データベース・オブジェクトを C# や Visual Basic のコードやフォームにドラッグ・アンド・ドロップすることで、選択されたオブジェクトを参照するコードを IDE で自動的に生成する。

[ツール] メニューから対応するコマンドを選択することで、Sybase Central と Interactive SQL を Visual Studio .NET から開くこともできます。

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 コンピュータに Visual Studio がインストール済みの場合、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、Ultra Light を Visual Studio と統合するプロセスを次の手順に従って手動で実行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 .NET 2003 の場合は、*install-dir¥ultralite¥UltraLite.NET¥assembly¥v1¥installULNet.exe* を実行します。

  Visual Studio 2005 の場合は、*install-dir¥ultralite¥UltraLite.NET¥assembly¥v2¥installULNet.exe* を実行します。

Visual Studio .NET での Ultra Light データベース接続の使用

SQL Anywhere Explorer を使用して、データ接続ノードで Ultra Light データベース接続を表示します。テーブル内のデータを表示するには、データ接続を作成してください。

SQL Anywhere Explorer でデータベース・テーブルを表示し、個々のテーブルを展開してカラムを表示できます。SQL Anywhere Explorer ウィンドウで選択されたオブジェクトのプロパティは、Visual Studio の [プロパティ] ウィンドウ枠に表示されます。

- **Visual Studio .NET で Ultra Light データベース接続を追加するには、次の手順に従います。**

1. [表示] − [SQL Anywhere Explorer] を選択して、SQL Anywhere Explorer を開きます。

Copyright © 2007, iAnywhere Solutions, Inc.

2. SQL Anywhere Explorer ウィンドウで、[データ接続] を右クリックし、[接続の追加] を選択します。

   [接続の追加] ダイアログが表示されます。

3. [Ultra Light] を選択し、[OK] をクリックします。

   [接続プロパティ] ダイアログが表示されます。

4. 適切な値を入力して、データベースに接続します。

5. [OK] をクリックします。

   データベースへの接続が確立され、接続が [データ接続] リストに追加されます。

♦ Visual Studio .NET から Ultra Light データベース接続を削除するには、次の手順に従います。

1. [表示] − [SQL Anywhere Explorer] を選択して、SQL Anywhere Explorer を開きます。

2. SQL Anywhere Explorer ウィンドウで、削除する Ultra Light データベース接続を右クリックし、[削除] を選択します。

   SQL Anywhere Explorer ウィンドウから接続が削除されます。

SQL Anywhere Explorer の設定

Visual Studio の [オプション] ダイアログには、SQL Anywhere Explorer を設定するのに使用できる設定が含まれています。オプションには、一般的なオプションと Ultra Light でのみ使用する固有のオプションがあります。

♦ SQL Anywhere Explorer オプションを使用するには、次の手順に従います。

1. Visual Studio の [ツール] メニューから [オプション] を選択します。

   [オプション] ダイアログが表示されます。

2. [オプション] ダイアログの左ウィンドウ枠で、[SQL Anywhere] を展開します。

3. [一般] をクリックして、必要に応じて SQL Anywhere Explorer の一般オプションを設定します。

   [出力ウィンドウに送信するクエリ結果を制限する] [出力] ウィンドウに表示するローの数を指定します。デフォルト値は 500 です。

   [オブジェクトのソート] Ultra Light 用 Explorer ウィンドウで、オブジェクト名またはオブジェクトの所有者名順にオブジェクトをソートするときに選択します。

   [テーブルまたはビューをデザイナにドラッグ・アンド・ドロップするときに UI コードを生成する] Windows Forms Designer にドラッグ・アンド・ドロップするテーブルまたはビューのコードを生成します。

 Copyright © 2007, iAnywhere Solutions, Inc.

[データ・アダプタ用に INSERT、UPDATE、DELETE コマンドを生成する] C# または Visual Basic ドキュメントにテーブルまたはビューをドラッグ・アンド・ドロップするときに、データ・アダプタ用の INSERT、UPDATE、DELETE コマンドを生成します。

[データ・アダプタ用にテーブル・マッピングを生成する] C# または Visual Basic ドキュメントにテーブルをドラッグ・アンド・ドロップするときに、データ・アダプタ用のテーブル・マッピングを生成します。

4. [Ultra Light] をクリックして、Ultra Light に固有のオプションを設定します。

   [テーブルをコードにドロップするときに生成する] アプリケーション・コードにドラッグ・アンド・ドロップするテーブルに特定タイプのコードを生成します。次のうちの 1 つを選択してください。

   - ULResultSet 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。
   - ULDataReader は、読み込み専用の結果セットを表します。
   - ULTable は、テーブル・データの格納、削除、更新、読み込みに使用するコードを表します。
   - ULDataAdapter は、テーブル・データをオフラインで使用するための方法です。

SQL Anywhere Explorer を使用した Ultra Light データベース・オブジェクトの追加

Visual Studio .NET では、SQL Anywhere Explorer から Visual Studio .NET デザイナに特定のデータベース・オブジェクトをドラッグ・アンド・ドロップすると、選択されたオブジェクトを参照する新しいコンポーネントが IDE によって自動的に作成されます。ドラッグ・アンド・ドロップ操作の設定を指定するには、Visual Studio .NET の [ツール] – [オプション] を選択します。

次の表に、SQL Anywhere Explorer からドラッグできる Ultra Light オブジェクトと、Visual Studio .NET Forms Designer またはコード・エディタにドロップしたときに作成されるコンポーネントを示します。

項目	結果
データ接続	データ接続を作成します。
テーブル	アダプタを作成します。

♦ SQL Anywhere Explorer を使用して、新しい Ultra Light データ・コンポーネントを作成するには、次の手順に従います。

1. データ・コンポーネントを追加するフォームまたはクラスを開きます。
2. SQL Anywhere Explorer で、使用する Ultra Light オブジェクトを選択します。
3. SQL Anywhere Explorer から Forms Designer またはコード・エディタにオブジェクトをドラッグします。

Copyright © 2007, iAnywhere Solutions, Inc.

SQL Anywhere Explorer を使用した Ultra Light テーブルの操作

SQL Anywhere Explorer では、Visual Studio .NET からデータベースの Ultra Light テーブルのプロパティとデータを表示できます。

♦ **Visual Studio .NET でテーブルを表示するには、次の手順に従います。**

1. SQL Anywhere Explorer を使用して、Ultra Light データベースに接続します。
2. [SQL Anywhere Explorer] ダイアログで、Ultra Light データベースを展開します。
3. テーブルを右クリックして、[データの取得] を選択します。

   選択したテーブルのデータが Visual Studio .NET の [出力] ウィンドウに表示されます。

Copyright © 2007, iAnywhere Solutions, Inc.

第 3 章

Ultra Light.NET 開発の概要

目次

Visual Studio .NET での SQL Anywhere ツールの使用 14
データベースへの接続 15
暗号化と難読化 17
SQL を使用したデータへのアクセスと操作 18
テーブル API によるデータ・アクセスと操作 22
トランザクションの管理 28
スキーマ情報へのアクセス 29
エラー処理 30
ユーザの認証 31
Ultra Light アプリケーションの同期 32

Copyright © 2007, iAnywhere Solutions, Inc.

Visual Studio .NET での SQL Anywhere ツールの使用

一部の SQL Anywhere ツールを Visual Studio .NET 2003 と 2005 に統合することで、SQL Anywhere データベースと Ultra Light データベースの両方に使用できる統合データベース開発環境を実現できます。

- SQL Anywhere Explorer を使用すると、テーブルやその中のデータを表示できるほか、データの操作や取得を行うためにデータベースへの接続を開くプログラムを設計できます。SQL Anywhere Explorer は、[表示] メニューから開くことができます。
- Sybase Central と Interactive SQL は、Visual Studio .NET を離れずに開くことができます。Sybase Central と Interactive SQL は、[ツール] メニューから開くことができます。

注意
Ultra Light 開発では、ビュー用のアダプタを追加したり、ストアド・プロシージャ用のコマンドを追加することはできません。これらの機能は SQL Anywhere データベースに対してのみ使用できます。

参照

- 「Visual Studio .NET での Ultra Light データベース接続の使用」 9 ページ
- 「SQL Anywhere Explorer を使用した Ultra Light データベース・オブジェクトの追加」 11 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

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。

  「トランザクションの管理」 28 ページを参照してください。

- **ユーザ認証** 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 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 ID を定義できます。

  「ユーザの認証」 31 ページを参照してください。

- **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。

  「Ultra Light アプリケーションの同期」 32 ページを参照してください。

- **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。

  「テーブル API によるデータ・アクセスと操作」 22 ページを参照してください。

- **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。

  「SQL を使用したデータへのアクセスと操作」 18 ページを参照してください。

「ULConnection クラス」 90 ページを参照してください。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

♦ **Ultra Light データベースに接続するには、次の手順に従います。**

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();
   ```

C# のコード・サンプル
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。

Copyright © 2007, iAnywhere Solutions, Inc.

暗号化と難読化

デフォルトでは、Ultra Light データベースは暗号化されていません。適切なデータベース作成パラメータを指定することで、データベースを作成するときに強力に暗号化したり簡単な難読化を適用したりできます。難読化とは、非常に弱い形態の、キーを使用しない暗号化のことで、(低レベル・ファイル検証ユーティリティが使用された場合など) データベース内のデータが偶発的に見られることを防ぐことだけを目的としています。

暗号化

強力な暗号化を使用してデータベースを作成するには、Sybase Central を使用してデータベースを作成するときに暗号化キーを指定します。データベースの作成に ULCreateDatabase を呼び出すか ulcreate ユーティリティを使用する場合は、該当するパラメータに暗号化キーを指定します。高い効果を得るために、文字、数字、特殊記号を組み合わせて暗号化キーを作成してください。暗号化キーは、長くするほど推測されにくくなります。

データベースが暗号化されると、暗号化キーは復元できなくなります。正しい暗号化キーを指定しないと、データベースにはまったくアクセスできません。暗号化キーは機密情報として扱い、適切に保管してください。

「Ultra Light DBKEY 接続パラメータ」 『Ultra Light - データベース管理とリファレンス』を参照してください。

既存の Ultra Light データベースの暗号化キーを変更するには、Connection.ChangeEncryptionKey メソッドを使用して新しい暗号化キーを適用します。

「ULConnection クラス」 90 ページと「ULConnectionParms クラス」 123 ページを参照してください。

データベースが暗号化された後は、データベースへの接続で正しい暗号化キーを指定する必要があります。指定しないと、接続は失敗します。

難読化

データベースを難読化するには、データベース作成パラメータとして **obfuscate=1** を指定します。データベースの暗号化と難読化のパラメータの詳細については、「Ultra Light で使用する作成時のデータベース・プロパティの選択」 『Ultra Light - データベース管理とリファレンス』を参照してください。

Copyright © 2007, iAnywhere Solutions, Inc.

SQL を使用したデータへのアクセスと操作

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。

テーブル API の使用方法の詳細については、「テーブル API によるデータ・アクセスと操作」 22 ページを参照してください。

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
- 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。SQL の機能の詳細については、「SQL 文」 『SQL Anywhere サーバ - SQL リファレンス』を参照してください。

データ操作：INSERT、UPDATE、DELETE

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。

「ULCommand クラス」 57 ページを参照してください。

SQL 文のパラメータのプレースホルダは、? 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での並び順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。

♦ ローを挿入するには、次の手順に従います。

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値を割り当てます。

   次のコードは、文字列パラメータを示します。

   ```
   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。

 Copyright © 2007, iAnywhere Solutions, Inc.

```
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();
```

♦ **ローを更新するには、次の手順に従います。**

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();
```

♦ **ローを削除するには、次の手順に従います。**

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();
```

Copyright © 2007, iAnywhere Solutions, Inc.

データの取得: SELECT

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。

♦ **SELECT 文を実行するには、次の手順に従います。**

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、オブジェクトのいくつかの種類の 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。以下のコードでは、SELECT 文の結果に文字列が含まれています。これはコンソール・ウィンドウへ出力されます。

```
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();
}
```

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。

Copyright © 2007, iAnywhere Solutions, Inc.

- **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・バッファが再配置されます。

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 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、スキーマ情報をコンソール・ウィンドウ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 © 2007, iAnywhere Solutions, Inc.

テーブル API によるデータ・アクセスと操作

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、テーブル API を使用したデータ・アクセスについて説明します。

SQL の詳細については、「SQL を使用したデータへのアクセスと操作」 18 ページを参照してください。

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

テーブルのローのナビゲーション

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。

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・バッファが再配置されます。

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 ) );
}
```


Copyright © 2007, iAnywhere Solutions, Inc.

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();
```

「ULTable クラス」 431 ページと「ULTableSchema クラス」 453 ページを参照してください。

Ultra Light のモードの使用

Ultra Light モードは、バッファ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。

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

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。

Copyright © 2007, iAnywhere Solutions, Inc.

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。「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』を参照してください。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

- **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。

♦ ローを検索するには、次の手順に従います。

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にのみ影響します。たとえば、次のコードは、バッファの値を 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。

   ```
   tCustomer.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。

「ULTable クラス」 431 ページを参照してください。

ローの更新

ローの更新手順を次に示します。

♦ ローを更新するには、次の手順に従います。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

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.BeginUpdate();
   ```

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。「トランザクションの管理」 28 ページを参照してください。

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。

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 入力可能なカラムの場合は 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。

 Copyright © 2007, iAnywhere Solutions, Inc.

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、挿入メソッドの一部としてコミットが実行されます。

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。

次のプロシージャは、ローを削除します。

♦ **ローを削除するには、次の手順に従います。**

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();
```

Copyright © 2007, iAnywhere Solutions, Inc.

トランザクションの管理

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。

接続に有効なトランザクションがある場合は、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);
// Carry out transaction operations here
txn.Commit();
```

Ultra Light の独立性レベル
Ultra Light は、IsolationLevel 列挙の IsolationLevel.ReadUncommitted メンバ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。

「ULConnection クラス」 90 ページと「ULTransaction クラス」 468 ページを参照してください。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

- **DatabaseSchema**　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 にアクセスします。

  「ULConnection クラス」 90 ページを参照してください。

- **TableSchema**　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 にアクセスします。

- **IndexSchema**　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使用します。

- **PublicationSchema**　パブリケーションに含まれるテーブルとカラムのリスト。パブリケーションはスキーマのみで構成されているため、Publication オブジェクトは存在しません。

  ULPublicationSchema オブジェクトを取得するには、ULDatabaseSchema.GetPublicationSchema メソッドを呼び出します。

「ULTableSchema クラス」 453 ページを参照してください。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

エラー・コードのリストについては、SQL Anywhere 10 - エラー・メッセージ『SQL Anywhere 10 - エラー・メッセージ』を参照してください。

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_COMMUNICATIONS_ERROR )
        {
          MessageBox.Show(
              "StreamErrorCode = " + _conn.SyncResult.StreamErrorCode.ToString() + "¥r¥n"
            + "StreamErrorContext = " + _conn.SyncResult.StreamErrorContext + "¥r¥n"
            + "StreamErrorID = " + _conn.SyncResult.StreamErrorID + "¥r¥n"
            + "StreamErrorSystem = " + _conn.SyncResult.StreamErrorSystem + "¥r¥n"
            );
        }
        else
        {
          MessageBox.Show(uEx.Message);
        }
      }
      catch(System.Exception ex )
      {
        MessageBox.Show(ex.Message);
      }
  }
```

次の項を参照してください。

- 「ULSyncProgressListener インタフェース」 421 ページ
- 「ULSyncResult クラス」 425 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ユーザの認証

新しいユーザは既存の接続から追加する必要があります。Ultra Light のすべてのデータベースは、デフォルトのユーザ ID DBA とパスワード sql を使用して作成されるため、まずこのユーザとして接続します。

ユーザ ID を直接変更することはできません。新しいユーザ ID を追加して、既存のユーザ ID を削除します。Ultra Light では、Ultra Light データベースごとに ID を 4 つまでサポートします。

「ユーザの認証」 31 ページを参照してください。

- **ユーザを追加する、または既存のユーザのパスワードを変更するには、次の手順に従います。**

1. 既存のユーザのユーザ ID とパスワードを使用してデータベースに接続します。
2. ULConnection.GrantConnectTo メソッドを使用して、希望するパスワードによるデータベースへのアクセスをユーザに付与します。

   新規ユーザを追加する場合も、既存のユーザのパスワードを変更する場合も、この手順は同じです。

   「ULConnection クラス」 90 ページを参照してください。

- **既存のユーザを削除するには、次の手順に従います。**

1. 既存のユーザのユーザ ID とパスワードを使用してデータベースに接続します。
2. Connection.RevokeConnectFrom メソッドを使用して既存のユーザを削除します。

Copyright © 2007, iAnywhere Solutions, Inc.

Ultra Light アプリケーションの同期

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。

同期の詳細については、「Ultra Light クライアント」 『Mobile Link - クライアント管理』を参照してください。

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 10 インストール環境の *Samples¥UltraLite.NET¥CustDB* サブディレクトリ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・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。いずれの場合でも、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。

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 承認の暗号化には、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 10 - 紹介』を参照してください。

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()
{
    // Sync
    try
    {
        // This line would be used to synchronize a specific publication called "high_priority"
        conn.SyncParms.PublicationMask = conn.Schema.GetPublicationSchema
("high_priority").Mask;

        // 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);
       }
    }
}
```


Copyright © 2007, iAnywhere Solutions, Inc.

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・ユーザのコンピュータ上の 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] ウィンドウで同期コマンド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・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 );
```

詳細とサンプル・コードについては、「ULActiveSyncListener インタフェース」 53 ページを参照してください。

Ultra Light は、ActiveSync メッセージを受信すると、指定されたリスナの 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・インタフェー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

Copyright © 2007, iAnywhere Solutions, Inc.

第 4 章

チュートリアル：Ultra Light.NET アプリケーションの構築

目次

Ultra Light.NET 開発のチュートリアル 36
レッスン 1：Visual Studio プロジェクトの作成 37
レッスン 2：Ultra Light データベースの作成 40
レッスン 3：データベースへの接続 41
レッスン 4：データの挿入、更新、削除 43
レッスン 5：アプリケーションのビルドと配置 48

Copyright © 2007, iAnywhere Solutions, Inc.

Ultra Light.NET 開発のチュートリアル

このチュートリアルでは、Microsoft Visual Studio で Ultra Light.NET 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では iAnywhere.Data.UltraLite ネームスペースに用意されている ADO.NET インタフェースを使用します。

このチュートリアルには、Visual Basic .NET 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 .NET プログラミング言語に精通している。
- Microsoft Visual Studio がコンピュータ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 .NET 2003 を使用して作成されており、Visual Studio のその他のバージョンとは微妙に異なる Visual Studio アクションやプロシージャに基づいている場合があります。
- Sybase Central で Ultra Light 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。

  「Sybase Central からの 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』を参照してください。

目的

このチュートリアルの目的は、Visual Studio 環境での Ultra Light.NET 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、Ultra Light を Visual Studio と統合するプロセスを次の手順に従って手動で実行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 .NET 2003 の場合は、*install-dir¥ultralite¥UltraLite.NET¥assembly¥v1¥* フォルダから *installULNet.exe* を実行します。

  Visual Studio 2005 の場合は、*install-dir¥ultralite¥UltraLite.NET¥assembly¥v2¥* フォルダから *installULNet.exe* を実行します。

Copyright © 2007, iAnywhere Solutions, Inc.

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.NET と C# のどちらも使用できます。

♦ **Visual Studio プロジェクトを作成するには、次の手順に従います。**

1. Visual Studio プロジェクトを作成します。

   ♦ Visual Studio の [ファイル] メニューから [新規作成] – [プロジェクト] を選択して、新規プロジェクトを作成します。

   [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 プロジェクト] フォルダまたは [Visual C# プロジェクト] フォルダを選択します。

   右ウィンドウ枠で、[スマート デバイス アプリケーション] を選択し、プログラミング言語として Visual Basic と C# のどちらを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。

   [スマート デバイス アプリケーション] ウィザードが表示されます。

   ♦ Pocket PC をターゲット・プラットフォームとして選択し、プロジェクト・タイプとして [Windows アプリケーション] を選択します。[OK] をクリックします。

   [デザイン] 作業領域が表示され、Form1 という名前のフォームが表示されます。

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] メニューで [参照の追加] を選択します。

      [参照の追加] ウィンドウが表示されます。

      b. 利用可能な参照のリストから [**iAnywhere.Data.UltraLite (CE)**] を選択します。[選択] 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。

      この参照がリストに表示されない場合は、[参照] をクリックし、SQL Anywhere インストール環境の *ultralite¥UltraLite.NET¥ce* サブディレクトリで探します。*iAnywhere.Data.UltraLite.dll* を選択し、[開く] をクリックします。

      c. 利用可能な参照のリストから [**iAnywhere.Data.UltraLite (CE) JA**] を選択します。[選択] 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。

      この参照がリストに表示されない場合は、[参照] をクリックし、SQL Anywhere インストール環境の *ultralite¥UltraLite.NET¥ce¥xx* サブディレクトリで探します。この場

Copyright © 2007, iAnywhere Solutions, Inc.

合、xx は言語を示す 2 文字の略語です。*iAnywhere.Data.UltraLite.resources.dll* を選択し、[開く] をクリックします。

d. [OK] をクリックし、アセンブリとリソースをプロジェクトに追加します。

- Ultra Light コンポーネントをプロジェクトにリンクします。

  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* サブディレクトリを参照します。

  b. [ファイルの種類] ドロップダウン・リストから [すべてのファイル] を選択して、DLL ファイルを表示します。

  c. 使用している CE デバイスのプロセッサに対応するフォルダを開きます。Visual Studio .NET 2003 の Pocket PC エミュレータの場合は x86 フォルダを開き、Visual Studio 2005 の場合は ARM フォルダを開きます。*ulnet10.dll* を選択します。*[開く] はクリックしません。*

  d. [開く] ボタンの矢印をクリックし、[**リンク ファイル**] を選択してプロジェクトにリンクします。

- Visual Studio 2005 を使用している場合は、dll の [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 [新しい場合はコピーする] に設定します。

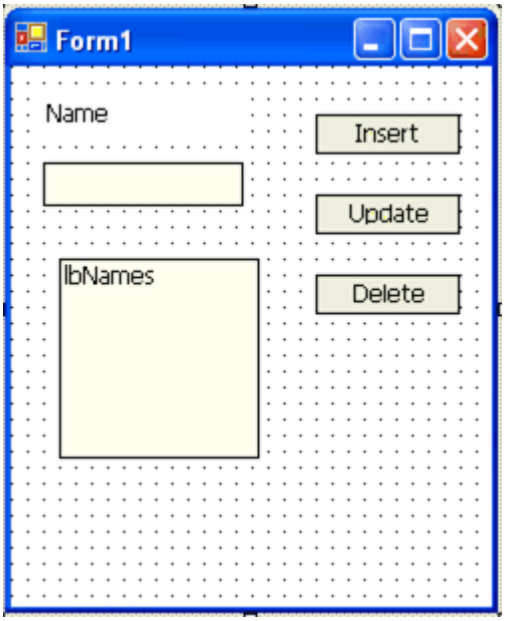
3. アプリケーションのフォームを作成します。

   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・メニューから [表示] – [ツールボックス] を選択します。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	名前	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。

 Copyright © 2007, iAnywhere Solutions, Inc.



4. ソリューションを構築および配置します。

   ソリューションを構築および配置し、Visual Studio.NET 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] メニューから [ソリューションのビルド] を選択します。プロジェクトが正常に構築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。

      参照されたアセンブリ「iAnywhere.Data.UltraLite.resources」はローカライズされたサテライト アセンブリです。

   b. [デバッグ] メニューから [開始] を選択します。

      これによって、アプリケーションがデバイスまたはエミュレータに配置され、起動されます。プロジェクト名に応じて、アプリケーションが *¥Program Files¥VBApp* または *¥Program Files¥CSApp* に配備されます。

      配置には時間がかかる場合があります。

   c. アプリケーションがエミュレータまたはターゲット・デバイスに配置されたことを確認します。

   d. [デバッグ] メニューから [デバッグの停止] を選択し、アプリケーションを閉じます。

Copyright © 2007, iAnywhere Solutions, Inc.

レッスン 2：Ultra Light データベースの作成

次の手順では、Ultra Light データベースを Sybase Central を使用して作成します。

「Sybase Central からの 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』を参照してください。

- **データベースを作成するには、次の手順に従います。**

1. Sybase Central を起動します。

   [スタート] – [プログラム] – [SQL Anywhere 10] – [Sybase Central] を選択します。

2. Sybase Central で 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次の特性を持つデータベースを作成します。

   - **データベース・ファイル名**　*VBApp.udb* または *CSApp.udb* (アプリケーションによって異なります)
   - **照合順**　デフォルトの照合の使い方
   - **大文字と小文字の区別**　このオプションはクリアしておきます。
   - **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 入力可	デフォルト値
ID	integer	いいえ	グローバル・オートインクリメント
Name	varchar(30)	いいえ	なし

   - **プライマリ・キー**　ID カラムを昇順に指定します。

3. 初期化済みの (空の) データベース・ファイルを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 のメニューから、[プロジェクト] – [既存項目の追加] を選択します。
   - 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、アプリケーションに応じてファイル *VBApp.udb* または *CSApp.udb* を選択します。*[開く] はクリックしません。*
   - [開く] ボタンの矢印をクリックし、[リンク ファイル] 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] を選択します。

     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] に設定します。Visual Studio 2005 を使用している場合は、[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。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

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作成した場合は、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。

♦ **アプリケーションに Ultra Light 接続を追加するには、次の手順に従います。**

1. フォームのソース・コードを開きます。

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

2. 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# アプリケーションの場合は、フォーム・コンポーネントが記述されたコードの後と、最初のメソッド宣言の前に、次のコードを追加します。

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

   Visual Basic プロジェクトの場合は、フォーム・コンポーネントの宣言 (FriendsWithEvents 宣言) の後と、Form1 のメソッド宣言の前に、次のコードを追加します。

   ```
   '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。

Copyright © 2007, iAnywhere Solutions, Inc.

- ulConnectionParms1 コントロールで設定されている接続パラメータを使用して、データベースへの接続を開きます。
- このチュートリアルで後ほど定義する RefreshListBox メソッドを呼び出します。
- 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。

  SQL Anywhere 10 - エラー・メッセージ『SQL Anywhere 10 - 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] メニューから [ソリューションのビルド] を選択します。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 class or namespace 'CSApp.Form1'」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、**Ultra Light**、**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。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

「テーブル API によるデータ・アクセスと操作」 22 ページを参照してください。

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を追加するには、次の手順に従います。

1. フォームを右クリックし、[コードの表示] を選択します。
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。

     SQL Anywhere 10 - エラー・メッセージ『SQL Anywhere 10 - エラー・メッセージ』を参照してください。

   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();
            NumRows = dr.RowCount;
            ids = new int[ NumRows ];
            while (dr.MoveNext())
            {
               lbNames.Items.Add(
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
                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 を実装するには、次の手順に従います。

1. [Insert] ボタン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。

Copyright © 2007, iAnywhere Solutions, Inc.

SQL Anywhere 10 - エラー・メッセージ『SQL Anywhere 10 - 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. [Update] ボタン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。

     SQL Anywhere 10 - エラー・メッセージ『SQL Anywhere 10 - 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);
        RowsUpdated = cmd.ExecuteNonQuery();
    }
    RefreshListBox();
}
catch( Exception err ) {
    MessageBox.Show(
        "Exception: " + err.Message);
}
```

Visual Basic の場合は、次のコードを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. [Delete] ボタン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。

     SQL Anywhere 10 - エラー・メッセージ『SQL Anywhere 10 - 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);
```


Copyright © 2007, iAnywhere Solutions, Inc.

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

Visual Basic の場合は、次のコードを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構築し、正常にコンパイルされることを確認します。

Copyright © 2007, iAnywhere Solutions, Inc.

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構築し、リモート・デバイスに配置します。

♦ **アプリケーションを配置するには、次の手順に従います。**

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 *ulnet10.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] を選択します。

   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、Pocket PC エミュレータに配置されます。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。「レッスン 3：データベースへの接続」 41 ページを参照してください。
- 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] に設定したことを確認します。Visual Studio 2005 を使用している場合は、[出力ディレクトリにコピー] を [常にコピーする] または [新しい場合はコピーする] に設定していることを確認します。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 *ulnet10.dll* への参照を追加したこと、または Windows CE インストーラ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。「レッスン 1：Visual Studio プロジェクトの作成」 37 ページを参照してください。
- エミュレータの状態を保存しないでエミュレータを終了できます。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。

 Copyright © 2007, iAnywhere Solutions, Inc.

♦ **アプリケーションをテストするには、次の手順に従います。**

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。

   リストボックスから名前を選択します。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。

   リストボックスから名前を選択します。[Delete] をクリックします。名前はリストボックスに表示されなくなります。

これでチュートリアルが完了しました。

Copyright © 2007, iAnywhere Solutions, Inc.

第 5 章

Ultra Light .NET 1.0 API リファレンス

目次

ULActiveSyncListener インタフェース 53
ULAuthStatusCode 列挙 56
ULCommand クラス 57
ULCommandBuilder クラス 82
ULConnection クラス 90
ULConnectionParms クラス 123
ULConnectionParms.UnusedEventHandler 委任 136
ULCreateParms クラス 137
ULCursorSchema クラス 147
ULDataAdapter クラス 156
ULDatabaseManager クラス 182
ULDatabaseSchema クラス 190
ULDataReader クラス 201
ULDateOrder enumeration 244
ULDbType 列挙 245
ULException クラス 249
ULFileTransfer クラス 253
ULFileTransferProgressData クラス 268
ULFileTransferProgressListener インタフェース 271
ULIndexSchema クラス 273
ULInfoMessageEventArgs クラス 282
ULInfoMessageEventHandler 委任 285
ULParameter クラス 286
ULParameterCollection クラス 303
ULPublicationSchema クラス 321
ULResultSet クラス 325
ULResultSetSchema クラス 351
ULRowUpdatedEventArgs クラス 353
ULRowUpdatedEventHandler 委任 357

Copyright © 2007, iAnywhere Solutions, Inc.

ULRowUpdatingEventArgs クラス 358
ULRowUpdatingEventHandler 委任 361
ULRuntimeType 列挙 362
ULServerSyncListener インタフェース 363
ULSQLCode 列挙 366
ULStreamErrorCode 列挙 376
ULStreamErrorContext 列挙 392
ULStreamErrorID 列挙 393
ULStreamType 列挙 395
ULSyncParms クラス 396
ULSyncProgressData クラス 411
ULSyncProgressListener インタフェース 421
ULSyncProgressState 列挙 423
ULSyncResult クラス 425
ULTable クラス 431
ULTableSchema クラス 453
ULTransaction クラス 468

 Copyright © 2007, iAnywhere Solutions, Inc.

ULActiveSyncListener インタフェース

UL 拡張： ActiveSync イベントを受信するリスナ・インタフェースです。

構文

Visual Basic

Public Interface **ULActiveSyncListener**

C#

public interface **ULActiveSyncListener**

ULActiveSyncListener メンバ

パブリック・メソッド

メンバ名	説明
ActiveSyncInvoked メソッド	ActiveSync 用の Mobile Link プロバイダ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。

構文

Visual Basic

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

C#

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

パラメータ

- **launchedByProvider**　ActiveSync 同期を実行するために、Mobile Link プロバイダ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。

Copyright © 2007, iAnywhere Solutions, Inc.

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・インタフェー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() を呼び出して、ActiveSync 用の Mobile Link プロバイダに対して通知する必要があります。

例

次のコード・フラグメントは、ActiveSync 要求を受信して、UI スレッドで同期を実行する方法を示しています。

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

Public Class MainWindow
  Inherits System.Windows.Forms.Form
  Implements ULActiveSyncListener
  Private conn As ULConnection

  Public Sub New(ByVal args() As String)

    MyBase.New()

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

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

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

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

  Public Sub ActiveSyncAction( _
      ByVal sender As Object, ByVal e As EventArgs _
    )
    ' Do active sync
    conn.Synchronize()
    ULConnection.DatabaseManager.SignalSyncIsComplete()
  End Sub
End Class
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
// 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
    // 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();
  }
}
```

参照

- 「ULActiveSyncListener インタフェース」 53 ページ
- 「ULActiveSyncListener メンバ」 53 ページ
- 「SignalSyncIsComplete メソッド」 188 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULAuthStatusCode 列挙

UL 拡張： Mobile Link ユーザ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。

構文

Visual Basic

```
Public Enum ULAuthStatusCode
  Inherits Short
```

C#

```
public enum ULAuthStatusCode :
  short
```

メンバ名

メンバ名	説明
EXPIRED	ユーザ 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)。
IN_USE	ユーザ 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)。
INVALID	ユーザ 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)。
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)。
VALID	ユーザ ID とパスワードは、同期時には有効でした (VALID = 1)。
VALID_BUT_EXPIRES_SOON	ユーザ 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)。

参照

- 「AuthStatus プロパティ」 426 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

ULCommand クラス

IN パラメータあり、または IN パラメータ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。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。

構文

Visual Basic

NotInheritable Public Class **ULCommand**
  Inherits Component

C#

public sealed class **ULCommand** :
  Component

備考

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 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。

方法	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 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 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。
ULCommand.ExecuteTable()	**UL 拡張：** 直接の操作用に、ULTable のデータベース・テーブルを取り出します。ULCommand.CommandText はテーブルの名前として解釈され、ULCommand.IndexName はテーブルのソート順の指定に使用できます。ULCommand.CommandType は System.Data.CommandType.TableDirect であることが必要です。

ULCommand.CommandText 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。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

実装： System.Data.IDbCommand、System.IDisposable

参照

- 「ULCommand メンバ」 58 ページ
- 「CreateCommand メソッド」 108 ページ
- 「Transaction プロパティ」 68 ページ
- 「ExecuteNonQuery メソッド」 71 ページ
- 「ExecuteReader メソッド」 72 ページ
- 「ExecuteResultSet メソッド」 75 ページ
- 「ExecuteScalar メソッド」 77 ページ
- 「ExecuteTable メソッド」 78 ページ
- 「ULTable クラス」 431 ページ
- 「CommandText プロパティ」 63 ページ
- CommandType.TableDirect
- System.ComponentModel.Component.Dispose
- 「CommandType プロパティ」 65 ページ
- 「IndexName プロパティ」 67 ページ
- IDbCommand
- IDisposable
- 「ULResultSet クラス」 325 ページ
- 「ULDataReader クラス」 201 ページ

ULCommand メンバ

パブリック・コンストラクタ

メンバ名	説明
ULCommand コンストラクタ	ULCommand オブジェクトを初期化します。
ULCommand コンストラクタ	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 コンストラクタ	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 コンストラクタ	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。

Copyright © 2007, iAnywhere Solutions, Inc.

パブリック・プロパティ

メンバ名	説明
CommandText プロパティ	ULCommand.CommandType 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。パラメータ化された文の場合、疑問符 (?) プレースホルダを使用してパラメータを渡します。
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 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 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を指定します。
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。このプロパティは、主に開発中の使用を目的とします。
Transaction プロパティ	ULCommand が実行される ULTransaction を指定します。
UpdatedRowSource プロパティ	ULDataAdapter の Update 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。

パブリック・メソッド

メンバ名	説明
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を指定するために ULParameter オブジェクトを提供します。
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。
ExecuteReader 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。
ExecuteResultSet メソッド	**UL 拡張：** SQL SELECT 文を実行し、ULResultSet として結果セットを返します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
ExecuteResultSet 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 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 のデータベース・テーブルを取り出します。ULCommand.CommandText はテーブルの名前として解釈され、ULCommand.IndexName はテーブルのソート順の指定に使用できます。
ExecuteTable 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。ULCommand.CommandText はテーブルの名前として解釈され、ULCommand.IndexName はテーブルのソート順の指定に使用できます。
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。

ULCommand コンストラクタ

ULCommand オブジェクトを初期化します。

構文

Visual Basic

Overloads Public Sub **New()**

C#

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

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「CreateCommand メソッド」 108 ページ
- 「ULCommand コンストラクタ」 61 ページ
- 「ULCommand コンストラクタ」 61 ページ
- 「ULCommand コンストラクタ」 62 ページ
- 「CommandText プロパティ」 63 ページ
- 「Connection プロパティ」 66 ページ
- 「Transaction プロパティ」 68 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULCommand コンストラクタ

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。

構文

Visual Basic

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

C#

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

パラメータ

- **cmdText**　ULCommand.CommandType 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を渡します。

備考

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

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「CreateCommand メソッド」 108 ページ
- 「ULCommand コンストラクタ」 60 ページ
- 「ULCommand コンストラクタ」 61 ページ
- 「ULCommand コンストラクタ」 62 ページ
- 「Connection プロパティ」 66 ページ
- 「Transaction プロパティ」 68 ページ
- 「CommandType プロパティ」 65 ページ
- CommandType.TableDirect

ULCommand コンストラクタ

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。

構文

Visual Basic

```
Overloads Public Sub New( _
  ByVal cmdText As String, _
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
  ByVal connection As ULConnection _
)
```

C#

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

パラメータ

- **cmdText**　ULCommand.CommandType 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を渡します。
- **connection**　現在の接続を示す ULConnection オブジェクト

備考

ULCommand オブジェクトに ULCommand.Transaction 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

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「CreateCommand メソッド」 108 ページ
- 「ULCommand コンストラクタ」 60 ページ
- 「ULCommand コンストラクタ」 61 ページ
- 「ULCommand コンストラクタ」 62 ページ
- 「Transaction プロパティ」 68 ページ
- 「CommandType プロパティ」 65 ページ
- CommandType.TableDirect
- 「ULConnection クラス」 90 ページ

ULCommand コンストラクタ

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。

構文

Visual Basic

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

C#

Copyright © 2007, iAnywhere Solutions, Inc.

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

パラメータ

- **cmdText**　ULCommand.CommandType 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を渡します。
- **connection**　現在の接続を示す ULConnection オブジェクト
- **transaction**　ULCommand が実行される ULTransaction

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「CreateCommand メソッド」 108 ページ
- 「ULCommand コンストラクタ」 60 ページ
- 「ULCommand コンストラクタ」 61 ページ
- 「ULCommand コンストラクタ」 61 ページ
- 「CommandType プロパティ」 65 ページ
- CommandType.TableDirect
- 「ULTransaction クラス」 468 ページ

CommandText プロパティ

ULCommand.CommandType 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。パラメータ化された文の場合、疑問符 (?) プレースホルダを使用してパラメータを渡します。

構文

Visual Basic

```
NotOverridable Public Property CommandText As String _
  Implements IDbCommand.CommandText
```

C#

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

プロパティ値

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。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

読み込み専用の結果セットの作成に使用する SELECT 文 (ULCommand.ExecuteReader() または ULCommand.ExecuteScalar) は、最後に FOR READ ONLY を指定することをおすすめします。テンポラリ・テーブルを使用する文で、パフォーマンスが大幅に向上することがあります。

実装

IDbCommand.CommandText

例

次の例は、パラメータ化されたプレースホルダの使用方法を示します。

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

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

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「ExecuteNonQuery メソッド」 71 ページ
- 「ExecuteReader メソッド」 72 ページ
- 「ExecuteResultSet メソッド」 75 ページ
- 「ExecuteScalar メソッド」 77 ページ
- 「ExecuteTable メソッド」 78 ページ
- 「CommandType プロパティ」 65 ページ
- CommandType.TableDirect

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。

構文

Visual Basic

```
NotOverridable Public Property CommandTimeout As Integer _
  Implements IDbCommand.CommandTimeout
```

C#

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

プロパティ値

値は常にゼロです。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

実装

IDbCommand.CommandTimeout

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。

構文

Visual Basic

```
NotOverridable Public Property CommandType As CommandType _
  Implements IDbCommand.CommandType
```

C#

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

プロパティ値

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

備考

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 を指定するときは、ULCommand.CommandText はデータベース・テーブルの名前でなければなりません。また、ULCommand.IndexName 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() または ULCommand.ExecuteReader() を使用します。
- System.Data.CommandType.Text - この CommandType を指定するときは、ULCommand.CommandText は SQL 文またはクエリでなければなりません。クエリ以外の SQL 文を実行するには、ULCommand.ExecuteNonQuery 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 を使用します。

例外

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

実装

IDbCommand.CommandType

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- CommandType
- CommandType.Text
- CommandType.TableDirect

Copyright © 2007, iAnywhere Solutions, Inc.

- 「CommandText プロパティ」 63 ページ
- 「IndexName プロパティ」 67 ページ
- 「ExecuteTable メソッド」 78 ページ
- 「ExecuteReader メソッド」 72 ページ
- CommandType.Text
- 「CommandText プロパティ」 63 ページ
- 「ExecuteNonQuery メソッド」 71 ページ
- 「ExecuteReader メソッド」 72 ページ
- 「ExecuteScalar メソッド」 77 ページ

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。

構文

Visual Basic

Public Property **Connection** As ULConnection

C#

public ULConnection **Connection** {get;set;}

プロパティ値

コマンドを実行する ULConnection オブジェクト。

備考

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 が厳密に型指定されたものです。

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「ULConnection クラス」 90 ページ
- IDbCommand.Connection

DesignTimeVisible プロパティ

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

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **DesignTimeVisible** As Boolean

C#

public bool **DesignTimeVisible** {get;set;}

プロパティ値

ULCommand インスタンスを参照できるようにする場合は true、このインスタンス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 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。

構文

Visual Basic

Public Property **IndexName** As String

C#

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

プロパティ値

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 クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「ExecuteTable メソッド」 78 ページ
- 「ExecuteReader メソッド」 72 ページ
- 「CommandType プロパティ」 65 ページ
- CommandType.TableDirect

Parameters プロパティ

現在の文のパラメータを指定します。

構文

Visual Basic

Public Readonly Property **Parameters** As ULParameterCollection

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public ULParameterCollection **Parameters** {get;}

プロパティ値

SQL 文のパラメータを保持する ULParameterCollection。デフォルト値は空のコレクションです。

備考

ULCommand.CommandText で疑問符を使用してパラメータを示します。コレクション内のパラメータは、疑問符のプレースホルダ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の数と同じでなければなりません。

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

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「ULParameterCollection クラス」 303 ページ
- 「ULParameter クラス」 286 ページ
- IDbCommand.Connection
- 「CommandText プロパティ」 63 ページ

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。このプロパティは、主に開発中の使用を目的とします。

構文

Visual Basic

Public Readonly Property **Plan** As String

C#

public string **Plan** {get;}

プロパティ値

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。

例外

- ULException クラス - SQL エラーが発生しました。

Transaction プロパティ

ULCommand が実行される ULTransaction を指定します。

 Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **Transaction** As ULTransaction

C#

public ULTransaction **Transaction** {get;set;}

プロパティ値

ULCommand が実行される ULTransaction。これは、ULCommand.Connection 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。

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 が厳密に型指定されたものです。

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「BeginTransaction メソッド」 103 ページ
- 「ULTransaction クラス」 468 ページ
- 「Connection プロパティ」 66 ページ
- IDbCommand.Transaction

UpdatedRowSource プロパティ

ULDataAdapter の Update 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。

構文

Visual Basic

NotOverridable Public Property **UpdatedRowSource** As UpdateRowSource _
  Implements IDbCommand.UpdatedRowSource

C#

public UpdateRowSource **UpdatedRowSource** {get;set;}

プロパティ値

System.Data.UpdateRowSource 値の 1 つ。デフォルト値は System.Data.UpdateRowSource.Both です。

Copyright © 2007, iAnywhere Solutions, Inc.

実装

IDbCommand.UpdatedRowSource

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- UpdateRowSource
- UpdateRowSource.Both

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。

構文

Visual Basic

```
NotOverridable Public Sub Cancel() _
  Implements IDbCommand.Cancel
```

C#

```
public 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。

実装

IDbCommand.Cancel

CreateParameter メソッド

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

構文

Visual Basic

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

C#

```
public 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。

Copyright © 2007, iAnywhere Solutions, Inc.

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 にプロパティを設定して、そのパラメータの値を指定できます。

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

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「ULParameter クラス」 286 ページ
- IDbCommand.CreateParameter

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。

構文

Visual Basic

```
NotOverridable Public Function ExecuteNonQuery() As Integer _
  Implements IDbCommand.ExecuteNonQuery
```

C#

```
public int ExecuteNonQuery();
```

戻り値

影響を受けたローの数。

備考

この文は、必要に応じて ULCommand.CommandText と ULCommand.Parameters 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 を System.Data.CommandType.TableDirect にすることはできません。

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 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 が無効です。

Copyright © 2007, iAnywhere Solutions, Inc.

実装

IDbCommand.ExecuteNonQuery

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「CommandText プロパティ」 63 ページ
- 「Parameters プロパティ」 67 ページ
- 「CommandType プロパティ」 65 ページ
- CommandType.TableDirect
- 「Connection プロパティ」 66 ページ
- 「Transaction プロパティ」 68 ページ
- 「CommandText プロパティ」 63 ページ

ExecuteReader メソッド

SQL SELECT 文を実行し、結果セットを返します。

構文

Visual Basic

Overloads Public Function **ExecuteReader()** As ULDataReader

C#

public ULDataReader **ExecuteReader();**

戻り値

ULDataReader オブジェクトとして返される結果セット。

備考

この文は、必要に応じて ULCommand.CommandText と ULCommand.Parameters 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()、ULCommand.ExecuteTable()、または ULDataAdapter を使用します。

ULCommand.CommandType が System.Data.CommandType.TableDirect である場合、ExecuteReader は ULCommand.ExecuteTable() を実行し、ULDataReader としての ULTable ダウンキャストを返します。

このメソッドを使用して読み込み専用の結果セットを作成する SELECT 文は、最後に FOR READ ONLY を指定することをおすすめします。テンポラリ・テーブルを使用する文で、パフォーマンスが大幅に向上することがあります。

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

例外

- ULException クラス - SQL エラーが発生しました。

Copyright © 2007, iAnywhere Solutions, Inc.

- InvalidOperationException - コマンドのステータスは有効ではありません。ULCommand.Connection 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 が無効です。

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「ExecuteReader メソッド」 73 ページ
- 「CommandText プロパティ」 63 ページ
- 「Parameters プロパティ」 67 ページ
- 「ULDataReader クラス」 201 ページ
- 「ExecuteResultSet メソッド」 75 ページ
- 「ExecuteTable メソッド」 78 ページ
- 「ULDataAdapter クラス」 156 ページ
- 「CommandType プロパティ」 65 ページ
- CommandType.TableDirect
- 「ExecuteTable メソッド」 78 ページ
- 「ULTable クラス」 431 ページ
- 「ULDataReader クラス」 201 ページ
- IDbCommand.ExecuteReader
- 「Connection プロパティ」 66 ページ
- 「Transaction プロパティ」 68 ページ

ExecuteReader 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。

構文

Visual Basic

```
Overloads Public Function ExecuteReader( _
  ByVal cmdBehavior As CommandBehavior _
) As ULDataReader
```

C#

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

パラメータ

- **cmdBehavior**　クエリの結果の記述と、接続への影響の記述の System.Data.CommandBehavior フラグのビット処理の組み合わせ。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。

Copyright © 2007, iAnywhere Solutions, Inc.

戻り値

ULDataReader オブジェクトとして返される結果セット。

備考

この文は、必要に応じて ULCommand.CommandText と ULCommand.Parameters 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(CommandBehavior)、ULCommand.ExecuteTable(CommandBehavior)、または ULDataAdapter を使用します。

ULCommand.CommandType が System.Data.CommandType.TableDirect である場合、ExecuteReader は ULCommand.ExecuteTable(CommandBehavior) を実行し、ULDataReader としての ULTable ダウンキャストを返します。

このメソッドを使用して読み込み専用の結果セットを作成する SELECT 文は、最後に FOR READ ONLY を指定することをおすすめします。テンポラリ・テーブルを使用する文で、パフォーマンスが大幅に向上することがあります。

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

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 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 が無効です。

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「ExecuteReader メソッド」 72 ページ
- 「CommandText プロパティ」 63 ページ
- 「Parameters プロパティ」 67 ページ
- 「ULDataReader クラス」 201 ページ
- 「ExecuteResultSet メソッド」 76 ページ
- 「ExecuteTable メソッド」 79 ページ
- 「ULDataAdapter クラス」 156 ページ
- 「CommandType プロパティ」 65 ページ
- CommandType.TableDirect
- 「ExecuteTable メソッド」 79 ページ
- 「ULTable クラス」 431 ページ
- 「ULDataReader クラス」 201 ページ
- IDbCommand.ExecuteReader
- CommandBehavior
- CommandBehavior.Default
- CommandBehavior.CloseConnection
- CommandBehavior.SchemaOnly

Copyright © 2007, iAnywhere Solutions, Inc.

- 「Connection プロパティ」 66 ページ
- 「Transaction プロパティ」 68 ページ
- 「CommandText プロパティ」 63 ページ

ExecuteResultSet メソッド

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

構文

Visual Basic

Overloads Public Function **ExecuteResultSet()** As ULResultSet

C#

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。

備考

この文は、必要に応じて ULCommand.CommandText と ULCommand.Parameters 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() または ULDataAdapter を使用します。

ULCommand.CommandType が System.Data.CommandType.TableDirect である場合、ExecuteReader は ULCommand.ExecuteTable() を実行し、ULResultSet としての ULTable ダウンキャストを返します。

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 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 が無効です。

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「ExecuteResultSet メソッド」 76 ページ
- 「ULCommand クラス」 57 ページ
- 「CommandText プロパティ」 63 ページ
- 「Parameters プロパティ」 67 ページ
- 「CommandType プロパティ」 65 ページ
- CommandType.TableDirect

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULResultSet クラス」 325 ページ
- 「ULTable クラス」 431 ページ

ExecuteResultSet 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 として結果セットを返します。

構文

Visual Basic

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

C#

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

パラメータ

- **cmdBehavior** クエリの結果の記述と、接続への影響の記述の System.Data.CommandBehavior フラグのビット処理の組み合わせ。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。

備考

この文は、必要に応じて ULCommand.CommandText と ULCommand.Parameters 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) または ULDataAdapter を使用します。

ULCommand.CommandType が System.Data.CommandType.TableDirect である場合、ExecuteReader は ULCommand.ExecuteTable(CommandBehavior) を実行し、ULResultSet としての ULTable ダウンキャストを返します。

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 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 が無効です。

 Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「ExecuteReader メソッド」 72 ページ
- 「ULResultSet クラス」 325 ページ
- 「CommandText プロパティ」 63 ページ
- 「Parameters プロパティ」 67 ページ
- 「ExecuteTable メソッド」 79 ページ
- 「ULDataAdapter クラス」 156 ページ
- 「CommandType プロパティ」 65 ページ
- CommandType.TableDirect
- 「ExecuteTable メソッド」 79 ページ
- 「ULTable クラス」 431 ページ
- CommandBehavior
- CommandBehavior.Default
- CommandBehavior.CloseConnection
- CommandBehavior.SchemaOnly
- 「Connection プロパティ」 66 ページ
- 「Transaction プロパティ」 68 ページ

ExecuteScalar メソッド

SQL SELECT 文を実行し、単一の値を返します。

構文

Visual Basic

```
NotOverridable Public Function ExecuteScalar() As Object _
  Implements IDbCommand.ExecuteScalar
```

C#

```
public 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)。

備考

この文は、必要に応じて ULCommand.CommandText と ULCommand.Parameters 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 が System.Data.CommandType.TableDirect である場合、ExecuteScalar は ULCommand.ExecuteTable() を実行し、最初のローの最初のカラムを返します。

Copyright © 2007, iAnywhere Solutions, Inc.

このメソッドを使用して読み込み専用の結果セットを作成する SELECT 文は、最後に FOR READ ONLY を指定することをおすすめします。テンポラリ・テーブルを使用する文で、パフォーマンスが大幅に向上することがあります。

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 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 が無効です。

実装

IDbCommand.ExecuteScalar

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「Connection プロパティ」 66 ページ
- 「Transaction プロパティ」 68 ページ
- 「CommandText プロパティ」 63 ページ
- 「IndexName プロパティ」 67 ページ
- CommandBehavior
- CommandBehavior.Default
- CommandBehavior.CloseConnection
- CommandBehavior.SchemaOnly
- 「CommandType プロパティ」 65 ページ
- 「Parameters プロパティ」 67 ページ
- CommandType.TableDirect

ExecuteTable メソッド

UL 拡張： 直接の操作用に、ULTable のデータベース・テーブルを取り出します。ULCommand.CommandText はテーブルの名前として解釈され、ULCommand.IndexName はテーブルのソート順の指定に使用できます。

構文

Visual Basic

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

C#

```
public ULTable ExecuteTable();
```

戻り値

ULTable オブジェクトとして返されるテーブル。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

ULCommand.CommandType は System.Data.CommandType.TableDirect に設定することが必要です。

ULCommand.IndexName が null 参照 (Visual Basic の Nothing) であると、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。

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 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 が無効です。

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「ExecuteTable メソッド」 79 ページ
- 「Connection プロパティ」 66 ページ
- 「Transaction プロパティ」 68 ページ
- 「CommandText プロパティ」 63 ページ
- 「IndexName プロパティ」 67 ページ
- CommandBehavior
- CommandBehavior.Default
- CommandBehavior.CloseConnection
- CommandBehavior.SchemaOnly
- 「CommandType プロパティ」 65 ページ
- CommandType.TableDirect
- 「ULTable クラス」 431 ページ

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

構文

Visual Basic

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

C#

```
public ULTable ExecuteTable(
```

```
    CommandBehavior cmdBehavior
);
```

パラメータ

- **cmdBehavior**　クエリの結果の記述と、接続への影響の記述の System.Data.CommandBehavior フラグのビット処理の組み合わせ。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返されるテーブル。

備考

ULCommand.CommandType は System.Data.CommandType.TableDirect に設定することが必要です。

ULCommand.IndexName が null 参照 (Visual Basic の Nothing) であると、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。

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 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 が無効です。

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「ExecuteTable メソッド」 78 ページ
- 「Connection プロパティ」 66 ページ
- 「Transaction プロパティ」 68 ページ
- 「CommandText プロパティ」 63 ページ
- 「IndexName プロパティ」 67 ページ
- CommandBehavior
- CommandBehavior.Default
- CommandBehavior.CloseConnection
- CommandBehavior.SchemaOnly
- 「CommandType プロパティ」 65 ページ
- CommandType.TableDirect

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
NotOverridable Public Sub Prepare() _
  Implements IDbCommand.Prepare
```

C#

```
public void Prepare();
```

備考

文を事前にコンパイルすると、パラメータ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。

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 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 が無効です。

実装

IDbCommand.Prepare

参照

- 「ULCommand クラス」 57 ページ
- 「ULCommand メンバ」 58 ページ
- 「Connection プロパティ」 66 ページ
- 「Transaction プロパティ」 68 ページ
- 「CommandText プロパティ」 63 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
NotInheritable Public Class ULCommandBuilder
  Inherits Component
```

C#

```
public sealed class ULCommandBuilder :
  Component
```

備考

ULDataAdapter 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 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 によって生成されます。

継承： System.ComponentModel.Component

実装： System.IDisposable

参照

- 「ULCommandBuilder メンバ」 82 ページ
- DataSet
- 「ULDataAdapter クラス」 156 ページ
- System.ComponentModel.Component
- IDisposable
- 「ULCommand クラス」 57 ページ
- 「ULConnection クラス」 90 ページ

ULCommandBuilder メンバ

パブリック・コンストラクタ

メンバ名	説明
ULCommandBuilder コンストラクタ	ULCommandBuilder オブジェクトを初期化します。
ULCommandBuilder コンストラクタ	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。

Copyright © 2007, iAnywhere Solutions, Inc.

パブリック・プロパティ

メンバ名	説明
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	スペースや予約語などの文字列が名前に含まれている Ultra Light データベース・オブジェクト (テーブルやカラム) 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	スペースや予約語などの文字列が名前に含まれている Ultra Light データベース・オブジェクト (テーブルやカラム) を指定するときに使用する末尾の文字または文字列を取得または設定します。

パブリック・メソッド

メンバ名	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。
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。
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。
RefreshSchema メソッド	この ULCommandBuilder に関連付けられたコマンドをクリアします。

ULCommandBuilder コンストラクタ

ULCommandBuilder オブジェクトを初期化します。

構文

Visual Basic

Overloads Public Sub **New()**

C#

public **ULCommandBuilder();**

参照

- 「ULCommandBuilder クラス」 82 ページ
- 「ULCommandBuilder メンバ」 82 ページ
- 「ULCommandBuilder コンストラクタ」 84 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULCommandBuilder コンストラクタ

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。

構文

Visual Basic

```
Overloads Public Sub New( _
  ByVal adapter As ULDataAdapter _
)
```

C#

```
public ULCommandBuilder(
  ULDataAdapter adapter
);
```

パラメータ

- **adapter**　ULDataAdapter オブジェクト。

参照

- 「ULCommandBuilder クラス」 82 ページ
- 「ULCommandBuilder メンバ」 82 ページ
- 「ULDataAdapter クラス」 156 ページ

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。

構文

Visual Basic

```
Public Property DataAdapter As ULDataAdapter
```

C#

```
public ULDataAdapter DataAdapter {get;set;}
```

プロパティ値

ULDataAdapter オブジェクト。

参照

- 「ULCommandBuilder クラス」 82 ページ
- 「ULCommandBuilder メンバ」 82 ページ
- 「ULDataAdapter クラス」 156 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

QuotePrefix プロパティ

スペースや予約語などの文字列が名前に含まれている Ultra Light データベース・オブジェクト (テーブルやカラム) 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。

構文

Visual Basic

Public Property **QuotePrefix** As String

C#

public string **QuotePrefix** {get;set;}

プロパティ値

使用する 1 つまたは複数の先頭文字。デフォルトは、空の文字列です。

参照

- 「ULCommandBuilder クラス」 82 ページ
- 「ULCommandBuilder メンバ」 82 ページ
- 「QuoteSuffix プロパティ」 85 ページ

QuoteSuffix プロパティ

スペースや予約語などの文字列が名前に含まれている Ultra Light データベース・オブジェクト (テーブルやカラム) を指定するときに使用する末尾の文字または文字列を取得または設定します。

構文

Visual Basic

Public Property **QuoteSuffix** As String

C#

public string **QuoteSuffix** {get;set;}

プロパティ値

使用する 1 つまたは複数の末尾文字。デフォルトは、空の文字列です。

参照

- 「ULCommandBuilder クラス」 82 ページ
- 「ULCommandBuilder メンバ」 82 ページ
- 「QuotePrefix プロパティ」 85 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Function **GetDeleteCommand()** As ULCommand

C#

public 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。

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、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、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 ULDataAdapter.Update または GetDeleteCommand メソッドを呼び出したときです。

例外

- InvalidOperationException - ULCommandBuilder.DataAdapter 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 に対してはサポートされていません。

  DeleteCommand に対する動的 SQL の生成は、キー・カラム情報を返さない SelectCommand に対してはサポートされていません。

参照

- 「ULCommandBuilder クラス」 82 ページ
- 「ULCommandBuilder メンバ」 82 ページ
- 「ULCommand クラス」 57 ページ
- 「RefreshSchema メソッド」 89 ページ
- 「SelectCommand プロパティ」 165 ページ
- 「Update メソッド」 176 ページ
- 「DataAdapter プロパティ」 84 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Function **GetInsertCommand()** As ULCommand

C#

public 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。

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、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、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 ULDataAdapter.Update または GetInsertCommand メソッドを呼び出したときです。

例外

- InvalidOperationException - ULCommandBuilder.DataAdapter 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 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 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 に対してはサポートされていません。

参照

- 「ULCommandBuilder クラス」 82 ページ
- 「ULCommandBuilder メンバ」 82 ページ
- 「ULCommand クラス」 57 ページ
- 「RefreshSchema メソッド」 89 ページ
- 「SelectCommand プロパティ」 165 ページ
- 「ULCommand クラス」 57 ページ
- 「DataAdapter プロパティ」 84 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Function **GetUpdateCommand()** As ULCommand

C#

public 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。

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、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 ULDataAdapter.Update または GetUpdateCommand メソッドを呼び出したときです。

例外

- InvalidOperationException - ULCommandBuilder.DataAdapter 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 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 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 に対してはサポートされていません。

  UpdateCommand に対する動的 SQL の生成は、キー・カラム情報を返さない SelectCommand に対してはサポートされていません。

参照

- 「ULCommandBuilder クラス」 82 ページ
- 「ULCommandBuilder メンバ」 82 ページ
- 「ULCommand クラス」 57 ページ
- 「RefreshSchema メソッド」 89 ページ
- 「SelectCommand プロパティ」 165 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「Update メソッド」 176 ページ
- 「DataAdapter プロパティ」 84 ページ

RefreshSchema メソッド

この ULCommandBuilder に関連付けられたコマンドをクリアします。

構文

Visual Basic

Public Sub **RefreshSchema()**

C#

public void **RefreshSchema();**

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 文を変更する場合、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、GetInsertCommand メソッド、GetDeleteCommand メソッド、および 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 ULDataAdapter.Update、GetInsertCommand、GetDeleteCommand、または GetUpdateCommand を呼び出したときです。

参照

- 「ULCommandBuilder クラス」 82 ページ
- 「ULCommandBuilder メンバ」 82 ページ
- 「RefreshSchema メソッド」 89 ページ
- 「SelectCommand プロパティ」 165 ページ
- 「GetInsertCommand メソッド」 87 ページ
- 「Update メソッド」 176 ページ
- 「GetInsertCommand メソッド」 87 ページ
- 「GetDeleteCommand メソッド」 86 ページ
- 「GetUpdateCommand メソッド」 88 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULConnection クラス

Ultra Light.NET データベースへの接続を示します。

構文

Visual Basic

NotInheritable Public Class **ULConnection**
  Inherits Component

C#

public sealed class **ULConnection** :
  Component

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 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 を使用して開きます。

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 を使用してアクセスできます。

実装： System.Data.IDbConnection、System.IDisposable

参照

- 「ULConnection メンバ」 90 ページ
- 「RuntimeType プロパティ」 183 ページ
- 「Open メソッド」 116 ページ
- 「Schema プロパティ」 100 ページ
- IDbConnection
- IDisposable

ULConnection メンバ

パブリックの静的フィールド (共有)

メンバ名	説明
INVALID_DATABASE_ID フィールド	**UL 拡張：** ULConnection.DatabaseID 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。

Copyright © 2007, iAnywhere Solutions, Inc.

パブリックの静的プロパティ (共有)

メンバ名	説明
DatabaseManager プロパティ	**UL 拡張：** シングルトンの ULDatabaseManager オブジェクトへのアクセスに使用します。

パブリック・コンストラクタ

メンバ名	説明
ULConnection コンストラクタ	ULConnection オブジェクトを初期化します。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。
ULConnection コンストラクタ	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。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。

パブリック・プロパティ

メンバ名	説明
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を指定します。接続文字列は、ULConnectionParms オブジェクトを使用して指定できます。
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。
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。
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。
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。

Copyright © 2007, iAnywhere Solutions, Inc.

パブリック・メソッド

メンバ名	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。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 または ULTransaction.Rollback で終了します。
BeginTransaction 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。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 または ULTransaction.Rollback で終了します。
ChangeDatabase メソッド	開いている ULConnection 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。
Close メソッド	データベース接続を閉じます。
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。
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。ULCommand のプロパティを使用して、その動作を制御できます。
ExecuteTable メソッド	**UL 拡張：** 直接の操作用に、ULTable のデータベース・テーブルを取り出します。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)。
ExecuteTable メソッド	**UL 拡張：** 直接の操作用に、ULTable のデータベース・テーブルを取り出します。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)。
ExecuteTable 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。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)。
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)。
GrantConnectTo メソッド	**UL 拡張：** 指定されたパスワードを持つユーザ ID に、Ultra Light データベースへのアクセスを許可します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	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。
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 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。
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。
Synchronize メソッド	**UL 拡張：** 指定されたリスナに送信されるプログレス・イベントとともに、現在の ULConnection.SyncParms を使用してデータベースを同期します。

パブリック・イベント

メンバ名	説明
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。

ULConnection コンストラクタ

ULConnection オブジェクトを初期化します。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。

構文

Visual Basic

Overloads Public Sub **New()**

C#

public **ULConnection();**

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 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 が設定されていないと、開くことができません。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「Open メソッド」 116 ページ
- 「ConnectionString プロパティ」 95 ページ

ULConnection コンストラクタ

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。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。

構文

Visual Basic

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

C#

```
public ULConnection(
  string  connectionString
);
```

パラメータ

- **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。

  パラメータのリストについては、ConnectionString プロパティを参照してください。

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 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。

例外

- ArgumentException - 指定された接続文字列が無効です。

Copyright © 2007, iAnywhere Solutions, Inc.

例

次のコードでは、Windows CE 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.DatabaseOnCE = "¥UltraLite¥MyDatabase.udb"
Dim conn As ULConnection = _
  New ULConnection( openParms.ToString() )
conn.Open()

// C#
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnCE = @"¥UltraLite¥MyDatabase.udb";
ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「Open メソッド」 116 ページ
- 「ULConnection コンストラクタ」 93 ページ
- 「RuntimeType プロパティ」 183 ページ
- 「ULConnectionParms クラス」 123 ページ
- 「ConnectionString プロパティ」 95 ページ

INVALID_DATABASE_ID フィールド

UL 拡張： ULConnection.DatabaseID 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。

構文

Visual Basic

Public Shared **INVALID_DATABASE_ID** As Long

C#

public const long **INVALID_DATABASE_ID**;

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「DatabaseID プロパティ」 98 ページ

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

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
NotOverridable Public Property ConnectionString As String _
  Implements IDbConnection.ConnectionString
```

C#

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

プロパティ値

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、この接続を開くためのパラメータ。デフォルトは、空の文字列 (無効な接続文字列) です。

備考

UL 拡張： Ultra Light.NET によって使用されるパラメータは 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。パラメータのリストについては、「Ultra Light の接続文字列パラメータのリファレンス」『Ultra Light - データベース管理とリファレンス』を参照してください。

パラメータ値には、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。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、ユーザ名 DBA のパスワードを変更するか、新しいユーザを作成して (GrantConnectTo を使用)、DBA ユーザを削除します (RevokeConnectFrom を使用)。

例外

- ArgumentException - 指定された接続文字列が無効です。
- InvalidOperationException - 接続が開いている間は、値を設定できません。

実装

IDbConnection.ConnectionString

例

次のコードでは、Windows CE 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.DatabaseOnCE = "¥UltraLite¥MyDatabase.udb"
Dim conn As ULConnection = New ULConnection
conn.ConnectionString = openParms.ToString()
conn.Open()

// C#
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnCE = @"¥UltraLite¥MyDatabase.udb";
ULConnection conn = new ULConnection();
conn.ConnectionString = openParms.ToString();
conn.Open();
```


Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「Open メソッド」 116 ページ
- 「ULConnectionParms クラス」 123 ページ
- 「GrantConnectTo メソッド」 115 ページ

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。

構文

Visual Basic

```
NotOverridable Public Readonly Property ConnectionTimeout As Integer _
  Implements IDbConnection.ConnectionTimeout
```

C#

```
public int ConnectionTimeout {get;}
```

プロパティ値

値は常にゼロです。

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。

実装

IDbConnection.ConnectionTimeout

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。

構文

Visual Basic

```
NotOverridable Public Readonly Property Database As String _
  Implements IDbConnection.Database
```

C#

```
public string  Database {get;}
```

プロパティ値

データベースの名前が含まれる文字列。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

Windows CE デバイスでは、ULConnection 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 は dbn、nt_file の順に接続文字列を参照します。

実装

IDbConnection.Database

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。

構文

Visual Basic

Public Property **DatabaseID** As Long

C#

public long **DatabaseID** {get;set;}

プロパティ値

現在のデータベースのデータベース ID の値。

備考

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。

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。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「GetDatabaseProperty メソッド」 194 ページ
- 「SetDatabaseOption メソッド」 199 ページ
- UInt32.MaxValue
- 「INVALID_DATABASE_ID フィールド」 95 ページ

DatabaseManager プロパティ

UL 拡張： シングルトンの ULDatabaseManager オブジェクトへのアクセスに使用します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Shared Readonly Property **DatabaseManager** As ULDatabaseManager

C#

public const ULDatabaseManager **DatabaseManager** {get;}

プロパティ値

シングルトンの ULDatabaseManager オブジェクトへの参照。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「ULDatabaseManager クラス」 182 ページ

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。

構文

Visual Basic

Public Readonly Property **GlobalAutoIncrementUsage** As Short

C#

public short **GlobalAutoIncrementUsage** {get;}

プロパティ値

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 ー 100 の範囲内の整数です。

備考

比率が 100% に近づいたら、ULConnection.DatabaseID を使用して、使用中のアプリケーションに新しいグローバル・データベース ID を設定してください。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「ULDatabaseManager クラス」 182 ページ
- 「DatabaseID プロパティ」 98 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

LastIdentity プロパティ

UL 拡張： 直前に使用した identity の値を返します。

構文

Visual Basic

Public Readonly Property **LastIdentity** As UInt64

C#

public ulong **LastIdentity** {get;}

プロパティ値

直前に使用した identity の値 (符号なし long)。

備考

直前に使用した identity の値。このプロパティは、次の SQL Anywhere 文と同義です。

```
SELECT @@identity
```

LastIdentity 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 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。

例外

- ULException クラス - SQL エラーが発生しました。

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。

構文

Visual Basic

Public Readonly Property **Schema** As ULDatabaseSchema

C#

public ULDatabaseSchema **Schema** {get;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

この接続が開かれているデータベースのスキーマを示す ULDatabaseSchema オブジェクトへの参照。

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。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「ULDatabaseSchema クラス」 190 ページ

State プロパティ

接続の現在のステータスを返します。

構文

Visual Basic

```
NotOverridable Public Readonly Property State As ConnectionState _
  Implements IDbConnection.State
```

C#

```
public ConnectionState State {get;}
```

プロパティ値

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。

実装

IDbConnection.State

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「StateChange イベント」 122 ページ
- ConnectionState.Open
- ConnectionState.Closed

SyncParms プロパティ

UL 拡張： この接続の同期設定を指定します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **SyncParms** As ULSyncParms

C#

public ULSyncParms **SyncParms** {get;}

プロパティ値

この接続によって同期に使用されるパラメータを参照する ULSyncParms オブジェクトへの参照。パラメータを修正すると、この接続で行われる次の同期に反映されます。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「Synchronize メソッド」 119 ページ
- 「SyncResult プロパティ」 102 ページ
- 「ULSyncParms クラス」 396 ページ

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。

構文

Visual Basic

Public Readonly Property **SyncResult** As ULSyncResult

C#

public ULSyncResult **SyncResult** {get;}

プロパティ値

この接続の最後の同期結果を参照する ULSyncResult オブジェクトへの参照。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「Synchronize メソッド」 119 ページ
- 「SyncParms プロパティ」 101 ページ
- 「SyncResult プロパティ」 102 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

BeginTransaction メソッド

トランザクション・オブジェクトを返します。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 または ULTransaction.Rollback で終了します。

構文

Visual Basic

Overloads Public Function **BeginTransaction()** As ULTransaction

C#

public ULTransaction **BeginTransaction();**

戻り値

新しいトランザクションを示す ULTransaction オブジェクト。

備考

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 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 を呼び出すまで、接続はオートコミット・モードに戻ります。

これは、System.Data.IDbConnection.BeginTransaction() が厳密に型指定されたものです。

例外

- ULException クラス - 接続が閉じています。
- InvalidOperationException - ULConnection は並列トランザクションをサポートしていません。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「BeginTransaction メソッド」 104 ページ
- 「Commit メソッド」 470 ページ
- 「Rollback メソッド」 471 ページ
- 「Transaction プロパティ」 68 ページ
- 「CreateCommand メソッド」 108 ページ
- IDbConnection.BeginTransaction

Copyright © 2007, iAnywhere Solutions, Inc.

BeginTransaction 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。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 または ULTransaction.Rollback で終了します。

構文

Visual Basic

```
Overloads Public Function BeginTransaction( _
  ByVal isolationLevel As IsolationLevel _
) As ULTransaction
```

C#

```
public ULTransaction BeginTransaction(
  IsolationLevel isolationLevel
);
```

パラメータ

- **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 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示す ULTransaction オブジェクト。

備考

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 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 を呼び出すまで、接続はオートコミット・モードに戻ります。

これは、System.Data.IDbConnection.BeginTransaction(System.Data.IsolationLevel) が厳密に型指定されたものです。

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 は並列トランザクションをサポートしていません。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「ULTransaction クラス」 468 ページ
- 「BeginTransaction メソッド」 103 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「Commit メソッド」 470 ページ
- 「Rollback メソッド」 471 ページ
- 「Transaction プロパティ」 68 ページ
- 「CreateCommand メソッド」 108 ページ
- IDbConnection.BeginTransaction
- IsolationLevel.ReadUncommitted

ChangeDatabase メソッド

開いている ULConnection の現在のデータベースを変更します。

構文

Visual Basic

```
NotOverridable Public Sub ChangeDatabase( _
  ByVal connectionString As String _
) _
  Implements IDbConnection.ChangeDatabase
```

C#

```
public void ChangeDatabase(
  string  connectionString
);
```

パラメータ

- **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 または dbf ではなく、完全な接続文字列です。

実装

IDbConnection.ChangeDatabase

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「ConnectionString プロパティ」 95 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Sub ChangeEncryptionKey( _
  ByVal newKey As String _
)
```

C#

```
public void ChangeEncryptionKey(
  string  newKey
);
```

パラメータ

- **newkey**　データベースの新しい暗号化キー。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「EncryptionKey プロパティ」 132 ページ

Close メソッド

データベース接続を閉じます。

構文

Visual Basic

```
NotOverridable Public Sub Close() _
  Implements IDbConnection.Close
```

C#

```
public void Close();
```

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、Close を複数回呼び出すことができます。

例外

- ULException クラス - SQL エラーが発生しました。

Copyright © 2007, iAnywhere Solutions, Inc.

実装

IDbConnection.Close

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。

構文

Visual Basic

```
Overloads Public Function CountUploadRows( _
  ByVal mask As Integer, _
  ByVal threshold As UInt32 _
) As UInt32
```

C#

```
public uint CountUploadRows(
  int mask,
  uint threshold
);
```

パラメータ

- **mask**　ローのチェック対象となるパブリケーションのセット。詳細については、ULPublicationSchema クラスを参照してください。
- **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。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「CountUploadRows メソッド」 107 ページ

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Overloads Public Function CountUploadRows( _
  ByVal mask As Integer, _
  ByVal threshold As Long _
) As Long
```

C#

```
public long CountUploadRows(
  int mask,
  long threshold
);
```

パラメータ

- **mask**　ローのチェック対象となるパブリケーションのセット。詳細については、ULPublicationSchema クラスを参照してください。
- **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。スレッショルド値は、[0,0x0fffffff] の範囲内であることが必要です。

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。

備考

このメソッドは、System.UInt32 型をネイティブでサポートしていない言語を対象として用意されています。この型がアプリケーションでサポートされている場合は、このメソッドの別のフォームを使用します。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「CountUploadRows メソッド」 107 ページ

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。ULCommand のプロパティを使用して、その動作を制御できます。

構文

Visual Basic

```
Public Function CreateCommand() As ULCommand
```


Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public ULCommand CreateCommand();
```

戻り値

新しい ULCommand オブジェクト。

備考

ULCommand.CommandText 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 が厳密に型指定されたものです。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「ULCommand クラス」 57 ページ
- 「CommandText プロパティ」 63 ページ
- IDbConnection.CreateCommand

ExecuteTable メソッド

UL 拡張： 直接の操作用に、ULTable のデータベース・テーブルを取り出します。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)。

構文

Visual Basic

```
Overloads Public Function ExecuteTable( _
  ByVal tableName As String _
) As ULTable
```

C#

```
public ULTable ExecuteTable(
  string  tableName
);
```

パラメータ

- **tableName**　開くテーブルの名前。

戻り値

ULTable オブジェクトとして返されるテーブル。

備考

このメソッドは、ULCommand インスタンスを必要としない 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() と iAnywhere.UltraLite.Table.Open() が置き換えられます)。

Copyright © 2007, iAnywhere Solutions, Inc.

例外

- ULException クラス - SQL エラーが発生しました。
- InvalidOperationException - *tableName* が無効です。

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()

// 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();
// }
```

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「ExecuteTable メソッド」 110 ページ
- 「ExecuteTable メソッド」 112 ページ
- 「ULTable クラス」 431 ページ
- 「ExecuteTable メソッド」 78 ページ
- 「ULCommand クラス」 57 ページ

ExecuteTable メソッド

UL 拡張： 直接の操作用に、ULTable のデータベース・テーブルを取り出します。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)。

構文

Visual Basic

```
Overloads Public Function ExecuteTable( _
   ByVal tableName As String, _
   ByVal indexName As String _
) As ULTable
```

C#

Copyright © 2007, iAnywhere Solutions, Inc.

```
public ULTable ExecuteTable(
  string  tableName,
  string  indexName
);
```

パラメータ

- **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返されるテーブル。

備考

このメソッドは、ULCommand インスタンスを必要としない 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() と iAnywhere.UltraLite.Table.Open() が置き換えられます)。

例外

- ULException クラス - SQL エラーが発生しました。
- InvalidOperationException - *tableName* が無効です。

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 インスタンス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()

// 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();
// }
```

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「ExecuteTable メソッド」 109 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ExecuteTable メソッド」 112 ページ
- 「ULTable クラス」 431 ページ
- 「ExecuteTable メソッド」 78 ページ
- 「ULCommand クラス」 57 ページ

ExecuteTable 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。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)。

構文

Visual Basic

```
Overloads Public Function ExecuteTable( _
  ByVal tableName As String, _
  ByVal indexName As String, _
  ByVal cmdBehavior As CommandBehavior _
) As ULTable
```

C#

```
public ULTable ExecuteTable(
  string  tableName,
  string  indexName,
  CommandBehavior cmdBehavior
);
```

パラメータ

- **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 フラグのビット処理の組み合わせ。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返されるテーブル。

備考

このメソッドは、ULCommand インスタンスを必要としない ULCommand.ExecuteTable (System.Data.CommandBehavior) 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() と iAnywhere.UltraLite.Table.Open() が置き換えられます)。

例外

- ULException クラス - SQL エラーが発生しました。

Copyright © 2007, iAnywhere Solutions, Inc.

- InvalidOperationException - *tableName* が無効です。

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 インスタンス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
' 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()

// C#
ULTable t = conn.ExecuteTable(
    "MyTable", "MyIndex", CommandBehavior.Default
  );
// 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(CommandBehavior.Default);
// }
```

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「ExecuteTable メソッド」 109 ページ
- 「ExecuteTable メソッド」 110 ページ
- 「ULTable クラス」 431 ページ
- 「ExecuteTable メソッド」 79 ページ
- 「ULCommand クラス」 57 ページ
- CommandBehavior
- CommandBehavior.Default
- CommandBehavior.CloseConnection
- CommandBehavior.SchemaOnly

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。

構文

Visual Basic

Public Function **GetLastDownloadTime(_**

```
  ByVal mask As Integer _
) As Date
```

C#

```
public DateTime GetLastDownloadTime(
  int mask
);
```

パラメータ

- **mask**　チェックするパブリケーションのマスク。詳細については、ULPublicationSchema クラスを参照してください。

戻り値

最後のダウンロードのタイムスタンプ。

備考

パラメータ *mask* は、単一のパブリケーションを参照するか、データベース全体の最後のダウンロード時刻の特殊定数 ULPublicationSchema.SYNC_ALL_DB であることが必要です。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「ResetLastDownloadTime メソッド」 116 ページ
- 「SYNC_ALL_DB フィールド」 322 ページ

GetNewUUID メソッド

UL 拡張： 新しい UUID を生成します (System.Guid)。

構文

Visual Basic

```
Public Function GetNewUUID() As Guid
```

C#

```
public Guid GetNewUUID();
```

戻り値

System.Guid として返される新しい UUID。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- Guid

GrantConnectTo メソッド

UL 拡張： 指定されたパスワードを持つユーザ ID に、Ultra Light データベースへのアクセスを許可します。

構文

Visual Basic

```
Public Sub GrantConnectTo( _
  ByVal uid As String, _
  ByVal pwd As String _
)
```

C#

```
public void GrantConnectTo(
  string  uid,
  string  pwd
);
```

パラメータ

- **uid**　データベースへのアクセス権を受け取るユーザ ID。ユーザ ID の最大長は 16 文字です。
- **pwd**　ユーザ ID に関連付けられるパスワード。最大長は 16 文字です。

備考

既存のユーザ ID が指定されていれば、この関数を使用してそのユーザ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認証が有効になっていた場合だけです。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「UserID プロパティ」 133 ページ
- 「Password プロパティ」 133 ページ
- 「ConnectionString プロパティ」 95 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
NotOverridable Public Sub Open() _
  Implements IDbConnection.Open
```

C#

```
public void Open();
```

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。

例外

- 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。
- InvalidOperationException - 接続がすでに開かれているか、接続文字列が ULConnection.ConnectionString で指定されていません。

実装

IDbConnection.Open

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「ConnectionString プロパティ」 95 ページ
- 「State プロパティ」 101 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。

構文

Visual Basic

```
Public Sub ResetLastDownloadTime( _
  ByVal mask As Integer _
)
```

C#

```
public void ResetLastDownloadTime(
  int mask
);
```


Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **mask**　リセットするパブリケーションのセット。詳細については、ULPublicationSchema クラスを参照してください。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「GetLastDownloadTime メソッド」 113 ページ

RevokeConnectFrom メソッド

UL 拡張： 指定されたユーザ 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。

構文

Visual Basic

```
Public Sub RevokeConnectFrom( _
  ByVal uid As String _
)
```

C#

```
public void RevokeConnectFrom(
  string  uid
);
```

パラメータ

- **uid**　データベースへのアクセス権を取り消すユーザ ID。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「GrantConnectTo メソッド」 115 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Sub **RollbackPartialDownload()**

C#

public void **RollbackPartialDownload();**

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「KeepPartialDownload プロパティ」 400 ページ
- 「ResumePartialDownload プロパティ」 404 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。

構文

Visual Basic

Public Sub **StartSynchronizationDelete()**

C#

public void **StartSynchronizationDelete();**

備考

この関数が呼び出されると、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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「StopSynchronizationDelete メソッド」 119 ページ
- 「Truncate メソッド」 451 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Sub **StopSynchronizationDelete()**

C#

public void **StopSynchronizationDelete();**

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「StartSynchronizationDelete メソッド」 118 ページ

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。

構文

Visual Basic

Overloads Public Sub **Synchronize()**

C#

public void **Synchronize();**

備考

結果の詳細なステータスは、この接続の ULConnection.SyncResult でレポートされます。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「Synchronize メソッド」 120 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「SyncParms プロパティ」 101 ページ
- 「SyncResult プロパティ」 102 ページ

Synchronize メソッド

UL 拡張： 指定されたリスナに送信されるプログレス・イベントとともに、現在の ULConnection.SyncParms を使用してデータベースを同期します。

構文

Visual Basic

```
Overloads Public Sub Synchronize( _
  ByVal listener As ULSyncProgressListener _
)
```

C#

```
public void Synchronize(
  ULSyncProgressListener listener
);
```

パラメータ

- **listener** 同期プログレス・イベントを受信するオブジェクト。

備考

リスナに送信された最後のイベントのステータスは、ULSyncProgressState.STATE_LAST です。

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 で報告されます。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「ULSyncProgressListener インタフェース」 421 ページ
- 「Synchronize メソッド」 119 ページ
- 「SyncParms プロパティ」 101 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「ULException クラス」 249 ページ
- 「SyncResult プロパティ」 102 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Event **InfoMessage** As ULInfoMessageEventHandler

C#

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 委任を作成し、このイベントにアタッチする必要があります。

例

次のコードでは、情報メッセージのイベント・ハンドラが定義されます。

```
' Visual Basic
Private Sub MyInfoMessageHandler( _
      obj As Object, args As ULInfoMessageEventArgs _
    )
  System.Console.WriteLine( _
      "InfoMesageHandler: " + args.NativeError + ", " _
      + args.Message _
    )
End Sub

// C#
private void MyInfoMessageHandler(
      object obj, ULInfoMessageEventArgs args
    )
{
  System.Console.WriteLine(
      "InfoMesageHandler: " + args.NativeError + ", "
      + args.Message
    );
}
```

次のコードでは、MyInfoMessageHandler が conn という接続に追加されます。

```
' Visual Basic
AddHandler conn.InfoMessage, AddressOf MyInfoMessageHandler

// C#
conn.InfoMessage +=
  new ULInfoMessageEventHandler(MyInfoMessageHandler);
```

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- 「ULInfoMessageEventHandler 委任」 285 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Event **StateChange** As StateChangeEventHandler

C#

public event 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 委任を作成し、このイベントにアタッチする必要があります。

例

次のコードでは、ステータス変更のイベント・ハンドラが定義されます。

```
' Visual Basic
Private Sub MyStateHandler( _
      obj As Object, args As StateChangeEventArgs _
    )
  System.Console.WriteLine( _
      "StateHandler: " + args.OriginalState + " to " _
      + args.CurrentState _
    )
End Sub

// 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

// C#
conn.StateChange += new StateChangeEventHandler(MyStateHandler);
```

参照

- 「ULConnection クラス」 90 ページ
- 「ULConnection メンバ」 90 ページ
- StateChangeEventHandler

Copyright © 2007, iAnywhere Solutions, Inc.

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。頻繁に使用する接続パラメータは、ULConnectionParms オブジェクトの個々のプロパティです。

構文

Visual Basic

```
Public Class ULConnectionParms
  Inherits Component
```

C#

```
public class ULConnectionParms :
  Component
```

備考

ULConnectionParms オブジェクトは、接続を開くパラメータ (ULConnection.Open)、またはデータベースを削除するパラメータ (ULDatabaseManager.DropDatabase) 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、接続パラメータをすべて指定したら、ULConnectionParms.ToString 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) コンストラクタで新しい ULConnection を作成したり、既存の ULConnection オブジェクトの ULConnection.ConnectionString 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 CE 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.DatabaseOnCE = "¥UltraLite¥mydb1.udb"
dbName.DatabaseOnDesktop = "somedir¥mydb2.udb"

// C#
ULConnectionParms dbName = new ULConnectionParms();
dbName.DatabaseOnCE = "¥¥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 CE 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 © 2007, iAnywhere Solutions, Inc.

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。詳細については、AdditionalParms プロパティを参照してください。

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 や ULConnectionParms.Password、データベースの ULConnectionParms.EncryptionKey、接続の ULConnectionParms.CacheSize 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 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 DBA で作成されます。このユーザの最初のパスワードは sql です。デフォルトでは、ユーザ 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を無効にするには、ULConnection.RevokeConnectFrom を使用します。ユーザを追加したりユーザのパスワードを変更するには、ULConnection.GrantConnectTo 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 を使用します。

詳細については、「Ultra Light の接続文字列パラメータのリファレンス」 『Ultra Light - データベース管理とリファレンス』を参照してください。

参照

- 「ULConnectionParms メンバ」 125 ページ
- 「Open メソッド」 116 ページ
- 「DropDatabase メソッド」 185 ページ
- 「ToString メソッド」 134 ページ
- 「ULConnection クラス」 90 ページ
- 「ULConnection コンストラクタ」 94 ページ
- 「ConnectionString プロパティ」 95 ページ
- 「UserID プロパティ」 133 ページ
- 「Password プロパティ」 133 ページ
- 「EncryptionKey プロパティ」 132 ページ
- 「CacheSize プロパティ」 130 ページ
- 「ConnectionName プロパティ」 131 ページ
- 「RevokeConnectFrom メソッド」 117 ページ
- 「GrantConnectTo メソッド」 115 ページ
- 「ChangeEncryptionKey メソッド」 105 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULConnectionParms メンバ

パブリック・コンストラクタ

メンバ名	説明
ULConnectionParms コンストラクタ	ULConnectionParms インスタンスを、そのデフォルト値で初期化します。

パブリック・プロパティ

メンバ名	説明
AdditionalParms プロパティ	「名前=値」の組み合わせをセミコロンで区切ったリストで、追加のパラメータを指定します。ここで指定されるのは、あまり一般的に使用されないパラメータです。
CacheSize プロパティ	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。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。
DatabaseOnCE プロパティ	Windows CE 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。
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。
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。
Password プロパティ	認証済みユーザのパスワードを指定します。
UserID プロパティ	データベースで認証されるユーザを指定します。

パブリック・メソッド

メンバ名	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。

パブリック・イベント

メンバ名	説明
UnusedEvent イベント	使われていません。

ULConnectionParms コンストラクタ

ULConnectionParms インスタンスを、そのデフォルト値で初期化します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Sub **New()**

C#

public **ULConnectionParms();**

AdditionalParms プロパティ

「名前=値」の組み合わせをセミコロンで区切ったリストで、追加のパラメータを指定します。ここで指定されるのは、あまり一般的に使用されないパラメータです。

構文

Visual Basic

Public Property **AdditionalParms** As String

C#

public string **AdditionalParms** {get;set;}

プロパティ値

「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。keyword=value リストの値は、ULConnection.ConnectionString のルールに従う必要があります。デフォルト値は null 参照 (Visual Basic の Nothing) です。

備考

ページ・サイズや予約サイズのパラメータの値は、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。

追加のパラメータを次に示します。

Copyright © 2007, iAnywhere Solutions, Inc.

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 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 (DatabaseOnCE または DatabaseOnDesktop) 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。 このパラメータが使用されるのは、ULConnection.Open を使用して接続を開いたときだけです。

Copyright © 2007, iAnywhere Solutions, Inc.

キーワード	説明
ordered_table_scans	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。 リリース 10.0.1 では、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、ordered_table_scans パラメータを 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。 ordered_table_scans 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。 次のパラメータ文字列を使用すると、Ultra Light は以前のリリースと同じように動作します。 createParms.**AdditionalParms = "ordered_table_scans=yes"** デフォルトは 0 (false、off、no) です。 このパラメータが使用されるのは、ULConnection.Open を使用して接続を開いたときだけです。 詳細については、「Ultra Light の接続文字列パラメータのリファレンス」『Ultra Light - データベース管理とリファレンス』を参照してください。

Copyright © 2007, iAnywhere Solutions, Inc.

キーワード	説明
reserve_size	Ultra Light 継続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、永続ストレージ・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、継続保管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ヘッドとデータの圧縮を考慮してください。テスト・データを入れてデータベースを実行し、継続保管ファイルのサイズを確認することをおすすめします。 reserve_size パラメータは、起動時に継続保管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。 次のパラメータ文字列では、起動時に継続保管ファイルのサイズが最低 2 MB 確保されます。 createParms.**AdditionalParms = "reserve_size=2m"** このパラメータが使用されるのは、ULConnection.Open を使用して接続を開いたときだけです。
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。 Ultra Light.NET で Ultra Light エンジンを使用する方法の詳細については、RuntimeType プロパティを参照してください。

例外

- ArgumentException - 値に無効な接続文字列が含まれていました。

参照

- 「ULConnectionParms クラス」 123 ページ
- 「ULConnectionParms メンバ」 125 ページ
- 「ConnectionString プロパティ」 95 ページ
- 「DatabaseOnCE プロパティ」 131 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「DatabaseOnDesktop プロパティ」 132 ページ
- 「Open メソッド」 116 ページ

CacheSize プロパティ

キャッシュのサイズを指定します。

構文

Visual Basic

Public Property **CacheSize** As String

C#

public string **CacheSize** {get;set;}

プロパティ値

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。

備考

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。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

ConnectionName プロパティ

接続の名前を指定します。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。

構文

Visual Basic

Public Property **ConnectionName** As String

C#

public string **ConnectionName** {get;set;}

プロパティ値

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。

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。

DatabaseOnCE プロパティ

Windows CE 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。

構文

Visual Basic

Public Property **DatabaseOnCE** As String

C#

public string **DatabaseOnCE** {get;set;}

プロパティ値

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。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Property **DatabaseOnDesktop** As String

C#

public string **DatabaseOnDesktop** {get;set;}

プロパティ値

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。

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。

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。

構文

Visual Basic

Public Property **EncryptionKey** As String

C#

public string **EncryptionKey** {get;set;}

プロパティ値

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。

備考

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。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

参照

- 「ULConnectionParms クラス」 123 ページ
- 「ULConnectionParms メンバ」 125 ページ
- 「ChangeEncryptionKey メソッド」 105 ページ

Password プロパティ

認証済みユーザのパスワードを指定します。

構文

Visual Basic

Public Property **Password** As String

C#

public string **Password** {get;set;}

プロパティ値

データベースのユーザ 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。

備考

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 ID DBA のパスワードは SQL に設定されます。

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。

参照

- 「ULConnectionParms クラス」 123 ページ
- 「ULConnectionParms メンバ」 125 ページ
- 「UserID プロパティ」 133 ページ

UserID プロパティ

データベースで認証されるユーザを指定します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **UserID** As String

C#

public string **UserID** {get;set;}

プロパティ値

データベースのユーザ 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。

備考

ユーザ ID の大文字と小文字は区別されません。

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で作成されます。

ユーザ ID とパスワードを両方とも入力しないと、ユーザ名 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、ユーザ名 DBA のパスワードを変更するか、新しいユーザを作成して (ULConnection.GrantConnectTo を使用)、DBA ユーザを削除します (ULConnection.RevokeConnectFrom を使用)。

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。

参照

- 「ULConnectionParms クラス」 123 ページ
- 「ULConnectionParms メンバ」 125 ページ
- 「Password プロパティ」 133 ページ
- 「GrantConnectTo メソッド」 115 ページ
- 「RevokeConnectFrom メソッド」 117 ページ

ToString メソッド

このインスタンスの文字列表現を返します。

構文

Visual Basic

Overrides Public Function **ToString()** As String

C#

public override string **ToString();**

 Copyright © 2007, iAnywhere Solutions, Inc.

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。

UnusedEvent イベント

使われていません。

構文

Visual Basic

Public Event **UnusedEvent** As ULConnectionParms.UnusedEventHandler

C#

public event ULConnectionParms.UnusedEventHandler **UnusedEvent**;

備考

この public Event は、Visual Basic .NET プロジェクトでこのクラスを組み込む場合の Visual Studio .NET のバグを修正するために用意されています。実用的な用途はありません。

Copyright © 2007, iAnywhere Solutions, Inc.

ULConnectionParms.UnusedEventHandler 委任

UL 拡張： 使われていません。

構文

Visual Basic

```
Delegate Sub ULConnectionParms.UnusedEventHandler( _
  ByVal sender As Object, _
  ByVal args As System.EventArgs _
)
```

C#

```
delegate void ULConnectionParms.UnusedEventHandler(
  object sender,
  System.EventArgs args
);
```

パラメータ

- **sender**　送信側のオブジェクト。
- **args**　イベント引数。

備考

この public Delegate は、Visual Basic .NET プロジェクトでこのクラスを組み込む場合の Visual Studio .NET のバグを修正するために用意されています。実用的な用途はありません。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Class **ULCreateParms**

C#

public class **ULCreateParms**

備考

ULCreateParms オブジェクトは、データベースを作成するときのパラメータを使用するために使用します (ULDatabaseManager.CreateDatabase(string,byte[],string))。

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、作成パラメータをすべて指定したら、ULCreateParms.ToString を使用して作成パラメータ文字列を作成します。結果として返される文字列は ULDatabaseManager.CreateDatabase(string,byte[],string) メソッドの createParms パラメータとして使用できます。

詳細については、「Ultra Light の接続文字列パラメータのリファレンス」 『Ultra Light - データベース管理とリファレンス』を参照してください。

参照

- 「ULCreateParms メンバ」 137 ページ
- 「GetDatabaseProperty メソッド」 194 ページ
- 「CreateDatabase メソッド」 183 ページ
- 「ToString メソッド」 146 ページ

ULCreateParms メンバ

パブリック・コンストラクタ

メンバ名	説明
ULCreateParms コンストラクタ	ULCreateParms インスタンスを、そのデフォルト値で初期化します。

パブリック・プロパティ

メンバ名	説明
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

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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
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
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
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)
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
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
Obfuscate プロパティ	新しいデータベースで難読化 (単純暗号化) を使用するかどうか
PageSize プロパティ	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)
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
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 PRECISION にトランケートされるときの小数点以下の最大桁数
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
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
TimestampIncrement プロパティ	2 つのユニークなタイムスタンプ間のナノ秒 (1,000,000 分の 1 秒) 単位の最小差
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)

パブリック・メソッド

メンバ名	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。

ULCreateParms コンストラクタ

ULCreateParms インスタンスを、そのデフォルト値で初期化します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Sub **New()**

C#

public **ULCreateParms();**

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

構文

Visual Basic

Public Property **CaseSensitive** As Boolean

C#

public bool **CaseSensitive** {get;set;}

プロパティ値

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。

備考

CaseSensitive 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 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

構文

Visual Basic

Public Property **ChecksumLevel** As Integer

C#

public int **ChecksumLevel** {get;set;}

プロパティ値

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。

Copyright © 2007, iAnywhere Solutions, Inc.

例外

- ArgumentException - 値が無効です。

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

構文

Visual Basic

Public Property **DateFormat** As String

C#

public string **DateFormat** {get;set;}

プロパティ値

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。

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。

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

構文

Visual Basic

Public Property **DateOrder** As ULDateOrder

C#

public ULDateOrder **DateOrder** {get;set;}

プロパティ値

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

- 「ULCreateParms クラス」 137 ページ
- 「ULCreateParms メンバ」 137 ページ
- 「ULDateOrder enumeration」 244 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

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)

構文

Visual Basic

Public Property **FIPS** As Boolean

C#

public bool **FIPS** {get;set;}

プロパティ値

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。

備考

新しいデータベースが作成されたときは、接続パラメータ EncryptionKey の値を指定して暗号化を有効にする必要があります。FIPS を true に設定している場合で、暗号化キーを指定しないと、ULDatabaseManager.CreateDatabase(string,byte[],string) 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

- 「ULCreateParms クラス」 137 ページ
- 「ULCreateParms メンバ」 137 ページ
- 「EncryptionKey プロパティ」 132 ページ
- 「CreateDatabase メソッド」 183 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

構文

Visual Basic

Public Property **MaxHashSize** As Integer

C#

public int **MaxHashSize** {get;set;}

プロパティ値

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。

例外

- ArgumentException - 値が無効です。

Copyright © 2007, iAnywhere Solutions, Inc.

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

構文

Visual Basic

Public Property **NearestCentury** As Integer

C#

public int **NearestCentury** {get;set;}

プロパティ値

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。

例外

- ArgumentException - 値が無効です。

Obfuscate プロパティ

新しいデータベースで難読化 (単純暗号化) を使用するかどうか

構文

Visual Basic

Public Property **Obfuscate** As Boolean

C#

public bool **Obfuscate** {get;set;}

プロパティ値

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。

備考

FIPS 暗号化が有効になっている場合 (ULCreateParms.FIPS) 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接続パラメータ EncryptionKey (DBKEY) に値を指定すると、暗号化キーは無視されます。

参照

- 「ULCreateParms クラス」 137 ページ
- 「ULCreateParms メンバ」 137 ページ
- 「FIPS プロパティ」 141 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

PageSize プロパティ

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)

構文

Visual Basic

Public Property **PageSize** As Integer

C#

public int **PageSize** {get;set;}

プロパティ値

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。

例外

- ArgumentException - 値が無効です。

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

構文

Visual Basic

Public Property **Precision** As Integer

C#

public int **Precision** {get;set;}

プロパティ値

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。

例外

- ArgumentException - 値が無効です。

参照

- 「ULCreateParms クラス」 137 ページ
- 「ULCreateParms メンバ」 137 ページ
- 「Scale プロパティ」 144 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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 PRECISION にトランケートされるときの小数点以下の最大桁数

構文

Visual Basic

Public Property **Scale** As Integer

C#

public int **Scale** {get;set;}

プロパティ値

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。

備考

Scale は、Precision よりも小さいか等しいことが必要です。Scale が Precision よりも大きいと、データベースの作成時にエラーが発生します。

例外

- ArgumentException - 値が無効です。

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

構文

Visual Basic

Public Property **TimeFormat** As String

C#

public string **TimeFormat** {get;set;}

プロパティ値

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。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

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

構文

Visual Basic

Public Property **TimestampFormat** As String

C#

public string **TimestampFormat** {get;set;}

プロパティ値

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。

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。

TimestampIncrement プロパティ

2 つのユニークなタイムスタンプ間のナノ秒 (1,000,000 分の 1 秒) 単位の最小差

構文

Visual Basic

Public Property **TimestampIncrement** As Integer

C#

public int **TimestampIncrement** {get;set;}

プロパティ値

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。

例外

- ArgumentException - 値が無効です。

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)

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **UTF8Encoding** As Boolean

C#

public bool **UTF8Encoding** {get;set;}

プロパティ値

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。

備考

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。

Symbian OS デバイスで使用するデータベースの場合、UTF8Encoding を true に設定する必要があります。

Palm OS デバイスで使用するデータベースの場合、UTF8Encoding を false のままにする必要があります。

ToString メソッド

このインスタンスの文字列表現を返します。

構文

Visual Basic

Overrides Public Function **ToString()** As String

C#

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。

Copyright © 2007, iAnywhere Solutions, Inc.

ULCursorSchema クラス

UL 拡張： Ultra Light.NET カーソルのスキーマを示します。

構文

Visual Basic

MustInherit Public Class **ULCursorSchema**

C#

public abstract class **ULCursorSchema**

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。

iAnywhere.UltraLite ネームスペースから移行するユーザへの注意： 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

- 「ULCursorSchema メンバ」 147 ページ
- 「ULTableSchema クラス」 453 ページ
- 「ULResultSetSchema クラス」 351 ページ

ULCursorSchema メンバ

パブリック・プロパティ

メンバ名	説明
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。

パブリック・メソッド

メンバ名	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 (SQL NUMERIC 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 (SQL NUMERIC 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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
GetColumnSize メソッド	カラムがサイズ指定されたカラム (SQL BINARY 型または CHAR 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 のカラムのスキーマが記述された System.Data.DataTable を返します。

保護メソッド

メンバ名	説明
Finalize メソッド	ULCursorSchema によって使用されるアンマネージ・リソースを解放します。

ColumnCount プロパティ

このカーソル内のカラム数を返します。

構文

Visual Basic

Public Readonly Property **ColumnCount** As Short

C#

public short **ColumnCount** {get;}

プロパティ値

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。

備考

カラム ID の範囲は、0 ー 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。

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 © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **IsOpen** As Boolean

C#

public bool **IsOpen** {get;}

プロパティ値

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。

構文

Visual Basic

Public Readonly Property **Name** As String

C#

public string **Name** {get;}

プロパティ値

文字列として返されるカーソル名。

Finalize メソッド

ULCursorSchema によって使用されるアンマネージ・リソースを解放します。

構文

Visual Basic

Overrides Protected Sub **Finalize()**

C#

protected override void **Finalize();**

GetColumnID メソッド

指定されたカラムのカラム ID を返します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Function GetColumnID( _
   ByVal name As String _
) As Short
```

C#

```
public short GetColumnID(
   string  name
);
```

パラメータ

- **name**　カラムの名前。

戻り値

指定されたカラムのカラム ID。

備考

カラム ID の範囲は、0 ー 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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULCursorSchema クラス」 147 ページ
- 「ULCursorSchema メンバ」 147 ページ
- 「ColumnCount プロパティ」 148 ページ
- 「ColumnCount プロパティ」 148 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。

構文

Visual Basic

```
Public Function GetColumnName( _
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
  ByVal columnID As Integer _
) As String
```

C#

```
public string  GetColumnName(
  int columnID
);
```

パラメータ

- **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。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULCursorSchema クラス」 147 ページ
- 「ULCursorSchema メンバ」 147 ページ
- 「ColumnCount プロパティ」 148 ページ
- 「ColumnCount プロパティ」 148 ページ

GetColumnPrecision メソッド

カラムが数値カラム (SQL NUMERIC 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。

構文

Visual Basic

```
Public Function GetColumnPrecision( _
  ByVal columnID As Integer _
) As Integer
```

C#

Copyright © 2007, iAnywhere Solutions, Inc.

```
public int GetColumnPrecision(
  int columnID
);
```

パラメータ

- **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 クラス」 147 ページ
- 「ULCursorSchema メンバ」 147 ページ
- 「GetColumnULDbType メソッド」 154 ページ
- 「ColumnCount プロパティ」 148 ページ

GetColumnScale メソッド

カラムが数値カラム (SQL NUMERIC 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。

構文

Visual Basic

```
Public Function GetColumnScale( _
  ByVal columnID As Integer _
) As Integer
```

C#

```
public int GetColumnScale(
  int columnID
);
```

パラメータ

- **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 クラス」 147 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULCursorSchema メンバ」 147 ページ
- 「GetColumnULDbType メソッド」 154 ページ
- 「ColumnCount プロパティ」 148 ページ

GetColumnSize メソッド

カラムがサイズ指定されたカラム (SQL BINARY 型または CHAR 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。

構文

Visual Basic

```
Public Function GetColumnSize( _
   ByVal columnID As Integer _
) As Integer
```

C#

```
public int GetColumnSize(
   int columnID
);
```

パラメータ

- **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 クラス」 147 ページ
- 「ULCursorSchema メンバ」 147 ページ
- 「GetColumnULDbType メソッド」 154 ページ
- 「ColumnCount プロパティ」 148 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。

構文

Visual Basic

```
Public Function GetColumnSQLName( _
   ByVal columnID As Integer _
) As String
```

Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public string  GetColumnSQLName(
   int columnID
);
```

パラメータ

- **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。

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、非エイリアス・カラム、非計算カラムの GetColumnName とは異なります。GetColumnSQLName は、常にカラムの名前を返します (プレフィクスとしてテーブル名は付きません)。この動作は、他の ADO.NET プロバイダ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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULCursorSchema クラス」 147 ページ
- 「ULCursorSchema メンバ」 147 ページ
- 「ColumnCount プロパティ」 148 ページ
- 「GetColumnName メソッド」 150 ページ
- 「ColumnCount プロパティ」 148 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。

構文

Visual Basic

```
Public Function GetColumnULDbType( _
   ByVal columnID As Integer _
) As ULDbType
```


Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public ULDbType GetColumnULDbType(
  int columnID
);
```

パラメータ

- **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 クラス」 147 ページ
- 「ULCursorSchema メンバ」 147 ページ
- 「ColumnCount プロパティ」 148 ページ
- 「ColumnCount プロパティ」 148 ページ
- 「ULDbType 列挙」 245 ページ

GetSchemaTable メソッド

ULDataReader のカラムのスキーマが記述された System.Data.DataTable を返します。

構文

Visual Basic

```
Public Function GetSchemaTable() As DataTable
```

C#

```
public DataTable GetSchemaTable();
```

戻り値

カラムのスキーマが記述された System.Data.DataTable。

備考

詳細については、GetSchemaTable メソッドを参照してください。

参照

- 「ULCursorSchema クラス」 147 ページ
- 「ULCursorSchema メンバ」 147 ページ
- DataTable
- 「ULDataReader クラス」 201 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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示します。

構文

Visual Basic

NotInheritable Public Class **ULDataAdapter**
  Inherits Component

C#

public sealed class **ULDataAdapter** :
  Component

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 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 または ULTable を使用すると、より効率的にデータにアクセスできます。

実装： System.Data.IDbDataAdapter、System.Data.IDataAdapter、System.IDisposable

参照

- 「ULDataAdapter メンバ」 156 ページ
- DataSet
- 「ULDataReader クラス」 201 ページ
- 「ULTable クラス」 431 ページ
- IDbDataAdapter
- IDataAdapter
- IDisposable

ULDataAdapter メンバ

パブリック・コンストラクタ

メンバ名	説明
ULDataAdapter コンストラクタ	ULDataAdapter オブジェクトを初期化します。
ULDataAdapter コンストラクタ	ULDataAdapter オブジェクトを、指定された SELECT 文で初期化します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
ULDataAdapter コンストラクタ	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 コンストラクタ	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。

パブリック・プロパティ

メンバ名	説明
AcceptChangesDuringFill プロパティ	System.Data.DataTable クラスに追加された後で、System.Data.DataRow で System.Data.DataRow.AcceptChanges を呼び出すかどうかを指定します。
ContinueUpdateOnError プロパティ	ローの更新時にエラーが検出されたときに例外を生成するかどうかを指定します。
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 (System.Data.DataSet) 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。
InsertCommand プロパティ	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、DbDataAdapter.Update (System.Data.DataSet) 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。
MissingMappingAction プロパティ	受信データと一致するテーブルまたはカラムがない場合に実行するアクションを決定します。
MissingSchemaAction プロパティ	既存の System.Data.DataSet スキーマが受信データと一致しない場合に実行するアクションを決定します。
SelectCommand プロパティ	ULDataAdapter.Fill(System.Data.DataSet) または ULDataAdapter.FillSchema (System.Data.DataSet,System.Data.SchemaType) 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。
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、ULDataAdapter.Update (System.Data.DataSet) 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 © 2007, iAnywhere Solutions, Inc.

パブリック・メソッド

メンバ名	説明
Fill メソッド	指定された System.Data.DataSet の指定された System.Data.DataTable で、ローの追加または再表示を行います。必要に応じて、System.Data.DataTable が作成され、System.Data.DataSet に追加されます。
Fill メソッド	指定された System.Data.DataSet の指定された System.Data.DataTable で、ローの追加または再表示を行います。必要に応じて、System.Data.DataTable が作成され、System.Data.DataSet に追加されます。
Fill メソッド	指定された System.Data.DataSet の指定された System.Data.DataTable で、指定された範囲のローの追加または再表示を行います。必要に応じて、System.Data.DataTable が作成され、System.Data.DataSet に追加されます。
Fill メソッド	指定された System.Data.DataTable でローの追加または再表示を行います。
FillSchema メソッド	System.Data.DataTable を "Table" という System.Data.DataSet に追加し、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。
FillSchema メソッド	System.Data.DataTable を "Table" という System.Data.DataSet に追加し、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。
FillSchema メソッド	System.Data.DataTable のスキーマがデータ・ソースのスキーマと一致するように設定します。
GetFillParameters メソッド	SELECT 文の実行時にユーザによって設定されるパラメータを返します。
Update メソッド	指定された System.Data.DataSet の "Table" という System.Data.DataTable への変更内容を使用して、データベース内のローを更新します。
Update メソッド	指定された System.Data.DataSet の System.Data.DataTable への変更内容を使用して、データベース内のローを更新します。
Update メソッド	指定された System.Data.DataTable への変更内容を使用して、データベース内のローを更新します。
Update メソッド	指定された System.Data.DataRow 配列への変更内容を使用して、データベース内のローを更新します。

パブリック・イベント

メンバ名	説明
FillError イベント	Fill 操作の実行時にエラーが検出されたときに発生します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。更新が試みられると、イベントが発生します。
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。更新が試みられると、イベントが発生します。

ULDataAdapter コンストラクタ

ULDataAdapter オブジェクトを初期化します。

構文

Visual Basic

Overloads Public Sub **New()**

C#

public **ULDataAdapter();**

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- 「ULDataAdapter コンストラクタ」 159 ページ
- 「ULDataAdapter コンストラクタ」 160 ページ
- 「ULDataAdapter コンストラクタ」 161 ページ

ULDataAdapter コンストラクタ

ULDataAdapter オブジェクトを、指定された SELECT 文で初期化します。

構文

Visual Basic

```
Overloads Public Sub New( _
  ByVal selectCommand As ULCommand _
)
```

C#

```
public ULDataAdapter(
  ULCommand selectCommand
);
```

Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **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 クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- 「ULDataAdapter コンストラクタ」 159 ページ
- 「ULDataAdapter コンストラクタ」 160 ページ
- 「ULDataAdapter コンストラクタ」 161 ページ
- 「ULCommand クラス」 57 ページ
- DbDataAdapter.Fill
- DataSet

ULDataAdapter コンストラクタ

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。

構文

Visual Basic

```
Overloads Public Sub New( _
  ByVal selectCommandText As String, _
  ByVal selectConnection As ULConnection _
)
```

C#

```
public ULDataAdapter(
  string  selectCommandText,
  ULConnection selectConnection
);
```

パラメータ

- **selectCommandText** ULDataAdapter の ULDataAdapter.SelectCommand 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 クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- 「ULDataAdapter コンストラクタ」 159 ページ
- 「ULDataAdapter コンストラクタ」 159 ページ
- 「ULDataAdapter コンストラクタ」 161 ページ
- 「SelectCommand プロパティ」 165 ページ
- 「ULConnection クラス」 90 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULDataAdapter コンストラクタ

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。

構文

Visual Basic

```
Overloads Public Sub New( _
  ByVal selectCommandText As String, _
  ByVal selectConnectionString As String _
)
```

C#

```
public ULDataAdapter(
  string  selectCommandText,
  string  selectConnectionString
);
```

パラメータ

- **selectCommandText**　ULDataAdapter の ULDataAdapter.SelectCommand 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 クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- 「ULDataAdapter コンストラクタ」 159 ページ
- 「ULDataAdapter コンストラクタ」 159 ページ
- 「ULDataAdapter コンストラクタ」 160 ページ
- 「SelectCommand プロパティ」 165 ページ

AcceptChangesDuringFill プロパティ

System.Data.DataTable クラスに追加された後で、System.Data.DataRow で System.Data.DataRow.AcceptChanges を呼び出すかどうかを指定します。

構文

Visual Basic

```
Public Property AcceptChangesDuringFill As Boolean
```

C#

```
public bool AcceptChangesDuringFill {get;set;}
```

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

ULDataAdapter が System.Data.DataRow クラスで System.Data.DataRow.AcceptChanges 関数を呼び出すように指定する場合は true、AcceptChanges を呼び出さず、新しく追加されたローを挿入されたローとして処理する場合は false。デフォルトは true です。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- DataRow.AcceptChanges
- DataRow
- DataTable

ContinueUpdateOnError プロパティ

ローの更新時にエラーが検出されたときに例外を生成するかどうかを指定します。

構文

Visual Basic

Public Property **ContinueUpdateOnError** As Boolean

C#

public bool **ContinueUpdateOnError** {get;set;}

プロパティ値

例外を生成しないで更新を継続する場合は true、例外を生成する場合は false。デフォルトは false です。

備考

ContinueUpdateOnError が true の場合、ローの更新時にエラーが発生しても例外はスローされません。ローの更新はスキップされ、エラー情報はローの System.Data.DataRow.RowError メソッドに格納されます。ULDataAdapter は後続のローの更新を継続します。

ContinueUpdateOnError が false の場合、エラーが発生すると例外がスローされます。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- DataRow.RowError

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(System.Data.DataSet) 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 © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **DeleteCommand** As ULCommand

C#

public ULCommand **DeleteCommand** {get;set;}

プロパティ値

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。

備考

DeleteCommand 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 は、既存の ULCommand への参照を保持します。

これは、System.Data.IDbDataAdapter.DeleteCommand が厳密に型指定されたものです。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- 「ULCommand クラス」 57 ページ
- DbDataAdapter.Update
- DataSet
- 「ULCommand クラス」 57 ページ
- IDbDataAdapter.DeleteCommand

InsertCommand プロパティ

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、DbDataAdapter.Update(System.Data.DataSet) 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。

構文

Visual Basic

Public Property **InsertCommand** As ULCommand

C#

public ULCommand **InsertCommand** {get;set;}

プロパティ値

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。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

InsertCommand 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 は、既存の ULCommand への参照を保持します。

これは、System.Data.IDbDataAdapter.InsertCommand が厳密に型指定されたものです。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- 「ULCommand クラス」 57 ページ
- DbDataAdapter.Update
- DataSet
- 「ULCommand クラス」 57 ページ
- IDbDataAdapter.InsertCommand

MissingMappingAction プロパティ

受信データと一致するテーブルまたはカラムがない場合に実行するアクションを決定します。

構文

Visual Basic

```
NotOverridable Public Property MissingMappingAction As MissingMappingAction _
  Implements IDataAdapter.MissingMappingAction
```

C#

```
public MissingMappingAction MissingMappingAction {get;set;}
```

プロパティ値

System.Data.MissingMappingAction 値の 1 つ。デフォルトは System.Data.MissingMappingAction.Passthrough です。

実装

IDataAdapter.MissingMappingAction

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- MissingMappingAction
- MissingMappingAction.Passthrough

Copyright © 2007, iAnywhere Solutions, Inc.

MissingSchemaAction プロパティ

既存の System.Data.DataSet スキーマが受信データと一致しない場合に実行するアクションを決定します。

構文

Visual Basic

```
NotOverridable Public Property MissingSchemaAction As MissingSchemaAction _
  Implements IDataAdapter.MissingSchemaAction
```

C#

```
public MissingSchemaAction MissingSchemaAction {get;set;}
```

プロパティ値

System.Data.MissingSchemaAction 値の 1 つ。デフォルトは System.Data.MissingSchemaAction.Add です。

実装

IDataAdapter.MissingSchemaAction

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- DataSet
- MissingSchemaAction
- MissingSchemaAction.Add

SelectCommand プロパティ

ULDataAdapter.Fill(System.Data.DataSet) または ULDataAdapter.FillSchema (System.Data.DataSet,System.Data.SchemaType) 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。

構文

Visual Basic

```
Public Property SelectCommand As ULCommand
```

C#

```
public ULCommand SelectCommand {get;set;}
```

プロパティ値

System.Data.DataSet を設定するために実行される ULCommand オブジェクト。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

SelectCommand 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 は、既存の ULCommand への参照を保持します。

SelectCommand がローを返さない場合、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) のコンストラクタにも指定できます。

これは、System.Data.IDbDataAdapter.SelectCommand が厳密に型指定されたものです。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- 「ULCommand クラス」 57 ページ
- DataSet
- 「Fill メソッド」 168 ページ
- 「FillSchema メソッド」 172 ページ
- 「ULDataAdapter コンストラクタ」 159 ページ
- 「ULDataAdapter コンストラクタ」 160 ページ
- 「ULDataAdapter コンストラクタ」 161 ページ
- IDbDataAdapter.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。

構文

Visual Basic

Public Readonly Property **TableMappings** As DataTableMappingCollection

C#

public DataTableMappingCollection **TableMappings** {get;}

プロパティ値

ソース・テーブルと System.Data.DataTables とのマスタ・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。

備考

変更を調整する場合、ULDataAdapter 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 が厳密に型指定されたものです。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- DataTable
- DataTableMapping
- DataTable
- DataTableMappingCollection
- DataSet
- IDataAdapter.TableMappings

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、ULDataAdapter.Update(System.Data.DataSet) 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。

構文

Visual Basic

Public Property **UpdateCommand** As ULCommand

C#

public ULCommand **UpdateCommand** {get;set;}

プロパティ値

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。

備考

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 は、既存の ULCommand 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 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 が厳密に型指定されたものです。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- 「ULCommand クラス」 57 ページ
- DataSet
- 「Update メソッド」 176 ページ
- 「UpdatedRowSource プロパティ」 69 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- IDbDataAdapter.UpdateCommand

Fill メソッド

指定された System.Data.DataSet の指定された System.Data.DataTable で、ローの追加または再表示を行います。必要に応じて、System.Data.DataTable が作成され、System.Data.DataSet に追加されます。

構文

Visual Basic

```
Overloads NotOverridable Public Function Fill( _
  ByVal dataSet As DataSet _
) As Integer _
  Implements IDataAdapter.Fill
```

C#

```
public int Fill(
  DataSet dataSet
);
```

パラメータ

- **dataSet**　レコードと (必要に応じて) スキーマを使用して設定する System.Data.DataSet。

戻り値

System.Data.DataSet で正常に追加または再表示されたローの数。

実装

IDataAdapter.Fill

Fill メソッド

指定された System.Data.DataSet の指定された System.Data.DataTable で、ローの追加または再表示を行います。必要に応じて、System.Data.DataTable が作成され、System.Data.DataSet に追加されます。

構文

Visual Basic

```
Overloads Public Function Fill( _
  ByVal dataSet As DataSet, _
  ByVal srcTable As String _
) As Integer
```

C#

```
public int Fill(
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
    DataSet dataSet,
    string  srcTable
);
```

パラメータ

- **dataSet**　レコードと (必要に応じて) スキーマを使用して設定する System.Data.DataSet。
- **srcTable**　テーブルのマッピングに使用するソース・テーブルの名前。

戻り値

System.Data.DataSet で正常に追加または再表示されたローの数。

備考

結果セットのサイズが大きい場合、パフォーマンスに重大な影響を及ぼす可能性があります。別の方法として、読み込み専用の結果セットで十分であれば、ULDataReader (場合によっては SQL 文 (ExecuteNonQuery) とともに) を使用して修正を行います。さらに別の方法として、そのデータベースへの読み込みと書き込みアクセスが可能な ULTable を使用できます。

ULDataAdapter.SelectCommand がローを返さない場合、System.Data.DataSet にテーブルが追加されず、例外も発生しません。

詳細については、「IDataAdapter.Fill」を参照してください。

例外

- System.ArgumentNullException - *dataSet* または *srcTable* パラメータが無効です。
- InvalidOperationException - ULDataAdapter.SelectCommand が無効です。または、テーブルのマッピングが見つからない上に DataAdapter.MissingMappingAction が System.Data.MissingMappingAction.Error に設定されています。あるいは、System.Data.DataTable が System.Data.DataSet で見つからず、ULDataAdapter.MissingSchemaAction が System.Data.MissingSchemaAction.Error に設定されています。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- DataTable
- DataSet
- 「ULDataReader クラス」 201 ページ
- 「ULTable クラス」 431 ページ
- 「SelectCommand プロパティ」 165 ページ
- IDataAdapter.Fill
- 「SelectCommand プロパティ」 165 ページ
- 「MissingMappingAction プロパティ」 164 ページ
- MissingMappingAction.Error
- 「MissingSchemaAction プロパティ」 165 ページ
- MissingSchemaAction.Error

Copyright © 2007, iAnywhere Solutions, Inc.

Fill メソッド

指定された System.Data.DataSet の指定された System.Data.DataTable で、指定された範囲のローの追加または再表示を行います。必要に応じて、System.Data.DataTable が作成され、System.Data.DataSet に追加されます。

構文

Visual Basic

```
Overloads Public Function Fill( _
  ByVal dataSet As DataSet, _
  ByVal startRecord As Integer, _
  ByVal maxRecords As Integer, _
  ByVal srcTable As String _
) As Integer
```

C#

```
public int Fill(
  DataSet dataSet,
  int startRecord,
  int maxRecords,
  string  srcTable
);
```

パラメータ

- **dataSet**　レコードと (必要に応じて) スキーマを使用して設定する System.Data.DataSet。
- **startRecord**　先頭の 0 から始まるレコード番号。
- **maxRecords**　System.Data.DataSet に読み込むレコードの最大数。値が 0 であると、先頭のレコードの後に見つかったすべてのレコードが取得されます。
- **srcTable**　テーブルのマッピングに使用するソース・テーブルの名前。

戻り値

System.Data.DataSet で正常に追加または再表示されたローの数。

備考

startRecord 引数を使用して、System.Data.DataSet にコピーされるレコードの数を制限した場合でも、ULDataAdapter クエリ内のすべてのレコードがデータベースからクライアントにフェッチされます。結果セットのサイズが大きい場合、パフォーマンスに重大な影響を及ぼす可能性があります。別の方法として、読み込み専用の結果セットで十分であれば、ULDataReader (場合によっては SQL 文 (ExecuteNonQuery) とともに) を使用して修正を行います。さらに別の方法として、そのデータベースへの読み込みと書き込みアクセスが可能な ULTable を使用できます。

ULDataAdapter.SelectCommand がローを返さない場合、System.Data.DataSet にテーブルが追加されず、例外も発生しません。

詳細については、「IDataAdapter.Fill」を参照してください。

Copyright © 2007, iAnywhere Solutions, Inc.

例外

- ArgumentException - *startRecord* または *maxRecords* パラメータがゼロ未満でした。
- System.ArgumentNullException - *dataSet* または *srcTable* パラメータが無効です。
- InvalidOperationException - ULDataAdapter.SelectCommand が無効です。または、テーブルのマッピングが見つからない上に DataAdapter.MissingMappingAction が System.Data.MissingMappingAction.Error に設定されています。あるいは、System.Data.DataTable が System.Data.DataSet で見つからず、ULDataAdapter.MissingSchemaAction が System.Data.MissingSchemaAction.Error に設定されています。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- DataTable
- DataSet
- 「ULDataReader クラス」 201 ページ
- 「ULTable クラス」 431 ページ
- 「SelectCommand プロパティ」 165 ページ
- IDataAdapter.Fill
- 「SelectCommand プロパティ」 165 ページ
- 「MissingMappingAction プロパティ」 164 ページ
- MissingMappingAction.Error
- 「MissingSchemaAction プロパティ」 165 ページ
- MissingSchemaAction.Error

Fill メソッド

指定された System.Data.DataTable でローの追加または再表示を行います。

構文

Visual Basic

```
Overloads Public Function Fill( _
  ByVal dataTable As DataTable _
) As Integer
```

C#

```
public int Fill(
  DataTable dataTable
);
```

パラメータ

- **dataTable**　レコードと (必要に応じて) スキーマを使用して設定する System.Data.DataTable。

Copyright © 2007, iAnywhere Solutions, Inc.

戻り値

System.Data.DataTable で正常に追加または再表示されたローの数。

備考

結果セットのサイズが大きい場合、パフォーマンスに重大な影響を及ぼす可能性があります。別の方法として、読み込み専用の結果セットで十分であれば、ULDataReader (場合によっては SQL 文 (ExecuteNonQuery) とともに) を使用して修正を行います。さらに別の方法として、そのデータベースへの読み込みと書き込みアクセスが可能な ULTable を使用できます。

ULDataAdapter.SelectCommand がローを返さない場合、テーブルが追加されず、例外も発生しません。

詳細については、「IDataAdapter.Fill」を参照してください。

例外

- System.ArgumentNullException - *dataTable* パラメータが無効です。
- InvalidOperationException - ULDataAdapter.SelectCommand が無効です。または、テーブルのマッピングが見つからない上に ULDataAdapter.MissingMappingAction が System.Data.MissingMappingAction.Error に設定されています。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- DataTable
- DataSet
- 「ULDataReader クラス」 201 ページ
- 「ULTable クラス」 431 ページ
- 「SelectCommand プロパティ」 165 ページ
- IDataAdapter.Fill
- 「SelectCommand プロパティ」 165 ページ
- 「MissingMappingAction プロパティ」 164 ページ
- MissingMappingAction.Error

FillSchema メソッド

System.Data.DataTable を "Table" という System.Data.DataSet に追加し、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。

構文

Visual Basic

```
Overloads NotOverridable Public Function FillSchema( _
   ByVal dataSet As DataSet, _
   ByVal schemaType As SchemaType _
) As DataTable() _
   Implements IDataAdapter.FillSchema
```

 Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public DataTable[] FillSchema(
  DataSet dataSet,
  SchemaType schemaType
);
```

パラメータ

- **dataSet**　スキーマを使用して設定する System.Data.DataSet。
- **schemaType**　スキーマを挿入する方法を指定する System.Data.SchemaType 値の 1 つ。

戻り値

System.Data.DataSet に追加された System.Data.DataTable オブジェクトのコレクションへの参照。

備考

詳細については、「IDataAdapter.FillSchema」を参照してください。

例外

- System.ArgumentNullException - *dataSet* パラメータが無効です。
- InvalidOperationException - ULDataAdapter.SelectCommand が無効です。または、テーブルのマッピングが見つからない上に ULDataAdapter.MissingMappingAction が System.Data.MissingMappingAction.Error に設定されています。

実装

IDataAdapter.FillSchema

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- DataTable
- DataSet
- 「ULDataReader クラス」 201 ページ
- 「ULTable クラス」 431 ページ
- 「SelectCommand プロパティ」 165 ページ
- IDataAdapter.Fill
- 「SelectCommand プロパティ」 165 ページ
- 「MissingMappingAction プロパティ」 164 ページ
- MissingMappingAction.Error

FillSchema メソッド

System.Data.DataTable を "Table" という System.Data.DataSet に追加し、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 © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Overloads Public Function FillSchema( _
  ByVal dataSet As DataSet, _
  ByVal schemaType As SchemaType, _
  ByVal srcTable As String _
) As DataTable()
```

C#

```
public DataTable[] FillSchema(
  DataSet dataSet,
  SchemaType schemaType,
  string  srcTable
);
```

パラメータ

- **dataSet**　スキーマを使用して設定する System.Data.DataSet。
- **schemaType**　スキーマを挿入する方法を指定する System.Data.SchemaType 値の 1 つ。
- **srcTable**　テーブルのマッピングに使用するソース・テーブルの名前。

戻り値

System.Data.DataSet に追加された System.Data.DataTable オブジェクトのコレクションへの参照。

備考

詳細については、「DbDataAdapter.FillSchema」を参照してください。

例外

- System.ArgumentNullException - *dataSet* または *srcTable* パラメータが無効です。
- InvalidOperationException - ULDataAdapter.SelectCommand が無効です。または、テーブルのマッピングが見つからない上に ULDataAdapter.MissingMappingAction が System.Data.MissingMappingAction.Error に設定されています。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- DataTable
- DataSet
- SchemaType
- 「ULDataReader クラス」 201 ページ
- 「ULTable クラス」 431 ページ
- 「SelectCommand プロパティ」 165 ページ
- IDataAdapter.Fill
- 「SelectCommand プロパティ」 165 ページ
- 「MissingMappingAction プロパティ」 164 ページ
- MissingMappingAction.Error

Copyright © 2007, iAnywhere Solutions, Inc.

FillSchema メソッド

System.Data.DataTable のスキーマがデータ・ソースのスキーマと一致するように設定します。

構文

Visual Basic

```
Overloads Public Function FillSchema( _
  ByVal dataTable As DataTable, _
  ByVal schemaType As SchemaType _
) As DataTable
```

C#

```
public DataTable FillSchema(
  DataTable dataTable,
  SchemaType schemaType
);
```

パラメータ

- **dataTable**　スキーマを使用して設定する System.Data.DataTable。
- **schemaType**　スキーマを挿入する方法を指定する System.Data.SchemaType 値の 1 つ。

戻り値

スキーマが含まれている System.Data.DataTable オブジェクトへの参照。

備考

詳細については、「DbDataAdapter.FillSchema」を参照してください。

例外

- System.ArgumentNullException - *dataTable* パラメータが無効です。
- InvalidOperationException - ULDataAdapter.SelectCommand が無効です。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- DataTable
- SchemaType
- 「SelectCommand プロパティ」 165 ページ

GetFillParameters メソッド

SELECT 文の実行時にユーザによって設定されるパラメータを返します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Function GetFillParameters() As ULParameter()
```

C#

```
public ULParameter[] GetFillParameters();
```

戻り値

ユーザによって設定されたパラメータが含まれる ULParameter オブジェクトの配列。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- 「ULParameter クラス」 286 ページ

Update メソッド

指定した System.Data.DataSet の "Table" という System.Data.DataTable への変更内容を使用して、データベース内のローを更新します。

構文

Visual Basic

```
Overloads NotOverridable Public Function Update( _
  ByVal dataSet As DataSet _
) As Integer _
  Implements IDataAdapter.Update
```

C#

```
public int Update(
  DataSet dataSet
);
```

パラメータ

- **dataSet**　更新に使用するレコードの System.Data.DataSet。

戻り値

System.Data.DataSet から正常に更新されたローの数。

備考

詳細については、「IDataAdapter.Update」を参照してください。

例外

- System.ArgumentNullException - *dataSet* パラメータが無効です。

Copyright © 2007, iAnywhere Solutions, Inc.

- InvalidOperationException - ULDataAdapter.SelectCommand が無効です。または、テーブルのマッピングが見つからない上に DataAdapter.MissingMappingAction が System.Data.MissingMappingAction.Error に設定されています。あるいは、System.Data.DataTable が System.Data.DataSet で見つかりません。

実装

IDataAdapter.Update

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- DataTable
- DataSet
- IDataAdapter.Update
- 「SelectCommand プロパティ」 165 ページ
- 「MissingMappingAction プロパティ」 164 ページ
- MissingMappingAction.Error

Update メソッド

指定された System.Data.DataSet の System.Data.DataTable への変更内容を使用して、データベース内のローを更新します。

構文

Visual Basic

```
Overloads Public Function Update( _
  ByVal dataSet As DataSet, _
  ByVal srcTable As String _
) As Integer
```

C#

```
public int Update(
  DataSet dataSet,
  string  srcTable
);
```

パラメータ

- **dataSet**　更新に使用するレコードの System.Data.DataSet。
- **srcTable**　テーブルのマッピングに使用するソース・テーブルの名前。

戻り値

System.Data.DataSet から正常に更新されたローの数。

備考

詳細については、「DbDataAdapter.Update」を参照してください。

Copyright © 2007, iAnywhere Solutions, Inc.

例外

- System.ArgumentNullException - *dataSet* または *srcTable* パラメータが無効です。
- InvalidOperationException - ULDataAdapter.SelectCommand が無効です。または、テーブルのマッピングが見つからない上に DataAdapter.MissingMappingAction が System.Data.MissingMappingAction.Error に設定されています。あるいは、System.Data.DataTable が System.Data.DataSet で見つかりません。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- DataTable
- DbDataAdapter.Update
- DataSet
- 「SelectCommand プロパティ」 165 ページ
- 「MissingMappingAction プロパティ」 164 ページ
- MissingMappingAction.Error

Update メソッド

指定された System.Data.DataTable への変更内容を使用して、データベース内のローを更新します。

構文

Visual Basic

```
Overloads Public Function Update( _
  ByVal dataTable As DataTable _
) As Integer
```

C#

```
public int Update(
  DataTable dataTable
);
```

パラメータ

- **dataTable**　更新の元になる System.Data.DataTable。

戻り値

System.Data.DataTable から正常に更新されたローの数。

備考

詳細については、「DbDataAdapter.Update」を参照してください。

例外

- System.ArgumentNullException - *dataTable* パラメータが無効です。

Copyright © 2007, iAnywhere Solutions, Inc.

- InvalidOperationException - ULDataAdapter.SelectCommand が無効です。または、テーブルのマッピングが見つからない上に ULDataAdapter.MissingMappingAction が System.Data.MissingMappingAction.Error に設定されています。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- DataTable
- DbDataAdapter.Update
- 「SelectCommand プロパティ」 165 ページ
- 「MissingMappingAction プロパティ」 164 ページ
- MissingMappingAction.Error

Update メソッド

指定された System.Data.DataRow 配列への変更内容を使用して、データベース内のローを更新します。

構文

Visual Basic

```
Overloads Public Function Update( _
  ByVal dataRows As DataRow() _
) As Integer
```

C#

```
public int Update(
  DataRow[] dataRows
);
```

パラメータ

- **dataRows**　更新の元になる System.Data.DataRow の配列。

戻り値

System.Data.DataRow 配列から正常に更新されたローの数。

備考

詳細については、「DbDataAdapter.Update」を参照してください。

例外

- System.ArgumentNullException - *dataRows* パラメータが無効です。
- InvalidOperationException - ULDataAdapter.SelectCommand が無効です。または、テーブルのマッピングが見つからない上に ULDataAdapter.MissingMappingAction が System.Data.MissingMappingAction.Error に設定されています。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- DataRow
- DbDataAdapter.Update
- 「SelectCommand プロパティ」 165 ページ
- 「MissingMappingAction プロパティ」 164 ページ
- MissingMappingAction.Error

FillError イベント

Fill 操作の実行時にエラーが検出されたときに発生します。

構文

Visual Basic

Public Event **FillError** As FillErrorEventHandler

C#

public event FillErrorEventHandler **FillError**;

備考

FillError イベントを使用して、エラーが発生した後に Fill 操作を継続するかどうかを決定できます。

FillError イベントが発生する例は、次のとおりです。

- DataSet に追加されるデータを、精度を落とさないで共通言語ランタイム・タイプに変換できない。
- DataSet の DataColumn に適用する必要がある制約に違反するデータが、追加されるローに含まれている。

ローの Fill エラー・イベントを処理するには、System.Data.FillErrorEventHandler 委任を作成し、このイベントにアタッチする必要があります。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- FillErrorEventHandler

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。更新が試みられると、イベントが発生します。

 Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Event **RowUpdated** As ULRowUpdatedEventHandler

C#

public event ULRowUpdatedEventHandler **RowUpdated**;

備考

ロー更新イベントを処理するには、ULRowUpdatedEventHandler 委任を作成し、このイベントにアタッチする必要があります。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- 「ULRowUpdatedEventHandler 委任」 357 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。更新が試みられると、イベントが発生します。

構文

Visual Basic

Public Event **RowUpdating** As ULRowUpdatingEventHandler

C#

public event ULRowUpdatingEventHandler **RowUpdating**;

備考

ロー更新イベントを処理するには、ULRowUpdatingEventHandler 委任を作成し、このイベントにアタッチする必要があります。

参照

- 「ULDataAdapter クラス」 156 ページ
- 「ULDataAdapter メンバ」 156 ページ
- 「ULRowUpdatedEventHandler 委任」 357 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULDatabaseManager クラス

UL 拡張： 同期リスナと Ultra Light.NET のランタイム・タイプを管理します。ULDatabaseManager クラスを使用すると、Ultra Light.NET データベースを削除できます。

構文

Visual Basic

NotInheritable Public Class **ULDatabaseManager**

C#

public sealed class **ULDatabaseManager**

備考

このクラスはシングルトン・クラスであり、その唯一のインスタンスには、静的な (Visual Basic の Shared) ULConnection.DatabaseManager のみを通じてアクセスできます。

Ultra Light.NET の Ultra Light エンジンのランタイムを使用するには、ULDatabaseManager.RuntimeType を適切な値に設定してから、他の Ultra Light.NET API を使用します。

参照

- 「ULDatabaseManager メンバ」 182 ページ
- 「DatabaseManager プロパティ」 98 ページ
- 「RuntimeType プロパティ」 183 ページ

ULDatabaseManager メンバ

パブリックの静的プロパティ (共有)

メンバ名	説明
RuntimeType プロパティ	Ultra Light.NET ランタイム・タイプを指定します。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。

パブリック・メソッド

メンバ名	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。 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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SetActiveSyncListener メソッド	ActiveSync 用 Mobile Link プロバイダからの ActiveSync の呼び出しを処理するリスナ・オブジェクトを指定します。
SetServerSyncListener メソッド	指定されたサーバ同期メッセージの処理に使用するリスナ・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。

RuntimeType プロパティ

Ultra Light.NET ランタイム・タイプを指定します。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。

構文

Visual Basic

Public Shared Property **RuntimeType** As ULRuntimeType

C#

public const ULRuntimeType **RuntimeType** {get;set;}

プロパティ値

アンマネージ Ultra Light.NET ランタイムのタイプを示す ULRuntimeType 値。

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

// C#
ULDatabaseManager.RuntimeType = ULRuntimeType.UL_ENGINE_CLIENT;
ULConnection conn = new ULConnection();
// The RuntimeType is now locked
```

参照

- 「ULDatabaseManager クラス」 182 ページ
- 「ULDatabaseManager メンバ」 182 ページ
- 「ULRuntimeType 列挙」 362 ページ

CreateDatabase メソッド

新しい Ultra Light データベースを作成します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Sub CreateDatabase( _
   ByVal connString As String, _
   ByVal collationData As Byte(), _
   ByVal createParms As String _
)
```

C#

```
public void CreateDatabase(
   string  connString,
   byte[] collationData,
   string  createParms
);
```

パラメータ

- **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。詳細については、ULConnectionParms クラスを参照してください。
- **collationData**　データベースが文字列を格納して比較する方法を指定する照合データ。
- **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。詳細については、ULCreateParms クラスを参照してください。

備考

照合を指定するには、SQL Anywhere インストール・ディレクトリの srcollations¥cs (C# プロジェクトの場合) または srcollations¥vb.net (Visual Basic プロジェクトの場合) サブディレクトリから、適切な照合データ・ソースを含める必要があります。プロジェクトに追加したら、照合の Data プロパティを使用して、CreateDatabase() メソッドに対して照合データを提供できます。

例外

- ULException クラス - SQL エラーが発生しました。

例

次のコードでは、Windows CE 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.DatabaseOnCE = "¥UltraLite¥MyDatabase.udb"
' Assumes file src¥ulcollations¥vb.net¥coll_1250LATIN2.vb is
' also compiled in the current project
ULConnection.DatabaseManager.CreateDatabase( _
    openParms.ToString(), _
    iAnywhere.UltraLite.Collations.Collation_1250LATIN2.Data, _
    "" _
  )
Dim conn As ULConnection = _
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
  New ULConnection( openParms.ToString() )
conn.Open()

// C#
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnCE = @"¥UltraLite¥MyDatabase.udb";
// Assumes file src¥ulcollations¥cs¥coll_1250LATIN2.cs is
// also compiled in the current project
ULConnection.DatabaseManager.CreateDatabase(
    openParms.ToString(),
    iAnywhere.UltraLite.Collations.Collation_1250LATIN2.Data,
    ""
  );
ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```

参照

- 「ULDatabaseManager クラス」 182 ページ
- 「ULDatabaseManager メンバ」 182 ページ
- 「Open メソッド」 116 ページ

DropDatabase メソッド

指定されたデータベースを削除します。

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。

構文

Visual Basic

```
Public Sub DropDatabase( _
  ByVal connString As String _
)
```

C#

```
public void DropDatabase(
  string  connString
);
```

パラメータ

- **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。詳細については、ULConnectionParms クラスを参照してください。

例外

- ULException クラス - SQL エラーが発生しました。

例

次のコードでは、Windows CE 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。

Copyright © 2007, iAnywhere Solutions, Inc.

```
' Visual Basic
Dim connParms As ULConnectionParms = New ULConnectionParms
connParms.DatabaseOnCE = "¥UltraLite¥MyDatabase.udb"
ULConnection.DatabaseManager.DropDatabase( _
   connParms.ToString() _
  )

// C#
ULConnectionParms connParms = new ULConnectionParms();
connParms.DatabaseOnCE = @"¥UltraLite¥MyDatabase.udb";
ULConnection.DatabaseManager.DropDatabase(
   connParms.ToString()
  );
ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```

参照

- 「ULDatabaseManager クラス」 182 ページ
- 「ULDatabaseManager メンバ」 182 ページ
- 「Open メソッド」 116 ページ

SetActiveSyncListener メソッド

ActiveSync 用 Mobile Link プロバイダからの ActiveSync の呼び出しを処理するリスナ・オブジェクトを指定します。

構文

Visual Basic

```
Public Sub SetActiveSyncListener( _
  ByVal appClassName As String, _
  ByVal listener As ULActiveSyncListener _
)
```

C#

```
public void SetActiveSyncListener(
  string  appClassName,
  ULActiveSyncListener listener
);
```

パラメータ

- **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。前のリスナを削除するには、null (Visual Basic の Nothing) を使用します。

備考

パラメータ *appClassName* 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* のみを使用できます。リスナが、特定の

Copyright © 2007, iAnywhere Solutions, Inc.

appClassName に登録されているときは、別の *appClassName* で SetServerSyncListener または SetActiveSyncListener を呼び出すと失敗します。

ActiveSync リスナを削除するには、*listener* パラメータとして null 参照 (Visual Basic の Nothing) を使用して SetActiveSyncListener を呼び出します。

すべてのリスナを削除するには、すべてのパラメータに null 参照 (Visual Basic の Nothing) を使用して SetServerSyncListener を呼び出します。

アプリケーションは、すべてのリスナを削除してから終了する必要があります。

例外

- ULException クラス - SQL エラーが発生しました。

例

SetActiveSyncListener の例については、ActiveSyncInvoked メソッドを参照してください。

参照

- 「ULDatabaseManager クラス」 182 ページ
- 「ULDatabaseManager メンバ」 182 ページ
- 「SetServerSyncListener メソッド」 187 ページ
- 「SetActiveSyncListener メソッド」 186 ページ
- 「ULActiveSyncListener インタフェース」 53 ページ
- 「ActiveSyncInvoked メソッド」 53 ページ

SetServerSyncListener メソッド

指定されたサーバ同期メッセージの処理に使用するリスナ・オブジェクトを指定します。

構文

Visual Basic

```
Public Sub SetServerSyncListener( _
  ByVal messageName As String, _
  ByVal appClassName As String, _
  ByVal listener As ULServerSyncListener _
)
```

C#

```
public void SetServerSyncListener(
  string  messageName,
  string  appClassName,
  ULServerSyncListener listener
);
```

パラメータ

- **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。前のリスナを削除するには、null (Visual Basic の Nothing) を使用します。

備考

パラメータ *appClassName* 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* のみを使用できます。リスナが、特定の *appClassName* に登録されているときは、別の *appClassName* で SetServerSyncListener または SetActiveSyncListener を呼び出すと失敗します。

特定のメッセージのリスナを削除するには、*listener* パラメータとして null 参照 (Visual Basic の Nothing) を使用して SetServerSyncListener を呼び出します。

すべてのリスナを削除するには、すべてのパラメータに null 参照 (Visual Basic の Nothing) を使用して SetServerSyncListener を呼び出します。

アプリケーションは、すべてのリスナを削除してから終了する必要があります。

例外

- ULException クラス - SQL エラーが発生しました。

例

SetServerSyncListener の例については、ServerSyncInvoked メソッドを参照してください。

参照

- 「ULDatabaseManager クラス」 182 ページ
- 「ULDatabaseManager メンバ」 182 ページ
- 「SetServerSyncListener メソッド」 187 ページ
- 「SetActiveSyncListener メソッド」 186 ページ
- 「ServerSyncInvoked メソッド」 363 ページ

SignalSyncIsComplete メソッド

ActiveSync 用の Mobile Link プロバイダに対して、アプリケーションが同期を完了したことを通知します。

構文

Visual Basic

```
Public Sub SignalSyncIsComplete()
```

C#

```
public void SignalSyncIsComplete();
```


Copyright © 2007, iAnywhere Solutions, Inc.

例

SignalSyncIsComplete の例については、ActiveSyncInvoked メソッドを参照してください。

参照

- 「ULDatabaseManager クラス」 182 ページ
- 「ULDatabaseManager メンバ」 182 ページ
- 「SignalSyncIsComplete メソッド」 188 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

NotInheritable Public Class **ULDatabaseSchema**

C#

public sealed class **ULDatabaseSchema**

備考

このクラスにはコンストラクタがありません。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 メンバ」 190 ページ
- 「ULDatabaseSchema クラス」 190 ページ
- 「Schema プロパティ」 100 ページ

ULDatabaseSchema メンバ

パブリック・プロパティ

メンバ名	説明
CollationName プロパティ	データベースの照合順の名前です。
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。

パブリック・メソッド

メンバ名	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。パブリケーション ID は、パブリケーション・マスクではありません。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
GetPublicationSchema メソッド	指定されたパブリケーションに対応するパブリケーション・スキーマを返します。
GetTableCountInPublications メソッド	指定されたパブリケーション・マスクに含まれるテーブルの数を返します。
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。

CollationName プロパティ

データベースの照合順の名前です。

構文

Visual Basic

Public Readonly Property **CollationName** As String

C#

public string **CollationName** {get;}

プロパティ値

データベースの照合順を示す文字列。

備考

データベースの照合順は、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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULDatabaseSchema クラス」 190 ページ
- 「ULDatabaseSchema メンバ」 190 ページ
- 「GetDatabaseProperty メソッド」 194 ページ

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **IsCaseSensitive** As Boolean

C#

public bool **IsCaseSensitive** {get;}

プロパティ値

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。

備考

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULDatabaseSchema クラス」 190 ページ
- 「ULDatabaseSchema メンバ」 190 ページ
- 「GetDatabaseProperty メソッド」 194 ページ
- 「UserID プロパティ」 133 ページ
- 「Password プロパティ」 133 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。

構文

Visual Basic

Public Readonly Property **IsOpen** As Boolean

C#

public bool **IsOpen** {get;}

プロパティ値

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。

備考

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。

Copyright © 2007, iAnywhere Solutions, Inc.

PublicationCount プロパティ

データベース内のパブリケーションの数

構文

Visual Basic

Public Readonly Property **PublicationCount** As Integer

C#

public int **PublicationCount** {get;}

プロパティ値

データベース内のパブリケーション数。接続が開いていない場合は 0。

備考

パブリケーション ID の範囲は、1 ー PublicationCount です。パブリケーション ID は、パブリケーション・マスクではありません。

注意：パブリケーションの ID、マスク、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、マスク、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 クラス」 190 ページ
- 「ULDatabaseSchema メンバ」 190 ページ
- 「GetPublicationName メソッド」 196 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。

構文

Visual Basic

Public Readonly Property **TableCount** As Integer

C#

public int **TableCount** {get;}

プロパティ値

データベース内のテーブル数。接続が開いていない場合は 0。

備考

テーブル ID の範囲は、1 ー TableCount です。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

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。

構文

Visual Basic

```
Public Function GetDatabaseProperty( _
  ByVal name As String _
) As String
```

C#

```
public string  GetDatabaseProperty(
  string  name
);
```

パラメータ

- **name**　値を取得するデータベース・プロパティの名前。プロパティ名には大文字と小文字の区別はありません。

戻り値

文字列として返されるプロパティの値。

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 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。
CollationName	データベースの照合順の名前。

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ	説明
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。
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。
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。
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ナノ秒 (1,000,000 分の 1 秒) 単位の最小差。

例外

- ULException クラス - SQL エラーが発生しました。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULDatabaseSchema クラス」 190 ページ
- 「ULDatabaseSchema メンバ」 190 ページ
- 「SetDatabaseOption メソッド」 199 ページ
- 「CollationName プロパティ」 191 ページ
- 「UserID プロパティ」 133 ページ
- 「Password プロパティ」 133 ページ
- TimeSpan
- 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。パブリケーション ID は、パブリケーション・マスクではありません。

構文

Visual Basic

```
Public Function GetPublicationName( _
  ByVal pubID As Integer _
) As String
```

C#

```
public string  GetPublicationName(
  int pubID
);
```

パラメータ

- **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。

備考

注意：パブリケーションの ID、マスク、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、マスク、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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULDatabaseSchema クラス」 190 ページ
- 「ULDatabaseSchema メンバ」 190 ページ
- 「PublicationCount プロパティ」 193 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「PublicationCount プロパティ」 193 ページ

GetPublicationSchema メソッド

指定されたパブリケーションに対応するパブリケーション・スキーマを返します。

構文

Visual Basic

```
Public Function GetPublicationSchema( _
  ByVal name As String _
) As ULPublicationSchema
```

C#

```
public ULPublicationSchema GetPublicationSchema(
  string  name
);
```

パラメータ

- **name**　パブリケーションの名前。

戻り値

指定されたパブリケーションを示す ULPublicationSchema 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

- 「ULDatabaseSchema クラス」 190 ページ
- 「ULDatabaseSchema メンバ」 190 ページ
- 「GetPublicationName メソッド」 196 ページ
- 「ULPublicationSchema クラス」 321 ページ

GetTableCountInPublications メソッド

指定されたパブリケーション・マスクに含まれるテーブルの数を返します。

構文

Visual Basic

```
Public Function GetTableCountInPublications( _
  ByVal mask As Integer _
) As Integer
```

C#

```
public int GetTableCountInPublications(
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
  int mask
);
```

パラメータ

- **mask**　チェックするパブリケーションのセット。パブリケーション・マスク作成の詳細については、ULPublicationSchema クラスを参照してください。

戻り値

パブリケーションのセットに含まれるテーブル数。

備考

カウントには、名前が _nosync で終わるテーブルは含まれません。

注意：パブリケーションの ID、マスク、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、マスク、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。

例外

- ULException クラス - SQL エラーが発生しました。

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。

構文

Visual Basic

```
Public Function GetTableName( _
  ByVal tableID As Integer _
) As String
```

C#

```
public string  GetTableName(
  int tableID
);
```

パラメータ

- **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

文字列として返されるテーブル名。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULDatabaseSchema クラス」 190 ページ
- 「ULDatabaseSchema メンバ」 190 ページ
- 「TableCount プロパティ」 193 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。

構文

Visual Basic

```
Public Sub SetDatabaseOption( _
  ByVal name As String, _
  ByVal value As String _
)
```

C#

```
public void SetDatabaseOption(
  string  name,
  string  value
);
```

パラメータ

- **name**　データベース・オプションの名前。オプション名には大文字と小文字の区別はありません。
- **value**　オプションの新しい値。

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。

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。
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。

Copyright © 2007, iAnywhere Solutions, Inc.

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULDatabaseSchema クラス」 190 ページ
- 「ULDatabaseSchema メンバ」 190 ページ
- 「GetDatabaseProperty メソッド」 194 ページ
- UInt32.MaxValue
- 「INVALID_DATABASE_ID フィールド」 95 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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示します。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。

構文

Visual Basic

```
Public Class ULDataReader
  Inherits MarshalByRefObject
```

C#

```
public class ULDataReader :
  MarshalByRefobject
```

備考

ULDataReader にはコンストラクタがありません。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 FOR READ ONLY", conn _
  )
Dim reader As ULDataReader = cmd.ExecuteReader()

// C#
ULCommand cmd = new ULCommand(
    "SELECT emp_id FROM employee FOR READ ONLY", 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。

実装： System.Data.IDataReader、System.Data.IDataRecord、System.IDisposable

参照

- 「ULDataReader メンバ」 202 ページ
- 「ULCommand クラス」 57 ページ
- 「ExecuteResultSet メソッド」 75 ページ
- 「ExecuteTable メソッド」 78 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULDataAdapter クラス」 156 ページ
- 「GetString メソッド」 232 ページ
- IDataReader
- IDataRecord
- IDisposable

ULDataReader メンバ

パブリック・プロパティ

メンバ名	説明
Depth プロパティ	現在のローのネストの深さを返します。最も外側のテーブルの深さは 0 です。
FieldCount プロパティ	このカーソル内のカラム数を返します。
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。
Item 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。C# では、このプロパティは ULDataReader クラスのインデクサです。
Item 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。C# では、このプロパティは ULDataReader クラスのインデクサです。
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。SELECT 文または CommandType.TableDirect テーブルの場合、この値は -1 です。
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。

パブリック・メソッド

メンバ名	説明
Close メソッド	カーソルを閉じます。
Dispose メソッド	ULDataReader 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。
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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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	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 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。ULDbType.Binary 型、ULDbType.LongBinary 型、ULDbType.UniqueIdentifier 型のカラムの場合にのみ有効です。
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 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。
GetDataTypeName メソッド	指定されたカラムのプロバイダ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。
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。
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。
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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。
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。

保護メソッド

メンバ名	説明
Finalize メソッド	ULDataReader によって使用されるアンマネージ・リソースを解放します。

Depth プロパティ

現在のローのネストの深さを返します。最も外側のテーブルの深さは 0 です。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
NotOverridable Public Readonly Property Depth As Integer _
  Implements IDataReader.Depth
```

C#

```
public int Depth {get;}
```

プロパティ値

Ultra Light.NET のすべての結果セットの深さはゼロです。

例外

- ULException クラス - ULDataReader が開かれていません。

実装

IDataReader.Depth

FieldCount プロパティ

このカーソル内のカラム数を返します。

構文

Visual Basic

```
NotOverridable Public Readonly Property FieldCount As Integer _
  Implements IDataRecord.FieldCount
```

C#

```
public int FieldCount {get;}
```

戻り値

整数として返される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。

実装

IDataRecord.FieldCount

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「ColumnCount プロパティ」 148 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Readonly Property **IsBOF** As Boolean

C#

public bool **IsBOF** {get;}

プロパティ値

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。

例外

- ULException クラス - SQL エラーが発生しました。

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。

構文

Visual Basic

NotOverridable Public Readonly Property **IsClosed** As Boolean _
  Implements IDataReader.IsClosed

C#

public bool **IsClosed** {get;}

プロパティ値

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。

実装

IDataReader.IsClosed

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。

構文

Visual Basic

Public Readonly Property **IsEOF** As Boolean

Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public bool IsEOF {get;}
```

プロパティ値

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。

例外

- ULException クラス - SQL エラーが発生しました。

Item 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。C# では、このプロパティは ULDataReader クラスのインデクサです。

構文

Visual Basic

```
NotOverridable Public Readonly Property Item As Object _
  Implements IDataRecord.Item
```

C#

```
public object Item {get;}
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

備考

このメソッドは、機能的には ULDataReader.GetValue(int) メソッドと同じです。

例外

- ULException クラス - SQL エラーが発生しました。

実装

IDataRecord.Item

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetFieldType メソッド」 222 ページ
- 「Item プロパティ」 208 ページ
- 「GetValue メソッド」 235 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「FieldCount プロパティ」 205 ページ

Item 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。C# では、このプロパティは ULDataReader クラスのインデクサです。

構文

Visual Basic

```
NotOverridable Public Readonly Property Item As Object _
  Implements IDataRecord.Item
```

C#

```
public object Item {get;}
```

パラメータ

- **name**　カラムの名前。

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。

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 ) )
```

例外

- ULException クラス - SQL エラーが発生しました。

実装

IDataRecord.Item

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「Item プロパティ」 207 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetValue メソッド」 235 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「GetFieldType メソッド」 222 ページ

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。SELECT 文または CommandType.TableDirect テーブルの場合、この値は -1 です。

構文

Visual Basic

```
NotOverridable Public Readonly Property RecordsAffected As Integer _
  Implements IDataReader.RecordsAffected
```

C#

```
public int RecordsAffected {get;}
```

プロパティ値

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。

実装

IDataReader.RecordsAffected

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- CommandType.TableDirect

RowCount プロパティ

UL 拡張： カーソル内のロー数を返します。

構文

Visual Basic

```
Public Readonly Property RowCount As Integer
```

C#

```
public int RowCount {get;}
```

プロパティ値

カーソル内のロー数。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「StartSynchronizationDelete メソッド」 118 ページ
- 「StopSynchronizationDelete メソッド」 119 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。

構文

Visual Basic

Public Readonly Property **Schema** As ULCursorSchema

C#

public ULCursorSchema **Schema** {get;}

プロパティ値

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。

備考

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示します。

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「ULTableSchema クラス」 453 ページ
- 「GetSchemaTable メソッド」 229 ページ
- 「ULResultSetSchema クラス」 351 ページ

Close メソッド

カーソルを閉じます。

 Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
NotOverridable Public Sub Close() _
  Implements IDataReader.Close
```

C#

```
public void Close();
```

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。

例外

- ULException クラス - SQL エラーが発生しました。

実装

IDataReader.Close

Dispose メソッド

ULDataReader 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。

構文

Visual Basic

```
NotOverridable Public Sub Dispose() _
  Implements IDisposable.Dispose
```

C#

```
public void Dispose();
```

例外

- ULException クラス - SQL エラーが発生しました。

実装

IDisposable.Dispose

Finalize メソッド

ULDataReader によって使用されるアンマネージ・リソースを解放します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Overrides Protected Sub Finalize()
```

C#

```
protected override void Finalize();
```

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。

構文

Visual Basic

```
NotOverridable Public Function GetBoolean( _
  ByVal columnID As Integer _
) As Boolean _
  Implements IDataRecord.GetBoolean
```

C#

```
public bool GetBoolean(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

実装

IDataRecord.GetBoolean

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- Boolean
- 「FieldCount プロパティ」 205 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
NotOverridable Public Function GetByte( _
  ByVal columnID As Integer _
) As Byte _
  Implements IDataRecord.GetByte
```

C#

```
public byte GetByte(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

System.Byte 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。

実装

IDataRecord.GetByte

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- Byte
- 「FieldCount プロパティ」 205 ページ

GetBytes 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。

構文

Visual Basic

```
Overloads NotOverridable Public Function GetBytes( _
```

```
  ByVal columnID As Integer, _
  ByVal srcOffset As Long, _
  ByVal dst As Byte(), _
  ByVal dstOffset As Integer, _
  ByVal count As Integer _
) As Long _
  Implements IDataRecord.GetBytes
```

C#

```
public long GetBytes(
  int columnID,
  long srcOffset,
  byte[] dst,
  int dstOffset,
  int count
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **srcOffset**　カラム値の開始位置。ゼロが値の最初です。
- **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。

備考

null 参照 (Visual Basic の Nothing) である *dst* バッファを渡すと、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* が *dst*.Length. より長い。

 Copyright © 2007, iAnywhere Solutions, Inc.

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。

例外

- ULException クラス - SQL エラーが発生しました。

実装

IDataRecord.GetBytes

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- 「GetBytes メソッド」 215 ページ
- 「ULDbType 列挙」 245 ページ
- Byte
- 「ULException クラス」 249 ページ
- 「ULSQLCode 列挙」 366 ページ
- 「FieldCount プロパティ」 205 ページ

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。ULDbType.Binary 型、ULDbType.LongBinary 型、ULDbType.UniqueIdentifier 型のカラムの場合にのみ有効です。

構文

Visual Basic

```
Overloads Public Function GetBytes( _
  ByVal columnID As Integer _
) As Byte()
```

C#

```
public byte[] GetBytes(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

System.Bytes 配列として返されるカラム値。

Copyright © 2007, iAnywhere Solutions, Inc.

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- 「GetBytes メソッド」 213 ページ
- Byte
- 「ULDbType 列挙」 245 ページ
- 「ULDbType 列挙」 245 ページ
- 「ULDbType 列挙」 245 ページ

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。

構文

Visual Basic

```
NotOverridable Public Function GetChar( _
  ByVal columnID As Integer _
) As Char _
  Implements IDataRecord.GetChar
```

C#

```
public char GetChar(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

実装

IDataRecord.GetChar

参照

- 「ULDataReader クラス」 201 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- 「GetString メソッド」 232 ページ
- 「FieldCount プロパティ」 205 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。

構文

Visual Basic

```
NotOverridable Public Function GetChars( _
  ByVal columnID As Integer, _
  ByVal srcOffset As Long, _
  ByVal dst As Char(), _
  ByVal dstOffset As Integer, _
  ByVal count As Integer _
) As Long _
  Implements IDataRecord.GetChars
```

C#

```
public long GetChars(
  int columnID,
  long srcOffset,
  char[] dst,
  int dstOffset,
  int count
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **srcOffset**　カラム値の開始位置。ゼロが値の最初です。
- **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。

戻り値

実際にコピーされた文字数。

備考

NULL 参照 (Visual Basic の Nothing) である *dst* バッファを渡すと、GetChars はフィールドの長さを文字数として返します。

Copyright © 2007, iAnywhere Solutions, Inc.

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* が *dst*.Length. 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。

例外

- ULException クラス - SQL エラーが発生しました。

実装

IDataRecord.GetChars

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- 「ULDbType 列挙」 245 ページ
- System.Char
- 「ULException クラス」 249 ページ
- 「ULSQLCode 列挙」 366 ページ
- 「FieldCount プロパティ」 205 ページ

GetData 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。

構文

Visual Basic

```
NotOverridable Public Function GetData( _
  ByVal i As Integer _
) As IDataReader _
  Implements IDataRecord.GetData
```

C#

Copyright © 2007, iAnywhere Solutions, Inc.

```
public IDataReader GetData(
   int i
);
```

パラメータ

- **i**　整数値。

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。

実装

IDataRecord.GetData

GetDataTypeName メソッド

指定されたカラムのプロバイダのデータ型の名前を返します。

構文

Visual Basic

```
NotOverridable Public Function GetDataTypeName( _
   ByVal columnID As Integer _
) As String _
   Implements IDataRecord.GetDataTypeName
```

C#

```
public string  GetDataTypeName(
   int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

実装

IDataRecord.GetDataTypeName

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetColumnULDbType メソッド」 154 ページ
- 「FieldCount プロパティ」 205 ページ
- 「ULDbType 列挙」 245 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。

構文

Visual Basic

```
NotOverridable Public Function GetDateTime( _
  ByVal columnID As Integer _
) As Date _
  Implements IDataRecord.GetDate
```

C#

```
public DateTime GetDateTime(
  int columnID
);
```

パラメータ

- **columnID** 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。カーソルの最初のカラムの ID 値はゼロです。

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。

実装

IDataRecord.GetDateTime

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- DateTime
- 「FieldCount プロパティ」 205 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
NotOverridable Public Function GetDecimal( _
  ByVal columnID As Integer _
) As Decimal _
  Implements IDataRecord.GetDecimal
```

C#

```
public decimal GetDecimal(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

実装

IDataRecord.GetDecimal

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- Decimal
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
NotOverridable Public Function GetDouble( _
  ByVal columnID As Integer _
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
) As Double _
  Implements IDataRecord.GetDouble
```

C#

```
public double GetDouble(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

実装

IDataRecord.GetDouble

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- Double
- 「FieldCount プロパティ」 205 ページ

GetFieldType メソッド

指定されたカラムに最適な System.Type を返します。

構文

Visual Basic

```
NotOverridable Public Function GetFieldType( _
  ByVal columnID As Integer _
) As Type _
  Implements IDataRecord.GetFieldType
```

C#

```
public Type GetFieldType(
  int columnID
);
```


Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

カラムの System.Type 値。

例外

- ULException クラス - SQL エラーが発生しました。

実装

IDataRecord.GetFieldType

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetDataTypeName メソッド」 219 ページ
- 「GetColumnULDbType メソッド」 154 ページ
- Type
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
NotOverridable Public Function GetFloat( _
  ByVal columnID As Integer _
) As Single _
  Implements IDataRecord.GetFloat
```

C#

```
public float  GetFloat(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

System.Single として返されるカラム値。

Copyright © 2007, iAnywhere Solutions, Inc.

例外

- ULException クラス - SQL エラーが発生しました。

実装

IDataRecord.GetFloat

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- Single
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
NotOverridable Public Function GetGuid( _
  ByVal columnID As Integer _
) As Guid _
  Implements IDataRecord.GetGuid
```

C#

```
public Guid GetGuid(
  int columnID
);
```

パラメータ

- **columnID** 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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

Guid として返されるカラム値。

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。

例外

- ULException クラス - SQL エラーが発生しました。

Copyright © 2007, iAnywhere Solutions, Inc.

実装

IDataRecord.GetGuid

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- 「GetColumnULDbType メソッド」 154 ページ
- 「GetColumnSize メソッド」 153 ページ
- Guid
- 「ULDbType 列挙」 245 ページ
- 「ULDbType 列挙」 245 ページ
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
NotOverridable Public Function GetInt16( _
  ByVal columnID As Integer _
) As Short _
  Implements IDataRecord.GetShort
```

C#

```
public short GetInt16(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

実装

IDataRecord.GetInt16

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- Int16
- 「FieldCount プロパティ」 205 ページ

GetInt32 メソッド

指定されたカラムの値を Int32 として返します。

構文

Visual Basic

```
NotOverridable Public Function GetInt32( _
   ByVal columnID As Integer _
) As Integer _
   Implements IDataRecord.GetInteger
```

C#

```
public int GetInt32(
   int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

実装

IDataRecord.GetInt32

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- Int32
- 「FieldCount プロパティ」 205 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

GetInt64 メソッド

指定されたカラムの値を Int64 として返します。

構文

Visual Basic

```
NotOverridable Public Function GetInt64( _
  ByVal columnID As Integer _
) As Long _
  Implements IDataRecord.GetLong
```

C#

```
public long GetInt64(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

実装

IDataRecord.GetInt64

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- Int64
- 「FieldCount プロパティ」 205 ページ

GetName メソッド

指定されたカラムの名前を返します。

構文

Visual Basic

```
NotOverridable Public Function GetName( _
  ByVal columnID As Integer _
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
) As String _
  Implements IDataRecord.GetName
```

C#

```
public string  GetName(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

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。

例外

- ULException クラス - SQL エラーが発生しました。

実装

IDataRecord.GetName

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「FieldCount プロパティ」 205 ページ
- 「GetSchemaTable メソッド」 229 ページ
- 「GetColumnName メソッド」 150 ページ
- 「FieldCount プロパティ」 205 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。

構文

Visual Basic

```
NotOverridable Public Function GetOrdinal( _
  ByVal columnName As String _
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
) As Integer _
  Implements IDataRecord.GetOrdinal
```

C#

```
public int GetOrdinal(
  string  columnName
);
```

パラメータ

- **columnName**　カラムの名前。

戻り値

指定されたカラムのカラム ID。

備考

カラム ID の範囲は、0 ー 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。

例外

- ULException クラス - SQL エラーが発生しました。

実装

IDataRecord.GetOrdinal

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetSchemaTable メソッド」 229 ページ
- 「FieldCount プロパティ」 205 ページ
- 「GetColumnID メソッド」 149 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
NotOverridable Public Function GetSchemaTable() As DataTable _
  Implements IDataReader.GetSchemaTable
```

C#

```
public 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。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

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 入力可能である場合は true、null 入力不可能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。

実装

IDataReader.GetSchemaTable

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「Schema プロパティ」 210 ページ
- 「FieldCount プロパティ」 205 ページ
- 「ULDbType 列挙」 245 ページ
- 「ULDbType 列挙」 245 ページ
- 「ULDbType 列挙」 245 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULDbType 列挙」 245 ページ
- 「ULDbType 列挙」 245 ページ
- 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。

構文

Visual Basic

```
NotOverridable Public Function GetString( _
  ByVal columnID As Integer _
) As String _
  Implements IDataRecord.GetString
```

C#

```
public string  GetString(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

実装

IDataRecord.GetString

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- String
- 「FieldCount プロパティ」 205 ページ

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Function GetTimeSpan( _
  ByVal columnID As Integer _
) As TimeSpan
```

C#

```
public TimeSpan GetTimeSpan(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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 クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- TimeSpan
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
Public Function GetUInt16( _
  ByVal columnID As Integer _
) As UInt16
```

C#

```
public ushort GetUInt16(
  int columnID
);
```

Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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 クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- UInt16
- 「FieldCount プロパティ」 205 ページ

GetUInt32 メソッド

指定されたカラムの値を UInt32 として返します。

構文

Visual Basic

```
Public Function GetUInt32( _
  ByVal columnID As Integer _
) As UInt32
```

C#

```
public uint GetUInt32(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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 クラス」 201 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- UInt32
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
Public Function GetUInt64( _
  ByVal columnID As Integer _
) As UInt64
```

C#

```
public ulong GetUInt64(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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 クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- UInt64
- 「FieldCount プロパティ」 205 ページ

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
NotOverridable Public Function GetValue( _
  ByVal columnID As Integer _
) As Object _
  Implements IDataRecord.GetValue
```

C#

```
public object GetValue(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

備考

このメソッドは、機能的には ULDataReader.this[int] と同じです。

例外

- ULException クラス - SQL エラーが発生しました。

実装

IDataRecord.GetValue

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- 「FieldCount プロパティ」 205 ページ
- 「Item プロパティ」 207 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。

構文

Visual Basic

```
NotOverridable Public Function GetValues( _
  ByVal values As Object() _
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
) As Integer _
  Implements IDataRecord.GetValues
```

C#

```
public int GetValues(
  object[] values
);
```

パラメータ

- **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。

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。

例外

- ULException クラス - SQL エラーが発生しました。
- System.ArgumentNullException - *values* 配列が NULL であるか、長さがゼロです。

実装

IDataRecord.GetValues

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「FieldCount プロパティ」 205 ページ
- 「GetFieldType メソッド」 222 ページ
- 「GetValue メソッド」 235 ページ
- オブジェクト

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
NotOverridable Public Function IsDBNull( _
  ByVal columnID As Integer _
) As Boolean _
  Implements IDataRecord.IsDBNull
```

C#

```
public bool IsDBNull(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

実装

IDataRecord.IsDBNull

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- 「FieldCount プロパティ」 205 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。

構文

Visual Basic

```
Public Sub MoveAfterLast()
```

C#

```
public void MoveAfterLast();
```


Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Sub **MoveBeforeFirst()**

C#

public void **MoveBeforeFirst();**

例外

- ULException クラス - SQL エラーが発生しました。

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。

構文

Visual Basic

Public Function **MoveFirst()** As Boolean

C#

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Function **MoveLast()** As Boolean

C#

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。

構文

Visual Basic

Public Function **MoveNext()** As Boolean

C#

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。

備考

このメソッドは、ULDataReader.Read メソッドと同じです。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「Read メソッド」 242 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Public Function MovePrevious() As Boolean
```

C#

```
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。

構文

Visual Basic

```
Public Function MoveRelative( _
  ByVal offset As Integer _
) As Boolean
```

C#

```
public bool MoveRelative(
  int offset
);
```

パラメータ

- **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。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「IsEOF プロパティ」 206 ページ
- 「IsBOF プロパティ」 206 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。

構文

Visual Basic

```
NotOverridable Public Function NextResult() As Boolean _
  Implements IDataReader.NextResult
```

C#

```
public 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。

備考

UL 拡張： Ultra Light.NET ではバッチ SQL 文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。

例外

- ULException クラス - ULDataReader が開かれていません。

実装

IDataReader.NextResult

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。

構文

Visual Basic

Copyright © 2007, iAnywhere Solutions, Inc.

```
NotOverridable Public Function Read() As Boolean _
  Implements IDataReader.Read
```

C#

```
public 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。

備考

このメソッドは、ULDataReader.MoveNext メソッドと同じです。

例外

- ULException クラス - SQL エラーが発生しました。

実装

IDataReader.Read

参照

- 「ULDataReader クラス」 201 ページ
- 「ULDataReader メンバ」 202 ページ
- 「MoveNext メソッド」 240 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULDateOrder enumeration

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。

構文

Visual Basic

Public Enum **ULDateOrder**

C#

public enum **ULDateOrder**

メンバ名

メンバ名	説明
DMY	日 (日、月、年の順に示す)
MDY	月 (月、日、年の順に示す)
YMD	年 (年、月、日の順に示す)

参照

- 「DateOrder プロパティ」 140 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULDbType 列挙

Ultra Light.NET データベースのデータ型を列挙します。

構文

Visual Basic

Public Enum **ULDbType**

C#

public enum **ULDbType**

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。

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

Copyright © 2007, iAnywhere Solutions, Inc.

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
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
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
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。

メンバ名

メンバ名	説明
BigInt	符号付き 64 ビット整数値
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。
Bit	1 ビット・フラグ
Char	指定された長さの文字データ。Ultra Light.NET では、この型は常にユニコード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。
Date	日付情報
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。
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。
Double	倍精度浮動小数点数 (8 バイト)

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。
Integer	符号なし 32 ビット整数値
LongBinary	可変長のバイナリ・データ
LongVarchar	可変長の文字データ。Ultra Light.NET では、この型は常にユニコード文字をサポートします。
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。
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。
SmallInt	符号付き 16 ビット整数値
Time	時刻情報
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。
TinyInt	符号なし 8 ビット整数値
UniqueIdentifier	ユニバーサル・ユニーク識別子 (UUID/GUID)
UnsignedBigInt	符号なし 64 ビット整数値
UnsignedInt	符号なし 32 ビット整数値
UnsignedSmallInt	符号なし 16 ビット整数値
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。
VarChar	指定された最大長の文字データ。Ultra Light.NET では、この型は常にユニコード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。

参照

- 「GetFieldType メソッド」 222 ページ
- 「GetDataTypeName メソッド」 219 ページ
- 「GetColumnULDbType メソッド」 154 ページ
- Byte
- Guid
- Boolean
- String
- DateTime
- Single

Copyright © 2007, iAnywhere Solutions, Inc.

- Int64
- Int32
- Int16
- TimeSpan
- UInt64
- UInt32
- UInt16

Copyright © 2007, iAnywhere Solutions, Inc.

ULException クラス

Ultra Light.NET データベースによって返される SQL エラーを示します。

構文

Visual Basic

```
NotInheritable Public Class ULException
  Inherits ApplicationException
```

C#

```
public sealed class ULException :
  ApplicationException
```

備考

NativeError にエラーが返されたことを示す SQL コード。

.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 メンバ」 249 ページ
- 「NativeError プロパティ」 251 ページ

ULException メンバ

パブリック・コンストラクタ

メンバ名	説明
ULException コンストラクタ	指定されたエラー・コードを持つ ULException を作成します。

パブリック・プロパティ

メンバ名	説明
HelpLink (Exception から継承)	この例外に関連付けられたヘルプ・ファイルへのリンクを取得または設定します。
InnerException (Exception から継承)	現在の例外を引き起こす System.Exception インスタンスを取得します。
Message (Exception から継承)	現在の例外を説明するメッセージを取得します。
NativeError プロパティ	データベースによって返される SQL コードを返します。
Source プロパティ	エラーを生成したプロバイダの名前を返します。
StackTrace (Exception から継承)	現在の例外がスローされた時点のコール・スタックのフレームを表す文字列を取得します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
TargetSite (Exception から継承)	現在の例外をスローしたメソッドを取得します。

パブリック・メソッド

メンバ名	説明
GetBaseException (Exception から継承)	派生クラスで上書きされる場合は、後続の 1 つ以上の例外の根本原因となる System.Exception を返します。
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。
ToString (Exception から継承)	現在の例外を表す文字列を作成して返します。

ULException コンストラクタ

指定されたエラー・コードを持つ ULException を作成します。

構文

Visual Basic

```
Overloads Public Sub New( _
  ByVal code As ULSQLCode, _
  ByVal s1 As String, _
  ByVal s2 As String, _
  ByVal s3 As String _
)
```

C#

```
public ULException(
  ULSQLCode code,
  string  s1,
  string  s2,
  string  s3
);
```

パラメータ

- **code**　例外のコード。
- **s1**　フォーマットされたメッセージの 1 番目の文字列
- **s2**　フォーマットされたメッセージの 2 番目の文字列
- **s3**　フォーマットされたメッセージの 3 番目の文字列

Copyright © 2007, iAnywhere Solutions, Inc.

備考

指定された iAnywhere.Data.UltraLite.ULSQLCode に対応するメッセージ文字列は、**iAnywhere.Data.UltraLite.resources** アセンブリから取得されます。リソースは、CultureInfo.CurrentUICulture、CultureInfo.CurrentCulture、国 "EN" の順序で国別に検索します。

参照

- 「ULException クラス」 249 ページ
- 「ULException メンバ」 249 ページ
- 「ULSQLCode 列挙」 366 ページ
- System.Globalization.CultureInfo.CurrentUICulture
- System.Globalization.CultureInfo.CurrentCulture

NativeError プロパティ

データベースによって返される SQL コードを返します。

構文

Visual Basic

Public Readonly Property **NativeError** As ULSQLCode

C#

public ULSQLCode **NativeError** {get;}

プロパティ値

データベースによって返される ULSQLCode 値。

参照

- 「ULException クラス」 249 ページ
- 「ULException メンバ」 249 ページ
- 「ULSQLCode 列挙」 366 ページ

Source プロパティ

エラーを生成したプロバイダの名前を返します。

構文

Visual Basic

Public Readonly Property **Source** As String

C#

public string **Source** {get;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

プロバイダが Ultra Light.NET であることを示す文字列値。

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。

構文

Visual Basic

```
Overrides Public Sub GetObjectData( _
  ByVal info As System.Runtime.Serialization.SerializationInfo, _
  ByVal context As System.Runtime.Serialization.StreamingContext _
) _
  Implements ISerializable.GetObjectData
```

C#

```
public override void GetObjectData(
  System.Runtime.Serialization.SerializationInfo info,
  System.Runtime.Serialization.StreamingContext context
);
```

パラメータ

- **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。

実装

ISerializable.GetObjectData

Copyright © 2007, iAnywhere Solutions, Inc.

ULFileTransfer クラス

UL 拡張： Mobile Link サーバを使用して、リモート・データベースからファイルを転送します。

構文

Visual Basic

NotInheritable Public Class **ULFileTransfer**

C#

public sealed class **ULFileTransfer**

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 メンバ」 253 ページ
- 「RuntimeType プロパティ」 183 ページ
- 「FileName プロパティ」 259 ページ
- 「Stream プロパティ」 261 ページ
- 「UserName プロパティ」 263 ページ
- 「Version プロパティ」 264 ページ

ULFileTransfer メンバ

パブリック・コンストラクタ

メンバ名	説明
ULFileTransfer コンストラクタ	ULFileTransfer オブジェクトを初期化します。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。

パブリック・プロパティ

メンバ名	説明
AuthenticationParms プロパティ	カスタム・ユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	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・ユーザ認証同期スクリプトからの戻り値を返します。
DestinationFileName 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。
DestinationPath 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。
DownloadedFile 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。
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。
ForceDownload プロパティ	ファイルが既に存在する場合に、ファイルのダウンロードを強制するかどうかを指定します。
Password プロパティ	UserName で指定されたユーザの Mobile Link パスワードです。
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を指定します。
UserName プロパティ	Mobile Link サーバが Mobile Link クライアントをユニークに識別するユーザ名です。
Version プロパティ	使用する同期スクリプトを指定します。

パブリック・メソッド

メンバ名	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 メソッド	指定されたリスナ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。

Copyright © 2007, iAnywhere Solutions, Inc.

ULFileTransfer コンストラクタ

ULFileTransfer オブジェクトを初期化します。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。

構文

Visual Basic

Public Sub **New()**

C#

public **ULFileTransfer();**

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。

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 クラス」 253 ページ
- 「ULFileTransfer メンバ」 253 ページ
- 「DownloadFile メソッド」 265 ページ
- 「RuntimeType プロパティ」 183 ページ
- 「FileName プロパティ」 259 ページ
- 「Stream プロパティ」 261 ページ
- 「UserName プロパティ」 263 ページ
- 「Version プロパティ」 264 ページ

AuthenticationParms プロパティ

カスタム・ユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。

構文

Visual Basic

Public Property **AuthenticationParms** As String()

C#

public string [] **AuthenticationParms** {get;set;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

それぞれに認証パラメータが格納された、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、認証パラメータは指定されません。

備考

最初の 255 文字のみが使用されます。また、各文字列は 128 文字以下であることが必要です (長すぎる文字列は、Mobile Link サーバに送信されるときにトランケートされます)。

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。

構文

Visual Basic

Public Readonly Property **AuthStatus** As ULAuthStatusCode

C#

public ULAuthStatusCode **AuthStatus** {get;}

プロパティ値

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

- 「ULFileTransfer クラス」 253 ページ
- 「ULFileTransfer メンバ」 253 ページ
- 「ULAuthStatusCode 列挙」 56 ページ

AuthValue プロパティ

カスタム・ユーザ認証同期スクリプトからの戻り値を返します。

構文

Visual Basic

Public Readonly Property **AuthValue** As Long

C#

public long **AuthValue** {get;}

プロパティ値

カスタム・ユーザ認証同期スクリプトから返された long integer。

 Copyright © 2007, iAnywhere Solutions, Inc.

DestinationFileName 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。

構文

Visual Basic

Public Property **DestinationFileName** As String

C#

public string **DestinationFileName** {get;set;}

プロパティ値

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 クラス」 253 ページ
- 「ULFileTransfer メンバ」 253 ページ
- 「FileName プロパティ」 259 ページ

DestinationPath 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。

構文

Visual Basic

Public Property **DestinationPath** As String

C#

public string **DestinationPath** {get;set;}

プロパティ値

ファイルのダウンロード先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。

備考

デフォルトのディレクトリは、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、DestinationPath 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、DestinationPath 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。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULFileTransfer クラス」 253 ページ
- 「ULFileTransfer メンバ」 253 ページ
- 「ForceDownload プロパティ」 259 ページ
- 「DownloadFile メソッド」 265 ページ

DownloadedFile 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。

構文

Visual Basic

Public Readonly Property **DownloadedFile** As Boolean

C#

public bool **DownloadedFile** {get;}

プロパティ値

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。

備考

DownloadFile() 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、DownloadedFile は false に設定されます。エラーが発生して DownloadFile() が false を返した場合は、DownloadedFile は false に設定されます。

参照

- 「ULFileTransfer クラス」 253 ページ
- 「ULFileTransfer メンバ」 253 ページ
- 「DownloadFile メソッド」 265 ページ
- 「DownloadFile メソッド」 265 ページ

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。

構文

Visual Basic

Public Readonly Property **FileAuthCode** As UInt16

C#

public ushort **FileAuthCode** {get;}

 Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

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。

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。

構文

Visual Basic

Public Property **FileName** As String

C#

public string **FileName** {get;set;}

プロパティ値

Mobile Link サーバ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。

備考

FileName は、Mobile Link を実行するサーバ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 サブディレクトリで、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 サーバの -ftr オプションで指定)。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。そのような情報を含めると、ファイルが見からなくなります。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 クラス」 253 ページ
- 「ULFileTransfer メンバ」 253 ページ
- 「DestinationFileName プロパティ」 257 ページ
- 「DownloadFile メソッド」 265 ページ

ForceDownload プロパティ

ファイルが既に存在する場合に、ファイルのダウンロードを強制するかどうかを指定します。

構文

Visual Basic

Public Property **ForceDownload** As Boolean

C#

public bool **ForceDownload** {get;set;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

ファイルが存在する場合に、ファイルのダウンロードを強制する場合は true、通常のダウンロード・チェックを実行する場合は false。デフォルトは false です。

備考

ForceDownload が true であると、変更が加えられていない場合でもファイルは常にダウンロードされ、部分的なダウンロードは破棄されます。ForceDownload が false であると、サーバのバージョンとクライアントのバージョンが異なる場合にのみ、ファイルはダウンロードされます。クライアントまたはサーバでファイルが変更される場合に ForceDownload を true に設定すると、クライアント・バージョンがサーバ・バージョンで上書きされることに注意してください。

参照

- 「ULFileTransfer クラス」 253 ページ
- 「ULFileTransfer メンバ」 253 ページ
- 「ResumePartialDownload プロパティ」 261 ページ
- 「DownloadFile メソッド」 265 ページ

Password プロパティ

UserName で指定されたユーザの Mobile Link パスワードです。

構文

Visual Basic

```
Public Property Password As String
```

C#

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

プロパティ値

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。

備考

Mobile Link ユーザ名とパスワードは他のデータベース・ユーザ 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 サーバに対して識別し、認証するために使用されます。

参照

- 「ULFileTransfer クラス」 253 ページ
- 「ULFileTransfer メンバ」 253 ページ
- 「UserName プロパティ」 263 ページ
- 「DownloadFile メソッド」 265 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Property **ResumePartialDownload** As Boolean

C#

public bool **ResumePartialDownload** {get;set;}

プロパティ値

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。

備考

Ultra Light.NET では、通信エラーや、ユーザによる 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 クラス」 253 ページ
- 「ULFileTransfer メンバ」 253 ページ
- 「ForceDownload プロパティ」 259 ページ
- 「DownloadFile メソッド」 265 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。

構文

Visual Basic

Public Property **Stream** As ULStreamType

C#

public ULStreamType **Stream** {get;set;}

プロパティ値

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。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

ほとんどの同期ストリームでは、Mobile Link サーバ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が必要です。これらのパラメータは、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

- 「ULFileTransfer クラス」 253 ページ
- 「ULFileTransfer メンバ」 253 ページ
- 「ULStreamType 列挙」 395 ページ
- 「StreamParms プロパティ」 263 ページ
- 「DownloadFile メソッド」 265 ページ
- 「ULStreamType 列挙」 395 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。

構文

Visual Basic

Public Readonly Property **StreamErrorCode** As ULStreamErrorCode

C#

public ULStreamErrorCode **StreamErrorCode** {get;}

プロパティ値

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。

参照

- 「ULFileTransfer クラス」 253 ページ
- 「ULFileTransfer メンバ」 253 ページ
- 「ULStreamErrorCode 列挙」 376 ページ
- 「ULStreamErrorCode 列挙」 376 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。

 Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **StreamErrorSystem** As Integer

C#

public int **StreamErrorSystem** {get;}

プロパティ値

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を指定します。

構文

Visual Basic

Public Property **StreamParms** As String

C#

public string **StreamParms** {get;set;}

プロパティ値

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を指定する文字列。デフォルト値は null 参照 (Visual Basic の Nothing) です。

備考

特定のストリームのタイプの設定方法については、「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」 『Mobile Link - クライアント管理』を参照してください。

StreamParms は、同期ストリームに使用されるすべてのパラメータが含まれている文字列です。パラメータは、名前=値の組み合わせ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 クラス」 253 ページ
- 「ULFileTransfer メンバ」 253 ページ
- 「Stream プロパティ」 261 ページ
- 「DownloadFile メソッド」 265 ページ
- 「ULStreamType 列挙」 395 ページ

UserName プロパティ

Mobile Link サーバが Mobile Link クライアントをユニークに識別するユーザ名です。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **UserName** As String

C#

public string **UserName** {get;set;}

プロパティ値

ユーザ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。

備考

Mobile Link サーバは、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 ユーザ名とパスワードは他のデータベース・ユーザ 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 サーバに対して識別し、認証するために使用されます。

参照

- 「ULFileTransfer クラス」 253 ページ
- 「ULFileTransfer メンバ」 253 ページ
- 「Password プロパティ」 260 ページ
- 「DownloadFile メソッド」 265 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。

構文

Visual Basic

Public Property **Version** As String

C#

public string **Version** {get;set;}

プロパティ値

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。

備考

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 クラス」 253 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULFileTransfer メンバ」 253 ページ
- 「DownloadFile メソッド」 265 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。

構文

Visual Basic

Overloads Public Function **DownloadFile()** As Boolean

C#

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 サーバによって ULFileTransfer.DestinationPath 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。ウンロードに影響を及ぼすその他のプロパティは、ULFileTransfer.DestinationFileName、ULFileTransfer.AuthenticationParmsULFileTransfer.ForceDownload、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対象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.DownloadedFile、ULFileTransfer.StreamErrorCode、および ULFileTransfer.StreamErrorSystem でレポートされます。

参照

- 「ULFileTransfer クラス」 253 ページ
- 「ULFileTransfer メンバ」 253 ページ
- 「DownloadFile メソッド」 266 ページ
- 「FileName プロパティ」 259 ページ
- 「DestinationPath プロパティ」 257 ページ
- 「Stream プロパティ」 261 ページ
- 「UserName プロパティ」 263 ページ
- 「Password プロパティ」 260 ページ
- 「Version プロパティ」 264 ページ
- 「DestinationFileName プロパティ」 257 ページ
- 「AuthenticationParms プロパティ」 255 ページ
- 「ForceDownload プロパティ」 259 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ResumePartialDownload プロパティ」 261 ページ
- 「AuthStatus プロパティ」 256 ページ
- 「AuthValue プロパティ」 256 ページ
- 「FileAuthCode プロパティ」 258 ページ
- 「DownloadedFile プロパティ」 258 ページ
- 「StreamErrorCode プロパティ」 262 ページ
- 「StreamErrorSystem プロパティ」 262 ページ
- 「StreamErrorCode プロパティ」 262 ページ

DownloadFile メソッド

指定されたリスナ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。

構文

Visual Basic

```
Overloads Public Function DownloadFile( _
  ByVal listener As ULFileTransferProgressListener _
) As Boolean
```

C#

```
public bool DownloadFile(
  ULFileTransferProgressListener listener
);
```

パラメータ

- **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 サーバによって ULFileTransfer.DestinationPath 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。ウンロードに影響を及ぼすその他のプロパティは、ULFileTransfer.DestinationFileName、ULFileTransfer.AuthenticationParmsULFileTransfer.ForceDownload、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対象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.DownloadedFile、ULFileTransfer.StreamErrorCode、および ULFileTransfer.StreamErrorSystem でレポートされます。

Copyright © 2007, iAnywhere Solutions, Inc.

エラーが派生すると、リスナにはデータが送信されないことがあります。

参照

- 「ULFileTransfer クラス」 253 ページ
- 「ULFileTransfer メンバ」 253 ページ
- 「FileName プロパティ」 259 ページ
- 「DestinationPath プロパティ」 257 ページ
- 「Stream プロパティ」 261 ページ
- 「UserName プロパティ」 263 ページ
- 「Password プロパティ」 260 ページ
- 「Version プロパティ」 264 ページ
- 「DestinationFileName プロパティ」 257 ページ
- 「AuthenticationParms プロパティ」 255 ページ
- 「ForceDownload プロパティ」 259 ページ
- 「ResumePartialDownload プロパティ」 261 ページ
- 「AuthStatus プロパティ」 256 ページ
- 「AuthValue プロパティ」 256 ページ
- 「FileAuthCode プロパティ」 258 ページ
- 「DownloadedFile プロパティ」 258 ページ
- 「StreamErrorCode プロパティ」 262 ページ
- 「StreamErrorSystem プロパティ」 262 ページ
- 「StreamErrorCode プロパティ」 262 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Class **ULFileTransferProgressData**

C#

public class **ULFileTransferProgressData**

参照

- 「ULFileTransferProgressData メンバ」 268 ページ
- 「ULFileTransferProgressListener インタフェース」 271 ページ

ULFileTransferProgressData メンバ

パブリックの静的フィールド (共有)

メンバ名	説明
FLAG_IS_BLOCKING フィールド	Mobile Link サーバ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

パブリック・プロパティ

メンバ名	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。
ResumedAtSize プロパティ	転送が再開されるファイル内の場所を返します。

FLAG_IS_BLOCKING フィールド

Mobile Link サーバ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

構文

Visual Basic

Public Shared **FLAG_IS_BLOCKING** As Integer

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public const int **FLAG_IS_BLOCKING**;

BytesReceived プロパティ

現在までに受信したバイト数を返します。

構文

Visual Basic

Public Readonly Property **BytesReceived** As UInt64

C#

public ulong **BytesReceived** {get;}

プロパティ値

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。

構文

Visual Basic

Public Readonly Property **FileSize** As UInt64

C#

public ulong **FileSize** {get;}

プロパティ値

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。

構文

Visual Basic

Public Readonly Property **Flags** As Integer

C#

public int **Flags** {get;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

フラグの組み合わせを保持する整数。

参照

- 「ULFileTransferProgressData クラス」 268 ページ
- 「ULFileTransferProgressData メンバ」 268 ページ
- 「FLAG_IS_BLOCKING フィールド」 268 ページ

ResumedAtSize プロパティ

転送が再開されるファイル内の場所を返します。

構文

Visual Basic

Public Readonly Property **ResumedAtSize** As UInt64

C#

public ulong **ResumedAtSize** {get;}

プロパティ値

以前に転送されたバイト数。

Copyright © 2007, iAnywhere Solutions, Inc.

ULFileTransferProgressListener インタフェース

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・インタフェースです。

構文

Visual Basic

Public Interface **ULFileTransferProgressListener**

C#

public interface **ULFileTransferProgressListener**

参照

- 「ULFileTransferProgressListener メンバ」 271 ページ
- 「DownloadFile メソッド」 266 ページ

ULFileTransferProgressListener メンバ

パブリック・メソッド

メンバ名	説明
FileTransferProgressed メソッド	ユーザ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。このメソッドは true を返して転送をキャンセルするか、false を返して続行する必要があります。

FileTransferProgressed メソッド

ユーザ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。このメソッドは true を返して転送をキャンセルするか、false を返して続行する必要があります。

構文

Visual Basic

```
Public Function FileTransferProgressed( _
  ByVal data As ULFileTransferProgressData _
) As Boolean
```

C#

```
public bool FileTransferProgressed(
  ULFileTransferProgressData data
);
```

パラメータ

- **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。

Copyright © 2007, iAnywhere Solutions, Inc.

戻り値

このメソッドは true を返して転送をキャンセルするか、false を返して続行する必要があります。

備考

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。

参照

- 「ULFileTransferProgressListener インタフェース」 271 ページ
- 「ULFileTransferProgressListener メンバ」 271 ページ
- 「ULFileTransferProgressData クラス」 268 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

NotInheritable Public Class **ULIndexSchema**

C#

public sealed class **ULIndexSchema**

備考

このクラスにはコンストラクタがありません。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

- 「ULIndexSchema メンバ」 273 ページ
- 「PrimaryKey プロパティ」 456 ページ
- 「GetIndex メソッド」 459 ページ
- 「GetOptimalIndex メソッド」 461 ページ
- 「ULTableSchema クラス」 453 ページ

ULIndexSchema メンバ

パブリック・プロパティ

メンバ名	説明
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 入力可能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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	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。

パブリック・メソッド

メンバ名	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。

構文

Visual Basic

Public Readonly Property **ColumnCount** As Short

C#

public short **ColumnCount** {get;}

プロパティ値

インデックス内のカラム数。

備考

インデックス内のカラム序数の範囲は、1 ー ColumnCount です。

カラムの序数とカウントは、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序数は、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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **IsForeignKey** As Boolean

C#

public bool **IsForeignKey** {get;}

プロパティ値

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。

備考

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。

例外

- ULException クラス - SQL エラーが発生しました。

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。

構文

Visual Basic

Public Readonly Property **IsForeignKeyCheckOnCommit** As Boolean

C#

public bool **IsForeignKeyCheckOnCommit** {get;}

プロパティ値

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。

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)。

参照

- 「ULIndexSchema クラス」 273 ページ
- 「ULIndexSchema メンバ」 273 ページ
- 「IsForeignKey プロパティ」 274 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

IsForeignKeyNullable プロパティ

外部キーが null 入力可能であるかどうかをチェックします。

構文

Visual Basic

Public Readonly Property **IsForeignKeyNullable** As Boolean

C#

public bool **IsForeignKeyNullable** {get;}

プロパティ値

外部キーが null 入力可能な場合は 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 (インデックスが外部キーではない場合を含む)。

参照

- 「ULIndexSchema クラス」 273 ページ
- 「ULIndexSchema メンバ」 273 ページ
- 「IsForeignKey プロパティ」 274 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。

構文

Visual Basic

Public Readonly Property **IsOpen** As Boolean

C#

public bool **IsOpen** {get;}

プロパティ値

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **IsPrimaryKey** As Boolean

C#

public bool **IsPrimaryKey** {get;}

プロパティ値

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。

備考

プライマリ・キー内のカラムでは null は許可されません。

例外

- ULException クラス - SQL エラーが発生しました。

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。

構文

Visual Basic

Public Readonly Property **IsUniqueIndex** As Boolean

C#

public bool **IsUniqueIndex** {get;}

プロパティ値

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。

備考

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。

例外

- ULException クラス - SQL エラーが発生しました。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **IsUniqueKey** As Boolean

C#

public bool **IsUniqueKey** {get;}

プロパティ値

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。

備考

ユニーク・キー内のカラムでは null は許可されません。

例外

- ULException クラス - SQL エラーが発生しました。

Name プロパティ

インデックスの名前を返します。

構文

Visual Basic

Public Readonly Property **Name** As String

C#

public string **Name** {get;}

プロパティ値

インデックスの名前を指定する文字列。

例外

- ULException クラス - SQL エラーが発生しました。

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。

構文

Visual Basic

Public Readonly Property **ReferencedIndexName** As String

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public string **ReferencedIndexName** {get;}

プロパティ値

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。

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)。

参照

- 「ULIndexSchema クラス」 273 ページ
- 「ULIndexSchema メンバ」 273 ページ
- 「IsForeignKey プロパティ」 274 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。

構文

Visual Basic

Public Readonly Property **ReferencedTableName** As String

C#

public string **ReferencedTableName** {get;}

プロパティ値

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。

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)。

参照

- 「ULIndexSchema クラス」 273 ページ
- 「ULIndexSchema メンバ」 273 ページ
- 「IsForeignKey プロパティ」 274 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Function GetColumnName( _
  ByVal colOrdinalInIndex As Short _
) As String
```

C#

```
public string  GetColumnName(
  short colOrdinalInIndex
);
```

パラメータ

- **colOrdinalInIndex**　インデックス内の対象のカラムの序数。値は、[1,ColumnCount] の範囲内であることが必要です。

戻り値

カラムの名前。

備考

カラムの序数とカウントは、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序数は、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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULIndexSchema クラス」 273 ページ
- 「ULIndexSchema メンバ」 273 ページ
- 「ColumnCount プロパティ」 274 ページ
- 「ColumnCount プロパティ」 274 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。

構文

Visual Basic

```
Public Function IsColumnDescending( _
  ByVal name As String _
) As Boolean
```

C#

```
public bool IsColumnDescending(
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
  string name
);
```

パラメータ

- **name**　カラムの名前。

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 クラス」 273 ページ
- 「ULIndexSchema メンバ」 273 ページ
- 「GetColumnName メソッド」 279 ページ
- 「ColumnCount プロパティ」 274 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULInfoMessageEventArgs クラス

ULConnection.InfoMessage イベントのデータを提供します。

構文

Visual Basic

NotInheritable Public Class **ULInfoMessageEventArgs**
  Inherits EventArgs

C#

public sealed class **ULInfoMessageEventArgs** :
  EventArgs

参照

- 「ULInfoMessageEventArgs メンバ」 282 ページ
- 「InfoMessage イベント」 121 ページ

ULInfoMessageEventArgs メンバ

パブリック・プロパティ

メンバ名	説明
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 SQL コードです。
Source プロパティ	メッセージを返す ADO.NET データ・プロバイダの名前です。

パブリック・メソッド

メンバ名	説明
ToString メソッド	ULConnection.InfoMessage イベントの文字列表現を返します。

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。

構文

Visual Basic

Public Readonly Property **Message** As String

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public string **Message** {get;}

プロパティ値

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 SQL コードです。

構文

Visual Basic

Public Readonly Property **NativeError** As ULSQLCode

C#

public ULSQLCode **NativeError** {get;}

プロパティ値

情報メッセージまたは警告の ULSQLCode 値。

参照

- 「ULInfoMessageEventArgs クラス」 282 ページ
- 「ULInfoMessageEventArgs メンバ」 282 ページ
- 「ULSQLCode 列挙」 366 ページ

Source プロパティ

メッセージを返す ADO.NET データ・プロバイダの名前です。

構文

Visual Basic

Public Readonly Property **Source** As String

C#

public string **Source** {get;}

プロパティ値

文字列 "Ultra Light.NET Data Provider"。

Copyright © 2007, iAnywhere Solutions, Inc.

ToString メソッド

ULConnection.InfoMessage イベントの文字列表現を返します。

構文

Visual Basic

Overrides Public Function **ToString()** As String

C#

public override string **ToString();**

プロパティ値

文字列 "Ultra Light.NET Data Provider"。

戻り値

情報メッセージまたは警告メッセージの文字列。

参照

- 「ULInfoMessageEventArgs クラス」 282 ページ
- 「ULInfoMessageEventArgs メンバ」 282 ページ
- 「InfoMessage イベント」 121 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

ULInfoMessageEventHandler 委任

ULConnection.InfoMessage イベントを処理するメソッドを示します。

構文

Visual Basic

```
Public Delegate Sub ULInfoMessageEventHandler( _
  ByVal obj As Object, _
  ByVal args As ULInfoMessageEventArgs _
)
```

C#

```
public delegate void ULInfoMessageEventHandler(
  object obj,
  ULInfoMessageEventArgs args
);
```

パラメータ

- **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

- 「InfoMessage イベント」 121 ページ
- 「ULInfoMessageEventArgs クラス」 282 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULParameter クラス

ULCommand のパラメータを示します。

構文

Visual Basic

```
NotInheritable Public Class ULParameter
  Inherits MarshalByRefObject
```

C#

```
public sealed class ULParameter :
  MarshalByRefobject
```

備考

ULParameter オブジェクトは、その多数のコンストラクタ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 ) )

// C#
ULParameter p = new ULParameter( "", (object)0 );
```

パラメータ (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は、位置パラメータとして扱われ、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、パラメータは IN パラメータ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。

実装： System.Data.IDbDataParameter、System.Data.IDataParameter

参照

- 「ULParameter メンバ」 287 ページ
- 「ULCommand クラス」 57 ページ
- 「CreateParameter メソッド」 70 ページ
- 「ULParameter コンストラクタ」 289 ページ
- 「Parameters プロパティ」 67 ページ
- 「Value プロパティ」 301 ページ
- IDbDataParameter
- IDataParameter

Copyright © 2007, iAnywhere Solutions, Inc.

ULParameter メンバ

パブリック・コンストラクタ

メンバ名	説明
ULParameter コンストラクタ	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 コンストラクタ	ULParameter オブジェクトを、指定されたパラメータ名と値で初期化します。
ULParameter コンストラクタ	ULParameter オブジェクトを、指定されたパラメータ名とデータ型で初期化します。このコンストラクタの使用はおすすめしません。これは、他のデータ・プロバイダとの互換性のために用意されています。
ULParameter コンストラクタ	ULParameter オブジェクトを、指定されたパラメータ名とデータ型で初期化します。このコンストラクタの使用はおすすめしません。これは、他のデータ・プロバイダとの互換性のために用意されています。
ULParameter コンストラクタ	ULParameter オブジェクトを、指定されたパラメータ名、データ型、長さで初期化します。このコンストラクタの使用はおすすめしません。これは、他のデータ・プロバイダとの互換性のために用意されています。
ULParameter コンストラクタ	指定されたパラメータ名、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。このコンストラクタの使用はおすすめしません。これは、他のデータ・プロバイダとの互換性のために用意されています。

パブリック・プロパティ

メンバ名	説明
DbType プロパティ	パラメータの System.Data.DbType を指定します。
Direction プロパティ	パラメータが入力専用、出力専用、双方向性、またはストアド・プロシージャ戻り値パラメータのいずれであるかを示す値です。
IsNullable プロパティ	パラメータが null 値を受け入れるかどうかを指定します。
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	パラメータの名前を指定します。
Precision プロパティ	ULParameter.Value を示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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。
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	パラメータの iAnywhere.Data.UltraLite.ULDbType を指定します。
Value プロパティ	パラメータの値を指定します。

パブリック・メソッド

メンバ名	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。

ULParameter コンストラクタ

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。

構文

Visual Basic

Overloads Public Sub **New()**

C#

public **ULParameter();**

例

次のコードでは、値が 3 である ULParameter パラメータが作成され、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 )

// C#
ULParameter p = new ULParameter();
p.Value = 3;
cmd.Parameters.Add( p );
```

参照

- 「ULParameter クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「Value プロパティ」 301 ページ
- 「ULParameter コンストラクタ」 289 ページ
- 「ULCommand クラス」 57 ページ

ULParameter コンストラクタ

ULParameter オブジェクトを、指定されたパラメータ名と値で初期化します。

構文

Visual Basic

```
Overloads Public Sub New( _
  ByVal parameterName As String, _
  ByVal value As Object _
)
```

C#

```
public ULParameter(
  string  parameterName,
  object value
);
```

パラメータ

- **parameterName**　パラメータの名前。名前のないパラメータの場合、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 はパラメータの名前を使用しません。
- **value**　パラメータの値である System.Object。

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。

例

次のコードでは、値が 0 である ULParameter パラメータが作成され、cmd という ULCommand が追加されます。

```
' Visual Basic
cmd.Parameters.Add( New ULParameter( "", CType( 0, Object ) ) )

// C#
cmd.Parameters.Add( new ULParameter( "", (object)0 ) );
```

参照

- 「ULParameter クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- 「ULParameter コンストラクタ」 288 ページ
- 「ULCommand クラス」 57 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- オブジェクト

ULParameter コンストラクタ

ULParameter オブジェクトを、指定されたパラメータ名とデータ型で初期化します。このコンストラクタの使用はおすすめしません。これは、他のデータ・プロバイダとの互換性のために用意されています。

構文

Visual Basic

```
Overloads Public Sub New( _
  ByVal parameterName As String, _
  ByVal dbType As ULDbType _
)
```

C#

```
public ULParameter(
  string  parameterName,
  ULDbType dbType
);
```

パラメータ

- **parameterName**　パラメータの名前。名前のないパラメータの場合、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 はパラメータの名前を使用しません。
- **dbType**　iAnywhere.Data.UltraLite.ULDbType 値の 1 つ。

備考

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- 「ULParameter コンストラクタ」 288 ページ
- 「ULParameter コンストラクタ」 289 ページ
- 「Value プロパティ」 301 ページ
- 「ULCommand クラス」 57 ページ
- 「ULDbType 列挙」 245 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULParameter コンストラクタ

ULParameter オブジェクトを、指定されたパラメータ名とデータ型で初期化します。このコンストラクタの使用はおすすめしません。これは、他のデータ・プロバイダとの互換性のために用意されています。

構文

Visual Basic

```
Overloads Public Sub New( _
  ByVal parameterName As String, _
  ByVal dbType As ULDbType, _
  ByVal size As Integer _
)
```

C#

```
public ULParameter(
  string  parameterName,
  ULDbType dbType,
  int size
);
```

パラメータ

- **parameterName**　パラメータの名前。名前のないパラメータの場合、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**　パラメータの長さ。

備考

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- 「ULParameter コンストラクタ」 288 ページ
- 「ULParameter コンストラクタ」 289 ページ
- 「Value プロパティ」 301 ページ
- 「ULDbType 列挙」 245 ページ

ULParameter コンストラクタ

ULParameter オブジェクトを、指定されたパラメータ名、データ型、長さで初期化します。このコンストラクタの使用はおすすめしません。これは、他のデータ・プロバイダとの互換性のために用意されています。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Overloads Public Sub New( _
  ByVal parameterName As String, _
  ByVal dbType As ULDbType, _
  ByVal size As Integer, _
  ByVal sourceColumn As String _
)
```

C#

```
public ULParameter(
  string  parameterName,
  ULDbType dbType,
  int size,
  string  sourceColumn
);
```

パラメータ

- **parameterName**　パラメータの名前。名前のないパラメータの場合、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**　パラメータの長さ。
- **sourceColumn**　マッピングするソース・カラムの名前。

備考

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- 「ULParameter コンストラクタ」 288 ページ
- 「ULParameter コンストラクタ」 289 ページ
- 「Value プロパティ」 301 ページ
- 「ULDbType 列挙」 245 ページ
- 「ULCommand クラス」 57 ページ

ULParameter コンストラクタ

指定されたパラメータ名、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。このコンストラクタの使用はおすすめしません。これは、他のデータ・プロバイダとの互換性のために用意されています。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Overloads 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#

```
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

- **parameterName** パラメータの名前。名前のないパラメータの場合、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** パラメータの値である System.Object。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

参照

- 「ULParameter クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- 「ULParameter コンストラクタ」 288 ページ
- 「ULParameter コンストラクタ」 289 ページ
- 「ULDbType 列挙」 245 ページ
- ParameterDirection
- DataRowVersion
- オブジェクト
- ParameterDirection.Input
- 「ULCommand クラス」 57 ページ

DbType プロパティ

パラメータの System.Data.DbType を指定します。

構文

Visual Basic

```
NotOverridable Public Property DbType As DbType _
  Implements IDataParameter.DbType
```

C#

```
public DbType DbType {get;set;}
```

プロパティ値

System.Data.DbType 値の 1 つ。

備考

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、サポートされている iAnywhere.Data.UltraLite.ULDbType に ULParameter.ULDbType を変更します。

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。

実装

IDataParameter.DbType

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULParameter クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- DbType
- DbType
- 「ULDbType プロパティ」 301 ページ
- 「ULDbType 列挙」 245 ページ

Direction プロパティ

パラメータが入力専用、出力専用、双方向性、またはストアド・プロシージャ戻り値パラメータのいずれであるかを示す値です。

構文

Visual Basic

```
NotOverridable Public Property Direction As ParameterDirection _
  Implements IDataParameter.Direction
```

C#

```
public ParameterDirection Direction {get;set;}
```

プロパティ値

System.Data.ParameterDirection 値の 1 つ。

備考

Ultra Light.NET では、パラメータは IN パラメータ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。

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。

実装

IDataParameter.Direction

参照

- 「ULParameter クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- ParameterDirection
- ParameterDirection.Input
- 「Value プロパティ」 301 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

IsNullable プロパティ

パラメータが null 値を受け入れるかどうかを指定します。

構文

Visual Basic

```
Public Property IsNullable As Boolean _
  Implements IDataParameter.IsNullable
```

C#

```
public bool IsNullable {get;set;}
```

プロパティ値

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。

備考

Ultra Light.NET では、パラメータは IN パラメータ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。

実装

IDataParameter.IsNullable

参照

- 「ULParameter クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- 「Value プロパティ」 301 ページ

Offset プロパティ

ULParameter.Value のオフセットを指定します。

構文

Visual Basic

```
Public Property Offset As Integer
```

C#

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

プロパティ値

値のオフセット。デフォルトは 0 です。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- 「Value プロパティ」 301 ページ

ParameterName プロパティ

パラメータの名前を指定します。

構文

Visual Basic

```
NotOverridable Public Property ParameterName As String _
  Implements IDataParameter.ParameterName
```

C#

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

プロパティ値

パラメータの名前を示す文字列。名前のないパラメータ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。

備考

Ultra Light.NET では、ULCommand はパラメータの名前を使用しません。すべてのパラメータは、位置パラメータとして扱われ、コマンドによって追加された順序で使用されます。

実装

IDataParameter.ParameterName

参照

- 「ULParameter クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- 「ULCommand クラス」 57 ページ

Precision プロパティ

ULParameter.Value を示すために使用される最大桁数を指定します。

構文

Visual Basic

```
NotOverridable Public Property Precision As Byte _
  Implements IDbDataParameter.Precision
```

C#

```
public byte Precision {get;set;}
```

プロパティ値

ULParameter.Value を示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・プロバイダが ULParameter.Value の精度を設定することを示します。

備考

Ultra Light.NET では、パラメータは IN パラメータ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。

例外

- ArgumentException - 値が 38 を超えています。

実装

IDbDataParameter.Precision

参照

- 「ULParameter クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- 「Value プロパティ」 301 ページ

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。

構文

Visual Basic

```
NotOverridable Public Property Scale As Byte _
  Implements IDbDataParameter.Scale
```

C#

```
public byte Scale {get;set;}
```

プロパティ値

ULParameter.Value が解決される小数点の桁の数。デフォルトは 0 です。

備考

Ultra Light.NET では、パラメータは IN パラメータ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 © 2007, iAnywhere Solutions, Inc.

実装

IDbDataParameter.Scale

参照

- 「ULParameter クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- 「Value プロパティ」 301 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。

構文

Visual Basic

```
NotOverridable Public Property Size As Integer _
  Implements IDbDataParameter.Size
```

C#

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

プロパティ値

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。

備考

Ultra Light.NET では、パラメータは IN パラメータ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。

実装

IDbDataParameter.Size

参照

- 「ULParameter クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- 「Value プロパティ」 301 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。

構文

Visual Basic

Copyright © 2007, iAnywhere Solutions, Inc.

```
NotOverridable Public Property SourceColumn As String _
  Implements IDataParameter.SourceColumn
```

C#

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

プロパティ値

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。

備考

Ultra Light.NET では、パラメータは IN パラメータ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。

実装

IDataParameter.SourceColumn

参照

- 「ULParameter クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- 「Value プロパティ」 301 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。

構文

Visual Basic

```
NotOverridable Public Property SourceVersion As DataRowVersion _
  Implements IDataParameter.SourceVersion
```

C#

```
public DataRowVersion SourceVersion {get;set;}
```

実装

IDataParameter.SourceVersion

参照

- 「ULParameter クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- 「Value プロパティ」 301 ページ
- DataRowVersion

Copyright © 2007, iAnywhere Solutions, Inc.

ULDbType プロパティ

パラメータの iAnywhere.Data.UltraLite.ULDbType を指定します。

構文

Visual Basic

Public Property **ULDbType** As ULDbType

C#

public ULDbType **ULDbType** {get;set;}

プロパティ値

iAnywhere.Data.UltraLite.ULDbType 値の 1 つ。

備考

ULDbType と ULParameter.DbType はリンクされます。このため、ULDbType を設定すると、サポートされている System.Data.DbType に ULParameter.DbType を変更します。

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- 「Value プロパティ」 301 ページ
- 「DbType プロパティ」 294 ページ
- DbType
- 「ULDbType 列挙」 245 ページ

Value プロパティ

パラメータの値を指定します。

構文

Visual Basic

NotOverridable Public Property **Value** As Object _
  Implements IDataParameter.Value

C#

public object **Value** {get;set;}

プロパティ値

パラメータの値を指定する System.Object。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

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。

実装

IDataParameter.Value

参照

- 「ULParameter クラス」 286 ページ
- 「ULParameter メンバ」 287 ページ
- 「ULException クラス」 249 ページ
- 「ULSQLCode 列挙」 366 ページ
- オブジェクト

ToString メソッド

このインスタンスの文字列表現を返します。

構文

Visual Basic

Overrides Public Function **ToString()** As String

C#

public override string **ToString();**

戻り値

パラメータの名前。

Copyright © 2007, iAnywhere Solutions, Inc.

ULParameterCollection クラス

ULCommand のすべてのパラメータを示します。

構文

Visual Basic

NotInheritable Public Class **ULParameterCollection**

C#

public sealed class **ULParameterCollection**

備考

コレクション内のすべてのパラメータは、位置パラメータとして扱われ、ULCommand.CommandText の疑問符のプレースホルダ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の数と同じでなければなりません。パラメータ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がありません。ULParameterCollection は、ULCommand.Parameters から取得します。

実装： System.Data.IDataParameterCollection

参照

- 「ULParameterCollection メンバ」 303 ページ
- 「ULCommand クラス」 57 ページ
- 「CommandText プロパティ」 63 ページ
- 「Parameters プロパティ」 67 ページ
- IDataParameterCollection

ULParameterCollection メンバ

パブリック・プロパティ

メンバ名	説明
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。
Item プロパティ	指定されたインデックスの ULParameter を返します。C# では、このプロパティは ULParameterCollection クラスのインデクサです。
Item プロパティ	指定された名前の ULParameter を返します。C# では、このプロパティは ULParameterCollection クラスのインデクサです。

Copyright © 2007, iAnywhere Solutions, Inc.

パブリック・メソッド

メンバ名	説明
Add メソッド	ULParameter をコレクションに追加します。
Add メソッド	ULParameter をコレクションに追加します。
Add メソッド	指定されたパラメータ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 メソッド	指定されたパラメータ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 メソッド	指定されたパラメータ名、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 メソッド	指定されたパラメータ名、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。
Clear メソッド	すべてのパラメータ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。
Contains 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。
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。
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。
IndexOf 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。
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を削除します。
RemoveAt メソッド	指定された名前のパラメータをコレクションから削除します。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
NotOverridable Public Readonly Property Count As Integer _
  Implements ICollection.Count
```

C#

```
public int Count {get;}
```

プロパティ値

コレクション内の ULParameter オブジェクトの数。

実装

ICollection.Count

Item プロパティ

指定されたインデックスの ULParameter を返します。C# では、このプロパティは ULParameterCollection クラスのインデクサです。

構文

Visual Basic

```
Public Property Item As ULParameter
```

C#

```
public ULParameter Item {get;set;}
```

パラメータ

- **index**　取り出すパラメータの 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のインデックス値はゼロです。

戻り値

指定されたインデックス位置の ULParameter。

例外

- System.IndexOutOfRangeException - インデックスが無効です。

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「ULParameter クラス」 286 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

Item プロパティ

指定された名前の ULParameter を返します。C# では、このプロパティは ULParameterCollection クラスのインデクサです。

構文

Visual Basic

Public Property **Item** As ULParameter

C#

public ULParameter **Item** {get;set;}

パラメータ

- **parameterName**　取り出すパラメータの名前。

戻り値

指定された名前の ULParameter。

例外

- System.ArgumentNullException - パラメータ名に null (Visual Basic の Nothing) を使用してパラメータを設定することはできません。
- System.IndexOutOfRangeException - 指定した名前のパラメータがありません。

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「Item プロパティ」 207 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetValue メソッド」 235 ページ
- 「GetFieldType メソッド」 222 ページ
- 「ULParameter クラス」 286 ページ

Add メソッド

ULParameter をコレクションに追加します。

構文

Visual Basic

```
Overloads NotOverridable Public Function Add( _
  ByVal value As Object _
) As Integer _
  Implements IList.Add
```


Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public int Add(
  object value
);
```

パラメータ

- **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。

備考

コレクション内のすべてのパラメータは、位置パラメータとして扱われ、ULCommand.CommandText 内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。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の数と同じでなければなりません。パラメータが不足している場合は、null が代用されます。

例外

- 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。
- System.ArgumentNullException - 値に null (Visual Basic の Nothing) を指定することはできません。
- System.InvalidCastException - 指定される値は ULParameter でなければなりません。

実装

IList.Add

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「Add メソッド」 307 ページ
- 「Add メソッド」 308 ページ
- 「CommandText プロパティ」 63 ページ
- 「ULParameter クラス」 286 ページ

Add メソッド

ULParameter をコレクションに追加します。

構文

Visual Basic

```
Overloads Public Function Add( _
```

```
  ByVal value As ULParameter _
) As ULParameter
```

C#

```
public ULParameter Add(
  ULParameter value
);
```

パラメータ

- **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。

備考

コレクション内のすべてのパラメータは、位置パラメータとして扱われ、ULCommand.CommandText 内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。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の数と同じでなければなりません。パラメータが不足している場合は、null が代用されます。

例外

- 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。
- System.ArgumentNullException - 値に null (Visual Basic の Nothing) を指定することはできません。

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「Add メソッド」 308 ページ
- 「ULParameter クラス」 286 ページ
- 「CommandText プロパティ」 63 ページ

Add メソッド

指定されたパラメータ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。

構文

Visual Basic

```
Overloads Public Function Add( _
  ByVal parameterName As String, _
  ByVal value As Object _
) As ULParameter
```


Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public ULParameter Add(
  string parameterName,
  object value
);
```

パラメータ

- **parameterName** パラメータの名前。名前のないパラメータの場合、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 はパラメータの名前を使用しません。
- **value** パラメータの値である 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は、位置パラメータとして扱われ、ULCommand.CommandText 内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。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の数と同じでなければなりません。パラメータ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。

例

次のコードでは、値が 0 である ULParameter パラメータが cmd という ULCommand に追加されます。

```
' Visual Basic
cmd.Parameters.Add( "", CType( 0, Object ) )

// C#
cmd.Parameters.Add( "", (object)0 );
```

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「Add メソッド」 307 ページ
- 「ULParameter クラス」 286 ページ
- 「CommandText プロパティ」 63 ページ
- 「ULCommand クラス」 57 ページ
- オブジェクト

Copyright © 2007, iAnywhere Solutions, Inc.

Add メソッド

指定されたパラメータ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。

構文

Visual Basic

```
Overloads Public Function Add( _
  ByVal parameterName As String, _
  ByVal ulDbType As ULDbType _
) As ULParameter
```

C#

```
public ULParameter Add(
  string  parameterName,
  ULDbType ulDbType
);
```

パラメータ

- **parameterName**　パラメータの名前。名前のないパラメータの場合、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は、位置パラメータとして扱われ、ULCommand.CommandText 内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。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の数と同じでなければなりません。パラメータが不足している場合は、null が代用されます。

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「Add メソッド」 307 ページ
- 「Add メソッド」 308 ページ
- 「ULParameter クラス」 286 ページ
- 「CommandText プロパティ」 63 ページ
- 「ULCommand クラス」 57 ページ
- 「ULDbType 列挙」 245 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

Add メソッド

指定されたパラメータ名、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。

構文

Visual Basic

```
Overloads Public Function Add( _
  ByVal parameterName As String, _
  ByVal ulDbType As ULDbType, _
  ByVal size As Integer _
) As ULParameter
```

C#

```
public ULParameter Add(
  string  parameterName,
  ULDbType ulDbType,
  int size
);
```

パラメータ

- **parameterName**　パラメータの名前。名前のないパラメータの場合、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は、位置パラメータとして扱われ、ULCommand.CommandText 内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。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の数と同じでなければなりません。パラメータが不足している場合は、null が代用されます。

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「Add メソッド」 307 ページ
- 「Add メソッド」 308 ページ
- 「ULParameter クラス」 286 ページ
- 「CommandText プロパティ」 63 ページ
- 「ULCommand クラス」 57 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULDbType 列挙」 245 ページ

Add メソッド

指定されたパラメータ名、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。

構文

Visual Basic

```
Overloads Public Function Add( _
  ByVal parameterName As String, _
  ByVal ulDbType As ULDbType, _
  ByVal size As Integer, _
  ByVal sourceColumn As String _
) As ULParameter
```

C#

```
public ULParameter Add(
  string  parameterName,
  ULDbType ulDbType,
  int size,
  string  sourceColumn
);
```

パラメータ

- **parameterName** パラメータの名前。名前のないパラメータの場合、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は、位置パラメータとして扱われ、ULCommand.CommandText 内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。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の数と同じでなければなりません。パラメータが不足している場合は、null が代用されます。

参照

- 「ULParameterCollection クラス」 303 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULParameterCollection メンバ」 303 ページ
- 「Add メソッド」 307 ページ
- 「Add メソッド」 308 ページ
- 「ULParameter クラス」 286 ページ
- 「CommandText プロパティ」 63 ページ
- 「ULCommand クラス」 57 ページ
- 「ULDbType 列挙」 245 ページ

Clear メソッド

すべてのパラメータをコレクションから削除します。

構文

Visual Basic

```
NotOverridable Public Sub Clear() _
  Implements IList.Clear
```

C#

```
public void Clear();
```

実装

IList.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。

構文

Visual Basic

```
Overloads NotOverridable Public Function Contains( _
  ByVal value As Object _
) As Boolean _
  Implements IList.Contains
```

C#

```
public bool Contains(
  object value
);
```

パラメータ

- **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。

Copyright © 2007, iAnywhere Solutions, Inc.

実装

IList.Contains

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「Contains メソッド」 314 ページ
- 「ULParameter クラス」 286 ページ

Contains 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。

構文

Visual Basic

```
Overloads NotOverridable Public Function Contains( _
  ByVal value As String _
) As Boolean _
  Implements IDataParameterCollection.Contains
```

C#

```
public bool Contains(
  string  value
);
```

パラメータ

- **value**　検索対象のパラメータの名前。

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。

実装

IDataParameterCollection.Contains

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「Contains メソッド」 313 ページ
- 「ULParameter クラス」 286 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
NotOverridable Public Sub CopyTo( _
  ByVal array As System.Array, _
  ByVal index As Integer _
) _
  Implements ICollection.CopyTo
```

C#

```
public void CopyTo(
  System.Array array,
  int index
);
```

パラメータ

- **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。

実装

ICollection.CopyTo

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「ULParameter クラス」 286 ページ

GetEnumerator メソッド

コレクションの列挙子を返します。

構文

Visual Basic

```
NotOverridable Public Function GetEnumerator() As System.Collections.IEnumerator _
  Implements IEnumerable.GetEnumerator
```

C#

```
public System.Collections.IEnumerator GetEnumerator();
```

戻り値

コレクション内のパラメータを列挙する ArrayList 列挙子。

実装

IEnumerable.GetEnumerator

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Overloads NotOverridable Public Function IndexOf( _
  ByVal value As Object _
) As Integer _
  Implements IList.IndexOf
```

C#

```
public int IndexOf(
  object value
);
```

パラメータ

- **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。パラメータが見つからない場合は -1。

例外

- System.InvalidCastException - 指定される値は ULParameter でなければなりません。

実装

IList.IndexOf

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「IndexOf メソッド」 316 ページ
- 「ULParameter クラス」 286 ページ

IndexOf 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。

構文

Visual Basic

```
Overloads NotOverridable Public Function IndexOf( _
  ByVal parameterName As String _
) As Integer _
  Implements IDataParameterCollection.IndexOf
```


Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public int IndexOf(
   string  parameterName
);
```

パラメータ

- **parameterName**　検索するパラメータの名前。

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。パラメータが見つからない場合は -1。

実装

IDataParameterCollection.IndexOf

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「IndexOf メソッド」 316 ページ
- 「ULParameter クラス」 286 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。

構文

Visual Basic

```
NotOverridable Public Sub Insert( _
   ByVal index As Integer, _
   ByVal value As Object _
) _
   Implements IList.Insert
```

C#

```
public void Insert(
   int index,
   object value
);
```

パラメータ

- **index**　コレクション内にパラメータ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。

Copyright © 2007, iAnywhere Solutions, Inc.

例外

- System.ArgumentNullException - null 参照 (Visual Basic の Nothing) を使用してパラメータを設定することはできません。
- System.IndexOutOfRangeException - インデックスが無効です。
- System.InvalidCastException - 指定される値は ULParameter でなければなりません。

実装

IList.Insert

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「ULParameter クラス」 286 ページ

Remove メソッド

ULParameter をコレクションから削除します。

構文

Visual Basic

```
NotOverridable Public Sub Remove( _
  ByVal value As Object _
) _
  Implements IList.Remove
```

C#

```
public void Remove(
  object value
);
```

パラメータ

- **value**　削除する ULParameter オブジェクト。

例外

- ArgumentException - コレクション内に、指定されたパラメータがありません。
- System.ArgumentNullException - null 参照 (Visual Basic の Nothing) を使用してパラメータを設定することはできません。
- System.InvalidCastException - 指定される値は ULParameter でなければなりません。

実装

IList.Remove

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「ULParameter クラス」 286 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を削除します。

構文

Visual Basic

```
Overloads NotOverridable Public Sub RemoveAt( _
  ByVal index As Integer _
) _
  Implements IList.RemoveAt
```

C#

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

パラメータ

- **index**　削除するパラメータの 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のインデックス値はゼロです。

例外

- System.IndexOutOfRangeException - インデックスが無効です。

実装

IList.RemoveAt

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「RemoveAt メソッド」 319 ページ
- 「Count プロパティ」 304 ページ

RemoveAt メソッド

指定された名前のパラメータをコレクションから削除します。

構文

Visual Basic

Copyright © 2007, iAnywhere Solutions, Inc.

```
Overloads NotOverridable Public Sub RemoveAt( _
  ByVal parameterName As String _
) _
  Implements IDataParameterCollection.RemoveAt
```

C#

```
public void RemoveAt(
  string  parameterName
);
```

パラメータ

- **parameterName**　取り出すパラメータの名前。

例外

- System.IndexOutOfRangeException - 指定された名前のパラメータがありません。

実装

IDataParameterCollection.RemoveAt

参照

- 「ULParameterCollection クラス」 303 ページ
- 「ULParameterCollection メンバ」 303 ページ
- 「RemoveAt メソッド」 319 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULPublicationSchema クラス

UL 拡張： Ultra Light パブリケーションのインデックスのスキーマを示します。

構文

Visual Basic

NotInheritable Public Class **ULPublicationSchema**

C#

public sealed class **ULPublicationSchema**

備考

このクラスにはコンストラクタがありません。パブリケーションのスキーマは、ULDatabaseSchema クラスの ULDatabaseSchema.GetPublicationSchema メソッドを使用して作成されます。

パブリケーション・マスクが必要な Ultra Light のメソッドには、実際にはチェック対象のパブリケーションのセットが必要です。セットは、個々のパブリケーションのパブリケーション・マスクの論理和で形成されます。次に例を示します。

```
' Visual Basic
Dim mask As Integer = pub1.Mask Or pub2.Mask

// C#
int mask = pub1.Mask | pub2.Mask;
```

特別な 2 つのマスク値も、このクラスによって提供されます。
ULPublicationSchema.SYNC_ALL_DB は、データベース全体に対応します。
ULPublicationSchema.SYNC_ALL_PUBS は、すべてのパブリケーションに対応します。

注意：パブリケーション・マスクは、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マスク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

- 「ULPublicationSchema メンバ」 322 ページ
- 「Mask プロパティ」 323 ページ
- 「GetLastDownloadTime メソッド」 113 ページ
- 「CountUploadRows メソッド」 107 ページ
- 「Schema プロパティ」 100 ページ
- 「SYNC_ALL_DB フィールド」 322 ページ
- 「SYNC_ALL_PUBS フィールド」 322 ページ
- 「GetPublicationSchema メソッド」 197 ページ
- 「ULDatabaseSchema クラス」 190 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULPublicationSchema メンバ

パブリックの静的フィールド (共有)

メンバ名	説明
SYNC_ALL_DB フィールド	データベース全体に対応するパブリケーション・マスクです。
SYNC_ALL_PUBS フィールド	すべてのパブリケーションに対応するパブリケーション・マスクです。

パブリック・プロパティ

メンバ名	説明
IsOpen プロパティ	パブリケーション・スキーマが開いているか、閉じているかを調べます。
Mask プロパティ	パブリケーションのパブリケーション・マスクを返します。
Name プロパティ	このパブリケーションの名前を返します。

SYNC_ALL_DB フィールド

データベース全体に対応するパブリケーション・マスクです。

構文

Visual Basic

Public Shared **SYNC_ALL_DB** As Integer

C#

public const int **SYNC_ALL_DB**;

SYNC_ALL_PUBS フィールド

すべてのパブリケーションに対応するパブリケーション・マスクです。

構文

Visual Basic

Public Shared **SYNC_ALL_PUBS** As Integer

C#

public const int **SYNC_ALL_PUBS**;

Copyright © 2007, iAnywhere Solutions, Inc.

IsOpen プロパティ

パブリケーション・スキーマが開いているか、閉じているかを調べます。

構文

Visual Basic

Public Readonly Property **IsOpen** As Boolean

C#

public bool **IsOpen** {get;}

プロパティ値

パブリケーション・スキーマが開いている場合は true、閉じている場合は false。

Mask プロパティ

パブリケーションのパブリケーション・マスクを返します。

構文

Visual Basic

Public Readonly Property **Mask** As Integer

C#

public int **Mask** {get;}

プロパティ値

パブリケーションのパブリケーション・マスク。

備考

パブリケーションの ID、マスク、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マスク、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。

例外

- ULException クラス - SQL エラーによる例外が発生した場合。スキーマのアップグレードでこのパブリケーションが削除されるか、名前が変更された場合は、PUBLICATION_NOT_FOUND が発行されます。

Name プロパティ

このパブリケーションの名前を返します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **Name** As String

C#

public string **Name** {get;}

プロパティ値

パブリケーションの名前を指定する文字列。

例外

- ULException クラス - SQL エラーが発生しました。スキーマのアップグレードでパブリケーションが削除されるか、名前が変更された場合は、PUBLICATION_NOT_FOUND が発行されます。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Public Class ULResultSet
  Inherits ULDataReader
```

C#

```
public class ULResultSet :
  ULDataReader
```

備考

このクラスにはコンストラクタがありません。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()

// 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。

継承： ULDataReader

実装： System.Data.IDataReader、System.Data.IDataRecord、System.IDisposable

参照

- 「ULResultSet メンバ」 326 ページ
- 「ExecuteResultSet メソッド」 75 ページ
- 「ULCommand クラス」 57 ページ
- 「ULResultSet クラス」 325 ページ
- 「ExecuteTable メソッド」 78 ページ
- 「ULDataAdapter クラス」 156 ページ
- 「ULDataReader クラス」 201 ページ
- IDataReader
- IDataRecord
- IDisposable

Copyright © 2007, iAnywhere Solutions, Inc.

ULResultSet メンバ

パブリック・プロパティ

メンバ名	説明
Depth プロパティ (ULDataReader から継承)	現在のローのネストの深さを返します。最も外側のテーブルの深さは 0 です。
FieldCount プロパティ (ULDataReader から継承)	このカーソル内のカラム数を返します。
IsBOF プロパティ (ULDataReader から継承)	**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 (ULDataReader から継承)	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 (ULDataReader から継承)	**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。
Item プロパティ (ULDataReader から継承)	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。C# では、このプロパティは ULDataReader クラスのインデクサです。
Item プロパティ (ULDataReader から継承)	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。C# では、このプロパティは ULDataReader クラスのインデクサです。
RecordsAffected プロパティ (ULDataReader から継承)	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。SELECT 文または CommandType.TableDirect テーブルの場合、この値は -1 です。
RowCount プロパティ (ULDataReader から継承)	**UL 拡張：** カーソル内のロー数を返します。
Schema プロパティ (ULDataReader から継承)	**UL 拡張：** このカーソルのスキーマを保持します。

パブリック・メソッド

メンバ名	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 (ULDataReader から継承)	カーソルを閉じます。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
Delete メソッド	現在の行を削除します。
Dispose メソッド (ULDataReader から継承)	ULDataReader 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。
GetBoolean メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	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 メソッド (ULDataReader から継承)	**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。ULDbType.Binary 型、ULDbType.LongBinary 型、ULDbType.UniqueIdentifier 型のカラムの場合にのみ有効です。
GetChar メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 メソッド (ULDataReader から継承)	このメソッドは Ultra Light.NET ではサポートされていません。
GetDataTypeName メソッド (ULDataReader から継承)	指定されたカラムのプロバイダのデータ型の名前を返します。
GetDateTime メソッド (ULDataReader から継承)	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。
GetDecimal メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	指定されたカラムの値を System.Double として返します。
GetFieldType メソッド (ULDataReader から継承)	指定されたカラムに最適な System.Type を返します。
GetFloat メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	指定されたカラムの値を UUID (System.Guid) として返します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
GetInt16 メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	指定されたカラムの値を Int32 として返します。
GetInt64 メソッド (ULDataReader から継承)	指定されたカラムの値を Int64 として返します。
GetName メソッド (ULDataReader から継承)	指定されたカラムの名前を返します。
GetOrdinal メソッド (ULDataReader から継承)	指定されたカラムのカラム ID を返します。
GetSchemaTable メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	指定されたカラムの値を UInt32 として返します。
GetUInt64 メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	現在のローのすべてのカラム値を返します。
IsDBNull メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**UL 拡張：** カーソルの最初のローに、カーソルを配置します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
MoveLast メソッド (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	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 (ULDataReader から継承)	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。
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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。

保護メソッド

メンバ名	説明
Finalize メソッド	ULResultSet がガーベジ・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。

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。

構文

Visual Basic

```
Public Sub AppendBytes( _
   ByVal columnID As Integer, _
   ByVal val As Byte(), _
   ByVal srcOffset As Integer, _
   ByVal count As Integer _
)
```

C#

```
public void AppendBytes(
   int columnID,
   byte[] val,
   int srcOffset,
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
    int count
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **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。

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。
- *srcOffset*+*count* が *val*.Length. 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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- Byte
- 「ULDbType 列挙」 245 ページ
- 「InsertBegin メソッド」 447 ページ
- 「Insert メソッド」 446 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULException クラス」 249 ページ
- 「ULSQLCode 列挙」 366 ページ
- 「ULException クラス」 249 ページ
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
Public Sub AppendChars( _
  ByVal columnID As Integer, _
  ByVal val As Char(), _
  ByVal srcOffset As Integer, _
  ByVal count As Integer _
)
```

C#

```
public void AppendChars(
  int columnID,
  char[] val,
  int srcOffset,
  int count
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **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。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

- *val* が null である。
- *srcOffset* が負である。
- *count* が負である。
- *srcOffset*+*count* が *value*.Length. 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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- System.Char
- 「ULDbType 列挙」 245 ページ
- 「InsertBegin メソッド」 447 ページ
- 「Insert メソッド」 446 ページ
- 「ULException クラス」 249 ページ
- 「ULSQLCode 列挙」 366 ページ
- 「FieldCount プロパティ」 205 ページ

Delete メソッド

現在の行を削除します。

構文

Visual Basic

Public Sub **Delete()**

C#

public void **Delete();**

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「StartSynchronizationDelete メソッド」 118 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「StopSynchronizationDelete メソッド」 119 ページ

Finalize メソッド

ULResultSet がガーベジ・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。

構文

Visual Basic

```
Overrides Protected Sub Finalize()
```

C#

```
protected override void Finalize();
```

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。

構文

Visual Basic

```
Public Sub SetBoolean( _
  ByVal columnID As Integer, _
  ByVal val As Boolean _
)
```

C#

```
public void SetBoolean(
  int columnID,
  bool val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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。

例外

- ULException クラス - SQL エラーが発生しました。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- Boolean
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
Public Sub SetByte( _
  ByVal columnID As Integer, _
  ByVal val As Byte _
)
```

C#

```
public void SetByte(
  int columnID,
  byte val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「GetFieldType メソッド」 222 ページ
- Byte
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
Public Sub SetBytes( _
   ByVal columnID As Integer, _
   ByVal val As Byte() _
)
```

C#

```
public void SetBytes(
   int columnID,
   byte[] val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- Byte
- 「Insert メソッド」 446 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 ページ
- 「ULDbType 列挙」 245 ページ
- 「ULDbType 列挙」 245 ページ
- 「ULDbType 列挙」 245 ページ

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。

構文

Visual Basic

```
Public Sub SetDateTime( _
  ByVal columnID As Integer, _
  ByVal val As Date _
)
```

C#

```
public void SetDateTime(
  int columnID,
  DateTime val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- DateTime
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「FieldCount プロパティ」 205 ページ

SetDBNull メソッド

カラムを NULL に設定します。

構文

Visual Basic

```
Public Sub SetDBNull( _
  ByVal columnID As Integer _
)
```

C#

```
public void SetDBNull(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「IsColumnNullable メソッド」 466 ページ
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 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。

構文

Visual Basic

Copyright © 2007, iAnywhere Solutions, Inc.

```
Public Sub SetDecimal( _
  ByVal columnID As Integer, _
  ByVal val As Decimal _
)
```

C#

```
public void SetDecimal(
  int columnID,
  decimal val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- Decimal
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
Public Sub SetDouble( _
  ByVal columnID As Integer, _
  ByVal val As Double _
)
```

Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public void SetDouble(
   int columnID,
   double val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- Double
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
Public Sub SetFloat( _
   ByVal columnID As Integer, _
   ByVal val As Single _
)
```

C#

```
public void SetFloat(
   int columnID,
   float  val
);
```


Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- Single
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
Public Sub SetGuid( _
  ByVal columnID As Integer, _
  ByVal val As Guid _
)
```

C#

```
public void SetGuid(
  int columnID,
  Guid val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetNewUUID メソッド」 114 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- 「GetColumnSize メソッド」 153 ページ
- Guid
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ
- 「ULDbType 列挙」 245 ページ
- 「ULDbType 列挙」 245 ページ
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
Public Sub SetInt16( _
  ByVal columnID As Integer, _
  ByVal val As Short _
)
```

C#

```
public void SetInt16(
  int columnID,
  short val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- Int16
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
Public Sub SetInt32( _
  ByVal columnID As Integer, _
  ByVal val As Integer _
)
```

C#

```
public void SetInt32(
  int columnID,
  int val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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 © 2007, iAnywhere Solutions, Inc.

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- Int32
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
Public Sub SetInt64( _
  ByVal columnID As Integer, _
  ByVal val As Long _
)
```

C#

```
public void SetInt64(
  int columnID,
  long val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- Int64
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
Public Sub SetString( _
  ByVal columnID As Integer, _
  ByVal val As String _
)
```

C#

```
public void SetString(
  int columnID,
  string  val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- 「Insert メソッド」 446 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 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。

構文

Visual Basic

```
Public Sub SetTimeSpan( _
  ByVal columnID As Integer, _
  ByVal val As TimeSpan _
)
```

C#

```
public void SetTimeSpan(
  int columnID,
  TimeSpan val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

備考

設定された値は、ミリ秒の精度であり、0 ー 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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- TimeSpan
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Public Sub SetToDefault( _
  ByVal columnID As Integer _
)
```

C#

```
public void SetToDefault(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- 「GetColumnDefaultValue メソッド」 458 ページ
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 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。

構文

Visual Basic

```
Public Sub SetUInt16( _
  ByVal columnID As Integer, _
```

```
  ByVal val As UInt16 _
)
```

C#

```
public void SetUInt16(
  int columnID,
  ushort val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「GetFieldType メソッド」 222 ページ
- UInt16
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
Public Sub SetUInt32( _
  ByVal columnID As Integer, _
  ByVal val As UInt32 _
)
```

C#

```
public void SetUInt32(
  int columnID,
```

 Copyright © 2007, iAnywhere Solutions, Inc.

```
  uint val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- UInt32
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

```
Public Sub SetUInt64( _
  ByVal columnID As Integer, _
  ByVal val As UInt64 _
)
```

C#

```
public void SetUInt64(
  int columnID,
  ulong val
);
```

Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULResultSet クラス」 325 ページ
- 「ULResultSet メンバ」 326 ページ
- 「GetOrdinal メソッド」 228 ページ
- 「Schema プロパティ」 210 ページ
- 「GetFieldType メソッド」 222 ページ
- UInt64
- 「Insert メソッド」 446 ページ
- 「Update メソッド」 350 ページ
- 「FieldCount プロパティ」 205 ページ

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。

構文

Visual Basic

Public Sub **Update()**

C#

public void **Update();**

例外

- ULException クラス - SQL エラーが発生しました。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
NotInheritable Public Class ULResultSetSchema
  Inherits ULCursorSchema
```

C#

```
public sealed class ULResultSetSchema :
  ULCursorSchema
```

備考

このクラスにはコンストラクタがありません。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・リーダが開かれている間だけです。

継承： ULCursorSchema

参照

- 「ULResultSetSchema メンバ」 351 ページ
- 「ULCommand クラス」 57 ページ
- 「ULDataReader クラス」 201 ページ
- 「ULResultSetSchema クラス」 351 ページ
- 「Schema プロパティ」 210 ページ
- 「ULCursorSchema クラス」 147 ページ

ULResultSetSchema メンバ

パブリック・プロパティ

メンバ名	説明
ColumnCount プロパティ (ULCursorSchema から継承)	このカーソル内のカラム数を返します。
IsOpen プロパティ (ULCursorSchema から継承)	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。

Copyright © 2007, iAnywhere Solutions, Inc.

パブリック・メソッド

メンバ名	説明
GetColumnID メソッド (ULCursorSchema から継承)	指定されたカラムのカラム ID を返します。
GetColumnName メソッド (ULCursorSchema から継承)	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 (ULCursorSchema から継承)	カラムが数値カラム (SQL NUMERIC 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 (ULCursorSchema から継承)	カラムが数値カラム (SQL NUMERIC 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 (ULCursorSchema から継承)	カラムがサイズ指定されたカラム (SQL BINARY 型または CHAR 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 (ULCursorSchema から継承)	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 (ULCursorSchema から継承)	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 (ULCursorSchema から継承)	ULDataReader のカラムのスキーマが記述された System.Data.DataTable を返します。

Name プロパティ

カーソルの名前を返します。

構文

Visual Basic

Public Readonly Property **Name** As String

C#

public string **Name** {get;}

プロパティ値

ULResultSetSchema を生成した SQL 文。

Copyright © 2007, iAnywhere Solutions, Inc.

ULRowUpdatedEventArgs クラス

ULDataAdapter.RowUpdated イベントのデータを提供します。

構文

Visual Basic

```
NotInheritable Public Class ULRowUpdatedEventArgs
  Inherits RowUpdatedEventArgs
```

C#

```
public sealed class ULRowUpdatedEventArgs :
  RowUpdatedEventArgs
```

備考

継承： System.Data.Common.RowUpdatedEventArgs

参照

- 「ULRowUpdatedEventArgs メンバ」 353 ページ
- 「RowUpdated イベント」 180 ページ
- RowUpdatedEventArgs

ULRowUpdatedEventArgs メンバ

パブリック・コンストラクタ

メンバ名	説明
ULRowUpdatedEventArgs コンストラクタ	ULRowUpdatedEventArgs クラスの新しいインスタンスを初期化します。

パブリック・プロパティ

メンバ名	説明
Command プロパティ	DbDataAdapter.Update(System.Data.DataRow [],System.Data.Common.DataTableMapping) が呼び出されたときに実行された ULCommand を返します。
Errors (RowUpdatedEventArgs から継承)	RowUpdatedEventArgs.Command の実行時に .NET Framework データ・プロバイダによって生成されるエラーを取得します。
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。SELECT 文の場合、この値は -1 です。
Row (RowUpdatedEventArgs から継承)	DbDataAdapter.Update によって送信された DataRow を取得します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
StatementType (RowUpdatedEventArgs から継承)	実行された SQL 文のタイプを取得します。
Status (RowUpdatedEventArgs から継承)	RowUpdatedEventArgs.Command プロパティの UpdateStatus を取得します。
TableMapping (RowUpdatedEventArgs から継承)	DbDataAdapter.Update によって送信された DataTableMapping を取得します。

ULRowUpdatedEventArgs コンストラクタ

ULRowUpdatedEventArgs クラスの新しいインスタンスを初期化します。

構文

Visual Basic

```
Public Sub New( _
  ByVal row As DataRow, _
  ByVal command As IDbCommand, _
  ByVal statementType As StatementType, _
  ByVal tableMapping As DataTableMapping _
)
```

C#

```
public ULRowUpdatedEventArgs(
  DataRow row,
  IDbCommand command,
  StatementType statementType,
  DataTableMapping tableMapping
);
```

パラメータ

- **row** DbDataAdapter.Update(System.Data.DataRow[],System.Data.Common.DataTableMapping) によって送信された System.Data.DataRow。
- **command** DbDataAdapter.Update(System.Data.DataRow [],System.Data.Common.DataTableMapping) が呼び出されたときに実行された 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(System.Data.DataRow [],System.Data.Common.DataTableMapping) によって送信された System.Data.Common.DataTableMapping。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULRowUpdatedEventArgs クラス」 353 ページ
- 「ULRowUpdatedEventArgs メンバ」 353 ページ
- DataRow
- IDbCommand
- StatementType
- DataTableMapping
- DbDataAdapter.Update

Command プロパティ

DbDataAdapter.Update(System.Data.DataRow[],System.Data.Common.DataTableMapping) が呼び出されたときに実行された ULCommand を返します。

構文

Visual Basic

Public Readonly Property **Command** As ULCommand

C#

public ULCommand **Command** {get;}

プロパティ値

更新で実行された ULCommand オブジェクト。

備考

これは、System.Data.Common.RowUpdatedEventArgs.Command が厳密に型指定されたものです。

参照

- 「ULRowUpdatedEventArgs クラス」 353 ページ
- 「ULRowUpdatedEventArgs メンバ」 353 ページ
- 「ULCommand クラス」 57 ページ
- DbDataAdapter.Update
- 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。SELECT 文の場合、この値は -1 です。

構文

Visual Basic

Public Readonly Property **RecordsAffected** As Integer

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public int **RecordsAffected** {get;}

プロパティ値

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。

Copyright © 2007, iAnywhere Solutions, Inc.

ULRowUpdatedEventHandler 委任

ULDataAdapter.RowUpdated イベントを処理するメソッドを示します。

構文

Visual Basic

```
Public Delegate Sub ULRowUpdatedEventHandler( _
  ByVal sender As Object, _
  ByVal e As ULRowUpdatedEventArgs _
)
```

C#

```
public delegate void ULRowUpdatedEventHandler(
  object sender,
  ULRowUpdatedEventArgs e
);
```

パラメータ

- **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

- 「RowUpdated イベント」 180 ページ
- 「ULRowUpdatedEventArgs クラス」 353 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULRowUpdatingEventArgs クラス

ULDataAdapter.RowUpdating イベントのデータを提供します。

構文

Visual Basic

NotInheritable Public Class **ULRowUpdatingEventArgs**
  Inherits RowUpdatingEventArgs

C#

public sealed class **ULRowUpdatingEventArgs** :
  RowUpdatingEventArgs

備考

継承： System.Data.Common.RowUpdatingEventArgs

参照

- 「ULRowUpdatingEventArgs メンバ」 358 ページ
- 「RowUpdating イベント」 181 ページ
- RowUpdatingEventArgs

ULRowUpdatingEventArgs メンバ

パブリック・コンストラクタ

メンバ名	説明
ULRowUpdatingEventArgs コンストラクタ	ULRowUpdatingEventArgs クラスの新しいインスタンスを初期化します。

パブリック・プロパティ

メンバ名	説明
Command プロパティ	DbDataAdapter.Update(System.Data.DataRow [],System.Data.Common.DataTableMapping) の実行時に実行する ULCommand を指定します。
Errors (RowUpdatingEventArgs から継承)	RowUpdatedEventArgs.Command の実行時に .NET Framework データ・プロバイダによって生成されるエラーを取得します。
Row (RowUpdatingEventArgs から継承)	DbDataAdapter.Update によって送信された DataRow を取得します。
StatementType (RowUpdatingEventArgs から継承)	実行する SQL 文のタイプを取得します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
Status (RowUpdatingEventArgs から継承)	RowUpdatedEventArgs.Command プロパティの UpdateStatus を取得します。
TableMapping (RowUpdatingEventArgs から継承)	DbDataAdapter.Update によって送信する DataTableMapping を取得します。

ULRowUpdatingEventArgs コンストラクタ

ULRowUpdatingEventArgs クラスの新しいインスタンスを初期化します。

構文

Visual Basic

```
Public Sub New( _
  ByVal row As DataRow, _
  ByVal command As IDbCommand, _
  ByVal statementType As StatementType, _
  ByVal tableMapping As DataTableMapping _
)
```

C#

```
public ULRowUpdatingEventArgs(
  DataRow row,
  IDbCommand command,
  StatementType statementType,
  DataTableMapping tableMapping
);
```

パラメータ

- **row** 更新する System.Data.DataRow。
- **command** 更新時に実行する 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(System.Data.DataRow [],System.Data.Common.DataTableMapping) によって送信された System.Data.Common.DataTableMapping。

参照

- 「ULRowUpdatingEventArgs クラス」 358 ページ
- 「ULRowUpdatingEventArgs メンバ」 358 ページ
- DataRow
- IDbCommand
- StatementType

Copyright © 2007, iAnywhere Solutions, Inc.

- DataTableMapping
- DbDataAdapter.Update

Command プロパティ

DbDataAdapter.Update(System.Data.DataRow[],System.Data.Common.DataTableMapping) の実行時に実行する ULCommand を指定します。

構文

Visual Basic

Public Property **Command** As ULCommand

C#

public ULCommand **Command** {get;set;}

プロパティ値

更新時に実行する ULCommand オブジェクト。

備考

これは、System.Data.Common.RowUpdatingEventArgs.Command が厳密に型指定されたものです。

参照

- 「ULRowUpdatingEventArgs クラス」 358 ページ
- 「ULRowUpdatingEventArgs メンバ」 358 ページ
- 「ULCommand クラス」 57 ページ
- DbDataAdapter.Update
- RowUpdatingEventArgs.Command

 Copyright © 2007, iAnywhere Solutions, Inc.

ULRowUpdatingEventHandler 委任

ULDataAdapter.RowUpdating イベントを処理するメソッドを示します。

構文

Visual Basic

```
Public Delegate Sub ULRowUpdatingEventHandler( _
  ByVal sender As Object, _
  ByVal e As ULRowUpdatingEventArgs _
)
```

C#

```
public delegate void ULRowUpdatingEventHandler(
  object sender,
  ULRowUpdatingEventArgs e
);
```

パラメータ

- **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

- 「RowUpdating イベント」 181 ページ
- 「ULRowUpdatingEventArgs クラス」 358 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULRuntimeType 列挙

UL 拡張： Ultra Light.NET ランタイムのタイプを列挙します。

構文

Visual Basic

```
Public Enum ULRuntimeType
  Inherits Short
```

C#

```
public enum ULRuntimeType :
  short
```

メンバ名

メンバ名	説明
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。
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。

参照

- 「RuntimeType プロパティ」 183 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULServerSyncListener インタフェース

UL 拡張： サーバの同期メッセージを受信するリスナ・インタフェースです。

構文

Visual Basic

Public Interface **ULServerSyncListener**

C#

public interface **ULServerSyncListener**

ULServerSyncListener メンバ

パブリック・メソッド

メンバ名	説明
ServerSyncInvoked メソッド	サーバ起動同期用の 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。

構文

Visual Basic

```
Public Sub ServerSyncInvoked( _
  ByVal messageName As String _
)
```

C#

```
public void ServerSyncInvoked(
  string  messageName
);
```

パラメータ

- **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・インタフェースにイベントを通知する必要があります。マルチス

Copyright © 2007, iAnywhere Solutions, Inc.

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

次のコード・フラグメントは、サーバ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。

```
' Visual Basic
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
    ...
  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

// C#
using iAnywhere.Data.UltraLite;
public class Form1 : System.Windows.Forms.Form, ULServerSyncListener
{
  private System.Windows.Forms.MainMenu mainMenu1;
  private ULConnection conn;

  public Form1()
  {
    //
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
    // Required for Windows Form Designer support
    //
    InitializeComponent();

    //
    // TODO: Add any constructor code after
    // InitializeComponent call
    //
    ULConnection.DatabaseManager.SetServerSyncListener(
        "myCompnay.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();
  }
}
```

Copyright © 2007, iAnywhere Solutions, Inc.

ULSQLCode 列挙

UL 拡張： Ultra Light.NET によってレポートされる可能性のある SQL コードを列挙します。

構文

Visual Basic

Public Enum **ULSQLCode**

C#

public enum **ULSQLCode**

メンバ名

メンバ名	説明
SQLE_AGGREGATES_NOT_ALLOWED	「集合関数の不正な使用です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ALIAS_NOT_UNIQUE	「エイリアス '%1' がユニークではありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ALIAS_NOT_YET_DEFINED	「エイリアス '%1' の定義は、最初の参照前に記述する必要があります。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_AMBIGUOUS_INDEX_NAME	「インデックス名 '%1' があいまいです。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ARGUMENT_CANNOT_BE_NULL	「プロシージャ '%2' の引数 %1 に NULL は指定できません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_BAD_ENCRYPTION_KEY	「暗号化キーが不正であるか、見つかりません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_BAD_PARAM_INDEX	「入力パラメータ・インデックスが範囲外です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_CANNOT_ACCESS_FILESYSTEM	「デバイス上のファイルシステムにアクセスできません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_CANNOT_CHANGE_ML_REMOTE_ID	「前回のアップロードのステータスが不明の場合、Mobile Link リモート ID は変更できません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_CANNOT_EXECUTE_STMT	「文を実行することができませんでした。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_CANNOT_MODIFY	「テーブル '%2' のカラム '%1' を変更できません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_CANNOT_REGISTER_LISTENER	「指定されたリスナを登録できませんでした。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SQLE_CLIENT_OUT_OF_MEMORY	「クライアントでメモリが不足しています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_COLUMN_AMBIGUOUS	「カラム '%1' が複数のテーブルで見つかりました。相関名が必要です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_COLUMN_CANNOT_BE_NULL	「テーブル '%2' のカラム '%1' は NULL 値を持つことはできません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_COLUMN_IN_INDEX	「インデックスのカラムを変更することはできません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_COLUMN_NOT_FOUND	「カラム '%1' が見つかりません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_COLUMN_NOT_INDEXED	「カラム '%1' は、それを含んでいるテーブルのどのインデックスにも属していません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_COLUMN_NOT_STREAMABLE	「操作が失敗しました。カラム '%1' のタイプがストリーミングをサポートしていません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_COMMUNICATIONS_ERROR	「通信エラーが発生しました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_CONNECTION_ALREADY_EXISTS	「この接続はすでに存在します。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_CONNECTION_NOT_FOUND	「接続が見つかりません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_CONNECTION_RESTORED	「Ultra Light の接続がリストアされました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_CONSTRAINT_NOT_FOUND	「制約 '%1' が見つかりません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_CONVERSION_ERROR	「値 %1 をデータ型 %2 に変換できません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_COULD_NOT_FIND_FUNCTION	「ダイナミック・ライブラリ '%2' に '%1' が見つかりませんでした。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_COULD_NOT_LOAD_LIBRARY	「ダイナミック・ライブラリ '%1' をロードできませんでした。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_CURSOR_ALREADY_OPEN	「カーソルはすでに開いています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_CURSOR_NOT_DECLARED	「カーソルが宣言されていません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SQLE_CURSOR_NOT_OPEN	「カーソルが開きません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_CURSOR_RESTORED	「Ultra Light のカーソル (あるいは結果セットまたはテーブル) がリストアされました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_CURSOROP_NOT_ALLOWED	「不正なカーソル処理をしようとしました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_DATABASE_ERROR	「データベースの内部エラー %1 -- トランザクションはロールバックされました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_DATABASE_NAME_REQUIRED	「サーバを起動するには、データベース名が必要です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_DATABASE_NOT_CREATED	「データベースの作成に失敗しました : %1」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_DATATYPE_NOT_ALLOWED	「式にサポートされていないデータ型があります。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_DBSPACE_FULL	「データベース領域が最大ファイル・サイズに達しています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_DESCRIBE_NONSELECT	「SELECT 文以外は記述できません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_DEVICE_ERROR	「I/O エラーです。%1 -- トランザクションはロールバックされました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照してください。
SQLE_DEVICE_IO_FAILED	「'%1' のファイル I/O に失敗しました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_DIV_ZERO_ERROR	「ゼロで除算しようとしました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_DOWNLOAD_CONFLICT	「既存のローと競合しているため、ダウンロードに失敗しました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_DOWNLOAD_RESTART_FAILED	「ダウンロードをリトライできません。アップロードが完了していません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_DROP_DATABASE_FAILED	「データベース '%1' の削除に失敗しました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_DUPLICATE_CURSOR_NAME	「カーソル名 '%1' はすでに存在します。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SQLE_DUPLICATE_FOREIGN_KEY	「テーブル '%2' の外部キー '%1' は、既存の外部キーと重複しています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_DUPLICATE_OPTION	「オプション '%1' が複数回指定されています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_DYNAMIC_MEMORY_EXHAUSTED	「動的メモリが足りません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ENCRYPTION_INITIALIZATION_FAILED	「暗号化 DLL を初期化できませんでした : '%1'」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ENCRYPTION_NOT_ENABLED	「暗号化が有効になっていません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ENCRYPTION_NOT_ENABLED_WARNING	「暗号化が有効になっていません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ENGINE_ALREADY_RUNNING	「データベース・サーバはすでに起動しています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ERROR	「実行時 SQL エラーです -- %1」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ERROR_CALLING_FUNCTION	「外部関数の呼び出しのためのリソースを割り付けられませんでした。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ERROR_IN_ASSIGNMENT	「割り当てのエラー」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_EXPRESSION_ERROR	「'%1' 付近に無効な式があります。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_FEATURE_NOT_ENABLED	「呼び出そうとしたメソッドは、お使いのアプリケーションでは使用できません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_FILE_BAD_DB	「指定されたデータベースを開始できません。'%1' は有効なデータベース・ファイルではありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_FILE_IN_USE	「指定されたデータベース・ファイルはすでに使用されています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_FILE_NOT_DB	「指定されたデータベースを開始できません。'%1' はデータベースではありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_FILE_VOLUME_NOT_FOUND	「データベース '%1' に対して指定したファイルシステム・ボリュームが見つかりません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SQLE_FILE_WRONG_VERSION	「指定されたデータベースを開始できません。'%1' は異なるバージョンのソフトウェアで作成されています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_FOREIGN_KEY_NAME_NOT_FOUND	「外部キー '%1' は見つかりません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_IDENTIFIER_TOO_LONG	「識別子 '%1' が長すぎます。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INCORRECT_VOLUME_ID	「'%1' のボリューム ID が不正です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INDEX_NAME_NOT_UNIQUE	「インデックス名 '%1' はユニークではありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INDEX_NOT_FOUND	「インデックス '%1' が見つかりません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INDEX_NOT_UNIQUE	「テーブル '%2' のインデックス '%1' はユニークでなければなりません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INTERRUPTED	「文の実行がユーザによって中断させられました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_CONSTRAINT_REF	「制約 '%1' への参照または操作が無効です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_DESCRIPTOR_INDEX	「記述子のインデックスが正しくありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_DESCRIPTOR_NAME	「SQL 記述子名が正しくありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_DISTINCT_AGGREGATE	「グループ化されたクエリに、複数の異なる集合関数が含まれています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_FOREIGN_KEY	「テーブル '%2' の外部キー '%1' に対応するプライマリ・キーの値がありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_FOREIGN_KEY_DEF	「外部キーのカラム '%1' にプライマリー・キーと異なる定義があります。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_GROUP_SELECT	「'%1' に対する関数またはカラムの参照も GROUP BY 句に記述する必要があります。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_INDEX_TYPE	「'%1' のインデックスの型の指定は不正です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SQLE_INVALID_LOGON	「ユーザ ID またはパスワードが無効です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_OPTION	「オプション '%1' が無効です。-- PUBLIC 設定がありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照してください。
SQLE_INVALID_OPTION_SETTING	「不正なオプション '%1' を設定しています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_OPTION_VALUE	「'%1' は '%2' に対して無効な値です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_ORDER	「ORDER BY 句の指定が不正です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_PARAMETER	「不正なパラメータです。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_PARSE_PARAMETER	「解析エラーです : %1」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_PUBLICATION_MASK	「指定されたパブリケーション・マスクが不正です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_SQL_IDENTIFIER	「SQL の識別子が無効です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_STATEMENT	「文が無効です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_INVALID_UNION	「UNION、INTERSECT、または EXCEPT の select リストの長さが一致していません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_KEYLESS_ENCRYPTION	「このデータベースはキー不使用の暗号化を使用するため、要求された操作を実行できません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_LOCKED	「'%2' のローは、ユーザ '%1' によってロックされています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_MAX_ROW_SIZE_EXCEEDED	「テーブル '%1' の最大ロー・サイズを超過します。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_MEMORY_ERROR	「メモリエラー -- トランザクションはロールバックされました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_METHOD_CANNOT_BE_CALLED	「メソッド '%1' は現時点では呼び出せません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_MIRROR_FILE_MISMATCH	「ミラー '%1' がデータベース '%2' と一致しません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SQLE_NAME_NOT_UNIQUE	「アイテム '%1' はすでに存在しています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_NO_COLUMN_NAME	「抽出されたテーブル '%1' にはカラム %2 に対する名前がありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_NO_CURRENT_ROW	「カーソルの現在のローがありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_NO_INDICATOR	「NULL に対して、インジケータ変数が用意されていません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_NO_MATCHING_SELECT_ITEM	「派生テーブル '%1' の select リストに '%2' と一致する式がありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_NO_PRIMARY_KEY	「テーブル '%1' にはプライマリ・キーが定義されていません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_NOERROR	SQLE_NOERROR(0) – このコードは、エラーまたは警告がなかったことを示します。
SQLE_NON_UPDATEABLE_COLUMN	「式を更新できません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_NON_UPDATEABLE_CURSOR	「FOR UPDATE が READ ONLY カーソルに誤って指定されました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_NOT_IMPLEMENTED	「'%1' の機能は実装されていません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_NOT_SUPPORTED_IN_ULTRALITE	「Ultra Light では使用できない機能です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_NOTFOUND	「ローが見つかりません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ONLY_ONE_TABLE	「カーソルの INSERT/DELETE は、1 度に 1 つのテーブルにしかできません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_OVERFLOW_ERROR	「値 %1 は、対象先にとって大きすぎます。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_PAGE_SIZE_INVALID	「無効なデータベース・ページ・サイズです。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_PARTIAL_DOWNLOAD_NOT_FOUND	「部分ダウンロードが見つかりませんでした。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_PERMISSION_DENIED	「パーミッションがありません : %1」 『SQL Anywhere 10 - エラー・メッセージ』を参照。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SQLE_PRIMARY_KEY_NOT_UNIQUE	「テーブル '%1' のプライマリ・キーがユニークではありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_PRIMARY_KEY_TWICE	「テーブルに 2 つのプライマリ・キーを定義することはできません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_PRIMARY_KEY_VALUE_REF	「テーブル '%1' 内のローのプライマリ・キーがテーブル '%3' 内の外部キー '%2' によって参照されています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_PUBLICATION_NOT_FOUND	「パブリケーション '%1' が見つかりません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_PUBLICATION_PREDICATE_IGNORED	「パブリケーションの述部は評価されませんでした。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_RESOURCE_GOVERNOR_EXCEEDED	「%1 のリソース・ガバナーが制限を超えています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ROW_DELETED_TO_MAINTAIN_REFERENTIAL_INTEGRITY	「参照整合性を保つためにテーブル %1 からローが削除されました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ROW_EXCEEDS_PAGE_SIZE	「ローがデータベースのページ・サイズを超えているため、ローを格納できません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_SCHEMA_UPGRADE_NOT_ALLOWED	「スキーマのアップグレードは現在認められません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_SERVER_SYNCHRONIZATION_ERROR	「サーバ %1 でエラーが発生したため、同期に失敗しました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_START_STOP_DATABASE_DENIED	「データベースの起動／停止の要求は拒否されました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_STATEMENT_ERROR	「SQL 文にエラーがあります。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_STRING_RIGHT_TRUNCATION	「文字列データの右側がトランケートされます。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_SUBQUERY_SELECT_LIST	「select リストの中にカラムが 2 つ以上指定されています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_SYNC_INFO_INVALID	「同期の情報が不完全か無効です。'%1' を確認してください。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_SYNC_INFO_REQUIRED	「同期の情報が指定されていません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_SYNC_NOT_REENTRANT	「同期処理に戻ることができませんでした。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SQLE_SYNC_STATUS_UNKNOWN	「最後の同期アップロードのステータスは不明です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_SYNTAX_ERROR	「'%1' %2 の近くに構文エラーがあります。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_TABLE_ALREADY_INCLUDED	「テーブル '%1' はすでにインクルードされています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_TABLE_IN_USE	「テーブルは使用されています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_TABLE_NOT_FOUND	「テーブル '%1' は見つかりませんでした。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_TOO_MANY_BLOB_REFS	「BLOB への参照が多すぎます。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_TOO_MANY_CONNECTIONS	「データベース・サーバに接続できる限界数を超えています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_TOO_MANY_PUBLICATIONS	「パブリケーション・マスクに指定されているパブリケーションが多すぎます。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_TOO_MANY_TEMP_TABLES	「接続しているテンポラリ・テーブルが多すぎます。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_TOO_MANY_USERS	「データベースのユーザが多すぎます。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ULTRALITE_DATABASE_NOT_FOUND	「データベース '%1' が見つかりませんでした。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ULTRALITE_OBJ_CLOSED	「閉じられたオブジェクトに対する操作は無効です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_ULTRALITE_WRITE_ACCESS_DENIED	「書き込みアクセスが拒否されました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_UNABLE_TO_CONNECT	「データベースが起動できません -- %1」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_UNABLE_TO_START_DATABASE	「指定されたデータベースを起動できません : %1」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_UNABLE_TO_START_DATABASE_VER_NEWER	「指定されたデータベースを起動できません : データベース %1 を起動するにはサーバをアップグレードする必要があります。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_UNCOMMITTED_TRANSACTIONS	「コミットされていないトランザクションの同期またはアップグレードはできません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SQLE_UNKNOWN_FUNC	「関数 '%1' はありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_UNKNOWN_OPTION	「'%1' は認識できないオプションです。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_UNKNOWN_USERID	「'%1' というユーザ ID はありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_UNRECOGNIZED_OPTION	「オプション '%1' が認識されません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_UPLOAD_FAILED_AT_SERVER	「同期サーバがアップロードのコミットに失敗しました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_VALUE_IS_NULL	「要求されたデータ型として NULL の結果を返すことができません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_VARIABLE_INVALID	「無効なホスト変数です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_WRONG_NUM_OF_INSERT_COLS	「INSERT コマンドへの値が正しくありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。
SQLE_WRONG_PARAMETER_COUNT	「関数 '%1' のパラメータ数が誤りです。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。

Copyright © 2007, iAnywhere Solutions, Inc.

ULStreamErrorCode 列挙

UL 拡張： 同期中にストリームによって報告される可能性のあるエラー・コードを列挙します。

構文

Visual Basic

```
Public Enum ULStreamErrorCode
  Inherits Short
```

C#

```
public enum ULStreamErrorCode :
  short
```

メンバ名

メンバ名	説明
ACTSYNC_NO_PORT	ActiveSync 同期を開始できるのは、ActiveSync だけです。開始するには、クレードル内にデバイスを置くか、ActiveSync マネージャの [同期] を選択します。アプリケーションから同期を開始するには、TCP/IP ソケットの同期ストリームを使用してください。
ACTSYNC_NOT_INSTALLED	ActiveSync プロバイダがインストールされていません。dbasinst を実行して ActiveSync プロバイダをインストールしてください (詳細についてはマニュアルを参照してください)。
CONNECT_TIMEOUT	接続試行がタイムアウトになりました。サーバが指定のホストとポートで実行されていないか、タイムアウト値を大きくして接続の試行に使用できる時間を増やす必要があります。
COULD_NOT_OPEN_FILE_FOR_WRITE	指定されたファイルを書き込み用に開くことができませんでした。このファイルが正しいファイルであること、またファイルが他のアプリケーションで使用されていないことを確認してください。
CREATE_RANDOM_OBJECT	セキュア・ネットワーク・レイヤが、乱数生成オブジェクトを作成できませんでした。システム・リソースを解放してから、もう一度接続と操作を行ってください。
DEQUEUING_CONNECTION	キューイングされている接続 (同期) 要求を取得しようとしているときに、Mobile Link サーバでエラーが発生しました。システム・リソースを解放してください。問題が解決しなければ、Mobile Link サーバを再起動してください。
DUN_DIAL_FAILED	指定されたダイヤルアップ・ネットワークに自動ダイヤルアップで接続を確立できませんでした。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
DUN_NOT_SUPPORTED	システム・サポートが十分でないため、ダイヤルアップに失敗しました。PocketPC では cellcore.dll を使用し、Windows では IE 4.0 以降の wininet.dll を使用する必要があります。他のプラットフォームではダイヤルアップはサポートされていません。
END_READ	ネットワークからの一連の読み込みを完了できません。参照：READ
END_WRITE	ネットワークへの一連の書き込みを完了できません。参照：WRITE
GENERATE_RANDOM	セキュア・ネットワーク・レイヤでは乱数が必要ですが、それを生成できませんでした。システム・リソースを解放してから、もう一度接続と操作を行ってください。
HTTP_AUTHENTICATION_FAILED	入力したユーザ ID とパスワードは拒否されました。正しく入力したかどうか確認してください。入力した場合は、正しいアクセス権があることをシステム管理者に確認してください。
HTTP_AUTHENTICATION_REQUIRED	HTTP サーバまたはゲートウェイが HTTP 認証を要求しました。HTTP 同期パラメータ http_userid と http_password を使用してユーザ ID とパスワードを指定してください。
HTTP_BAD_STATUS_CODE	ステータス行を調べて失敗の原因を判別してください。
HTTP_BUFFER_SIZE_OUT_OF_RANGE	HTTP バッファ・サイズを修正してください。有効なバッファ・サイズは正の値で、ホスト・プラットフォームにとって大きすぎないように指定してください。
HTTP_CHUNK_LEN_BAD_CHARACTER	固定長の HTTP 本文を使用してください。
HTTP_CHUNK_LEN_ENCODED_MISSING	固定長の HTTP 本文を使用してください。
HTTP_CLIENT_ID_NOT_SET	クライアント ID が HTTP クライアント・コードに渡されませんでした。修正プログラムについては、弊社製品の保守契約を結んでいるサポート・センタにお問い合わせください。
HTTP_CONTENT_TYPE_NOT_SPECIFIED	未知の内容タイプが指定されました。マニュアルを参照し、サポートされる内容タイプに変更してください。
HTTP_CRLF_ENCODED_MISSING	使用しているプロキシが Mobile Link と互換性がない可能性があります。設定を確認してください。
HTTP_CRLF_MISSING	使用しているプロキシが Mobile Link と互換性がない可能性があります。設定を確認してください。
HTTP_EXPECTED_POST	使用しているプロキシが Mobile Link と互換性がない可能性があります。設定を確認してください。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
HTTP_EXTRA_DATA_END_READ	HTTP 本文に余分なデータが追加されています。このデータは、プロキシ・エージェントによって追加された可能性があります。プロキシを削除してください。
HTTP_HEADER_PARSE_ERROR	HTTP ヘッダを解析しようとしてエラーが発生しました。ヘッダの形式が間違っている可能性があります。
HTTP_INTERNAL_HEADER_STATE	HTTP ヘッダを復号化できませんでした。これは通常は起こるはずのない内部エラーです。保守契約を結んでいるサポート・センタに問い合わせてください。
HTTP_INTERNAL_REQUEST_TYPE	HTTP 要求タイプを判別できませんでした。これは通常は起こるはずのない内部エラーです。保守契約を結んでいるサポート・センタに問い合わせてください。
HTTP_INVALID_CHARACTER	HTTP ヘッダで予期しない文字が読み込まれました。ヘッダの形式が間違っているか、通信先から HTTP が送信されていない可能性があります。
HTTP_INVALID_SESSION_KEY	未知のセッション・キー・タイプが指定されました。マニュアルを参照し、サポートされるセッション・キー・タイプに変更してください。
HTTP_MALFORMED_SESSION_COOKIE	同期セッションの管理用の HTTP cookie が破損しています。cookie が破損している箇所を特定してください。クライアント・エラーか、HTTP の中間が正しく動作していないことが主な原因と考えられます。
HTTP_NO_CONTD_CONNECTION	リモート・サイトから送信される次の HTTP 要求の待機中に、サーバがタイムアウトになりました。この要求がサーバに届かなかった原因を判別するか、永続的な接続を試みてください。
HTTP_NO_PASSWORD	HTTP 認証にユーザ ID が入力されましたが、パスワードが入力されていません。認証には両方が必要です。
HTTP_NO_USERID	HTTP 認証にパスワードが入力されましたが、ユーザ ID が入力されていません。認証には両方が必要です。
HTTP_PROXY_AUTHENTICATION_FAILED	入力したユーザ ID とパスワードがプロキシ・サーバで拒否されました。正しく入力したかどうか確認してください。入力した場合は、正しいアクセス権があることをシステム管理者に確認してください。
HTTP_PROXY_AUTHENTICATION_REQUIRED	HTTP プロキシが HTTP 認証を要求しました。HTTP 同期パラメータ http_proxy_userid と http_proxy_password を使用してユーザ ID とパスワードを指定してください。
HTTP_SERVER_AUTH_FAILED	サーバから送信された認証情報ヘッダに不正な値が含まれているため、認証に失敗しました。正規の HTTP サーバに接続していることを確認してください。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
HTTP_UNABLE_TO_PARSE_COOKIE	破損している set cookie ヘッダを判別してください。
HTTP_UNKNOWN_TRANSFER_ENCODING	未知の転送エンコードがどのように生成されているかを判別してください。
HTTP_UNSUPPORTED_AUTH_ALGORITHM	サーバが要求した HTTP ダイジェスト認証アルゴリズムは、サポートされていません。サポートされているのは「MD5」と「MD5-sess」だけです。
HTTP_VERSION	要求された HTTP バージョンはサポートされていません。マニュアルを調べて、サポート対象の HTTP バージョンを指定してください。パブリケーション時のサポート対象 HTTP バージョンは 1.0 と 1.1 です。
INCONSISTENT_FIPS	Mobile Link サーバのコマンド・ラインで -fips スイッチを使用するには、すべてのセキュア・ストリームが FIPS に準拠している必要があります。セキュア・ストリームに fips オプションが設定されていない場合は、自動的に FIPS 準拠 (fips=y など) になります。セキュア・ストリームから fips オプションを削除するか、fips=y で有効にしてください。
INIT_RANDOM	セキュア・ネットワーク・レイヤが、乱数ジェネレータを初期化できませんでした。システム・リソースを解放してから、もう一度接続と操作を行ってください。
INTERNAL	ネットワーク・レイヤで内部エラーが発生しました。保守契約を結んでいるサポート・センタに問い合わせてください。
INTERNAL_API	ネットワーク・レイヤで内部エラーが発生しました。保守契約を結んでいるサポート・センタに問い合わせてください。
INTERNAL_PROTOCOL_NOT_LOADED	同期プロトコルをロードできませんでした。Ultra Light を使用している場合は、適切な ULEnable メソッドを呼び出していることを確認してください。
INTERRUPTED	現在の操作は呼び出し元によって中断されました。
INVALID_COMPRESSION_TYPE	指定された圧縮型が認識されませんでした。
INVALID_LOCAL_PATH	ダウンロードされたファイルのローカル・パスが無効です。詳細については、マニュアルを参照してください。
INVALID_NETWORK_LIBRARY	指定されたネットワーク・インタフェース DLL または共有オブジェクトがロードできませんでした。不正か破損している可能性があります。
INVALID_SYNC_PROTOCOL	指定されたプロトコルは有効な同期プロトコルではありません。
LIBRARY_ENTRY_POINT_NOT_FOUND	指定されたライブラリのエントリ・ポイントが見つかりませんでした。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
LOAD_LIBRARY_FAILURE	パスでダイナミック・ライブラリを見つけることができませんでした。同期に TLS 暗号化を使用しようとしている場合は、適切なライセンスを取得していることを確認してください。
LOAD_NETWORK_LIBRARY	ネットワーク・インタフェース・ライブラリを検出してロードできませんでした (またはそのいずれか)。次の項目を確認してください。 1) ソケット・レイヤが適切にインストールされている。適切なネットワーク・インタフェース・ライブラリ (または DLL か共有オブジェクト) が存在しており、アクセス可能になっていなければなりません。 2) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。
MEMORY_ALLOCATION	ネットワーク・レイヤで十分なバイト数の記憶領域を割り付けることができませんでした。システム・メモリを解放してもう一度操作を行ってください。システム・メモリを解放する方法は、オペレーティング・システムとその設定方法によって異なります。最も簡単な方法は、アクティブな処理の数を減らすことです。詳細については、使用しているオペレーティング・システムのマニュアルを参照してください。
MISSING_PARAMETER	必要なパラメータが指定されていません。
NETWORK_LIBRARY_VERSION_MISMATCH	バージョンが間違っているため、ネットワーク・インタフェース DLL または共有オブジェクトがロードできませんでした。
NO_ECC_FIPS	この圧縮処理を行うことができませんでした。保守契約を結んでいるサポート・センタに問い合わせてください。
NONE	このコードは、ネットワーク・エラーが発生しなかったか、未知のネットワーク・エラーが発生したことを示します。
NOT_IMPLEMENTED	実装されていない内部機能が要求されました。保守契約を結んでいるサポート・センタに問い合わせてください。
PARAMETER	ネットワーク・パラメータの形式は、"name=value; [name2=value2[;...]]" です。このコードは、不正なパラメータ値を示しています。該当するパラメータ名について説明したマニュアルを参照して、パラメータ値を修正してください。
PARAMETER_NOT_BOOLEAN	ネットワーク・パラメータの形式は、"name=value; [name2=value2[;...]]" です。このパラメータ値はブール値ではありません。パラメータ指定の違反箇所を探して、パラメータ値を 0 (off または False に対応) または 1 (on または True に対応) に変更してください。
PARAMETER_NOT_HEX	ネットワーク・パラメータの形式は、"name=value; [name2=value2[;...]]" です。このパラメータ値は、16 進数値 (基数 16) ではありません。パラメータ指定の違反箇所を探して、パラメータ値を 16 進数値に変更してください。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
PARAMETER_NOT_UINT32	ネットワーク・パラメータの形式は、"name=value; [name2=value2[;...]]" です。このパラメータ値は、符号なし整数ではありません。パラメータ指定の違反箇所を探して、パラメータ値を符号なしの整数に変更してください。
PARAMETER_NOT_UINT32_RANGE	ネットワーク・パラメータの形式は、"name=value; [name2=value2[;...]]" です。このパラメータ値は、符号なし整数値または範囲のいずれでもありません。パラメータ指定の違反箇所を探して、パラメータ値を符号なしの整数値か範囲に変更してください。符号なし範囲の形式は NNN-NNN です。
PARSE	ネットワーク・パラメータの形式は、"name=value; [name2=value2[;...]]" です。オプションで、カッコ内にパラメータの全リストを入力できます。入力された文字列はこの表記規則に従っていません。文字列を検査して、フォーマット上の問題を修正してからもう一度操作を行ってください。
PROTOCOL_ERROR	予期しない値またはトークンが読み込まれました。
READ	ネットワーク・レイヤによって指定されたバイト数を読み込めませんでした。読み込みは、より規模の大きいネットワーク操作の一部として行われます。たとえば、ネットワーク・レイヤによっては、サブ・レイヤを設定して、ここで上位レイヤの基本操作の一部として読み込みと書き込みを行います。通常、読み込みエラーは次のような事柄が原因で発生します。 1) ネットワークに、読み込みを失敗させる問題が生じた。もう一度接続と操作を行ってください。 2) 接続がタイムアウトした。もう一度接続と操作を行ってください。 3) 反対側の接続が完全に終了した。クライアントかサーバ (またはその両方) のエラー・ログを参照して、接続が削除された理由を調べてください。出力されたエラー・ログを調べて原因を解明し、もう一度操作を行ってください。 4) 接続の反対側で処理がアボートされた。クライアントかサーバ (またはその両方) のエラー出力ログを参照して、処理がアボートされた理由を調べてください。通常の方法以外のやり方で処理が停止された場合は、出力ログにエラーは記録されません。もう一度接続と操作を行ってください。 5) システム・リソースが不足しており、読み込みを実行できない。システム・リソースを解放してから、もう一度接続と操作を行ってください。再度行ってもうまくいかない場合は、ネットワーク管理者に問い合わせてください。
READ_TIMEOUT	ネットワーク・レイヤによって指定されたバイト数を指定時間内に読み込めませんでした。ネットワークが正常に機能しており、送信側アプリケーションがまだ動作中であることを確認してください。
SACI_ERROR_CLIENT	SACI 実装によってエラーがレポートされました。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SACI_ERROR_SERVER	SACI ネットワーク・ライブラリがエラーをレポートしています。SACI ネットワーク・ライブラリのプロバイダを参照して、問題を解決してください。
SECURE_ADD_CERTIFICATE	セキュア・ネットワーク・レイヤが、証明書チェーンに証明書を追加できませんでした。システム・リソースを解放して、もう一度操作を行ってください。
SECURE_ADD_TRUSTED_CERTIFICATE	セキュア・ネットワーク・レイヤが、証明書チェーンに信用された証明書を追加できませんでした。システム・リソースの不足が主な原因と考えられます。システム・リソースを解放して、もう一度操作を行ってください。
SECURE_CERTIFICATE_COMMON_NAME	指定された共通名は証明書チェーンに存在しません。次の項目を確認してください。 1) 共通名が正しく入力されている。 2) 適切な証明書ファイルが指定されている。 3) 証明書チェーンに共通名が存在している。readcert ユーティリティを使用して上記を確認できます。
SECURE_CERTIFICATE_COMPANY_NAME	指定された組織名は証明書チェーンに存在しません。次の項目を確認してください。 1) 組織名が正しく入力されている。 2) 適切な証明書ファイルが指定されている。 3) 証明書チェーンに組織名が存在している。readcert ユーティリティを使用して上記を確認できます。
SECURE_CERTIFICATE_COMPANY_UNIT	指定された組織単位は証明書チェーンに存在しません。次の項目を確認してください。 1) 会社名が正しく入力されている。 2) 適切な証明書ファイルが指定されている。 3) 証明書チェーンに会社名が存在している。readcert ユーティリティを使用して上記を確認できます。
SECURE_CERTIFICATE_COUNT	指定されたファイルに証明書が入っていません。次の項目を確認してください。 1) 証明書ファイル名が適切に指定されている。 2) 証明書ファイルに証明書が 1 つ以上入っている。 3) 証明書ファイルに適切な証明書が入っている。
SECURE_CERTIFICATE_EXPIRED	証明書チェーンにある証明書の有効期限が切れています。有効期限が切れていない新しい証明書を入手して、再度操作を行ってください。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SECURE_CERTIFICATE_EXPIRY_DATE	証明書の有効期日を読み込むことができませんでした。次の項目を確認してください。 1) パスワードが正しく入力されている。 2) 証明書ファイルに証明書が 1 つ以上入っている。 3) 証明書ファイルに適切な証明書が入っている。 4) 証明書ファイルは破損していない。
SECURE_CERTIFICATE_FILE_NOT_FOUND	証明書ファイルを開けられませんでした。次の項目を確認してください。 1) 証明書ファイル名が適切に指定されている。 2) 証明書ファイルが存在する。 3) 証明書ファイルに証明書が 1 つ以上入っている。 4) 証明書ファイルに適切な証明書が入っている。 5) 証明書ファイルを開こうとしているプログラムにファイルを読み込む権限がある。この項目は、ユーザ・パーミッションかファイル・パーミッション (またはその両方) を持つオペレーティング・システムだけに適用されます。
SECURE_CERTIFICATE_NOT_TRUSTED	サーバの証明書は信用される権限保持者によって署名されていません。次の項目を確認してください。 1) 証明書ファイル名が適切に指定されている。 2) 証明書ファイルに証明書が 1 つ以上入っている。 3) 証明書ファイルに適切な証明書が入っている。 4) 信用されるルート証明書のクライアント側のリストに、サーバ側のルート証明書が含まれている。
SECURE_CERTIFICATE_ROOT	チェーンのルート証明書が不正です。このエラーは、パブリケーション時に定義されましたが使用されませんでした。
SECURE_CREATE_CERTIFICATE	セキュア・ネットワーク・レイヤが、証明書用の記憶領域を割り付けることができませんでした。システム・リソースを解放して、もう一度操作を行ってください。
SECURE_CREATE_PRIVATE_KEY_OBJECT	プライベート・キーを読み込む前に、セキュア・ネットワーク・レイヤがプライベート・キーのオブジェクトを作成できませんでした。システム・リソースの不足が主な原因と考えられます。システム・リソースを解放して、もう一度操作を行ってください。
SECURE_DUPLICATE_CONTEXT	セキュア・ネットワーク・レイヤが、セキュリティ・コンテキストを複製できませんでした。システム・リソースを解放して、もう一度操作を行ってください。

メンバ名	説明
SECURE_EXPORT_CERTIFICATE	セキュア・ネットワーク・レイヤが、証明書をコピーできませんでした。システム・リソースを解放して、もう一度操作を行ってください。
SECURE_HANDSHAKE	安全なハンドシェイクに失敗しました。次の項目を確認してください。 1) クライアントでは、正しいホスト・マシンとポート番号が指定されている。 2) サーバでは、正しいポート番号が指定されている。 3) クライアントとサーバの両方で、正しい証明書ファイルが指定されている。
SECURE_IMPORT_CERT_FROM_SYSTEM_STORE	システム証明書ストアから証明書をインポートできませんでした。
SECURE_IMPORT_CERTIFICATE	セキュア・ネットワーク・レイヤが、証明書をインポートできませんでした。次の項目を確認してください。 1) 証明書ファイル名が適切に指定されている。 2) 証明書ファイルが存在する。 3) 証明書ファイルに証明書が 1 つ以上入っている。 4) 証明書ファイルに適切な証明書が入っている。
SECURE_NO_CERTS_IN_SYS_STORE	システム証明書ストアで証明書が見つかりませんでした。
SECURE_NO_SERVER_CERTIFICATE	サーバの証明書が指定されていません。通信のセキュリティを保護するには、サーバの証明書が必要です。指定するファイルには、サーバの証明書とプライベート・キーが含まれる必要があります。
SECURE_NO_SERVER_CERTIFICATE_PASSWORD	サーバ証明書のパスワードが入力されていません。サーバの暗号化されたプライベート・キーを復号化するには、パスワードが必要です。
SECURE_NO_TRUSTED_ROOTS	信用できるルート証明書が指定されていません。通信のセキュリティを保護するには、信用できるルート証明書が少なくとも 1 つ必要です。
SECURE_OPEN_SYSTEM_CERT_STORE	システム証明書ストアを開けませんでした。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SECURE_READ_CERTIFICATE	証明書ファイルが読み込めませんでした。次の項目を確認してください。 1) パスワードが正しく入力されている。 2) 証明書ファイルに証明書が 1 つ以上入っている。 3) 証明書ファイルに適切な証明書が入っている。 4) 証明書ファイルは破損していない。
SECURE_READ_PRIVATE_KEY	証明書ファイルからプライベート・キーを読み込めませんでした。次の項目を確認してください。 1) パスワードが正しく入力されている。 2) 証明書ファイルに証明書が 1 つ以上入っている。 3) 証明書ファイルに適切な証明書が入っている。 4) 証明書ファイルは破損していない。
SECURE_REDUNDANT_SERVER_CERTIFICATE_PASSWORD	サーバのプライベート・キーがパスワードで暗号化されていないのに、パスワードが指定されました。
SECURE_SET_IO	セキュア・ネットワーク・レイヤをネットワーク・レイヤに付加できませんでした。システム・リソースを解放して、もう一度操作を行ってください。
SECURE_SET_PRIVATE_KEY	プライベート・キーを使用できませんでした。次の項目を確認してください。 1) パスワードが正しく入力されている。 2) 証明書ファイルに証明書が 1 つ以上入っている。 3) 証明書ファイルに適切な証明書が入っている。 4) 証明書ファイルは破損していない。
SECURE_TRUSTED_CERTIFICATE_FILE_NOT_FOUND	証明書ファイルが見つかりませんでした。次の項目を確認してください。 1) 証明書ファイル名が適切に指定されている。 2) 証明書ファイルが存在する。 3) 証明書ファイルに証明書が 1 つ以上入っている。 4) 証明書ファイルに適切な証明書が入っている。 5) 証明書ファイルを開こうとしているプログラムにファイル参照の権限がある。この項目は、ユーザ・パーミッションかファイル・パーミッション (またはその両方) を持つオペレーティング・システムだけに適用されます。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SECURE_TRUSTED_CERTIFICATE_READ	セキュア・ネットワーク・レイヤが、信用された証明書ファイルを読み込むことができませんでした。次の項目を確認してください。 1) 証明書ファイル名が適切に指定されている。 2) 証明書ファイルが存在する。 3) 証明書ファイルに証明書が 1 つ以上入っている。 4) 証明書ファイルに適切な証明書が入っている。 5) 証明書ファイルを開こうとしているプログラムにファイル参照の権限がある。この項目は、ユーザ・パーミッションかファイル・パーミッション (またはその両方) を持つオペレーティング・システムだけに適用されます。
SEED_RANDOM	セキュア・ネットワーク・レイヤが、乱数ジェネレータにシードを設定できませんでした。システム・リソースを解放してから、もう一度接続と操作を行ってください。
SERVER_ERROR	サーバによってエラーがレポートされました。詳細については、Mobile Link の管理者に問い合わせてください。
SHUTTING_DOWN	シャットダウン中、Mobile Link サーバがネットワーク・レイヤでエラーを検出しました。シャットダウン時に保留になっているネットワーク操作が影響を受けた可能性があります。
SOCKET_BIND	ネットワーク・レイヤが、指定ポートにソケットをバインドできませんでした。次の項目を確認してください。 1) (サーバのみ) ポートがまだ使用されていないことを確認する。ポートが使用中の場合は、そのポートで受信しているアプリケーションを停止するか、別のポートを指定してください。 2) (サーバのみ) ポートの使用に関するファイアウォールの制限事項がないことを確認する。 3) (クライアントのみ) client_port オプションが使用された場合は、指定したポートがまだ使用されていないことを確認する。クライアント・ポートを 1 つだけ指定した場合は、NNN-NNN などの範囲を使用することを検討してください。範囲が指定されていた場合は、その範囲を広げるか、別の範囲にすることを検討してください。 4) (クライアントのみ) client_port オプションが使用された場合は、ポートの使用に関するファイアウォールの制限事項がないことを確認する。
SOCKET_CLEANUP	ネットワーク・レイヤがソケット・レイヤをクリーンアップできませんでした。これは、すべての接続が終了した後にだけ発生するエラーです。したがって現在の接続には影響しません。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SOCKET_CLOSE	ネットワーク・レイヤがソケットを閉じることができませんでした。フラッシュされなかった保留中の書き込みが原因で、ネットワーク・セッションが不完全なまま終了した可能性があります (または、終了していない可能性があります)。次の項目を確認してください。 1) ネットワーク接続のどちらか一方の側でエラーが発生した。 2) 接続の反対側は正常に動作している。 3) マシンがネットワークに接続されたままになっており、ネットワークからの応答がある。
SOCKET_CONNECT	ネットワーク・レイヤがソケットを接続できませんでした。次の項目を確認してください。 1) マシンがネットワークに接続されている。 2) ソケット・レイヤは適切に初期化されている。 3) 適切なホスト・マシンとポートが指定されている。 4) ホスト・サーバが正常に稼働しており、適切なポートで受信している。 5) ホスト・マシンが、適切なソケット・タイプ (TCP/IP と UDP) で受信を行っている。 6) client_port オプションが使用された場合は、ポートの使用に関するファイアウォールの制限事項がないことを確認する。 7) オープンしているソケット数について、デバイスに制限が設けられている場合は、その値に達していないことを確認する。 8) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。
SOCKET_CREATE_TCPIP	ネットワーク・レイヤが TCP/IP ソケットを作成できませんでした。次の項目を確認してください。 1) マシンがネットワークに接続されている。 2) ソケット・レイヤは適切に初期化されている。 5) オープンしているソケット数について、デバイスに制限が設けられている場合は、その値に達していないことを確認する。 6) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SOCKET_CREATE_UDP	ネットワーク・レイヤが UDP ソケットを作成できませんでした。次の項目を確認してください。 1) マシンがネットワークに接続されている。 2) ソケット・レイヤは適切に初期化されている。 3) client_port オプションが使用された場合は、指定したポートがまだ使用されていないことを確認する。クライアント・ポートを 1 つだけ指定した場合は、NNN-NNN などの範囲を使用することを検討してください。範囲が指定されていた場合は、その範囲を広げるか、別の範囲にすることを検討してください。 4) client_port オプションが使用された場合は、ポートの使用に関するファイアウォールの制限事項がないことを確認する。 5) オープンしているソケット数について、デバイスに制限が設けられている場合は、その値に達していないことを確認する。 6) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。
SOCKET_GET_HOST_BY_ADDR	ネットワーク・レイヤが、IP アドレスを使用してホスト名を取得できませんでした。このエラーは、パブリケーション時に定義されましたが使用されませんでした。
SOCKET_GET_NAME	ネットワーク・レイヤがソケットのローカル名を判別できませんでした。TCP/IP 接続には、各接続の両端にポート専用に付加されたソケットがあります。ソケットのローカル名にはこのポート番号が含まれており、これは、接続時にネットワークによって割り当てられます。次の項目を確認してください。 1) マシンがネットワークに接続されたままになっており、ネットワークからの応答がある。 2) 接続の反対側は正常に動作している。 3) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。
SOCKET_GET_OPTION	ネットワーク・レイヤがソケット・オプションを取得できませんでした。このエラーは、接続が失われたことを示す最初の徴候であると考えられます。次の項目を確認してください。 1) マシンがネットワークに接続されたままになっており、ネットワークからの応答がある。 2) 接続の反対側は正常に動作している。 3) 使用できるシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SOCKET_HOST_NAME_NOT_FOUND	指定されたホスト名が見つかりませんでした。次の項目を確認してください。 1) ホスト名が正しく指定されている。 2) ホストがアクセス可能である。多くのシステムには、名前を指定したホストへのアクセスを検証できる ping ユーティリティが組み込まれています。 3) ドメイン・ネーム・サーバ (DNS) またはそれに相当するものが使用できる。DNS を使用できない場合は、ホスト名の代わりにホストの IP 番号 (例：NNN.NNN.NNN.NNN) を指定してみてください。 4) HOSTS ファイルに、ホスト名を IP 番号にマッピングするエントリが入っている。
SOCKET_LISTEN	サーバがソケットで受信できません。バックログとは、ある一定期間に保留中になっている可能性がある、キューイングされた接続要求の最大数のことです。次の項目を確認してください。 1) マシンがネットワークに接続されたままになっており、ネットワークからの応答がある。 2) 現在のマシンでソケット・リスナが動作するのを妨げるファイアウォールやその他の制限事項がない。 3) マシンに制限がある場合、バックログ設定が制限内に収まっている。 4) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。
SOCKET_LIVENESS_OUT_OF_RANGE	無効な活性タイムアウト値が指定されました。活性タイムアウト値は、0 ー 65535 の間の整数にしてください。
SOCKET_LOCALHOST_NAME_NOT_FOUND	ネットワーク・レイヤが "localhost" の IP アドレスを判別できませんでした。次の項目を確認してください。 1) ドメイン・ネーム・サーバ (DNS) またはそれに相当するものが使用できる。DNS が使用できない場合には、代わりに localhost の IP 番号を明示的に指定してみます (通常は 127.0.0.1)。 2) HOSTS ファイルに、"localhost" 名を IP 番号にマッピングするエントリが入っている。 3) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。
SOCKET_PORT_OUT_OF_RANGE	不正なポート番号を指定しています。ポート番号は、0 ー 65535 の間の整数にしてください。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SOCKET_SELECT	ネットワーク・レイヤで、読み込みまたは書き込み可能状態のソケットに対して待ち状態に入ろうとするエラーが発生しました。次の項目を確認してください。 1) ネットワークにマシンが接続されており、ネットワークからの応答がある。 2) 接続の反対側は正常に動作している。 3) 使用できるシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。
SOCKET_SET_OPTION	ネットワーク・レイヤがソケット・オプションを設定できませんでした。このエラーは、接続が失われたことを示す最初の徴候であると考えられます。次の項目を確認してください。 1) マシンがネットワークに接続されたままになっており、ネットワークからの応答がある。 2) 接続の反対側は正常に動作している。 3) 使用できるシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。
SOCKET_SHUTDOWN	ネットワーク・レイヤがソケットをシャトダウンできませんでした。次の項目を確認してください。 1) ネットワークにマシンが接続されており、ネットワークからの応答がある。 2) 接続の反対側は正常に動作している。 3) 使用できるシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。
SOCKET_STARTUP	ネットワーク・レイヤがソケット・レイヤを初期化できませんでした。次の項目を確認してください。 1) ソケット・レイヤが適切にインストールされている。正しいネットワーク・インタフェース・ライブラリが存在し、アクセス可能になっていなければなりません。 2) ネットワークにマシンが接続されており、ネットワークからの応答がある。 3) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。
UNEXPECTED_HTTP_REQUEST_TYPE	この HTTP 要求タイプはこの時点では予期されていませんでした。Mobile Link クライアントではない HTTP クライアントが主な原因と考えられます。
UNRECOGNIZED_TLS_TYPE	TLS タイプが無効です。有効なタイプについては、マニュアルを参照してください。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
VALUE_OUT_OF_RANGE	指定された値が、パラメータに許可される値の範囲内ではありませんでした。マニュアルでパラメータの説明を参照して、値に許可される範囲を確認してください。
WOULD_BLOCK	ブロッキングが不要または予期されないところで、要求された操作がブロックした可能性があります。
WRITE	指定されたバイト数をネットワーク・レイヤに書き込むことができません。書き込みは、より規模の大きいネットワーク操作の一部として行われます。たとえば、ネットワーク・レイヤによっては、サブ・レイヤを設定して、ここで上位レイヤの基本操作の一部として読み込みと書き込みを行います。通常、書き込みエラーは次のような事柄が原因で発生します。 1) ネットワークに、書き込みを失敗させる問題が生じた。もう一度接続と操作を行ってください。 2) 接続がタイムアウトした。もう一度接続と操作を行ってください。 3) 反対側の接続が完全に終了した。クライアントかサーバ (またはその両方) のエラー・ログを参照して、接続が削除された理由を調べてください。出力されたエラー・ログを調べて原因を解明し、もう一度操作を行ってください。 4) 接続の反対側で処理がアボートされた。クライアントかサーバ (またはその両方) のエラー出力ログを参照して、処理がアボートされた理由を調べてください。通常の方法以外のやり方で処理が停止された場合は、出力ログにエラーは記録されません。もう一度接続と操作を行ってください。 5) システム・リソースが不足しており、書き込みを実行できない。システム・リソースを解放してから、もう一度接続と操作を行ってください。再度行ってもうまくいかない場合は、ネットワーク管理者に問い合わせてください。
WRITE_TIMEOUT	ネットワーク・レイヤによって指定されたバイト数を指定時間内に書き込めませんでした。ネットワークが正常に機能しており、受信側アプリケーションがまだ動作中であることを確認してください。

参照

- 「StreamErrorCode プロパティ」 428 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULStreamErrorContext 列挙

UL 拡張： ストリーム・エラーが発生したときに実行される基本的なネットワーク・オペレーションを列挙します。

構文

Visual Basic

Public Enum **ULStreamErrorContext**
  Inherits Short

C#

public enum **ULStreamErrorContext** :
  short

メンバ名

メンバ名	説明
CLOSE	CLOSE(6)
CREATE	CREATE(3)
DESTROY	DESTROY(4)
END_READ	END_READ(11)
END_WRITE	END_WRITE(10)
GETVALUE	GETVALUE(14)
OPEN	OPEN(5)
READ	READ(7)
REGISTER	REGISTER(1)
SOFTSHUTDOWN	SOFTSHUTDOWN(13)
UNKNOWN	UNKNOWN(0)
UNREGISTER	UNREGISTER(2)
WRITE	WRITE(8)
WRITE_FLUSH	WRITE_FLUSH(9)
YIELD	YIELD(12)

参照

- 「StreamErrorContext プロパティ」 428 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

ULStreamErrorID 列挙

UL 拡張： 同期中にエラーを報告する可能性のあるネットワーク・レイヤを列挙します。

構文

Visual Basic

```
Public Enum ULStreamErrorID
  Inherits Short
```

C#

```
public enum ULStreamErrorID :
  short
```

メンバ名

メンバ名	説明
ACTIVESYNC	ACTIVESYNC(22)
CERTICOM	CERTICOM(11)
CERTICOM_SSL	CERTICOM_SSL(13)
CERTICOM_TLS	CERTICOM_TLS(14)
DH_CAST	DH_CAST(9)
EMAIL	EMAIL(20)
FAKE	FAKE(2)
FILE	FILE(21)
HTTP	HTTP(7)
HTTPS	HTTPS(8)
JAVA_CERTICOM	JAVA_CERTICOM(12)
JAVA_RSA	JAVA_RSA(24)
NETTECH	NETTECH(5)
OPEN_SSL_RSA	OPEN_SSL_RSA(25)
PALM_CONDUIT	PALM_CONDUIT(3)
PALM_SS	PALM_SS(4)
PALM_SSL	PALM_SSL(26)
REPLAY	REPLAY(17)

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
RIMBB	RIMBB(6)
RSA_TLS	RSA_TLS(23)
SECURE	SECURE(10)
SERIAL	SERIAL(1)
STRM	STRM(18)
TCPIP	TCPIP(0)
[UDP]	UDP(19)
WIRELESS	WIRELESS(16)
WIRESTRM	WIRESTRM(15)

参照

- 「StreamErrorID プロパティ」 429 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULStreamType 列挙

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。

構文

Visual Basic

```
Public Enum ULStreamType
  Inherits Short
```

C#

```
public enum ULStreamType :
  short
```

備考

特定のストリームのタイプの設定方法については、「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」『Mobile Link - クライアント管理』を参照してください。

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 承認の暗号化には、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 10 - 紹介』を参照してください。

メンバ名

メンバ名	説明
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 サーバは Web サーバとして機能します。Ultra Light アプリケーションは、サーバへのデータ送信のために POST 要求を送り、サーバ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・セキュリティを適用した HTTP)。
TCPIP	TCP/IP を介して同期します。
TLS	TLS (TCP/IP を介して同期)

参照

- 「Stream プロパティ」 406 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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を表します。

構文

Visual Basic

NotInheritable Public Class **ULSyncParms**

C#

public sealed class **ULSyncParms**

備考

このクラスにはコンストラクタがありません。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

- 「ULSyncParms メンバ」 397 ページ
- 「ULConnection クラス」 90 ページ
- 「SyncParms プロパティ」 101 ページ
- 「Synchronize メソッド」 119 ページ
- 「DownloadOnly プロパティ」 399 ページ
- 「PingOnly プロパティ」 403 ページ
- 「ResumePartialDownload プロパティ」 404 ページ
- 「UploadOnly プロパティ」 408 ページ
- 「ULSQLCode 列挙」 366 ページ
- 「Synchronize メソッド」 119 ページ
- 「Stream プロパティ」 406 ページ
- 「Version プロパティ」 409 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULSyncParms メンバ

パブリック・プロパティ

メンバ名	説明
AuthenticationParms プロパティ	カスタム・ユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。
CheckpointStore プロパティ	クライアントが、追加のストア・チェックポイントを実行して、同期中にデータベース・ストアのサイズの増加を制御するかどうかを指定します。
DisableConcurrency プロパティ	同期の実行中に Ultra Light への同時アクセスを無効にするか、有効にするかを指定します。
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 サーバに ping のみを行うかどうかを指定します。
PublicationMask 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 サーバにダウンロード確認を送信するかどうかを指定します。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。
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を指定します。
TableOrder プロパティ	統合データベースにテーブルがアップロードされる順序を指定します。
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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
UserName プロパティ	Mobile Link サーバが Mobile Link クライアントをユニークに識別するユーザ名です。
Version プロパティ	使用する同期スクリプトを指定します。

パブリック・メソッド

メンバ名	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。

AuthenticationParms プロパティ

カスタム・ユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。

構文

Visual Basic

Public Property **AuthenticationParms** As String()

C#

public string [] **AuthenticationParms** {get;set;}

プロパティ値

それぞれに認証パラメータが格納された、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、認証パラメータは指定されません。

備考

最初の 255 文字のみが使用されます。また、各文字列は 128 文字以下であることが必要です (長すぎる文字列は、Mobile Link サーバに送信されるときにトランケートされます)。

CheckpointStore プロパティ

クライアントが、追加のストア・チェックポイントを実行して、同期中にデータベース・ストアのサイズの増加を制御するかどうかを指定します。

構文

Visual Basic

Public Property **CheckpointStore** As Boolean

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public bool **CheckpointStore** {get;set;}

プロパティ値

クライアントが追加のストア・チェックポイントを実行する必要があると指定する場合は true。デフォルトは false で、必要なチェックポイントのみが実行されます。

備考

チェックポイント操作は、アプリケーションの I/O 操作を増やすため、同期が低速になります。このオプションは、多くの更新を伴う大量のダウンロードに最適です。低速なフラッシュ・メモリを使用するデバイスでは、追加のチェックポイントの実行によって、望ましくないパフォーマンスの低下が生じることがあります。

DisableConcurrency プロパティ

同期の実行中に Ultra Light への同時アクセスを無効にするか、有効にするかを指定します。

構文

Visual Basic

Public Property **DisableConcurrency** As Boolean

C#

public bool **DisableConcurrency** {get;set;}

プロパティ値

同期の実行中に Ultra Light への同時アクセス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。

備考

デフォルトでは、1 つのスレッドが同期しているときに、別のスレッドが Ultra Light の操作を行うことがあります。同時同期が無効になっていると、別のスレッドは同期が完了するまで Ultra Light の呼び出しをブロックします。

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。

構文

Visual Basic

Public Property **DownloadOnly** As Boolean

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public bool **DownloadOnly** {get;set;}

プロパティ値

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。

備考

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 クラス」 396 ページ
- 「ULSyncParms メンバ」 397 ページ
- 「UploadOnly プロパティ」 408 ページ
- 「DownloadOnly プロパティ」 399 ページ
- 「PingOnly プロパティ」 403 ページ
- 「ResumePartialDownload プロパティ」 404 ページ
- 「UploadOnly プロパティ」 408 ページ
- 「ULSQLCode 列挙」 366 ページ
- 「Synchronize メソッド」 119 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。

構文

Visual Basic

Public Property **KeepPartialDownload** As Boolean

C#

public bool **KeepPartialDownload** {get;set;}

プロパティ値

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。

備考

Ultra Light.NET では、通信エラーや、ユーザによる 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

 Copyright © 2007, iAnywhere Solutions, Inc.

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 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、PublicationMask 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 フィールドと DisableConcurrency 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。

詳細については、「失敗したダウンロードの再開」 『Mobile Link - サーバ管理』を参照してください。

参照

- 「ULSyncParms クラス」 396 ページ
- 「ULSyncParms メンバ」 397 ページ
- 「PartialDownloadRetained プロパティ」 427 ページ
- 「ResumePartialDownload プロパティ」 404 ページ
- 「RollbackPartialDownload メソッド」 117 ページ
- 「ResumePartialDownload プロパティ」 404 ページ
- 「UserName プロパティ」 408 ページ
- 「DisableConcurrency プロパティ」 399 ページ
- 「RollbackPartialDownload メソッド」 117 ページ

NewPassword プロパティ

UserName で指定されたユーザの新しい Mobile Link パスワードを指定します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **NewPassword** As String

C#

public string **NewPassword** {get;set;}

プロパティ値

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。

備考

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 クラス」 396 ページ
- 「ULSyncParms メンバ」 397 ページ
- 「UserName プロパティ」 408 ページ

Password プロパティ

UserName で指定されたユーザの Mobile Link パスワードです。

構文

Visual Basic

Public Property **Password** As String

C#

public string **Password** {get;set;}

プロパティ値

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。

備考

Mobile Link ユーザ名とパスワードは他のデータベース・ユーザ 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 サーバに対して識別し、認証するために使用されます。

参照

- 「ULSyncParms クラス」 396 ページ
- 「ULSyncParms メンバ」 397 ページ
- 「NewPassword プロパティ」 401 ページ
- 「UserName プロパティ」 408 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

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 サーバに ping のみを行うかどうかを指定します。

構文

Visual Basic

Public Property **PingOnly** As Boolean

C#

public bool **PingOnly** {get;set;}

プロパティ値

クライアントが Mobile Link サーバに 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。

備考

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 クラス」 396 ページ
- 「ULSyncParms メンバ」 397 ページ
- 「DownloadOnly プロパティ」 399 ページ
- 「PingOnly プロパティ」 403 ページ
- 「ResumePartialDownload プロパティ」 404 ページ
- 「UploadOnly プロパティ」 408 ページ
- 「ULSQLCode 列挙」 366 ページ
- 「Synchronize メソッド」 119 ページ

PublicationMask プロパティ

同期させるパブリケーションを指定します。

構文

Visual Basic

Public Property **PublicationMask** As Integer

C#

public int **PublicationMask** {get;set;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

パブリケーション・マスク のビット処理の組み合わせとして、特別値 ULPublicationSchema.SYNC_ALL_PUBS、または特別値 ULPublicationSchema.SYNC_ALL_DB。デフォルトは ULPublicationSchema.SYNC_ALL_DB です。パブリケーション・マスクの詳細については、ULPublicationSchema クラスを参照してください。

参照

- 「ULSyncParms クラス」 396 ページ
- 「ULSyncParms メンバ」 397 ページ
- 「Mask プロパティ」 323 ページ
- 「SYNC_ALL_PUBS フィールド」 322 ページ
- 「SYNC_ALL_DB フィールド」 322 ページ
- 「SYNC_ALL_DB フィールド」 322 ページ
- 「ULPublicationSchema クラス」 321 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。

構文

Visual Basic

Public Property **ResumePartialDownload** As Boolean

C#

public bool **ResumePartialDownload** {get;set;}

プロパティ値

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。

備考

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。

部分的なダウンロードの詳細については、KeepPartialDownload プロパティを参照してください。

参照

- 「ULSyncParms クラス」 396 ページ
- 「ULSyncParms メンバ」 397 ページ
- 「DownloadOnly プロパティ」 399 ページ
- 「PingOnly プロパティ」 403 ページ
- 「ResumePartialDownload プロパティ」 404 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「UploadOnly プロパティ」 408 ページ
- 「ULSQLCode 列挙」 366 ページ
- 「Synchronize メソッド」 119 ページ
- 「PartialDownloadRetained プロパティ」 427 ページ

SendColumnNames プロパティ

同期中に、クライアントが Mobile Link サーバにカラム名を送信するかどうかを指定します。

構文

Visual Basic

Public Property **SendColumnNames** As Boolean

C#

public bool **SendColumnNames** {get;set;}

プロパティ値

クライアントがカラム名を Mobile Link サーバ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。

備考

カラム名は、Mobile Link サーバ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 サーバがダイレクト・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 サーバにダウンロード確認を送信するかどうかを指定します。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。

構文

Visual Basic

Public Property **SendDownloadAck** As Boolean

C#

public bool **SendDownloadAck** {get;set;}

プロパティ値

True に設定すると、クライアントが Mobile Link サーバにダウンロード確認を送信することを指定します。False に設定すると、ダウンロード確認は送信されません。デフォルト値は False です。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

クライアントがダウンロード確認を送信する場合、Mobile Link サーバのデータベース・ワーカ・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。

構文

Visual Basic

Public Property **Stream** As ULStreamType

C#

public ULStreamType **Stream** {get;set;}

プロパティ値

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。

備考

ほとんどの同期ストリームでは、Mobile Link サーバ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が必要です。これらのパラメータは、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

- 「ULSyncParms クラス」 396 ページ
- 「ULSyncParms メンバ」 397 ページ
- 「ULStreamType 列挙」 395 ページ
- 「StreamParms プロパティ」 406 ページ
- 「ULStreamType 列挙」 395 ページ

StreamParms プロパティ

同期ストリームの設定パラメータを指定します。

構文

Visual Basic

Public Property **StreamParms** As String

 Copyright © 2007, iAnywhere Solutions, Inc.

C#

public string **StreamParms** {get;set;}

プロパティ値

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を指定する文字列。デフォルト値は null 参照 (Visual Basic の Nothing) です。

備考

特定のストリームのタイプの設定方法については、「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」 『Mobile Link - クライアント管理』を参照してください。

StreamParms は、同期ストリームに使用されるすべてのパラメータが含まれている文字列です。パラメータは、名前=値の組み合わせ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 クラス」 396 ページ
- 「ULSyncParms メンバ」 397 ページ
- 「Stream プロパティ」 406 ページ
- 「ULStreamType 列挙」 395 ページ

TableOrder プロパティ

統合データベースにテーブルがアップロードされる順序を指定します。

構文

Visual Basic

Public Property **TableOrder** As String

C#

public string **TableOrder** {get;set;}

プロパティ値

テーブル名がカンマで区切られたリスト形式の文字列。テーブル名は、一重または二重引用符で囲むことができます。デフォルト値は null 参照 (Visual Basic の Nothing) で、テーブルのデフォルトの順序は上書きされません。

備考

統合データベースの外部キーがリモートの Ultra Light データベースの外部キーに一致し、外部キー循環がない場合は、通常、この機能を使用する必要はありません。しかし、外部キー循環に含まれるテーブルがある場合は、循環に含まれるすべてのテーブルを TableOrder フィールドにリストする必要があります。統合データベース内の外部キーの関係が異なるテーブルがある場合も、これらのテーブルを TableOrder フィールドにリストします。

Copyright © 2007, iAnywhere Solutions, Inc.

指定しないテーブルは、リモート・データベースで定義した外部キーに基づいて適切に保存されます。

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。

構文

Visual Basic

Public Property **UploadOnly** As Boolean

C#

public bool **UploadOnly** {get;set;}

プロパティ値

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。

備考

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 クラス」 396 ページ
- 「ULSyncParms メンバ」 397 ページ
- 「DownloadOnly プロパティ」 399 ページ
- 「PingOnly プロパティ」 403 ページ
- 「ResumePartialDownload プロパティ」 404 ページ
- 「UploadOnly プロパティ」 408 ページ
- 「ULSQLCode 列挙」 366 ページ
- 「Synchronize メソッド」 119 ページ

UserName プロパティ

Mobile Link サーバが Mobile Link クライアントをユニークに識別するユーザ名です。

構文

Visual Basic

Public Property **UserName** As String

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public string **UserName** {get;set;}

プロパティ値

ユーザ名を指定する文字列。このパラメータ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。

備考

Mobile Link サーバでは、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。このユーザ名とパスワードは他のデータベース・ユーザ 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 サーバに対して識別し、認証するために使用されます。

参照

- 「ULSyncParms クラス」 396 ページ
- 「ULSyncParms メンバ」 397 ページ
- 「Password プロパティ」 402 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。

構文

Visual Basic

Public Property **Version** As String

C#

public string **Version** {get;set;}

プロパティ値

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。

備考

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。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Sub CopyFrom( _
  ByVal src As ULSyncParms _
)
```

C#

```
public void CopyFrom(
  ULSyncParms src
);
```

パラメータ

- **src** コピー元のオブジェクト。

参照

- 「ULSyncParms クラス」 396 ページ
- 「ULSyncParms メンバ」 397 ページ
- 「ULSyncParms クラス」 396 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Class **ULSyncProgressData**

C#

public class **ULSyncProgressData**

参照

- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressListener インタフェース」 421 ページ

ULSyncProgressData メンバ

パブリックの静的フィールド (共有)

メンバ名	説明
FLAG_IS_BLOCKING フィールド	Mobile Link サーバ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。

パブリック・プロパティ

メンバ名	説明
ErrorMessage プロパティ	同期中に発生したエラーを説明するエラー・メッセージを返します。
Flags プロパティ	現在の状態に関連する追加情報を示す、現在の同期フラグを返します。
ReceivedBytes プロパティ	現在までに受信したバイト数を返します。この情報は、すべてのステータスで更新されます。
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。この情報は、すべてのステータスで更新されます。
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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。
SQLCode プロパティ	同期の SQL コードを返します。
State プロパティ	現在の同期のステータスを返します。
SyncParms プロパティ	接続の SyncParms オブジェクトへの参照を返します。
SyncResult プロパティ	接続の SyncResult オブジェクトへの参照を返します。このオブジェクトは、ULSyncProgressState 列挙イベントと ULSyncProgressState 列挙イベントでのみ更新されます。
TableCount プロパティ	現在までに送信または受信したテーブルの数です (TableTotal の TableCount)。
TableIndex 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 ー DatabaseSchema.TableCount までの番号が付けられています)。
TableTotal プロパティ	同期中のテーブルの数を返します。

FLAG_IS_BLOCKING フィールド

Mobile Link サーバ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

構文

Visual Basic

Public Shared **FLAG_IS_BLOCKING** As Integer

C#

public const int **FLAG_IS_BLOCKING**;

ErrorMessage プロパティ

同期中に発生したエラーを説明するエラー・メッセージを返します。

構文

Visual Basic

Public Readonly Property **ErrorMessage** As String

C#

public string **ErrorMessage** {get;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

同期中に発生したエラーを説明する文字列。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 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。

構文

Visual Basic

Public Readonly Property **Flags** As Integer

C#

public int **Flags** {get;}

プロパティ値

フラグの組み合わせを保持する整数。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「FLAG_IS_BLOCKING フィールド」 412 ページ

ReceivedBytes プロパティ

現在までに受信したバイト数を返します。この情報は、すべてのステータスで更新されます。

構文

Visual Basic

Public Readonly Property **ReceivedBytes** As Long

C#

public long **ReceivedBytes** {get;}

プロパティ値

現在までに受信したバイト数。

参照

- 「ULSyncProgressData クラス」 411 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「ULSyncProgressState 列挙」 423 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。

構文

Visual Basic

Public Readonly Property **ReceivedDeletes** As Integer

C#

public int **ReceivedDeletes** {get;}

プロパティ値

現在までに受信した削除済みのローの数。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「ULSyncProgressState 列挙」 423 ページ

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。

構文

Visual Basic

Public Readonly Property **ReceivedInserts** As Integer

C#

public int **ReceivedInserts** {get;}

プロパティ値

現在までに受信した挿入済みのローの数。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「ULSyncProgressState 列挙」 423 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Readonly Property **ReceivedUpdates** As Integer

C#

public int **ReceivedUpdates** {get;}

プロパティ値

現在までに受信した更新済みのローの数。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「ULSyncProgressState 列挙」 423 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。この情報は、すべてのステータスで更新されます。

構文

Visual Basic

Public Readonly Property **SentBytes** As Long

C#

public long **SentBytes** {get;}

プロパティ値

現在までに送信されたバイト数。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「ULSyncProgressState 列挙」 423 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **SentDeletes** As Integer

C#

public int **SentDeletes** {get;}

プロパティ値

現在までに送信された削除済みのローの数。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「ULSyncProgressState 列挙」 423 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。

構文

Visual Basic

Public Readonly Property **SentInserts** As Integer

C#

public int **SentInserts** {get;}

プロパティ値

現在までに送信された挿入済みのローの数。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「ULSyncProgressState 列挙」 423 ページ

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **SentUpdates** As Integer

C#

public int **SentUpdates** {get;}

プロパティ値

現在までに送信された更新済みのローの数。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「ULSyncProgressState 列挙」 423 ページ

SQLCode プロパティ

同期の SQL コードを返します。

構文

Visual Basic

Public Readonly Property **SQLCode** As ULSQLCode

C#

public ULSQLCode **SQLCode** {get;}

プロパティ値

同期エラーの ULSQLCode 列挙値。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 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。

構文

Visual Basic

Public Readonly Property **State** As ULSyncProgressState

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public ULSyncProgressState **State** {get;}

プロパティ値

現在の同期のステータスを指定する ULSyncProgressState 列挙値の 1 つ。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 ページ

SyncParms プロパティ

接続の SyncParms オブジェクトへの参照を返します。

構文

Visual Basic

Public Readonly Property **SyncParms** As ULSyncParms

C#

public ULSyncParms **SyncParms** {get;}

プロパティ値

SyncParms プロパティオブジェクトへの参照。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「SyncParms プロパティ」 101 ページ

SyncResult プロパティ

接続の SyncResult オブジェクトへの参照を返します。このオブジェクトは、ULSyncProgressState 列挙イベントと ULSyncProgressState 列挙イベントでのみ更新されます。

構文

Visual Basic

Public Readonly Property **SyncResult** As ULSyncResult

 Copyright © 2007, iAnywhere Solutions, Inc.

C#

public ULSyncResult **SyncResult** {get;}

プロパティ値

SyncResult プロパティオブジェクトへの参照。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「ULSyncProgressState 列挙」 423 ページ

TableCount プロパティ

現在までに送信または受信したテーブルの数です (TableTotal の TableCount)。

構文

Visual Basic

Public Readonly Property **TableCount** As Integer

C#

public int **TableCount** {get;}

プロパティ値

送信または受信したテーブルの数。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「ULSyncProgressState 列挙」 423 ページ

TableIndex 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 ー DatabaseSchema.TableCount までの番号が付けられています)。

構文

Visual Basic

Public Readonly Property **TableIndex** As Integer

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public int **TableIndex** {get;}

プロパティ値

現在同期中のテーブルのインデックス。テーブルには、1 ーTableCount プロパティまでの番号が付けられています。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「ULSyncProgressState 列挙」 423 ページ

TableTotal プロパティ

同期中のテーブルの数を返します。

構文

Visual Basic

Public Readonly Property **TableTotal** As Integer

C#

public int **TableTotal** {get;}

プロパティ値

同期中のテーブルの数。

参照

- 「ULSyncProgressData クラス」 411 ページ
- 「ULSyncProgressData メンバ」 411 ページ
- 「ULSyncProgressState 列挙」 423 ページ
- 「ULSyncProgressState 列挙」 423 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULSyncProgressListener インタフェース

UL 拡張： 同期プログレス・イベントを受信するリスナ・インタフェースです。

構文

Visual Basic

Public Interface **ULSyncProgressListener**

C#

public interface **ULSyncProgressListener**

参照

- 「ULSyncProgressListener メンバ」 421 ページ
- 「Synchronize メソッド」 120 ページ

ULSyncProgressListener メンバ

パブリック・メソッド

メンバ名	説明
SyncProgressed メソッド	ユーザに進行状況を通知するために、同期処理中に起動されます。このメソッドは true を返して同期をキャンセルするか、false を返して続行する必要があります。

SyncProgressed メソッド

ユーザに進行状況を通知するために、同期処理中に起動されます。このメソッドは true を返して同期をキャンセルするか、false を返して続行する必要があります。

構文

Visual Basic

```
Public Function SyncProgressed( _
  ByVal data As ULSyncProgressData _
) As Boolean
```

C#

```
public bool SyncProgressed(
  ULSyncProgressData data
);
```

パラメータ

- **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。

Copyright © 2007, iAnywhere Solutions, Inc.

戻り値

このメソッドは true を返して同期をキャンセルするか、false を返して続行する必要があります。

備考

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

- 「ULSyncProgressListener インタフェース」 421 ページ
- 「ULSyncProgressListener メンバ」 421 ページ
- 「ULSyncProgressData クラス」 411 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULSyncProgressState 列挙

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。

構文

Visual Basic

Public Enum **ULSyncProgressState**
  Inherits Short

C#

public enum **ULSyncProgressState** :
  short

メンバ名

メンバ名	説明
STATE_CANCELLED	同期がキャンセルされました。
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。ReceivedBytes プロパティ、ReceivedInserts プロパティ、ReceivedUpdates プロパティ、ReceivedDeletes プロパティを参照してください。
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。
STATE_DISCONNECTING	同期ストリームが閉じられようとしています。
STATE_DONE	同期が正常に完了しました。接続の SyncResult プロパティオブジェクトが更新されています。
STATE_ERROR	同期は完了しましたが、エラーが発生しました。詳細については、SyncResult プロパティ、ErrorMessage プロパティ、SQLCode プロパティを確認してください。
STATE_FINISHING_UPLOAD	アップロードが完了しています。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。SentBytes プロパティ、SentInserts プロパティ、SentUpdates プロパティ、SentDeletes プロパティを参照してください。
STATE_LAST	同期の最後のステータスです。このステータスは、常にその他のステータスの後に入力されます。他の ULSyncProgressData フィールドには、有効なデータがないことがあります。
STATE_RECEIVING_DATA	現在のテーブルのデータを受信しています。ReceivedBytes プロパティ、ReceivedInserts プロパティ、ReceivedUpdates プロパティ、ReceivedDeletes プロパティが更新されています。
STATE_RECEIVING_TABLE	テーブルの受信中です。進行状況をモニタするには、TableIndex プロパティと TableCount プロパティを使用します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
STATE_RECEIVING_UPLOAD_ACK	アップロードが完了したという確認が受信されています。
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。
STATE_SENDING_DATA	現在のテーブルのデータが送信されています。SentBytes プロパティ、SentInserts プロパティ、SentUpdates プロパティ、SentDeletes プロパティが更新されています。
STATE_SENDING_DOWNLOAD_ACK	ダウンロードが完了したという確認が送信されています。
STATE_SENDING_HEADER	同期ストリームが開かれ、ヘッダが送信されようとしています。
STATE_SENDING_TABLE	テーブルの送信中です。進行状況をモニタするには、TableIndex プロパティと TableCount プロパティを使用します。
STATE_STARTING	同期処理はまだ行われていません。

参照

- 「ULSyncProgressData クラス」 411 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULSyncResult クラス

UL 拡張： 前回の同期のステータスを表します。

構文

Visual Basic

Public Class **ULSyncResult**

C#

public class **ULSyncResult**

備考

このクラスにはコンストラクタがありません。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

- 「ULSyncResult メンバ」 425 ページ
- 「SyncResult プロパティ」 102 ページ
- 「Synchronize メソッド」 119 ページ

ULSyncResult メンバ

パブリック・プロパティ

メンバ名	説明
AuthStatus プロパティ	前回行わ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。
StreamErrorContext プロパティ	ストリーム・エラーが発生したときに実行される基本的なネットワーク・オペレーションを返します。
StreamErrorID プロパティ	エラーをレポートするネットワーク・レイヤの ID 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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。

保護メソッド

メンバ名	説明
Finalize メソッド	ULSyncResult がガーベジ・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。

AuthStatus プロパティ

前回行われた同期の認証ステータス・コードを返します。

構文

Visual Basic

Public Readonly Property **AuthStatus** As ULAuthStatusCode

C#

public ULAuthStatusCode **AuthStatus** {get;}

プロパティ値

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

- 「ULSyncResult クラス」 425 ページ
- 「ULSyncResult メンバ」 425 ページ
- 「ULAuthStatusCode 列挙」 56 ページ

AuthValue プロパティ

カスタム・ユーザ認証同期スクリプトからの戻り値を返します。

構文

Visual Basic

Public Readonly Property **AuthValue** As Long

C#

public long **AuthValue** {get;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

カスタム・ユーザ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。

構文

Visual Basic

Public Readonly Property **IgnoredRows** As Boolean

C#

public bool **IgnoredRows** {get;}

プロパティ値

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

- 「ULSyncResult クラス」 425 ページ
- 「ULSyncResult メンバ」 425 ページ
- 「DownloadOnly プロパティ」 399 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。

構文

Visual Basic

Public Readonly Property **PartialDownloadRetained** As Boolean

C#

public bool **PartialDownloadRetained** {get;}

プロパティ値

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

- 「ULSyncResult クラス」 425 ページ
- 「ULSyncResult メンバ」 425 ページ
- 「KeepPartialDownload プロパティ」 400 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Readonly Property **StreamErrorCode** As ULStreamErrorCode

C#

public ULStreamErrorCode **StreamErrorCode** {get;}

プロパティ値

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。

参照

- 「ULSyncResult クラス」 425 ページ
- 「ULSyncResult メンバ」 425 ページ
- 「ULStreamErrorCode 列挙」 376 ページ
- 「ULStreamErrorCode 列挙」 376 ページ

StreamErrorContext プロパティ

ストリーム・エラーが発生したときに実行される基本的なネットワーク・オペレーションを返します。

構文

Visual Basic

Public Readonly Property **StreamErrorContext** As ULStreamErrorContext

C#

public ULStreamErrorContext **StreamErrorContext** {get;}

プロパティ値

ストリーム・エラーが発生したときに実行される基本的なネットワーク・オペレーションを示す ULStreamErrorContext 値の 1 つ。

参照

- 「ULSyncResult クラス」 425 ページ
- 「ULSyncResult メンバ」 425 ページ
- 「ULStreamErrorContext 列挙」 392 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

StreamErrorID プロパティ

エラーをレポートするネットワーク・レイヤの ID を返します。

構文

Visual Basic

Public Readonly Property **StreamErrorID** As ULStreamErrorID

C#

public ULStreamErrorID **StreamErrorID** {get;}

プロパティ値

エラーをレポートするネットワーク・レイヤの ID を示す ULStreamErrorID 値の 1 つ。

参照

- 「ULSyncResult クラス」 425 ページ
- 「ULSyncResult メンバ」 425 ページ
- 「ULStreamErrorID 列挙」 393 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。

構文

Visual Basic

Public Readonly Property **StreamErrorSystem** As Integer

C#

public int **StreamErrorSystem** {get;}

プロパティ値

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。

構文

Visual Basic

Public Readonly Property **Timestamp** As Date

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public DateTime **Timestamp** {get;}

プロパティ値

前回の同期のタイムスタンプを指定する System.DateTime。

参照

- 「ULSyncResult クラス」 425 ページ
- 「ULSyncResult メンバ」 425 ページ
- 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。

構文

Visual Basic

Public Readonly Property **UploadOK** As Boolean

C#

public bool **UploadOK** {get;}

プロパティ値

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。

Finalize メソッド

ULSyncResult がガーベジ・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。

構文

Visual Basic

Overrides Protected Sub **Finalize()**

C#

protected override void **Finalize();**

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Public Class ULTable
  Inherits ULResultSet
```

C#

```
public class ULTable :
  ULResultSet
```

備考

このクラスにはコンストラクタがありません。テーブルは、ULCommand の ULCommand.ExecuteTable() を使用して作成されます。

継承： ULResultSet

実装： System.Data.IDataReader、System.Data.IDataRecord、System.IDisposable

参照

- 「ULTable メンバ」 431 ページ
- 「ExecuteTable メソッド」 78 ページ
- 「ULCommand クラス」 57 ページ
- 「ULResultSet クラス」 325 ページ
- IDataReader
- IDataRecord
- IDisposable

ULTable メンバ

パブリック・プロパティ

メンバ名	説明
Depth プロパティ (ULDataReader から継承)	現在のローのネストの深さを返します。最も外側のテーブルの深さは 0 です。
FieldCount プロパティ (ULDataReader から継承)	このカーソル内のカラム数を返します。
IsBOF プロパティ (ULDataReader から継承)	**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 (ULDataReader から継承)	カーソルが現在開いているかどうかを確認します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
IsEOF プロパティ (ULDataReader から継承)	**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。
Item プロパティ (ULDataReader から継承)	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。C# では、このプロパティは ULDataReader クラスのインデクサです。
Item プロパティ (ULDataReader から継承)	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。C# では、このプロパティは ULDataReader クラスのインデクサです。
RecordsAffected プロパティ (ULDataReader から継承)	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。SELECT 文または CommandType.TableDirect テーブルの場合、この値は -1 です。
RowCount プロパティ (ULDataReader から継承)	**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。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。

パブリック・メソッド

メンバ名	説明
AppendBytes メソッド (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULDataReader から継承)	カーソルを閉じます。
Delete メソッド (ULResultSet から継承)	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 メソッド (ULDataReader から継承)	ULDataReader 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。
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。
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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	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。
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の一部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。
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。
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 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。
GetBoolean メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	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 メソッド (ULDataReader から継承)	**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。ULDbType.Binary 型、ULDbType.LongBinary 型、ULDbType.UniqueIdentifier 型のカラムの場合にのみ有効です。
GetChar メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 メソッド (ULDataReader から継承)	このメソッドは Ultra Light.NET ではサポートされていません。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
GetDataTypeName メソッド (ULDataReader から継承)	指定されたカラムのプロバイダのデータ型の名前を返します。
GetDateTime メソッド (ULDataReader から継承)	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。
GetDecimal メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	指定されたカラムの値を System.Double として返します。
GetFieldType メソッド (ULDataReader から継承)	指定されたカラムに最適な System.Type を返します。
GetFloat メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	指定されたカラムの値を Int32 として返します。
GetInt64 メソッド (ULDataReader から継承)	指定されたカラムの値を Int64 として返します。
GetName メソッド (ULDataReader から継承)	指定されたカラムの名前を返します。
GetOrdinal メソッド (ULDataReader から継承)	指定されたカラムのカラム ID を返します。
GetSchemaTable メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	指定されたカラムの値を UInt32 として返します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
GetUInt64 メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	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。 挿入を行う前に ULTable.InsertBegin を呼び出してください。
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 (ULDataReader から継承)	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。
LookupBackward メソッド	テーブルを最初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。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。
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 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。
MoveAfterLast メソッド (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**UL 拡張：** カーソルの最後のローに、カーソルを配置します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
MoveNext メソッド (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	カラムを NULL に設定します。
SetDecimal メソッド (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	指定されたカラムの値を、Int64 を使用して設定します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SetString メソッド (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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	テーブルの現在のローを更新する準備を行います。

保護メソッド

メンバ名	説明
Finalize メソッド	ULTable がガーベジ・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。

Schema プロパティ

テーブル・スキーマを保持します。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。

構文

Visual Basic

Public Readonly Property **Schema** As ULTableSchema

C#

public ULTableSchema **Schema** {get;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

テーブル・スキーマを示す ULTableSchema オブジェクト。

備考

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示します。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「ULTableSchema クラス」 453 ページ

DeleteAllRows メソッド

テーブルのすべてのローを削除します。

構文

Visual Basic

Public Sub **DeleteAllRows()**

C#

public void **DeleteAllRows();**

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「Truncate メソッド」 451 ページ
- 「StopSynchronizationDelete メソッド」 119 ページ

Finalize メソッド

ULTable がガーベジ・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。

 Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Overrides Protected Sub **Finalize()**

C#

protected override void **Finalize();**

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。

構文

Visual Basic

Public Sub **FindBegin()**

C#

public void **FindBegin();**

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「FindFirst メソッド」 439 ページ
- 「FindFirst メソッド」 440 ページ
- 「FindLast メソッド」 441 ページ
- 「FindLast メソッド」 442 ページ

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。

構文

Visual Basic

Overloads Public Function **FindFirst()** As Boolean

Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public bool FindFirst();
```

戻り値

成功した場合は true、失敗した場合は false。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「FindBegin メソッド」 439 ページ
- 「FindNext メソッド」 443 ページ
- 「FindPrevious メソッド」 445 ページ
- 「FindFirst メソッド」 440 ページ
- 「IsEOF プロパティ」 206 ページ
- 「FindBegin メソッド」 439 ページ

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の一部に完全に一致するローを検索します。

構文

Visual Basic

```
Overloads Public Function FindFirst( _
   ByVal numColumns As Short _
) As Boolean
```

C#

```
public bool FindFirst(
   short numColumns
);
```


Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **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。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「FindBegin メソッド」 439 ページ
- 「FindNext メソッド」 444 ページ
- 「FindPrevious メソッド」 445 ページ
- 「FindFirst メソッド」 439 ページ
- 「IsEOF プロパティ」 206 ページ
- 「FindBegin メソッド」 439 ページ

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。

構文

Visual Basic

```
Overloads Public Function FindLast() As Boolean
```

C#

```
public bool FindLast();
```

戻り値

成功した場合は true、失敗した場合は false。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「FindBegin メソッド」 439 ページ
- 「FindNext メソッド」 443 ページ
- 「FindPrevious メソッド」 445 ページ
- 「FindLast メソッド」 442 ページ
- 「IsBOF プロパティ」 206 ページ
- 「FindBegin メソッド」 439 ページ

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の一部に完全に一致するローを検索します。

構文

Visual Basic

```
Overloads Public Function FindLast( _
  ByVal numColumns As Short _
) As Boolean
```

C#

```
public bool FindLast(
  short numColumns
);
```

パラメータ

- **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。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「FindBegin メソッド」 439 ページ
- 「FindNext メソッド」 444 ページ
- 「FindPrevious メソッド」 445 ページ
- 「FindLast メソッド」 441 ページ
- 「IsBOF プロパティ」 206 ページ
- 「FindBegin メソッド」 439 ページ

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。

構文

Visual Basic

Overloads Public Function **FindNext()** As Boolean

C#

public bool **FindNext();**

戻り値

成功した場合は true、失敗した場合は false。

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。

例外

- ULException クラス - SQL エラーが発生しました。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「FindFirst メソッド」 439 ページ
- 「FindNext メソッド」 444 ページ
- 「IsEOF プロパティ」 206 ページ

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。

構文

Visual Basic

```
Overloads Public Function FindNext( _
  ByVal numColumns As Short _
) As Boolean
```

C#

```
public bool FindNext(
  short numColumns
);
```

パラメータ

- **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。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「FindFirst メソッド」 440 ページ
- 「FindNext メソッド」 443 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「FindFirst メソッド」 439 ページ
- 「IsEOF プロパティ」 206 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。

構文

Visual Basic

Overloads Public Function **FindPrevious()** As Boolean

C#

public bool **FindPrevious();**

戻り値

成功した場合は true、失敗した場合は false。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「FindLast メソッド」 441 ページ
- 「FindPrevious メソッド」 445 ページ
- 「IsBOF プロパティ」 206 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。

構文

Visual Basic

Overloads Public Function **FindPrevious(_**

Copyright © 2007, iAnywhere Solutions, Inc.

```
  ByVal numColumns As Short _
) As Boolean
```

C#

```
public bool FindPrevious(
  short numColumns
);
```

パラメータ

- **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。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「FindLast メソッド」 441 ページ
- 「FindLast メソッド」 442 ページ
- 「FindPrevious メソッド」 445 ページ
- 「IsBOF プロパティ」 206 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。

挿入を行う前に ULTable.InsertBegin を呼び出してください。

構文

Visual Basic

```
Public Sub Insert()
```

C#

```
public void Insert();
```


Copyright © 2007, iAnywhere Solutions, Inc.

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「InsertBegin メソッド」 447 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。

構文

Visual Basic

Public Sub **InsertBegin()**

C#

public void **InsertBegin();**

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「Insert メソッド」 446 ページ
- 「Insert メソッド」 446 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Overloads Public Function **LookupBackward()** As Boolean

C#

public bool **LookupBackward();**

戻り値

成功した場合は true、失敗した場合は false。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「LookupBegin メソッド」 449 ページ
- 「LookupBackward メソッド」 448 ページ
- 「IsBOF プロパティ」 206 ページ

LookupBackward メソッド

テーブルを最初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。

構文

Visual Basic

```
Overloads Public Function LookupBackward( _
   ByVal numColumns As Short _
) As Boolean
```

C#

```
public bool LookupBackward(
   short numColumns
);
```

 Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **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。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「LookupBegin メソッド」 449 ページ
- 「LookupBackward メソッド」 448 ページ
- 「IsBOF プロパティ」 206 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。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。

構文

Visual Basic

Public Sub **LookupBegin()**

C#

public void **LookupBegin();**

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「LookupForward メソッド」 450 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「LookupForward メソッド」 450 ページ
- 「LookupBackward メソッド」 447 ページ
- 「LookupBackward メソッド」 448 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。

構文

Visual Basic

Overloads Public Function **LookupForward()** As Boolean

C#

public bool **LookupForward();**

戻り値

成功した場合は true、失敗した場合は false。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「LookupBegin メソッド」 449 ページ
- 「LookupForward メソッド」 450 ページ
- 「IsEOF プロパティ」 206 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。

 Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Overloads Public Function LookupForward( _
  ByVal numColumns As Short _
) As Boolean
```

C#

```
public bool LookupForward(
  short numColumns
);
```

パラメータ

- **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。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「LookupBegin メソッド」 449 ページ
- 「LookupForward メソッド」 450 ページ
- 「IsEOF プロパティ」 206 ページ
- 「LookupBegin メソッド」 449 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Sub **Truncate()**

C#

public void **Truncate();**

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「DeleteAllRows メソッド」 438 ページ

UpdateBegin メソッド

テーブルの現在のローを更新する準備を行います。

構文

Visual Basic

Public Sub **UpdateBegin()**

C#

public void **UpdateBegin();**

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、ULResultSet.Update() が呼び出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。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTable クラス」 431 ページ
- 「ULTable メンバ」 431 ページ
- 「Update メソッド」 350 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

ULTableSchema クラス

UL 拡張： Ultra Light テーブルのスキーマを示します。

構文

Visual Basic

```
NotInheritable Public Class ULTableSchema
  Inherits ULCursorSchema
```

C#

```
public sealed class ULTableSchema :
  ULCursorSchema
```

備考

このクラスにはコンストラクタがありません。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。

継承： ULCursorSchema

参照

- 「ULTableSchema メンバ」 453 ページ
- 「ULTableSchema クラス」 453 ページ
- 「Schema プロパティ」 437 ページ
- 「ULCursorSchema クラス」 147 ページ

ULTableSchema メンバ

パブリック・プロパティ

メンバ名	説明
ColumnCount プロパティ (ULCursorSchema から継承)	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 (ULCursorSchema から継承)	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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。

パブリック・メソッド

メンバ名	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 (ULCursorSchema から継承)	指定されたカラムのカラム ID を返します。
GetColumnName メソッド (ULCursorSchema から継承)	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 (ULCursorSchema から継承)	カラムが数値カラム (SQL NUMERIC 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 (ULCursorSchema から継承)	カラムが数値カラム (SQL NUMERIC 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 (ULCursorSchema から継承)	カラムがサイズ指定されたカラム (SQL BINARY 型または CHAR 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 (ULCursorSchema から継承)	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 (ULCursorSchema から継承)	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 (ULCursorSchema から継承)	ULDataReader のカラムのスキーマが記述された System.Data.DataTable を返します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。
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。
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 入力可能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。

IndexCount プロパティ

テーブルのインデックス数を返します。

構文

Visual Basic

Public Readonly Property **IndexCount** As Integer

C#

public int **IndexCount** {get;}

プロパティ値

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。

備考

インデックス ID の範囲は、1 ー 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。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Readonly Property **IsNeverSynchronized** As Boolean

C#

public bool **IsNeverSynchronized** {get;}

プロパティ値

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。

備考

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、「未同期」テーブルと呼ばれることもあります。

例外

- ULException クラス - SQL エラーが発生しました。

Name プロパティ

テーブルの名前を返します。

構文

Visual Basic

Public Readonly Property **Name** As String

C#

public string **Name** {get;}

プロパティ値

文字列として返されるテーブル名。

例外

- ULException クラス - SQL エラーが発生しました。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **PrimaryKey** As ULIndexSchema

C#

public ULIndexSchema **PrimaryKey** {get;}

プロパティ値

テーブルのプライマリ・キーを示す 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

- 「ULTableSchema クラス」 453 ページ
- 「ULTableSchema メンバ」 453 ページ
- 「ULIndexSchema クラス」 273 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。

構文

Visual Basic

Public Readonly Property **UploadUnchangedRows** As Boolean

C#

public bool **UploadUnchangedRows** {get;}

プロパティ値

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。

備考

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。

例外

- ULException クラス - SQL エラーが発生しました。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Public Function GetColumnDefaultValue( _
   ByVal columnID As Integer _
) As String
```

C#

```
public string  GetColumnDefaultValue(
   int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

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

- 「ULTableSchema クラス」 453 ページ
- 「ULTableSchema メンバ」 453 ページ
- 「ColumnCount プロパティ」 148 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。

構文

Visual Basic

```
Public Function GetColumnPartitionSize( _
   ByVal columnID As Integer _
) As UInt64
```

C#

```
public ulong GetColumnPartitionSize(
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

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)。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTableSchema クラス」 453 ページ
- 「ULTableSchema メンバ」 453 ページ
- 「IsColumnGlobalAutoIncrement メソッド」 465 ページ
- 「ColumnCount プロパティ」 148 ページ
- 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。

構文

Visual Basic

```
Public Function GetIndex( _
  ByVal name As String _
) As ULIndexSchema
```

C#

```
public ULIndexSchema GetIndex(
  string  name
);
```

パラメータ

- **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示す ULIndexSchema オブジェクト。

Copyright © 2007, iAnywhere Solutions, Inc.

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTableSchema クラス」 453 ページ
- 「ULTableSchema メンバ」 453 ページ
- 「ULIndexSchema クラス」 273 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。

構文

Visual Basic

```
Public Function GetIndexName( _
  ByVal indexID As Integer _
) As String
```

C#

```
public string  GetIndexName(
  int indexID
);
```

パラメータ

- **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。

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。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTableSchema クラス」 453 ページ
- 「ULTableSchema メンバ」 453 ページ
- 「IndexCount プロパティ」 455 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

GetOptimalIndex メソッド

指定し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。

構文

Visual Basic

```
Public Function GetOptimalIndex( _
  ByVal columnID As Integer _
) As ULIndexSchema
```

C#

```
public ULIndexSchema GetOptimalIndex(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

戻り値

指定されたカラムの最適なインデックスを示す ULIndexSchema オブジェクト。

備考

インデックス内の最初のカラムが指定されますが、インデックスには複数のカラムがある場合があります。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTableSchema クラス」 453 ページ
- 「ULTableSchema メンバ」 453 ページ
- 「ColumnCount プロパティ」 148 ページ
- 「ULIndexSchema クラス」 273 ページ

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。

構文

Visual Basic

```
Public Function GetPublicationPredicate( _
  ByVal pubName As String _
) As String
```

Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public string  GetPublicationPredicate(
   string  pubName
);
```

パラメータ

- **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。

構文

Visual Basic

```
Public Function IsColumnAutoIncrement( _
   ByVal columnID As Integer _
) As Boolean
```

C#

```
public bool IsColumnAutoIncrement(
   int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

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

- 「ULTableSchema クラス」 453 ページ
- 「ULTableSchema メンバ」 453 ページ
- 「ColumnCount プロパティ」 148 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Public Function IsColumnCurrentDate( _
  ByVal columnID As Integer _
) As Boolean
```

C#

```
public bool IsColumnCurrentDate(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

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

- 「ULTableSchema クラス」 453 ページ
- 「ULTableSchema メンバ」 453 ページ
- 「ColumnCount プロパティ」 148 ページ
- 「ULDbType 列挙」 245 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。

構文

Visual Basic

```
Public Function IsColumnCurrentTime( _
  ByVal columnID As Integer _
) As Boolean
```

C#

Copyright © 2007, iAnywhere Solutions, Inc.

```
public bool IsColumnCurrentTime(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

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

- 「ULTableSchema クラス」 453 ページ
- 「ULTableSchema メンバ」 453 ページ
- 「ColumnCount プロパティ」 148 ページ
- 「ULDbType 列挙」 245 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。

構文

Visual Basic

```
Public Function IsColumnCurrentTimestamp( _
  ByVal columnID As Integer _
) As Boolean
```

C#

```
public bool IsColumnCurrentTimestamp(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTableSchema クラス」 453 ページ
- 「ULTableSchema メンバ」 453 ページ
- 「ColumnCount プロパティ」 148 ページ
- 「ULDbType 列挙」 245 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。

構文

Visual Basic

```
Public Function IsColumnGlobalAutoIncrement( _
  ByVal columnID As Integer _
) As Boolean
```

C#

```
public bool IsColumnGlobalAutoIncrement(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

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 クラス」 453 ページ
- 「ULTableSchema メンバ」 453 ページ
- 「GetColumnPartitionSize メソッド」 458 ページ
- 「DatabaseID プロパティ」 98 ページ
- 「ColumnCount プロパティ」 148 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Public Function IsColumnNewUUID( _
  ByVal columnID As Integer _
) As Boolean
```

C#

```
public bool IsColumnNewUUID(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

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

- 「ULTableSchema クラス」 453 ページ
- 「ULTableSchema メンバ」 453 ページ
- 「ColumnCount プロパティ」 148 ページ
- Guid

IsColumnNullable メソッド

指定されたカラムが null 入力可能であるかどうかをチェックします。

構文

Visual Basic

```
Public Function IsColumnNullable( _
  ByVal columnID As Integer _
) As Boolean
```

C#

```
public bool IsColumnNullable(
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

戻り値

カラムが null 入力可能な場合は true、null 入力不可能な場合は false。

例外

- ULException クラス - SQL エラーが発生しました。

参照

- 「ULTableSchema クラス」 453 ページ
- 「ULTableSchema メンバ」 453 ページ
- 「ColumnCount プロパティ」 148 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。

構文

Visual Basic

```
Public Function IsInPublication( _
  ByVal pubName As String _
) As Boolean
```

C#

```
public bool IsInPublication(
  string  pubName
);
```

パラメータ

- **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。

Copyright © 2007, iAnywhere Solutions, Inc.

ULTransaction クラス

SQL トランザクションを示します。

構文

Visual Basic

NotInheritable Public Class **ULTransaction**

C#

public sealed class **ULTransaction**

備考

ULTransaction にはコンストラクタがありません。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。

実装： System.Data.IDbTransaction、System.IDisposable

参照

- 「ULTransaction メンバ」 468 ページ
- 「BeginTransaction メソッド」 103 ページ
- 「Transaction プロパティ」 68 ページ
- IDbTransaction
- IDisposable

ULTransaction メンバ

パブリック・プロパティ

メンバ名	説明
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。

パブリック・メソッド

メンバ名	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 メソッド	ULTransaction によって使用されたリソースを解放し、コミットされていないコマンドをロールバックします。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
Rollback メソッド	トランザクションの未処理の変更をデータベースに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。

構文

Visual Basic

Public Readonly Property **Connection** As ULConnection

C#

public ULConnection **Connection** {get;}

プロパティ値

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)。

備考

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

参照

- 「ULTransaction クラス」 468 ページ
- 「ULTransaction メンバ」 468 ページ
- 「BeginTransaction メソッド」 103 ページ
- 「ULConnection クラス」 90 ページ
- IDbTransaction.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。

構文

Visual Basic

NotOverridable Public Readonly Property **IsolationLevel** As IsolationLevel _
  Implements IDbTransaction.IsolationLevel

C#

public IsolationLevel **IsolationLevel** {get;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

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。

実装

IDbTransaction.IsolationLevel

参照

- 「ULTransaction クラス」 468 ページ
- 「ULTransaction メンバ」 468 ページ
- 「BeginTransaction メソッド」 103 ページ
- IsolationLevel
- IsolationLevel.ReadUncommitted

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。

構文

Visual Basic

```
NotOverridable Public Sub Commit() _
  Implements IDbTransaction.Commit
```

C#

```
public 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。

実装

IDbTransaction.Commit

参照

- 「ULTransaction クラス」 468 ページ
- 「ULTransaction メンバ」 468 ページ
- 「Rollback メソッド」 471 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

Dispose メソッド

ULTransaction によって使用されたリソースを解放し、コミットされていないコマンドをロールバックします。

構文

Visual Basic

```
NotOverridable Public Sub Dispose() _
  Implements IDisposable.Dispose
```

C#

```
public void Dispose();
```

実装

IDisposable.Dispose

Rollback メソッド

トランザクションの未処理の変更をデータベースにロールバックします。

構文

Visual Basic

```
NotOverridable Public Sub Rollback() _
  Implements IDbTransaction.Rollback
```

C#

```
public 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。

実装

IDbTransaction.Rollback

参照

- 「ULTransaction クラス」 468 ページ
- 「ULTransaction メンバ」 468 ページ
- 「Commit メソッド」 470 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

第 6 章

Ultra Light .NET 2.0 API リファレンス

目次

ULActiveSyncListener インタフェース 475
ULAuthStatusCode 列挙 479
ULBulkCopy クラス 480
ULBulkCopyColumnMapping クラス 492
ULBulkCopyColumnMappingCollection クラス 499
ULBulkCopyOptions 列挙 508
ULCommand クラス 509
ULCommandBuilder クラス 545
ULConnection クラス 555
ULConnectionParms クラス 592
ULConnectionParms.UnusedEventHandler 委任 605
ULConnectionStringBuilder クラス 606
ULCreateParms クラス 624
ULCursorSchema クラス 636
ULDataAdapter クラス 645
ULDatabaseManager クラス 657
ULDatabaseSchema クラス 665
ULDataReader クラス 676
ULDateOrder enumeration 714
ULDbType 列挙 715
ULException クラス 719
ULFactory クラス 723
ULFileTransfer クラス 729
ULFileTransferProgressData クラス 745
ULFileTransferProgressListener インタフェース 749
ULIndexSchema クラス 751
ULInfoMessageEventArgs クラス 760
ULInfoMessageEventHandler 委任 763
ULMetaDataCollectionNames クラス 764

Copyright © 2007, iAnywhere Solutions, Inc.

ULParameter クラス **773**
ULParameterCollection クラス **790**
ULPublicationSchema クラス **810**
ULResultSet クラス **814**
ULResultSetSchema クラス **839**
ULRowsCopiedEventArgs クラス **842**
ULRowsCopiedEventHandler 委任 **845**
ULRowUpdatedEventArgs クラス **846**
ULRowUpdatedEventHandler 委任 **850**
ULRowUpdatingEventArgs クラス **851**
ULRowUpdatingEventHandler 委任 **854**
ULRuntimeType 列挙 **855**
ULServerSyncListener インタフェース **856**
ULSQLCode 列挙 **859**
ULStreamErrorCode 列挙 **871**
ULStreamErrorContext 列挙 **892**
ULStreamErrorID 列挙 **893**
ULStreamType 列挙 **895**
ULSyncParms クラス **896**
ULSyncProgressData クラス **912**
ULSyncProgressListener インタフェース **923**
ULSyncProgressState 列挙 **925**
ULSyncResult クラス **927**
ULTable クラス **933**
ULTableSchema クラス **954**
ULTransaction クラス **969**

Copyright © 2007, iAnywhere Solutions, Inc.

ULActiveSyncListener インタフェース

UL 拡張： ActiveSync イベントを受信するリスナ・インタフェースです。

構文

Visual Basic

Public Interface **ULActiveSyncListener**

C#

public interface **ULActiveSyncListener**

参照

- 「ULActiveSyncListener メンバ」 475 ページ

ULActiveSyncListener メンバ

パブリック・メソッド

メンバ名	説明
ActiveSyncInvoked メソッド	ActiveSync 用の Mobile Link プロバイダ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。

参照

- 「ULActiveSyncListener インタフェース」 475 ページ

ActiveSyncInvoked メソッド

ActiveSync 用の Mobile Link プロバイダ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。

構文

Visual Basic

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

C#

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

Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **launchedByProvider** ActiveSync 同期を実行するために、Mobile Link プロバイダ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・インタフェー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() を呼び出して、ActiveSync 用の Mobile Link プロバイダに対して通知する必要があります。

例

次のコード・フラグメントは、ActiveSync 要求を受信して、UI スレッドで同期を実行する方法を示しています。

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

Public Class MainWindow
  Inherits System.Windows.Forms.Form
  Implements ULActiveSyncListener
  Private conn As ULConnection

  Public Sub New(ByVal args() As String)

    MyBase.New()

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

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

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

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


Copyright © 2007, iAnywhere Solutions, Inc.

```
  Public Sub ActiveSyncAction( _
      ByVal sender As Object, ByVal e As EventArgs _
    )
    ' Do active sync
    conn.Synchronize()
    ULConnection.DatabaseManager.SignalSyncIsComplete()
  End Sub
End Class


// 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
    // 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();
  }
}
```

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULActiveSyncListener インタフェース」 475 ページ
- 「ULActiveSyncListener メンバ」 475 ページ
- 「SignalSyncIsComplete メソッド」 664 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULAuthStatusCode 列挙

UL 拡張： Mobile Link ユーザ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。

構文

Visual Basic

Public Enum **ULAuthStatusCode**

C#

public enum **ULAuthStatusCode**

メンバ

メンバ名	説明	値
EXPIRED	ユーザ 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
IN_USE	ユーザ 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
INVALID	ユーザ 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
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 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 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

参照

- 「AuthStatus プロパティ」 928 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。このクラスは継承できません。

構文

Visual Basic

Public NotInheritable Class **ULBulkCopy**
  Implements IDisposable

C#

public sealed class **ULBulkCopy** : IDisposable

備考

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

参照

- 「ULBulkCopy メンバ」 480 ページ

ULBulkCopy メンバ

パブリック・コンストラクタ

メンバ名	説明
ULBulkCopy コンストラクタ	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。

パブリック・プロパティ

メンバ名	説明
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。各バッチが終了すると、バッチ内のローがサーバに送信されます。
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。
ColumnMappings プロパティ	ULBulkCopyColumnMapping 項目のコレクションを返します。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。
DestinationTableName プロパティ	サーバ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。

Copyright © 2007, iAnywhere Solutions, Inc.

パブリック・メソッド

メンバ名	説明
Close メソッド	ULBulkCopy インスタンスを閉じます。
Dispose メソッド	ULBulkCopy インスタンスを破棄します。
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。

パブリック・イベント

メンバ名	説明
ULRowsCopied イベント	NotifyAfter 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。

参照

- 「ULBulkCopy クラス」 480 ページ

ULBulkCopy コンストラクタ

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。

参照

- 「ULConnection クラス」 555 ページ

ULBulkCopy(ULConnection) コンストラクタ

構文

Visual Basic

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

C#

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

パラメータ

- **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。接続が開いていない場合は、WriteToServer に例外がスローされます。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

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

参照

- 「ULBulkCopy クラス」 480 ページ
- 「ULBulkCopy メンバ」 480 ページ
- 「ULBulkCopy コンストラクタ」 481 ページ
- 「ULConnection クラス」 555 ページ

ULBulkCopy(String) コンストラクタ

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。

構文

Visual Basic

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

C#

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

パラメータ

- **connectionString**　ULBulkCopy インスタンス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。

  パラメータのリストについては、「ConnectionString プロパティ」 560 ページを参照してください。

備考

この構文は、connectionString を使用して WriteToServer の実行中に接続を開きます。WriteToServer 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。

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

実装： System.IDisposable

参照

- 「ULBulkCopy クラス」 480 ページ
- 「ULBulkCopy メンバ」 480 ページ
- 「ULBulkCopy コンストラクタ」 481 ページ
- 「ULConnectionParms クラス」 592 ページ
- IDisposable

Copyright © 2007, iAnywhere Solutions, Inc.

ULBulkCopy(String, ULBulkCopyOptions) コンストラクタ

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。

構文

Visual Basic

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

C#

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

パラメータ

- **connectionString**　ULBulkCopy インスタンス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。

  パラメータのリストについては、「ConnectionString プロパティ」 560 ページを参照してください。

- **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 列挙の値の組み合わせ。

備考

この構文は、connectionString を使用して WriteToServer の実行中に接続を開きます。WriteToServer が終了すると、接続が閉じます。

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

参照

- 「ULBulkCopy クラス」 480 ページ
- 「ULBulkCopy メンバ」 480 ページ
- 「ULBulkCopy コンストラクタ」 481 ページ
- 「ULBulkCopyOptions 列挙」 508 ページ

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

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

構文

Visual Basic

```
Public Sub New( _
```

```
  ByVal connection As ULConnection, _
  ByVal copyOptions As ULBulkCopyOptions, _
  ByVal externalTransaction As ULTransaction _
)
```

C#

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

パラメータ

- **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。接続が開いていない場合は、WriteToServer 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 列挙の値の組み合わせ。
- **externalTransaction** バルク・コピーが発生する、既存の ULTransaction インスタンス。externalTransaction が null 参照 (Visual Basic の Nothing) でないと、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

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

参照

- 「ULBulkCopy クラス」 480 ページ
- 「ULBulkCopy メンバ」 480 ページ
- 「ULBulkCopy コンストラクタ」 481 ページ
- 「ULConnection クラス」 555 ページ
- 「ULTransaction クラス」 969 ページ

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。各バッチが終了すると、バッチ内のローがサーバに送信されます。

構文

Visual Basic

```
Public Property BatchSize As Integer
```

C#

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

 Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

各バッチの中のロー数。デフォルトは 0 です。

備考

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。

参照

- 「ULBulkCopy クラス」 480 ページ
- 「ULBulkCopy メンバ」 480 ページ

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。

構文

Visual Basic

Public Property **BulkCopyTimeout** As Integer

C#

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

プロパティ値

デフォルト値は 30 秒です。

備考

値が 0 の場合、制限はありません。この場合、待機時間が無限になる可能性があるため、値は 0 にしないでください。

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 が発生します。

0 未満の値を設定すると、エラーになります。

参照

- 「ULBulkCopy クラス」 480 ページ
- 「ULBulkCopy メンバ」 480 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ColumnMappings プロパティ

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

構文

Visual Basic

Public Readonly Property **ColumnMappings** As ULBulkCopyColumnMappingCollection

C#

public ULBulkCopyColumnMappingCollection **ColumnMappings** { get;}

プロパティ値

デフォルトでは、空のコレクションです。

備考

WriteToServer の実行中は、プロパティを変更できません。

WriteToServer の実行時に ColumnMappings が空の場合、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プされ、2 番目は 2 番目にマップされます。以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 入力可能なカラムである場合に行われます。

参照

- 「ULBulkCopy クラス」 480 ページ
- 「ULBulkCopy メンバ」 480 ページ
- 「ULBulkCopyColumnMapping クラス」 492 ページ

DestinationTableName プロパティ

サーバ上の送信先テーブルの名前を取得または設定します。

構文

Visual Basic

Public Property **DestinationTableName** As String

C#

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

プロパティ値

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

備考

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

 Copyright © 2007, iAnywhere Solutions, Inc.

WriteToServer への呼び出しの前に値が設定されていない場合、InvalidOperationException が発生します。

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。

参照

- 「ULBulkCopy クラス」 480 ページ
- 「ULBulkCopy メンバ」 480 ページ

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。

構文

Visual Basic

Public Property **NotifyAfter** As Integer

C#

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

プロパティ値

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。

備考

WriteToServer の実行時に加えられる NotifyAfter への変更は、次の通知まで反映されません。

0 未満の値を設定すると、エラーになります。

NotifyAfter と BulkCopyTimeOut の値は相互に排他的なため、データベースにローが送信されなかったり、コミットされない場合であっても、イベントは起動します。

参照

- 「ULBulkCopy クラス」 480 ページ
- 「ULBulkCopy メンバ」 480 ページ
- 「BulkCopyTimeOut プロパティ」 485 ページ

Close メソッド

ULBulkCopy インスタンスを閉じます。

構文

Visual Basic

Public Sub **Close()**

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public void **Close();**

参照

- 「ULBulkCopy クラス」 480 ページ
- 「ULBulkCopy メンバ」 480 ページ

Dispose メソッド

ULBulkCopy インスタンスを破棄します。

構文

Visual Basic

NotOverridable Public Sub **Dispose()**

C#

public void **Dispose();**

参照

- 「ULBulkCopy クラス」 480 ページ
- 「ULBulkCopy メンバ」 480 ページ

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。

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。

構文

Visual Basic

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

C#

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


Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **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 クラス」 480 ページ
- 「ULBulkCopy メンバ」 480 ページ
- 「WriteToServer メソッド」 488 ページ
- DataRow
- 「DestinationTableName プロパティ」 486 ページ

WriteToServer(DataTable) メソッド

指定された System.Data.DataTable のすべてのローを、ULBulkCopy オブジェクトの DestinationTableName で指定される送信先テーブルにコピーします。

構文

Visual Basic

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

C#

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

パラメータ

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

参照

- 「ULBulkCopy クラス」 480 ページ
- 「ULBulkCopy メンバ」 480 ページ
- 「WriteToServer メソッド」 488 ページ
- 「DestinationTableName プロパティ」 486 ページ
- DataTable

WriteToServer(IDataReader) メソッド

指定された System.Data.IDataReader のすべてのローを、ULBulkCopy オブジェクトの DestinationTableName で指定される送信先テーブルにコピーします。

構文

Visual Basic

```
Public Sub WriteToServer( _
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
  ByVal reader As IDataReader _
)
```

C#

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

パラメータ

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

参照

- 「ULBulkCopy クラス」 480 ページ
- 「ULBulkCopy メンバ」 480 ページ
- 「WriteToServer メソッド」 488 ページ
- IDataReader
- 「DestinationTableName プロパティ」 486 ページ

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 オブジェクトの DestinationTableName で指定される送信先テーブルにコピーします。

構文

Visual Basic

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

C#

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

パラメータ

- **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。

備考

rowState 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 クラス」 480 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULBulkCopy メンバ」 480 ページ
- 「WriteToServer メソッド」 488 ページ
- 「DestinationTableName プロパティ」 486 ページ
- DataTable
- DataRowState

ULRowsCopied イベント

NotifyAfter 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。

構文

Visual Basic

Public Event **ULRowsCopied** As ULRowsCopiedEventHandler

C#

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 クラス」 480 ページ
- 「ULBulkCopy メンバ」 480 ページ
- 「NotifyAfter プロパティ」 487 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。このクラスは継承できません。

構文

Visual Basic

Public NotInheritable Class **ULBulkCopyColumnMapping**

C#

public sealed class **ULBulkCopyColumnMapping**

備考

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

参照

- 「ULBulkCopyColumnMapping メンバ」 492 ページ
- 「ULBulkCopy クラス」 480 ページ

ULBulkCopyColumnMapping メンバ

パブリック・コンストラクタ

メンバ名	説明
ULBulkCopyColumnMapping コンストラクタ	「ULBulkCopyColumnMapping クラス」 492 ページの新しいインスタンスを初期化します。

パブリック・プロパティ

メンバ名	説明
DestinationColumn プロパティ	マップ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	マップされる送信先データベース・テーブルのカラムの序数を指定します。
SourceColumn プロパティ	データ・ソースにマップ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・カラムの並び順を指定します。

参照

- 「ULBulkCopyColumnMapping クラス」 492 ページ
- 「ULBulkCopy クラス」 480 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULBulkCopyColumnMapping コンストラクタ

「ULBulkCopyColumnMapping クラス」 492 ページの新しいインスタンスを初期化します。

ULBulkCopyColumnMapping() コンストラクタ

新しいカラム・マッピングを作成します。

構文

Visual Basic

```
Public Sub New()
```

C#

```
public  ULBulkCopyColumnMapping();
```

備考

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

参照

- 「ULBulkCopyColumnMapping クラス」 492 ページ
- 「ULBulkCopyColumnMapping メンバ」 492 ページ
- 「ULBulkCopyColumnMapping コンストラクタ」 493 ページ

ULBulkCopyColumnMapping(Int32, Int32) コンストラクタ

カラムの序数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。

構文

Visual Basic

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

C#

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

パラメータ

- **sourceColumnOrdinal**　データ・ソース内のソース・カラムの並び順。データ・ソースの最初のカラムの序数位置は 0 です。

Copyright © 2007, iAnywhere Solutions, Inc.

- **destinationColumnOrdinal**　送信先テーブル内の送信先カラムの並び順。テーブルの最初のカラムの序数位置は 0 です。

備考

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

参照

- 「ULBulkCopyColumnMapping クラス」 492 ページ
- 「ULBulkCopyColumnMapping メンバ」 492 ページ
- 「ULBulkCopyColumnMapping コンストラクタ」 493 ページ

ULBulkCopyColumnMapping(Int32, String) コンストラクタ

ソース・カラムを参照するカラムの序数と、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。

構文

Visual Basic

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

C#

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

パラメータ

- **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

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

参照

- 「ULBulkCopyColumnMapping クラス」 492 ページ
- 「ULBulkCopyColumnMapping メンバ」 492 ページ
- 「ULBulkCopyColumnMapping コンストラクタ」 493 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULBulkCopyColumnMapping(String, Int32) コンストラクタ

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序数を使用して、新しいカラム・マッピングを作成します。

構文

Visual Basic

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

C#

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

パラメータ

- **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

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

参照

- 「ULBulkCopyColumnMapping クラス」 492 ページ
- 「ULBulkCopyColumnMapping メンバ」 492 ページ
- 「ULBulkCopyColumnMapping コンストラクタ」 493 ページ

ULBulkCopyColumnMapping(String, String) コンストラクタ

カラムの序数を使用してソース・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。

構文

Visual Basic

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

C#

```
public  ULBulkCopyColumnMapping(
```

```
  string  sourceColumn,
  string  destinationColumn
);
```

パラメータ

- **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

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

参照

- 「ULBulkCopyColumnMapping クラス」 492 ページ
- 「ULBulkCopyColumnMapping メンバ」 492 ページ
- 「ULBulkCopyColumnMapping コンストラクタ」 493 ページ

DestinationColumn プロパティ

マップ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。

構文

Visual Basic

```
Public Property DestinationColumn As String
```

C#

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

プロパティ値

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

備考

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 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

- 「ULBulkCopyColumnMapping クラス」 492 ページ
- 「ULBulkCopyColumnMapping メンバ」 492 ページ
- 「DestinationOrdinal プロパティ」 497 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

- 「DestinationOrdinal プロパティ」 497 ページ

DestinationOrdinal プロパティ

マップされる送信先データベース・テーブルのカラムの序数を指定します。

構文

Visual Basic

Public Property **DestinationOrdinal** As Integer

C#

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

プロパティ値

マップされるカラムの送信先テーブルにおける序数を指定する整数。プロパティが設定されていない場合は -1。

備考

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

- 「ULBulkCopyColumnMapping クラス」 492 ページ
- 「ULBulkCopyColumnMapping メンバ」 492 ページ
- 「DestinationColumn プロパティ」 496 ページ
- 「DestinationColumn プロパティ」 496 ページ

SourceColumn プロパティ

データ・ソースにマップされるカラムの名前を指定します。

構文

Visual Basic

Public Property **SourceColumn** As String

C#

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

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

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)。

備考

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 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

- 「ULBulkCopyColumnMapping クラス」 492 ページ
- 「ULBulkCopyColumnMapping メンバ」 492 ページ
- 「SourceOrdinal プロパティ」 498 ページ
- 「SourceOrdinal プロパティ」 498 ページ

SourceOrdinal プロパティ

データ・ソース内のソース・カラムの並び順を指定します。

構文

Visual Basic

Public Property **SourceOrdinal** As Integer

C#

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

プロパティ値

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

備考

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

- 「ULBulkCopyColumnMapping クラス」 492 ページ
- 「ULBulkCopyColumnMapping メンバ」 492 ページ
- 「SourceColumn プロパティ」 497 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

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。このクラスは継承できません。

構文

Visual Basic

Public NotInheritable Class **ULBulkCopyColumnMappingCollection**
  Inherits CollectionBase

C#

public sealed class **ULBulkCopyColumnMappingCollection** : CollectionBase

備考

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

参照

- 「ULBulkCopyColumnMappingCollection メンバ」 499 ページ
- 「ULBulkCopyColumnMapping クラス」 492 ページ

ULBulkCopyColumnMappingCollection メンバ

パブリック・プロパティ

メンバ名	説明
Capacity (CollectionBase から継承)	
Count (CollectionBase から継承)	
Item 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。

パブリック・メソッド

メンバ名	説明
Add メソッド	指定された ULBulkCopyColumnMapping をコレクションに追加します。
Clear (CollectionBase から継承)	

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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 の要素を、特定のインデックスを先頭に、ULBulkCopyColumnMapping 項目の配列にコピーします。
GetEnumerator (CollectionBase から継承)	
IndexOf メソッド	コレクション内で指定された ULBulkCopyColumnMapping のインデックスを返します。
Remove メソッド	指定された ULBulkCopyColumnMapping 要素を ULBulkCopyColumnMappingCollection 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。

参照

- 「ULBulkCopyColumnMappingCollection クラス」 499 ページ
- 「ULBulkCopyColumnMapping クラス」 492 ページ

Item 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。

構文

Visual Basic

```
Public Readonly Property Item ( _
   ByVal index As Integer _
) As ULBulkCopyColumnMapping
```

C#

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

パラメータ

- **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。

プロパティ値

ULBulkCopyColumnMapping オブジェクトが返されます。

参照

- 「ULBulkCopyColumnMappingCollection クラス」 499 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULBulkCopyColumnMappingCollection メンバ」 499 ページ
- 「ULBulkCopyColumnMapping クラス」 492 ページ

Add メソッド

指定された ULBulkCopyColumnMapping 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 をコレクションに追加します。

構文

Visual Basic

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

C#

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

パラメータ

- **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

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

参照

- 「ULBulkCopyColumnMappingCollection クラス」 499 ページ
- 「ULBulkCopyColumnMappingCollection メンバ」 499 ページ
- 「Add メソッド」 501 ページ
- 「ULBulkCopyColumnMapping クラス」 492 ページ

Add(Int32, Int32) 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序数を使用して、新しい ULBulkCopyColumnMapping インスタンスを作成し、マッピングをコレクションに追加します。

構文

Visual Basic

Copyright © 2007, iAnywhere Solutions, Inc.

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

C#

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

パラメータ

- **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

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

参照

- 「ULBulkCopyColumnMappingCollection クラス」 499 ページ
- 「ULBulkCopyColumnMappingCollection メンバ」 499 ページ
- 「Add メソッド」 501 ページ
- 「ULBulkCopyColumnMapping クラス」 492 ページ

Add(Int32, String) メソッド

カラム序数を使用してソース・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 を作成し、このマッピングをコレクションに追加します。

構文

Visual Basic

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

C#

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

 Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **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

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

参照

- 「ULBulkCopyColumnMappingCollection クラス」 499 ページ
- 「ULBulkCopyColumnMappingCollection メンバ」 499 ページ
- 「Add メソッド」 501 ページ
- 「ULBulkCopyColumnMapping クラス」 492 ページ

Add(String, Int32) メソッド

カラム名を使用してソース・カラムを参照し、カラム序数を使用して送信先カラムを参照する、新しい ULBulkCopyColumnMapping を作成し、このマッピングをコレクションに追加します。

カラムの序数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。

構文

Visual Basic

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

C#

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

パラメータ

- **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

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

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULBulkCopyColumnMappingCollection クラス」 499 ページ
- 「ULBulkCopyColumnMappingCollection メンバ」 499 ページ
- 「Add メソッド」 501 ページ
- 「ULBulkCopyColumnMapping クラス」 492 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 を作成し、マッピングをコレクションに追加します。

構文

Visual Basic

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

C#

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

パラメータ

- **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

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

参照

- 「ULBulkCopyColumnMappingCollection クラス」 499 ページ
- 「ULBulkCopyColumnMappingCollection メンバ」 499 ページ
- 「Add メソッド」 501 ページ
- 「ULBulkCopyColumnMapping クラス」 492 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

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

C#

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

パラメータ

- **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

- 「ULBulkCopyColumnMappingCollection クラス」 499 ページ
- 「ULBulkCopyColumnMappingCollection メンバ」 499 ページ
- 「ULBulkCopyColumnMapping クラス」 492 ページ

CopyTo メソッド

ULBulkCopyColumnMappingCollection の要素を、特定のインデックスを先頭に、ULBulkCopyColumnMapping 項目の配列にコピーします。

構文

Visual Basic

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

C#

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

パラメータ

- **array**　ULBulkCopyColumnMappingCollection 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。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULBulkCopyColumnMappingCollection クラス」 499 ページ
- 「ULBulkCopyColumnMappingCollection メンバ」 499 ページ
- 「ULBulkCopyColumnMapping クラス」 492 ページ

IndexOf メソッド

コレクション内で指定された ULBulkCopyColumnMapping のインデックスを返します。

構文

Visual Basic

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

C#

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

パラメータ

- **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

- 「ULBulkCopyColumnMappingCollection クラス」 499 ページ
- 「ULBulkCopyColumnMappingCollection メンバ」 499 ページ
- 「ULBulkCopyColumnMapping クラス」 492 ページ

Remove メソッド

指定された ULBulkCopyColumnMapping 要素を ULBulkCopyColumnMappingCollection から削除します。

構文

Visual Basic

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


Copyright © 2007, iAnywhere Solutions, Inc.

C#

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

パラメータ

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

参照

- 「ULBulkCopyColumnMappingCollection クラス」 499 ページ
- 「ULBulkCopyColumnMappingCollection メンバ」 499 ページ
- 「ULBulkCopyColumnMapping クラス」 492 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。

構文

Visual Basic

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

C#

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

パラメータ

- **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。

参照

- 「ULBulkCopyColumnMappingCollection クラス」 499 ページ
- 「ULBulkCopyColumnMappingCollection メンバ」 499 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULBulkCopyOptions 列挙

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処理フラグです。

構文

Visual Basic

Public Enum **ULBulkCopyOptions**

C#

public enum **ULBulkCopyOptions**

備考

ULBulkCopyOptions 列挙は、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。

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

メンバ

メンバ名	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。	0
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。	1
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。指定しないと、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、コンストラクタに ULTransaction オブジェクトも指定すると、System.ArgumentException が発生します。	2

参照

- 「ULBulkCopy クラス」 480 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

ULCommand クラス

IN パラメータあり、または IN パラメータ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。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。このクラスは継承できません。

構文

Visual Basic

```
Public NotInheritable Class ULCommand
  Inherits DbCommand
  Implements ICloneable
```

C#

```
public sealed class ULCommand : DbCommand,
  ICloneable
```

備考

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 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。

方法	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 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 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。
ULCommand.ExecuteTable()	**UL 拡張：** 直接の操作用に、ULTable のデータベース・テーブルを取り出します。ULCommand.CommandText はテーブルの名前として解釈され、ULCommand.IndexName はテーブルのソート順の指定に使用できます。ULCommand.CommandType は System.Data.CommandType.TableDirect であることが必要です。

Copyright © 2007, iAnywhere Solutions, Inc.

ULCommand.CommandText 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。

継承： System.Data.Common.DbCommand

実装： System.Data.IDbCommand、System.IDisposable

参照

- 「ULCommand メンバ」 510 ページ
- 「CreateCommand メソッド」 574 ページ
- 「Transaction プロパティ」 521 ページ
- 「ExecuteNonQuery メソッド」 535 ページ
- 「ExecuteReader() メソッド」 535 ページ
- 「ExecuteResultSet() メソッド」 538 ページ
- 「ExecuteScalar メソッド」 540 ページ
- 「ExecuteTable() メソッド」 541 ページ
- 「ULTable クラス」 933 ページ
- 「CommandText プロパティ」 516 ページ
- CommandType.TableDirect
- Component.Dispose
- 「CommandType プロパティ」 517 ページ
- 「IndexName プロパティ」 519 ページ
- DbCommand
- IDbCommand
- IDisposable
- 「ULResultSet クラス」 814 ページ
- 「ULDataReader クラス」 676 ページ

ULCommand メンバ

パブリック・コンストラクタ

メンバ名	説明
ULCommand コンストラクタ	「ULCommand クラス」 509 ページの新しいインスタンスを初期化します。

Copyright © 2007, iAnywhere Solutions, Inc.

パブリック・プロパティ

メンバ名	説明
CommandText プロパティ	ULCommand.CommandType 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。パラメータ化された文の場合、疑問符 (?) プレースホルダを使用してパラメータを渡します。
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 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 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を指定します。
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。このプロパティは、主に開発中の使用を目的とします。
Transaction プロパティ	ULCommand が実行される ULTransaction を指定します。
UpdatedRowSource プロパティ	ULDataAdapter の Update 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。

パブリック・メソッド

メンバ名	説明
BeginExecuteNonQuery メソッド	コールバック・プロシージャと状態情報を指定して、この ULCommand 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 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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	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 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 のデータベース・テーブルを取り出します。ULCommand.CommandText はテーブルの名前として解釈され、ULCommand.IndexName はテーブルのソート順の指定に使用できます。
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。

参照

- 「ULCommand クラス」 509 ページ
- 「CreateCommand メソッド」 574 ページ
- 「Transaction プロパティ」 521 ページ
- 「ExecuteNonQuery メソッド」 535 ページ
- 「ExecuteReader() メソッド」 535 ページ
- 「ExecuteResultSet() メソッド」 538 ページ
- 「ExecuteScalar メソッド」 540 ページ
- 「ExecuteTable() メソッド」 541 ページ
- 「ULTable クラス」 933 ページ
- 「CommandText プロパティ」 516 ページ
- CommandType.TableDirect
- Component.Dispose
- 「CommandType プロパティ」 517 ページ
- 「IndexName プロパティ」 519 ページ
- DbCommand
- IDbCommand
- IDisposable
- 「ULResultSet クラス」 814 ページ
- 「ULDataReader クラス」 676 ページ

ULCommand コンストラクタ

「ULCommand クラス」 509 ページの新しいインスタンスを初期化します。

ULCommand() コンストラクタ

ULCommand オブジェクトを初期化します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Sub New()
```

C#

```
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

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「ULCommand コンストラクタ」 512 ページ
- 「CreateCommand メソッド」 574 ページ
- 「ULCommand(String) コンストラクタ」 513 ページ
- 「ULCommand(String, ULConnection) コンストラクタ」 514 ページ
- 「ULCommand(String, ULConnection, ULTransaction) コンストラクタ」 515 ページ
- 「CommandText プロパティ」 516 ページ
- 「Connection プロパティ」 518 ページ
- 「Transaction プロパティ」 521 ページ

ULCommand(String) コンストラクタ

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。

構文

Visual Basic

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

C#

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

パラメータ

- **cmdText**　ULCommand.CommandType 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を渡します。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

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

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「ULCommand コンストラクタ」 512 ページ
- 「CreateCommand メソッド」 574 ページ
- 「ULCommand() コンストラクタ」 512 ページ
- 「ULCommand(String, ULConnection) コンストラクタ」 514 ページ
- 「ULCommand(String, ULConnection, ULTransaction) コンストラクタ」 515 ページ
- 「Connection プロパティ」 518 ページ
- 「Transaction プロパティ」 521 ページ
- 「CommandType プロパティ」 517 ページ
- CommandType.TableDirect

ULCommand(String, ULConnection) コンストラクタ

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。

構文

Visual Basic

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

C#

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

パラメータ

- **cmdText**　ULCommand.CommandType 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を渡します。
- **connection**　現在の接続を示す ULConnection オブジェクト。

備考

ULCommand オブジェクトに ULCommand.Transaction 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。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「ULCommand コンストラクタ」 512 ページ
- 「CreateCommand メソッド」 574 ページ
- 「ULCommand() コンストラクタ」 512 ページ
- 「ULCommand(String) コンストラクタ」 513 ページ
- 「ULCommand(String, ULConnection, ULTransaction) コンストラクタ」 515 ページ
- 「Transaction プロパティ」 521 ページ
- 「CommandType プロパティ」 517 ページ
- CommandType.TableDirect
- 「ULConnection クラス」 555 ページ

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

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。

構文

Visual Basic

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

C#

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

パラメータ

- **cmdText**　ULCommand.CommandType 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を渡します。
- **connection**　現在の接続を示す ULConnection オブジェクト。
- **transaction**　ULCommand が実行される ULTransaction。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「ULCommand コンストラクタ」 512 ページ
- 「CreateCommand メソッド」 574 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULCommand() コンストラクタ」 512 ページ
- 「ULCommand(String) コンストラクタ」 513 ページ
- 「ULCommand(String, ULConnection) コンストラクタ」 514 ページ
- 「CommandType プロパティ」 517 ページ
- CommandType.TableDirect
- 「ULTransaction クラス」 969 ページ

CommandText プロパティ

ULCommand.CommandType 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。パラメータ化された文の場合、疑問符 (?) プレースホルダを使用してパラメータを渡します。

構文

Visual Basic

Public Overrides Property **CommandText** As String

C#

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

プロパティ値

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。

備考

読み込み専用の結果セットの作成に使用する SELECT 文 (ULCommand.ExecuteReader() または ULCommand.ExecuteScalar) は、最後に FOR READ ONLY を指定することをおすすめします。テンポラリ・テーブルを使用する文で、パフォーマンスが大幅に向上することがあります。

例

次の例は、パラメータ化されたプレースホルダの使用方法を示します。

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

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

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「ExecuteNonQuery メソッド」 535 ページ
- 「ExecuteReader() メソッド」 535 ページ
- 「ExecuteResultSet() メソッド」 538 ページ
- 「ExecuteScalar メソッド」 540 ページ
- 「ExecuteTable() メソッド」 541 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「CommandType プロパティ」 517 ページ
- CommandType.TableDirect

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。

構文

Visual Basic

Public Overrides Property **CommandTimeout** As Integer

C#

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

プロパティ値

値は常にゼロです。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ

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。

構文

Visual Basic

Public Overrides Property **CommandType** As CommandType

C#

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

プロパティ値

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

備考

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 を指定するときは、ULCommand.CommandText はデータベース・テーブルの名前でなければなりません。また、ULCommand.IndexName 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() または ULCommand.ExecuteReader() を使用します。

Copyright © 2007, iAnywhere Solutions, Inc.

- System.Data.CommandType.Text - この CommandType を指定するときは、ULCommand.CommandText は SQL 文またはクエリでなければなりません。クエリ以外の SQL 文を実行するには、ULCommand.ExecuteNonQuery 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 を使用します。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- CommandType
- CommandType.Text
- CommandType.TableDirect
- 「CommandText プロパティ」 516 ページ
- 「IndexName プロパティ」 519 ページ
- 「ExecuteTable() メソッド」 541 ページ
- 「ExecuteReader() メソッド」 535 ページ
- CommandType.Text
- 「CommandText プロパティ」 516 ページ
- 「ExecuteNonQuery メソッド」 535 ページ
- 「ExecuteReader() メソッド」 535 ページ
- 「ExecuteScalar メソッド」 540 ページ

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。

構文

Visual Basic

Public Property **Connection** As ULConnection

C#

public ULConnection **Connection** { get; set; }

プロパティ値

コマンドを実行する ULConnection オブジェクト。

備考

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

- 「ULCommand クラス」 509 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULCommand メンバ」 510 ページ
- 「ULConnection クラス」 555 ページ
- IDbCommand.Connection
- DbCommand.Connection

DesignTimeVisible プロパティ

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

構文

Visual Basic

Public Overrides Property **DesignTimeVisible** As Boolean

C#

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

プロパティ値

ULCommand インスタンスを参照できるようにする場合は true、このインスタンス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。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ

IndexName プロパティ

UL 拡張： ULCommand.CommandType 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。

構文

Visual Basic

Public Property **IndexName** As String

C#

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

プロパティ値

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 クラス」 509 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULCommand メンバ」 510 ページ
- 「ExecuteTable() メソッド」 541 ページ
- 「ExecuteReader() メソッド」 535 ページ
- 「CommandType プロパティ」 517 ページ
- CommandType.TableDirect

Parameters プロパティ

現在の文のパラメータを指定します。

構文

Visual Basic

Public Readonly Property **Parameters** As ULParameterCollection

C#

public ULParameterCollection **Parameters** { get;}

プロパティ値

SQL 文のパラメータを保持する ULParameterCollection。デフォルト値は空のコレクションです。

備考

ULCommand.CommandText で疑問符を使用してパラメータを示します。コレクション内のパラメータは、疑問符のプレースホルダ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

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「ULParameterCollection クラス」 790 ページ
- 「ULParameter クラス」 773 ページ
- IDbCommand.Connection
- DbCommand.Connection
- 「CommandText プロパティ」 516 ページ

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。このプロパティは、主に開発中の使用を目的とします。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **Plan** As String

C#

public string **Plan** { get;}

プロパティ値

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。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ

Transaction プロパティ

ULCommand が実行される ULTransaction を指定します。

構文

Visual Basic

Public Property **Transaction** As ULTransaction

C#

public ULTransaction **Transaction** { get; set; }

プロパティ値

ULCommand が実行される ULTransaction。これは、ULCommand.Connection 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。

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

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「BeginTransaction() メソッド」 568 ページ
- 「ULTransaction クラス」 969 ページ
- 「Connection プロパティ」 518 ページ
- IDbCommand.Transaction

Copyright © 2007, iAnywhere Solutions, Inc.

- DbCommand.Transaction

UpdatedRowSource プロパティ

ULDataAdapter の Update 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。

構文

Visual Basic

Public Overrides Property **UpdatedRowSource** As UpdateRowSource

C#

public override UpdateRowSource **UpdatedRowSource** { get; set; }

プロパティ値

System.Data.UpdateRowSource 値の 1 つ。デフォルト値は System.Data.UpdateRowSource.Both です。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- UpdateRowSource
- UpdateRowSource.Both

BeginExecuteNonQuery メソッド

コールバック・プロシージャと状態情報を指定して、この ULCommand 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

構文

Visual Basic

Public Function **BeginExecuteNonQuery()** As IAsyncResult

C#

public IAsyncResult **BeginExecuteNonQuery();**

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「BeginExecuteNonQuery メソッド」 522 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

BeginExecuteNonQuery(AsyncCallback, Object) メソッド

コールバック・プロシージャと状態情報を指定して、この ULCommand で記述される SQL 文の非同期実行を開始します。

構文

Visual Basic

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

C#

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

パラメータ

- **callback**　コマンドの実行が終了すると起動される System.AsyncCallback 委任。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・プロシージャに渡される、ユーザ定義の状態オブジェクト。コールバック・プロシージャ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) を起動する場合にも必要です。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「BeginExecuteNonQuery メソッド」 522 ページ
- 「EndExecuteNonQuery メソッド」 528 ページ
- IAsyncResult.AsyncState
- AsyncCallback
- IAsyncResult

BeginExecuteReader メソッド

この ULCommand 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 © 2007, iAnywhere Solutions, Inc.

BeginExecuteReader() メソッド

この ULCommand 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。

構文

Visual Basic

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

C#

```
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 インスタンスを返す、EndExecuteReader (IAsyncResult) を起動する場合にも必要です。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「BeginExecuteReader メソッド」 523 ページ
- 「EndExecuteReader メソッド」 532 ページ
- IAsyncResult
- 「ULDataReader クラス」 676 ページ

BeginExecuteReader(CommandBehavior) メソッド

CommandBehavior 値の 1 つを使用してこの ULCommand 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。

構文

Visual Basic

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

C#

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

パラメータ

- **cmdBehavior**　クエリの結果の記述と、接続への影響の記述の System.Data.CommandBehavior フラグのビット処理の組み合わせ。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。

Copyright © 2007, iAnywhere Solutions, Inc.

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 インスタンスを返す、EndExecuteReader (IAsyncResult) を起動する場合にも必要です。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「BeginExecuteReader メソッド」 523 ページ
- 「EndExecuteReader メソッド」 532 ページ
- CommandBehavior
- CommandBehavior.Default
- CommandBehavior.CloseConnection
- CommandBehavior.SchemaOnly
- IAsyncResult
- 「EndExecuteReader メソッド」 532 ページ
- 「ULDataReader クラス」 676 ページ

BeginExecuteReader(AsyncCallback, Object) メソッド

コールバック・プロシージャと状態情報を指定して、この ULCommand 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。

構文

Visual Basic

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

C#

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

パラメータ

- **callback**　コマンドの実行が終了すると起動される System.AsyncCallback 委任。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・プロシージャに渡される、ユーザ定義のステータス・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。

Copyright © 2007, iAnywhere Solutions, Inc.

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 インスタンスを返す、EndExecuteReader (IAsyncResult) を起動する場合にも必要です。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「BeginExecuteReader メソッド」 523 ページ
- 「EndExecuteReader メソッド」 532 ページ
- AsyncCallback
- IAsyncResult.AsyncState
- IAsyncResult.AsyncState
- 「EndExecuteReader メソッド」 532 ページ
- IAsyncResult
- 「ULDataReader クラス」 676 ページ

BeginExecuteReader(AsyncCallback, Object, CommandBehavior) メソッド

コールバック・プロシージャと状態情報を指定して、CommandBehavior 値の 1 つを使用してこの ULCommand 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。

構文

Visual Basic

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

C#

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

パラメータ

- **callback**　コマンドの実行が終了すると起動される System.AsyncCallback 委任。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・プロシージャに渡される、ユーザ定義のステータス・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 フラグのビット処理の組み合わせ。Ultra Light.NET で有効なのは、

Copyright © 2007, iAnywhere Solutions, Inc.

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 インスタンスを返す、EndExecuteReader (IAsyncResult) を起動する場合にも必要です。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「BeginExecuteReader メソッド」 523 ページ
- AsyncCallback
- IAsyncResult.AsyncState
- CommandBehavior
- CommandBehavior.Default
- CommandBehavior.CloseConnection
- CommandBehavior.SchemaOnly
- IAsyncResult
- 「EndExecuteReader メソッド」 532 ページ
- 「ULDataReader クラス」 676 ページ

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。

構文

Visual Basic

Public Overrides Sub **Cancel()**

C#

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。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

CreateParameter メソッド

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

構文

Visual Basic

Public Function **CreateParameter()** As ULParameter

C#

public 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。

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 にプロパティを設定して、そのパラメータ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

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「ULParameter クラス」 773 ページ
- IDbCommand.CreateParameter
- DbCommand.CreateParameter

EndExecuteNonQuery メソッド

SQL 文の非同期実行を終了します。

構文

Visual Basic

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

C#

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

 Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

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

戻り値

影響を受けたローの数 (ExecuteNonQuery と同じ動作)。

備考

BeginExecuteNonQuery を呼び出すごとに、EndExecuteNonQuery を呼び出す必要があります。呼び出しは、BeginExecuteNonQuery が返されてから行います。ADO.NET はスレッドに対応していないため、各自で BeginExecuteNonQuery が返されたことを確認する必要があります。EndExecuteNonQuery 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 を呼び出して、BeginExecuteReader 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、EndExecuteNonQuery 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。

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 will block until the command completes
Dim rowCount As Integer = _
  cmd.EndExecuteNonQuery( res )

// C#
ULCommand cmd = new ULCommand(
    "UPDATE Departments"
    + " SET DepartmentName = 'Engineering'"
    + " WHERE DepartmentID=100",
    conn
  );
IAsyncResult res = cmd.BeginExecuteNonQuery();
// perform other work
// this will block 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'" _
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
    + " WHERE DepartmentID=100", _
    conn _
  )
Dim res As IAsyncResult res = _
  cmd.BeginExecuteNonQuery()
While( !res.IsCompleted )
  ' do other work
End While
' this will block until the command completes
Dim rowCount As Integer = _
  cmd.EndExecuteNonQuery( res )

// C#
ULCommand cmd = new ULCommand(
    "UPDATE Departments"
    + " SET DepartmentName = 'Engineering'"
    + " WHERE DepartmentID=100",
    conn
  );
IAsyncResult res = cmd.BeginExecuteNonQuery();
while( !res.IsCompleted ) {
  // do other work
}
// this will block 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。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 will not block because the command is finished
Dim rowCount As Integer = _
  cmd.EndExecuteNonQuery( res )

// 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 will not block because the command is finished
int rowCount = cmd.EndExecuteNonQuery( res );
```

BeginExecuteNonQuery の呼び出し時にコールバック関数を指定する。BeginExecuteNonQuery 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。

Copyright © 2007, iAnywhere Solutions, Inc.

```
' Visual Basic
Private Sub callbackFunction(ByVal ar As IAsyncResult)
  Dim cmd As ULCommand = _
    CType(ar.AsyncState, ULCommand)
  ' this wona ?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
  ' will be called when the command completes
End Sub


// C#
private void callbackFunction( IAsyncResult ar )
{
  ULCommand cmd = (ULCommand) ar.AsyncState;
  // this wona ?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
  // will be 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・インタフェースの更新に関する通常の注意が適用されます。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「BeginExecuteNonQuery() メソッド」 522 ページ
- IAsyncResult

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

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

C#

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

パラメータ

- **asyncResult** BeginExecuteReader への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 と同じ動作)。

備考

BeginExecuteReader を呼び出すごとに、EndExecuteReader を呼び出す必要があります。呼び出しは、BeginExecuteReader が返されてから行います。ADO.NET はスレッドに対応していないため、各自で BeginExecuteReader が返されたことを確認する必要があります。EndExecuteReader 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。EndExecuteReade 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 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。

EndExecuteReader を呼び出す。EndExecuteReader 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 will block until the command completes
Dim reader As ULDataReader = _
  cmd.EndExecuteReader( res )

// C#
ULCommand cmd = new ULCommand(
    "SELECT * FROM Departments", conn
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
  );
IAsyncResult res = cmd.BeginExecuteReader();
// perform other work
// this will block 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 )
  ' do other work
End While
' this will block 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 ) {
  // do other work
}
// this will block 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。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 will not block because the command is finished
Dim reader As ULDataReader = _
  cmd.EndExecuteReader( res )

// C#
ULCommand cmd = new ULCommand(
    "SELECT * FROM Departments", conn
  );
IAsyncResult res = cmd.BeginExecuteReader();
// perform other work
WaitHandle wh = res.AsyncWaitHandle;
wh.WaitOne();
// this will not block because the command is finished
ULDataReader reader = cmd.EndExecuteReader( res );
```

Copyright © 2007, iAnywhere Solutions, Inc.

BeginExecuteReader の呼び出し時にコールバック関数を指定する。BeginExecuteReader 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 wona ?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
  ' will be called when the command completes
End Sub


// C#
private void callbackFunction( IAsyncResult ar )
{
  ULCommand cmd = (ULCommand) ar.AsyncState;
  // this wona ?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
  // will be 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・インタフェースの更新に関する通常の注意が適用されます。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「BeginExecuteReader() メソッド」 524 ページ
- 「ULDataReader クラス」 676 ページ
- IAsyncResult

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

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

C#

public override int **ExecuteNonQuery();**

戻り値

影響を受けたローの数。

備考

この文は、必要に応じて ULCommand.CommandText と ULCommand.Parameters 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 を System.Data.CommandType.TableDirect にすることはできません。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「CommandText プロパティ」 516 ページ
- 「Parameters プロパティ」 520 ページ
- 「CommandType プロパティ」 517 ページ
- CommandType.TableDirect
- 「Connection プロパティ」 518 ページ
- 「Transaction プロパティ」 521 ページ
- 「CommandText プロパティ」 516 ページ

ExecuteReader メソッド

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Function **ExecuteReader()** As ULDataReader

C#

public ULDataReader **ExecuteReader();**

戻り値

ULDataReader オブジェクトとして返される結果セット。

備考

この文は、必要に応じて ULCommand.CommandText と ULCommand.Parameters 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()、ULCommand.ExecuteTable()、または ULDataAdapter を使用します。

ULCommand.CommandType が System.Data.CommandType.TableDirect である場合、ExecuteReader は ULCommand.ExecuteTable() を実行し、ULDataReader としての ULTable ダウンキャストを返します。

このメソッドを使用して読み込み専用の結果セットを作成する SELECT 文は、最後に FOR READ ONLY を指定することをおすすめします。テンポラリ・テーブルを使用する文で、パフォーマンスが大幅に向上することがあります。

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

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「ExecuteReader メソッド」 535 ページ
- 「ExecuteReader(CommandBehavior) メソッド」 537 ページ
- 「CommandText プロパティ」 516 ページ
- 「Parameters プロパティ」 520 ページ
- 「ULDataReader クラス」 676 ページ
- 「ExecuteResultSet() メソッド」 538 ページ
- 「ExecuteTable() メソッド」 541 ページ
- 「ULDataAdapter クラス」 645 ページ
- 「CommandType プロパティ」 517 ページ
- CommandType.TableDirect
- 「ExecuteTable() メソッド」 541 ページ
- 「ULTable クラス」 933 ページ
- 「ULDataReader クラス」 676 ページ
- IDbCommand.ExecuteReader
- DbCommand.ExecuteReader
- 「Connection プロパティ」 518 ページ
- 「Transaction プロパティ」 521 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

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

C#

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

パラメータ

- **cmdBehavior** クエリの結果の記述と、接続への影響の記述の System.Data.CommandBehavior フラグのビット処理の組み合わせ。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。

備考

この文は、必要に応じて ULCommand.CommandText と ULCommand.Parameters 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(CommandBehavior)、ULCommand.ExecuteTable(CommandBehavior)、または ULDataAdapter を使用します。

ULCommand.CommandType が System.Data.CommandType.TableDirect である場合、ExecuteReader は ULCommand.ExecuteTable(CommandBehavior) を実行し、ULDataReader としての ULTable ダウンキャストを返します。

このメソッドを使用して読み込み専用の結果セットを作成する SELECT 文は、最後に FOR READ ONLY を指定することをおすすめします。テンポラリ・テーブルを使用する文で、パフォーマンスが大幅に向上することがあります。

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 クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「ExecuteReader メソッド」 535 ページ
- 「ExecuteReader() メソッド」 535 ページ
- 「CommandText プロパティ」 516 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「Parameters プロパティ」 520 ページ
- 「ULDataReader クラス」 676 ページ
- 「ExecuteResultSet(CommandBehavior) メソッド」 539 ページ
- 「ExecuteTable(CommandBehavior) メソッド」 542 ページ
- 「ULDataAdapter クラス」 645 ページ
- 「CommandType プロパティ」 517 ページ
- CommandType.TableDirect
- 「ExecuteTable(CommandBehavior) メソッド」 542 ページ
- 「ULTable クラス」 933 ページ
- 「ULDataReader クラス」 676 ページ
- IDbCommand.ExecuteReader
- DbCommand.ExecuteReader
- CommandBehavior
- CommandBehavior.Default
- CommandBehavior.CloseConnection
- CommandBehavior.SchemaOnly
- 「Connection プロパティ」 518 ページ
- 「Transaction プロパティ」 521 ページ
- 「CommandText プロパティ」 516 ページ

ExecuteResultSet メソッド

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

構文

Visual Basic

Public Function **ExecuteResultSet()** As ULResultSet

C#

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。

備考

この文は、必要に応じて ULCommand.CommandText と ULCommand.Parameters 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() または ULDataAdapter を使用します。

Copyright © 2007, iAnywhere Solutions, Inc.

ULCommand.CommandType が System.Data.CommandType.TableDirect である場合、ExecuteReader は ULCommand.ExecuteTable() を実行し、ULResultSet としての ULTable ダウンキャストを返します。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「ExecuteResultSet メソッド」 538 ページ
- 「ExecuteResultSet(CommandBehavior) メソッド」 539 ページ
- 「ULCommand クラス」 509 ページ
- 「CommandText プロパティ」 516 ページ
- 「Parameters プロパティ」 520 ページ
- 「CommandType プロパティ」 517 ページ
- CommandType.TableDirect
- 「ULResultSet クラス」 814 ページ
- 「ULTable クラス」 933 ページ

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 として結果セットを返します。

構文

Visual Basic

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

C#

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

パラメータ

- **cmdBehavior**　クエリの結果の記述と、接続への影響の記述の System.Data.CommandBehavior フラグのビット処理の組み合わせ。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。

備考

この文は、必要に応じて ULCommand.CommandText と ULCommand.Parameters 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

Copyright © 2007, iAnywhere Solutions, Inc.

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) または ULDataAdapter を使用します。

ULCommand.CommandType が System.Data.CommandType.TableDirect である場合、ExecuteReader は ULCommand.ExecuteTable(CommandBehavior) を実行し、ULResultSet としての ULTable ダウンキャストを返します。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「ExecuteResultSet メソッド」 538 ページ
- 「ExecuteReader() メソッド」 535 ページ
- 「ULResultSet クラス」 814 ページ
- 「CommandText プロパティ」 516 ページ
- 「Parameters プロパティ」 520 ページ
- 「ExecuteTable(CommandBehavior) メソッド」 542 ページ
- 「ULDataAdapter クラス」 645 ページ
- 「CommandType プロパティ」 517 ページ
- CommandType.TableDirect
- 「ExecuteTable(CommandBehavior) メソッド」 542 ページ
- 「ULTable クラス」 933 ページ
- CommandBehavior
- CommandBehavior.Default
- CommandBehavior.CloseConnection
- CommandBehavior.SchemaOnly
- 「Connection プロパティ」 518 ページ
- 「Transaction プロパティ」 521 ページ

ExecuteScalar メソッド

SQL SELECT 文を実行し、単一の値を返します。

構文

Visual Basic

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

C#

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)。

 Copyright © 2007, iAnywhere Solutions, Inc.

備考

この文は、必要に応じて ULCommand.CommandText と ULCommand.Parameters 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 が System.Data.CommandType.TableDirect である場合、ExecuteScalar は ULCommand.ExecuteTable() を実行し、最初のローの最初のカラムを返します。

このメソッドを使用して読み込み専用の結果セットを作成する SELECT 文は、最後に FOR READ ONLY を指定することをおすすめします。テンポラリ・テーブルを使用する文で、パフォーマンスが大幅に向上することがあります。

参照

- 「ULCommand クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「Connection プロパティ」 518 ページ
- 「Transaction プロパティ」 521 ページ
- 「CommandText プロパティ」 516 ページ
- 「IndexName プロパティ」 519 ページ
- CommandBehavior
- CommandBehavior.Default
- CommandBehavior.CloseConnection
- CommandBehavior.SchemaOnly
- 「CommandType プロパティ」 517 ページ
- 「Parameters プロパティ」 520 ページ
- CommandType.TableDirect

ExecuteTable メソッド

UL 拡張： 直接の操作用に、ULTable のデータベース・テーブルを取り出します。ULCommand.CommandText はテーブルの名前として解釈され、ULCommand.IndexName はテーブルのソート順の指定に使用できます。

ExecuteTable() メソッド

UL 拡張： 直接の操作用に、ULTable のデータベース・テーブルを取り出します。ULCommand.CommandText はテーブルの名前として解釈され、ULCommand.IndexName はテーブルのソート順の指定に使用できます。

構文

Visual Basic

Public Function **ExecuteTable()** As ULTable

Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public ULTable ExecuteTable();
```

戻り値

ULTable オブジェクトとして返されるテーブル。

備考

ULCommand.CommandType は System.Data.CommandType.TableDirect に設定することが必要です。

ULCommand.IndexName が null 参照 (Visual Basic の Nothing) であると、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 クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「ExecuteTable メソッド」 541 ページ
- 「ExecuteTable(CommandBehavior) メソッド」 542 ページ
- 「Connection プロパティ」 518 ページ
- 「Transaction プロパティ」 521 ページ
- 「CommandText プロパティ」 516 ページ
- 「IndexName プロパティ」 519 ページ
- CommandBehavior
- CommandBehavior.Default
- CommandBehavior.CloseConnection
- CommandBehavior.SchemaOnly
- 「CommandType プロパティ」 517 ページ
- CommandType.TableDirect
- 「ULTable クラス」 933 ページ

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

構文

Visual Basic

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

C#

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


Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **cmdBehavior**　クエリの結果の記述と、接続への影響の記述の System.Data.CommandBehavior フラグのビット処理の組み合わせ。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返されるテーブル。

備考

ULCommand.CommandType は System.Data.CommandType.TableDirect に設定することが必要です。

ULCommand.IndexName が null 参照 (Visual Basic の Nothing) であると、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 クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「ExecuteTable メソッド」 541 ページ
- 「ExecuteTable() メソッド」 541 ページ
- 「Connection プロパティ」 518 ページ
- 「Transaction プロパティ」 521 ページ
- 「CommandText プロパティ」 516 ページ
- 「IndexName プロパティ」 519 ページ
- CommandBehavior
- CommandBehavior.Default
- CommandBehavior.CloseConnection
- CommandBehavior.SchemaOnly
- 「CommandType プロパティ」 517 ページ
- CommandType.TableDirect

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。

構文

Visual Basic

Public Overrides Sub **Prepare()**

C#

public override void **Prepare();**

Copyright © 2007, iAnywhere Solutions, Inc.

備考

文を事前にコンパイルすると、パラメータ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 クラス」 509 ページ
- 「ULCommand メンバ」 510 ページ
- 「Connection プロパティ」 518 ページ
- 「Transaction プロパティ」 521 ページ
- 「CommandText プロパティ」 516 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Class **ULCommandBuilder**
  Inherits DbCommandBuilder

C#

public class **ULCommandBuilder** : DbCommandBuilder

備考

ULDataAdapter 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 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 によって生成されます。

継承： System.ComponentModel.Component

実装： System.IDisposable

例

次の例では、ULCommand を ULDataAdapter と ULConnection 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 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)

    'code to modify data in DataSet here

    'Without the ULCommandBuilder this line would fail
    adapter.Update(dataSet, tableName)

    Return dataSet
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
End Function

// 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);

          //code to modify data in DataSet here

        //Without the ULCommandBuilder this line would fail
        adapter.Update(dataSet, tableName);

        return dataSet;
    }
}
```

参照

- 「ULCommandBuilder メンバ」 546 ページ
- DataSet
- 「ULDataAdapter クラス」 645 ページ
- コンポーネント
- IDisposable
- 「ULCommand クラス」 509 ページ
- 「ULConnection クラス」 555 ページ

ULCommandBuilder メンバ

パブリック・コンストラクタ

メンバ名	説明
ULCommandBuilder コンストラクタ	「ULCommandBuilder クラス」 545 ページの新しいインスタンスを初期化します。

パブリック・プロパティ

メンバ名	説明
CatalogLocation (DbCommandBuilder から継承)	

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
CatalogSeparator (DbCommandBuilder から継承)	
ConflictOption (DbCommandBuilder から継承)	
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 (DbCommandBuilder から継承)	
QuoteSuffix (DbCommandBuilder から継承)	
SchemaSeparator (DbCommandBuilder から継承)	
SetAllValues (DbCommandBuilder から継承)	

パブリック・メソッド

メンバ名	説明
GetDeleteCommand メソッド	データ・ソースでの削除に必要な自動生成された DbCommand オブジェクトを取得します。
GetInsertCommand メソッド	データ・ソースでの挿入に必要な自動生成された DbCommand オブジェクトを取得します。
GetUpdateCommand メソッド	データ・ソースでの更新に必要な自動生成された DbCommand オブジェクトを取得します。
QuoteIdentifier (DbCommandBuilder から継承)	
RefreshSchema (DbCommandBuilder から継承)	
UnquoteIdentifier (DbCommandBuilder から継承)	

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULCommandBuilder クラス」 545 ページ
- DataSet
- 「ULDataAdapter クラス」 645 ページ
- コンポーネント
- IDisposable
- 「ULCommand クラス」 509 ページ
- 「ULConnection クラス」 555 ページ

ULCommandBuilder コンストラクタ

「ULCommandBuilder クラス」 545 ページの新しいインスタンスを初期化します。

ULCommandBuilder() コンストラクタ

ULCommandBuilder オブジェクトを初期化します。

構文

Visual Basic

```
Public Sub New()
```

C#

```
public  ULCommandBuilder();
```

参照

- 「ULCommandBuilder クラス」 545 ページ
- 「ULCommandBuilder メンバ」 546 ページ
- 「ULCommandBuilder コンストラクタ」 548 ページ
- 「ULCommandBuilder(ULDataAdapter) コンストラクタ」 548 ページ

ULCommandBuilder(ULDataAdapter) コンストラクタ

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。

構文

Visual Basic

```
Public Sub New( _
  ByVal adapter As ULDataAdapter _
)
```

C#

```
public  ULCommandBuilder(
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
  ULDataAdapter adapter
);
```

パラメータ

- **adapter**　ULDataAdapter オブジェクト。

参照

- 「ULCommandBuilder クラス」 545 ページ
- 「ULCommandBuilder メンバ」 546 ページ
- 「ULCommandBuilder コンストラクタ」 548 ページ
- 「ULDataAdapter クラス」 645 ページ

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。

構文

Visual Basic

```
Public Property DataAdapter As ULDataAdapter
```

C#

```
public ULDataAdapter DataAdapter  { get; set; }
```

プロパティ値

ULDataAdapter オブジェクト。

参照

- 「ULCommandBuilder クラス」 545 ページ
- 「ULCommandBuilder メンバ」 546 ページ
- 「ULDataAdapter クラス」 645 ページ

GetDeleteCommand メソッド

データ・ソースでの削除に必要な自動生成された DbCommand オブジェクトを取得します。

GetDeleteCommand(Boolean) 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。

構文

Visual Basic

```
Public Function GetDeleteCommand( _
```

```
  ByVal useColumnsForParameterNames As Boolean _
) As ULCommand
```

C#

```
public ULCommand GetDeleteCommand(
  bool useColumnsForParameterNames
);
```

パラメータ

- **useColumnsForParameterNames** true の場合、カラム名に一致するパラメータ名を生成します (可能な場合)。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。

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、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

- 「ULCommandBuilder クラス」 545 ページ
- 「ULCommandBuilder メンバ」 546 ページ
- 「GetDeleteCommand メソッド」 549 ページ
- 「ULCommand クラス」 509 ページ
- DbCommandBuilder.RefreshSchema
- 「SelectCommand プロパティ」 652 ページ
- DbDataAdapter.Update
- DbCommandBuilder.DataAdapter

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。

構文

Visual Basic

```
Public Function GetDeleteCommand() As ULCommand
```

C#

```
public ULCommand GetDeleteCommand();
```


Copyright © 2007, iAnywhere Solutions, Inc.

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。

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、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

- 「ULCommandBuilder クラス」 545 ページ
- 「ULCommandBuilder メンバ」 546 ページ
- 「GetDeleteCommand メソッド」 549 ページ
- 「ULCommand クラス」 509 ページ
- DbCommandBuilder.RefreshSchema
- 「SelectCommand プロパティ」 652 ページ
- DbDataAdapter.Update
- DbCommandBuilder.DataAdapter

GetInsertCommand メソッド

データ・ソースでの挿入に必要な自動生成された DbCommand オブジェクトを取得します。

GetInsertCommand(Boolean) 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。

構文

Visual Basic

```
Public Function GetInsertCommand( _
  ByVal useColumnsForParameterNames As Boolean _
) As ULCommand
```

C#

```
public ULCommand GetInsertCommand(
  bool useColumnsForParameterNames
);
```

パラメータ

- **useColumnsForParameterNames** true の場合、カラム名に一致するパラメータ名を生成します (可能な場合)。false の場合、@p1、@p2 などを生成します。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

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、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) または GetInsertCommand メソッドを呼び出したときです。

参照

- 「ULCommandBuilder クラス」 545 ページ
- 「ULCommandBuilder メンバ」 546 ページ
- 「GetInsertCommand メソッド」 551 ページ
- 「ULCommand クラス」 509 ページ
- DbCommandBuilder.RefreshSchema
- 「SelectCommand プロパティ」 652 ページ
- DbDataAdapter.Update
- DbCommandBuilder.DataAdapter

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。

構文

Visual Basic

Public Function **GetInsertCommand()** As ULCommand

C#

public 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。

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、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) または GetInsertCommand メソッドを呼び出したときです。

参照

- 「ULCommandBuilder クラス」 545 ページ
- 「ULCommandBuilder メンバ」 546 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「GetInsertCommand メソッド」 551 ページ
- 「ULCommand クラス」 509 ページ
- DbCommandBuilder.RefreshSchema
- 「SelectCommand プロパティ」 652 ページ
- 「ULCommand クラス」 509 ページ
- DbCommandBuilder.DataAdapter

GetUpdateCommand メソッド

データ・ソースでの更新に必要な自動生成された DbCommand オブジェクトを取得します。

GetUpdateCommand(Boolean) 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。

構文

Visual Basic

```
Public Function GetUpdateCommand( _
  ByVal useColumnsForParameterNames As Boolean _
) As ULCommand
```

C#

```
public ULCommand GetUpdateCommand(
  bool useColumnsForParameterNames
);
```

パラメータ

- **useColumnsForParameterNames** true の場合、カラム名に一致するパラメータ名を生成します (可能な場合)。false の場合、@p1、@p2 などを生成します。

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。

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

- 「ULCommandBuilder クラス」 545 ページ
- 「ULCommandBuilder メンバ」 546 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「GetUpdateCommand メソッド」 553 ページ
- 「ULCommand クラス」 509 ページ
- DbCommandBuilder.RefreshSchema
- 「SelectCommand プロパティ」 652 ページ
- DbDataAdapter.Update
- DbCommandBuilder.DataAdapter

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。

構文

Visual Basic

Public Function **GetUpdateCommand()** As ULCommand

C#

public 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。

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

- 「ULCommandBuilder クラス」 545 ページ
- 「ULCommandBuilder メンバ」 546 ページ
- 「GetUpdateCommand メソッド」 553 ページ
- 「ULCommand クラス」 509 ページ
- DbCommandBuilder.RefreshSchema
- 「SelectCommand プロパティ」 652 ページ
- DbDataAdapter.Update
- DbCommandBuilder.DataAdapter

Copyright © 2007, iAnywhere Solutions, Inc.

ULConnection クラス

Ultra Light.NET データベースへの接続を示します。このクラスは継承できません。

構文

Visual Basic

```
Public NotInheritable Class ULConnection
  Inherits DbConnection
```

C#

```
public sealed class ULConnection : DbConnection
```

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 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 を使用して開きます。

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 を使用してアクセスできます。

実装： System.Data.IDbConnection、System.IDisposable

参照

- 「ULConnection メンバ」 555 ページ
- 「RuntimeType プロパティ」 658 ページ
- 「Open メソッド」 584 ページ
- 「Schema プロパティ」 565 ページ
- IDbConnection
- IDisposable

ULConnection メンバ

パブリック・コンストラクタ

メンバ名	説明
ULConnection コンストラクタ	「ULConnection クラス」 555 ページの新しいインスタンスを初期化します。

Copyright © 2007, iAnywhere Solutions, Inc.

パブリック・フィールド

メンバ名	説明
INVALID_DATABASE_ID フィールド	**UL 拡張：** ULConnection.DatabaseID 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。このフィールドは定数であり、読み込み専用です。

パブリック・プロパティ

メンバ名	説明
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を指定します。接続文字列は、ULConnectionParms オブジェクトを使用して指定できます。
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。
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。
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。
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。
DatabaseManager プロパティ	**UL 拡張：** シングルトンの ULDatabaseManager オブジェクトへのアクセスに使用します。
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。

パブリック・メソッド

メンバ名	説明
BeginTransaction メソッド	データベース・トランザクションを起動します。
ChangeDatabase メソッド	開いている ULConnection の現在のデータベースを変更します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。
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。ULCommand のプロパティを使用して、その動作を制御できます。
EnlistTransaction (DbConnection から継承)	
ExecuteTable メソッド	**UL 拡張：** 直接の操作用に、ULTable のデータベース・テーブルを取り出します。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)。
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)。
GetSchema メソッド	この DbConnection のデータ・ソースに関するスキーマ情報を返します。
GrantConnectTo メソッド	**UL 拡張：** 指定されたパスワードを持つユーザ ID に、Ultra Light データベースへのアクセスを許可します。
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。
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 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。
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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	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。

パブリック・イベント

メンバ名	説明
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。

参照

- 「ULConnection クラス」 555 ページ
- 「RuntimeType プロパティ」 658 ページ
- 「Open メソッド」 584 ページ
- 「Schema プロパティ」 565 ページ
- IDbConnection
- IDisposable

ULConnection コンストラクタ

「ULConnection クラス」 555 ページの新しいインスタンスを初期化します。

ULConnection() コンストラクタ

ULConnection オブジェクトを初期化します。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。

構文

Visual Basic

```
Public Sub New()
```

C#

```
public  ULConnection();
```

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 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 が設定されていないと、開くことができません。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「ULConnection コンストラクタ」 558 ページ
- 「Open メソッド」 584 ページ
- 「ConnectionString プロパティ」 560 ページ

ULConnection(String) コンストラクタ

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。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。

構文

Visual Basic

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

C#

```
public  ULConnection(
  string  connectionString
);
```

パラメータ

- **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。

  パラメータのリストについては、「ConnectionString プロパティ」 560 ページを参照してください。

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 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。

例

次のコードでは、Windows CE 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.DatabaseOnCE = "¥UltraLite¥MyDatabase.udb"
Dim conn As ULConnection = _
  New ULConnection( openParms.ToString() )
conn.Open()
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
// C#
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnCE = @"¥UltraLite¥MyDatabase.udb";
ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「ULConnection コンストラクタ」 558 ページ
- 「Open メソッド」 584 ページ
- 「ULConnection() コンストラクタ」 558 ページ
- 「RuntimeType プロパティ」 658 ページ
- 「ULConnectionParms クラス」 592 ページ
- 「ConnectionString プロパティ」 560 ページ

INVALID_DATABASE_ID フィールド

UL 拡張： ULConnection.DatabaseID 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。このフィールドは定数で、読み込み専用です。

構文

Visual Basic

Public Shared **INVALID_DATABASE_ID** As Long

C#

public const long **INVALID_DATABASE_ID** ;

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「DatabaseID プロパティ」 563 ページ

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

構文

Visual Basic

Public Overrides Property **ConnectionString** As String

C#

public override string **ConnectionString** { get; set; }

 Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

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、この接続を開くためのパラメータ。デフォルトは、空の文字列 (無効な接続文字列) です。

備考

UL 拡張： Ultra Light.NET によって使用されるパラメータは 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。パラメータのリストについては、「Ultra Light の接続文字列パラメータのリファレンス」『Ultra Light - データベース管理とリファレンス』を参照してください。

パラメータ値には、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。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、ユーザ名 DBA のパスワードを変更するか、新しいユーザを作成して (GrantConnectTo を使用)、DBA ユーザを削除します (RevokeConnectFrom を使用)。

例

次のコードでは、Windows CE 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.DatabaseOnCE = "¥UltraLite¥MyDatabase.udb"
Dim conn As ULConnection = New ULConnection
conn.ConnectionString = openParms.ToString()
conn.Open()

// C#
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnCE = @"¥UltraLite¥MyDatabase.udb";
ULConnection conn = new ULConnection();
conn.ConnectionString = openParms.ToString();
conn.Open();
```

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「Open メソッド」 584 ページ
- 「ULConnectionParms クラス」 592 ページ
- 「GrantConnectTo メソッド」 583 ページ

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。

構文

Visual Basic

Public Overrides Readonly Property **ConnectionTimeout** As Integer

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public override int **ConnectionTimeout** { get;}

プロパティ値

値は常にゼロです。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ

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。

構文

Visual Basic

Public Overrides Readonly Property **DataSource** As String

C#

public override string **DataSource** { get;}

プロパティ値

値は常に空の文字列です。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ

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。

構文

Visual Basic

Public Overrides Readonly Property **Database** As String

C#

public override string **Database** { get;}

プロパティ値

データベースの名前が含まれる文字列。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

Windows CE デバイスでは、ULConnection 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 は dbn、nt_file の順に接続文字列を参照します。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ

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。

構文

Visual Basic

Public Property **DatabaseID** As Long

C#

public long **DatabaseID** { get; set; }

プロパティ値

現在のデータベースのデータベース ID の値。

備考

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

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「GetDatabaseProperty メソッド」 669 ページ
- 「SetDatabaseOption メソッド」 674 ページ
- UInt32.MaxValue
- 「INVALID_DATABASE_ID フィールド」 560 ページ

DatabaseManager プロパティ

UL 拡張： シングルトンの ULDatabaseManager オブジェクトへのアクセスに使用します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Shared Readonly Property **DatabaseManager** As ULDatabaseManager

C#

public const ULDatabaseManager **DatabaseManager** { get;}

プロパティ値

シングルトンの ULDatabaseManager オブジェクトへの参照。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「ULDatabaseManager クラス」 657 ページ

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。

構文

Visual Basic

Public Readonly Property **GlobalAutoIncrementUsage** As Short

C#

public short **GlobalAutoIncrementUsage** { get;}

プロパティ値

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 ー 100 の範囲内の整数です。

備考

比率が 100% に近づいたら、ULConnection.DatabaseID を使用して、使用中のアプリケーションに新しいグローバル・データベース ID を設定してください。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「ULDatabaseManager クラス」 657 ページ
- 「DatabaseID プロパティ」 563 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。

 Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **LastIdentity** As UInt64

C#

public ulong **LastIdentity** { get;}

プロパティ値

直前に使用した identity の値 (符号なし long)。

備考

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 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 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。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ

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。

構文

Visual Basic

Public Readonly Property **Schema** As ULDatabaseSchema

C#

public ULDatabaseSchema **Schema** { get;}

プロパティ値

この接続が開かれているデータベースのスキーマを示す ULDatabaseSchema オブジェクトへの参照。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「ULDatabaseSchema クラス」 665 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。

構文

Visual Basic

Public Overrides Readonly Property **ServerVersion** As String

C#

public override string **ServerVersion** { get;}

プロパティ値

値は常に空の文字列です。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ

State プロパティ

接続の現在のステータスを返します。

構文

Visual Basic

Public Overrides Readonly Property **State** As ConnectionState

C#

public override ConnectionState **State** { get;}

プロパティ値

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。

参照

- 「ULConnection クラス」 555 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULConnection メンバ」 555 ページ
- 「StateChange イベント」 590 ページ
- ConnectionState.Open
- ConnectionState.Closed

SyncParms プロパティ

UL 拡張： この接続の同期設定を指定します。

構文

Visual Basic

Public Readonly Property **SyncParms** As ULSyncParms

C#

public ULSyncParms **SyncParms** { get;}

プロパティ値

この接続によって同期に使用されるパラメータを参照する ULSyncParms オブジェクトへの参照。パラメータを修正すると、この接続で行われる次の同期に反映されます。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「Synchronize() メソッド」 587 ページ
- 「SyncResult プロパティ」 567 ページ
- 「ULSyncParms クラス」 896 ページ

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。

構文

Visual Basic

Public Readonly Property **SyncResult** As ULSyncResult

C#

public ULSyncResult **SyncResult** { get;}

プロパティ値

この接続の最後の同期結果を参照する ULSyncResult オブジェクトへの参照。

参照

- 「ULConnection クラス」 555 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULConnection メンバ」 555 ページ
- 「Synchronize() メソッド」 587 ページ
- 「SyncParms プロパティ」 567 ページ
- 「SyncResult プロパティ」 567 ページ

BeginTransaction メソッド

データベース・トランザクションを起動します。

BeginTransaction() メソッド

トランザクション・オブジェクトを返します。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 または ULTransaction.Rollback で終了します。

構文

Visual Basic

Public Function **BeginTransaction()** As ULTransaction

C#

public ULTransaction **BeginTransaction();**

戻り値

新しいトランザクションを示す ULTransaction オブジェクト。

備考

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 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 を呼び出すまで、接続はオートコミット・モードに戻ります。

これは、System.Data.IDbConnection.BeginTransaction() と System.Data.Common.DbConnection.BeginTransaction() が厳密に型指定されたものです。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「BeginTransaction メソッド」 568 ページ
- 「BeginTransaction(IsolationLevel) メソッド」 569 ページ
- 「Commit メソッド」 971 ページ
- 「Rollback メソッド」 972 ページ
- 「Transaction プロパティ」 521 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「CreateCommand メソッド」 574 ページ
- IDbConnection.BeginTransaction
- DbConnection.BeginTransaction

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。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 または ULTransaction.Rollback で終了します。

構文

Visual Basic

```
Public Function BeginTransaction( _
  ByVal isolationLevel As IsolationLevel _
) As ULTransaction
```

C#

```
public ULTransaction BeginTransaction(
  IsolationLevel isolationLevel
);
```

パラメータ

- **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 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示す ULTransaction オブジェクト。

備考

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 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 を呼び出すまで、接続はオートコミット・モードに戻ります。

これは、System.Data.IDbConnection.BeginTransaction(System.Data.IsolationLevel) と System.Data.Common.DbConnection.BeginTransaction(System.Data.IsolationLevel) が厳密に型指定されたものです。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「BeginTransaction メソッド」 568 ページ
- 「ULTransaction クラス」 969 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「BeginTransaction() メソッド」 568 ページ
- 「Commit メソッド」 971 ページ
- 「Rollback メソッド」 972 ページ
- 「Transaction プロパティ」 521 ページ
- 「CreateCommand メソッド」 574 ページ
- IDbConnection.BeginTransaction
- DbConnection.BeginTransaction
- IsolationLevel.ReadUncommitted

ChangeDatabase メソッド

開いている ULConnection の現在のデータベースを変更します。

構文

Visual Basic

```
Public Overrides Sub ChangeDatabase( _
  ByVal connectionString As String _
)
```

C#

```
public override void ChangeDatabase(
  string  connectionString
);
```

パラメータ

- **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 または dbf ではなく、完全な接続文字列です。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「ConnectionString プロパティ」 560 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。

構文

Visual Basic

```
Public Sub ChangeEncryptionKey( _
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
  ByVal newKey As String _
)
```

C#

```
public void ChangeEncryptionKey(
  string  newKey
);
```

パラメータ

- **newkey**　データベースの新しい暗号化キー。

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。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「EncryptionKey プロパティ」 601 ページ

ChangePassword メソッド

接続文字列に示されるユーザのパスワードを、指定される新しいパスワードに変更します。

構文

Visual Basic

```
Public Shared Sub ChangePassword( _
  ByVal connectionString As String, _
  ByVal newPassword As String _
)
```

C#

```
public static void ChangePassword(
  string  connectionString,
  string  newPassword
);
```

パラメータ

- **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 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。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

Close メソッド

データベース接続を閉じます。

構文

Visual Basic

Public Overrides Sub **Close()**

C#

public override void **Close();**

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、Close を複数回呼び出すことができます。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ

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。

CountUploadRows(Int32, UInt32) 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。

構文

Visual Basic

```
Public Function CountUploadRows( _
   ByVal mask As Integer, _
   ByVal threshold As UInt32 _
) As UInt32
```

C#

```
public uint CountUploadRows(
   int mask,
   uint threshold
);
```

パラメータ

- **mask**　ローのチェック対象となるパブリケーションのセット。詳細については、「ULPublicationSchema クラス」 810 ページを参照してください。

Copyright © 2007, iAnywhere Solutions, Inc.

- **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。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「CountUploadRows メソッド」 572 ページ
- 「CountUploadRows(Int32, Int64) メソッド」 573 ページ

CountUploadRows(Int32, Int64) 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。

構文

Visual Basic

```
Public Function CountUploadRows( _
  ByVal mask As Integer, _
  ByVal threshold As Long _
) As Long
```

C#

```
public long CountUploadRows(
  int mask,
  long threshold
);
```

パラメータ

- **mask**　ローのチェック対象となるパブリケーションのセット。詳細については、「ULPublicationSchema クラス」 810 ページを参照してください。
- **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。スレッショルド値は、[0,0x0fffffff] の範囲内であることが必要です。

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。

備考

このメソッドは、System.UInt32 型をネイティブでサポートしていない言語を対象として用意されています。この型がアプリケーションでサポートされている場合は、このメソッドの別のフォームを使用します。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「CountUploadRows メソッド」 572 ページ
- 「CountUploadRows(Int32, UInt32) メソッド」 572 ページ

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。ULCommand のプロパティを使用して、その動作を制御できます。

構文

Visual Basic

Public Function **CreateCommand()** As ULCommand

C#

public ULCommand **CreateCommand();**

戻り値

新しい ULCommand オブジェクト。

備考

ULCommand.CommandText 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 と System.Data.Common.DbConnection.CreateCommand() が厳密に型指定されたものです。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「ULCommand クラス」 509 ページ
- 「CommandText プロパティ」 516 ページ
- IDbConnection.CreateCommand
- DbConnection.CreateCommand

ExecuteTable メソッド

UL 拡張： 直接の操作用に、ULTable のデータベース・テーブルを取り出します。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)。

Copyright © 2007, iAnywhere Solutions, Inc.

ExecuteTable(String) メソッド

UL 拡張： 直接の操作用に、ULTable のデータベース・テーブルを取り出します。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)。

構文

Visual Basic

```
Public Function ExecuteTable( _
  ByVal tableName As String _
) As ULTable
```

C#

```
public ULTable ExecuteTable(
  string  tableName
);
```

パラメータ

- **tableName**　開くテーブルの名前。

戻り値

ULTable オブジェクトとして返されるテーブル。

備考

このメソッドは、ULCommand インスタンスを必要としない 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() と iAnywhere.UltraLite.Table.Open() が置き換えられます)。

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()


// 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;
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
//     t = cmd.ExecuteTable();
// }
```

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「ExecuteTable メソッド」 574 ページ
- 「ExecuteTable(String, String) メソッド」 576 ページ
- 「ExecuteTable(String, String, CommandBehavior) メソッド」 577 ページ
- 「ULTable クラス」 933 ページ
- 「ExecuteTable() メソッド」 541 ページ
- 「ULCommand クラス」 509 ページ

ExecuteTable(String, String) メソッド

UL 拡張： 直接の操作用に、ULTable のデータベース・テーブルを取り出します。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)。

構文

Visual Basic

```
Public Function ExecuteTable( _
   ByVal tableName As String, _
   ByVal indexName As String _
) As ULTable
```

C#

```
public ULTable ExecuteTable(
   string  tableName,
   string  indexName
);
```

パラメータ

- **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返されるテーブル。

備考

このメソッドは、ULCommand インスタンスを必要としない 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() と iAnywhere.UltraLite.Table.Open() が置き換えられます)。

Copyright © 2007, iAnywhere Solutions, Inc.

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 インスタンス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()


// 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();
// }
```

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「ExecuteTable メソッド」 574 ページ
- 「ExecuteTable(String) メソッド」 575 ページ
- 「ExecuteTable(String, String, CommandBehavior) メソッド」 577 ページ
- 「ULTable クラス」 933 ページ
- 「ExecuteTable() メソッド」 541 ページ
- 「ULCommand クラス」 509 ページ

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。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)。

構文

Visual Basic

```
Public Function ExecuteTable( _
   ByVal tableName As String, _
   ByVal indexName As String, _
   ByVal cmdBehavior As CommandBehavior _
) As ULTable
```

C#

Copyright © 2007, iAnywhere Solutions, Inc.

```
public ULTable ExecuteTable(
  string  tableName,
  string  indexName,
  CommandBehavior cmdBehavior
);
```

パラメータ

- **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 フラグのビット処理の組み合わせ。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返されるテーブル。

備考

このメソッドは、ULCommand インスタンスを必要としない ULCommand.ExecuteTable (System.Data.CommandBehavior) 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() と iAnywhere.UltraLite.Table.Open() が置き換えられます)。

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 インスタンス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
' 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()


// C#
ULTable t = conn.ExecuteTable(
    "MyTable", "MyIndex", CommandBehavior.Default
  );
// 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(CommandBehavior.Default);
// }
```


Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「ExecuteTable メソッド」 574 ページ
- 「ExecuteTable(String) メソッド」 575 ページ
- 「ExecuteTable(String, String) メソッド」 576 ページ
- 「ULTable クラス」 933 ページ
- 「ExecuteTable(CommandBehavior) メソッド」 542 ページ
- 「ULCommand クラス」 509 ページ
- CommandBehavior
- CommandBehavior.Default
- CommandBehavior.CloseConnection
- CommandBehavior.SchemaOnly

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。

構文

Visual Basic

```
Public Function GetLastDownloadTime( _
  ByVal mask As Integer _
) As Date
```

C#

```
public DateTime GetLastDownloadTime(
  int mask
);
```

パラメータ

- **mask**　チェックするパブリケーションのマスク。詳細については、「ULPublicationSchema クラス」 810 ページを参照してください。

戻り値

最後のダウンロードのタイムスタンプ。

備考

パラメータ

mask は、単一のパブリケーションを参照するか、データベース全体の最後のダウンロード時刻の特殊定数 ULPublicationSchema.SYNC_ALL_DB であることが必要です。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「ResetLastDownloadTime メソッド」 585 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「SYNC_ALL_DB フィールド」 811 ページ

GetNewUUID メソッド

UL 拡張： 新しい UUID を生成します (System.Guid)。

構文

Visual Basic

Public Function **GetNewUUID()** As Guid

C#

public Guid **GetNewUUID();**

戻り値

System.Guid として返される新しい UUID。

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

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- Guid

GetSchema メソッド

この DbConnection のデータ・ソースに関するスキーマ情報を返します。

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。

構文

Visual Basic

Public Overrides Function **GetSchema()** As DataTable

C#

public override DataTable **GetSchema();**

備考

使用できるメタデータの詳細については、「GetSchema(String, String[]) メソッド」 581 ページを参照してください。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「GetSchema メソッド」 580 ページ

GetSchema(String) メソッド

この ULConnection 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。

構文

Visual Basic

```
Public Overrides Function GetSchema( _
   ByVal collection As String _
) As DataTable
```

C#

```
public override DataTable GetSchema(
   string  collection
);
```

パラメータ

- **collection**　メタデータ・コレクションの名前。指定しない場合、MetaDataCollections が使用されます。

備考

使用できるメタデータの詳細については、「GetSchema(String, String[]) メソッド」 581 ページを参照してください。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「GetSchema メソッド」 580 ページ
- 「ULConnection クラス」 555 ページ

GetSchema(String, String[]) メソッド

この ULConnection 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。

構文

Visual Basic

```
Public Overrides Function GetSchema( _
   ByVal collection As String, _
```

```
  ByVal restrictions As String() _
) As DataTable
```

C#

```
public override DataTable GetSchema(
  string  collection,
  string [] restrictions
);
```

パラメータ

- **collection**　メタデータ・コレクションの名前。指定しない場合、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。

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。
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。
パブリケーション	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。

Copyright © 2007, iAnywhere Solutions, Inc.

コレクション	メタデータ
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 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するには、フィルタ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 コレクションは、Table、TableType、SyncType という 3 つの制限を持ちます。

Table コレクションをフィルタするには、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・テーブルに関する情報を返します。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「GetSchema メソッド」 580 ページ
- 「ULConnection クラス」 555 ページ
- 「ULMetaDataCollectionNames クラス」 764 ページ

GrantConnectTo メソッド

UL 拡張： 指定されたパスワードを持つユーザ ID に、Ultra Light データベースへのアクセスを許可します。

構文

Visual Basic

```
Public Sub GrantConnectTo( _
  ByVal uid As String, _
  ByVal pwd As String _
)
```

C#

```
public void GrantConnectTo(
   string  uid,
   string  pwd
);
```

パラメータ

- **uid**　データベースへのアクセス権を受け取るユーザ ID。ユーザ ID の最大長は 16 文字です。
- **pwd**　ユーザ ID に関連付けられるパスワード。最大長は 16 文字です。

備考

既存のユーザ ID が指定されていれば、この関数を使用してそのユーザ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認証が有効になっていた場合だけです。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「UserID プロパティ」 602 ページ
- 「Password プロパティ」 602 ページ
- 「ConnectionString プロパティ」 560 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。

構文

Visual Basic

```
Public Overrides Sub Open()
```

C#

```
public override void Open();
```

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 クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「ConnectionString プロパティ」 560 ページ
- 「State プロパティ」 566 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Public Sub ResetLastDownloadTime( _
  ByVal mask As Integer _
)
```

C#

```
public void ResetLastDownloadTime(
  int mask
);
```

パラメータ

- **mask** リセットするパブリケーションのセット。詳細については、「ULPublicationSchema クラス」 810 ページを参照してください。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「GetLastDownloadTime メソッド」 579 ページ

RevokeConnectFrom メソッド

UL 拡張： 指定されたユーザ 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。

構文

Visual Basic

```
Public Sub RevokeConnectFrom( _
  ByVal uid As String _
)
```

C#

```
public void RevokeConnectFrom(
  string  uid
);
```

パラメータ

- **uid** データベースへのアクセス権を取り消すユーザ ID。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「GrantConnectTo メソッド」 583 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。

構文

Visual Basic

Public Sub **RollbackPartialDownload()**

C#

public void **RollbackPartialDownload();**

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「KeepPartialDownload プロパティ」 901 ページ
- 「ResumePartialDownload プロパティ」 904 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。

構文

Visual Basic

Public Sub **StartSynchronizationDelete()**

C#

public void **StartSynchronizationDelete();**

備考

この関数が呼び出されると、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 クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「StopSynchronizationDelete メソッド」 587 ページ
- 「Truncate メソッド」 952 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Sub **StopSynchronizationDelete()**

C#

public void **StopSynchronizationDelete();**

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 クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「StartSynchronizationDelete メソッド」 586 ページ

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() 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。

構文

Visual Basic

Public Sub **Synchronize()**

C#

public void **Synchronize();**

備考

結果の詳細なステータスは、この接続の ULConnection.SyncResult でレポートされます。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「Synchronize メソッド」 587 ページ
- 「Synchronize(ULSyncProgressListener) メソッド」 588 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「SyncParms プロパティ」 567 ページ
- 「SyncResult プロパティ」 567 ページ

Synchronize(ULSyncProgressListener) メソッド

UL 拡張： 指定されたリスナに送信されるプログレス・イベントとともに、現在の ULConnection.SyncParms を使用してデータベースを同期します。

構文

Visual Basic

```
Public Sub Synchronize( _
  ByVal listener As ULSyncProgressListener _
)
```

C#

```
public void Synchronize(
  ULSyncProgressListener listener
);
```

パラメータ

- **listener**　同期プログレス・イベントを受信するオブジェクト。

備考

リスナに送信された最後のイベントのステータスは、ULSyncProgressState.STATE_LAST です。

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 で報告されます。

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「Synchronize メソッド」 587 ページ
- 「ULSyncProgressListener インタフェース」 923 ページ
- 「Synchronize() メソッド」 587 ページ
- 「SyncParms プロパティ」 567 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「ULException クラス」 719 ページ
- 「SyncResult プロパティ」 567 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Event **InfoMessage** As ULInfoMessageEventHandler

C#

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 委任を作成し、このイベントにアタッチする必要があります。

例

次のコードでは、情報メッセージのイベント・ハンドラが定義されます。

```
' Visual Basic
Private Sub MyInfoMessageHandler( _
    obj As Object, args As ULInfoMessageEventArgs _
  )
  System.Console.WriteLine( _
    "InfoMesageHandler: " + args.NativeError + ", " _
    + args.Message _
  )
End Sub

// C#
private void MyInfoMessageHandler(
    object obj, ULInfoMessageEventArgs args
  )
{
  System.Console.WriteLine(
    "InfoMesageHandler: " + args.NativeError + ", "
    + args.Message
  );
}
```

次のコードでは、MyInfoMessageHandler が conn という接続に追加されます。

```
' Visual Basic
AddHandler conn.InfoMessage, AddressOf MyInfoMessageHandler

// C#
conn.InfoMessage +=
  new ULInfoMessageEventHandler(MyInfoMessageHandler);
```

イベント・データ

- **NativeError**　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 SQL コードです。
- **Message**　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。
- **Source**　メッセージを返す ADO.NET データ・プロバイダの名前です。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- 「ULInfoMessageEventHandler 委任」 763 ページ

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。

構文

Visual Basic

Public Overrides Event **StateChange** As StateChangeEventHandler

C#

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 委任を作成し、このイベントにアタッチする必要があります。

例

次のコードでは、ステータス変更のイベント・ハンドラが定義されます。

```
' Visual Basic
Private Sub MyStateHandler( _
      obj As Object, args As StateChangeEventArgs _
    )
  System.Console.WriteLine( _
      "StateHandler: " + args.OriginalState + " to " _
      + args.CurrentState _
    )
End Sub

// 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

// C#
conn.StateChange += new StateChangeEventHandler(MyStateHandler);
```


Copyright © 2007, iAnywhere Solutions, Inc.

イベント・データ

- **CurrentState**
- **OriginalState**

参照

- 「ULConnection クラス」 555 ページ
- 「ULConnection メンバ」 555 ページ
- StateChangeEventHandler

Copyright © 2007, iAnywhere Solutions, Inc.

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。頻繁に使用する接続パラメータは、ULConnectionParms オブジェクトの個々のプロパティです。

構文

Visual Basic

Public Class **ULConnectionParms**
  Inherits Component

C#

public class **ULConnectionParms** : Component

備考

ULConnectionParms オブジェクトは、接続を開くパラメータ (ULConnection.Open)、またはデータベースを削除するパラメータ (ULDatabaseManager.DropDatabase) 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、接続パラメータをすべて指定したら、ULConnectionParms.ToString 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) コンストラクタで新しい ULConnection を作成したり、既存の ULConnection オブジェクトの ULConnection.ConnectionString 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 CE 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.DatabaseOnCE = "¥UltraLite¥mydb1.udb"
dbName.DatabaseOnDesktop = "somedir¥mydb2.udb"

// C#
ULConnectionParms dbName = new ULConnectionParms();
dbName.DatabaseOnCE = "¥¥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 CE 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 © 2007, iAnywhere Solutions, Inc.

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。詳細については、「AdditionalParms プロパティ」 595 ページを参照してください。

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 や ULConnectionParms.Password、データベースの ULConnectionParms.EncryptionKey、接続の ULConnectionParms.CacheSize 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 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 DBA で作成されます。このユーザの最初のパスワードは sql です。デフォルトでは、ユーザ 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を無効にするには、ULConnection.RevokeConnectFrom を使用します。ユーザを追加したりユーザのパスワードを変更するには、ULConnection.GrantConnectTo 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 を使用します。

詳細については、「Ultra Light の接続文字列パラメータのリファレンス」 『Ultra Light - データベース管理とリファレンス』を参照してください。

参照

- 「ULConnectionParms メンバ」 594 ページ
- 「Open メソッド」 584 ページ
- 「DropDatabase メソッド」 660 ページ
- 「ToString メソッド」 603 ページ
- 「ULConnection クラス」 555 ページ
- 「ULConnection(String) コンストラクタ」 559 ページ
- 「ConnectionString プロパティ」 560 ページ
- 「UserID プロパティ」 602 ページ
- 「Password プロパティ」 602 ページ
- 「EncryptionKey プロパティ」 601 ページ
- 「CacheSize プロパティ」 599 ページ
- 「ConnectionName プロパティ」 599 ページ
- 「RevokeConnectFrom メソッド」 585 ページ
- 「GrantConnectTo メソッド」 583 ページ
- 「ChangeEncryptionKey メソッド」 570 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULConnectionParms メンバ

パブリック・コンストラクタ

メンバ名	説明
ULConnectionParms コンストラクタ	ULConnectionParms インスタンスを、そのデフォルト値で初期化します。

パブリック・プロパティ

メンバ名	説明
AdditionalParms プロパティ	「名前=値」の組み合わせをセミコロンで区切ったリストで、追加のパラメータを指定します。ここで指定されるのは、あまり一般的に使用されないパラメータです。
CacheSize プロパティ	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。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。
DatabaseOnCE プロパティ	Windows CE 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。
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。
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。
Password プロパティ	認証済みユーザのパスワードを指定します。
UserID プロパティ	データベースで認証されるユーザを指定します。

パブリック・メソッド

メンバ名	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。

パブリック・イベント

メンバ名	説明
UnusedEvent イベント	使われていません。

参照

- 「ULConnectionParms クラス」 592 ページ
- 「Open メソッド」 584 ページ
- 「DropDatabase メソッド」 660 ページ
- 「ToString メソッド」 603 ページ
- 「ULConnection クラス」 555 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULConnection(String) コンストラクタ」 559 ページ
- 「ConnectionString プロパティ」 560 ページ
- 「UserID プロパティ」 602 ページ
- 「Password プロパティ」 602 ページ
- 「EncryptionKey プロパティ」 601 ページ
- 「CacheSize プロパティ」 599 ページ
- 「ConnectionName プロパティ」 599 ページ
- 「RevokeConnectFrom メソッド」 585 ページ
- 「GrantConnectTo メソッド」 583 ページ
- 「ChangeEncryptionKey メソッド」 570 ページ

ULConnectionParms コンストラクタ

ULConnectionParms インスタンスを、そのデフォルト値で初期化します。

構文

Visual Basic

Public Sub **New()**

C#

public **ULConnectionParms();**

参照

- 「ULConnectionParms クラス」 592 ページ
- 「ULConnectionParms メンバ」 594 ページ

AdditionalParms プロパティ

「名前=値」の組み合わせをセミコロンで区切ったリストで、追加のパラメータを指定します。ここで指定されるのは、あまり一般的に使用されないパラメータです。

構文

Visual Basic

Public Property **AdditionalParms** As String

C#

public string **AdditionalParms** { get; set; }

プロパティ値

「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。keyword=value リストの値は、ULConnection.ConnectionString のルールに従う必要があります。デフォルト値は null 参照 (Visual Basic の Nothing) です。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

ページ・サイズや予約サイズのパラメータの値は、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。

追加のパラメータを次に示します。

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 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 (DatabaseOnCE または DatabaseOnDesktop) 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。 このパラメータが使用されるのは、ULConnection.Open を使用して接続を開いたときだけです。

Copyright © 2007, iAnywhere Solutions, Inc.

キーワード	説明
ordered_table_scans	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。 リリース 10.0.1 では、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、ordered_table_scans パラメータを 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。 ordered_table_scans 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。 次のパラメータ文字列を使用すると、Ultra Light は以前のリリースと同じように動作します。 createParms.**AdditionalParms = "ordered_table_scans=yes"** デフォルトは 0 (false、off、no) です。 このパラメータが使用されるのは、ULConnection.Open を使用して接続を開いたときだけです。 詳細については、「Ultra Light の接続文字列パラメータのリファレンス」『Ultra Light - データベース管理とリファレンス』を参照してください。

Copyright © 2007, iAnywhere Solutions, Inc.

キーワード	説明
reserve_size	Ultra Light 継続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、永続ストレージ・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、継続保管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ヘッドとデータの圧縮を考慮してください。テスト・データを入れてデータベースを実行し、継続保管ファイルのサイズを確認することをおすすめします。 reserve_size パラメータは、起動時に継続保管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。 次のパラメータ文字列では、起動時に継続保管ファイルのサイズが最低 2 MB 確保されます。 createParms.**AdditionalParms = "reserve_size=2m"** このパラメータが使用されるのは、ULConnection.Open を使用して接続を開いたときだけです。
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。 Ultra Light.NET で Ultra Light エンジンを使用する方法の詳細については、「RuntimeType プロパティ」 658 ページを参照してください。

参照

- 「ULConnectionParms クラス」 592 ページ
- 「ULConnectionParms メンバ」 594 ページ
- 「ConnectionString プロパティ」 560 ページ
- 「DatabaseOnCE プロパティ」 600 ページ
- 「DatabaseOnDesktop プロパティ」 600 ページ
- 「Open メソッド」 584 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

CacheSize プロパティ

キャッシュのサイズを指定します。

構文

Visual Basic

Public Property **CacheSize** As String

C#

public string **CacheSize** { get; set; }

プロパティ値

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。

備考

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。

参照

- 「ULConnectionParms クラス」 592 ページ
- 「ULConnectionParms メンバ」 594 ページ

ConnectionName プロパティ

接続の名前を指定します。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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **ConnectionName** As String

C#

public string **ConnectionName** { get; set; }

プロパティ値

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。

参照

- 「ULConnectionParms クラス」 592 ページ
- 「ULConnectionParms メンバ」 594 ページ

DatabaseOnCE プロパティ

Windows CE 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。

構文

Visual Basic

Public Property **DatabaseOnCE** As String

C#

public string **DatabaseOnCE** { get; set; }

プロパティ値

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。

参照

- 「ULConnectionParms クラス」 592 ページ
- 「ULConnectionParms メンバ」 594 ページ

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。

 Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **DatabaseOnDesktop** As String

C#

public string **DatabaseOnDesktop** { get; set; }

プロパティ値

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。

参照

- 「ULConnectionParms クラス」 592 ページ
- 「ULConnectionParms メンバ」 594 ページ

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。

構文

Visual Basic

Public Property **EncryptionKey** As String

C#

public string **EncryptionKey** { get; set; }

プロパティ値

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。

備考

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

- 「ULConnectionParms クラス」 592 ページ
- 「ULConnectionParms メンバ」 594 ページ
- 「ChangeEncryptionKey メソッド」 570 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

Password プロパティ

認証済みユーザのパスワードを指定します。

構文

Visual Basic

Public Property **Password** As String

C#

public string **Password** { get; set; }

プロパティ値

データベースのユーザ 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。

備考

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 ID DBA のパスワードは SQL に設定されます。

参照

- 「ULConnectionParms クラス」 592 ページ
- 「ULConnectionParms メンバ」 594 ページ
- 「UserID プロパティ」 602 ページ

UserID プロパティ

データベースで認証されるユーザを指定します。

構文

Visual Basic

Public Property **UserID** As String

C#

public string **UserID** { get; set; }

プロパティ値

データベースのユーザ 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。

備考

ユーザ ID の大文字と小文字は区別されません。

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で作成されます。

Copyright © 2007, iAnywhere Solutions, Inc.

ユーザ ID とパスワードを両方とも入力しないと、ユーザ名 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、ユーザ名 DBA のパスワードを変更するか、新しいユーザを作成して (ULConnection.GrantConnectTo を使用)、DBA ユーザを削除します (ULConnection.RevokeConnectFrom を使用)。

参照

- 「ULConnectionParms クラス」 592 ページ
- 「ULConnectionParms メンバ」 594 ページ
- 「Password プロパティ」 602 ページ
- 「GrantConnectTo メソッド」 583 ページ
- 「RevokeConnectFrom メソッド」 585 ページ

ToString メソッド

このインスタンスの文字列表現を返します。

構文

Visual Basic

Public Overrides Function **ToString()** As String

C#

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。

参照

- 「ULConnectionParms クラス」 592 ページ
- 「ULConnectionParms メンバ」 594 ページ

UnusedEvent イベント

使われていません。

構文

Visual Basic

Public Event **UnusedEvent** As ULConnectionParms.UnusedEventHandler

C#

public event ULConnectionParms.UnusedEventHandler **UnusedEvent** ;

Copyright © 2007, iAnywhere Solutions, Inc.

備考

この public Event は、Visual Basic .NET プロジェクトでこのクラスを組み込む場合の Visual Studio .NET のバグを修正するために用意されています。実用的な用途はありません。

参照

- 「ULConnectionParms クラス」 592 ページ
- 「ULConnectionParms メンバ」 594 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

ULConnectionParms.UnusedEventHandler 委任

UL 拡張： 使われていません。

構文

Visual Basic

```
Public Delegate Sub ULConnectionParms.UnusedEventHandler( _
  ByVal sender As Object, _
  ByVal args As EventArgs _
)
```

C#

```
public delegate void ULConnectionParms.UnusedEventHandler(
  object sender,
  EventArgs args
);
```

備考

この public Delegate は、Visual Basic .NET プロジェクトでこのクラスを組み込む場合の Visual Studio .NET のバグを修正するために用意されています。実用的な用途はありません。

Copyright © 2007, iAnywhere Solutions, Inc.

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。頻繁に使用する接続パラメータは、ULConnectionStringBuilder オブジェクトの個々のプロパティです。このクラスは継承できません。

構文

Visual Basic

Public NotInheritable Class **ULConnectionStringBuilder**
  Inherits DbConnectionStringBuilder

C#

public sealed class **ULConnectionStringBuilder** : DbConnectionstring Builder

備考

制限： SAConnectionStringBuilder 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、接続を開くパラメータ (ULConnection.Open)、またはデータベースを削除するパラメータ (ULDatabaseManager.DropDatabase) 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) コンストラクタで新しい ULConnection を作成したり、既存の ULConnection オブジェクトの ULConnection.ConnectionString 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 CE 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.DatabaseOnCE = "¥UltraLite¥mydb1.udb"
dbName.DatabaseOnDesktop = "somedir¥mydb2.udb"

// C#
ULConnectionStringBuilder dbName = new ULConnectionStringBuilder();
dbName.DatabaseOnCE = "¥¥UltraLite¥¥mydb1.udb";
dbName.DatabaseOnDesktop = @"somedir¥mydb2.udb";
```


Copyright © 2007, iAnywhere Solutions, Inc.

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 CE 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。詳細については、「DatabaseName プロパティ」 613 ページを参照してください。

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 や ULConnectionStringBuilder.Password、データベースの ULConnectionStringBuilder.DatabaseKey、接続の ULConnectionStringBuilder.CacheSize 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 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 DBA で作成されます。このユーザの最初のパスワードは sql です。デフォルトでは、ユーザ 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を無効にするには、ULConnection.RevokeConnectFrom を使用します。ユーザを追加したりユーザのパスワードを変更するには、ULConnection.GrantConnectTo 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 を使用します。

詳細については、「Ultra Light の接続文字列パラメータのリファレンス」 『Ultra Light - データベース管理とリファレンス』を参照してください。

参照

- 「ULConnectionStringBuilder メンバ」 608 ページ
- 「Open メソッド」 584 ページ
- 「DropDatabase メソッド」 660 ページ
- DbConnectionStringBuilder.ConnectionString
- 「ULConnection クラス」 555 ページ
- 「ULConnection(String) コンストラクタ」 559 ページ
- 「ConnectionString プロパティ」 560 ページ
- 「DatabaseName プロパティ」 613 ページ
- 「UserID プロパティ」 619 ページ
- 「Password プロパティ」 617 ページ
- 「DatabaseKey プロパティ」 612 ページ
- 「CacheSize プロパティ」 611 ページ
- 「ConnectionName プロパティ」 612 ページ
- 「RevokeConnectFrom メソッド」 585 ページ
- 「GrantConnectTo メソッド」 583 ページ
- 「ChangeEncryptionKey メソッド」 570 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULConnectionStringBuilder メンバ

パブリック・コンストラクタ

メンバ名	説明
ULConnectionStringBuilder コンストラクタ	「ULConnectionStringBuilder クラス」 606 ページの新しいインスタンスを初期化します。

パブリック・プロパティ

メンバ名	説明
BrowsableConnectionString (DbConnectionStringBuilder から継承)	
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。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。
ConnectionString (DbConnectionStringBuilder から継承)	
Count (DbConnectionStringBuilder から継承)	
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。
DatabaseOnCE プロパティ	**UL 拡張：** Windows CE 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。
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。
IsFixedSize (DbConnectionStringBuilder から継承)	
IsReadOnly (DbConnectionStringBuilder から継承)	
Item 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。
Keys (DbConnectionStringBuilder から継承)	

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	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。
UserID プロパティ	データベースで認証されるユーザを指定します。
Values (DbConnectionStringBuilder から継承)	

パブリック・メソッド

メンバ名	説明
Add (DbConnectionStringBuilder から継承)	
Clear (DbConnectionStringBuilder から継承)	
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。
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 インスタンスから削除します。
ShouldSerialize (DbConnectionStringBuilder から継承)	
ToString (DbConnectionStringBuilder から継承)	
TryGetValue メソッド	入力されたキーに対応する値を、この ULConnectionStringBuilder から取り出します。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「Open メソッド」 584 ページ
- 「DropDatabase メソッド」 660 ページ
- DbConnectionStringBuilder.ConnectionString
- 「ULConnection クラス」 555 ページ
- 「ULConnection(String) コンストラクタ」 559 ページ
- 「ConnectionString プロパティ」 560 ページ
- 「DatabaseName プロパティ」 613 ページ
- 「UserID プロパティ」 619 ページ
- 「Password プロパティ」 617 ページ
- 「DatabaseKey プロパティ」 612 ページ
- 「CacheSize プロパティ」 611 ページ
- 「ConnectionName プロパティ」 612 ページ
- 「RevokeConnectFrom メソッド」 585 ページ
- 「GrantConnectTo メソッド」 583 ページ
- 「ChangeEncryptionKey メソッド」 570 ページ

ULConnectionStringBuilder コンストラクタ

「ULConnectionStringBuilder クラス」 606 ページの新しいインスタンスを初期化します。

ULConnectionStringBuilder() コンストラクタ

ULConnectionStringBuilder インスタンスを、そのデフォルト値で初期化します。

構文

Visual Basic

Public Sub **New()**

C#

public **ULConnectionStringBuilder();**

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ
- 「ULConnectionStringBuilder コンストラクタ」 610 ページ

ULConnectionStringBuilder(String) コンストラクタ

ULConnectionStringBuilder インスタンスを、指定された接続文字列で初期化します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

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

C#

```
public  ULConnectionStringBuilder(
  string  connectionString
);
```

パラメータ

- **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。

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ
- 「ULConnectionStringBuilder コンストラクタ」 610 ページ

CacheSize プロパティ

UL 拡張： キャッシュのサイズを指定します。

構文

Visual Basic

```
Public Property CacheSize As String
```

C#

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

プロパティ値

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。

備考

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。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ

ConnectionName プロパティ

接続の名前を指定します。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。

構文

Visual Basic

Public Property **ConnectionName** As String

C#

public string **ConnectionName** { get; set; }

プロパティ値

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。

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ

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。

構文

Visual Basic

Public Property **DatabaseKey** As String

C#

public string **DatabaseKey** { get; set; }

 Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

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。

備考

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

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ
- 「ChangeEncryptionKey メソッド」 570 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。

構文

Visual Basic

```
Public Property DatabaseName As String
```

C#

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

プロパティ値

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。

備考

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 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 (DatabaseOnCE または DatabaseOnDesktop) 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 クラス」 606 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULConnectionStringBuilder メンバ」 608 ページ
- 「DatabaseOnCE プロパティ」 614 ページ
- 「DatabaseOnDesktop プロパティ」 614 ページ

DatabaseOnCE プロパティ

UL 拡張： Windows CE 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。

構文

Visual Basic

Public Property **DatabaseOnCE** As String

C#

public string **DatabaseOnCE** { get; set; }

プロパティ値

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。

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 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。

構文

Visual Basic

Public Property **DatabaseOnDesktop** As String

C#

public string **DatabaseOnDesktop** { get; set; }

プロパティ値

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。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ

Item 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。

構文

Visual Basic

```
Public Overrides Default Property Item ( _
   ByVal keyword As String _
) As Object
```

C#

```
public override object this [
   string  keyword
] { get; set; }
```

パラメータ

- **keyword**　接続キーワードの名前。

プロパティ値

指定された接続キーワードの値を表すオブジェクト。

備考

接続キーワードと、ULConnectionStringBuilder 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	「CacheSize プロパティ」 611 ページ
ce_file	「DatabaseOnCE プロパティ」 614 ページ
con	「ConnectionName プロパティ」 612 ページ
dbkey	「DatabaseKey プロパティ」 612 ページ
dbn	「DatabaseName プロパティ」 613 ページ
nt_file	「DatabaseOnDesktop プロパティ」 614 ページ
pwd	「Password プロパティ」 617 ページ
reserve_size	「ReserveSize プロパティ」 617 ページ
start	「StartLine プロパティ」 618 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

キーワード	対応するプロパティ
uid	「UserID プロパティ」 619 ページ

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ
- 「CacheSize プロパティ」 611 ページ
- 「DatabaseOnCE プロパティ」 614 ページ
- 「ConnectionName プロパティ」 612 ページ
- 「DatabaseKey プロパティ」 612 ページ
- 「DatabaseName プロパティ」 613 ページ
- 「DatabaseOnDesktop プロパティ」 614 ページ
- 「Password プロパティ」 617 ページ
- 「ReserveSize プロパティ」 617 ページ
- 「StartLine プロパティ」 618 ページ
- 「UserID プロパティ」 619 ページ

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。

構文

Visual Basic

Public Property **OrderedTableScans** As String

C#

public string **OrderedTableScans** { get; set; }

プロパティ値

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。

備考

リリース 10.0.1 では、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

 Copyright © 2007, iAnywhere Solutions, Inc.

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 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。

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ

Password プロパティ

認証済みユーザのパスワードを指定します。

構文

Visual Basic

Public Property **Password** As String

C#

public string **Password** { get; set; }

プロパティ値

データベースのユーザ 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。

備考

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 ID DBA のパスワードは SQL に設定されます。

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ
- 「UserID プロパティ」 619 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。

構文

Visual Basic

Public Property **ReserveSize** As String

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public string **ReserveSize** { get; set; }

プロパティ値

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。

備考

予約サイズ・パラメータの値は、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、永続ストレージ・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、継続保管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ヘッドとデータの圧縮を考慮してください。テスト・データを入れてデータベースを実行し、継続保管ファイルのサイズを確認することをおすすめします。

reserve_size パラメータは、起動時に継続保管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。

次のパラメータ文字列では、起動時に継続保管ファイルのサイズが最低 2 MB 確保されます。

```
connParms.ReserveSize = "2m"
```

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ

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。

構文

Visual Basic

Public Property **StartLine** As String

C#

public string **StartLine** { get; set; }

 Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

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。

備考

現在実行中でないエンジンに接続する場合は、StartLine (START) 接続パラメータだけを指定します。

Ultra Light.NET で Ultra Light エンジンを使用する方法の詳細については、「RuntimeType プロパティ」 658 ページを参照してください。

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ
- 「RuntimeType プロパティ」 658 ページ

UserID プロパティ

データベースで認証されるユーザを指定します。

構文

Visual Basic

```
Public Property UserID As String
```

C#

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

プロパティ値

データベースのユーザ 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。

備考

ユーザ ID の大文字と小文字は区別されません。

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で作成されます。

ユーザ ID とパスワードを両方とも入力しないと、ユーザ名 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、ユーザ名 DBA のパスワードを変更するか、新しいユーザを作成して (ULConnection.GrantConnectTo を使用)、DBA ユーザを削除します (ULConnection.RevokeConnectFrom を使用)。

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ
- 「Password プロパティ」 617 ページ
- 「GrantConnectTo メソッド」 583 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「RevokeConnectFrom メソッド」 585 ページ

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。

構文

Visual Basic

```
Public Overrides Function ContainsKey( _
   ByVal keyword As String _
) As Boolean
```

C#

```
public override bool ContainsKey(
   string  keyword
);
```

パラメータ

- **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。

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ

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。

構文

Visual Basic

```
Public Overrides Function EquivalentTo( _
   ByVal connectionStringBuilder As DbConnectionStringBuilder _
) As Boolean
```

C#

```
public override bool EquivalentTo(
   DbConnectionstring Builder connectionStringBuilder
);
```

 Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **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 を返します。

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ
- 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。

構文

Visual Basic

```
Public Shared Function GetShortName( _
  ByVal keyword As String _
) As String
```

C#

```
public static string  GetShortName(
  string  keyword
);
```

パラメータ

- **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。

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ

Remove メソッド

指定されたキーが設定されたエントリを ULConnectionStringBuilder インスタンスから削除します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Overrides Function Remove( _
  ByVal keyword As String _
) As Boolean
```

C#

```
public override bool Remove(
  string  keyword
);
```

パラメータ

- **keyword**　接続キーワードの名前。

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。

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ

TryGetValue メソッド

入力されたキーに対応する値を、この ULConnectionStringBuilder から取り出します。

構文

Visual Basic

```
Public Overrides Function TryGetValue( _
  ByVal keyword As String, _
  ByVal value As Object _
) As Boolean
```

C#

```
public override bool TryGetValue(
  string  keyword,
  object value
);
```

パラメータ

- **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。

Copyright © 2007, iAnywhere Solutions, Inc.

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 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、値パラメータに null 値 (Visual Basic の Nothing) が設定されます。

参照

- 「ULConnectionStringBuilder クラス」 606 ページ
- 「ULConnectionStringBuilder メンバ」 608 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Class **ULCreateParms**

C#

public class **ULCreateParms**

備考

ULCreateParms オブジェクトは、データベースを作成するときのパラメータを使用するために使用します (ULDatabaseManager.CreateDatabase(string,byte[],string))。

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、作成パラメータをすべて指定したら、ULCreateParms.ToString を使用して作成パラメータ文字列を作成します。結果として返される文字列は ULDatabaseManager.CreateDatabase(string,byte[],string) メソッドの createParms パラメータとして使用できます。

詳細については、「Ultra Light の接続文字列パラメータのリファレンス」 『Ultra Light - データベース管理とリファレンス』を参照してください。

例

次のコードでは、Windows CE 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.DatabaseOnCE = "¥UltraLite¥MyDatabase.udb"
' Assumes file coll_1250LATIN2.vb is
' also compiled in the current project
ULConnection.DatabaseManager.CreateDatabase( _
    openParms.ToString(), _
    iAnywhere.UltraLite.Collations.Collation_1250LATIN2.Data, _
    createParms.ToString() _
  )
Dim conn As ULConnection = _
  New ULConnection( openParms.ToString() )
conn.Open()


// C#
ULCreateParms createParms = new ULCreateParms();
createParms.CaseSensitive = true;
```

 Copyright © 2007, iAnywhere Solutions, Inc.

```
createParms.UTF8Encoding = true;
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnCE = @"¥UltraLite¥MyDatabase.udb";
// Assumes file coll_1250LATIN2.vb is
// also compiled in the current project
ULConnection.DatabaseManager.CreateDatabase(
    openParms.ToString(),
    iAnywhere.UltraLite.Collations.Collation_1250LATIN2.Data,
    createParms.ToString()
  );
ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```

参照

- 「ULCreateParms メンバ」 625 ページ
- 「GetDatabaseProperty メソッド」 669 ページ
- 「CreateDatabase メソッド」 659 ページ
- 「ToString メソッド」 634 ページ

ULCreateParms メンバ

パブリック・コンストラクタ

メンバ名	説明
ULCreateParms コンストラクタ	ULCreateParms インスタンスを、そのデフォルト値で初期化します。

パブリック・プロパティ

メンバ名	説明
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
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
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
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
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)
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
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
Obfuscate プロパティ	新しいデータベースで難読化 (単純暗号化) を使用するかどうか

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
PageSize プロパティ	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)
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
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 PRECISION にトランケートされるときの小数点以下の最大桁数
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
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
TimestampIncrement プロパティ	2 つのユニークなタイムスタンプ間のナノ秒 (1,000,000 分の 1 秒) 単位の最小差
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)

パブリック・メソッド

メンバ名	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。

参照

- 「ULCreateParms クラス」 624 ページ
- 「GetDatabaseProperty メソッド」 669 ページ
- 「CreateDatabase メソッド」 659 ページ
- 「ToString メソッド」 634 ページ

ULCreateParms コンストラクタ

ULCreateParms インスタンスを、そのデフォルト値で初期化します。

構文

Visual Basic

Public Sub **New()**

C#

public **ULCreateParms();**

参照

- 「ULCreateParms クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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

構文

Visual Basic

Public Property **CaseSensitive** As Boolean

C#

public bool **CaseSensitive** { get; set; }

プロパティ値

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。

備考

CaseSensitive 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 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。

参照

- 「ULCreateParms クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ

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

構文

Visual Basic

Public Property **ChecksumLevel** As Integer

C#

public int **ChecksumLevel** { get; set; }

プロパティ値

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。

参照

- 「ULCreateParms クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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

構文

Visual Basic

Public Property **DateFormat** As String

C#

public string **DateFormat** { get; set; }

プロパティ値

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。

参照

- 「ULCreateParms クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ

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

構文

Visual Basic

Public Property **DateOrder** As ULDateOrder

C#

public ULDateOrder **DateOrder** { get; set; }

プロパティ値

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

- 「ULCreateParms クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ
- 「ULDateOrder enumeration」 714 ページ

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)

 Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **FIPS** As Boolean

C#

public bool **FIPS** { get; set; }

プロパティ値

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。

備考

新しいデータベースが作成されたときは、接続パラメータ EncryptionKey の値を指定して暗号化を有効にする必要があります。FIPS を true に設定している場合で、暗号化キーを指定しないと、ULDatabaseManager.CreateDatabase(string,byte[],string) 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

- 「ULCreateParms クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ
- 「EncryptionKey プロパティ」 601 ページ
- 「CreateDatabase メソッド」 659 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

構文

Visual Basic

Public Property **MaxHashSize** As Integer

C#

public int **MaxHashSize** { get; set; }

プロパティ値

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。

参照

- 「ULCreateParms クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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

構文

Visual Basic

Public Property **NearestCentury** As Integer

C#

public int **NearestCentury** { get; set; }

プロパティ値

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。

参照

- 「ULCreateParms クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ

Obfuscate プロパティ

新しいデータベースで難読化 (単純暗号化) を使用するかどうか

構文

Visual Basic

Public Property **Obfuscate** As Boolean

C#

public bool **Obfuscate** { get; set; }

プロパティ値

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。

備考

FIPS 暗号化が有効になっている場合 (ULCreateParms.FIPS) 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接続パラメータ EncryptionKey (DBKEY) に値を指定すると、暗号化キーは無視されます。

参照

- 「ULCreateParms クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ
- 「FIPS プロパティ」 628 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

PageSize プロパティ

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)

構文

Visual Basic

Public Property **PageSize** As Integer

C#

public int **PageSize** { get; set; }

プロパティ値

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。

参照

- 「ULCreateParms クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ

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

構文

Visual Basic

Public Property **Precision** As Integer

C#

public int **Precision** { get; set; }

プロパティ値

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 クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ
- 「Scale プロパティ」 631 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 PRECISION にトランケートされるときの小数点以下の最大桁数

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **Scale** As Integer

C#

public int **Scale** { get; set; }

プロパティ値

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。

備考

Scale は、Precision よりも小さいか等しいことが必要です。Scale が Precision よりも大きいと、データベースの作成時にエラーが発生します。

参照

- 「ULCreateParms クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ

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

構文

Visual Basic

Public Property **TimeFormat** As String

C#

public string **TimeFormat** { get; set; }

プロパティ値

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。

参照

- 「ULCreateParms クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ

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

 Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **TimestampFormat** As String

C#

public string **TimestampFormat** { get; set; }

プロパティ値

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。

参照

- 「ULCreateParms クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ

TimestampIncrement プロパティ

2 つのユニークなタイムスタンプ間のナノ秒 (1,000,000 分の 1 秒) 単位の最小差

構文

Visual Basic

Public Property **TimestampIncrement** As Integer

C#

public int **TimestampIncrement** { get; set; }

プロパティ値

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。

参照

- 「ULCreateParms クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ

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)

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **UTF8Encoding** As Boolean

C#

public bool **UTF8Encoding** { get; set; }

プロパティ値

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。

備考

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。

Symbian OS デバイスで使用するデータベースの場合、UTF8Encoding を true に設定する必要があります。

Palm OS デバイスで使用するデータベースの場合、UTF8Encoding を false のままにする必要があります。

参照

- 「ULCreateParms クラス」 624 ページ
- 「ULCreateParms メンバ」 625 ページ

ToString メソッド

このインスタンスの文字列表現を返します。

構文

Visual Basic

Public Overrides Function **ToString()** As String

C#

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。

参照

- 「ULCreateParms クラス」 624 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULCreateParms メンバ」 625 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULCursorSchema クラス

UL 拡張： Ultra Light.NET カーソルのスキーマを示します。これは抽象クラスであるため、インスタンス化はできません。

構文

Visual Basic

MustInherit Public Class **ULCursorSchema**

C#

public abstract class **ULCursorSchema**

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。

iAnywhere.UltraLite ネームスペースから移行するユーザへの注意： 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

- 「ULCursorSchema メンバ」 636 ページ
- 「ULTableSchema クラス」 954 ページ
- 「ULResultSetSchema クラス」 839 ページ

ULCursorSchema メンバ

パブリック・プロパティ

メンバ名	説明
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。

パブリック・メソッド

メンバ名	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 (SQL NUMERIC 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。
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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
GetColumnScale メソッド	カラムが数値カラム (SQL NUMERIC 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 (SQL BINARY 型または CHAR 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。
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 のカラムのスキーマが記述された System.Data.DataTable を返します。

参照

- 「ULCursorSchema クラス」 636 ページ
- 「ULTableSchema クラス」 954 ページ
- 「ULResultSetSchema クラス」 839 ページ

ColumnCount プロパティ

このカーソル内のカラム数を返します。

構文

Visual Basic

Public Readonly Property **ColumnCount** As Short

C#

public short **ColumnCount** { get;}

プロパティ値

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。

備考

カラム ID の範囲は、0 ー 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。

参照

- 「ULCursorSchema クラス」 636 ページ
- 「ULCursorSchema メンバ」 636 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Readonly Property **IsOpen** As Boolean

C#

public bool **IsOpen** { get;}

プロパティ値

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。

参照

- 「ULCursorSchema クラス」 636 ページ
- 「ULCursorSchema メンバ」 636 ページ

Name プロパティ

カーソルの名前を返します。

構文

Visual Basic

Public Readonly Property **Name** As String

C#

public string **Name** { get;}

プロパティ値

文字列として返されるカーソル名。

参照

- 「ULCursorSchema クラス」 636 ページ
- 「ULCursorSchema メンバ」 636 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。

構文

Visual Basic

 Copyright © 2007, iAnywhere Solutions, Inc.

```
Public Function GetColumnID( _
  ByVal name As String _
) As Short
```

C#

```
public short GetColumnID(
  string  name
);
```

パラメータ

- **name**　カラムの名前。

戻り値

指定されたカラムのカラム ID。

備考

カラム ID の範囲は、0 ー 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 クラス」 636 ページ
- 「ULCursorSchema メンバ」 636 ページ
- 「ColumnCount プロパティ」 637 ページ
- 「ColumnCount プロパティ」 637 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。

構文

Visual Basic

```
Public Function GetColumnName( _
  ByVal columnID As Integer _
) As String
```

C#

```
public string  GetColumnName(
  int columnID
);
```

Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **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。

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 クラス」 636 ページ
- 「ULCursorSchema メンバ」 636 ページ
- 「ColumnCount プロパティ」 637 ページ
- 「ColumnCount プロパティ」 637 ページ

GetColumnPrecision メソッド

カラムが数値カラム (SQL NUMERIC 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。

構文

Visual Basic

```
Public Function GetColumnPrecision( _
   ByVal columnID As Integer _
) As Integer
```

C#

```
public int GetColumnPrecision(
   int columnID
);
```

パラメータ

- **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。

 Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULCursorSchema クラス」 636 ページ
- 「ULCursorSchema メンバ」 636 ページ
- 「GetColumnULDbType メソッド」 643 ページ
- 「ColumnCount プロパティ」 637 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。

構文

Visual Basic

```
Public Function GetColumnSQLName( _
  ByVal columnID As Integer _
) As String
```

C#

```
public string  GetColumnSQLName(
  int columnID
);
```

パラメータ

- **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。

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、非エイリアス・カラム、非計算カラムの GetColumnName とは異なります。GetColumnSQLName は、常にカラムの名前を返します (プレフィクスとしてテーブル名は付きません)。この動作は、他の ADO.NET プロバイダ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 クラス」 636 ページ
- 「ULCursorSchema メンバ」 636 ページ
- 「ColumnCount プロパティ」 637 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「GetColumnName メソッド」 639 ページ
- 「ColumnCount プロパティ」 637 ページ

GetColumnScale メソッド

カラムが数値カラム (SQL NUMERIC 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。

構文

Visual Basic

```
Public Function GetColumnScale( _
   ByVal columnID As Integer _
) As Integer
```

C#

```
public int GetColumnScale(
   int columnID
);
```

パラメータ

- **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。

参照

- 「ULCursorSchema クラス」 636 ページ
- 「ULCursorSchema メンバ」 636 ページ
- 「GetColumnULDbType メソッド」 643 ページ
- 「ColumnCount プロパティ」 637 ページ

GetColumnSize メソッド

カラムがサイズ指定されたカラム (SQL BINARY 型または CHAR 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。

構文

Visual Basic

```
Public Function GetColumnSize( _
   ByVal columnID As Integer _
) As Integer
```

C#

```
public int GetColumnSize(
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
  int columnID
);
```

パラメータ

- **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。

参照

- 「ULCursorSchema クラス」 636 ページ
- 「ULCursorSchema メンバ」 636 ページ
- 「GetColumnULDbType メソッド」 643 ページ
- 「ColumnCount プロパティ」 637 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。

構文

Visual Basic

```
Public Function GetColumnULDbType( _
  ByVal columnID As Integer _
) As ULDbType
```

C#

```
public ULDbType GetColumnULDbType(
  int columnID
);
```

パラメータ

- **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。

参照

- 「ULCursorSchema クラス」 636 ページ
- 「ULCursorSchema メンバ」 636 ページ
- 「ColumnCount プロパティ」 637 ページ
- 「ColumnCount プロパティ」 637 ページ
- 「ULDbType 列挙」 715 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

GetSchemaTable メソッド

ULDataReader のカラムのスキーマが記述された System.Data.DataTable を返します。

構文

Visual Basic

Public Function **GetSchemaTable()** As DataTable

C#

public DataTable **GetSchemaTable();**

戻り値

カラムのスキーマが記述された System.Data.DataTable。

備考

詳細については、「GetSchemaTable メソッド」 701 ページを参照してください。

参照

- 「ULCursorSchema クラス」 636 ページ
- 「ULCursorSchema メンバ」 636 ページ
- DataTable
- 「ULDataReader クラス」 676 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

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示します。このクラスは継承できません。

構文

Visual Basic

Public NotInheritable Class **ULDataAdapter**
  Inherits DbDataAdapter

C#

public sealed class **ULDataAdapter** : DbDataAdapter

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 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 または ULTable を使用すると、より効率的にデータにアクセスできます。

継承： System.Data.Common.DbDataAdapter

実装： System.Data.IDbDataAdapter、System.Data.IDataAdapter、System.IDisposable

参照

- 「ULDataAdapter メンバ」 645 ページ
- DataSet
- 「ULDataReader クラス」 676 ページ
- 「ULTable クラス」 933 ページ
- DbDataAdapter
- IDbDataAdapter
- IDataAdapter
- IDisposable

ULDataAdapter メンバ

パブリック・コンストラクタ

メンバ名	説明
ULDataAdapter コンストラクタ	「ULDataAdapter クラス」 645 ページの新しいインスタンスを初期化します。

Copyright © 2007, iAnywhere Solutions, Inc.

パブリック・プロパティ

メンバ名	説明
AcceptChangesDuringFill (DataAdapter から継承)	
AcceptChangesDuringUpdate (DataAdapter から継承)	
ContinueUpdateOnError (DataAdapter から継承)	
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 (System.Data.DataSet) 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。
FillLoadOption (DataAdapter から継承)	
InsertCommand プロパティ	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、DbDataAdapter.Update (System.Data.DataSet) 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。
MissingMappingAction (DataAdapter から継承)	
MissingSchemaAction (DataAdapter から継承)	
ReturnProviderSpecificTypes (DataAdapter から継承)	
SelectCommand プロパティ	System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet) または System.Data.Common.DbDataAdapter.FillSchema (System.Data.DataSet,System.Data.SchemaType) 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 (DbDataAdapter から継承)	
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 (System.Data.DataSet) 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 © 2007, iAnywhere Solutions, Inc.

パブリック・メソッド

メンバ名	説明
Fill (DbDataAdapter から継承)	DataSet または DataTable に入力します。
FillSchema (DbDataAdapter から継承)	DataTable を DataSet に追加し、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。
GetFillParameters メソッド	SELECT 文の実行時にユーザによって設定されるパラメータを返します。
ResetFillLoadOption (DataAdapter から継承)	
ShouldSerializeAcceptChangesDuringFill (DataAdapter から継承)	
ShouldSerializeFillLoadOption (DataAdapter から継承)	
Update (DbDataAdapter から継承)	DataSet 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。

パブリック・イベント

メンバ名	説明
FillError (DataAdapter から継承)	
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。更新が試みられると、イベントが発生します。
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。更新が試みられると、イベントが発生します。

参照

- 「ULDataAdapter クラス」 645 ページ
- DataSet
- 「ULDataReader クラス」 676 ページ
- 「ULTable クラス」 933 ページ
- DbDataAdapter
- IDbDataAdapter
- IDataAdapter
- IDisposable

ULDataAdapter コンストラクタ

「ULDataAdapter クラス」 645 ページの新しいインスタンスを初期化します。

Copyright © 2007, iAnywhere Solutions, Inc.

ULDataAdapter() コンストラクタ

ULDataAdapter オブジェクトを初期化します。

構文

Visual Basic

```
Public Sub New()
```

C#

```
public  ULDataAdapter();
```

参照

- 「ULDataAdapter クラス」 645 ページ
- 「ULDataAdapter メンバ」 645 ページ
- 「ULDataAdapter コンストラクタ」 647 ページ
- 「ULDataAdapter(ULCommand) コンストラクタ」 648 ページ
- 「ULDataAdapter(String, ULConnection) コンストラクタ」 649 ページ
- 「ULDataAdapter(String, String) コンストラクタ」 649 ページ

ULDataAdapter(ULCommand) コンストラクタ

ULDataAdapter オブジェクトを、指定された SELECT 文で初期化します。

構文

Visual Basic

```
Public Sub New( _
  ByVal selectCommand As ULCommand _
)
```

C#

```
public  ULDataAdapter(
  ULCommand selectCommand
);
```

パラメータ

- **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 クラス」 645 ページ
- 「ULDataAdapter メンバ」 645 ページ
- 「ULDataAdapter コンストラクタ」 647 ページ
- 「ULDataAdapter() コンストラクタ」 648 ページ
- 「ULDataAdapter(String, ULConnection) コンストラクタ」 649 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULDataAdapter(String, String) コンストラクタ」 649 ページ
- 「ULCommand クラス」 509 ページ
- DbDataAdapter.Fill
- DataSet

ULDataAdapter(String, ULConnection) コンストラクタ

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。

構文

Visual Basic

```
Public Sub New( _
  ByVal selectCommandText As String, _
  ByVal selectConnection As ULConnection _
)
```

C#

```
public  ULDataAdapter(
  string  selectCommandText,
  ULConnection selectConnection
);
```

パラメータ

- **selectCommandText**　ULDataAdapter の ULDataAdapter.SelectCommand 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 クラス」 645 ページ
- 「ULDataAdapter メンバ」 645 ページ
- 「ULDataAdapter コンストラクタ」 647 ページ
- 「ULDataAdapter() コンストラクタ」 648 ページ
- 「ULDataAdapter(ULCommand) コンストラクタ」 648 ページ
- 「ULDataAdapter(String, String) コンストラクタ」 649 ページ
- 「SelectCommand プロパティ」 652 ページ
- 「ULConnection クラス」 555 ページ

ULDataAdapter(String, String) コンストラクタ

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。

構文

Visual Basic

```
Public Sub New( _
```

```
  ByVal selectCommandText As String, _
  ByVal selectConnectionString As String _
)
```

C#

```
public ULDataAdapter(
  string selectCommandText,
  string selectConnectionString
);
```

パラメータ

- **selectCommandText** ULDataAdapter の ULDataAdapter.SelectCommand 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 クラス」 645 ページ
- 「ULDataAdapter メンバ」 645 ページ
- 「ULDataAdapter コンストラクタ」 647 ページ
- 「ULDataAdapter() コンストラクタ」 648 ページ
- 「ULDataAdapter(ULCommand) コンストラクタ」 648 ページ
- 「ULDataAdapter(String, ULConnection) コンストラクタ」 649 ページ
- 「SelectCommand プロパティ」 652 ページ

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(System.Data.DataSet) 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。

構文

Visual Basic

```
Public Property DeleteCommand As ULCommand
```

C#

```
public ULCommand DeleteCommand { get; set; }
```

プロパティ値

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。

備考

DeleteCommand 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 は、既存の ULCommand への参照を保持します。

Copyright © 2007, iAnywhere Solutions, Inc.

これは、System.Data.IDbDataAdapter.DeleteCommand と System.Data.Common.DbDataAdapter.DeleteCommand が厳密に型指定されたものです。

参照

- 「ULDataAdapter クラス」 645 ページ
- 「ULDataAdapter メンバ」 645 ページ
- 「ULCommand クラス」 509 ページ
- DbDataAdapter.Update
- DataSet
- 「ULCommand クラス」 509 ページ
- IDbDataAdapter.DeleteCommand
- DbDataAdapter.DeleteCommand

InsertCommand プロパティ

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、DbDataAdapter.Update(System.Data.DataSet) 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。

構文

Visual Basic

Public Property **InsertCommand** As ULCommand

C#

public ULCommand **InsertCommand** { get; set; }

プロパティ値

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。

備考

InsertCommand 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 は、既存の ULCommand への参照を保持します。

これは、System.Data.IDbDataAdapter.InsertCommand と System.Data.Common.DbDataAdapter.InsertCommand が厳密に型指定されたものです。

参照

- 「ULDataAdapter クラス」 645 ページ
- 「ULDataAdapter メンバ」 645 ページ
- 「ULCommand クラス」 509 ページ
- DbDataAdapter.Update
- DataSet
- 「ULCommand クラス」 509 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- IDbDataAdapter.InsertCommand
- DbDataAdapter.InsertCommand

SelectCommand プロパティ

System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet) または System.Data.Common.DbDataAdapter.FillSchema(System.Data.DataSet,System.Data.SchemaType) 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。

構文

Visual Basic

Public Property **SelectCommand** As ULCommand

C#

public ULCommand **SelectCommand** { get; set; }

プロパティ値

System.Data.DataSet を設定するために実行される ULCommand オブジェクト。

備考

SelectCommand 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 は、既存の ULCommand への参照を保持します。

SelectCommand がローを返さない場合、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) のコンストラクタにも指定できます。

これは、System.Data.IDbDataAdapter.SelectCommand と System.Data.Common.DbDataAdapter.SelectCommand が厳密に型指定されたものです。

参照

- 「ULDataAdapter クラス」 645 ページ
- 「ULDataAdapter メンバ」 645 ページ
- 「ULCommand クラス」 509 ページ
- DataSet
- DbDataAdapter.Fill
- DbDataAdapter.FillSchema
- 「ULDataAdapter(ULCommand) コンストラクタ」 648 ページ
- 「ULDataAdapter(String, ULConnection) コンストラクタ」 649 ページ
- 「ULDataAdapter(String, String) コンストラクタ」 649 ページ
- IDbDataAdapter.SelectCommand
- DbDataAdapter.SelectCommand

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Readonly Property **TableMappings** As DataTableMappingCollection

C#

public DataTableMappingCollection **TableMappings** { get;}

プロパティ値

ソース・テーブルと System.Data.DataTables とのマスタ・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。

備考

変更を調整する場合、ULDataAdapter 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 が厳密に型指定されたものです。

参照

- 「ULDataAdapter クラス」 645 ページ
- 「ULDataAdapter メンバ」 645 ページ
- DataTable
- DataTableMapping
- DataTable
- DataTableMappingCollection
- DataSet
- IDataAdapter.TableMappings

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(System.Data.DataSet) 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。

構文

Visual Basic

Public Property **UpdateCommand** As ULCommand

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public ULCommand **UpdateCommand** { get; set; }

プロパティ値

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。

備考

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 は、既存の ULCommand 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 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 と System.Data.Common.DbDataAdapter.DeleteCommand が厳密に型指定されたものです。

参照

- 「ULDataAdapter クラス」 645 ページ
- 「ULDataAdapter メンバ」 645 ページ
- 「ULCommand クラス」 509 ページ
- DataSet
- DbDataAdapter.Update
- 「UpdatedRowSource プロパティ」 522 ページ
- IDbDataAdapter.UpdateCommand
- DbDataAdapter.DeleteCommand

GetFillParameters メソッド

SELECT 文の実行時にユーザによって設定されるパラメータを返します。

構文

Visual Basic

Public Function **GetFillParameters()** As ULParameter

C#

public ULParameter **GetFillParameters();**

戻り値

ユーザによって設定されたパラメータが含まれる ULParameter オブジェクトの配列。

備考

これは、System.Data.Common.DbDataAdapter.GetFillParameters が厳密に型指定されたものです。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULDataAdapter クラス」 645 ページ
- 「ULDataAdapter メンバ」 645 ページ
- DbDataAdapter.GetFillParameters
- 「ULParameter クラス」 773 ページ

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。更新が試みられると、イベントが発生します。

構文

Visual Basic

Public Event **RowUpdated** As ULRowUpdatedEventHandler

C#

public event ULRowUpdatedEventHandler **RowUpdated** ;

備考

ロー更新イベントを処理するには、ULRowUpdatedEventHandler 委任を作成し、このイベントにアタッチする必要があります。

イベント・データ

- **Command** DbDataAdapter.Update(System.Data.DataRow [],System.Data.Common.DataTableMapping) が呼び出されたときに実行された ULCommand を返します。
- **RecordsAffected** 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。SELECT 文の場合、この値は -1 です。
- **Command**
- **Errors**
- **Row**
- **RowCount**
- **StatementType**
- **Status**
- **TableMapping**

参照

- 「ULDataAdapter クラス」 645 ページ
- 「ULDataAdapter メンバ」 645 ページ
- 「ULRowUpdatedEventHandler 委任」 850 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。更新が試みられると、イベントが発生します。

構文

Visual Basic

Public Event **RowUpdating** As ULRowUpdatingEventHandler

C#

public event ULRowUpdatingEventHandler **RowUpdating** ;

備考

ロー更新イベントを処理するには、ULRowUpdatingEventHandler 委任を作成し、このイベントにアタッチする必要があります。

イベント・データ

- **Command** DbDataAdapter.Update(System.Data.DataRow [],System.Data.Common.DataTableMapping) の実行時に実行する ULCommand を指定します。
- **Command**
- **Errors**
- **Row**
- **StatementType**
- **Status**
- **TableMapping**

参照

- 「ULDataAdapter クラス」 645 ページ
- 「ULDataAdapter メンバ」 645 ページ
- 「ULRowUpdatedEventHandler 委任」 850 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULDatabaseManager クラス

UL 拡張： 同期リスナと Ultra Light.NET のランタイム・タイプを管理します。ULDatabaseManager クラスを使用すると、Ultra Light.NET データベースを削除できます。このクラスは継承できません。

構文

Visual Basic

Public NotInheritable Class **ULDatabaseManager**

C#

public sealed class **ULDatabaseManager**

備考

このクラスはシングルトン・クラスであり、その唯一のインスタンスには、静的な (Visual Basic の Shared) ULConnection.DatabaseManager のみを通じてアクセスできます。

Ultra Light.NET の Ultra Light エンジンのランタイムを使用するには、ULDatabaseManager.RuntimeType 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

// C#
ULDatabaseManager.RuntimeType = ULRuntimeType.UL_ENGINE_CLIENT;
ULConnection conn = new ULConnection();
// The RuntimeType is now locked
```

参照

- 「ULDatabaseManager メンバ」 658 ページ
- 「DatabaseManager プロパティ」 563 ページ
- 「RuntimeType プロパティ」 658 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULDatabaseManager メンバ

パブリック・プロパティ

メンバ名	説明
RuntimeType プロパティ	Ultra Light.NET ランタイム・タイプを指定します。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。

パブリック・メソッド

メンバ名	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。 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。
SetActiveSyncListener メソッド	ActiveSync 用 Mobile Link プロバイダからの ActiveSync の呼び出しを処理するリスナ・オブジェクトを指定します。
SetServerSyncListener メソッド	指定されたサーバ同期メッセージの処理に使用するリスナ・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。

参照

- 「ULDatabaseManager クラス」 657 ページ
- 「DatabaseManager プロパティ」 563 ページ
- 「RuntimeType プロパティ」 658 ページ

RuntimeType プロパティ

Ultra Light.NET ランタイム・タイプを指定します。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。

構文

Visual Basic

```
Public Shared Property RuntimeType As ULRuntimeType
```

C#

```
public const ULRuntimeType RuntimeType  { get; set; }
```


Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

アンマネージ Ultra Light.NET ランタイムのタイプを示す ULRuntimeType 値。

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

// C#
ULDatabaseManager.RuntimeType = ULRuntimeType.UL_ENGINE_CLIENT;
ULConnection conn = new ULConnection();
// The RuntimeType is now locked
```

参照

- 「ULDatabaseManager クラス」 657 ページ
- 「ULDatabaseManager メンバ」 658 ページ
- 「ULRuntimeType 列挙」 855 ページ

CreateDatabase メソッド

新しい Ultra Light データベースを作成します。

構文

Visual Basic

```
Public Sub CreateDatabase( _
   ByVal connString As String, _
   ByVal collationData As Byte(), _
   ByVal createParms As String _
)
```

C#

```
public void CreateDatabase(
   string  connString,
   byte[] collationData,
   string  createParms
);
```

パラメータ

- **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。詳細については、「ULConnectionParms クラス」 592 ページを参照してください。
- **collationData**　データベースが文字列を格納して比較する方法を指定する照合データ。

Copyright © 2007, iAnywhere Solutions, Inc.

- **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。詳細については、「ULCreateParms クラス」 624 ページを参照してください。

備考

照合を指定するには、SQL Anywhere インストール・ディレクトリの srcollations¥cs (C# プロジェクトの場合) または srcollations¥vb.net (Visual Basic プロジェクトの場合) サブディレクトリから、適切な照合データ・ソースを含める必要があります。プロジェクトに追加したら、照合の Data プロパティを使用して、CreateDatabase() メソッドに対して照合データを提供できます。

例

次のコードでは、Windows CE 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.DatabaseOnCE = "¥UltraLite¥MyDatabase.udb"
' Assumes file src¥ulcollations¥vb.net¥coll_1250LATIN2.vb is
' also compiled in the current project
ULConnection.DatabaseManager.CreateDatabase( _
    openParms.ToString(), _
    iAnywhere.UltraLite.Collations.Collation_1250LATIN2.Data, _
    "" _
  )
Dim conn As ULConnection = _
  New ULConnection( openParms.ToString() )
conn.Open()


// C#
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnCE = @"¥UltraLite¥MyDatabase.udb";
// Assumes file src¥ulcollations¥cs¥coll_1250LATIN2.cs is
// also compiled in the current project
ULConnection.DatabaseManager.CreateDatabase(
    openParms.ToString(),
    iAnywhere.UltraLite.Collations.Collation_1250LATIN2.Data,
    ""
  );
ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```

参照

- 「ULDatabaseManager クラス」 657 ページ
- 「ULDatabaseManager メンバ」 658 ページ
- 「Open メソッド」 584 ページ

DropDatabase メソッド

指定されたデータベースを削除します。

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。

 Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Sub DropDatabase( _
  ByVal connString As String _
)
```

C#

```
public void DropDatabase(
  string  connString
);
```

パラメータ

- **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。詳細については、「ULConnectionParms クラス」 592 ページを参照してください。

例

次のコードでは、Windows CE 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.DatabaseOnCE = "¥UltraLite¥MyDatabase.udb"
ULConnection.DatabaseManager.DropDatabase( _
   connParms.ToString() _
  )


// C#
ULConnectionParms connParms = new ULConnectionParms();
connParms.DatabaseOnCE = @"¥UltraLite¥MyDatabase.udb";
ULConnection.DatabaseManager.DropDatabase(
   connParms.ToString()
  );
ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```

参照

- 「ULDatabaseManager クラス」 657 ページ
- 「ULDatabaseManager メンバ」 658 ページ
- 「Open メソッド」 584 ページ

SetActiveSyncListener メソッド

ActiveSync 用 Mobile Link プロバイダからの ActiveSync の呼び出しを処理するリスナ・オブジェクトを指定します。

構文

Visual Basic

Copyright © 2007, iAnywhere Solutions, Inc.

```
Public Sub SetActiveSyncListener( _
  ByVal appClassName As String, _
  ByVal listener As ULActiveSyncListener _
)
```

C#

```
public void SetActiveSyncListener(
  string  appClassName,
  ULActiveSyncListener listener
);
```

パラメータ

- **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。前のリスナを削除するには、null (Visual Basic の Nothing) を使用します。

備考

パラメータ *appClassName* 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* のみを使用できます。リスナが、特定の *appClassName* に登録されているときは、別の *appClassName* で SetServerSyncListener または SetActiveSyncListener を呼び出すと失敗します。

ActiveSync リスナを削除するには、*listener* パラメータとして null 参照 (Visual Basic の Nothing) を使用して SetActiveSyncListener を呼び出します。

すべてのリスナを削除するには、すべてのパラメータに null 参照 (Visual Basic の Nothing) を使用して SetServerSyncListener を呼び出します。

アプリケーションは、すべてのリスナを削除してから終了する必要があります。

例

SetActiveSyncListener の例については、「ActiveSyncInvoked メソッド」 475 ページを参照してください。

参照

- 「ULDatabaseManager クラス」 657 ページ
- 「ULDatabaseManager メンバ」 658 ページ
- 「SetServerSyncListener メソッド」 662 ページ
- 「SetActiveSyncListener メソッド」 661 ページ
- 「ULActiveSyncListener インタフェース」 475 ページ
- 「ActiveSyncInvoked メソッド」 475 ページ

SetServerSyncListener メソッド

指定されたサーバ同期メッセージの処理に使用するリスナ・オブジェクトを指定します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Sub SetServerSyncListener( _
  ByVal messageName As String, _
  ByVal appClassName As String, _
  ByVal listener As ULServerSyncListener _
)
```

C#

```
public void SetServerSyncListener(
  string  messageName,
  string  appClassName,
  ULServerSyncListener listener
);
```

パラメータ

- **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。前のリスナを削除するには、null (Visual Basic の Nothing) を使用します。

備考

パラメータ *appClassName* 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* のみを使用できます。リスナが、特定の *appClassName* に登録されているときは、別の *appClassName* で SetServerSyncListener または SetActiveSyncListener を呼び出すと失敗します。

特定のメッセージのリスナを削除するには、*listener* パラメータとして null 参照 (Visual Basic の Nothing) を使用して SetServerSyncListener を呼び出します。

すべてのリスナを削除するには、すべてのパラメータに null 参照 (Visual Basic の Nothing) を使用して SetServerSyncListener を呼び出します。

アプリケーションは、すべてのリスナを削除してから終了する必要があります。

例

SetServerSyncListener の例については、「ServerSyncInvoked メソッド」 856 ページを参照してください。

参照

- 「ULDatabaseManager クラス」 657 ページ
- 「ULDatabaseManager メンバ」 658 ページ
- 「SetServerSyncListener メソッド」 662 ページ
- 「SetActiveSyncListener メソッド」 661 ページ
- 「ServerSyncInvoked メソッド」 856 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

SignalSyncIsComplete メソッド

ActiveSync 用の Mobile Link プロバイダに対して、アプリケーションが同期を完了したことを通知します。

構文

Visual Basic

Public Sub **SignalSyncIsComplete()**

C#

public void **SignalSyncIsComplete();**

例

SignalSyncIsComplete の例については、「ActiveSyncInvoked メソッド」 475 ページを参照してください。

参照

- 「ULDatabaseManager クラス」 657 ページ
- 「ULDatabaseManager メンバ」 658 ページ
- 「SignalSyncIsComplete メソッド」 664 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

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。このクラスは継承できません。

構文

Visual Basic

Public NotInheritable Class **ULDatabaseSchema**

C#

public sealed class **ULDatabaseSchema**

備考

このクラスにはコンストラクタがありません。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 メンバ」 665 ページ
- 「ULDatabaseSchema クラス」 665 ページ
- 「Schema プロパティ」 565 ページ

ULDatabaseSchema メンバ

パブリック・プロパティ

メンバ名	説明
CollationName プロパティ	データベースの照合順の名前です。
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。

パブリック・メソッド

メンバ名	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。パブリケーション ID は、パブリケーション・マスクではありません。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
GetPublicationSchema メソッド	指定されたパブリケーションに対応するパブリケーション・スキーマを返します。
GetTableCountInPublications メソッド	指定されたパブリケーション・マスクに含まれるテーブルの数を返します。
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。

参照

- 「ULDatabaseSchema クラス」 665 ページ
- 「ULDatabaseSchema クラス」 665 ページ
- 「Schema プロパティ」 565 ページ

CollationName プロパティ

データベースの照合順の名前です。

構文

Visual Basic

Public Readonly Property **CollationName** As String

C#

public string **CollationName** { get;}

プロパティ値

データベースの照合順を示す文字列。

備考

データベースの照合順は、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。

参照

- 「ULDatabaseSchema クラス」 665 ページ
- 「ULDatabaseSchema メンバ」 665 ページ
- 「GetDatabaseProperty メソッド」 669 ページ

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **IsCaseSensitive** As Boolean

C#

public bool **IsCaseSensitive** { get;}

プロパティ値

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。

備考

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 クラス」 665 ページ
- 「ULDatabaseSchema メンバ」 665 ページ
- 「GetDatabaseProperty メソッド」 669 ページ
- 「UserID プロパティ」 602 ページ
- 「Password プロパティ」 602 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。

構文

Visual Basic

Public Readonly Property **IsOpen** As Boolean

C#

public bool **IsOpen** { get;}

プロパティ値

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。

備考

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。

参照

- 「ULDatabaseSchema クラス」 665 ページ
- 「ULDatabaseSchema メンバ」 665 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Readonly Property **PublicationCount** As Integer

C#

public int **PublicationCount** { get;}

プロパティ値

データベース内のパブリケーション数。接続が開いていない場合は 0。

備考

パブリケーション ID の範囲は、1 ー PublicationCount です。パブリケーション ID は、パブリケーション・マスクではありません。

注意：パブリケーションの ID、マスク、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、マスク、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 クラス」 665 ページ
- 「ULDatabaseSchema メンバ」 665 ページ
- 「GetPublicationName メソッド」 671 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。

構文

Visual Basic

Public Readonly Property **TableCount** As Integer

C#

public int **TableCount** { get;}

プロパティ値

データベース内のテーブル数。接続が開いていない場合は 0。

備考

テーブル ID の範囲は、1 ー TableCount です。

 Copyright © 2007, iAnywhere Solutions, Inc.

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。

参照

- 「ULDatabaseSchema クラス」 665 ページ
- 「ULDatabaseSchema メンバ」 665 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。

構文

Visual Basic

```
Public Function GetDatabaseProperty( _
  ByVal name As String _
) As String
```

C#

```
public string  GetDatabaseProperty(
  string  name
);
```

パラメータ

- **name**　値を取得するデータベース・プロパティの名前。プロパティ名には大文字と小文字の区別はありません。

戻り値

文字列として返されるプロパティの値。

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 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。

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ	説明
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。
CollationName	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。
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。
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。
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。

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ	説明
timestamp_increment	2 つのユニークなタイムスタンプ間のナノ秒 (1,000,000 分の 1 秒) 単位の最小差。

参照

- 「ULDatabaseSchema クラス」 665 ページ
- 「ULDatabaseSchema メンバ」 665 ページ
- 「SetDatabaseOption メソッド」 674 ページ
- 「CollationName プロパティ」 666 ページ
- 「UserID プロパティ」 602 ページ
- 「Password プロパティ」 602 ページ
- TimeSpan
- 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。パブリケーション ID は、パブリケーション・マスクではありません。

構文

Visual Basic

```
Public Function GetPublicationName( _
   ByVal publD As Integer _
) As String
```

C#

```
public string  GetPublicationName(
   int publD
);
```

パラメータ

- **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。

備考

注意：パブリケーションの ID、マスク、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、マスク、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 クラス」 665 ページ
- 「ULDatabaseSchema メンバ」 665 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「PublicationCount プロパティ」 668 ページ
- 「PublicationCount プロパティ」 668 ページ

GetPublicationSchema メソッド

指定されたパブリケーションに対応するパブリケーション・スキーマを返します。

構文

Visual Basic

```
Public Function GetPublicationSchema( _
   ByVal name As String _
) As ULPublicationSchema
```

C#

```
public ULPublicationSchema GetPublicationSchema(
   string  name
);
```

パラメータ

- **name**　パブリケーションの名前。

戻り値

指定されたパブリケーションを示す ULPublicationSchema オブジェクト。

参照

- 「ULDatabaseSchema クラス」 665 ページ
- 「ULDatabaseSchema メンバ」 665 ページ
- 「GetPublicationName メソッド」 671 ページ
- 「ULPublicationSchema クラス」 810 ページ

GetTableCountInPublications メソッド

指定されたパブリケーション・マスクに含まれるテーブルの数を返します。

構文

Visual Basic

```
Public Function GetTableCountInPublications( _
   ByVal mask As Integer _
) As Integer
```

C#

```
public int GetTableCountInPublications(
   int mask
);
```


Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **mask**　チェックするパブリケーションのセット。パブリケーション・マスク作成の詳細については、「ULPublicationSchema クラス」 810 ページを参照してください。

戻り値

パブリケーションのセットに含まれるテーブル数。

備考

カウントには、名前が _nosync で終わるテーブルは含まれません。

注意：パブリケーションの ID、マスク、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、マスク、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 クラス」 665 ページ
- 「ULDatabaseSchema メンバ」 665 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。

構文

Visual Basic

```
Public Function GetTableName( _
  ByVal tableID As Integer _
) As String
```

C#

```
public string  GetTableName(
  int tableID
);
```

パラメータ

- **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

文字列として返されるテーブル名。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULDatabaseSchema クラス」 665 ページ
- 「ULDatabaseSchema メンバ」 665 ページ
- 「TableCount プロパティ」 668 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。

構文

Visual Basic

```
Public Sub SetDatabaseOption( _
  ByVal name As String, _
  ByVal value As String _
)
```

C#

```
public void SetDatabaseOption(
  string  name,
  string  value
);
```

パラメータ

- **name**　データベース・オプションの名前。オプション名には大文字と小文字の区別はありません。
- **value**　オプションの新しい値。

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。

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。
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 クラス」 665 ページ
- 「ULDatabaseSchema メンバ」 665 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「GetDatabaseProperty メソッド」 669 ページ
- UInt32.MaxValue
- 「INVALID_DATABASE_ID フィールド」 560 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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示します。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。

構文

Visual Basic

```
Public Class ULDataReader
  Inherits DbDataReader
```

C#

```
public class ULDataReader : DbDataReader
```

備考

ULDataReader にはコンストラクタがありません。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 FOR READ ONLY", conn _
  )
Dim reader As ULDataReader = cmd.ExecuteReader()

// C#
ULCommand cmd = new ULCommand(
    "SELECT emp_id FROM employee FOR READ ONLY", 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。

継承： System.Data.Common.DbDataReader

実装： System.Data.IDataReader、System.Data.IDataRecord、System.IDisposable

参照

- 「ULDataReader メンバ」 677 ページ
- 「ULCommand クラス」 509 ページ
- 「ExecuteResultSet() メソッド」 538 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

- 「ExecuteTable() メソッド」 541 ページ
- 「ULDataAdapter クラス」 645 ページ
- 「GetString メソッド」 703 ページ
- DbDataReader
- IDataReader
- IDataRecord
- IDisposable

ULDataReader メンバ

パブリック・プロパティ

メンバ名	説明
Depth プロパティ	現在のローのネストの深さを返します。最も外側のテーブルの深さは 0 です。
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。
Item プロパティ	指定されたカラムの値を Object のインスタンスとして取得します。
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。SELECT 文または CommandType.TableDirect テーブルの場合、この値は -1 です。
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。
VisibleFieldCount (DbDataReader から継承)	

パブリック・メソッド

メンバ名	説明
Close メソッド	カーソルを閉じます。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
Dispose (DbDataReader から継承)	
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。ULDbType.Binary 型、ULDbType.LongBinary 型、ULDbType.UniqueIdentifier 型のカラムの場合にのみ有効です。
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 (DbDataReader から継承)	
GetDataTypeName メソッド	指定されたカラムのプロバイダ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。
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 (DbDataReader から継承)	

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
GetProviderSpecificValue (DbDataReader から継承)	
GetProviderSpecificValues (DbDataReader から継承)	
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。
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。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULDataReader クラス」 676 ページ
- 「ULCommand クラス」 509 ページ
- 「ExecuteResultSet() メソッド」 538 ページ
- 「ExecuteTable() メソッド」 541 ページ
- 「ULDataAdapter クラス」 645 ページ
- 「GetString メソッド」 703 ページ
- DbDataReader
- IDataReader
- IDataRecord
- IDisposable

Depth プロパティ

現在のローのネストの深さを返します。最も外側のテーブルの深さは 0 です。

構文

Visual Basic

Public Overrides Readonly Property **Depth** As Integer

C#

public override int **Depth** { get;}

プロパティ値

Ultra Light.NET のすべての結果セットの深さはゼロです。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ

FieldCount プロパティ

このカーソル内のカラム数を返します。

構文

Visual Basic

Public Overrides Readonly Property **FieldCount** As Integer

C#

public override int **FieldCount** { get;}

プロパティ値

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。

 Copyright © 2007, iAnywhere Solutions, Inc.

備考

このメソッドは、ULCursorSchema.ColumnCount メソッドと同じです。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「ColumnCount プロパティ」 637 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。

構文

Visual Basic

Public Overrides Readonly Property **HasRows** As Boolean

C#

public override bool **HasRows** { get;}

プロパティ値

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。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ

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。

構文

Visual Basic

Public Readonly Property **IsBOF** As Boolean

C#

public bool **IsBOF** { get;}

プロパティ値

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。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Overrides Readonly Property **IsClosed** As Boolean

C#

public override bool **IsClosed** { get;}

プロパティ値

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。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ

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。

構文

Visual Basic

Public Readonly Property **IsEOF** As Boolean

C#

public bool **IsEOF** { get;}

プロパティ値

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。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ

Item プロパティ

指定されたカラムの値を Object のインスタンスとして取得します。

Copyright © 2007, iAnywhere Solutions, Inc.

Item(Int32) 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。C# では、このプロパティは ULDataReader クラスのインデクサです。

構文

Visual Basic

```
Public Overrides Default Readonly Property Item ( _
  ByVal columnID As Integer _
) As Object
```

C#

```
public override object this [
  int columnID
] { get;}
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

プロパティ値

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。

備考

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 クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「Item プロパティ」 682 ページ
- 「GetFieldType メソッド」 695 ページ
- 「Item(String) プロパティ」 683 ページ
- 「GetValue メソッド」 707 ページ
- 「FieldCount プロパティ」 680 ページ

Item(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。C# では、このプロパティは ULDataReader クラスのインデクサです。

構文

Visual Basic

```
Public Overrides Default Readonly Property Item ( _
  ByVal name As String _
) As Object
```

Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public override object this [
   string  name
] { get;}
```

パラメータ

- **name**　カラムの名前。

プロパティ値

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。

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 クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「Item プロパティ」 682 ページ
- 「Item(Int32) プロパティ」 683 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetValue メソッド」 707 ページ
- 「GetFieldType メソッド」 695 ページ

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。SELECT 文または CommandType.TableDirect テーブルの場合、この値は -1 です。

構文

Visual Basic

```
Public Overrides Readonly Property RecordsAffected As Integer
```

C#

```
public override int RecordsAffected  { get;}
```


Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

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

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- CommandType.TableDirect

RowCount プロパティ

UL 拡張： カーソル内のロー数を返します。

構文

Visual Basic

Public Readonly Property **RowCount** As Integer

C#

public int **RowCount** { get;}

プロパティ値

カーソル内のロー数。

備考

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

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「StartSynchronizationDelete メソッド」 586 ページ
- 「StopSynchronizationDelete メソッド」 587 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。

構文

Visual Basic

Public Readonly Property **Schema** As ULCursorSchema

C#

public ULCursorSchema **Schema** { get;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

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。

備考

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示します。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「ULTableSchema クラス」 954 ページ
- 「GetSchemaTable メソッド」 701 ページ
- 「ULResultSetSchema クラス」 839 ページ

Close メソッド

カーソルを閉じます。

構文

Visual Basic

```
Public Overrides Sub Close()
```

C#

```
public override void Close();
```

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。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ

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。

構文

Visual Basic

```
Public Overrides Function GetBoolean( _
   ByVal columnID As Integer _
) As Boolean
```

 Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public override bool GetBoolean(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

System.Boolean として返されるカラム値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- Boolean
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Overrides Function GetByte( _
  ByVal columnID As Integer _
) As Byte
```

C#

```
public override byte GetByte(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

System.Byte として返されるカラム値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- Byte
- 「FieldCount プロパティ」 680 ページ

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。ULDbType.Binary 型、ULDbType.LongBinary 型、ULDbType.UniqueIdentifier 型のカラムの場合にのみ有効です。

GetBytes(Int32, Int64, Byte[], Int32, Int32) 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。

構文

Visual Basic

```
Public Overrides Function GetBytes( _
   ByVal columnID As Integer, _
   ByVal srcOffset As Long, _
   ByVal dst As Byte(), _
   ByVal dstOffset As Integer, _
   ByVal count As Integer _
) As Long
```

C#

```
public override long GetBytes(
   int columnID,
   long srcOffset,
   byte[] dst,
   int dstOffset,
   int count
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **srcOffset**　カラム値の開始位置。ゼロが値の最初です。
- **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 © 2007, iAnywhere Solutions, Inc.

備考

null 参照 (Visual Basic の Nothing) である *dst* バッファを渡すと、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* が *dst*.Length. 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 クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetBytes メソッド」 688 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- 「GetBytes(Int32) メソッド」 689 ページ
- 「ULDbType 列挙」 715 ページ
- Byte
- 「ULException クラス」 719 ページ
- 「ULSQLCode 列挙」 859 ページ
- 「FieldCount プロパティ」 680 ページ

GetBytes(Int32) 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。ULDbType.Binary 型、ULDbType.LongBinary 型、ULDbType.UniqueIdentifier 型のカラムの場合にのみ有効です。

構文

Visual Basic

```
Public Function GetBytes( _
  ByVal columnID As Integer _
) As Byte
```

C#

```
public byte GetBytes(
  int columnID
);
```

Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

System.Bytes 配列として返されるカラム値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetBytes メソッド」 688 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- 「GetBytes(Int32, Int64, Byte[], Int32, Int32) メソッド」 688 ページ
- Byte
- 「ULDbType 列挙」 715 ページ
- 「ULDbType 列挙」 715 ページ
- 「ULDbType 列挙」 715 ページ

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。

構文

Visual Basic

```
Public Overrides Function GetChar( _
  ByVal columnID As Integer _
) As Char
```

C#

```
public override char GetChar(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「GetFieldType メソッド」 695 ページ
- 「GetString メソッド」 703 ページ
- 「FieldCount プロパティ」 680 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。

構文

Visual Basic

```
Public Overrides Function GetChars( _
  ByVal columnID As Integer, _
  ByVal srcOffset As Long, _
  ByVal dst As Char(), _
  ByVal dstOffset As Integer, _
  ByVal count As Integer _
) As Long
```

C#

```
public override long GetChars(
  int columnID,
  long srcOffset,
  char[] dst,
  int dstOffset,
  int count
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **srcOffset**　カラム値の開始位置。ゼロが値の最初です。
- **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。

戻り値

実際にコピーされた文字数。

備考

NULL 参照 (Visual Basic の Nothing) である *dst* バッファを渡すと、GetChars はフィールドの長さを文字数として返します。

Copyright © 2007, iAnywhere Solutions, Inc.

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* が *dst*.Length. 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 クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- 「ULDbType 列挙」 715 ページ
- Char
- 「ULException クラス」 719 ページ
- 「ULSQLCode 列挙」 859 ページ
- 「FieldCount プロパティ」 680 ページ

GetDataTypeName メソッド

指定されたカラムのプロバイダのデータ型の名前を返します。

構文

Visual Basic

```
Public Overrides Function GetDataTypeName( _
   ByVal columnID As Integer _
) As String
```

C#

```
public override string  GetDataTypeName(
   int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

Copyright © 2007, iAnywhere Solutions, Inc.

戻り値

カラムの ULDbType に対応する文字列。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetColumnULDbType メソッド」 643 ページ
- 「FieldCount プロパティ」 680 ページ
- 「ULDbType 列挙」 715 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。

構文

Visual Basic

```
Public Overrides Function GetDateTime( _
  ByVal columnID As Integer _
) As Date
```

C#

```
public override DateTime GetDateTime(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

System.DateTime として返されるカラム値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- DateTime
- 「FieldCount プロパティ」 680 ページ

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Overrides Function GetDecimal( _
  ByVal columnID As Integer _
) As Decimal
```

C#

```
public override decimal GetDecimal(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

System.Decimal として返されるカラム値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- Decimal
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Overrides Function GetDouble( _
  ByVal columnID As Integer _
) As Double
```

C#

```
public override double GetDouble(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

Copyright © 2007, iAnywhere Solutions, Inc.

戻り値

System.Double として返されるカラム値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- Double
- 「FieldCount プロパティ」 680 ページ

GetEnumerator メソッド

ULDataReader の反復処理を実行する System.Collections.IEnumerator を返します。

構文

Visual Basic

```
Public Overrides Function GetEnumerator() As IEnumerator
```

C#

```
public override IEnumerator GetEnumerator();
```

戻り値

ULDataReader の System.Collections.IEnumerator。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 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。

構文

Visual Basic

```
Public Overrides Function GetFieldType( _
  ByVal columnID As Integer _
) As Type
```

C#

```
public override Type GetFieldType(
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

カラムの System.Type 値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetDataTypeName メソッド」 692 ページ
- 「GetColumnULDbType メソッド」 643 ページ
- Type
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Overrides Function GetFloat( _
  ByVal columnID As Integer _
) As Single
```

C#

```
public override float  GetFloat(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

System.Single として返されるカラム値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- Single
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Overrides Function GetGuid( _
  ByVal columnID As Integer _
) As Guid
```

C#

```
public override Guid GetGuid(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

Guid として返されるカラム値。

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 クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- 「GetColumnULDbType メソッド」 643 ページ
- 「GetColumnSize メソッド」 642 ページ
- Guid
- 「ULDbType 列挙」 715 ページ
- 「ULDbType 列挙」 715 ページ
- 「FieldCount プロパティ」 680 ページ

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Overrides Function GetInt16( _
  ByVal columnID As Integer _
) As Short
```

C#

```
public override short GetInt16(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

System.Int16 として返されるカラム値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- Int16
- 「FieldCount プロパティ」 680 ページ

GetInt32 メソッド

指定されたカラムの値を Int32 として返します。

構文

Visual Basic

```
Public Overrides Function GetInt32( _
  ByVal columnID As Integer _
) As Integer
```

C#

```
public override int GetInt32(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

Copyright © 2007, iAnywhere Solutions, Inc.

戻り値

Int32 として返されるカラム値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- Int32
- 「FieldCount プロパティ」 680 ページ

GetInt64 メソッド

指定されたカラムの値を Int64 として返します。

構文

Visual Basic

```
Public Overrides Function GetInt64( _
  ByVal columnID As Integer _
) As Long
```

C#

```
public override long GetInt64(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

Int64 として返されるカラム値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- Int64
- 「FieldCount プロパティ」 680 ページ

GetName メソッド

指定されたカラムの名前を返します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Overrides Function GetName( _
  ByVal columnID As Integer _
) As String
```

C#

```
public override string  GetName(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

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。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「FieldCount プロパティ」 680 ページ
- 「GetSchemaTable メソッド」 701 ページ
- 「GetColumnName メソッド」 639 ページ
- 「FieldCount プロパティ」 680 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。

構文

Visual Basic

```
Public Overrides Function GetOrdinal( _
  ByVal columnName As String _
) As Integer
```


Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public override int GetOrdinal(
  string  columnName
);
```

パラメータ

- **columnName**　カラムの名前。

戻り値

指定されたカラムのカラム ID。

備考

カラム ID の範囲は、0 ー 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 クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetSchemaTable メソッド」 701 ページ
- 「FieldCount プロパティ」 680 ページ
- 「GetColumnID メソッド」 638 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。

構文

Visual Basic

```
Public Overrides Function GetSchemaTable() As DataTable
```

C#

```
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 © 2007, iAnywhere Solutions, Inc.

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。
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 入力可能である場合は true、null 入力不可能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。

Copyright © 2007, iAnywhere Solutions, Inc.

DataTable カラム	説明
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。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「Schema プロパティ」 685 ページ
- 「FieldCount プロパティ」 680 ページ
- 「ULDbType 列挙」 715 ページ
- 「ULDbType 列挙」 715 ページ
- 「ULDbType 列挙」 715 ページ
- 「ULDbType 列挙」 715 ページ
- 「ULDbType 列挙」 715 ページ
- 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。

構文

Visual Basic

```
Public Overrides Function GetString( _
   ByVal columnID As Integer _
) As String
```

Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public override string  GetString(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

System.String として返されるカラム値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- String
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Function GetTimeSpan( _
  ByVal columnID As Integer _
) As TimeSpan
```

C#

```
public TimeSpan GetTimeSpan(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

System.TimeSpan として返されるカラム値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- TimeSpan
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Function GetUInt16( _
   ByVal columnID As Integer _
) As UInt16
```

C#

```
public ushort GetUInt16(
   int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

System.UInt16 として返されるカラム値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- UInt16
- 「FieldCount プロパティ」 680 ページ

GetUInt32 メソッド

指定されたカラムの値を UInt32 として返します。

構文

Visual Basic

```
Public Function GetUInt32( _
   ByVal columnID As Integer _
) As UInt32
```

Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public uint GetUInt32(
   int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

UInt32 として返されるカラム値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- UInt32
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Function GetUInt64( _
   ByVal columnID As Integer _
) As UInt64
```

C#

```
public ulong GetUInt64(
   int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

System.UInt64 として返されるカラム値。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- UInt64
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Overrides Function GetValue( _
  ByVal columnID As Integer _
) As Object
```

C#

```
public override object GetValue(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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。

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 クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- 「FieldCount プロパティ」 680 ページ
- 「Item(Int32) プロパティ」 683 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。

構文

Visual Basic

Copyright © 2007, iAnywhere Solutions, Inc.

```
Public Overrides Function GetValues( _
  ByVal values As Object() _
) As Integer
```

C#

```
public override int GetValues(
  object[] values
);
```

パラメータ

- **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。

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 クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「FieldCount プロパティ」 680 ページ
- 「GetFieldType メソッド」 695 ページ
- 「GetValue メソッド」 707 ページ
- オブジェクト

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。

構文

Visual Basic

```
Public Overrides Function IsDBNull( _
  ByVal columnID As Integer _
) As Boolean
```

C#

Copyright © 2007, iAnywhere Solutions, Inc.

```
public override bool IsDBNull(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

戻り値

値が NULL の場合は true、NULL でない場合は false。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- 「FieldCount プロパティ」 680 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。

構文

Visual Basic

```
Public Sub MoveAfterLast()
```

C#

```
public void MoveAfterLast();
```

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ

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。

構文

Visual Basic

```
Public Sub MoveBeforeFirst()
```

C#

```
public void MoveBeforeFirst();
```

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ

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。

構文

Visual Basic

Public Function **MoveFirst()** As Boolean

C#

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。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ

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。

構文

Visual Basic

Public Function **MoveLast()** As Boolean

C#

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。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Function **MoveNext()** As Boolean

C#

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。

備考

このメソッドは、ULDataReader.Read メソッドと同じです。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「Read メソッド」 713 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。

構文

Visual Basic

Public Function **MovePrevious()** As Boolean

C#

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。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Public Function MoveRelative( _
   ByVal offset As Integer _
) As Boolean
```

C#

```
public bool MoveRelative(
   int offset
);
```

パラメータ

- **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。

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 クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「IsEOF プロパティ」 682 ページ
- 「IsBOF プロパティ」 681 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。

構文

Visual Basic

```
Public Overrides Function NextResult() As Boolean
```

C#

```
public override bool NextResult();
```


Copyright © 2007, iAnywhere Solutions, Inc.

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。

備考

UL 拡張： Ultra Light.NET ではバッチ SQL 文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。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ

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。

構文

Visual Basic

Public Overrides Function **Read()** As Boolean

C#

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。

備考

このメソッドは、ULDataReader.MoveNext メソッドと同じです。

参照

- 「ULDataReader クラス」 676 ページ
- 「ULDataReader メンバ」 677 ページ
- 「MoveNext メソッド」 711 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULDateOrder enumeration

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。

構文

Visual Basic

Public Enum **ULDateOrder**

C#

public enum **ULDateOrder**

メンバ

メンバ名	説明	値
DMY	日 (日、月、年の順に示す)	2
MDY	月 (月、日、年の順に示す)	1
YMD	年 (年、月、日の順に示す)	0

参照

- 「DateOrder プロパティ」 628 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULDbType 列挙

Ultra Light.NET データベースのデータ型を列挙します。

構文

Visual Basic

Public Enum **ULDbType**

C#

public enum **ULDbType**

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。

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

Copyright © 2007, iAnywhere Solutions, Inc.

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
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
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
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。

メンバ

メンバ名	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、この型は常にユニコード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
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

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
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、この型は常にユニコード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
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
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、この型は常にユニコード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

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「GetFieldType メソッド」 695 ページ
- 「GetDataTypeName メソッド」 692 ページ
- 「GetColumnULDbType メソッド」 643 ページ
- Byte
- Guid
- Boolean
- String
- DateTime
- Single
- Int64
- Int32
- Int16
- TimeSpan
- UInt64
- UInt32
- UInt16

Copyright © 2007, iAnywhere Solutions, Inc.

ULException クラス

Ultra Light.NET データベースによって返される SQL エラーを示します。このクラスは継承できません。

構文

Visual Basic

```
Public NotInheritable Class ULException
  Inherits ApplicationException
```

C#

```
public sealed class ULException : ApplicationException
```

備考

NativeError にエラーが返されたことを示す SQL コード。

.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 メンバ」 719 ページ
- 「NativeError プロパティ」 721 ページ

ULException メンバ

パブリック・コンストラクタ

メンバ名	説明
ULException コンストラクタ	指定されたエラー・コードを持つ ULException を作成します。

パブリック・プロパティ

メンバ名	説明
Data (Exception から継承)	
HelpLink (Exception から継承)	
InnerException (Exception から継承)	
Message (Exception から継承)	
NativeError プロパティ	データベースによって返される SQL コードを返します。
Source プロパティ	エラーを生成したプロバイダの名前を返します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
StackTrace (Exception から継承)	
TargetSite (Exception から継承)	

パブリック・メソッド

メンバ名	説明
GetBaseException (Exception から継承)	
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。
GetType (Exception から継承)	
ToString (Exception から継承)	

参照

- 「ULException クラス」 719 ページ
- 「NativeError プロパティ」 721 ページ

ULException コンストラクタ

指定されたエラー・コードを持つ ULException を作成します。

構文

Visual Basic

```
Public Sub New( _
  ByVal code As ULSQLCode, _
  ByVal s1 As String, _
  ByVal s2 As String, _
  ByVal s3 As String _
)
```

C#

```
public  ULException(
  ULSQLCode code,
  string  s1,
  string  s2,
  string  s3
);
```

パラメータ

- **code**　例外のコード

Copyright © 2007, iAnywhere Solutions, Inc.

- **s1**　フォーマットされたメッセージの 1 番目の文字列
- **s2**　フォーマットされたメッセージの 2 番目の文字列
- **s3**　フォーマットされたメッセージの 3 番目の文字列

備考

指定された iAnywhere.Data.UltraLite.ULSQLCode に対応するメッセージ文字列は、**iAnywhere.Data.UltraLite.resources** アセンブリから取得されます。リソースは、CultureInfo.CurrentUICulture、CultureInfo.CurrentCulture、国 "EN" の順序で国別に検索します。

参照

- 「ULException クラス」 719 ページ
- 「ULException メンバ」 719 ページ
- 「ULSQLCode 列挙」 859 ページ
- CultureInfo.CurrentUICulture
- CultureInfo.CurrentCulture

NativeError プロパティ

データベースによって返される SQL コードを返します。

構文

Visual Basic

Public Readonly Property **NativeError** As ULSQLCode

C#

public ULSQLCode **NativeError** { get;}

プロパティ値

データベースによって返される ULSQLCode 値。

参照

- 「ULException クラス」 719 ページ
- 「ULException メンバ」 719 ページ
- 「ULSQLCode 列挙」 859 ページ

Source プロパティ

エラーを生成したプロバイダの名前を返します。

構文

Visual Basic

Public Overrides Readonly Property **Source** As String

Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public override string  Source  { get;}
```

プロパティ値

プロバイダが Ultra Light.NET であることを示す文字列値。

参照

- 「ULException クラス」 719 ページ
- 「ULException メンバ」 719 ページ

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。

構文

Visual Basic

```
Public Overrides Sub GetObjectData( _
  ByVal info As SerializationInfo, _
  ByVal context As StreamingContext _
)
```

C#

```
public override void GetObjectData(
  SerializationInfo info,
  StreamingContext context
);
```

パラメータ

- **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。

参照

- 「ULException クラス」 719 ページ
- 「ULException メンバ」 719 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULFactory クラス

データ・ソース・クラスの iAnywhere.Data.UltraLite プロバイダの実装のインスタンスを作成する、メソッドのセットを示します。これは静的クラスであるため、継承またはインスタンス化はできません。

構文

Visual Basic

Public NotInheritable Class **ULFactory**
  Inherits DbProviderFactory

C#

public sealed class **ULFactory** : DbProviderFactory

備考

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()

// 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・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。

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

継承： System.Data.Common.DbProviderFactory

参照

- 「ULFactory メンバ」 724 ページ
- DbProviderFactories
- DbProviderFactory

Copyright © 2007, iAnywhere Solutions, Inc.

ULFactory メンバ

パブリック・フィールド

メンバ名	説明
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。このフィールドは読み込み専用です。

パブリック・プロパティ

メンバ名	説明
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。

パブリック・メソッド

メンバ名	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。
CreateDataSourceEnumerator (DbProviderFactory から継承)	
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 (DbProviderFactory から継承)	

参照

- 「ULFactory クラス」 723 ページ
- DbProviderFactories
- DbProviderFactory

Copyright © 2007, iAnywhere Solutions, Inc.

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。このフィールドは読み込み専用です。

構文

Visual Basic

Public Shared Readonly **Instance** As ULFactory

C#

public const ULFactory **Instance** ;

備考

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。詳細については、「ULFactory クラス」 723 ページを参照してください。

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

参照

- 「ULFactory クラス」 723 ページ
- 「ULFactory メンバ」 724 ページ
- DbProviderFactories.GetFactory

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。

構文

Visual Basic

Public Overrides Readonly Property **CanCreateDataSourceEnumerator** As Boolean

C#

public override bool **CanCreateDataSourceEnumerator** { get;}

プロパティ値

ULFactory で CreateDataSourceEnumerator() メソッドを実装していないことを示す false。

参照

- 「ULFactory クラス」 723 ページ
- 「ULFactory メンバ」 724 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Overrides Function **CreateCommand()** As DbCommand

C#

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

- 「ULFactory クラス」 723 ページ
- 「ULFactory メンバ」 724 ページ
- DbCommand
- 「ULCommand クラス」 509 ページ

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。

構文

Visual Basic

Public Overrides Function **CreateCommandBuilder()** As DbCommandBuilder

C#

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

- 「ULFactory クラス」 723 ページ
- 「ULFactory メンバ」 724 ページ
- DbCommandBuilder
- 「ULCommandBuilder クラス」 545 ページ

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 © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Overrides Function **CreateConnection()** As DbConnection

C#

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

- 「ULFactory クラス」 723 ページ
- 「ULFactory メンバ」 724 ページ
- DbConnection
- 「ULConnection クラス」 555 ページ

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。

構文

Visual Basic

Public Overrides Function **CreateConnectionStringBuilder()** As DbConnectionStringBuilder

C#

public override DbConnectionstring Builder **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

- 「ULFactory クラス」 723 ページ
- 「ULFactory メンバ」 724 ページ
- DbConnectionStringBuilder
- 「ULConnectionStringBuilder クラス」 606 ページ

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 © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Overrides Function **CreateDataAdapter()** As DbDataAdapter

C#

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

- 「ULFactory クラス」 723 ページ
- 「ULFactory メンバ」 724 ページ
- DbDataAdapter
- 「ULDataAdapter クラス」 645 ページ

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。

構文

Visual Basic

Public Overrides Function **CreateParameter()** As DbParameter

C#

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

- 「ULFactory クラス」 723 ページ
- 「ULFactory メンバ」 724 ページ
- DbParameter
- 「ULParameter クラス」 773 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULFileTransfer クラス

UL 拡張： Mobile Link サーバを使用して、リモート・データベースからファイルを転送します。このクラスは継承できません。

構文

Visual Basic

Public NotInheritable Class **ULFileTransfer**

C#

public sealed class **ULFileTransfer**

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 メンバ」 729 ページ
- 「RuntimeType プロパティ」 658 ページ
- 「FileName プロパティ」 735 ページ
- 「Stream プロパティ」 738 ページ
- 「UserName プロパティ」 740 ページ
- 「Version プロパティ」 740 ページ

ULFileTransfer メンバ

パブリック・コンストラクタ

メンバ名	説明
ULFileTransfer コンストラクタ	ULFileTransfer オブジェクトを初期化します。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。

パブリック・プロパティ

メンバ名	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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
AuthValue プロパティ	カスタム・ユーザ認証同期スクリプトからの戻り値を返します。
AuthenticationParms プロパティ	カスタム・ユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。
DestinationFileName 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。
DestinationPath 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。
DownloadedFile 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。
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。
ForceDownload プロパティ	ファイルが既に存在する場合に、ファイルのダウンロードを強制するかどうかを指定します。
Password プロパティ	UserName で指定されたユーザの Mobile Link パスワードです。
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を指定します。
UserName プロパティ	Mobile Link サーバが Mobile Link クライアントをユニークに識別するユーザ名です。
Version プロパティ	使用する同期スクリプトを指定します。

パブリック・メソッド

メンバ名	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。

参照

- 「ULFileTransfer クラス」 729 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「RuntimeType プロパティ」 658 ページ
- 「FileName プロパティ」 735 ページ
- 「Stream プロパティ」 738 ページ
- 「UserName プロパティ」 740 ページ
- 「Version プロパティ」 740 ページ

ULFileTransfer コンストラクタ

ULFileTransfer オブジェクトを初期化します。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。

構文

Visual Basic

Public Sub **New()**

C#

public **ULFileTransfer();**

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。

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 クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「DownloadFile() メソッド」 741 ページ
- 「RuntimeType プロパティ」 658 ページ
- 「FileName プロパティ」 735 ページ
- 「Stream プロパティ」 738 ページ
- 「UserName プロパティ」 740 ページ
- 「Version プロパティ」 740 ページ

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **AuthStatus** As ULAuthStatusCode

C#

public ULAuthStatusCode **AuthStatus** { get;}

プロパティ値

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

- 「ULFileTransfer クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「ULAuthStatusCode 列挙」 479 ページ

AuthValue プロパティ

カスタム・ユーザ認証同期スクリプトからの戻り値を返します。

構文

Visual Basic

Public Readonly Property **AuthValue** As Long

C#

public long **AuthValue** { get;}

プロパティ値

カスタム・ユーザ認証同期スクリプトから返された long integer。

参照

- 「ULFileTransfer クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ

AuthenticationParms プロパティ

カスタム・ユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。

構文

Visual Basic

Public Property **AuthenticationParms** As String

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public string **AuthenticationParms** { get; set; }

プロパティ値

それぞれに認証パラメータが格納された、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、認証パラメータは指定されません。

備考

最初の 255 文字のみが使用されます。また、各文字列は 128 文字以下であることが必要です (長すぎる文字列は、Mobile Link サーバに送信されるときにトランケートされます)。

参照

- 「ULFileTransfer クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ

DestinationFileName 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。

構文

Visual Basic

Public Property **DestinationFileName** As String

C#

public string **DestinationFileName** { get; set; }

プロパティ値

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 クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「FileName プロパティ」 735 ページ

DestinationPath 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。

構文

Visual Basic

Public Property **DestinationPath** As String

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public string **DestinationPath** { get; set; }

プロパティ値

ファイルのダウンロード先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。

備考

デフォルトのディレクトリは、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、DestinationPath 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、DestinationPath 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 クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「ForceDownload プロパティ」 736 ページ
- 「DownloadFile() メソッド」 741 ページ

DownloadedFile 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。

構文

Visual Basic

Public Readonly Property **DownloadedFile** As Boolean

C#

public bool **DownloadedFile** { get;}

プロパティ値

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。

備考

DownloadFile() 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、DownloadedFile は false に設定されます。エラーが発生して DownloadFile() が false を返した場合は、DownloadedFile は false に設定されます。

参照

- 「ULFileTransfer クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「DownloadFile() メソッド」 741 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「DownloadFile() メソッド」 741 ページ

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。

構文

Visual Basic

Public Readonly Property **FileAuthCode** As UInt16

C#

public ushort **FileAuthCode** { get;}

プロパティ値

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。

参照

- 「ULFileTransfer クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ

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。

構文

Visual Basic

Public Property **FileName** As String

C#

public string **FileName** { get; set; }

プロパティ値

Mobile Link サーバ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。

備考

FileName は、Mobile Link を実行するサーバ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 サブディレクトリで、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 サーバの -ftr オプションで指定)。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。そのような情報を含めると、ファイルが見からなくなります。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。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULFileTransfer クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「DestinationFileName プロパティ」 733 ページ
- 「DownloadFile() メソッド」 741 ページ

ForceDownload プロパティ

ファイルが既に存在する場合に、ファイルのダウンロードを強制するかどうかを指定します。

構文

Visual Basic

Public Property **ForceDownload** As Boolean

C#

public bool **ForceDownload** { get; set; }

プロパティ値

ファイルが存在する場合に、ファイルのダウンロードを強制する場合は true、通常のダウンロード・チェックを実行する場合は false。デフォルトは false です。

備考

ForceDownload が true であると、変更が加えられていない場合でもファイルは常にダウンロードされ、部分的なダウンロードは破棄されます。ForceDownload が false であると、サーバのバージョンとクライアントのバージョンが異なる場合にのみ、ファイルはダウンロードされます。クライアントまたはサーバでファイルが変更される場合に ForceDownload を true に設定すると、クライアント・バージョンがサーバ・バージョンで上書きされることに注意してください。

参照

- 「ULFileTransfer クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「ResumePartialDownload プロパティ」 737 ページ
- 「DownloadFile() メソッド」 741 ページ

Password プロパティ

UserName で指定されたユーザの Mobile Link パスワードです。

構文

Visual Basic

Public Property **Password** As String

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public string **Password** { get; set; }

プロパティ値

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。

備考

Mobile Link ユーザ名とパスワードは他のデータベース・ユーザ 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 サーバに対して識別し、認証するために使用されます。

参照

- 「ULFileTransfer クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「UserName プロパティ」 740 ページ
- 「DownloadFile() メソッド」 741 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。

構文

Visual Basic

Public Property **ResumePartialDownload** As Boolean

C#

public bool **ResumePartialDownload** { get; set; }

プロパティ値

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。

備考

Ultra Light.NET では、通信エラーや、ユーザによる 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 クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「ForceDownload プロパティ」 736 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「DownloadFile() メソッド」 741 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。

構文

Visual Basic

Public Property **Stream** As ULStreamType

C#

public ULStreamType **Stream** { get; set; }

プロパティ値

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。

備考

ほとんどの同期ストリームでは、Mobile Link サーバ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が必要です。これらのパラメータは、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

- 「ULFileTransfer クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「ULStreamType 列挙」 895 ページ
- 「StreamParms プロパティ」 739 ページ
- 「DownloadFile() メソッド」 741 ページ
- 「ULStreamType 列挙」 895 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。

構文

Visual Basic

Public Readonly Property **StreamErrorCode** As ULStreamErrorCode

C#

public ULStreamErrorCode **StreamErrorCode** { get;}

 Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

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。

参照

- 「ULFileTransfer クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「ULStreamErrorCode 列挙」 871 ページ
- 「ULStreamErrorCode 列挙」 871 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。

構文

Visual Basic

Public Readonly Property **StreamErrorSystem** As Integer

C#

public int **StreamErrorSystem** { get;}

プロパティ値

ストリーム・エラー・システム固有のコードを示す整数。

参照

- 「ULFileTransfer クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ

StreamParms プロパティ

同期ストリームの設定パラメータを指定します。

構文

Visual Basic

Public Property **StreamParms** As String

C#

public string **StreamParms** { get; set; }

プロパティ値

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 © 2007, iAnywhere Solutions, Inc.

備考

特定のストリームのタイプの設定方法については、「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」『Mobile Link - クライアント管理』を参照してください。

StreamParms は、同期ストリームに使用されるすべてのパラメータが含まれている文字列です。パラメータは、名前=値の組み合わせ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 クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「Stream プロパティ」 738 ページ
- 「DownloadFile() メソッド」 741 ページ
- 「ULStreamType 列挙」 895 ページ

UserName プロパティ

Mobile Link サーバが Mobile Link クライアントをユニークに識別するユーザ名です。

構文

Visual Basic

Public Property **UserName** As String

C#

public string **UserName** { get; set; }

プロパティ値

ユーザ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。

備考

Mobile Link サーバは、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 ユーザ名とパスワードは他のデータベース・ユーザ 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 サーバに対して識別し、認証するために使用されます。

参照

- 「ULFileTransfer クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「Password プロパティ」 736 ページ
- 「DownloadFile() メソッド」 741 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **Version** As String

C#

public string **Version** { get; set; }

プロパティ値

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。

備考

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 クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「DownloadFile() メソッド」 741 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。

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。

構文

Visual Basic

Public Function **DownloadFile()** As Boolean

C#

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 サーバによって ULFileTransfer.DestinationPath にダウンロードされます。このとき、ULFileTransfer.Stream、

Copyright © 2007, iAnywhere Solutions, Inc.

ULFileTransfer.UserName、ULFileTransfer.Password、および ULFileTransfer.Version が使用されます。ウンロードに影響を及ぼすその他のプロパティは、ULFileTransfer.DestinationFileName、ULFileTransfer.AuthenticationParmsULFileTransfer.ForceDownload、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対象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.DownloadedFile、ULFileTransfer.StreamErrorCode、および ULFileTransfer.StreamErrorSystem でレポートされます。

参照

- 「ULFileTransfer クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「DownloadFile メソッド」 741 ページ
- 「DownloadFile(ULFileTransferProgressListener) メソッド」 742 ページ
- 「FileName プロパティ」 735 ページ
- 「DestinationPath プロパティ」 733 ページ
- 「Stream プロパティ」 738 ページ
- 「UserName プロパティ」 740 ページ
- 「Password プロパティ」 736 ページ
- 「Version プロパティ」 740 ページ
- 「DestinationFileName プロパティ」 733 ページ
- 「AuthenticationParms プロパティ」 732 ページ
- 「ForceDownload プロパティ」 736 ページ
- 「ResumePartialDownload プロパティ」 737 ページ
- 「AuthStatus プロパティ」 731 ページ
- 「AuthValue プロパティ」 732 ページ
- 「FileAuthCode プロパティ」 735 ページ
- 「DownloadedFile プロパティ」 734 ページ
- 「StreamErrorCode プロパティ」 738 ページ
- 「StreamErrorSystem プロパティ」 739 ページ
- 「StreamErrorCode プロパティ」 738 ページ

DownloadFile(ULFileTransferProgressListener) メソッド

指定されたリスナ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。

構文

Visual Basic

```
Public Function DownloadFile( _
   ByVal listener As ULFileTransferProgressListener _
) As Boolean
```


Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public bool DownloadFile(
  ULFileTransferProgressListener listener
);
```

パラメータ

- **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 サーバによって ULFileTransfer.DestinationPath 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。ウンロードに影響を及ぼすその他のプロパティは、ULFileTransfer.DestinationFileName、ULFileTransfer.AuthenticationParmsULFileTransfer.ForceDownload、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対象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.DownloadedFile、ULFileTransfer.StreamErrorCode、および ULFileTransfer.StreamErrorSystem でレポートされます。

エラーが派生すると、リスナにはデータが送信されないことがあります。

参照

- 「ULFileTransfer クラス」 729 ページ
- 「ULFileTransfer メンバ」 729 ページ
- 「DownloadFile メソッド」 741 ページ
- 「FileName プロパティ」 735 ページ
- 「DestinationPath プロパティ」 733 ページ
- 「Stream プロパティ」 738 ページ
- 「UserName プロパティ」 740 ページ
- 「Password プロパティ」 736 ページ
- 「Version プロパティ」 740 ページ
- 「DestinationFileName プロパティ」 733 ページ
- 「AuthenticationParms プロパティ」 732 ページ
- 「ForceDownload プロパティ」 736 ページ
- 「ResumePartialDownload プロパティ」 737 ページ
- 「AuthStatus プロパティ」 731 ページ
- 「AuthValue プロパティ」 732 ページ
- 「FileAuthCode プロパティ」 735 ページ
- 「DownloadedFile プロパティ」 734 ページ
- 「StreamErrorCode プロパティ」 738 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「StreamErrorSystem プロパティ」 739 ページ
- 「StreamErrorCode プロパティ」 738 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Class **ULFileTransferProgressData**

C#

public class **ULFileTransferProgressData**

参照

- 「ULFileTransferProgressData メンバ」 745 ページ
- 「ULFileTransferProgressListener インタフェース」 749 ページ

ULFileTransferProgressData メンバ

パブリック・フィールド

メンバ名	説明
FLAG_IS_BLOCKING フィールド	Mobile Link サーバ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このフィールドは定数で、読み込み専用です。

パブリック・プロパティ

メンバ名	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。
ResumedAtSize プロパティ	転送が再開されるファイル内の場所を返します。

参照

- 「ULFileTransferProgressData クラス」 745 ページ
- 「ULFileTransferProgressListener インタフェース」 749 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

FLAG_IS_BLOCKING フィールド

Mobile Link サーバ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
このフィールドは定数で、読み込み専用です。

構文

Visual Basic

Public Shared **FLAG_IS_BLOCKING** As Integer

C#

public const int **FLAG_IS_BLOCKING** ;

参照

- 「ULFileTransferProgressData クラス」 745 ページ
- 「ULFileTransferProgressData メンバ」 745 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。

構文

Visual Basic

Public Readonly Property **BytesReceived** As UInt64

C#

public ulong **BytesReceived** { get;}

プロパティ値

現在までに受信したバイト数。

参照

- 「ULFileTransferProgressData クラス」 745 ページ
- 「ULFileTransferProgressData メンバ」 745 ページ

FileSize プロパティ

転送されるファイルのサイズを返します。

構文

Visual Basic

Public Readonly Property **FileSize** As UInt64

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public ulong **FileSize** { get;}

プロパティ値

ファイルのサイズ (バイト単位)。

参照

- 「ULFileTransferProgressData クラス」 745 ページ
- 「ULFileTransferProgressData メンバ」 745 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、現在のファイル転送フラグを返します。

構文

Visual Basic

Public Readonly Property **Flags** As Integer

C#

public int **Flags** { get;}

プロパティ値

フラグの組み合わせを保持する整数。

参照

- 「ULFileTransferProgressData クラス」 745 ページ
- 「ULFileTransferProgressData メンバ」 745 ページ
- 「FLAG_IS_BLOCKING フィールド」 746 ページ

ResumedAtSize プロパティ

転送が再開されるファイル内の場所を返します。

構文

Visual Basic

Public Readonly Property **ResumedAtSize** As UInt64

C#

public ulong **ResumedAtSize** { get;}

プロパティ値

以前に転送されたバイト数。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULFileTransferProgressData クラス」 745 ページ
- 「ULFileTransferProgressData メンバ」 745 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULFileTransferProgressListener インタフェース

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・インタフェースです。

構文

Visual Basic

Public Interface **ULFileTransferProgressListener**

C#

public interface **ULFileTransferProgressListener**

参照

- 「ULFileTransferProgressListener メンバ」 749 ページ
- 「DownloadFile(ULFileTransferProgressListener) メソッド」 742 ページ

ULFileTransferProgressListener メンバ

パブリック・メソッド

メンバ名	説明
FileTransferProgressed メソッド	ユーザ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。このメソッドは true を返して転送をキャンセルするか、false を返して続行する必要があります。

参照

- 「ULFileTransferProgressListener インタフェース」 749 ページ
- 「DownloadFile(ULFileTransferProgressListener) メソッド」 742 ページ

FileTransferProgressed メソッド

ユーザ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。このメソッドは true を返して転送をキャンセルするか、false を返して続行する必要があります。

構文

Visual Basic

```
Public Function FileTransferProgressed( _
  ByVal data As ULFileTransferProgressData _
) As Boolean
```

C#

```
public bool FileTransferProgressed(
```

```
  ULFileTransferProgressData data
);
```

パラメータ

- **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 true を返して転送をキャンセルするか、false を返して続行する必要があります。

備考

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。

参照

- 「ULFileTransferProgressListener インタフェース」 749 ページ
- 「ULFileTransferProgressListener メンバ」 749 ページ
- 「ULFileTransferProgressData クラス」 745 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。このクラスは継承できません。

構文

Visual Basic

Public NotInheritable Class **ULIndexSchema**

C#

public sealed class **ULIndexSchema**

備考

このクラスにはコンストラクタがありません。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

- 「ULIndexSchema メンバ」 751 ページ
- 「PrimaryKey プロパティ」 958 ページ
- 「GetIndex メソッド」 960 ページ
- 「GetOptimalIndex メソッド」 962 ページ
- 「ULTableSchema クラス」 954 ページ

ULIndexSchema メンバ

パブリック・プロパティ

メンバ名	説明
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 入力可能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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	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。

パブリック・メソッド

メンバ名	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。

参照

- 「ULIndexSchema クラス」 751 ページ
- 「PrimaryKey プロパティ」 958 ページ
- 「GetIndex メソッド」 960 ページ
- 「GetOptimalIndex メソッド」 962 ページ
- 「ULTableSchema クラス」 954 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。

構文

Visual Basic

Public Readonly Property **ColumnCount** As Short

C#

public short **ColumnCount** { get;}

プロパティ値

インデックス内のカラム数。

備考

インデックス内のカラム序数の範囲は、1 ー ColumnCount です。

Copyright © 2007, iAnywhere Solutions, Inc.

カラムの序数とカウントは、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序数は、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 クラス」 751 ページ
- 「ULIndexSchema メンバ」 751 ページ

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。

構文

Visual Basic

Public Readonly Property **IsForeignKey** As Boolean

C#

public bool **IsForeignKey** { get;}

プロパティ値

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。

備考

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。

参照

- 「ULIndexSchema クラス」 751 ページ
- 「ULIndexSchema メンバ」 751 ページ

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。

構文

Visual Basic

Public Readonly Property **IsForeignKeyCheckOnCommit** As Boolean

C#

public bool **IsForeignKeyCheckOnCommit** { get;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

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 クラス」 751 ページ
- 「ULIndexSchema メンバ」 751 ページ
- 「IsForeignKey プロパティ」 753 ページ

IsForeignKeyNullable プロパティ

外部キーが null 入力可能であるかどうかをチェックします。

構文

Visual Basic

Public Readonly Property **IsForeignKeyNullable** As Boolean

C#

public bool **IsForeignKeyNullable** { get;}

プロパティ値

外部キーが null 入力可能な場合は true、null 入力不可の場合は false。

参照

- 「ULIndexSchema クラス」 751 ページ
- 「ULIndexSchema メンバ」 751 ページ
- 「IsForeignKey プロパティ」 753 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。

構文

Visual Basic

Public Readonly Property **IsOpen** As Boolean

C#

public bool **IsOpen** { get;}

プロパティ値

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。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULIndexSchema クラス」 751 ページ
- 「ULIndexSchema メンバ」 751 ページ

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。

構文

Visual Basic

Public Readonly Property **IsPrimaryKey** As Boolean

C#

public bool **IsPrimaryKey** { get;}

プロパティ値

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。

備考

プライマリ・キー内のカラムでは null は許可されません。

参照

- 「ULIndexSchema クラス」 751 ページ
- 「ULIndexSchema メンバ」 751 ページ

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。

構文

Visual Basic

Public Readonly Property **IsUniqueIndex** As Boolean

C#

public bool **IsUniqueIndex** { get;}

プロパティ値

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。

備考

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。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULIndexSchema クラス」 751 ページ
- 「ULIndexSchema メンバ」 751 ページ

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。

構文

Visual Basic

Public Readonly Property **IsUniqueKey** As Boolean

C#

public bool **IsUniqueKey** { get;}

プロパティ値

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。

備考

ユニーク・キー内のカラムでは null は許可されません。

参照

- 「ULIndexSchema クラス」 751 ページ
- 「ULIndexSchema メンバ」 751 ページ

Name プロパティ

インデックスの名前を返します。

構文

Visual Basic

Public Readonly Property **Name** As String

C#

public string **Name** { get;}

プロパティ値

インデックスの名前を指定する文字列。

参照

- 「ULIndexSchema クラス」 751 ページ
- 「ULIndexSchema メンバ」 751 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Readonly Property **ReferencedIndexName** As String

C#

public string **ReferencedIndexName** { get;}

プロパティ値

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 クラス」 751 ページ
- 「ULIndexSchema メンバ」 751 ページ
- 「IsForeignKey プロパティ」 753 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。

構文

Visual Basic

Public Readonly Property **ReferencedTableName** As String

C#

public string **ReferencedTableName** { get;}

プロパティ値

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 クラス」 751 ページ
- 「ULIndexSchema メンバ」 751 ページ
- 「IsForeignKey プロパティ」 753 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Function GetColumnName( _
  ByVal colOrdinalInIndex As Short _
) As String
```

C#

```
public string  GetColumnName(
  short colOrdinalInIndex
);
```

パラメータ

- **colOrdinalInIndex**　インデックス内の対象のカラムの序数。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

備考

カラムの序数とカウントは、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序数は、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 クラス」 751 ページ
- 「ULIndexSchema メンバ」 751 ページ
- 「ColumnCount プロパティ」 752 ページ
- 「ColumnCount プロパティ」 752 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。

構文

Visual Basic

```
Public Function IsColumnDescending( _
  ByVal name As String _
) As Boolean
```

C#

```
public bool IsColumnDescending(
  string  name
);
```


Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **name**　カラムの名前。

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。

参照

- 「ULIndexSchema クラス」 751 ページ
- 「ULIndexSchema メンバ」 751 ページ
- 「GetColumnName メソッド」 757 ページ
- 「ColumnCount プロパティ」 752 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULInfoMessageEventArgs クラス

ULConnection.InfoMessage イベントのデータを提供します。このクラスは継承できません。

構文

Visual Basic

Public NotInheritable Class **ULInfoMessageEventArgs**
  Inherits EventArgs

C#

public sealed class **ULInfoMessageEventArgs** : EventArgs

参照

- 「ULInfoMessageEventArgs メンバ」 760 ページ
- 「InfoMessage イベント」 588 ページ

ULInfoMessageEventArgs メンバ

パブリック・プロパティ

メンバ名	説明
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 SQL コードです。
Source プロパティ	メッセージを返す ADO.NET データ・プロバイダの名前です。

パブリック・メソッド

メンバ名	説明
ToString メソッド	ULConnection.InfoMessage イベントの文字列表現を返します。

参照

- 「ULInfoMessageEventArgs クラス」 760 ページ
- 「InfoMessage イベント」 588 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **Message** As String

C#

public string **Message** { get;}

プロパティ値

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。

参照

- 「ULInfoMessageEventArgs クラス」 760 ページ
- 「ULInfoMessageEventArgs メンバ」 760 ページ

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 SQL コードです。

構文

Visual Basic

Public Readonly Property **NativeError** As ULSQLCode

C#

public ULSQLCode **NativeError** { get;}

プロパティ値

情報メッセージまたは警告の ULSQLCode 値。

参照

- 「ULInfoMessageEventArgs クラス」 760 ページ
- 「ULInfoMessageEventArgs メンバ」 760 ページ
- 「ULSQLCode 列挙」 859 ページ

Source プロパティ

メッセージを返す ADO.NET データ・プロバイダの名前です。

構文

Visual Basic

Public Readonly Property **Source** As String

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public string **Source** { get;}

プロパティ値

文字列 "Ultra Light.NET Data Provider"。

参照

- 「ULInfoMessageEventArgs クラス」 760 ページ
- 「ULInfoMessageEventArgs メンバ」 760 ページ

ToString メソッド

ULConnection.InfoMessage イベントの文字列表現を返します。

構文

Visual Basic

Public Overrides Function **ToString()** As String

C#

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。

参照

- 「ULInfoMessageEventArgs クラス」 760 ページ
- 「ULInfoMessageEventArgs メンバ」 760 ページ
- 「InfoMessage イベント」 588 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULInfoMessageEventHandler 委任

ULConnection.InfoMessage イベントを処理するメソッドを示します。

構文

Visual Basic

```
Public Delegate Sub ULInfoMessageEventHandler( _
  ByVal obj As Object, _
  ByVal args As ULInfoMessageEventArgs _
)
```

C#

```
public delegate void ULInfoMessageEventHandler(
  object obj,
  ULInfoMessageEventArgs args
);
```

参照

- 「InfoMessage イベント」 588 ページ
- 「ULInfoMessageEventArgs クラス」 760 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。このクラスは継承できません。

構文

Visual Basic

Public NotInheritable Class **ULMetaDataCollectionNames**

C#

public sealed class **ULMetaDataCollectionNames**

参照

- 「ULMetaDataCollectionNames メンバ」 764 ページ
- 「GetSchema(String, String[]) メソッド」 581 ページ

ULMetaDataCollectionNames メンバ

パブリック・プロパティ

メンバ名	説明
Columns プロパティ	Columns コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。
DataSourceInformation プロパティ	DataSourceInformation コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。
DataTypes プロパティ	DataTypes コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。
ForeignKeys プロパティ	ForeignKeys コレクションを示す ULConnection.GetSchema (String) で使用する定数を提供します。
IndexColumns プロパティ	IndexColumns コレクションを示す ULConnection.GetSchema (String) で使用する定数を提供します。
Indexes プロパティ	Indexes コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。
MetaDataCollections プロパティ	MetaDataCollections コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。
Publications プロパティ	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。
ReservedWords プロパティ	ReservedWords コレクションを示す ULConnection.GetSchema (String) で使用する定数を提供します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
Restrictions プロパティ	Restrictions コレクションを示す ULConnection.GetSchema (String) で使用する定数を提供します。
Tables プロパティ	Tables コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。

参照

- 「ULMetaDataCollectionNames クラス」 764 ページ
- 「GetSchema(String, String[]) メソッド」 581 ページ

Columns プロパティ

Columns コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。

構文

Visual Basic

Public Shared Readonly Property **Columns** As String

C#

public const string **Columns** { get;}

プロパティ値

Columns コレクションの名前を示す文字列。

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 )

// C#
DataTable schema =
  conn.GetSchema( ULMetaDataCollectionNames.Columns );
```

参照

- 「ULMetaDataCollectionNames クラス」 764 ページ
- 「ULMetaDataCollectionNames メンバ」 764 ページ
- 「GetSchema(String) メソッド」 581 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

DataSourceInformation プロパティ

DataSourceInformation コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。

構文

Visual Basic

Public Shared Readonly Property **DataSourceInformation** As String

C#

public const string **DataSourceInformation** { get;}

プロパティ値

DataSourceInformation コレクションの名前を示す文字列。

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 )

// C#
DataTable schema =
  conn.GetSchema( ULMetaDataCollectionNames.DataSourceInformation );
```

参照

- 「ULMetaDataCollectionNames クラス」 764 ページ
- 「ULMetaDataCollectionNames メンバ」 764 ページ
- 「GetSchema(String) メソッド」 581 ページ

DataTypes プロパティ

DataTypes コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。

構文

Visual Basic

Public Shared Readonly Property **DataTypes** As String

C#

public const string **DataTypes** { get;}

プロパティ値

DataTypes コレクションの名前を示す文字列。

Copyright © 2007, iAnywhere Solutions, Inc.

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 )

// C#
DataTable schema =
  conn.GetSchema( ULMetaDataCollectionNames.DataTypes );
```

参照

- 「ULMetaDataCollectionNames クラス」 764 ページ
- 「ULMetaDataCollectionNames メンバ」 764 ページ
- 「GetSchema(String) メソッド」 581 ページ

ForeignKeys プロパティ

ForeignKeys コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。

構文

Visual Basic

Public Shared Readonly Property **ForeignKeys** As String

C#

public const string **ForeignKeys** { get;}

プロパティ値

ForeignKeys コレクションの名前を示す文字列。

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 )

// C#
DataTable schema =
  conn.GetSchema( ULMetaDataCollectionNames.ForeignKeys );
```

参照

- 「ULMetaDataCollectionNames クラス」 764 ページ
- 「ULMetaDataCollectionNames メンバ」 764 ページ
- 「GetSchema(String) メソッド」 581 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

IndexColumns プロパティ

IndexColumns コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。

構文

Visual Basic

Public Shared Readonly Property **IndexColumns** As String

C#

public const string **IndexColumns** { get;}

プロパティ値

IndexColumns コレクションの名前を示す文字列。

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 )

// C#
DataTable schema =
  conn.GetSchema( ULMetaDataCollectionNames.IndexColumns );
```

参照

- 「ULMetaDataCollectionNames クラス」 764 ページ
- 「ULMetaDataCollectionNames メンバ」 764 ページ
- 「GetSchema(String) メソッド」 581 ページ

Indexes プロパティ

Indexes コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。

構文

Visual Basic

Public Shared Readonly Property **Indexes** As String

C#

public const string **Indexes** { get;}

プロパティ値

Indexes コレクションの名前を示す文字列。

Copyright © 2007, iAnywhere Solutions, Inc.

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 )

// C#
DataTable schema =
  conn.GetSchema( ULMetaDataCollectionNames.Indexes );
```

参照

- 「ULMetaDataCollectionNames クラス」 764 ページ
- 「ULMetaDataCollectionNames メンバ」 764 ページ
- 「GetSchema(String) メソッド」 581 ページ

MetaDataCollections プロパティ

MetaDataCollections コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。

構文

Visual Basic

Public Shared Readonly Property **MetaDataCollections** As String

C#

public const string **MetaDataCollections** { get;}

プロパティ値

MetaDataCollections コレクションの名前を示す文字列。

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 )

// C#
DataTable schema =
  conn.GetSchema( ULMetaDataCollectionNames.MetaDataCollections );
```

参照

- 「ULMetaDataCollectionNames クラス」 764 ページ
- 「ULMetaDataCollectionNames メンバ」 764 ページ
- 「GetSchema(String) メソッド」 581 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

Publications プロパティ

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。

構文

Visual Basic

Public Shared Readonly Property **Publications** As String

C#

public const string **Publications** { get;}

参照

- 「ULMetaDataCollectionNames クラス」 764 ページ
- 「ULMetaDataCollectionNames メンバ」 764 ページ
- 「GetSchema(String, String[]) メソッド」 581 ページ

ReservedWords プロパティ

ReservedWords コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。

構文

Visual Basic

Public Shared Readonly Property **ReservedWords** As String

C#

public const string **ReservedWords** { get;}

プロパティ値

ReservedWords コレクションの名前を示す文字列。

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 )

// C#
DataTable schema =
  conn.GetSchema( ULMetaDataCollectionNames.ReservedWords );
```

参照

- 「ULMetaDataCollectionNames クラス」 764 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULMetaDataCollectionNames メンバ」 764 ページ
- 「GetSchema(String) メソッド」 581 ページ

Restrictions プロパティ

Restrictions コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。

構文

Visual Basic

Public Shared Readonly Property **Restrictions** As String

C#

public const string **Restrictions** { get;}

プロパティ値

Restrictions コレクションの名前を示す文字列。

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 )

// C#
DataTable schema =
  conn.GetSchema( ULMetaDataCollectionNames.Restrictions );
```

参照

- 「ULMetaDataCollectionNames クラス」 764 ページ
- 「ULMetaDataCollectionNames メンバ」 764 ページ
- 「GetSchema(String) メソッド」 581 ページ

Tables プロパティ

Tables コレクションを示す ULConnection.GetSchema(String) で使用する定数を提供します。

構文

Visual Basic

Public Shared Readonly Property **Tables** As String

C#

public const string **Tables** { get;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

Tables コレクションの名前を示す文字列。

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 )

// C#
DataTable schema =
  conn.GetSchema( ULMetaDataCollectionNames.Tables );
```

参照

- 「ULMetaDataCollectionNames クラス」 764 ページ
- 「ULMetaDataCollectionNames メンバ」 764 ページ
- 「GetSchema(String) メソッド」 581 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULParameter クラス

ULCommand のパラメータを示します。このクラスは継承できません。

構文

Visual Basic

```
Public NotInheritable Class ULParameter
  Inherits DbParameter
  Implements ICloneable
```

C#

```
public sealed class ULParameter : DbParameter,
  ICloneable
```

備考

ULParameter オブジェクトは、その多数のコンストラクタ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 ) )

// C#
ULParameter p = new ULParameter( "", (object)0 );
```

パラメータ (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は、位置パラメータとして扱われ、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、パラメータは IN パラメータ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。

継承： System.Data.Common.DbParameter

実装： System.Data.IDbDataParameter、System.Data.IDataParameter

参照

- 「ULParameter メンバ」 774 ページ
- 「ULCommand クラス」 509 ページ
- 「CreateParameter メソッド」 528 ページ
- 「ULParameter(String, Object) コンストラクタ」 776 ページ
- 「Parameters プロパティ」 520 ページ
- 「Value プロパティ」 788 ページ
- DbParameter
- IDbDataParameter
- IDataParameter

Copyright © 2007, iAnywhere Solutions, Inc.

ULParameter メンバ

パブリック・コンストラクタ

メンバ名	説明
ULParameter コンストラクタ	「ULParameter クラス」 773 ページの新しいインスタンスを初期化します。

パブリック・プロパティ

メンバ名	説明
DbType プロパティ	パラメータの System.Data.DbType を指定します。
Direction プロパティ	パラメータが入力専用、出力専用、双方向性、またはストアド・プロシージャ戻り値パラメータのいずれであるかを示す値です。
IsNullable プロパティ	パラメータが null 値を受け入れるかどうかを指定します。
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	パラメータの名前を指定します。
Precision プロパティ	ULParameter.Value を示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	
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	パラメータの iAnywhere.Data.UltraLite.ULDbType を指定します。
Value プロパティ	パラメータの値を指定します。

パブリック・メソッド

メンバ名	説明
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現を返します。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULParameter クラス」 773 ページ
- 「ULCommand クラス」 509 ページ
- 「CreateParameter メソッド」 528 ページ
- 「ULParameter(String, Object) コンストラクタ」 776 ページ
- 「Parameters プロパティ」 520 ページ
- 「Value プロパティ」 788 ページ
- DbParameter
- IDbDataParameter
- IDataParameter

ULParameter コンストラクタ

「ULParameter クラス」 773 ページの新しいインスタンスを初期化します。

ULParameter() コンストラクタ

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。

構文

Visual Basic

Public Sub **New()**

C#

public **ULParameter();**

例

次のコードでは、値が 3 である ULParameter パラメータが作成され、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 )

// C#
ULParameter p = new ULParameter();
p.Value = 3;
cmd.Parameters.Add( p );
```

参照

- 「ULParameter クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「ULParameter コンストラクタ」 775 ページ
- 「Value プロパティ」 788 ページ
- 「ULParameter(String, Object) コンストラクタ」 776 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULCommand クラス」 509 ページ

ULParameter(String, Object) コンストラクタ

ULParameter オブジェクトを、指定されたパラメータ名と値で初期化します。

構文

Visual Basic

```
Public Sub New( _
  ByVal parameterName As String, _
  ByVal value As Object _
)
```

C#

```
public  ULParameter(
  string  parameterName,
  object value
);
```

パラメータ

- **parameterName**　パラメータの名前。名前のないパラメータの場合、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 はパラメータの名前を使用しません。
- **value**　パラメータの値である System.Object。

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。

例

次のコードでは、値が 0 である ULParameter パラメータが作成され、cmd という ULCommand が追加されます。

```
' Visual Basic
cmd.Parameters.Add( New ULParameter( "", CType( 0, Object ) ) )

// C#
cmd.Parameters.Add( new ULParameter( "", (object)0 ) );
```

参照

- 「ULParameter クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「ULParameter コンストラクタ」 775 ページ
- 「ULParameter() コンストラクタ」 775 ページ
- 「ULCommand クラス」 509 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- オブジェクト

ULParameter(String, ULDbType) コンストラクタ

ULParameter オブジェクトを、指定されたパラメータ名とデータ型で初期化します。このコンストラクタの使用はおすすめしません。これは、他のデータ・プロバイダとの互換性のために用意されています。

構文

Visual Basic

```
Public Sub New( _
  ByVal parameterName As String, _
  ByVal dbType As ULDbType _
)
```

C#

```
public  ULParameter(
  string  parameterName,
  ULDbType dbType
);
```

パラメータ

- **parameterName**　パラメータの名前。名前のないパラメータの場合、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 はパラメータの名前を使用しません。
- **dbType**　iAnywhere.Data.UltraLite.ULDbType 値の 1 つ。

備考

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「ULParameter コンストラクタ」 775 ページ
- 「ULParameter() コンストラクタ」 775 ページ
- 「ULParameter(String, Object) コンストラクタ」 776 ページ
- 「Value プロパティ」 788 ページ
- 「ULCommand クラス」 509 ページ
- 「ULDbType 列挙」 715 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULParameter(String, ULDbType, Int32) コンストラクタ

ULParameter オブジェクトを、指定されたパラメータ名とデータ型で初期化します。このコンストラクタの使用はおすすめしません。これは、他のデータ・プロバイダとの互換性のために用意されています。

構文

Visual Basic

```
Public Sub New( _
  ByVal parameterName As String, _
  ByVal dbType As ULDbType, _
  ByVal size As Integer _
)
```

C#

```
public  ULParameter(
  string  parameterName,
  ULDbType dbType,
  int size
);
```

パラメータ

- **parameterName**　パラメータの名前。名前のないパラメータの場合、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**　パラメータの長さ。

備考

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「ULParameter コンストラクタ」 775 ページ
- 「ULParameter() コンストラクタ」 775 ページ
- 「ULParameter(String, Object) コンストラクタ」 776 ページ
- 「Value プロパティ」 788 ページ
- 「ULDbType 列挙」 715 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULParameter(String, ULDbType, Int32, String) コンストラクタ

ULParameter オブジェクトを、指定されたパラメータ名、データ型、長さで初期化します。このコンストラクタの使用はおすすめしません。これは、他のデータ・プロバイダとの互換性のために用意されています。

構文

Visual Basic

```
Public Sub New( _
  ByVal parameterName As String, _
  ByVal dbType As ULDbType, _
  ByVal size As Integer, _
  ByVal sourceColumn As String _
)
```

C#

```
public  ULParameter(
  string  parameterName,
  ULDbType dbType,
  int size,
  string  sourceColumn
);
```

パラメータ

- **parameterName**　パラメータの名前。名前のないパラメータの場合、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**　パラメータの長さ。
- **sourceColumn**　マッピングするソース・カラムの名前。

備考

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「ULParameter コンストラクタ」 775 ページ
- 「ULParameter() コンストラクタ」 775 ページ
- 「ULParameter(String, Object) コンストラクタ」 776 ページ
- 「Value プロパティ」 788 ページ
- 「ULDbType 列挙」 715 ページ
- 「ULCommand クラス」 509 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULParameter(String, ULDbType, Int32, ParameterDirection, Boolean, Byte, Byte, String, DataRowVersion, Object) コンストラクタ

指定されたパラメータ名、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。このコンストラクタの使用はおすすめしません。これは、他のデータ・プロバイダとの互換性のために用意されています。

構文

Visual Basic

```
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#

```
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

- **parameterName** パラメータの名前。名前のないパラメータの場合、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。

Copyright © 2007, iAnywhere Solutions, Inc.

- **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**　パラメータの値である System.Object。

参照

- 「ULParameter クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「ULParameter コンストラクタ」 775 ページ
- 「ULParameter() コンストラクタ」 775 ページ
- 「ULParameter(String, Object) コンストラクタ」 776 ページ
- 「ULDbType 列挙」 715 ページ
- ParameterDirection
- DataRowVersion
- オブジェクト
- ParameterDirection.Input
- 「ULCommand クラス」 509 ページ

DbType プロパティ

パラメータの System.Data.DbType を指定します。

構文

Visual Basic

```
Public Overrides Property DbType As DbType
```

C#

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

プロパティ値

System.Data.DbType 値の 1 つ。

備考

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、サポートされている iAnywhere.Data.UltraLite.ULDbType に ULParameter.ULDbType を変更します。

参照

- 「ULParameter クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- DbType
- DbType
- 「ULDbType プロパティ」 787 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULDbType 列挙」 715 ページ

Direction プロパティ

パラメータが入力専用、出力専用、双方向性、またはストアド・プロシージャ戻り値パラメータのいずれであるかを示す値です。

構文

Visual Basic

Public Overrides Property **Direction** As ParameterDirection

C#

public override ParameterDirection **Direction** { get; set; }

プロパティ値

System.Data.ParameterDirection 値の 1 つ。

備考

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- ParameterDirection
- ParameterDirection.Input
- 「Value プロパティ」 788 ページ

IsNullable プロパティ

パラメータが null 値を受け入れるかどうかを指定します。

構文

Visual Basic

Public Overrides Property **IsNullable** As Boolean

C#

public override bool **IsNullable** { get; set; }

プロパティ値

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。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「Value プロパティ」 788 ページ

Offset プロパティ

ULParameter.Value のオフセットを指定します。

構文

Visual Basic

Public Property **Offset** As Integer

C#

public int **Offset** { get; set; }

プロパティ値

値のオフセット。デフォルトは 0 です。

備考

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「Value プロパティ」 788 ページ

ParameterName プロパティ

パラメータの名前を指定します。

構文

Visual Basic

Public Overrides Property **ParameterName** As String

C#

public override string **ParameterName** { get; set; }

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

パラメータの名前を示す文字列。名前のないパラメータ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。

備考

Ultra Light.NET では、ULCommand はパラメータの名前を使用しません。すべてのパラメータは、位置パラメータとして扱われ、コマンドによって追加された順序で使用されます。

参照

- 「ULParameter クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「ULCommand クラス」 509 ページ

Precision プロパティ

ULParameter.Value を示すために使用される最大桁数を指定します。

構文

Visual Basic

Public Property **Precision** As Byte

C#

public byte **Precision** { get; set; }

プロパティ値

ULParameter.Value を示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・プロバイダが ULParameter.Value の精度を設定することを示します。

備考

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「Value プロパティ」 788 ページ

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **Scale** As Byte

C#

public byte **Scale** { get; set; }

プロパティ値

ULParameter.Value が解決される小数点の桁の数。デフォルトは 0 です。

備考

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「Value プロパティ」 788 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。

構文

Visual Basic

Public Overrides Property **Size** As Integer

C#

public override int **Size** { get; set; }

プロパティ値

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。

備考

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「Value プロパティ」 788 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Overrides Property **SourceColumn** As String

C#

public override string **SourceColumn** { get; set; }

プロパティ値

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。

備考

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「Value プロパティ」 788 ページ

SourceColumnNullMapping プロパティ

構文

Visual Basic

Public Overrides Property **SourceColumnNullMapping** As Boolean

C#

public override bool **SourceColumnNullMapping** { get; set; }

参照

- 「ULParameter クラス」 773 ページ
- 「ULParameter メンバ」 774 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Overrides Property **SourceVersion** As DataRowVersion

C#

public override DataRowVersion **SourceVersion** { get; set; }

参照

- 「ULParameter クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「Value プロパティ」 788 ページ
- DataRowVersion

ULDbType プロパティ

パラメータの iAnywhere.Data.UltraLite.ULDbType を指定します。

構文

Visual Basic

Public Property **ULDbType** As ULDbType

C#

public ULDbType **ULDbType** { get; set; }

プロパティ値

iAnywhere.Data.UltraLite.ULDbType 値の 1 つ。

備考

ULDbType と ULParameter.DbType はリンクされます。このため、ULDbType を設定すると、サポートされている System.Data.DbType に ULParameter.DbType を変更します。

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「Value プロパティ」 788 ページ
- 「DbType プロパティ」 781 ページ
- DbType
- 「ULDbType 列挙」 715 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

Value プロパティ

パラメータの値を指定します。

構文

Visual Basic

Public Overrides Property **Value** As Object

C#

public override object **Value** { get; set; }

プロパティ値

パラメータの値を指定する System.Object。

備考

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

- 「ULParameter クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ
- 「ULException クラス」 719 ページ
- 「ULSQLCode 列挙」 859 ページ
- オブジェクト

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。

構文

Visual Basic

Public Overrides Sub **ResetDbType()**

C#

public override void **ResetDbType();**

備考

Ultra Light.NET では、パラメータは IN パラメータ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 クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「Value プロパティ」 788 ページ

ToString メソッド

このインスタンスの文字列表現を返します。

構文

Visual Basic

Public Overrides Function **ToString()** As String

C#

public override string **ToString();**

戻り値

パラメータの名前。

参照

- 「ULParameter クラス」 773 ページ
- 「ULParameter メンバ」 774 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULParameterCollection クラス

ULCommand のすべてのパラメータを示します。このクラスは継承できません。

構文

Visual Basic

```
Public NotInheritable Class ULParameterCollection
  Inherits DbParameterCollection
```

C#

```
public sealed class ULParameterCollection : DbParameterCollection
```

備考

コレクション内のすべてのパラメータは、位置パラメータとして扱われ、ULCommand.CommandText の疑問符のプレースホルダ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の数と同じでなければなりません。パラメータ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がありません。ULParameterCollection は、ULCommand.Parameters から取得します。

継承： System.Data.Common.DbParameterCollection

実装： System.Data.IDataParameterCollection

参照

- 「ULParameterCollection メンバ」 790 ページ
- 「ULCommand クラス」 509 ページ
- 「CommandText プロパティ」 516 ページ
- 「Parameters プロパティ」 520 ページ
- DbParameterCollection
- IDataParameterCollection

ULParameterCollection メンバ

パブリック・プロパティ

メンバ名	説明
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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。
Item プロパティ	コレクション内の DbParameter を取得または設定します。
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。

パブリック・メソッド

メンバ名	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をコレクションから削除します。
Contains メソッド	特定のプロパティが設定された DbParameter がコレクション内に存在するかどうかを示します。
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。
IndexOf メソッド	指定された DbParameter オブジェクトのインデックスを返します。
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	指定された DbParameter オブジェクトをコレクションから削除します。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULCommand クラス」 509 ページ
- 「CommandText プロパティ」 516 ページ
- 「Parameters プロパティ」 520 ページ
- DbParameterCollection
- IDataParameterCollection

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Overrides Readonly Property **Count** As Integer

C#

public override int **Count** { get;}

プロパティ値

コレクション内の ULParameter オブジェクトの数。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ

IsFixedSize プロパティ

ULParameterCollection のサイズが固定かどうかを示します。

構文

Visual Basic

Public Overrides Readonly Property **IsFixedSize** As Boolean

C#

public override bool **IsFixedSize** { get;}

プロパティ値

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。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Overrides Readonly Property **IsReadOnly** As Boolean

C#

public override bool **IsReadOnly** { get;}

プロパティ値

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。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ

IsSynchronized プロパティ

ULParameterCollection が同期されるかどうかを示します。

構文

Visual Basic

Public Overrides Readonly Property **IsSynchronized** As Boolean

C#

public override bool **IsSynchronized** { get;}

プロパティ値

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。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ

Item プロパティ

コレクション内の DbParameter を取得または設定します。

Item(Int32) プロパティ

指定されたインデックスの ULParameter を返します。C# では、このプロパティは ULParameterCollection クラスのインデクサです。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Property Item ( _
  ByVal index As Integer _
) As ULParameter
```

C#

```
public ULParameter this [
  int index
] { get; set; }
```

パラメータ

- **index**　取り出すパラメータの 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のインデックス値はゼロです。

プロパティ値

指定されたインデックス位置の ULParameter。

備考

これは、DbParameterCollection.this[int] が厳密に型指定されたものです。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「Item プロパティ」 793 ページ
- 「ULParameter クラス」 773 ページ
- DbParameterCollection.Item

Item(String) プロパティ

指定された名前の ULParameter を返します。C# では、このプロパティは ULParameterCollection クラスのインデクサです。

構文

Visual Basic

```
Public Property Item ( _
  ByVal parameterName As String _
) As ULParameter
```

C#

```
public ULParameter this [
  string  parameterName
] { get; set; }
```


Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **parameterName**　取り出すパラメータの名前。

プロパティ値

指定された名前の ULParameter。

備考

これは、DbParameterCollection.this[string] が厳密に型指定されたものです。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「Item プロパティ」 793 ページ
- 「Item(Int32) プロパティ」 683 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetValue メソッド」 707 ページ
- 「GetFieldType メソッド」 695 ページ
- 「ULParameter クラス」 773 ページ

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。

構文

Visual Basic

Public Overrides Readonly Property **SyncRoot** As Object

C#

public override object **SyncRoot** { get;}

プロパティ値

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。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ

Add メソッド

ULParameter をコレクションに追加します。

Copyright © 2007, iAnywhere Solutions, Inc.

Add(Object) メソッド

ULParameter をコレクションに追加します。

構文

Visual Basic

```
Public Overrides Function Add( _
  ByVal value As Object _
) As Integer
```

C#

```
public override int Add(
  object value
);
```

パラメータ

- **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。

備考

コレクション内のすべてのパラメータは、位置パラメータとして扱われ、ULCommand.CommandText 内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。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の数と同じでなければなりません。パラメータが不足している場合は、null が代用されます。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「Add メソッド」 795 ページ
- 「Add(ULParameter) メソッド」 796 ページ
- 「Add(String, Object) メソッド」 797 ページ
- 「CommandText プロパティ」 516 ページ
- 「ULParameter クラス」 773 ページ

Add(ULParameter) メソッド

ULParameter をコレクションに追加します。

構文

Visual Basic

```
Public Function Add( _
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
  ByVal value As ULParameter _
) As ULParameter
```

C#

```
public ULParameter Add(
  ULParameter value
);
```

パラメータ

- **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。

備考

コレクション内のすべてのパラメータは、位置パラメータとして扱われ、ULCommand.CommandText 内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。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の数と同じでなければなりません。パラメータが不足している場合は、null が代用されます。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「Add メソッド」 795 ページ
- 「Add(String, Object) メソッド」 797 ページ
- 「ULParameter クラス」 773 ページ
- 「CommandText プロパティ」 516 ページ

Add(String, Object) メソッド

指定されたパラメータ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。

構文

Visual Basic

```
Public Function Add( _
  ByVal parameterName As String, _
  ByVal value As Object _
) As ULParameter
```

C#

```
public ULParameter Add(
  string  parameterName,
```

```
  object value
);
```

パラメータ

- **parameterName**　パラメータの名前。名前のないパラメータの場合、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 はパラメータの名前を使用しません。
- **value**　パラメータの値である 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は、位置パラメータとして扱われ、ULCommand.CommandText 内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。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の数と同じでなければなりません。パラメータ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。

例

次のコードでは、値が 0 である ULParameter パラメータが cmd という ULCommand に追加されます。

```
' Visual Basic
cmd.Parameters.Add( "", CType( 0, Object ) )

// C#
cmd.Parameters.Add( "", (object)0 );
```

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「Add メソッド」 795 ページ
- 「Add(ULParameter) メソッド」 796 ページ
- 「ULParameter クラス」 773 ページ
- 「CommandText プロパティ」 516 ページ
- 「ULCommand クラス」 509 ページ
- オブジェクト

Add(String, ULDbType) メソッド

指定されたパラメータ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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Function Add( _
  ByVal parameterName As String, _
  ByVal ulDbType As ULDbType _
) As ULParameter
```

C#

```
public ULParameter Add(
  string  parameterName,
  ULDbType ulDbType
);
```

パラメータ

- **parameterName**　パラメータの名前。名前のないパラメータの場合、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は、位置パラメータとして扱われ、ULCommand.CommandText 内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。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の数と同じでなければなりません。パラメータが不足している場合は、null が代用されます。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「Add メソッド」 795 ページ
- 「Add(ULParameter) メソッド」 796 ページ
- 「Add(String, Object) メソッド」 797 ページ
- 「ULParameter クラス」 773 ページ
- 「CommandText プロパティ」 516 ページ
- 「ULCommand クラス」 509 ページ
- 「ULDbType 列挙」 715 ページ

Add(String, ULDbType, Int32) メソッド

指定されたパラメータ名、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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Function Add( _
  ByVal parameterName As String, _
  ByVal ulDbType As ULDbType, _
  ByVal size As Integer _
) As ULParameter
```

C#

```
public ULParameter Add(
  string  parameterName,
  ULDbType ulDbType,
  int size
);
```

パラメータ

- **parameterName**　パラメータの名前。名前のないパラメータの場合、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は、位置パラメータとして扱われ、ULCommand.CommandText 内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。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の数と同じでなければなりません。パラメータが不足している場合は、null が代用されます。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「Add メソッド」 795 ページ
- 「Add(ULParameter) メソッド」 796 ページ
- 「Add(String, Object) メソッド」 797 ページ
- 「ULParameter クラス」 773 ページ
- 「CommandText プロパティ」 516 ページ
- 「ULCommand クラス」 509 ページ
- 「ULDbType 列挙」 715 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

Add(String, ULDbType, Int32, String) メソッド

指定されたパラメータ名、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。

構文

Visual Basic

```
Public Function Add( _
   ByVal parameterName As String, _
   ByVal ulDbType As ULDbType, _
   ByVal size As Integer, _
   ByVal sourceColumn As String _
) As ULParameter
```

C#

```
public ULParameter Add(
   string  parameterName,
   ULDbType ulDbType,
   int size,
   string  sourceColumn
);
```

パラメータ

- **parameterName**　パラメータの名前。名前のないパラメータの場合、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は、位置パラメータとして扱われ、ULCommand.CommandText 内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。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の数と同じでなければなりません。パラメータが不足している場合は、null が代用されます。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「Add メソッド」 795 ページ
- 「Add(ULParameter) メソッド」 796 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「Add(String, Object) メソッド」 797 ページ
- 「ULParameter クラス」 773 ページ
- 「CommandText プロパティ」 516 ページ
- 「ULCommand クラス」 509 ページ
- 「ULDbType 列挙」 715 ページ

AddRange 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。

構文

Visual Basic

```
Public Overrides Sub AddRange( _
  ByVal values As Array _
)
```

C#

```
public override void AddRange(
  Array values
);
```

パラメータ

- **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

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「AddRange メソッド」 802 ページ
- 「ULParameter クラス」 773 ページ

AddRange(ULParameter[]) メソッド

ULParameterCollection の末尾に値の配列を追加します。

構文

Visual Basic

```
Public Sub AddRange( _
  ByVal values As ULParameter() _
)
```

 Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public void AddRange(
  ULParameter[] values
);
```

パラメータ

- **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

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「AddRange メソッド」 802 ページ
- 「ULParameter クラス」 773 ページ
- DbParameterCollection.AddRange
- 「ULParameterCollection クラス」 790 ページ

Clear メソッド

すべてのパラメータをコレクションから削除します。

構文

Visual Basic

```
Public Overrides Sub Clear()
```

C#

```
public override void Clear();
```

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ

Contains メソッド

特定のプロパティが設定された DbParameter がコレクション内に存在するかどうかを示します。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Overrides Function Contains( _
  ByVal value As Object _
) As Boolean
```

C#

```
public override bool Contains(
  object value
);
```

パラメータ

- **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 クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「Contains メソッド」 803 ページ
- 「Contains(String) メソッド」 804 ページ
- 「ULParameter クラス」 773 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。

構文

Visual Basic

```
Public Overrides Function Contains( _
  ByVal value As String _
) As Boolean
```

C#

```
public override bool Contains(
  string  value
);
```

パラメータ

- **value**　検索対象のパラメータの名前。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「Contains メソッド」 803 ページ
- 「Contains(Object) メソッド」 803 ページ
- 「ULParameter クラス」 773 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。

構文

Visual Basic

```
Public Overrides Sub CopyTo( _
  ByVal array As Array, _
  ByVal index As Integer _
)
```

C#

```
public override void CopyTo(
  Array array,
  int index
);
```

パラメータ

- **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

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「ULParameter クラス」 773 ページ

GetEnumerator メソッド

コレクションの列挙子を返します。

構文

Visual Basic

```
Public Overrides Function GetEnumerator() As IEnumerator
```

C#

```
public override IEnumerator GetEnumerator();
```

Copyright © 2007, iAnywhere Solutions, Inc.

戻り値

コレクション内のパラメータを列挙する ArrayList 列挙子。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ

IndexOf メソッド

指定された DbParameter オブジェクトのインデックスを返します。

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。

構文

Visual Basic

```
Public Overrides Function IndexOf( _
  ByVal value As Object _
) As Integer
```

C#

```
public override int IndexOf(
  object value
);
```

パラメータ

- **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。パラメータが見つからない場合は -1。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「IndexOf メソッド」 806 ページ
- 「IndexOf(String) メソッド」 806 ページ
- 「ULParameter クラス」 773 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Overrides Function IndexOf( _
  ByVal parameterName As String _
) As Integer
```

C#

```
public override int IndexOf(
  string  parameterName
);
```

パラメータ

- **parameterName**　検索するパラメータの名前。

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。パラメータが見つからない場合は -1。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「IndexOf メソッド」 806 ページ
- 「IndexOf(Object) メソッド」 806 ページ
- 「ULParameter クラス」 773 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。

構文

Visual Basic

```
Public Overrides Sub Insert( _
  ByVal index As Integer, _
  ByVal value As Object _
)
```

C#

```
public override void Insert(
  int index,
  object value
);
```

パラメータ

- **index**　コレクション内にパラメータ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。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「ULParameter クラス」 773 ページ

Remove メソッド

ULParameter をコレクションから削除します。

構文

Visual Basic

```
Public Overrides Sub Remove( _
  ByVal value As Object _
)
```

C#

```
public override void Remove(
  object value
);
```

パラメータ

- **value**　削除する ULParameter オブジェクト。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「ULParameter クラス」 773 ページ

RemoveAt メソッド

指定された DbParameter オブジェクトをコレクションから削除します。

RemoveAt(Int32) 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を削除します。

構文

Visual Basic

```
Public Overrides Sub RemoveAt( _
  ByVal index As Integer _
)
```

C#

Copyright © 2007, iAnywhere Solutions, Inc.

```
public override void RemoveAt(
  int index
);
```

パラメータ

- **index**　削除するパラメータの 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のインデックス値はゼロです。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「RemoveAt メソッド」 808 ページ
- 「RemoveAt(String) メソッド」 809 ページ
- 「Count プロパティ」 792 ページ

RemoveAt(String) メソッド

指定された名前のパラメータをコレクションから削除します。

構文

Visual Basic

```
Public Overrides Sub RemoveAt( _
  ByVal parameterName As String _
)
```

C#

```
public override void RemoveAt(
  string  parameterName
);
```

パラメータ

- **parameterName**　取り出すパラメータの名前。

参照

- 「ULParameterCollection クラス」 790 ページ
- 「ULParameterCollection メンバ」 790 ページ
- 「RemoveAt メソッド」 808 ページ
- 「RemoveAt(Int32) メソッド」 808 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULPublicationSchema クラス

UL 拡張： Ultra Light パブリケーションのインデックスのスキーマを示します。このクラスは継承できません。

構文

Visual Basic

Public NotInheritable Class **ULPublicationSchema**

C#

public sealed class **ULPublicationSchema**

備考

このクラスにはコンストラクタがありません。パブリケーションのスキーマは、ULDatabaseSchema クラスの ULDatabaseSchema.GetPublicationSchema メソッドを使用して作成されます。

パブリケーション・マスクが必要な Ultra Light のメソッドには、実際にはチェック対象のパブリケーションのセットが必要です。セットは、個々のパブリケーションのパブリケーション・マスクの論理和で形成されます。次に例を示します。

```
' Visual Basic
Dim mask As Integer = pub1.Mask Or pub2.Mask

// C#
int mask = pub1.Mask | pub2.Mask;
```

特別な 2 つのマスク値も、このクラスによって提供されます。
ULPublicationSchema.SYNC_ALL_DB は、データベース全体に対応します。
ULPublicationSchema.SYNC_ALL_PUBS は、すべてのパブリケーションに対応します。

注意：パブリケーション・マスクは、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マスク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

- 「ULPublicationSchema メンバ」 811 ページ
- 「Mask プロパティ」 812 ページ
- 「GetLastDownloadTime メソッド」 579 ページ
- 「CountUploadRows(Int32, UInt32) メソッド」 572 ページ
- 「Schema プロパティ」 565 ページ
- 「SYNC_ALL_DB フィールド」 811 ページ
- 「SYNC_ALL_PUBS フィールド」 812 ページ
- 「GetPublicationSchema メソッド」 672 ページ
- 「ULDatabaseSchema クラス」 665 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULPublicationSchema メンバ

パブリック・フィールド

メンバ名	説明
SYNC_ALL_DB フィールド	データベース全体に対応するパブリケーション・マスクです。このフィールドは定数で、読み込み専用です。
SYNC_ALL_PUBS フィールド	すべてのパブリケーションに対応するパブリケーション・マスクです。このフィールドは定数で、読み込み専用です。

パブリック・プロパティ

メンバ名	説明
IsOpen プロパティ	パブリケーション・スキーマが開いているか、閉じているかを調べます。
Mask プロパティ	パブリケーションのパブリケーション・マスクを返します。
Name プロパティ	このパブリケーションの名前を返します。

参照

- 「ULPublicationSchema クラス」 810 ページ
- 「Mask プロパティ」 812 ページ
- 「GetLastDownloadTime メソッド」 579 ページ
- 「CountUploadRows(Int32, UInt32) メソッド」 572 ページ
- 「Schema プロパティ」 565 ページ
- 「SYNC_ALL_DB フィールド」 811 ページ
- 「SYNC_ALL_PUBS フィールド」 812 ページ
- 「GetPublicationSchema メソッド」 672 ページ
- 「ULDatabaseSchema クラス」 665 ページ

SYNC_ALL_DB フィールド

データベース全体に対応するパブリケーション・マスクです。このフィールドは定数で、読み込み専用です。

構文

Visual Basic

Public Shared **SYNC_ALL_DB** As Integer

C#

public const int **SYNC_ALL_DB** ;

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULPublicationSchema クラス」 810 ページ
- 「ULPublicationSchema メンバ」 811 ページ

SYNC_ALL_PUBS フィールド

すべてのパブリケーションに対応するパブリケーション・マスクです。このフィールドは定数で、読み込み専用です。

構文

Visual Basic

Public Shared **SYNC_ALL_PUBS** As Integer

C#

public const int **SYNC_ALL_PUBS** ;

参照

- 「ULPublicationSchema クラス」 810 ページ
- 「ULPublicationSchema メンバ」 811 ページ

IsOpen プロパティ

パブリケーション・スキーマが開いているか、閉じているかを調べます。

構文

Visual Basic

Public Readonly Property **IsOpen** As Boolean

C#

public bool **IsOpen** { get;}

プロパティ値

パブリケーション・スキーマが開いている場合は true、閉じている場合は false。

参照

- 「ULPublicationSchema クラス」 810 ページ
- 「ULPublicationSchema メンバ」 811 ページ

Mask プロパティ

パブリケーションのパブリケーション・マスクを返します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **Mask** As Integer

C#

public int **Mask** { get;}

プロパティ値

パブリケーションのパブリケーション・マスク。

備考

パブリケーションの ID、マスク、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マスク、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

- 「ULPublicationSchema クラス」 810 ページ
- 「ULPublicationSchema メンバ」 811 ページ

Name プロパティ

このパブリケーションの名前を返します。

構文

Visual Basic

Public Readonly Property **Name** As String

C#

public string **Name** { get;}

プロパティ値

パブリケーションの名前を指定する文字列。

参照

- 「ULPublicationSchema クラス」 810 ページ
- 「ULPublicationSchema メンバ」 811 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Class **ULResultSet**
  Inherits ULDataReader

C#

public class **ULResultSet** : ULDataReader

備考

このクラスにはコンストラクタがありません。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()

// 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。

継承： ULDataReader

実装： System.Data.IDataReader、System.Data.IDataRecord、System.IDisposable

参照

- 「ULResultSet メンバ」 815 ページ
- 「ExecuteResultSet() メソッド」 538 ページ
- 「ULCommand クラス」 509 ページ
- 「ULResultSet クラス」 814 ページ
- 「ExecuteTable() メソッド」 541 ページ
- 「ULDataAdapter クラス」 645 ページ
- 「ULDataReader クラス」 676 ページ
- IDataReader
- IDataRecord
- IDisposable

Copyright © 2007, iAnywhere Solutions, Inc.

ULResultSet メンバ

パブリック・プロパティ

メンバ名	説明
Depth プロパティ (ULDataReader から継承)	現在のローのネストの深さを返します。最も外側のテーブルの深さは 0 です。
FieldCount プロパティ (ULDataReader から継承)	このカーソル内のカラム数を返します。
HasRows プロパティ (ULDataReader から継承)	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 (ULDataReader から継承)	**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 (ULDataReader から継承)	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 (ULDataReader から継承)	**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。
Item プロパティ (ULDataReader から継承)	指定されたカラムの値を Object のインスタンスとして取得します。
RecordsAffected プロパティ (ULDataReader から継承)	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。SELECT 文または CommandType.TableDirect テーブルの場合、この値は -1 です。
RowCount プロパティ (ULDataReader から継承)	**UL 拡張：** カーソル内のロー数を返します。
Schema プロパティ (ULDataReader から継承)	**UL 拡張：** このカーソルのスキーマを保持します。
VisibleFieldCount (DbDataReader から継承)	

パブリック・メソッド

メンバ名	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 (ULDataReader から継承)	カーソルを閉じます。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
Delete メソッド	現在の行を削除します。
Dispose (DbDataReader から継承)	
GetBoolean メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	**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。ULDbType.Binary 型、ULDbType.LongBinary 型、ULDbType.UniqueIdentifier 型のカラムの場合にのみ有効です。
GetChar メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (DbDataReader から継承)	
GetDataTypeName メソッド (ULDataReader から継承)	指定されたカラムのプロバイダのデータ型の名前を返します。
GetDateTime メソッド (ULDataReader から継承)	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。
GetDecimal メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 から継承)	ULDataReader の反復処理を実行する System.Collections.IEnumerator を返します。
GetFieldType メソッド (ULDataReader から継承)	指定されたカラムに最適な System.Type を返します。
GetFloat メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	指定されたカラムの値を System.Int16 として返します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
GetInt32 メソッド (ULDataReader から継承)	指定されたカラムの値を Int32 として返します。
GetInt64 メソッド (ULDataReader から継承)	指定されたカラムの値を Int64 として返します。
GetName メソッド (ULDataReader から継承)	指定されたカラムの名前を返します。
GetOrdinal メソッド (ULDataReader から継承)	指定されたカラムのカラム ID を返します。
GetProviderSpecificFieldType (DbDataReader から継承)	
GetProviderSpecificValue (DbDataReader から継承)	
GetProviderSpecificValues (DbDataReader から継承)	
GetSchemaTable メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	指定されたカラムの値を UInt32 として返します。
GetUInt64 メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	現在のローのすべてのカラム値を返します。
IsDBNull メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	**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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
MoveBeforeFirst メソッド (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	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 (ULDataReader から継承)	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。
SetDBNull メソッド	カラムを NULL に設定します。
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。
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。
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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。

参照

- 「ULResultSet クラス」 814 ページ
- 「ExecuteResultSet() メソッド」 538 ページ
- 「ULCommand クラス」 509 ページ
- 「ULResultSet クラス」 814 ページ
- 「ExecuteTable() メソッド」 541 ページ
- 「ULDataAdapter クラス」 645 ページ
- 「ULDataReader クラス」 676 ページ
- IDataReader
- IDataRecord
- IDisposable

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。

構文

Visual Basic

```
Public Sub AppendBytes( _
  ByVal columnID As Integer, _
  ByVal val As Byte(), _
```

```
  ByVal srcOffset As Integer, _
  ByVal count As Integer _
)
```

C#

```
public void AppendBytes(
  int columnID,
  byte[] val,
  int srcOffset,
  int count
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **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。

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。
- *srcOffset*+*count* が *val*.Length. 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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- Byte
- 「ULDbType 列挙」 715 ページ
- 「InsertBegin メソッド」 948 ページ
- 「Insert メソッド」 948 ページ
- 「ULException クラス」 719 ページ
- 「ULSQLCode 列挙」 859 ページ
- 「ULException クラス」 719 ページ
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Sub AppendChars( _
  ByVal columnID As Integer, _
  ByVal val As Char(), _
  ByVal srcOffset As Integer, _
  ByVal count As Integer _
)
```

C#

```
public void AppendChars(
  int columnID,
  char[] val,
  int srcOffset,
  int count
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **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。

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。

Copyright © 2007, iAnywhere Solutions, Inc.

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*.Length. 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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- Char
- 「ULDbType 列挙」 715 ページ
- 「InsertBegin メソッド」 948 ページ
- 「Insert メソッド」 948 ページ
- 「ULException クラス」 719 ページ
- 「ULSQLCode 列挙」 859 ページ
- 「FieldCount プロパティ」 680 ページ

Delete メソッド

現在の行を削除します。

構文

Visual Basic

Public Sub **Delete()**

C#

public void **Delete();**

参照

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「StartSynchronizationDelete メソッド」 586 ページ
- 「StopSynchronizationDelete メソッド」 587 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Public Sub SetBoolean( _
  ByVal columnID As Integer, _
  ByVal val As Boolean _
)
```

C#

```
public void SetBoolean(
  int columnID,
  bool val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- Boolean
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Sub SetByte( _
  ByVal columnID As Integer, _
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
  ByVal val As Byte _
)
```

C#

```
public void SetByte(
  int columnID,
  byte val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- Byte
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Sub SetBytes( _
  ByVal columnID As Integer, _
  ByVal val As Byte() _
)
```

C#

```
public void SetBytes(
  int columnID,
  byte[] val
);
```


Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- Byte
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 ページ
- 「ULDbType 列挙」 715 ページ
- 「ULDbType 列挙」 715 ページ
- 「ULDbType 列挙」 715 ページ

SetDBNull メソッド

カラムを NULL に設定します。

構文

Visual Basic

```
Public Sub SetDBNull( _
  ByVal columnID As Integer _
)
```

C#

```
public void SetDBNull(
  int columnID
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

Copyright © 2007, iAnywhere Solutions, Inc.

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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「IsColumnNullable メソッド」 967 ページ
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Sub SetDateTime( _
  ByVal columnID As Integer, _
  ByVal val As Date _
)
```

C#

```
public void SetDateTime(
  int columnID,
  DateTime val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- DateTime
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 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。

構文

Visual Basic

```
Public Sub SetDecimal( _
  ByVal columnID As Integer, _
  ByVal val As Decimal _
)
```

C#

```
public void SetDecimal(
  int columnID,
  decimal val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- Decimal
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Public Sub SetDouble( _
  ByVal columnID As Integer, _
  ByVal val As Double _
)
```

C#

```
public void SetDouble(
  int columnID,
  double val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- Double
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Sub SetFloat( _
  ByVal columnID As Integer, _
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
  ByVal val As Single _
)
```

C#

```
public void SetFloat(
  int columnID,
  float  val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- Single
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Sub SetGuid( _
  ByVal columnID As Integer, _
  ByVal val As Guid _
)
```

C#

```
public void SetGuid(
  int columnID,
  Guid val
);
```

Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetNewUUID メソッド」 580 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- 「GetColumnSize メソッド」 642 ページ
- Guid
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「ULDbType 列挙」 715 ページ
- 「ULDbType 列挙」 715 ページ
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Sub SetInt16( _
  ByVal columnID As Integer, _
  ByVal val As Short _
)
```

C#

```
public void SetInt16(
  int columnID,
  short val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

Copyright © 2007, iAnywhere Solutions, Inc.

- **val**　カラムの新しい値。

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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- Int16
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Sub SetInt32( _
  ByVal columnID As Integer, _
  ByVal val As Integer _
)
```

C#

```
public void SetInt32(
  int columnID,
  int val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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

- 「ULResultSet クラス」 814 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- Int32
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Sub SetInt64( _
   ByVal columnID As Integer, _
   ByVal val As Long _
)
```

C#

```
public void SetInt64(
   int columnID,
   long val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- Int64
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Public Sub SetString( _
  ByVal columnID As Integer, _
  ByVal val As String _
)
```

C#

```
public void SetString(
  int columnID,
  string  val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 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。

構文

Visual Basic

```
Public Sub SetTimeSpan( _
  ByVal columnID As Integer, _
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
  ByVal val As TimeSpan _
)
```

C#

```
public void SetTimeSpan(
  int columnID,
  TimeSpan val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

備考

設定された値は、ミリ秒の精度であり、0 ー 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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- TimeSpan
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Sub SetToDefault( _
  ByVal columnID As Integer _
)
```

C#

```
public void SetToDefault(
  int columnID
);
```


Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。

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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- 「GetColumnDefaultValue メソッド」 959 ページ
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 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。

構文

Visual Basic

```
Public Sub SetUInt16( _
  ByVal columnID As Integer, _
  ByVal val As UInt16 _
)
```

C#

```
public void SetUInt16(
  int columnID,
  ushort val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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 © 2007, iAnywhere Solutions, Inc.

参照

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「GetFieldType メソッド」 695 ページ
- UInt16
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Sub SetUInt32( _
  ByVal columnID As Integer, _
  ByVal val As UInt32 _
)
```

C#

```
public void SetUInt32(
  int columnID,
  uint val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- UInt32
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「FieldCount プロパティ」 680 ページ

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。

構文

Visual Basic

```
Public Sub SetUInt64( _
  ByVal columnID As Integer, _
  ByVal val As UInt64 _
)
```

C#

```
public void SetUInt64(
  int columnID,
  ulong val
);
```

パラメータ

- **columnID**　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。カーソルの最初のカラムの ID 値はゼロです。
- **val**　カラムの新しい値。

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

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ
- 「GetOrdinal メソッド」 700 ページ
- 「Schema プロパティ」 685 ページ
- 「GetFieldType メソッド」 695 ページ
- UInt64
- 「Insert メソッド」 948 ページ
- 「Update メソッド」 837 ページ
- 「FieldCount プロパティ」 680 ページ

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Sub **Update()**

C#

public void **Update();**

参照

- 「ULResultSet クラス」 814 ページ
- 「ULResultSet メンバ」 815 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。このクラスは継承できません。

構文

Visual Basic

```
Public NotInheritable Class ULResultSetSchema
  Inherits ULCursorSchema
```

C#

```
public sealed class ULResultSetSchema : ULCursorSchema
```

備考

このクラスにはコンストラクタがありません。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・リーダが開かれている間だけです。

継承： ULCursorSchema

参照

- 「ULResultSetSchema メンバ」 839 ページ
- 「ULCommand クラス」 509 ページ
- 「ULDataReader クラス」 676 ページ
- 「ULResultSetSchema クラス」 839 ページ
- 「Schema プロパティ」 685 ページ
- 「ULCursorSchema クラス」 636 ページ

ULResultSetSchema メンバ

パブリック・プロパティ

メンバ名	説明
ColumnCount プロパティ (ULCursorSchema から継承)	このカーソル内のカラム数を返します。
IsOpen プロパティ (ULCursorSchema から継承)	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。

Copyright © 2007, iAnywhere Solutions, Inc.

パブリック・メソッド

メンバ名	説明
GetColumnID メソッド (ULCursorSchema から継承)	指定されたカラムのカラム ID を返します。
GetColumnName メソッド (ULCursorSchema から継承)	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 (ULCursorSchema から継承)	カラムが数値カラム (SQL NUMERIC 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。
GetColumnSQLName メソッド (ULCursorSchema から継承)	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。
GetColumnScale メソッド (ULCursorSchema から継承)	カラムが数値カラム (SQL NUMERIC 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 (ULCursorSchema から継承)	カラムがサイズ指定されたカラム (SQL BINARY 型または CHAR 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。
GetColumnULDbType メソッド (ULCursorSchema から継承)	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 (ULCursorSchema から継承)	ULDataReader のカラムのスキーマが記述された System.Data.DataTable を返します。

参照

- 「ULResultSetSchema クラス」 839 ページ
- 「ULCommand クラス」 509 ページ
- 「ULDataReader クラス」 676 ページ
- 「ULResultSetSchema クラス」 839 ページ
- 「Schema プロパティ」 685 ページ
- 「ULCursorSchema クラス」 636 ページ

Name プロパティ

カーソルの名前を返します。

構文

Visual Basic

```
Public Overrides Readonly Property Name As String
```

C#

```
public override string  Name  { get;}
```


Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

ULResultSetSchema を生成した SQL 文。

参照

- 「ULResultSetSchema クラス」 839 ページ
- 「ULResultSetSchema メンバ」 839 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULRowsCopiedEventArgs クラス

ULRowsCopiedEventHandler に渡される引数のセットを示します。このクラスは継承できません。

構文

Visual Basic

Public NotInheritable Class **ULRowsCopiedEventArgs**

C#

public sealed class **ULRowsCopiedEventArgs**

備考

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

参照

- 「ULRowsCopiedEventArgs メンバ」 842 ページ
- 「ULRowsCopiedEventHandler 委任」 845 ページ

ULRowsCopiedEventArgs メンバ

パブリック・コンストラクタ

メンバ名	説明
ULRowsCopiedEventArgs コンストラクタ	ULRowsCopiedEventArgs オブジェクトの新しいインスタンスを作成します。

パブリック・プロパティ

メンバ名	説明
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。

参照

- 「ULRowsCopiedEventArgs クラス」 842 ページ
- 「ULRowsCopiedEventHandler 委任」 845 ページ

ULRowsCopiedEventArgs コンストラクタ

ULRowsCopiedEventArgs オブジェクトの新しいインスタンスを作成します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Sub New( _
  ByVal rowsCopied As Long _
)
```

C#

```
public  ULRowsCopiedEventArgs(
  long rowsCopied
);
```

パラメータ

- **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

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

参照

- 「ULRowsCopiedEventArgs クラス」 842 ページ
- 「ULRowsCopiedEventArgs メンバ」 842 ページ

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。

構文

Visual Basic

```
Public Property Abort As Boolean
```

C#

```
public bool Abort  { get; set; }
```

備考

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

参照

- 「ULRowsCopiedEventArgs クラス」 842 ページ
- 「ULRowsCopiedEventArgs メンバ」 842 ページ

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **RowsCopied** As Long

C#

public long **RowsCopied** { get;}

プロパティ値

コピーされたロー数を示す long integer。

備考

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

参照

- 「ULRowsCopiedEventArgs クラス」 842 ページ
- 「ULRowsCopiedEventArgs メンバ」 842 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULRowsCopiedEventHandler 委任

ULBulkCopy.ULRowsCopied イベントを処理するメソッドを示します。

構文

Visual Basic

```
Public Delegate Sub ULRowsCopiedEventHandler( _
  ByVal sender As Object, _
  ByVal rowsCopiedEventArgs As ULRowsCopiedEventArgs _
)
```

C#

```
public delegate void ULRowsCopiedEventHandler(
  object sender,
  ULRowsCopiedEventArgs rowsCopiedEventArgs
);
```

備考

制限： ULRowsCopiedEventHandler 委任は、.NET Compact Framework 2.0 では使用できません。

参照

- 「ULRowsCopied イベント」 491 ページ
- オブジェクト

Copyright © 2007, iAnywhere Solutions, Inc.

ULRowUpdatedEventArgs クラス

ULDataAdapter.RowUpdated イベントのデータを提供します。このクラスは継承できません。

構文

Visual Basic

```
Public NotInheritable Class ULRowUpdatedEventArgs
  Inherits RowUpdatedEventArgs
```

C#

```
public sealed class ULRowUpdatedEventArgs : RowUpdatedEventArgs
```

備考

継承： System.Data.Common.RowUpdatedEventArgs

参照

- 「ULRowUpdatedEventArgs メンバ」 846 ページ
- 「RowUpdated イベント」 655 ページ
- RowUpdatedEventArgs

ULRowUpdatedEventArgs メンバ

パブリック・コンストラクタ

メンバ名	説明
ULRowUpdatedEventArgs コンストラクタ	ULRowUpdatedEventArgs クラスの新しいインスタンスを初期化します。

パブリック・プロパティ

メンバ名	説明
Command プロパティ	DbDataAdapter.Update(System.Data.DataRow [],System.Data.Common.DataTableMapping) が呼び出されたときに実行された ULCommand を返します。
Errors (RowUpdatedEventArgs から継承)	
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。SELECT 文の場合、この値は -1 です。
Row (RowUpdatedEventArgs から継承)	

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
RowCount (RowUpdatedEventArgs から継承)	
StatementType (RowUpdatedEventArgs から継承)	
Status (RowUpdatedEventArgs から継承)	
TableMapping (RowUpdatedEventArgs から継承)	

パブリック・メソッド

メンバ名	説明
CopyToRows (RowUpdatedEventArgs から継承)	バッチ更新操作の実行中に処理されるローにアクセスできます。

参照

- 「ULRowUpdatedEventArgs クラス」 846 ページ
- 「RowUpdated イベント」 655 ページ
- RowUpdatedEventArgs

ULRowUpdatedEventArgs コンストラクタ

ULRowUpdatedEventArgs クラスの新しいインスタンスを初期化します。

構文

Visual Basic

```
Public Sub New( _
   ByVal row As DataRow, _
   ByVal command As IDbCommand, _
   ByVal statementType As StatementType, _
   ByVal tableMapping As DataTableMapping _
)
```

C#

```
public  ULRowUpdatedEventArgs(
   DataRow row,
   IDbCommand command,
   StatementType statementType,
```

```
  DataTableMapping tableMapping
);
```

パラメータ

- **row** DbDataAdapter.Update(System.Data.DataRow[],System.Data.Common.DataTableMapping) によって送信された System.Data.DataRow。
- **command** DbDataAdapter.Update(System.Data.DataRow [],System.Data.Common.DataTableMapping) が呼び出されたときに実行された 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(System.Data.DataRow [],System.Data.Common.DataTableMapping) によって送信された System.Data.Common.DataTableMapping。

参照

- 「ULRowUpdatedEventArgs クラス」 846 ページ
- 「ULRowUpdatedEventArgs メンバ」 846 ページ
- DataRow
- IDbCommand
- StatementType
- DataTableMapping
- DbDataAdapter.Update

Command プロパティ

DbDataAdapter.Update(System.Data.DataRow[],System.Data.Common.DataTableMapping) が呼び出されたときに実行された ULCommand を返します。

構文

Visual Basic

```
Public Readonly Property Command As ULCommand
```

C#

```
public ULCommand Command  { get;}
```

プロパティ値

更新で実行された ULCommand オブジェクト。

備考

これは、System.Data.Common.RowUpdatedEventArgs.Command が厳密に型指定されたものです。

参照

- 「ULRowUpdatedEventArgs クラス」 846 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULRowUpdatedEventArgs メンバ」 846 ページ
- 「ULCommand クラス」 509 ページ
- DbDataAdapter.Update
- 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。SELECT 文の場合、この値は -1 です。

構文

Visual Basic

Public Readonly Property **RecordsAffected** As Integer

C#

public int **RecordsAffected** { get;}

プロパティ値

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。

参照

- 「ULRowUpdatedEventArgs クラス」 846 ページ
- 「ULRowUpdatedEventArgs メンバ」 846 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULRowUpdatedEventHandler 委任

ULDataAdapter.RowUpdated イベントを処理するメソッドを示します。

構文

Visual Basic

```
Public Delegate Sub ULRowUpdatedEventHandler( _
   ByVal sender As Object, _
   ByVal e As ULRowUpdatedEventArgs _
)
```

C#

```
public delegate void ULRowUpdatedEventHandler(
   object sender,
   ULRowUpdatedEventArgs e
);
```

参照

- 「RowUpdated イベント」 655 ページ
- 「ULRowUpdatedEventArgs クラス」 846 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

ULRowUpdatingEventArgs クラス

ULDataAdapter.RowUpdating イベントのデータを提供します。このクラスは継承できません。

構文

Visual Basic

```
Public NotInheritable Class ULRowUpdatingEventArgs
  Inherits RowUpdatingEventArgs
```

C#

```
public sealed class ULRowUpdatingEventArgs : RowUpdatingEventArgs
```

備考

継承： System.Data.Common.RowUpdatingEventArgs

参照

- 「ULRowUpdatingEventArgs メンバ」 851 ページ
- 「RowUpdating イベント」 656 ページ
- RowUpdatingEventArgs

ULRowUpdatingEventArgs メンバ

パブリック・コンストラクタ

メンバ名	説明
ULRowUpdatingEventArgs コンストラクタ	ULRowUpdatingEventArgs クラスの新しいインスタンスを初期化します。

パブリック・プロパティ

メンバ名	説明
Command プロパティ	DbDataAdapter.Update(System.Data.DataRow [],System.Data.Common.DataTableMapping) の実行時に実行する ULCommand を指定します。
Errors (RowUpdatingEventArgs から継承)	
Row (RowUpdatingEventArgs から継承)	
StatementType (RowUpdatingEventArgs から継承)	

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
Status (RowUpdatingEventArgs から継承)	
TableMapping (RowUpdatingEventArgs から継承)	

参照

- 「ULRowUpdatingEventArgs クラス」 851 ページ
- 「RowUpdating イベント」 656 ページ
- RowUpdatingEventArgs

ULRowUpdatingEventArgs コンストラクタ

ULRowUpdatingEventArgs クラスの新しいインスタンスを初期化します。

構文

Visual Basic

```
Public Sub New( _
   ByVal row As DataRow, _
   ByVal command As IDbCommand, _
   ByVal statementType As StatementType, _
   ByVal tableMapping As DataTableMapping _
)
```

C#

```
public  ULRowUpdatingEventArgs(
   DataRow row,
   IDbCommand command,
   StatementType statementType,
   DataTableMapping tableMapping
);
```

パラメータ

- **row**　更新する System.Data.DataRow。
- **command**　更新時に実行する 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(System.Data.DataRow [],System.Data.Common.DataTableMapping) によって送信された System.Data.Common.DataTableMapping。

参照

- 「ULRowUpdatingEventArgs クラス」 851 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULRowUpdatingEventArgs メンバ」 851 ページ
- DataRow
- IDbCommand
- StatementType
- DataTableMapping
- DbDataAdapter.Update

Command プロパティ

DbDataAdapter.Update(System.Data.DataRow[],System.Data.Common.DataTableMapping) の実行時に実行する ULCommand を指定します。

構文

Visual Basic

Public Property **Command** As ULCommand

C#

public ULCommand **Command** { get; set; }

プロパティ値

更新時に実行する ULCommand オブジェクト。

備考

これは、System.Data.Common.RowUpdatingEventArgs.Command が厳密に型指定されたものです。

参照

- 「ULRowUpdatingEventArgs クラス」 851 ページ
- 「ULRowUpdatingEventArgs メンバ」 851 ページ
- 「ULCommand クラス」 509 ページ
- DbDataAdapter.Update
- RowUpdatingEventArgs.Command

Copyright © 2007, iAnywhere Solutions, Inc.

ULRowUpdatingEventHandler 委任

ULDataAdapter.RowUpdating イベントを処理するメソッドを示します。

構文

Visual Basic

```
Public Delegate Sub ULRowUpdatingEventHandler( _
   ByVal sender As Object, _
   ByVal e As ULRowUpdatingEventArgs _
)
```

C#

```
public delegate void ULRowUpdatingEventHandler(
   object sender,
   ULRowUpdatingEventArgs e
);
```

参照

- 「RowUpdating イベント」 656 ページ
- 「ULRowUpdatingEventArgs クラス」 851 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULRuntimeType 列挙

UL 拡張： Ultra Light.NET ランタイムのタイプを列挙します。

構文

Visual Basic

Public Enum **ULRuntimeType**

C#

public enum **ULRuntimeType**

メンバ

メンバ名	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

- 「RuntimeType プロパティ」 658 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULServerSyncListener インタフェース

UL 拡張： サーバの同期メッセージを受信するリスナ・インタフェースです。

構文

Visual Basic

Public Interface **ULServerSyncListener**

C#

public interface **ULServerSyncListener**

参照

- 「ULServerSyncListener メンバ」 856 ページ

ULServerSyncListener メンバ

パブリック・メソッド

メンバ名	説明
ServerSyncInvoked メソッド	サーバ起動同期用の 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。

参照

- 「ULServerSyncListener インタフェース」 856 ページ

ServerSyncInvoked メソッド

サーバ起動同期用の 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。

構文

Visual Basic

```
Public Sub ServerSyncInvoked( _
  ByVal messageName As String _
)
```

C#

```
public void ServerSyncInvoked(
  string  messageName
);
```


Copyright © 2007, iAnywhere Solutions, Inc.

パラメータ

- **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・インタフェー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

次のコード・フラグメントは、サーバ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。

```
' Visual Basic
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
    ...
  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
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
// C#
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(
        "myCompnay.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();
  }
}
```

参照

- 「ULServerSyncListener インタフェース」 856 ページ
- 「ULServerSyncListener メンバ」 856 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

ULSQLCode 列挙

UL 拡張： Ultra Light.NET によってレポートされる可能性のある SQL コードを列挙します。

構文

Visual Basic

Public Enum **ULSQLCode**

C#

public enum **ULSQLCode**

メンバ

メンバ名	説明	値
SQLE_AGGREGATES_NOT_ALLOWED	「集合関数の不正な使用です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-150
SQLE_ALIAS_NOT_UNIQUE	「エイリアス '%1' がユニークではありません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-830
SQLE_ALIAS_NOT_YET_DEFINED	「エイリアス '%1' の定義は、最初の参照前に記述する必要があります。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-831
SQLE_AMBIGUOUS_INDEX_NAME	「インデックス名 '%1' があいまいです。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-678
SQLE_ARGUMENT_CANNOT_BE_NULL	「プロシージャ '%2' の引数 %1 に NULL は指定できません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-90
SQLE_BAD_ENCRYPTION_KEY	「暗号化キーが不正であるか、見つかりません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-840
SQLE_BAD_PARAM_INDEX	「入力パラメータ・インデックスが範囲外です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-689
SQLE_CANNOT_ACCESS_FILESYSTEM	「デバイス上のファイルシステムにアクセスできません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1108
SQLE_CANNOT_CHANGE_ML_REMOTE_ID	「前回のアップロードのステータスが不明の場合、Mobile Link リモート ID は変更できません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1118

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SQLE_CANNOT_EXECUTE_STMT	「文を実行することができませんでした。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	111
SQLE_CANNOT_MODIFY	「テーブル '%2' のカラム '%1' を変更できません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-191
SQLE_CANNOT_REGISTER_LISTENER	「指定されたリスナを登録できませんでした。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-992
SQLE_CLIENT_OUT_OF_MEMORY	「クライアントでメモリが不足しています。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-876
SQLE_COLUMN_AMBIGUOUS	「カラム '%1' が複数のテーブルで見つかりました。相関名が必要です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-144
SQLE_COLUMN_CANNOT_BE_NULL	「テーブル '%2' のカラム '%1' は NULL 値を持つことはできません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-195
SQLE_COLUMN_IN_INDEX	「インデックスのカラムを変更することはできません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-127
SQLE_COLUMN_NOT_FOUND	「カラム '%1' が見つかりません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-143
SQLE_COLUMN_NOT_INDEXED	「カラム '%1' は、それを含んでいるテーブルのどのインデックスにも属していません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1101
SQLE_COLUMN_NOT_STREAMABLE	「操作が失敗しました。カラム '%1' のタイプがストリーミングをサポートしていません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1100
SQLE_COMMUNICATIONS_ERROR	「通信エラーが発生しました。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-85
SQLE_CONNECTION_ALREADY_EXISTS	「この接続はすでに存在します。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-955
SQLE_CONNECTION_NOT_FOUND	「接続が見つかりません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-108
SQLE_CONNECTION_RESTORED	「Ultra Light の接続がリストアされました。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	133

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SQLE_CONSTRAINT_NOT_FOUND	「制約 '%1' が見つかりません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-929
SQLE_CONVERSION_ERROR	「値 %1 をデータ型 %2 に変換できません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-157
SQLE_COULD_NOT_FIND_FUNCTION	「ダイナミック・ライブラリ '%2' に '%1' が見つかりませんでした。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-621
SQLE_COULD_NOT_LOAD_LIBRARY	「ダイナミック・ライブラリ '%1' をロードできませんでした。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-620
SQLE_CURSOR_ALREADY_OPEN	「カーソルはすでに開いています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-172
SQLE_CURSOR_NOT_DECLARED	「カーソルが宣言されていません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-170
SQLE_CURSOR_NOT_OPEN	「カーソルが開きません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-180
SQLE_CURSOR_RESTORED	「Ultra Light のカーソル (あるいは結果セットまたはテーブル) がリストアされました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	134
SQLE_CURSOROP_NOT_ALLOWED	「不正なカーソル処理をしようとしました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-187
SQLE_DATABASE_ERROR	「データベースの内部エラー %1 -- トランザクションはロールバックされました。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-301
SQLE_DATABASE_NAME_REQUIRED	「サーバを起動するには、データベース名が必要です。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-87
SQLE_DATABASE_NOT_CREATED	「データベースの作成に失敗しました : %1」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-645
SQLE_DATATYPE_NOT_ALLOWED	「式にサポートされていないデータ型があります。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-624
SQLE_DBSPACE_FULL	「データベース領域が最大ファイル・サイズに達しています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-604

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SQLE_DESCRIBE_NONSELECT	「SELECT 文以外は記述できません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-160
SQLE_DEVICE_ERROR	「I/O エラーです。%1 -- トランザクションはロールバックされました。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-305
SQLE_DEVICE_IO_FAILED	「'%1' のファイル I/O に失敗しました。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-974
SQLE_DIV_ZERO_ERROR	「ゼロで除算しようとしました。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-628
SQLE_DOWNLOAD_CONFLICT	「既存のローと競合しているため、ダウンロードに失敗しました。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-839
SQLE_DOWNLOAD_RESTART_FAILED	「ダウンロードをリトライできません。アップロードが完了していません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1102
SQLE_DROP_DATABASE_FAILED	「データベース '%1' の削除に失敗しました。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-651
SQLE_DUPLICATE_CURSOR_NAME	「カーソル名 '%1' はすでに存在します。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-683
SQLE_DUPLICATE_FOREIGN_KEY	「テーブル '%2' の外部キー '%1' は、既存の外部キーと重複しています。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-251
SQLE_DUPLICATE_OPTION	「オプション '%1' が複数回指定されています。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	139
SQLE_DYNAMIC_MEMORY_EXHAUSTED	「動的メモリが足りません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-78
SQLE_ENCRYPTION_INITIALIZATION_FAILED	「暗号化 DLL を初期化できませんでした : '%1'」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-984
SQLE_ENCRYPTION_NOT_ENABLED	「暗号化が有効になっていません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1143
SQLE_ENCRYPTION_NOT_ENABLED_WARNING	「暗号化が有効になっていません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	140

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SQLE_ENGINE_ALREADY_RUNNING	「データベース・サーバはすでに起動しています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-96
SQLE_ERROR	「実行時 SQL エラーです -- %1」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-300
SQLE_ERROR_CALLING_FUNCTION	「外部関数の呼び出しのためのリソースを割り付けられませんでした。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-622
SQLE_ERROR_IN_ASSIGNMENT	「割り当てのエラー」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-641
SQLE_EXPRESSION_ERROR	「'%1' 付近に無効な式があります。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-156
SQLE_FEATURE_NOT_ENABLED	「呼び出そうとしたメソッドは、お使いのアプリケーションでは使用できません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1092
SQLE_FILE_BAD_DB	「指定されたデータベースを開始できません。'%1' は有効なデータベース・ファイルではありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1006
SQLE_FILE_IN_USE	「指定されたデータベース・ファイルはすでに使用されています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-816
SQLE_FILE_NOT_DB	「指定されたデータベースを開始できません。'%1' はデータベースではありません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1004
SQLE_FILE_VOLUME_NOT_FOUND	「データベース '%1' に対して指定したファイル・システム・ボリュームが見つかりません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1112
SQLE_FILE_WRONG_VERSION	「指定されたデータベースを開始できません。'%1' は異なるバージョンのソフトウェアで作成されています。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1005
SQLE_FOREIGN_KEY_NAME_NOT_FOUND	「外部キー '%1' は見つかりません。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-145
SQLE_IDENTIFIER_TOO_LONG	「識別子 '%1' が長すぎます。」 『SQL Anywhere 10 - エラー・メッセージ』を参照。	-250

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SQLE_INCORRECT_VOLUME_ID	「'%1' のボリューム ID が不正です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-975
SQLE_INDEX_NAME_NOT_UNIQUE	「インデックス名 '%1' はユニークではありません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-111
SQLE_INDEX_NOT_FOUND	「インデックス '%1' が見つかりません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-183
SQLE_INDEX_NOT_UNIQUE	「テーブル '%2' のインデックス '%1' はユニークでなければなりません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-196
SQLE_INTERRUPTED	「文の実行がユーザによって中断させられました。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-299
SQLE_INVALID_CONSTRAINT_REF	「制約 '%1' への参照または操作が無効です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-937
SQLE_INVALID_DESCRIPTOR_INDEX	「記述子のインデックスが正しくありません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-640
SQLE_INVALID_DESCRIPTOR_NAME	「SQL 記述子名が正しくありません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-642
SQLE_INVALID_DISTINCT_AGGREGATE	「グループ化されたクエリに、複数の異なる集合関数が含まれています。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-863
SQLE_INVALID_FOREIGN_KEY	「テーブル '%2' の外部キー '%1' に対応するプライマリ・キーの値がありません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-194
SQLE_INVALID_FOREIGN_KEY_DEF	「外部キーのカラム '%1' にプライマリー・キーと異なる定義があります。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-113
SQLE_INVALID_GROUP_SELECT	「'%1' に対する関数またはカラムの参照も GROUP BY 句に記述する必要があります。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-149
SQLE_INVALID_INDEX_TYPE	「'%1' のインデックスの型の指定は不正です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-650

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SQLE_INVALID_LOGON	「ユーザ ID またはパスワードが無効です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-103
SQLE_INVALID_OPTION	「オプション '%1' が無効です。 -- PUBLIC 設定がありません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-200
SQLE_INVALID_OPTION_SETTING	「不正なオプション '%1' を設定しています。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-201
SQLE_INVALID_OPTION_VALUE	「'%1' は '%2' に対して無効な値です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1053
SQLE_INVALID_ORDER	「ORDER BY 句の指定が不正です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-152
SQLE_INVALID_PARAMETER	「不正なパラメータです。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-735
SQLE_INVALID_PARSE_PARAMETER	「解析エラーです : %1」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-95
SQLE_INVALID_PUBLICATION_MASK	「指定されたパブリケーション・マスクが不正です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1105
SQLE_INVALID_SQL_IDENTIFIER	「SQL の識別子が無効です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-760
SQLE_INVALID_STATEMENT	「文が無効です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-130
SQLE_INVALID_UNION	「UNION、INTERSECT、または EXCEPT の select リストの長さが一致していません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-153
SQLE_KEYLESS_ENCRYPTION	「このデータベースはキー不使用の暗号化を使用するため、要求された操作を実行できません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1109
SQLE_LOCKED	「'%2' のローは、ユーザ '%1' によってロックされています。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-210
SQLE_MAX_ROW_SIZE_EXCEEDED	「テーブル '%1' の最大ロー・サイズを超過します。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1132

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SQLE_MEMORY_ERROR	「メモリエラー -- トランザクションはロールバックされました。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-309
SQLE_METHOD_CANNOT_BE_CALLED	「メソッド '%1' は現時点では呼び出せません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-669
SQLE_MIRROR_FILE_MISMATCH	「ミラー '%1' がデータベース '%2' と一致しません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1138
SQLE_NAME_NOT_UNIQUE	「アイテム '%1' はすでに存在しています。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-110
SQLE_NO_COLUMN_NAME	「抽出されたテーブル '%1' にはカラム %2 に対する名前がありません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-163
SQLE_NO_CURRENT_ROW	「カーソルの現在のローがありません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-197
SQLE_NO_INDICATOR	「NULL に対して、インジケータ変数が用意されていません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-181
SQLE_NO_MATCHING_SELECT_ITEM	「派生テーブル '%1' の select リストに '%2' と一致する式がありません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-812
SQLE_NO_PRIMARY_KEY	「テーブル '%1' にはプライマリ・キーが定義されていません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-118
SQLE_NOERROR	SQLE_NOERROR(0) – このコードは、エラーまたは警告がなかったことを示します。	0
SQLE_NON_UPDATEABLE_COLUMN	「式を更新できません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-190
SQLE_NON_UPDATEABLE_CURSOR	「FOR UPDATE が READ ONLY カーソルに誤って指定されました。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-813
SQLE_NOT_IMPLEMENTED	「'%1' の機能は実装されていません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-134
SQLE_NOT_SUPPORTED_IN_ULTRALITE	「Ultra Light では使用できない機能です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-749

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SQLE_NOTFOUND	「ローが見つかりません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	100
SQLE_ONLY_ONE_TABLE	「カーソルの INSERT/DELETE は、1 度に 1 つのテーブルにしかできません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-199
SQLE_OVERFLOW_ERROR	「値 %1 は、対象先にとって大きすぎます。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-158
SQLE_PAGE_SIZE_INVALID	「無効なデータベース・ページ・サイズです。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-644
SQLE_PARTIAL_DOWNLOAD_NOT_FOUND	「部分ダウンロードが見つかりませんでした。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1103
SQLE_PERMISSION_DENIED	「パーミッションがありません : %1」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-121
SQLE_PRIMARY_KEY_NOT_UNIQUE	「テーブル '%1' のプライマリ・キーがユニークではありません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-193
SQLE_PRIMARY_KEY_TWICE	「テーブルに 2 つのプライマリ・キーを定義することはできません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-126
SQLE_PRIMARY_KEY_VALUE_REF	「テーブル '%1' 内のローのプライマリ・キーがテーブル '%3' 内の外部キー '%2' によって参照されています。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-198
SQLE_PUBLICATION_NOT_FOUND	「パブリケーション '%1' が見つかりません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-280
SQLE_PUBLICATION_PREDICATE_IGNORED	「パブリケーションの述部は評価されませんでした。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	138
SQLE_RESOURCE_GOVERNOR_EXCEEDED	「%1 のリソース・ガバナーが制限を超えています。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-685
SQLE_ROW_DELETED_TO_MAINTAIN_REFERENTIAL_INTEGRITY	「参照整合性を保つためにテーブル %1 からローが削除されました。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	137

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SQLE_ROW_EXCEEDS_PAGE_SIZE	「ローがデータベースのページ・サイズを超えているため、ローを格納できません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1117
SQLE_SCHEMA_UPGRADE_NOT_ALLOWED	「スキーマのアップグレードは現在認められません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-953
SQLE_SERVER_SYNCHRONIZATION_ERROR	「サーバ %1 でエラーが発生したため、同期に失敗しました。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-857
SQLE_START_STOP_DATABASE_DENIED	「データベースの起動／停止の要求は拒否されました。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-75
SQLE_STATEMENT_ERROR	「SQL 文にエラーがあります。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-132
SQLE_STRING_RIGHT_TRUNCATION	「文字列データの右側がトランケートされます。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-638
SQLE_SUBQUERY_SELECT_LIST	「select リストの中にカラムが 2 つ以上指定されています。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-151
SQLE_SYNC_INFO_INVALID	「同期の情報が不完全か無効です。'%1' を確認してください。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-956
SQLE_SYNC_INFO_REQUIRED	「同期の情報が指定されていません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1111
SQLE_SYNC_NOT_REENTRANT	「同期処理に戻ることができませんでした。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1110
SQLE_SYNC_STATUS_UNKNOWN	「最後の同期アップロードのステータスは不明です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-952
SQLE_SYNTAX_ERROR	「'%1' %2 の近くに構文エラーがあります。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-131
SQLE_TABLE_ALREADY_INCLUDED	「テーブル '%1' はすでにインクルードされています。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-822
SQLE_TABLE_IN_USE	「テーブルは使用されています。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-214

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SQLE_TABLE_NOT_FOUND	「テーブル '%1' は見つかりませんでした。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-141
SQLE_TOO_MANY_BLOB_REFS	「BLOB への参照が多すぎます。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1107
SQLE_TOO_MANY_CONNECTIONS	「データベース・サーバに接続できる限界数を超えています。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-102
SQLE_TOO_MANY_PUBLICATIONS	「パブリケーション・マスクに指定されているパブリケーションが多すぎます。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1106
SQLE_TOO_MANY_TEMP_TABLES	「接続しているテンポラリ・テーブルが多すぎます。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-817
SQLE_TOO_MANY_USERS	「データベースのユーザが多すぎます。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1104
SQLE_ULTRALITE_DATABASE_NOT_FOUND	「データベース '%1' が見つかりませんでした。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-954
SQLE_ULTRALITE_OBJ_CLOSED	「閉じられたオブジェクトに対する操作は無効です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-908
SQLE_ULTRALITE_WRITE_ACCESS_DENIED	「書き込みアクセスが拒否されました。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1086
SQLE_UNABLE_TO_CONNECT	「データベースが起動できません -- %1」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-105
SQLE_UNABLE_TO_START_DATABASE	「指定されたデータベースを起動できません : %1」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-82
SQLE_UNABLE_TO_START_DATABASE_VER_NEWER	「指定されたデータベースを起動できません : データベース %1 を起動するにはサーバをアップグレードする必要があります。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-934
SQLE_UNCOMMITTED_TRANSACTIONS	「コミットされていないトランザクションの同期またはアップグレードはできません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-755
SQLE_UNKNOWN_FUNC	「関数 '%1' はありません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-148

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SQLE_UNKNOWN_OPTION	「'%1' は認識できないオプションです。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	120
SQLE_UNKNOWN_USERID	「'%1' というユーザ ID はありません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-140
SQLE_UNRECOGNIZED_OPTION	「オプション '%1' が認識されません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1002
SQLE_UPLOAD_FAILED_AT_SERVER	「同期サーバがアップロードのコミットに失敗しました。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-794
SQLE_VALUE_IS_NULL	「要求されたデータ型として NULL の結果を返すことができません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-1050
SQLE_VARIABLE_INVALID	「無効なホスト変数です。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-155
SQLE_WRONG_NUM_OF_INSERT_COLS	「INSERT コマンドへの値が正しくありません。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-207
SQLE_WRONG_PARAMETER_COUNT	「関数 '%1' のパラメータ数が誤りです。」『SQL Anywhere 10 - エラー・メッセージ』を参照。	-154

Copyright © 2007, iAnywhere Solutions, Inc.

ULStreamErrorCode 列挙

UL 拡張： 同期中にストリームによって報告される可能性のあるエラー・コードを列挙します。

構文

Visual Basic

Public Enum **ULStreamErrorCode**

C#

public enum **ULStreamErrorCode**

メンバ

メンバ名	説明	値
ACTSYNC_NO_PORT	ActiveSync 同期を開始できるのは、ActiveSync だけです。開始するには、クレードル内にデバイスを置くか、ActiveSync マネージャの [同期] を選択します。アプリケーションから同期を開始するには、TCP/IP ソケットの同期ストリームを使用してください。	75
ACTSYNC_NOT_INSTALLED	ActiveSync プロバイダがインストールされていません。dbasinst を実行して ActiveSync プロバイダをインストールしてください (詳細についてはマニュアルを参照してください)。	76
CONNECT_TIMEOUT	接続試行がタイムアウトになりました。サーバが指定のホストとポートで実行されていないか、タイムアウト値を大きくして接続の試行に使用できる時間を増やす必要があります。	241
COULD_NOT_OPEN_FILE_FOR_WRITE	指定されたファイルを書き込み用に開くことができませんでした。このファイルが正しいファイルであること、またファイルが他のアプリケーションで使用されていないことを確認してください。	230
CREATE_RANDOM_OBJECT	セキュア・ネットワーク・レイヤが、乱数生成オブジェクトを作成できませんでした。システム・リソースを解放してから、もう一度接続と操作を行ってください。	17
DEQUEUING_CONNECTION	キューイングされている接続 (同期) 要求を取得しようとしているときに、Mobile Link サーバでエラーが発生しました。システム・リソースを解放してください。問題が解決しなければ、Mobile Link サーバを再起動してください。	19

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
DUN_DIAL_FAILED	指定されたダイヤルアップ・ネットワークに自動ダイヤルアップで接続を確立できませんでした。	204
DUN_NOT_SUPPORTED	システム・サポートが十分でないため、ダイヤルアップに失敗しました。PocketPC では cellcore.dll を使用し、Windows では IE 4.0 以降の wininet.dll を使用する必要があります。他のプラットフォームではダイヤルアップはサポートされていません。	203
END_READ	ネットワークからの一連の読み込みを完了できません。参照：READ	11
END_WRITE	ネットワークへの一連の書き込みを完了できません。参照：WRITE	10
GENERATE_RANDOM	セキュア・ネットワーク・レイヤでは乱数が必要ですが、それを生成できませんでした。システム・リソースを解放してから、もう一度接続と操作を行ってください。	14
HTTP_AUTHENTICATION_FAILED	入力したユーザ ID とパスワードは拒否されました。正しく入力したかどうか確認してください。入力した場合は、正しいアクセス権があることをシステム管理者に確認してください。	211
HTTP_AUTHENTICATION_REQUIRED	HTTP サーバまたはゲートウェイが HTTP 認証を要求しました。HTTP 同期パラメータ http_userid と http_password を使用してユーザ ID とパスワードを指定してください。	209
HTTP_BAD_STATUS_CODE	ステータス行を調べて失敗の原因を判別してください。	86
HTTP_BUFFER_SIZE_OUT_OF_RANGE	HTTP バッファ・サイズを修正してください。有効なバッファ・サイズは正の値で、ホスト・プラットフォームにとって大きすぎないように指定してください。	79
HTTP_CHUNK_LEN_BAD_CHARACTER	固定長の HTTP 本文を使用してください。	85
HTTP_CHUNK_LEN_ENCODED_MISSING	固定長の HTTP 本文を使用してください。	84
HTTP_CLIENT_ID_NOT_SET	クライアント ID が HTTP クライアント・コードに渡されませんでした。修正プログラムについては、弊社製品の保守契約を結んでいるサポート・センタにお問い合わせください。	78

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
HTTP_CONTENT_TYPE_NOT_SPECIFIED	未知の内容タイプが指定されました。マニュアルを参照し、サポートされる内容タイプに変更してください。	77
HTTP_CRLF_ENCODED_MISSING	使用しているプロキシが Mobile Link と互換性がない可能性があります。設定を確認してください。	81
HTTP_CRLF_MISSING	使用しているプロキシが Mobile Link と互換性がない可能性があります。設定を確認してください。	82
HTTP_EXPECTED_POST	使用しているプロキシが Mobile Link と互換性がない可能性があります。設定を確認してください。	89
HTTP_EXTRA_DATA_END_READ	HTTP 本文に余分なデータが追加されています。このデータは、プロキシ・エージェントによって追加された可能性があります。プロキシを削除してください。	80
HTTP_HEADER_PARSE_ERROR	HTTP ヘッダを解析しようとしてエラーが発生しました。ヘッダの形式が間違っている可能性があります。	216
HTTP_INTERNAL_HEADER_STATE	HTTP ヘッダを復号化できませんでした。これは通常は起こるはずのない内部エラーです。保守契約を結んでいるサポート・センタに問い合わせてください。	236
HTTP_INTERNAL_REQUEST_TYPE	HTTP 要求タイプを判別できませんでした。これは通常は起こるはずのない内部エラーです。保守契約を結んでいるサポート・センタに問い合わせてください。	237
HTTP_INVALID_CHARACTER	HTTP ヘッダで予期しない文字が読み込まれました。ヘッダの形式が間違っているか、通信先から HTTP が送信されていない可能性があります。	219
HTTP_INVALID_SESSION_KEY	未知のセッション・キー・タイプが指定されました。マニュアルを参照し、サポートされるセッション・キー・タイプに変更してください。	244
HTTP_MALFORMED_SESSION_COOKIE	同期セッションの管理用の HTTP cookie が破損しています。cookie が破損している箇所を特定してください。クライアント・エラーか、HTTP の中間が正しく動作していないことが主な原因と考えられます。	235

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
HTTP_NO_CONTD_CONNECTION	リモート・サイトから送信される次の HTTP 要求の待機中に、サーバがタイムアウトになりました。この要求がサーバに届かなかった原因を判別するか、永続的な接続を試みてください。	83
HTTP_NO_PASSWORD	HTTP 認証にユーザ ID が入力されましたが、パスワードが入力されていません。認証には両方が必要です。	214
HTTP_NO_USERID	HTTP 認証にパスワードが入力されましたが、ユーザ ID が入力されていません。認証には両方が必要です。	213
HTTP_PROXY_AUTHENTICATION_FAILED	入力したユーザ ID とパスワードがプロキシ・サーバで拒否されました。正しく入力したかどうか確認してください。入力した場合は、正しいアクセス権があることをシステム管理者に確認してください。	212
HTTP_PROXY_AUTHENTICATION_REQUIRED	HTTP プロキシが HTTP 認証を要求しました。HTTP 同期パラメータ http_proxy_userid と http_proxy_password を使用してユーザ ID とパスワードを指定してください。	210
HTTP_SERVER_AUTH_FAILED	サーバから送信された認証情報ヘッダに不正な値が含まれているため、認証に失敗しました。正規の HTTP サーバに接続していることを確認してください。	217
HTTP_UNABLE_TO_PARSE_COOKIE	破損している set cookie ヘッダを判別してください。	88
HTTP_UNKNOWN_TRANSFER_ENCODING	未知の転送エンコードがどのように生成されているかを判別してください。	87
HTTP_UNSUPPORTED_AUTH_ALGORITHM	サーバが要求した HTTP ダイジェスト認証アルゴリズムは、サポートされていません。サポートされているのは「MD5」と「MD5-sess」だけです。	215
HTTP_VERSION	要求された HTTP バージョンはサポートされていません。マニュアルを調べて、サポート対象の HTTP バージョンを指定してください。パブリケーション時のサポート対象 HTTP バージョンは 1.0 と 1.1 です。	54

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
INCONSISTENT_FIPS	Mobile Link サーバのコマンド・ラインで -fips スイッチを使用するには、すべてのセキュア・ストリームが FIPS に準拠している必要があります。セキュア・ストリームに fips オプションが設定されていない場合は、自動的に FIPS 準拠 (fips=y など) になります。セキュア・ストリームから fips オプションを削除するか、fips=y で有効にしてください。	242
INIT_RANDOM	セキュア・ネットワーク・レイヤが、乱数ジェネレータを初期化できませんでした。システム・リソースを解放してから、もう一度接続と操作を行ってください。	15
INTERNAL	ネットワーク・レイヤで内部エラーが発生しました。保守契約を結んでいるサポート・センタに問い合わせてください。	220
INTERNAL_API	ネットワーク・レイヤで内部エラーが発生しました。保守契約を結んでいるサポート・センタに問い合わせてください。	238
INTERNAL_PROTOCOL_NOT_LOADED	同期プロトコルをロードできませんでした。Ultra Light を使用している場合は、適切な ULEnable メソッドを呼び出していることを確認してください。	227
INTERRUPTED	現在の操作は呼び出し元によって中断されました。	218
INVALID_COMPRESSION_TYPE	指定された圧縮型が認識されませんでした。	232
INVALID_LOCAL_PATH	ダウンロードされたファイルのローカル・パスが無効です。詳細については、マニュアルを参照してください。	243
INVALID_NETWORK_LIBRARY	指定されたネットワーク・インタフェース DLL または共有オブジェクトがロードできませんでした。不正か破損している可能性があります。	245
INVALID_SYNC_PROTOCOL	指定されたプロトコルは有効な同期プロトコルではありません。	226
LIBRARY_ENTRY_POINT_NOT_FOUND	指定されたライブラリのエントリ・ポイントが見つかりませんでした。	225
LOAD_LIBRARY_FAILURE	パスでダイナミック・ライブラリを見つけることができませんでした。同期に TLS 暗号化を使用しようとしている場合は、適切なライセンスを取得していることを確認してください。	224

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
LOAD_NETWORK_LIBRARY	ネットワーク・インタフェース・ライブラリを検出してロードできませんでした (またはそのいずれか)。次の項目を確認してください。 1) ソケット・レイヤが適切にインストールされている。適切なネットワーク・インタフェース・ライブラリ (または DLL か共有オブジェクト) が存在しており、アクセス可能になっていなければなりません。 2) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。	73
MEMORY_ALLOCATION	ネットワーク・レイヤで十分なバイト数の記憶領域を割り付けることができませんでした。システム・メモリを解放してもう一度操作を行ってください。システム・メモリを解放する方法は、オペレーティング・システムとその設定方法によって異なります。最も簡単な方法は、アクティブな処理の数を減らすことです。詳細については、使用しているオペレーティング・システムのマニュアルを参照してください。	6
MISSING_PARAMETER	必要なパラメータが指定されていません。	229
NETWORK_LIBRARY_VERSION_MISMATCH	バージョンが間違っているため、ネットワーク・インタフェース DLL または共有オブジェクトがロードできませんでした。	248
NO_ECC_FIPS	この圧縮処理を行うことができませんでした。保守契約を結んでいるサポート・センタに問い合わせてください。	239
NONE	このコードは、ネットワーク・エラーが発生しなかったか、未知のネットワーク・エラーが発生したことを示します。	0
NOT_IMPLEMENTED	実装されていない内部機能が要求されました。保守契約を結んでいるサポート・センタに問い合わせてください。	12
PARAMETER	ネットワーク・パラメータの形式は、"name=value;[name2=value2[;...]]" です。このコードは、不正なパラメータ値を示しています。該当するパラメータ名について説明したマニュアルを参照して、パラメータ値を修正してください。	1

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
PARAMETER_NOT_BOOLEAN	ネットワーク・パラメータの形式は、"name=value;[name2=value2[;...]]" です。このパラメータ値はブール値ではありません。パラメータ指定の違反箇所を探して、パラメータ値を 0 (off または False に対応) または 1 (on または True に対応) に変更してください。	4
PARAMETER_NOT_HEX	ネットワーク・パラメータの形式は、"name=value;[name2=value2[;...]]" です。このパラメータ値は、16 進数値 (基数 16) ではありません。パラメータ指定の違反箇所を探して、パラメータ値を 16 進数値に変更してください。	5
PARAMETER_NOT_UINT32	ネットワーク・パラメータの形式は、"name=value;[name2=value2[;...]]" です。このパラメータ値は、符号なし整数ではありません。パラメータ指定の違反箇所を探して、パラメータ値を符号なしの整数に変更してください。	2
PARAMETER_NOT_UINT32_RANGE	ネットワーク・パラメータの形式は、"name=value;[name2=value2[;...]]" です。このパラメータ値は、符号なし整数値または範囲のいずれでもありません。パラメータ指定の違反箇所を探して、パラメータ値を符号なしの整数値か範囲に変更してください。符号なし範囲の形式は NNN-NNN です。	3
PARSE	ネットワーク・パラメータの形式は、"name=value;[name2=value2[;...]]" です。オプションで、カッコ内にパラメータの全リストを入力できます。入力された文字列はこの表記規則に従っていません。文字列を検査して、フォーマット上の問題を修正してからもう一度操作を行ってください。	7
PROTOCOL_ERROR	予期しない値またはトークンが読み込まれました。	231

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
READ	ネットワーク・レイヤによって指定されたバイト数を読み込めませんでした。読み込みは、より規模の大きいネットワーク操作の一部として行われます。たとえば、ネットワーク・レイヤによっては、サブ・レイヤを設定して、ここで上位レイヤの基本操作の一部として読み込みと書き込みを行います。通常、読み込みエラーは次のような事柄が原因で発生します。 1) ネットワークに、読み込みを失敗させる問題が生じた。もう一度接続と操作を行ってください。 2) 接続がタイムアウトした。もう一度接続と操作を行ってください。 3) 反対側の接続が完全に終了した。クライアントかサーバ (またはその両方) のエラー・ログを参照して、接続が削除された理由を調べてください。出力されたエラー・ログを調べて原因を解明し、もう一度操作を行ってください。 4) 接続の反対側で処理がアボートされた。クライアントかサーバ (またはその両方) のエラー出力ログを参照して、処理がアボートされた理由を調べてください。通常の方法以外のやり方で処理が停止された場合は、出力ログにエラーは記録されません。もう一度接続と操作を行ってください。 5) システム・リソースが不足しており、読み込みを実行できない。システム・リソースを解放してから、もう一度接続と操作を行ってください。再度行ってもうまくいかない場合は、ネットワーク管理者に問い合わせてください。	8
READ_TIMEOUT	ネットワーク・レイヤによって指定されたバイト数を指定時間内に読み込めませんでした。ネットワークが正常に機能しており、送信側アプリケーションがまだ動作中であることを確認してください。	201
SACI_ERROR_CLIENT	SACI 実装によってエラーがレポートされました。	246
SACI_ERROR_SERVER	SACI ネットワーク・ライブラリがエラーをレポートしています。SACI ネットワーク・ライブラリのプロバイダを参照して、問題を解決してください。	247

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SECURE_ADD_CERTIFICATE	セキュア・ネットワーク・レイヤが、証明書チェーンに証明書を追加できませんでした。システム・リソースを解放して、もう一度操作を行ってください。	39
SECURE_ADD_TRUSTED_CERTIFICATE	セキュア・ネットワーク・レイヤが、証明書チェーンに信用された証明書を追加できませんでした。システム・リソースの不足が主な原因と考えられます。システム・リソースを解放して、もう一度操作を行ってください。	48
SECURE_CERTIFICATE_COMMON_NAME	指定された共通名は証明書チェーンに存在しません。次の項目を確認してください。 1) 共通名が正しく入力されている。 2) 適切な証明書ファイルが指定されている。 3) 証明書チェーンに共通名が存在している。readcert ユーティリティを使用して上記を確認できます。	52
SECURE_CERTIFICATE_COMPANY_NAME	指定された組織名は証明書チェーンに存在しません。次の項目を確認してください。 1) 組織名が正しく入力されている。 2) 適切な証明書ファイルが指定されている。 3) 証明書チェーンに組織名が存在している。readcert ユーティリティを使用して上記を確認できます。	21
SECURE_CERTIFICATE_COMPANY_UNIT	指定された組織単位は証明書チェーンに存在しません。次の項目を確認してください。 1) 会社名が正しく入力されている。 2) 適切な証明書ファイルが指定されている。 3) 証明書チェーンに会社名が存在している。readcert ユーティリティを使用して上記を確認できます。	51
SECURE_CERTIFICATE_COUNT	指定されたファイルに証明書が入っていません。次の項目を確認してください。 1) 証明書ファイル名が適切に指定されている。 2) 証明書ファイルに証明書が 1 つ以上入っている。 3) 証明書ファイルに適切な証明書が入っている。	42

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SECURE_CERTIFICATE_EXPIRED	証明書チェーンにある証明書の有効期限が切れています。有効期限が切れていない新しい証明書を入手して、再度操作を行ってください。	50
SECURE_CERTIFICATE_EXPIRY_DATE	証明書の有効期日を読み込むことができませんでした。次の項目を確認してください。 1) パスワードが正しく入力されている。 2) 証明書ファイルに証明書が 1 つ以上入っている。 3) 証明書ファイルに適切な証明書が入っている。 4) 証明書ファイルは破損していない。	37
SECURE_CERTIFICATE_FILE_NOT_FOUND	証明書ファイルを開けられませんでした。次の項目を確認してください。 1) 証明書ファイル名が適切に指定されている。 2) 証明書ファイルが存在する。 3) 証明書ファイルに証明書が 1 つ以上入っている。 4) 証明書ファイルに適切な証明書が入っている。 5) 証明書ファイルを開こうとしているプログラムにファイルを読み込む権限がある。この項目は、ユーザ・パーミッションかファイル・パーミッション (またはその両方) を持つオペレーティング・システムだけに適用されます。	33
SECURE_CERTIFICATE_NOT_TRUSTED	サーバの証明書は信用される権限保持者によって署名されていません。次の項目を確認してください。 1) 証明書ファイル名が適切に指定されている。 2) 証明書ファイルに証明書が 1 つ以上入っている。 3) 証明書ファイルに適切な証明書が入っている。 4) 信用されるルート証明書のクライアント側のリストに、サーバ側のルート証明書が含まれている。	24
SECURE_CERTIFICATE_ROOT	チェーンのルート証明書が不正です。このエラーは、パブリケーション時に定義されましたが使用されませんでした。	20

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SECURE_CREATE_CERTIFICATE	セキュア・ネットワーク・レイヤが、証明書用の記憶領域を割り付けることができませんでした。システム・リソースを解放して、もう一度操作を行ってください。	43
SECURE_CREATE_PRIVATE_KEY_OBJECT	プライベート・キーを読み込む前に、セキュア・ネットワーク・レイヤがプライベート・キーのオブジェクトを作成できませんでした。システム・リソースの不足が主な原因と考えられます。システム・リソースを解放して、もう一度操作を行ってください。	49
SECURE_DUPLICATE_CONTEXT	セキュア・ネットワーク・レイヤが、セキュリティ・コンテキストを複製できませんでした。システム・リソースを解放して、もう一度操作を行ってください。	25
SECURE_EXPORT_CERTIFICATE	セキュア・ネットワーク・レイヤが、証明書をコピーできませんでした。システム・リソースを解放して、もう一度操作を行ってください。	38
SECURE_HANDSHAKE	安全なハンドシェイクに失敗しました。次の項目を確認してください。 1) クライアントでは、正しいホスト・マシンとポート番号が指定されている。 2) サーバでは、正しいポート番号が指定されている。 3) クライアントとサーバの両方で、正しい証明書ファイルが指定されている。	53
SECURE_IMPORT_CERT_FROM_SYSTEM_STORE	システム証明書ストアから証明書をインポートできませんでした。	222
SECURE_IMPORT_CERTIFICATE	セキュア・ネットワーク・レイヤが、証明書をインポートできませんでした。次の項目を確認してください。 1) 証明書ファイル名が適切に指定されている。 2) 証明書ファイルが存在する。 3) 証明書ファイルに証明書が 1 つ以上入っている。 4) 証明書ファイルに適切な証明書が入っている。	44
SECURE_NO_CERTS_IN_SYS_STORE	システム証明書ストアで証明書が見つかりませんでした。	223

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SECURE_NO_SERVER_CERTIFICATE	サーバの証明書が指定されていません。通信のセキュリティを保護するには、サーバの証明書が必要です。指定するファイルには、サーバの証明書とプライベート・キーが含まれる必要があります。	205
SECURE_NO_SERVER_CERTIFICATE_PASSWORD	サーバ証明書のパスワードが入力されていません。サーバの暗号化されたプライベート・キーを復号化するには、パスワードが必要です。	206
SECURE_NO_TRUSTED_ROOTS	信用できるルート証明書が指定されていません。通信のセキュリティを保護するには、信用できるルート証明書が少なくとも 1 つ必要です。	207
SECURE_OPEN_SYSTEM_CERT_STORE	システム証明書ストアを開けませんでした。	221
SECURE_READ_CERTIFICATE	証明書ファイルが読み込めませんでした。次の項目を確認してください。 1) パスワードが正しく入力されている。 2) 証明書ファイルに証明書が 1 つ以上入っている。 3) 証明書ファイルに適切な証明書が入っている。 4) 証明書ファイルは破損していない。	34
SECURE_READ_PRIVATE_KEY	証明書ファイルからプライベート・キーを読み込めませんでした。次の項目を確認してください。 1) パスワードが正しく入力されている。 2) 証明書ファイルに証明書が 1 つ以上入っている。 3) 証明書ファイルに適切な証明書が入っている。 4) 証明書ファイルは破損していない。	35
SECURE_REDUNDANT_SERVER_CERTIFICATE_PASSWORD	サーバのプライベート・キーがパスワードで暗号化されていないのに、パスワードが指定されました。	208
SECURE_SET_IO	セキュア・ネットワーク・レイヤをネットワーク・レイヤに付加できませんでした。システム・リソースを解放して、もう一度操作を行ってください。	26

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SECURE_SET_PRIVATE_KEY	プライベート・キーを使用できませんでした。次の項目を確認してください。 1) パスワードが正しく入力されている。 2) 証明書ファイルに証明書が 1 つ以上入っている。 3) 証明書ファイルに適切な証明書が入っている。 4) 証明書ファイルは破損していない。	36
SECURE_TRUSTED_CERTIFICATE_FILE_NOT_FOUND	証明書ファイルが見つかりませんでした。次の項目を確認してください。 1) 証明書ファイル名が適切に指定されている。 2) 証明書ファイルが存在する。 3) 証明書ファイルに証明書が 1 つ以上入っている。 4) 証明書ファイルに適切な証明書が入っている。 5) 証明書ファイルを開こうとしているプログラムにファイル参照の権限がある。この項目は、ユーザ・パーミッションかファイル・パーミッション (またはその両方) を持つオペレーティング・システムだけに適用されます。	40
SECURE_TRUSTED_CERTIFICATE_READ	セキュア・ネットワーク・レイヤが、信用された証明書ファイルを読み込むことができませんでした。次の項目を確認してください。 1) 証明書ファイル名が適切に指定されている。 2) 証明書ファイルが存在する。 3) 証明書ファイルに証明書が 1 つ以上入っている。 4) 証明書ファイルに適切な証明書が入っている。 5) 証明書ファイルを開こうとしているプログラムにファイル参照の権限がある。この項目は、ユーザ・パーミッションかファイル・パーミッション (またはその両方) を持つオペレーティング・システムだけに適用されます。	41
SEED_RANDOM	セキュア・ネットワーク・レイヤが、乱数ジェネレータにシードを設定できませんでした。システム・リソースを解放してから、もう一度接続と操作を行ってください。	16

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SERVER_ERROR	サーバによってエラーがレポートされました。詳細については、Mobile Link の管理者に問い合わせてください。	228
SHUTTING_DOWN	シャットダウン中、Mobile Link サーバがネットワーク・レイヤでエラーを検出しました。シャットダウン時に保留になっているネットワーク操作が影響を受けた可能性があります。	18
SOCKET_BIND	ネットワーク・レイヤが、指定ポートにソケットをバインドできませんでした。次の項目を確認してください。 1) (サーバのみ) ポートがまだ使用されていないことを確認する。ポートが使用中の場合は、そのポートで受信しているアプリケーションを停止するか、別のポートを指定してください。 2) (サーバのみ) ポートの使用に関するファイアウォールの制限事項がないことを確認する。 3) (クライアントのみ) client_port オプションが使用された場合は、指定したポートがまだ使用されていないことを確認する。クライアント・ポートを 1 つだけ指定した場合は、NNN-NNN などの範囲を使用することを検討してください。範囲が指定されていた場合は、その範囲を広げるか、別の範囲にすることを検討してください。 4) (クライアントのみ) client_port オプションが使用された場合は、ポートの使用に関するファイアウォールの制限事項がないことを確認する。	60
SOCKET_CLEANUP	ネットワーク・レイヤがソケット・レイヤをクリーンアップできませんでした。これは、すべての接続が終了した後にだけ発生するエラーです。したがって現在の接続には影響しません。	61

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SOCKET_CLOSE	ネットワーク・レイヤがソケットを閉じることができませんでした。フラッシュされなかった保留中の書き込みが原因で、ネットワーク・セッションが不完全なまま終了した可能性があります (または、終了していない可能性があります)。次の項目を確認してください。 1) ネットワーク接続のどちらか一方の側でエラーが発生した。 2) 接続の反対側は正常に動作している。 3) マシンがネットワークに接続されたままになっており、ネットワークからの応答がある。	62
SOCKET_CONNECT	ネットワーク・レイヤがソケットを接続できませんでした。次の項目を確認してください。 1) マシンがネットワークに接続されている。 2) ソケット・レイヤは適切に初期化されている。 3) 適切なホスト・マシンとポートが指定されている。 4) ホスト・サーバが正常に稼働しており、適切なポートで受信している。 5) ホスト・マシンが、適切なソケット・タイプ (TCP/IP と UDP) で受信を行っている。 6) client_port オプションが使用された場合は、ポートの使用に関するファイアウォールの制限事項がないことを確認する。 7) オープンしているソケット数について、デバイスに制限が設けられている場合は、その値に達していないことを確認する。 8) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。	63

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SOCKET_CREATE_TCPIP	ネットワーク・レイヤが TCP/IP ソケットを作成できませんでした。次の項目を確認してください。 1) マシンがネットワークに接続されている。 2) ソケット・レイヤは適切に初期化されている。 5) オープンしているソケット数について、デバイスに制限が設けられている場合は、その値に達していないことを確認する。 6) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。	58
SOCKET_CREATE_UDP	ネットワーク・レイヤが UDP ソケットを作成できませんでした。次の項目を確認してください。 1) マシンがネットワークに接続されている。 2) ソケット・レイヤは適切に初期化されている。 3) client_port オプションが使用された場合は、指定したポートがまだ使用されていないことを確認する。クライアント・ポートを 1 つだけ指定した場合は、NNN-NNN などの範囲を使用することを検討してください。範囲が指定されていた場合は、その範囲を広げるか、別の範囲にすることを検討してください。 4) client_port オプションが使用された場合は、ポートの使用に関するファイアウォールの制限事項がないことを確認する。 5) オープンしているソケット数について、デバイスに制限が設けられている場合は、その値に達していないことを確認する。 6) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。	59
SOCKET_GET_HOST_BY_ADDR	ネットワーク・レイヤが、IP アドレスを使用してホスト名を取得できませんでした。このエラーは、パブリケーション時に定義されましたが使用されませんでした。	72

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SOCKET_GET_NAME	ネットワーク・レイヤがソケットのローカル名を判別できませんでした。TCP/IP 接続には、各接続の両端にポート専用に付加されたソケットがあります。ソケットのローカル名にはこのポート番号が含まれており、これは、接続時にネットワークによって割り当てられます。次の項目を確認してください。 1) マシンがネットワークに接続されたままになっており、ネットワークからの応答がある。 2) 接続の反対側は正常に動作している。 3) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。	64
SOCKET_GET_OPTION	ネットワーク・レイヤがソケット・オプションを取得できませんでした。このエラーは、接続が失われたことを示す最初の徴候であると考えられます。次の項目を確認してください。 1) マシンがネットワークに接続されたままになっており、ネットワークからの応答がある。 2) 接続の反対側は正常に動作している。 3) 使用できるシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。	65
SOCKET_HOST_NAME_NOT_FOUND	指定されたホスト名が見つかりませんでした。次の項目を確認してください。 1) ホスト名が正しく指定されている。 2) ホストがアクセス可能である。多くのシステムには、名前を指定したホストへのアクセスを検証できる ping ユーティリティが組み込まれています。 3) ドメイン・ネーム・サーバ (DNS) またはそれに相当するものが使用できる。DNS を使用できない場合は、ホスト名の代わりにホストの IP 番号 (例：NNN.NNN.NNN.NNN) を指定してみてください。 4) HOSTS ファイルに、ホスト名を IP 番号にマッピングするエントリが入っている。	57

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SOCKET_LISTEN	サーバがソケットで受信できません。バックログとは、ある一定期間に保留中になっている可能性がある、キューイングされた接続要求の最大数のことです。次の項目を確認してください。 1) マシンがネットワークに接続されたままになっており、ネットワークからの応答がある。 2) 現在のマシンでソケット・リスナが動作するのを妨げるファイアウォールやその他の制限事項がない。 3) マシンに制限がある場合、バックログ設定が制限内に収まっている。 4) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。	67
SOCKET_LIVENESS_OUT_OF_RANGE	無効な活性タイムアウト値が指定されました。活性タイムアウト値は、0 ー 65535 の間の整数にしてください。	200
SOCKET_LOCALHOST_NAME_NOT_FOUND	ネットワーク・レイヤが "localhost" の IP アドレスを判別できませんでした。次の項目を確認してください。 1) ドメイン・ネーム・サーバ (DNS) またはそれに相当するものが使用できる。DNS が使用できない場合には、代わりに localhost の IP 番号を明示的に指定してみます (通常は 127.0.0.1)。 2) HOSTS ファイルに、"localhost" 名を IP 番号にマッピングするエントリが入っている。 3) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。	71
SOCKET_PORT_OUT_OF_RANGE	不正なポート番号を指定しています。ポート番号は、0 ー 65535 の間の整数にしてください。	74

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
SOCKET_SELECT	ネットワーク・レイヤで、読み込みまたは書き込み可能状態のソケットに対して待ち状態に入ろうとするエラーが発生しました。次の項目を確認してください。 1) ネットワークにマシンが接続されており、ネットワークからの応答がある。 2) 接続の反対側は正常に動作している。 3) 使用できるシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。	69
SOCKET_SET_OPTION	ネットワーク・レイヤがソケット・オプションを設定できませんでした。このエラーは、接続が失われたことを示す最初の徴候であると考えられます。次の項目を確認してください。 1) マシンがネットワークに接続されたままになっており、ネットワークからの応答がある。 2) 接続の反対側は正常に動作している。 3) 使用できるシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。	66
SOCKET_SHUTDOWN	ネットワーク・レイヤがソケットをシャトダウンできませんでした。次の項目を確認してください。 1) ネットワークにマシンが接続されており、ネットワークからの応答がある。 2) 接続の反対側は正常に動作している。 3) 使用できるシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。	68
SOCKET_STARTUP	ネットワーク・レイヤがソケット・レイヤを初期化できませんでした。次の項目を確認してください。 1) ソケット・レイヤが適切にインストールされている。正しいネットワーク・インタフェース・ライブラリが存在し、アクセス可能になっていなければなりません。 2) ネットワークにマシンが接続されており、ネットワークからの応答がある。 3) 使用可能なシステム・リソースが十分にある。不足していれば、システム・リソースを解放してください。	70

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
UNEXPECTED_HTTP_REQUEST_TYPE	この HTTP 要求タイプはこの時点では予期されていませんでした。Mobile Link クライアントではない HTTP クライアントが主な原因と考えられます。	234
UNRECOGNIZED_TLS_TYPE	TLS タイプが無効です。有効なタイプについては、マニュアルを参照してください。	240
VALUE_OUT_OF_RANGE	指定された値が、パラメータに許可される値の範囲内ではありませんでした。マニュアルでパラメータの説明を参照して、値に許可される範囲を確認してください。	233
WOULD_BLOCK	ブロッキングが不要または予期されないところで、要求された操作がブロックした可能性があります。	13
WRITE	指定されたバイト数をネットワーク・レイヤに書き込むことができません。書き込みは、より規模の大きいネットワーク操作の一部として行われます。たとえば、ネットワーク・レイヤによっては、サブ・レイヤを設定して、ここで上位レイヤの基本操作の一部として読み込みと書き込みを行います。通常、書き込みエラーは次のような事柄が原因で発生します。 1) ネットワークに、書き込みを失敗させる問題が生じた。もう一度接続と操作を行ってください。 2) 接続がタイムアウトした。もう一度接続と操作を行ってください。 3) 反対側の接続が完全に終了した。クライアントかサーバ (またはその両方) のエラー・ログを参照して、接続が削除された理由を調べてください。出力されたエラー・ログを調べて原因を解明し、もう一度操作を行ってください。 4) 接続の反対側で処理がアボートされた。クライアントかサーバ (またはその両方) のエラー出力ログを参照して、処理がアボートされた理由を調べてください。通常の方法以外のやり方で処理が停止された場合は、出力ログにエラーは記録されません。もう一度接続と操作を行ってください。 5) システム・リソースが不足しており、書き込みを実行できない。システム・リソースを解放してから、もう一度接続と操作を行ってください。再度行ってもうまくいかない場合は、ネットワーク管理者に問い合わせてください。	9

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
WRITE_TIMEOUT	ネットワーク・レイヤによって指定されたバイト数を指定時間内に書き込めませんでした。ネットワークが正常に機能しており、受信側アプリケーションがまだ動作中であることを確認してください。	202

参照

- 「StreamErrorCode プロパティ」 930 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULStreamErrorContext 列挙

UL 拡張： ストリーム・エラーが発生したときに実行される基本的なネットワーク・オペレーションを列挙します。

構文

Visual Basic

Public Enum **ULStreamErrorContext**

C#

public enum **ULStreamErrorContext**

メンバ

メンバ名	説明	値
CLOSE	CLOSE(6)	6
CREATE	CREATE(3)	3
DESTROY	DESTROY(4)	4
END_READ	END_READ(11)	11
END_WRITE	END_WRITE(10)	10
GETVALUE	GETVALUE(14)	14
OPEN	OPEN(5)	5
READ	READ(7)	7
REGISTER	REGISTER(1)	1
SOFTSHUTDOWN	SOFTSHUTDOWN(13)	13
UNKNOWN	UNKNOWN(0)	0
UNREGISTER	UNREGISTER(2)	2
WRITE	WRITE(8)	8
WRITE_FLUSH	WRITE_FLUSH(9)	9
YIELD	YIELD(12)	12

参照

- 「StreamErrorContext プロパティ」 930 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULStreamErrorID 列挙

UL 拡張： 同期中にエラーを報告する可能性のあるネットワーク・レイヤを列挙します。

構文

Visual Basic

Public Enum **ULStreamErrorID**

C#

public enum **ULStreamErrorID**

メンバ

メンバ名	説明	値
ACTIVESYNC	ACTIVESYNC(22)	22
CERTICOM	CERTICOM(11)	11
CERTICOM_SSL	CERTICOM_SSL(13)	13
CERTICOM_TLS	CERTICOM_TLS(14)	14
DH_CAST	DH_CAST(9)	9
EMAIL	EMAIL(20)	20
FAKE	FAKE(2)	2
FILE	FILE(21)	21
HTTP	HTTP(7)	7
HTTPS	HTTPS(8)	8
JAVA_CERTICOM	JAVA_CERTICOM(12)	12
JAVA_RSA	JAVA_RSA(24)	24
NETTECH	NETTECH(5)	5
OPEN_SSL_RSA	OPEN_SSL_RSA(25)	25
PALM_CONDUIT	PALM_CONDUIT(3)	3
PALM_SS	PALM_SS(4)	4
PALM_SSL	PALM_SSL(26)	26
REPLAY	REPLAY(17)	17

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
RIMBB	RIMBB(6)	6
RSA_TLS	RSA_TLS(23)	23
SECURE	SECURE(10)	10
SERIAL	SERIAL(1)	1
STRM	STRM(18)	18
TCPIP	TCPIP(0)	0
[UDP]	UDP(19)	19
WIRELESS	WIRELESS(16)	16
WIRESTRM	WIRESTRM(15)	15

参照

- 「StreamErrorID プロパティ」 931 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULStreamType 列挙

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。

構文

Visual Basic

Public Enum **ULStreamType**

C#

public enum **ULStreamType**

備考

特定のストリームのタイプの設定方法については、「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」『Mobile Link - クライアント管理』を参照してください。

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 承認の暗号化には、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 10 - 紹介』を参照してください。

メンバ

メンバ名	説明	値
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 サーバは Web サーバとして機能します。Ultra Light アプリケーションは、サーバへのデータ送信のために POST 要求を送り、サーバからのデータの読み込みのために GET 要求を送ります。	1
HTTPS	HTTPS を介して同期します (トランスポート・レイヤ・セキュリティを適用した HTTP)。	2
TCPIP	TCP/IP を介して同期します。	0
TLS	TLS (TCP/IP を介して同期)	3

参照

- 「Stream プロパティ」 907 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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を表します。このクラスは継承できません。

構文

Visual Basic

Public NotInheritable Class **ULSyncParms**

C#

public sealed class **ULSyncParms**

備考

このクラスにはコンストラクタがありません。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

- 「ULSyncParms メンバ」 897 ページ
- 「ULConnection クラス」 555 ページ
- 「SyncParms プロパティ」 567 ページ
- 「Synchronize() メソッド」 587 ページ
- 「DownloadOnly プロパティ」 900 ページ
- 「PingOnly プロパティ」 903 ページ
- 「ResumePartialDownload プロパティ」 904 ページ
- 「UploadOnly プロパティ」 909 ページ
- 「ULSQLCode 列挙」 859 ページ
- 「Synchronize() メソッド」 587 ページ
- 「Stream プロパティ」 907 ページ
- 「Version プロパティ」 910 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULSyncParms メンバ

パブリック・プロパティ

メンバ名	説明
AuthenticationParms プロパティ	カスタム・ユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。
CheckpointStore プロパティ	クライアントが、追加のストア・チェックポイントを実行して、同期中にデータベース・ストアのサイズの増加を制御するかどうかを指定します。
DisableConcurrency プロパティ	同期の実行中に Ultra Light への同時アクセスを無効にするか、有効にするかを指定します。
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 サーバに ping のみを行うかどうかを指定します。
PublicationMask 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 サーバにダウンロード確認を送信するかどうかを指定します。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。
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を指定します。
TableOrder プロパティ	統合データベースにテーブルがアップロードされる順序を指定します。
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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
UserName プロパティ	Mobile Link サーバが Mobile Link クライアントをユニークに識別するユーザ名です。
Version プロパティ	使用する同期スクリプトを指定します。

パブリック・メソッド

メンバ名	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。

参照

- 「ULSyncParms クラス」 896 ページ
- 「ULConnection クラス」 555 ページ
- 「SyncParms プロパティ」 567 ページ
- 「Synchronize() メソッド」 587 ページ
- 「DownloadOnly プロパティ」 900 ページ
- 「PingOnly プロパティ」 903 ページ
- 「ResumePartialDownload プロパティ」 904 ページ
- 「UploadOnly プロパティ」 909 ページ
- 「ULSQLCode 列挙」 859 ページ
- 「Synchronize() メソッド」 587 ページ
- 「Stream プロパティ」 907 ページ
- 「Version プロパティ」 910 ページ

AuthenticationParms プロパティ

カスタム・ユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。

構文

Visual Basic

Public Property **AuthenticationParms** As String

C#

public string **AuthenticationParms** { get; set; }

プロパティ値

それぞれに認証パラメータが格納された、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、認証パラメータは指定されません。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

最初の 255 文字のみが使用されます。また、各文字列は 128 文字以下であることが必要です (長すぎる文字列は、Mobile Link サーバに送信されるときにトランケートされます)。

参照

- 「ULSyncParms クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ

CheckpointStore プロパティ

クライアントが、追加のストア・チェックポイントを実行して、同期中にデータベース・ストアのサイズの増加を制御するかどうかを指定します。

構文

Visual Basic

Public Property **CheckpointStore** As Boolean

C#

public bool **CheckpointStore** { get; set; }

プロパティ値

クライアントが追加のストア・チェックポイントを実行する必要があると指定する場合は true。デフォルトは false で、必要なチェックポイントのみが実行されます。

備考

チェックポイント操作は、アプリケーションの I/O 操作を増やすため、同期が低速になります。このオプションは、多くの更新を伴う大量のダウンロードに最適です。低速なフラッシュ・メモリを使用するデバイスでは、追加のチェックポイントの実行によって、望ましくないパフォーマンスの低下が生じることがあります。

参照

- 「ULSyncParms クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ

DisableConcurrency プロパティ

同期の実行中に Ultra Light への同時アクセスを無効にするか、有効にするかを指定します。

構文

Visual Basic

Public Property **DisableConcurrency** As Boolean

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public bool **DisableConcurrency** { get; set; }

プロパティ値

同期の実行中に Ultra Light への同時アクセス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。

備考

デフォルトでは、1 つのスレッドが同期しているときに、別のスレッドが Ultra Light の操作を行うことがあります。同時同期が無効になっていると、別のスレッドは同期が完了するまで Ultra Light の呼び出しをブロックします。

参照

- 「ULSyncParms クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ

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。

構文

Visual Basic

Public Property **DownloadOnly** As Boolean

C#

public bool **DownloadOnly** { get; set; }

プロパティ値

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。

備考

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 クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ
- 「UploadOnly プロパティ」 909 ページ
- 「DownloadOnly プロパティ」 900 ページ
- 「PingOnly プロパティ」 903 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

- 「ResumePartialDownload プロパティ」 904 ページ
- 「UploadOnly プロパティ」 909 ページ
- 「ULSQLCode 列挙」 859 ページ
- 「Synchronize() メソッド」 587 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。

構文

Visual Basic

Public Property **KeepPartialDownload** As Boolean

C#

public bool **KeepPartialDownload** { get; set; }

プロパティ値

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。

備考

Ultra Light.NET では、通信エラーや、ユーザによる 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 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、PublicationMask 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

Copyright © 2007, iAnywhere Solutions, Inc.

だけです。KeepPartialDownload フィールドと DisableConcurrency 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。

詳細については、「失敗したダウンロードの再開」『Mobile Link - サーバ管理』を参照してください。

参照

- 「ULSyncParms クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ
- 「PartialDownloadRetained プロパティ」 929 ページ
- 「ResumePartialDownload プロパティ」 904 ページ
- 「RollbackPartialDownload メソッド」 586 ページ
- 「ResumePartialDownload プロパティ」 904 ページ
- 「UserName プロパティ」 909 ページ
- 「DisableConcurrency プロパティ」 899 ページ
- 「RollbackPartialDownload メソッド」 586 ページ

NewPassword プロパティ

UserName で指定されたユーザの新しい Mobile Link パスワードを指定します。

構文

Visual Basic

Public Property **NewPassword** As String

C#

public string **NewPassword** { get; set; }

プロパティ値

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。

備考

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 クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ
- 「UserName プロパティ」 909 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

Password プロパティ

UserName で指定されたユーザの Mobile Link パスワードです。

構文

Visual Basic

Public Property **Password** As String

C#

public string **Password** { get; set; }

プロパティ値

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。

備考

Mobile Link ユーザ名とパスワードは他のデータベース・ユーザ 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 サーバに対して識別し、認証するために使用されます。

参照

- 「ULSyncParms クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ
- 「NewPassword プロパティ」 902 ページ
- 「UserName プロパティ」 909 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 サーバに ping のみを行うかどうかを指定します。

構文

Visual Basic

Public Property **PingOnly** As Boolean

C#

public bool **PingOnly** { get; set; }

プロパティ値

クライアントが Mobile Link サーバに 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。

備考

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

Copyright © 2007, iAnywhere Solutions, Inc.

せん。これらの複数のパラメータが 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 クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ
- 「DownloadOnly プロパティ」 900 ページ
- 「PingOnly プロパティ」 903 ページ
- 「ResumePartialDownload プロパティ」 904 ページ
- 「UploadOnly プロパティ」 909 ページ
- 「ULSQLCode 列挙」 859 ページ
- 「Synchronize() メソッド」 587 ページ

PublicationMask プロパティ

同期させるパブリケーションを指定します。

構文

Visual Basic

Public Property **PublicationMask** As Integer

C#

public int **PublicationMask** { get; set; }

プロパティ値

パブリケーション・マスク のビット処理の組み合わせとして、特別値 ULPublicationSchema.SYNC_ALL_PUBS、または特別値 ULPublicationSchema.SYNC_ALL_DB。デフォルトは ULPublicationSchema.SYNC_ALL_DB です。パブリケーション・マスクの詳細については、「ULPublicationSchema クラス」 810 ページを参照してください。

参照

- 「ULSyncParms クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ
- 「Mask プロパティ」 812 ページ
- 「SYNC_ALL_PUBS フィールド」 812 ページ
- 「SYNC_ALL_DB フィールド」 811 ページ
- 「SYNC_ALL_DB フィールド」 811 ページ
- 「ULPublicationSchema クラス」 810 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Property **ResumePartialDownload** As Boolean

C#

public bool **ResumePartialDownload** { get; set; }

プロパティ値

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。

備考

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。

部分的なダウンロードの詳細については、「KeepPartialDownload プロパティ」 901 ページを参照してください。

参照

- 「ULSyncParms クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ
- 「DownloadOnly プロパティ」 900 ページ
- 「PingOnly プロパティ」 903 ページ
- 「ResumePartialDownload プロパティ」 904 ページ
- 「UploadOnly プロパティ」 909 ページ
- 「ULSQLCode 列挙」 859 ページ
- 「Synchronize() メソッド」 587 ページ
- 「PartialDownloadRetained プロパティ」 929 ページ

SendColumnNames プロパティ

同期中に、クライアントが Mobile Link サーバにカラム名を送信するかどうかを指定します。

構文

Visual Basic

Public Property **SendColumnNames** As Boolean

C#

public bool **SendColumnNames** { get; set; }

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

クライアントがカラム名を Mobile Link サーバ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。

備考

カラム名は、Mobile Link サーバ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 サーバがダイレクト・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。

参照

- 「ULSyncParms クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ

SendDownloadAck プロパティ

同期中に、クライアントが Mobile Link サーバにダウンロード確認を送信するかどうかを指定します。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。

構文

Visual Basic

Public Property **SendDownloadAck** As Boolean

C#

public bool **SendDownloadAck** { get; set; }

プロパティ値

True に設定すると、クライアントが Mobile Link サーバ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。デフォルト値は False です。

備考

クライアントがダウンロード確認を送信する場合、Mobile Link サーバのデータベース・ワーカ・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 サーバは、次の同期のため、より早く解放されます。

参照

- 「ULSyncParms クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Property **Stream** As ULStreamType

C#

public ULStreamType **Stream** { get; set; }

プロパティ値

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。

備考

ほとんどの同期ストリームでは、Mobile Link サーバ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が必要です。これらのパラメータは、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

- 「ULSyncParms クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ
- 「ULStreamType 列挙」 895 ページ
- 「StreamParms プロパティ」 907 ページ
- 「ULStreamType 列挙」 895 ページ

StreamParms プロパティ

同期ストリームの設定パラメータを指定します。

構文

Visual Basic

Public Property **StreamParms** As String

C#

public string **StreamParms** { get; set; }

プロパティ値

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 © 2007, iAnywhere Solutions, Inc.

備考

特定のストリームのタイプの設定方法については、「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」『Mobile Link - クライアント管理』を参照してください。

StreamParms は、同期ストリームに使用されるすべてのパラメータが含まれている文字列です。パラメータは、名前=値の組み合わせ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 クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ
- 「Stream プロパティ」 907 ページ
- 「ULStreamType 列挙」 895 ページ

TableOrder プロパティ

統合データベースにテーブルがアップロードされる順序を指定します。

構文

Visual Basic

Public Property **TableOrder** As String

C#

public string **TableOrder** { get; set; }

プロパティ値

テーブル名がカンマで区切られたリスト形式の文字列。テーブル名は、一重または二重引用符で囲むことができます。デフォルト値は null 参照 (Visual Basic の Nothing) で、テーブルのデフォルトの順序は上書きされません。

備考

統合データベースの外部キーがリモートの Ultra Light データベースの外部キーに一致し、外部キー循環がない場合は、通常、この機能を使用する必要はありません。しかし、外部キー循環に含まれるテーブルがある場合は、循環に含まれるすべてのテーブルを TableOrder フィールドにリストする必要があります。統合データベース内の外部キーの関係が異なるテーブルがある場合も、これらのテーブルを TableOrder フィールドにリストします。

指定しないテーブルは、リモート・データベースで定義した外部キーに基づいて適切に保存されます。

参照

- 「ULSyncParms クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Property **UploadOnly** As Boolean

C#

public bool **UploadOnly** { get; set; }

プロパティ値

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。

備考

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 クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ
- 「DownloadOnly プロパティ」 900 ページ
- 「PingOnly プロパティ」 903 ページ
- 「ResumePartialDownload プロパティ」 904 ページ
- 「UploadOnly プロパティ」 909 ページ
- 「ULSQLCode 列挙」 859 ページ
- 「Synchronize() メソッド」 587 ページ

UserName プロパティ

Mobile Link サーバが Mobile Link クライアントをユニークに識別するユーザ名です。

構文

Visual Basic

Public Property **UserName** As String

C#

public string **UserName** { get; set; }

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

ユーザ名を指定する文字列。このパラメータ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。

備考

Mobile Link サーバでは、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。このユーザ名とパスワードは他のデータベース・ユーザ 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 サーバに対して識別し、認証するために使用されます。

参照

- 「ULSyncParms クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ
- 「Password プロパティ」 903 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。

構文

Visual Basic

```
Public Property Version As String
```

C#

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

プロパティ値

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。

備考

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。

参照

- 「ULSyncParms クラス」 896 ページ
- 「ULSyncParms メンバ」 897 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。

 Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Sub CopyFrom( _
  ByVal src As ULSyncParms _
)
```

C#

```
public void CopyFrom(
  ULSyncParms src
);
```

パラメータ

- **src**　コピー元のオブジェクト。

参照

- 「ULSyncParms クラス」 896 ページ
- 「ULSyncParms メンバ」 897 ページ
- 「ULSyncParms クラス」 896 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Class **ULSyncProgressData**

C#

public class **ULSyncProgressData**

参照

- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressListener インタフェース」 923 ページ

ULSyncProgressData メンバ

パブリック・フィールド

メンバ名	説明
FLAG_IS_BLOCKING フィールド	Mobile Link サーバ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。このフィールドは定数で、読み込み専用です。

パブリック・プロパティ

メンバ名	説明
ErrorMessage プロパティ	同期中に発生したエラーを説明するエラー・メッセージを返します。
Flags プロパティ	現在の状態に関連する追加情報を示す、現在の同期フラグを返します。
ReceivedBytes プロパティ	現在までに受信したバイト数を返します。この情報は、すべてのステータスで更新されます。
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。
SQLCode プロパティ	同期の SQL コードを返します。
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。この情報は、すべてのステータスで更新されます。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。
SyncParms プロパティ	接続の SyncParms オブジェクトへの参照を返します。
SyncResult プロパティ	接続の SyncResult オブジェクトへの参照を返します。このオブジェクトは、「ULSyncProgressState 列挙」 925 ページイベントと「ULSyncProgressState 列挙」 925 ページイベントでのみ更新されます。
TableCount プロパティ	現在までに送信または受信したテーブルの数です (TableTotal の TableCount)。
TableIndex 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 ー DatabaseSchema.TableCount までの番号が付けられています)。
TableTotal プロパティ	同期中のテーブルの数を返します。

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressListener インタフェース」 923 ページ

FLAG_IS_BLOCKING フィールド

Mobile Link サーバ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このフィールドは定数で、読み込み専用です。

構文

Visual Basic

Public Shared **FLAG_IS_BLOCKING** As Integer

C#

public const int **FLAG_IS_BLOCKING** ;

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ErrorMessage プロパティ

同期中に発生したエラーを説明するエラー・メッセージを返します。

構文

Visual Basic

Public Readonly Property **ErrorMessage** As String

C#

public string **ErrorMessage** { get;}

プロパティ値

同期中に発生したエラーを説明する文字列。

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 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。

構文

Visual Basic

Public Readonly Property **Flags** As Integer

C#

public int **Flags** { get;}

プロパティ値

フラグの組み合わせを保持する整数。

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「FLAG_IS_BLOCKING フィールド」 913 ページ

ReceivedBytes プロパティ

現在までに受信したバイト数を返します。この情報は、すべてのステータスで更新されます。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **ReceivedBytes** As Long

C#

public long **ReceivedBytes** { get;}

プロパティ値

現在までに受信したバイト数。

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「ULSyncProgressState 列挙」 925 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。

構文

Visual Basic

Public Readonly Property **ReceivedDeletes** As Integer

C#

public int **ReceivedDeletes** { get;}

プロパティ値

現在までに受信した削除済みのローの数。

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「ULSyncProgressState 列挙」 925 ページ

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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **ReceivedInserts** As Integer

C#

public int **ReceivedInserts** { get;}

プロパティ値

現在までに受信した挿入済みのローの数。

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「ULSyncProgressState 列挙」 925 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。

構文

Visual Basic

Public Readonly Property **ReceivedUpdates** As Integer

C#

public int **ReceivedUpdates** { get;}

プロパティ値

現在までに受信した更新済みのローの数。

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「ULSyncProgressState 列挙」 925 ページ

SQLCode プロパティ

同期の SQL コードを返します。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Readonly Property **SQLCode** As ULSQLCode

C#

public ULSQLCode **SQLCode** { get;}

プロパティ値

同期エラーの「ULSQLCode 列挙」 859 ページ値。

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 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。この情報は、すべてのステータスで更新されます。

構文

Visual Basic

Public Readonly Property **SentBytes** As Long

C#

public long **SentBytes** { get;}

プロパティ値

現在までに送信されたバイト数。

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「ULSyncProgressState 列挙」 925 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。

構文

Visual Basic

Public Readonly Property **SentDeletes** As Integer

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public int **SentDeletes** { get;}

プロパティ値

現在までに送信された削除済みのローの数。

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「ULSyncProgressState 列挙」 925 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。

構文

Visual Basic

Public Readonly Property **SentInserts** As Integer

C#

public int **SentInserts** { get;}

プロパティ値

現在までに送信された挿入済みのローの数。

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「ULSyncProgressState 列挙」 925 ページ

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。

構文

Visual Basic

Public Readonly Property **SentUpdates** As Integer

C#

public int **SentUpdates** { get;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

現在までに送信された更新済みのローの数。

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「ULSyncProgressState 列挙」 925 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。

構文

Visual Basic

Public Readonly Property **State** As ULSyncProgressState

C#

public ULSyncProgressState **State** { get;}

プロパティ値

現在の同期のステータスを指定する「ULSyncProgressState 列挙」 925 ページ値の 1 つ。

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 ページ

SyncParms プロパティ

接続の SyncParms オブジェクトへの参照を返します。

構文

Visual Basic

Public Readonly Property **SyncParms** As ULSyncParms

C#

public ULSyncParms **SyncParms** { get;}

プロパティ値

「SyncParms プロパティ」 567 ページオブジェクトへの参照。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「SyncParms プロパティ」 567 ページ

SyncResult プロパティ

接続の SyncResult オブジェクトへの参照を返します。このオブジェクトは、「ULSyncProgressState 列挙」 925 ページイベントと「ULSyncProgressState 列挙」 925 ページイベントでのみ更新されます。

構文

Visual Basic

Public Readonly Property **SyncResult** As ULSyncResult

C#

public ULSyncResult **SyncResult** { get;}

プロパティ値

「SyncResult プロパティ」 567 ページオブジェクトへの参照。

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「ULSyncProgressState 列挙」 925 ページ

TableCount プロパティ

現在までに送信または受信したテーブルの数です (TableTotal の TableCount)。

構文

Visual Basic

Public Readonly Property **TableCount** As Integer

C#

public int **TableCount** { get;}

プロパティ値

送信または受信したテーブルの数。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「ULSyncProgressState 列挙」 925 ページ

TableIndex 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 ー DatabaseSchema.TableCount までの番号が付けられています)。

構文

Visual Basic

Public Readonly Property **TableIndex** As Integer

C#

public int **TableIndex** { get;}

プロパティ値

現在同期中のテーブルのインデックス。テーブルには、1 ー「TableCount プロパティ」 668 ページ までの番号が付けられています。

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「ULSyncProgressState 列挙」 925 ページ

TableTotal プロパティ

同期中のテーブルの数を返します。

構文

Visual Basic

Public Readonly Property **TableTotal** As Integer

C#

public int **TableTotal** { get;}

プロパティ値

同期中のテーブルの数。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULSyncProgressData クラス」 912 ページ
- 「ULSyncProgressData メンバ」 912 ページ
- 「ULSyncProgressState 列挙」 925 ページ
- 「ULSyncProgressState 列挙」 925 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULSyncProgressListener インタフェース

UL 拡張： 同期プログレス・イベントを受信するリスナ・インタフェースです。

構文

Visual Basic

Public Interface **ULSyncProgressListener**

C#

public interface **ULSyncProgressListener**

参照

- 「ULSyncProgressListener メンバ」 923 ページ
- 「Synchronize(ULSyncProgressListener) メソッド」 588 ページ

ULSyncProgressListener メンバ

パブリック・メソッド

メンバ名	説明
SyncProgressed メソッド	ユーザに進行状況を通知するために、同期処理中に起動されます。このメソッドは true を返して同期をキャンセルするか、false を返して続行する必要があります。

参照

- 「ULSyncProgressListener インタフェース」 923 ページ
- 「Synchronize(ULSyncProgressListener) メソッド」 588 ページ

SyncProgressed メソッド

ユーザに進行状況を通知するために、同期処理中に起動されます。このメソッドは true を返して同期をキャンセルするか、false を返して続行する必要があります。

構文

Visual Basic

```
Public Function SyncProgressed( _
   ByVal data As ULSyncProgressData _
) As Boolean
```

C#

```
public bool SyncProgressed(
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
  ULSyncProgressData data
);
```

パラメータ

- **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 true を返して同期をキャンセルするか、false を返して続行する必要があります。

備考

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

- 「ULSyncProgressListener インタフェース」 923 ページ
- 「ULSyncProgressListener メンバ」 923 ページ
- 「ULSyncProgressData クラス」 912 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULSyncProgressState 列挙

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。

構文

Visual Basic

Public Enum **ULSyncProgressState**

C#

public enum **ULSyncProgressState**

メンバ

メンバ名	説明	値
STATE_CANCELLED	同期がキャンセルされました。	15
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。「ReceivedBytes プロパティ」 914 ページ、「ReceivedInserts プロパティ」 915 ページ、「ReceivedUpdates プロパティ」 916 ページ、「ReceivedDeletes プロパティ」 915 ページを参照してください。	9
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_DISCONNECTING	同期ストリームが閉じられようとしています。	11
STATE_DONE	同期が正常に完了しました。接続の「SyncResult プロパティ」 920 ページオブジェクトが更新されています。	12
STATE_ERROR	同期は完了しましたが、エラーが発生しました。詳細については、「SyncResult プロパティ」 920 ページ、「ErrorMessage プロパティ」 914 ページ、「SQLCode プロパティ」 916 ページを確認してください。	13
STATE_FINISHING_UPLOAD	アップロードが完了しています。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。「SentBytes プロパティ」 917 ページ、「SentInserts プロパティ」 918 ページ、「SentUpdates プロパティ」 918 ページ、「SentDeletes プロパティ」 917 ページを参照してください。	5

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明	値
STATE_LAST	同期の最後のステータスです。このステータスは、常にその他のステータスの後に入力されます。他の ULSyncProgressData フィールドには、有効なデータがないことがあります。	128
STATE_RECEIVING_DATA	現在のテーブルのデータを受信しています。「ReceivedBytes プロパティ」 914 ページ、「ReceivedInserts プロパティ」 915 ページ、「ReceivedUpdates プロパティ」 916 ページ、「ReceivedDeletes プロパティ」 915 ページが更新されています。	8
STATE_RECEIVING_TABLE	テーブルの受信中です。進行状況をモニタするには、「TableIndex プロパティ」 921 ページと「TableCount プロパティ」 920 ページを使用します。	7
STATE_RECEIVING_UPLOAD_ACK	アップロード完了の確認の受信中	6
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。	14
STATE_SENDING_DATA	現在のテーブルのデータが送信されています。「SentBytes プロパティ」 917 ページ、「SentInserts プロパティ」 918 ページ、「SentUpdates プロパティ」 918 ページ、「SentDeletes プロパティ」 917 ページが更新されています。	4
STATE_SENDING_DOWNLOAD_ACK	ダウンロードが完了したという確認が送信されています。	10
STATE_SENDING_HEADER	同期ストリームが開かれ、ヘッダが送信されようとしています。	2
STATE_SENDING_TABLE	テーブルの送信中です。進行状況をモニタするには、「TableIndex プロパティ」 921 ページと「TableCount プロパティ」 920 ページを使用します。	3
STATE_STARTING	同期処理はまだ行われていません。	0

参照

- 「ULSyncProgressData クラス」 912 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULSyncResult クラス

UL 拡張： 前回の同期のステータスを表します。

構文

Visual Basic

Public Class **ULSyncResult**

C#

public class **ULSyncResult**

備考

このクラスにはコンストラクタがありません。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

- 「ULSyncResult メンバ」 927 ページ
- 「SyncResult プロパティ」 567 ページ
- 「Synchronize() メソッド」 587 ページ

ULSyncResult メンバ

パブリック・プロパティ

メンバ名	説明
AuthStatus プロパティ	前回行わ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。
StreamErrorContext プロパティ	ストリーム・エラーが発生したときに実行される基本的なネットワーク・オペレーションを返します。
StreamErrorID プロパティ	エラーをレポートするネットワーク・レイヤの ID 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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。

参照

- 「ULSyncResult クラス」 927 ページ
- 「SyncResult プロパティ」 567 ページ
- 「Synchronize() メソッド」 587 ページ

AuthStatus プロパティ

前回行われた同期の認証ステータス・コードを返します。

構文

Visual Basic

Public Readonly Property **AuthStatus** As ULAuthStatusCode

C#

public ULAuthStatusCode **AuthStatus** { get;}

プロパティ値

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

- 「ULSyncResult クラス」 927 ページ
- 「ULSyncResult メンバ」 927 ページ
- 「ULAuthStatusCode 列挙」 479 ページ

AuthValue プロパティ

カスタム・ユーザ認証同期スクリプトからの戻り値を返します。

構文

Visual Basic

Public Readonly Property **AuthValue** As Long

C#

public long **AuthValue** { get;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

カスタム・ユーザ認証同期スクリプトから返された long integer。

参照

- 「ULSyncResult クラス」 927 ページ
- 「ULSyncResult メンバ」 927 ページ

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。

構文

Visual Basic

Public Readonly Property **IgnoredRows** As Boolean

C#

public bool **IgnoredRows** { get;}

プロパティ値

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

- 「ULSyncResult クラス」 927 ページ
- 「ULSyncResult メンバ」 927 ページ
- 「DownloadOnly プロパティ」 900 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。

構文

Visual Basic

Public Readonly Property **PartialDownloadRetained** As Boolean

C#

public bool **PartialDownloadRetained** { get;}

プロパティ値

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。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULSyncResult クラス」 927 ページ
- 「ULSyncResult メンバ」 927 ページ
- 「KeepPartialDownload プロパティ」 901 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。

構文

Visual Basic

Public Readonly Property **StreamErrorCode** As ULStreamErrorCode

C#

public ULStreamErrorCode **StreamErrorCode** { get;}

プロパティ値

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。

参照

- 「ULSyncResult クラス」 927 ページ
- 「ULSyncResult メンバ」 927 ページ
- 「ULStreamErrorCode 列挙」 871 ページ
- 「ULStreamErrorCode 列挙」 871 ページ

StreamErrorContext プロパティ

ストリーム・エラーが発生したときに実行される基本的なネットワーク・オペレーションを返します。

構文

Visual Basic

Public Readonly Property **StreamErrorContext** As ULStreamErrorContext

C#

public ULStreamErrorContext **StreamErrorContext** { get;}

プロパティ値

ストリーム・エラーが発生したときに実行される基本的なネットワーク・オペレーションを示す ULStreamErrorContext 値の 1 つ。

Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULSyncResult クラス」 927 ページ
- 「ULSyncResult メンバ」 927 ページ
- 「ULStreamErrorContext 列挙」 892 ページ

StreamErrorID プロパティ

エラーをレポートするネットワーク・レイヤの ID を返します。

構文

Visual Basic

Public Readonly Property **StreamErrorID** As ULStreamErrorID

C#

public ULStreamErrorID **StreamErrorID** { get;}

プロパティ値

エラーをレポートするネットワーク・レイヤの ID を示す ULStreamErrorID 値の 1 つ。

参照

- 「ULSyncResult クラス」 927 ページ
- 「ULSyncResult メンバ」 927 ページ
- 「ULStreamErrorID 列挙」 893 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。

構文

Visual Basic

Public Readonly Property **StreamErrorSystem** As Integer

C#

public int **StreamErrorSystem** { get;}

プロパティ値

ストリーム・エラー・システム固有のコードを示す整数。

参照

- 「ULSyncResult クラス」 927 ページ
- 「ULSyncResult メンバ」 927 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

Timestamp プロパティ

前回の同期のタイムスタンプを返します。

構文

Visual Basic

Public Readonly Property **Timestamp** As Date

C#

public DateTime **Timestamp** { get;}

プロパティ値

前回の同期のタイムスタンプを指定する System.DateTime。

参照

- 「ULSyncResult クラス」 927 ページ
- 「ULSyncResult メンバ」 927 ページ
- 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。

構文

Visual Basic

Public Readonly Property **UploadOK** As Boolean

C#

public bool **UploadOK** { get;}

プロパティ値

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。

参照

- 「ULSyncResult クラス」 927 ページ
- 「ULSyncResult メンバ」 927 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Public Class ULTable
  Inherits ULResultSet
```

C#

```
public class ULTable : ULResultSet
```

備考

このクラスにはコンストラクタがありません。テーブルは、ULCommand の ULCommand.ExecuteTable() を使用して作成されます。

継承： ULResultSet

実装： System.Data.IDataReader、System.Data.IDataRecord、System.IDisposable

参照

- 「ULTable メンバ」 933 ページ
- 「ExecuteTable() メソッド」 541 ページ
- 「ULCommand クラス」 509 ページ
- 「ULResultSet クラス」 814 ページ
- IDataReader
- IDataRecord
- IDisposable

ULTable メンバ

パブリック・プロパティ

メンバ名	説明
Depth プロパティ (ULDataReader から継承)	現在のローのネストの深さを返します。最も外側のテーブルの深さは 0 です。
FieldCount プロパティ (ULDataReader から継承)	このカーソル内のカラム数を返します。
HasRows プロパティ (ULDataReader から継承)	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 (ULDataReader から継承)	**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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
IsClosed プロパティ (ULDataReader から継承)	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 (ULDataReader から継承)	**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。
Item プロパティ (ULDataReader から継承)	指定されたカラムの値を Object のインスタンスとして取得します。
RecordsAffected プロパティ (ULDataReader から継承)	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。SELECT 文または CommandType.TableDirect テーブルの場合、この値は -1 です。
RowCount プロパティ (ULDataReader から継承)	**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。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。
VisibleFieldCount (DbDataReader から継承)	

パブリック・メソッド

メンバ名	説明
AppendBytes メソッド (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULDataReader から継承)	カーソルを閉じます。
Delete メソッド (ULResultSet から継承)	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 (DbDataReader から継承)	
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。
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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	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。
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 (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	**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。ULDbType.Binary 型、ULDbType.LongBinary 型、ULDbType.UniqueIdentifier 型のカラムの場合にのみ有効です。
GetChar メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (DbDataReader から継承)	
GetDataTypeName メソッド (ULDataReader から継承)	指定されたカラムのプロバイダのデータ型の名前を返します。
GetDateTime メソッド (ULDataReader から継承)	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。
GetDecimal メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 から継承)	ULDataReader の反復処理を実行する System.Collections.IEnumerator を返します。
GetFieldType メソッド (ULDataReader から継承)	指定されたカラムに最適な System.Type を返します。
GetFloat メソッド (ULDataReader から継承)	指定されたカラムの値を System.Single として返します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
GetGuid メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	指定されたカラムの値を Int32 として返します。
GetInt64 メソッド (ULDataReader から継承)	指定されたカラムの値を Int64 として返します。
GetName メソッド (ULDataReader から継承)	指定されたカラムの名前を返します。
GetOrdinal メソッド (ULDataReader から継承)	指定されたカラムのカラム ID を返します。
GetProviderSpecificFieldType (DbDataReader から継承)	
GetProviderSpecificValue (DbDataReader から継承)	
GetProviderSpecificValues (DbDataReader から継承)	
GetSchemaTable メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	指定されたカラムの値を UInt32 として返します。
GetUInt64 メソッド (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULDataReader から継承)	現在のローのすべてのカラム値を返します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。 挿入を行う前に ULTable.InsertBegin を呼び出してください。
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 (ULDataReader から継承)	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。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。
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。
MoveAfterLast メソッド (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	**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 (ULDataReader から継承)	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 (ULDataReader から継承)	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 (ULResultSet から継承)	指定されたカラムの値を、System.Boolean を使用して設定します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
SetByte メソッド (ULResultSet から継承)	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 (ULResultSet から継承)	指定されたカラムの値を、System.Bytes を使用して設定します。
SetDBNull メソッド (ULResultSet から継承)	カラムを NULL に設定します。
SetDateTime メソッド (ULResultSet から継承)	指定されたカラムの値を、="System.DateTime を使用して設定します。
SetDecimal メソッド (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	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 (ULResultSet から継承)	指定されたカラムの値を、System.UInt64 を使用して設定します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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 (ULResultSet から継承)	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	テーブルの現在のローを更新する準備を行います。

参照

- 「ULTable クラス」 933 ページ
- 「ExecuteTable() メソッド」 541 ページ
- 「ULCommand クラス」 509 ページ
- 「ULResultSet クラス」 814 ページ
- IDataReader
- IDataRecord
- IDisposable

Schema プロパティ

テーブル・スキーマを保持します。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。

構文

Visual Basic

Public Readonly Property **Schema** As ULTableSchema

C#

public ULTableSchema **Schema** { get;}

プロパティ値

テーブル・スキーマを示す ULTableSchema オブジェクト。

備考

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示します。

参照

- 「ULTable クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「ULTableSchema クラス」 954 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

DeleteAllRows メソッド

テーブルのすべてのローを削除します。

構文

Visual Basic

Public Sub **DeleteAllRows()**

C#

public void **DeleteAllRows();**

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 クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「Truncate メソッド」 952 ページ
- 「StopSynchronizationDelete メソッド」 587 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。

構文

Visual Basic

Public Sub **FindBegin()**

C#

public void **FindBegin();**

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 クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「FindFirst() メソッド」 941 ページ
- 「FindFirst(Int16) メソッド」 941 ページ
- 「FindLast() メソッド」 943 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「FindLast(Int16) メソッド」 943 ページ

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() 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。

構文

Visual Basic

Public Function **FindFirst()** As Boolean

C#

public bool **FindFirst();**

戻り値

成功した場合は true、失敗した場合は false。

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 クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「FindFirst メソッド」 941 ページ
- 「FindBegin メソッド」 940 ページ
- 「FindNext() メソッド」 944 ページ
- 「FindPrevious() メソッド」 946 ページ
- 「FindFirst(Int16) メソッド」 941 ページ
- 「IsEOF プロパティ」 682 ページ
- 「FindBegin メソッド」 940 ページ

FindFirst(Int16) 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Function FindFirst( _
  ByVal numColumns As Short _
) As Boolean
```

C#

```
public bool FindFirst(
  short numColumns
);
```

パラメータ

- **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。

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 クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「FindFirst メソッド」 941 ページ
- 「FindBegin メソッド」 940 ページ
- 「FindNext(Int16) メソッド」 945 ページ
- 「FindPrevious(Int16) メソッド」 947 ページ
- 「FindFirst() メソッド」 941 ページ
- 「IsEOF プロパティ」 682 ページ
- 「FindBegin メソッド」 940 ページ

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。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Function **FindLast()** As Boolean

C#

public bool **FindLast();**

戻り値

成功した場合は true、失敗した場合は false。

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 クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「FindLast メソッド」 942 ページ
- 「FindBegin メソッド」 940 ページ
- 「FindNext() メソッド」 944 ページ
- 「FindPrevious() メソッド」 946 ページ
- 「FindLast(Int16) メソッド」 943 ページ
- 「IsBOF プロパティ」 681 ページ
- 「FindBegin メソッド」 940 ページ

FindLast(Int16) 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。

構文

Visual Basic

```
Public Function FindLast( _
   ByVal numColumns As Short _
) As Boolean
```

C#

```
public bool FindLast(
  short numColumns
);
```

パラメータ

- **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。

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 クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「FindLast メソッド」 942 ページ
- 「FindBegin メソッド」 940 ページ
- 「FindNext(Int16) メソッド」 945 ページ
- 「FindPrevious(Int16) メソッド」 947 ページ
- 「FindLast() メソッド」 943 ページ
- 「IsBOF プロパティ」 681 ページ
- 「FindBegin メソッド」 940 ページ

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() 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。

構文

Visual Basic

```
Public Function FindNext() As Boolean
```


Copyright © 2007, iAnywhere Solutions, Inc.

C#

```
public bool FindNext();
```

戻り値

成功した場合は true、失敗した場合は false。

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 クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「FindNext メソッド」 944 ページ
- 「FindFirst() メソッド」 941 ページ
- 「FindNext(Int16) メソッド」 945 ページ
- 「IsEOF プロパティ」 682 ページ

FindNext(Int16) 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。

構文

Visual Basic

```
Public Function FindNext( _
   ByVal numColumns As Short _
) As Boolean
```

C#

```
public bool FindNext(
   short numColumns
);
```

パラメータ

- **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。

Copyright © 2007, iAnywhere Solutions, Inc.

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 クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「FindNext メソッド」 944 ページ
- 「FindFirst(Int16) メソッド」 941 ページ
- 「FindNext() メソッド」 944 ページ
- 「FindFirst() メソッド」 941 ページ
- 「IsEOF プロパティ」 682 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。

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。

構文

Visual Basic

Public Function **FindPrevious()** As Boolean

C#

public bool **FindPrevious();**

戻り値

成功した場合は true、失敗した場合は false。

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 クラス」 933 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULTable メンバ」 933 ページ
- 「FindPrevious メソッド」 946 ページ
- 「FindLast() メソッド」 943 ページ
- 「FindPrevious(Int16) メソッド」 947 ページ
- 「IsBOF プロパティ」 681 ページ

FindPrevious(Int16) 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。

構文

Visual Basic

```
Public Function FindPrevious( _
  ByVal numColumns As Short _
) As Boolean
```

C#

```
public bool FindPrevious(
  short numColumns
);
```

パラメータ

- **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。

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 クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「FindPrevious メソッド」 946 ページ
- 「FindLast() メソッド」 943 ページ
- 「FindLast(Int16) メソッド」 943 ページ
- 「FindPrevious() メソッド」 946 ページ
- 「IsBOF プロパティ」 681 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

挿入を行う前に ULTable.InsertBegin を呼び出してください。

構文

Visual Basic

Public Sub **Insert()**

C#

public void **Insert();**

参照

- 「ULTable クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「InsertBegin メソッド」 948 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。

構文

Visual Basic

Public Sub **InsertBegin()**

C#

public void **InsertBegin();**

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 クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「Insert メソッド」 948 ページ
- 「Insert メソッド」 948 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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() 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。

構文

Visual Basic

Public Function **LookupBackward()** As Boolean

C#

public bool **LookupBackward();**

戻り値

成功した場合は true、失敗した場合は false。

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 クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「LookupBackward メソッド」 949 ページ
- 「LookupBegin メソッド」 950 ページ
- 「LookupBackward(Int16) メソッド」 949 ページ
- 「IsBOF プロパティ」 681 ページ

LookupBackward(Int16) メソッド

テーブルを最初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。

構文

Visual Basic

Public Function **LookupBackward(_**

Copyright © 2007, iAnywhere Solutions, Inc.

```
  ByVal numColumns As Short _
) As Boolean
```

C#

```
public bool LookupBackward(
  short numColumns
);
```

パラメータ

- **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。

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 クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「LookupBackward メソッド」 949 ページ
- 「LookupBegin メソッド」 950 ページ
- 「LookupBackward(Int16) メソッド」 949 ページ
- 「IsBOF プロパティ」 681 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。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。

構文

Visual Basic

```
Public Sub LookupBegin()
```

C#

```
public void LookupBegin();
```


Copyright © 2007, iAnywhere Solutions, Inc.

参照

- 「ULTable クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「LookupForward() メソッド」 951 ページ
- 「LookupForward(Int16) メソッド」 952 ページ
- 「LookupBackward() メソッド」 949 ページ
- 「LookupBackward(Int16) メソッド」 949 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。

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。

構文

Visual Basic

Public Function **LookupForward()** As Boolean

C#

public bool **LookupForward();**

戻り値

成功した場合は true、失敗した場合は false。

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 クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「LookupForward メソッド」 951 ページ
- 「LookupBegin メソッド」 950 ページ
- 「LookupForward(Int16) メソッド」 952 ページ
- 「IsEOF プロパティ」 682 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

LookupForward(Int16) 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。

構文

Visual Basic

```
Public Function LookupForward( _
  ByVal numColumns As Short _
) As Boolean
```

C#

```
public bool LookupForward(
  short numColumns
);
```

パラメータ

- **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。

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 クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「LookupForward メソッド」 951 ページ
- 「LookupBegin メソッド」 950 ページ
- 「LookupForward() メソッド」 951 ページ
- 「IsEOF プロパティ」 682 ページ
- 「LookupBegin メソッド」 950 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

Public Sub **Truncate()**

C#

public void **Truncate();**

参照

- 「ULTable クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「DeleteAllRows メソッド」 940 ページ

UpdateBegin メソッド

テーブルの現在のローを更新する準備を行います。

構文

Visual Basic

Public Sub **UpdateBegin()**

C#

public void **UpdateBegin();**

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、ULResultSet.Update() が呼び出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。

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

- 「ULTable クラス」 933 ページ
- 「ULTable メンバ」 933 ページ
- 「Update メソッド」 837 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

ULTableSchema クラス

UL 拡張： Ultra Light テーブルのスキーマを示します。このクラスは継承できません。

構文

Visual Basic

```
Public NotInheritable Class ULTableSchema
  Inherits ULCursorSchema
```

C#

```
public sealed class ULTableSchema : ULCursorSchema
```

備考

このクラスにはコンストラクタがありません。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。

継承： ULCursorSchema

参照

- 「ULTableSchema メンバ」 954 ページ
- 「ULTableSchema クラス」 954 ページ
- 「Schema プロパティ」 939 ページ
- 「ULCursorSchema クラス」 636 ページ

ULTableSchema メンバ

パブリック・プロパティ

メンバ名	説明
ColumnCount プロパティ (ULCursorSchema から継承)	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 (ULCursorSchema から継承)	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。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。

パブリック・メソッド

メンバ名	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 (ULCursorSchema から継承)	指定されたカラムのカラム ID を返します。
GetColumnName メソッド (ULCursorSchema から継承)	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 (ULCursorSchema から継承)	カラムが数値カラム (SQL NUMERIC 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。
GetColumnSQLName メソッド (ULCursorSchema から継承)	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。
GetColumnScale メソッド (ULCursorSchema から継承)	カラムが数値カラム (SQL NUMERIC 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 (ULCursorSchema から継承)	カラムがサイズ指定されたカラム (SQL BINARY 型または CHAR 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。
GetColumnULDbType メソッド (ULCursorSchema から継承)	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 (ULCursorSchema から継承)	ULDataReader のカラムのスキーマが記述された System.Data.DataTable を返します。

Copyright © 2007, iAnywhere Solutions, Inc.

メンバ名	説明
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。
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。
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 入力可能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。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema クラス」 954 ページ
- 「Schema プロパティ」 939 ページ
- 「ULCursorSchema クラス」 636 ページ

IndexCount プロパティ

テーブルのインデックス数を返します。

構文

Visual Basic

Public Readonly Property **IndexCount** As Integer

C#

public int **IndexCount** { get;}

プロパティ値

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。

Copyright © 2007, iAnywhere Solutions, Inc.

備考

インデックス ID の範囲は、1 ー 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。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ

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。

構文

Visual Basic

Public Readonly Property **IsNeverSynchronized** As Boolean

C#

public bool **IsNeverSynchronized** { get;}

プロパティ値

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。

備考

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、「未同期」テーブルと呼ばれることもあります。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ

Name プロパティ

テーブルの名前を返します。

構文

Visual Basic

Public Overrides Readonly Property **Name** As String

Copyright © 2007, iAnywhere Solutions, Inc.

C#

public override string **Name** { get;}

プロパティ値

文字列として返されるテーブル名。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ

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。

構文

Visual Basic

Public Readonly Property **PrimaryKey** As ULIndexSchema

C#

public ULIndexSchema **PrimaryKey** { get;}

プロパティ値

テーブルのプライマリ・キーを示す ULIndexSchema オブジェクト。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ
- 「ULIndexSchema クラス」 751 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。

構文

Visual Basic

Public Readonly Property **UploadUnchangedRows** As Boolean

C#

public bool **UploadUnchangedRows** { get;}

Copyright © 2007, iAnywhere Solutions, Inc.

プロパティ値

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。

備考

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。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 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。

構文

Visual Basic

```
Public Function GetColumnDefaultValue( _
  ByVal columnID As Integer _
) As String
```

C#

```
public string  GetColumnDefaultValue(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

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)。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ
- 「ColumnCount プロパティ」 637 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Public Function GetColumnPartitionSize( _
  ByVal columnID As Integer _
) As UInt64
```

C#

```
public ulong GetColumnPartitionSize(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

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)。

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 クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ
- 「IsColumnGlobalAutoIncrement メソッド」 965 ページ
- 「ColumnCount プロパティ」 637 ページ
- 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。

構文

Visual Basic

```
Public Function GetIndex( _
  ByVal name As String _
) As ULIndexSchema
```

C#

Copyright © 2007, iAnywhere Solutions, Inc.

```
public ULIndexSchema GetIndex(
  string  name
);
```

パラメータ

- **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示す ULIndexSchema オブジェクト。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ
- 「ULIndexSchema クラス」 751 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。

構文

Visual Basic

```
Public Function GetIndexName( _
  ByVal indexID As Integer _
) As String
```

C#

```
public string  GetIndexName(
  int indexID
);
```

パラメータ

- **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。

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 クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ
- 「IndexCount プロパティ」 956 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

```
Public Function GetOptimalIndex( _
  ByVal columnID As Integer _
) As ULIndexSchema
```

C#

```
public ULIndexSchema GetOptimalIndex(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

戻り値

指定されたカラムの最適なインデックスを示す ULIndexSchema オブジェクト。

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

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ
- 「ColumnCount プロパティ」 637 ページ
- 「ULIndexSchema クラス」 751 ページ

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。

構文

Visual Basic

```
Public Function GetPublicationPredicate( _
  ByVal pubName As String _
) As String
```

C#

```
public string  GetPublicationPredicate(
```


Copyright © 2007, iAnywhere Solutions, Inc.

```
  string  pubName
);
```

パラメータ

- **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。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ

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。

構文

Visual Basic

```
Public Function IsColumnAutoIncrement( _
  ByVal columnID As Integer _
) As Boolean
```

C#

```
public bool IsColumnAutoIncrement(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

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。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ
- 「ColumnCount プロパティ」 637 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Function IsColumnCurrentDate( _
  ByVal columnID As Integer _
) As Boolean
```

C#

```
public bool IsColumnCurrentDate(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

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。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ
- 「ColumnCount プロパティ」 637 ページ
- 「ULDbType 列挙」 715 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。

構文

Visual Basic

```
Public Function IsColumnCurrentTime( _
  ByVal columnID As Integer _
) As Boolean
```

C#

```
public bool IsColumnCurrentTime(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ
- 「ColumnCount プロパティ」 637 ページ
- 「ULDbType 列挙」 715 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。

構文

Visual Basic

```
Public Function IsColumnCurrentTimestamp( _
  ByVal columnID As Integer _
) As Boolean
```

C#

```
public bool IsColumnCurrentTimestamp(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

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。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ
- 「ColumnCount プロパティ」 637 ページ
- 「ULDbType 列挙」 715 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。

Copyright © 2007, iAnywhere Solutions, Inc.

構文

Visual Basic

```
Public Function IsColumnGlobalAutoIncrement( _
  ByVal columnID As Integer _
) As Boolean
```

C#

```
public bool IsColumnGlobalAutoIncrement(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

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。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ
- 「GetColumnPartitionSize メソッド」 960 ページ
- 「DatabaseID プロパティ」 563 ページ
- 「ColumnCount プロパティ」 637 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。

構文

Visual Basic

```
Public Function IsColumnNewUUID( _
  ByVal columnID As Integer _
) As Boolean
```

C#

```
public bool IsColumnNewUUID(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

Copyright © 2007, iAnywhere Solutions, Inc.

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。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ
- 「ColumnCount プロパティ」 637 ページ
- Guid

IsColumnNullable メソッド

指定されたカラムが null 入力可能であるかどうかをチェックします。

構文

Visual Basic

```
Public Function IsColumnNullable( _
  ByVal columnID As Integer _
) As Boolean
```

C#

```
public bool IsColumnNullable(
  int columnID
);
```

パラメータ

- **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。テーブルの最初のカラムの ID 値はゼロです。

戻り値

カラムが null 入力可能な場合は true、null 入力不可能な場合は false。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ
- 「ColumnCount プロパティ」 637 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。

構文

Visual Basic

```
Public Function IsInPublication( _
```

Copyright © 2007, iAnywhere Solutions, Inc.

```
  ByVal pubName As String _
) As Boolean
```

C#

```
public bool IsInPublication(
  string  pubName
);
```

パラメータ

- **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。

参照

- 「ULTableSchema クラス」 954 ページ
- 「ULTableSchema メンバ」 954 ページ

 Copyright © 2007, iAnywhere Solutions, Inc.

ULTransaction クラス

SQL トランザクションを示します。このクラスは継承できません。

構文

Visual Basic

Public NotInheritable Class **ULTransaction**
  Inherits DbTransaction

C#

public sealed class **ULTransaction** : DbTransaction

備考

ULTransaction にはコンストラクタがありません。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。

継承： System.Data.Common.DbTransaction

実装： System.Data.IDbTransaction、System.IDisposable

参照

- 「ULTransaction メンバ」 969 ページ
- 「BeginTransaction() メソッド」 568 ページ
- 「Transaction プロパティ」 521 ページ
- DbTransaction
- IDbTransaction
- IDisposable

ULTransaction メンバ

パブリック・プロパティ

メンバ名	説明
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。

Copyright © 2007, iAnywhere Solutions, Inc.

パブリック・メソッド

メンバ名	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 (DbTransaction から継承)	
Rollback メソッド	トランザクションの未処理の変更をデータベースにロールバックします。

参照

- 「ULTransaction クラス」 969 ページ
- 「BeginTransaction() メソッド」 568 ページ
- 「Transaction プロパティ」 521 ページ
- DbTransaction
- IDbTransaction
- IDisposable

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。

構文

Visual Basic

Public Readonly Property **Connection** As ULConnection

C#

public ULConnection **Connection** { get;}

プロパティ値

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)。

備考

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

- 「ULTransaction クラス」 969 ページ
- 「ULTransaction メンバ」 969 ページ
- 「BeginTransaction() メソッド」 568 ページ
- 「ULConnection クラス」 555 ページ
- IDbTransaction.Connection
- DbCommand.Connection

Copyright © 2007, iAnywhere Solutions, Inc.

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。

構文

Visual Basic

Public Overrides Readonly Property **IsolationLevel** As IsolationLevel

C#

public override IsolationLevel **IsolationLevel** { get;}

プロパティ値

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

- 「ULTransaction クラス」 969 ページ
- 「ULTransaction メンバ」 969 ページ
- 「BeginTransaction() メソッド」 568 ページ
- IsolationLevel
- IsolationLevel.ReadUncommitted

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。

構文

Visual Basic

Public Overrides Sub **Commit()**

C#

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 クラス」 969 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

- 「ULTransaction メンバ」 969 ページ
- 「Rollback メソッド」 972 ページ

Rollback メソッド

トランザクションの未処理の変更をデータベースにロールバックします。

構文

Visual Basic

Public Overrides Sub **Rollback()**

C#

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 クラス」 969 ページ
- 「ULTransaction メンバ」 969 ページ
- 「Commit メソッド」 971 ページ

Copyright © 2007, iAnywhere Solutions, Inc.

索引

A

Abort プロパティ [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 843
AcceptChangesDuringFill プロパティ [UL .NET 1.0]
  iAnywhere.Data.UltraLite ネームスペース, 161
ActiveSyncInvoked メソッド [UL .NET 1.0]
  iAnywhere.Data.UltraLite ネームスペース, 53
ActiveSyncInvoked メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 475
ActiveSync 同期
  Ultra Light.NET, 33
Add(Int32, Int32) メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 501
Add(Int32, String) メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 502
AdditionalParms プロパティ [UL .NET 1.0]
  iAnywhere.Data.UltraLite ネームスペース, 126
AdditionalParms プロパティ [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 595
Add(Object) メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 796
AddRange(Array) メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 802
AddRange(ULParameter[]) メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 802
AddRange メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 802
Add(String, Int32) メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 503
Add(String, Object) メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 797
Add(String, String) メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 504
Add(String, ULDbType, Int32, String) メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 801
Add(String, ULDbType, Int32) メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 799
Add(String, ULDbType) メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 798
Add(ULBulkCopyColumnMapping) メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 501
Add(ULParameter) メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 796
Add メソッド [UL .NET 1.0]
  iAnywhere.Data.UltraLite ネームスペース, 306, 307, 308, 310, 311, 312
Add メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 501, 795
AppendBytes メソッド [UL .NET 1.0]
  iAnywhere.Data.UltraLite ネームスペース, 330
AppendBytes メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 819
AppendChars メソッド [UL .NET 1.0]
  iAnywhere.Data.UltraLite ネームスペース, 332
AppendChars メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 821
AuthenticationParms プロパティ [UL .NET 1.0]
  iAnywhere.Data.UltraLite ネームスペース, 255, 398
AuthenticationParms プロパティ [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 732, 898
AuthStatus プロパティ [UL .NET 1.0]
  iAnywhere.Data.UltraLite ネームスペース, 256, 426
AuthStatus プロパティ [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 731, 928
AuthValue プロパティ [UL .NET 1.0]
  iAnywhere.Data.UltraLite ネームスペース, 256, 426
AuthValue プロパティ [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 732, 928
AutoCommit モード
  Ultra Light.NET 開発, 28

B

BatchSize プロパティ [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 484
BeginExecuteNonQuery() メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 522
BeginExecuteNonQuery(AsyncCallback, Object) メソッド [UL .NET 2.0]
  iAnywhere.Data.UltraLite ネームスペース, 523
BeginExecuteNonQuery メソッド [UL .NET 2.0]

Copyright © 2007, iAnywhere Solutions, Inc.

iAnywhere.Data.UltraLite ネームスペース, 522
BeginExecuteReader() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 524
BeginExecuteReader(AsyncCallback, Object, CommandBehavior) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 526
BeginExecuteReader(AsyncCallback, Object) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 525
BeginExecuteReader(CommandBehavior) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 524
BeginExecuteReader メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 523
BeginTransaction() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 568
BeginTransaction(IsolationLevel) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 569
BeginTransaction メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 103, 104
BeginTransaction メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 568
BulkCopyTimeout プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 485
BytesReceived プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 269
BytesReceived プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 746

C

CacheSize プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 130
CacheSize プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 599, 611
Cancel メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 70
Cancel メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 527
CanCreateDataSourceEnumerator プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 725
CaseSensitive プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 139
CaseSensitive プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 627
ChangeDatabase メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 105
ChangeDatabase メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 570
ChangeEncryptionKey メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 105
ChangeEncryptionKey メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 570
ChangePassword メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 571
CheckpointStore プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 398
CheckpointStore プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 899
ChecksumLevel プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 139
ChecksumLevel プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 627
Clear メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 313
Clear メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 803
Close メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 106, 210
Close メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 487, 572, 686
CollationName プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 191
CollationName プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 666
ColumnCount プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 148, 274
ColumnCount プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 637, 752
ColumnMappings プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 486
Columns プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 765
CommandText プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 63
CommandText プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 516

 Copyright © 2007, iAnywhere Solutions, Inc.

CommandTimeout プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 64
CommandTimeout プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 517
CommandType プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 65
CommandType プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 517
Command プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 355, 360
Command プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 848, 853
Commit メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 470
Commit メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 971
ConnectionName プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 131
ConnectionName プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 599, 612
ConnectionString プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 95
ConnectionString プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 560
ConnectionTimeout プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 97
ConnectionTimeout プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 561
Connection クラス
Ultra Light.NET , 15
Connection プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 66, 469
Connection プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 518, 970
ContainsKey メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 620
Contains(Object) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 803
Contains(String) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 804
Contains メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 313, 314
Contains メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 504, 803
ContinueUpdateOnError プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 162
CopyFrom メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 409
CopyFrom メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 910
CopyTo メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 314
CopyTo メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 505, 805
CountUploadRows(Int32, Int64) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 573
CountUploadRows(Int32, UInt32) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 572
CountUploadRows メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 107
CountUploadRows メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 572
Count プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 304
Count プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 792
CreateCommandBuilder メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 726
CreateCommand メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 108
CreateCommand メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 574, 726
CreateConnectionStringBuilder メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 727
CreateConnection メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 726
CreateDataAdapter メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 727
CreateDatabase メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 183
CreateDatabase メソッド [UL .NET 2.0]

Copyright © 2007, iAnywhere Solutions, Inc.

iAnywhere.Data.UltraLite ネームスペース, 659
CreateParameter メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 70
CreateParameter メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 528, 728

D

DataAdapter プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 84
DataAdapter プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 549
DatabaseID プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 98
DatabaseID プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 563
DatabaseKey プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 612
DatabaseManager クラス
Ultra Light.NET , 15
DatabaseManager プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 98
DatabaseManager プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 563
DatabaseName プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 613
DatabaseOnCE プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 131
DatabaseOnCE プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 600, 614
DatabaseOnDesktop プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 132
DatabaseOnDesktop プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 600, 614
Database プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 97
Database プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 562
DataSourceInformation プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 766
DataSource プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 562
DataTypes プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 766
DateFormat プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 140
DateFormat プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 628
DateOrder プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 140
DateOrder プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 628
DbType プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 294
DbType プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 781
DeleteAllRows メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 438
DeleteAllRows メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 940
DeleteCommand プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 162
DeleteCommand プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 650
Delete メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 333
Delete メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 822
Depth プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 204
Depth プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 680
DesignTimeVisible プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 66
DesignTimeVisible プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 519
DestinationColumn プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 496
DestinationFileName プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 257
DestinationFileName プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 733
DestinationOrdinal プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 497
DestinationPath プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 257
DestinationPath プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 733
DestinationTableName プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 486
Direction プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 295

 Copyright © 2007, iAnywhere Solutions, Inc.

Direction プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 782
DisableConcurrency プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 399
DisableConcurrency プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 899
Dispose メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 211, 471
Dispose メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 488
DML
Ultra Light.NET , 18
DownloadedFile プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 258
DownloadedFile プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 734
DownloadFile() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 741
DownloadFile(ULFileTransferProgressListener) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 742
DownloadFile メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 265, 266
DownloadFile メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 741
DownloadOnly プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 399
DownloadOnly プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 900
DropDatabase メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 185
DropDatabase メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 660

E

EncryptionKey プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 132
EncryptionKey プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 601
EndExecuteNonQuery メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 528
EndExecuteReader メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 532
EquivalentTo メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 620
ErrorMessage プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 412
ErrorMessage プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 914
ExecuteNonQuery メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 71
ExecuteNonQuery メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 535
ExecuteReader() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 535
ExecuteReader(CommandBehavior) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 537
ExecuteReader メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 72, 73
ExecuteReader メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 535
ExecuteResultSet() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 538
ExecuteResultSet(CommandBehavior) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 539
ExecuteResultSet メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 75, 76
ExecuteResultSet メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 538
ExecuteScalar メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 77
ExecuteScalar メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 540
ExecuteTable() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 541
ExecuteTable(CommandBehavior) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 542
ExecuteTable(String, String, CommandBehavior) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 577
ExecuteTable(String, String) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 576
ExecuteTable(String) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 575
ExecuteTable メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 78, 79, 109, 110, 112

Copyright © 2007, iAnywhere Solutions, Inc.

ExecuteTable メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 541, 574
Explorer、Ultra Light 用 (参照 SQL Anywhere Explorer、Ultra Light 用)

F

FieldCount プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 205
FieldCount プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 680
FileAuthCode プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 258
FileAuthCode プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 735
FileName プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 259
FileName プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 735
FileSize プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 269
FileSize プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 746
FileTransferProgressed メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 271
FileTransferProgressed メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 749
FillError イベント [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 180
FillSchema メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 172, 173, 175
Fill メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 168, 170, 171
Finalize メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 149, 211, 334, 430, 438
FindBegin メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 439
FindBegin メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 940
FindFirst() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 941
FindFirst(Int16) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 941
FindFirst メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 439, 440
FindFirst メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 941
FindLast() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 943
FindLast(Int16) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 943
FindLast メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 441, 442
FindLast メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 942
FindNext() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 944
FindNext(Int16) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 945
FindNext メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 443, 444
FindNext メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 944
FindPrevious() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 946
FindPrevious(Int16) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 947
FindPrevious メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 445
FindPrevious メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 946
FIPS プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 141
FIPS プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 628
FLAG_IS_BLOCKING フィールド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 268, 412
FLAG_IS_BLOCKING フィールド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 746, 913
Flags プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 269, 413
Flags プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 747, 914
ForceDownload プロパティ [UL .NET 1.0]

 Copyright © 2007, iAnywhere Solutions, Inc.

iAnywhere.Data.UltraLite ネームスペース, 259
ForceDownload プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 736
ForeignKeys プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 767

G

GetBoolean メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 212
GetBoolean メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 686
GetBytes(Int32, Int64, Byte[], Int32, Int32) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 688
GetBytes(Int32) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 689
GetBytes メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 213, 215
GetBytes メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 688
GetByte メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 213
GetByte メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 687
GetChars メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 217
GetChars メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 691
GetChar メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 216
GetChar メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 690
GetColumnDefaultValue メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 458
GetColumnDefaultValue メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 959
GetColumnID メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 149
GetColumnID メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 638
GetColumnName メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 150, 279
GetColumnName メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 639, 757
GetColumnPartitionSize メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 458
GetColumnPartitionSize メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 960
GetColumnPrecision メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 151
GetColumnPrecision メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 640
GetColumnScale メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 152
GetColumnScale メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 642
GetColumnSize メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 153
GetColumnSize メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 642
GetColumnSQLName メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 153
GetColumnSQLName メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 641
GetColumnULDbType メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 154
GetColumnULDbType メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 643
GetDatabaseProperty メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 194
GetDatabaseProperty メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 669
GetDataTypeName メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 219
GetDataTypeName メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 692
GetData メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 218
GetDateTime メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 220
GetDateTime メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 693
GetDecimal メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 221
GetDecimal メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 693
GetDeleteCommand() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 550
GetDeleteCommand(Boolean) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 549

Copyright © 2007, iAnywhere Solutions, Inc.

GetDeleteCommand メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 86
GetDeleteCommand メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 549
GetDouble メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 221
GetDouble メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 694
GetEnumerator メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 315
GetEnumerator メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 695, 805
GetFieldType メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 222
GetFieldType メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 695
GetFillParameters メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 175
GetFillParameters メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 654
GetFloat メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 223
GetFloat メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 696
GetGuid メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 224
GetGuid メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 697
GetIndexName メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 460
GetIndexName メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 961
GetIndex メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 459
GetIndex メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 960
GetInsertCommand() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 552
GetInsertCommand(Boolean) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 551
GetInsertCommand メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 87
GetInsertCommand メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 551
GetInt16 メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 225
GetInt16 メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 697
GetInt32 メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 226
GetInt32 メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 698
GetInt64 メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 227
GetInt64 メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 699
GetLastDownloadTime メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 113
GetLastDownloadTime メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 579
GetName メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 227
GetName メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 699
GetNewUUID メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 114
GetNewUUID メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 580
GetObjectData メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 252
GetObjectData メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 722
GetOptimalIndex メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 461
GetOptimalIndex メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 962
GetOrdinal メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 228
GetOrdinal メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 700
GetPublicationName メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 196
GetPublicationName メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 671
GetPublicationPredicate メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 461
GetPublicationPredicate メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 962
GetPublicationSchema メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 197
GetPublicationSchema メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 672
GetSchema() メソッド [UL .NET 2.0]

 Copyright © 2007, iAnywhere Solutions, Inc.

iAnywhere.Data.UltraLite ネームスペース, 580
GetSchema(String, String) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 581
GetSchema(String) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 581
GetSchemaTable メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 155, 229
GetSchemaTable メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 644, 701
GetSchema メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 580
GetShortName メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 621
GetString メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 232
GetString メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 703
GetTableCountInPublications メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 197
GetTableCountInPublications メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 672
GetTableName メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 198
GetTableName メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 673
GetTimeSpan メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 232
GetTimeSpan メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 704
GetUInt16 メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 233
GetUInt16 メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 705
GetUInt32 メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 234
GetUInt32 メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 705
GetUInt64 メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 235
GetUInt64 メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 706
GetUpdateCommand() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 554
GetUpdateCommand(Boolean) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 553
GetUpdateCommand メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 88
GetUpdateCommand メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 553
GetValues メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 236
GetValues メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 707
GetValue メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 235
GetValue メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 707
GlobalAutoIncrementUsage プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 99
GlobalAutoIncrementUsage プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 564
grantConnectTo メソッド
Ultra Light.NET 開発, 31
GrantConnectTo メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 115
GrantConnectTo メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 583

H

HasRows プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 681

I

iAnywhere.Data.UltraLite ネームスペース
説明, 2
iAnywhere.Data.UltraLite ネームスペース (.NET 1.0)
iAnywhere.Data.UltraLite ネームスペース, 2
iAnywhere.Data.UltraLite ネームスペース [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 52
iAnywhere.Data.UltraLite ネームスペース [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 474
iAnywhere デベロッパー・コミュニティ ニュースグループ, xvii
IgnoredRows プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 427
IgnoredRows プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 929

Copyright © 2007, iAnywhere Solutions, Inc.

IndexColumns プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 768
IndexCount プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 455
IndexCount プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 956
Indexes プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 768
IndexName プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 67
IndexName プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 519
IndexOf(Object) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 806
IndexOf(String) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 806
IndexOf メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 316
IndexOf メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 506, 806
IndexSchema クラス
Ultra Light.NET 開発, 29
InfoMessage イベント [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 121
InfoMessage イベント [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 588
InsertBegin メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 447
InsertBegin メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 948
InsertCommand プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 163
InsertCommand プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 651
Insert メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 317, 446
Insert メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 807, 948
install-dir
マニュアルの使用方法, xiv
Instance フィールド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 725
Interactive SQL
Visual Studio .NET から開く, 9
INVALID_DATABASE_ID フィールド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 95
INVALID_DATABASE_ID フィールド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 560
IsBOF プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 206
IsBOF プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 681
IsCaseSensitive プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 191
IsCaseSensitive プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 666
IsClosed プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 206
IsClosed プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 682
IsColumnAutoIncrement メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 462
IsColumnAutoIncrement メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 963
IsColumnCurrentDate メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 463
IsColumnCurrentDate メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 963
IsColumnCurrentTimestamp メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 464
IsColumnCurrentTimestamp メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 965
IsColumnCurrentTime メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 463
IsColumnCurrentTime メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 964
IsColumnDescending メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 280
IsColumnDescending メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 758
IsColumnGlobalAutoIncrement メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 465
IsColumnGlobalAutoIncrement メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 965
IsColumnNewUUID メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 466
IsColumnNewUUID メソッド [UL .NET 2.0]

 Copyright © 2007, iAnywhere Solutions, Inc.

iAnywhere.Data.UltraLite ネームスペース, 966
IsColumnNullable メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 466
IsColumnNullable メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 967
IsDBNull メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 237
IsDBNull メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 708
IsEOF プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 206
IsEOF プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 682
IsFixedSize プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 792
IsForeignKeyCheckOnCommit プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 275
IsForeignKeyCheckOnCommit プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 753
IsForeignKeyNullable プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 276
IsForeignKeyNullable プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 754
IsForeignKey プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 274
IsForeignKey プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 753
IsInPublication メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 467
IsInPublication メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 967
IsNeverSynchronized プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 456
IsNeverSynchronized プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 957
IsNullable プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 296
IsNullable プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 782
IsolationLevel プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 469
IsolationLevel プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 971
IsOpen プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 148, 192, 276, 323
IsOpen プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 638, 667, 754, 812
IsPrimaryKey プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 276
IsPrimaryKey プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 755
IsReadOnly プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 792
IsSynchronized プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 793
IsUniqueIndex プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 277
IsUniqueIndex プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 755
IsUniqueKey プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 277
IsUniqueKey プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 756
Item(Int32) プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 683, 793
Item(String) プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 683, 794
Item プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 207, 208, 305, 306
Item プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 500, 615, 682, 793

K

KeepPartialDownload プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 400
KeepPartialDownload プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 901

L

LastIdentity プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 100
LastIdentity プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 564
LookupBackward() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 949

LookupBackward(Int16) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 949
LookupBackward メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 447, 448
LookupBackward メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 949
LookupBegin メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 449
LookupBegin メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 950
LookupForward() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 951
LookupForward(Int16) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 952
LookupForward メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 450
LookupForward メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 951

M

Mask プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 323
Mask プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 812
MaxHashSize プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 141
MaxHashSize プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 629
Message プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 282
Message プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 760
MetaDataCollections プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 769
MissingMappingAction プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 164
MissingSchemaAction プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 165
MoveAfterLast メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 238
MoveAfterLast メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 709
MoveBeforeFirst メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 239
MoveBeforeFirst メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 709
moveFirst クラス
Ultra Light.NET データ取得の例, 20
moveFirst メソッド (Table クラス)
Ultra Light.NET 開発, 22
MoveFirst メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 239
MoveFirst メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 710
MoveLast メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 239
MoveLast メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 710
moveNext クラス
Ultra Light.NET データ取得の例, 20
moveNext メソッド (Table クラス)
Ultra Light.NET 開発, 22
MoveNext メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 240
MoveNext メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 711
MovePrevious メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 241
MovePrevious メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 711
MoveRelative メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 241
MoveRelative メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 712

N

Name プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 149, 278, 323, 352, 456
Name プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 638, 756, 813, 840, 957
NativeError プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 251, 283
NativeError プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 721, 761
NearestCentury プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 142
NearestCentury プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 630
NewPassword プロパティ [UL .NET 1.0]

 Copyright © 2007, iAnywhere Solutions, Inc.

iAnywhere.Data.UltraLite ネームスペース, 401
NewPassword プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 902
NextResult メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 242
NextResult メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 712
NotifyAfter プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 487

O

Obfuscate プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 142
Obfuscate プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 630
Offset プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 296
Offset プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 783
Open メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 116
Open メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 584
OrderedTableScans プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 616

P

PageSize プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 143
PageSize プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 631
ParameterName プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 297
ParameterName プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 783
Parameters プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 67
Parameters プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 520
PartialDownloadRetained プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 427
PartialDownloadRetained プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 929
Password プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 133, 260, 402
Password プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 602, 617, 736, 903
PDF
マニュアル, x
PingOnly プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 403
PingOnly プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 903
Plan プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 68
Plan プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 520
Precision プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 143, 297
Precision プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 631, 784
Prepare メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 80
Prepare メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 543
PrimaryKey プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 456
PrimaryKey プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 958
PublicationCount プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 193
PublicationCount プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 668
PublicationMask プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 403
PublicationMask プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 904
PublicationSchema クラス
Ultra Light.NET 開発, 29
Publications プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 770

Q

QuotePrefix プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 85
QuoteSuffix プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 85

R

Read メソッド [UL .NET 1.0]

iAnywhere.Data.UltraLite ネームスペース, 242
Read メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 713
ReceivedBytes プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 413
ReceivedBytes プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 914
ReceivedDeletes プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 414
ReceivedDeletes プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 915
ReceivedInserts プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 414
ReceivedInserts プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 915
ReceivedUpdates プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 415
ReceivedUpdates プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 916
RecordsAffected プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 209, 355
RecordsAffected プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 684, 849
ReferencedIndexName プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 278
ReferencedIndexName プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 757
ReferencedTableName プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 279
ReferencedTableName プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 757
RefreshSchema メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 89
RemoveAt(Int32) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 808
RemoveAt(String) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 809
RemoveAt メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 319
RemoveAt メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 507, 808
Remove メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 318
Remove メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 506, 621, 808
ReservedWords プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 770
ReserveSize プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 617
ResetDbType メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 788
ResetLastDownloadTime メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 116
ResetLastDownloadTime メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 585
Restrictions プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 771
ResumedAtSize プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 270
ResumedAtSize プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 747
ResumePartialDownload プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 261, 404
ResumePartialDownload プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 737, 904
RevokeConnectFrom メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 117
RevokeConnectFrom メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 585
revokeConnectionFrom メソッド
Ultra Light.NET 開発, 31
RollbackPartialDownload メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 117
RollbackPartialDownload メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 586
Rollback メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 471
Rollback メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 972
RowCount プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 209
RowCount プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 685
RowsCopied プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 843
RowUpdated イベント [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 180
RowUpdated イベント [UL .NET 2.0]

 Copyright © 2007, iAnywhere Solutions, Inc.

iAnywhere.Data.UltraLite ネームスペース, 655
RowUpdating イベント [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 181
RowUpdating イベント [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 656
RuntimeType プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 183
RuntimeType プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 658

S

samples-dir
マニュアルの使用方法, xiv
Scale プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 144, 298
Scale プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 631, 784
Schema プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 100, 210, 437
Schema プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 565, 685, 939
SelectCommand プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 165
SelectCommand プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 652
SELECT 文
Ultra Light.NET の例, 20
SendColumnNames プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 405
SendColumnNames プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 905
SendDownloadAck プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 405
SendDownloadAck プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 906
SentBytes プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 415
SentBytes プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 917
SentDeletes プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 415
SentDeletes プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 917
SentInserts プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 416
SentInserts プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 918
SentUpdates プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 416
SentUpdates プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 918
ServerSyncInvoked メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 363
ServerSyncInvoked メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 856
ServerVersion プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 566
SetActiveSyncListener メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 186
SetActiveSyncListener メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 661
SetBoolean メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 334
SetBoolean メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 823
SetBytes メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 336
SetBytes メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 824
SetByte メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 335
SetByte メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 823
SetDatabaseOption メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 199
SetDatabaseOption メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 674
SetDateTime メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 337
SetDateTime メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 826
SetDBNull メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 338
SetDBNull メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 825
SetDecimal メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 338
SetDecimal メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 827
SetDouble メソッド [UL .NET 1.0]

Copyright © 2007, iAnywhere Solutions, Inc.

iAnywhere.Data.UltraLite ネームスペース, 339
SetDouble メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 828
SetFloat メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 340
SetFloat メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 828
SetGuid メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 341
SetGuid メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 829
SetInt16 メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 342
SetInt16 メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 830
SetInt32 メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 343
SetInt32 メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 831
SetInt64 メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 344
SetInt64 メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 832
SetServerSyncListener メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 187
SetServerSyncListener メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 662
SetString メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 345
SetString メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 833
SetTimeSpan メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 346
SetTimeSpan メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 833
SetToDefault メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 347
SetToDefault メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 834
SetUInt16 メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 347
SetUInt16 メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 835
SetUInt32 メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 348
SetUInt32 メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 836
SetUInt64 メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 349
SetUInt64 メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 837
SignalSyncIsComplete メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 188
SignalSyncIsComplete メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 664
Size プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 299
Size プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 785
SourceColumnNullMapping プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 786
SourceColumn プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 299
SourceColumn プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 497, 786
SourceOrdinal プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 498
SourceVersion プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 300
SourceVersion プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 786
Source プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 251, 283
Source プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 721, 761
SQL Anywhere
マニュアル, x
SQL Anywhere Explorer
Ultra Light 制限事項, 14
Ultra Light でサポートされるプログラミング言語, 9
Ultra Litht.NET 説明, 14
SQL Anywhere Explorer の設定
Ultra Light 説明, 10
SQL Anywhere Explorer を使用した Ultra Light テーブルの操作
説明, 12
SQLCode プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 417
SQLCode プロパティ [UL .NET 2.0]

 Copyright © 2007, iAnywhere Solutions, Inc.

iAnywhere.Data.UltraLite ネームスペース, 916
SQL 結果セットのナビゲーション
Ultra Light.NET, 20
StartLine プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 618
StartSynchronizationDelete メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 118
StartSynchronizationDelete メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 586
StateChange イベント [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 122
StateChange イベント [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 590
State プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 101, 417
State プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 566, 919
StopSynchronizationDelete メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 119
StopSynchronizationDelete メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 587
StreamErrorCode プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 262, 428
StreamErrorCode プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 738, 930
StreamErrorContext プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 428
StreamErrorContext プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 930
StreamErrorID プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 429
StreamErrorID プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 931
StreamErrorSystem プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 262, 429
StreamErrorSystem プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 739, 931
StreamParms プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 263, 406
StreamParms プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 739, 907
Stream プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 261, 406
Stream プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 738, 907
Sybase Central
Visual Studio .NET から開く, 9
SYNC_ALL_DB フィールド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 322
SYNC_ALL_DB フィールド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 811
SYNC_ALL_PUBS フィールド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 322
SYNC_ALL_PUBS フィールド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 812
Synchronize() メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 587
Synchronize(ULSyncProgressListener) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 588
Synchronize メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 119, 120
Synchronize メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 587
SyncParms プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 101, 418
SyncParms プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 567, 919
SyncProgressed メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 421
SyncProgressed メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 923
SyncResult プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 102, 418
SyncResult プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 567, 920
SyncRoot プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 795

Copyright © 2007, iAnywhere Solutions, Inc.

T

TableCount プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 193, 419
TableCount プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 668, 920
TableIndex プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 419
TableIndex プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 921
TableMappings プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 166
TableMappings プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 653
TableOrder プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 407
TableOrder プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 908
TableSchema クラス
Ultra Light.NET 開発, 29
Tables プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 771
TableTotal プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 420
TableTotal プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 921
TimeFormat プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 144
TimeFormat プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 632
TimestampFormat プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 145
TimestampFormat プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 632
TimestampIncrement プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 145
TimestampIncrement プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 633
Timestamp プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 429
Timestamp プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 932
ToString メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 134, 146, 284, 302
ToString メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 603, 634, 762, 789
Transaction プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 68
Transaction プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 521
Truncate メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 451
Truncate メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 952
TryGetValue メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 622

U

ULActiveSyncListener インタフェース [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 53
ULActiveSyncListener インタフェース [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 475
ULActiveSyncListener メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 475
ULAuthStatusCode 列挙 [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 56
ULAuthStatusCode 列挙 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 479
ULBulkCopyColumnMapping() コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 493
ULBulkCopyColumnMappingCollection クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 499
ULBulkCopyColumnMappingCollection メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 499
ULBulkCopyColumnMapping(Int32, Int32) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 493
ULBulkCopyColumnMapping(Int32, String) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 494
ULBulkCopyColumnMapping(String, Int32) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 495
ULBulkCopyColumnMapping(String, String) コンストラクタ [UL .NET 2.0]

Copyright © 2007, iAnywhere Solutions, Inc.

iAnywhere.Data.UltraLite ネームスペース, 495
ULBulkCopyColumnMapping クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 492
ULBulkCopyColumnMapping コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 493
ULBulkCopyColumnMapping メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 492
ULBulkCopyOptions 列挙 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 508
ULBulkCopy(String, ULBulkCopyOptions) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 483
ULBulkCopy(String) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 482
ULBulkCopy(ULConnection, ULBulkCopyOptions, ULTransaction) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 483
ULBulkCopy(ULConnection) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 481
ULBulkCopy クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 480
ULBulkCopy コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 481
ULBulkCopy メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 480
ULCommand() コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 512
ULCommandBuilder() コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 548
ULCommandBuilder(ULDataAdapter) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 548
ULCommandBuilder クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 82
ULCommandBuilder クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 545
ULCommandBuilder コンストラクタ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 83, 84
ULCommandBuilder コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 548
ULCommandBuilder メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 546
ULCommand(String, ULConnection, ULTransaction) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 515
ULCommand(String, ULConnection) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 514
ULCommand(String) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 513
ULCommand クラス
Ultra Light.NET データ取得の例, 20
Ultra Light.NET データ操作の例, 18
ULCommand クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 57
ULCommand クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 509
ULCommand コンストラクタ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 60, 61, 62
ULCommand コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 512
ULCommand メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 510
ULConnection() コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 558
ULConnectionParms.UnusedEventHandler 委任 [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 136
ULConnectionParms.UnusedEventHandler 委任 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 605
ULConnectionParms クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 123
ULConnectionParms クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 592
ULConnectionParms コンストラクタ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 125
ULConnectionParms コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 595
ULConnectionParms メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 594
ULConnectionStringBuilder() コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 610
ULConnectionStringBuilder(String) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 610
ULConnectionStringBuilder クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 606

Copyright © 2007, iAnywhere Solutions, Inc.

ULConnectionStringBuilder コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 610
ULConnectionStringBuilder メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 608
ULConnection(String) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 559
ULConnection クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 90
ULConnection クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 555
ULConnection コンストラクタ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 93, 94
ULConnection コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 558
ULConnection メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 555
ULCreateParms クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 137
ULCreateParms クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 624
ULCreateParms コンストラクタ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 138
ULCreateParms コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 626
ULCreateParms メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 625
ULCursorSchema クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 147
ULCursorSchema クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 636
ULCursorSchema メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 636
ULDataAdapter() コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 648
ULDataAdapter(String, String) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 649
ULDataAdapter(String, ULConnection) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 649
ULDataAdapter(ULCommand) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 648
ULDataAdapter クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 156
ULDataAdapter クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 645
ULDataAdapter コンストラクタ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 159, 160, 161
ULDataAdapter コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 647
ULDataAdapter メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 645
ULDatabaseManager クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 182
ULDatabaseManager クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 657
ULDatabaseManager メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 658
ULDatabaseSchema クラス
iAnywhere.Data.UltraLite.NET ネームスペース, 29
ULDatabaseSchema クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 190
ULDatabaseSchema クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 665
ULDatabaseSchema メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 665
ULDataReader クラス
Ultra Light.NET データ取得の例, 20
ULDataReader クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 201
ULDataReader クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 676
ULDataReader メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 677
ULDateOrder 列挙 [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 244
ULDateOrder 列挙 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 714
ULDbType プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 301
ULDbType プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 787
ULDbType 列挙 [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 245
ULDbType 列挙 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 715
ULException クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 249
ULException クラス [UL .NET 2.0]

 Copyright © 2007, iAnywhere Solutions, Inc.

iAnywhere.Data.UltraLite ネームスペース, 719
ULException コンストラクタ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 250
ULException コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 720
ULException メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 719
ULFactory クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 723
ULFactory メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 724
ULFileTransferProgressData クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 268
ULFileTransferProgressData クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 745
ULFileTransferProgressData メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 745
ULFileTransferProgressListener インタフェース [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 271
ULFileTransferProgressListener インタフェース [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 749
ULFileTransferProgressListener メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 749
ULFileTransfer クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 253
ULFileTransfer クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 729
ULFileTransfer コンストラクタ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 255
ULFileTransfer コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 731
ULFileTransfer メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 729
ULIndexSchema クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 273
ULIndexSchema クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 751
ULIndexSchema メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 751
ULInfoMessageEventArgs クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 282
ULInfoMessageEventArgs クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 760
ULInfoMessageEventArgs メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 760
ULInfoMessageEventHandler 委任 [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 285
ULInfoMessageEventHandler 委任 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 763
ULMetaDataCollectionNames クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 764
ULMetaDataCollectionNames メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 764
ULParameter() コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 775
ULParameterCollection クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 303
ULParameterCollection クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 790
ULParameterCollection メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 790
ULParameter(String, Object) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 776
ULParameter(String, ULDbType, Int32, ParameterDirection, Boolean, Byte, Byte, String, DataRowVersion, Object) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.SQLAnywhere ネームスペース, 780
ULParameter(String, ULDbType, Int32, String) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 779
ULParameter(String, ULDbType, Int32) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 778
ULParameter(String, ULDbType) コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 777
ULParameter クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 286
ULParameter クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 773
ULParameter コンストラクタ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 288, 289, 290, 291, 292
ULParameter コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 775
ULParameter メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 774
ULPublicationSchema クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 321

Copyright © 2007, iAnywhere Solutions, Inc.

ULPublicationSchema クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 810
ULPublicationSchema メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 811
ULResultSetSchema クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 351
ULResultSetSchema クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 839
ULResultSetSchema メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 839
ULResultSet クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 325
ULResultSet クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 814
ULResultSet メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 815
ULRowsCopiedEventArgs クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 842
ULRowsCopiedEventArgs コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 842
ULRowsCopiedEventArgs メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 842
ULRowsCopiedEventHandler 委任 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 845
ULRowsCopied イベント [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 491
ULRowUpdatedEventArgs クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 353
ULRowUpdatedEventArgs クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 846
ULRowUpdatedEventArgs コンストラクタ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 354
ULRowUpdatedEventArgs コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 847
ULRowUpdatedEventArgs メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 846
ULRowUpdatedEventHandler 委任 [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 357
ULRowUpdatedEventHandler 委任 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 850
ULRowUpdatingEventArgs クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 358
ULRowUpdatingEventArgs クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 851
ULRowUpdatingEventArgs コンストラクタ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 359
ULRowUpdatingEventArgs コンストラクタ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 852
ULRowUpdatingEventArgs メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 851
ULRowUpdatingEventHandler 委任 [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 361
ULRowUpdatingEventHandler 委任 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 854
ULRuntimeType 列挙 [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 362
ULRuntimeType 列挙 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 855
ULServerSyncListener インタフェース [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 363
ULServerSyncListener インタフェース [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 856
ULServerSyncListener メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 856
ULSQLCode 列挙 [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 366
ULSQLCode 列挙 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 859
ULStreamErrorCode 列挙 [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 376
ULStreamErrorCode 列挙 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 871
ULStreamErrorContext 列挙 [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 392
ULStreamErrorContext 列挙 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 892
ULStreamErrorID 列挙 [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 393
ULStreamErrorID 列挙 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 893
ULStreamType 列挙 [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 395
ULStreamType 列挙 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 895
ULSyncParms クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 396
ULSyncParms クラス [UL .NET 2.0]

 Copyright © 2007, iAnywhere Solutions, Inc.

iAnywhere.Data.UltraLite ネームスペース, 896
ULSyncParms メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 897
ULSyncProgressData クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 411
ULSyncProgressData クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 912
ULSyncProgressData メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 912
ULSyncProgressListener インタフェース [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 421
ULSyncProgressListener インタフェース [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 923
ULSyncProgressListener メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 923
ULSyncProgressState 列挙 [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 423
ULSyncProgressState 列挙 [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 925
ULSyncResult クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 425
ULSyncResult クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 927
ULSyncResult メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 927
ULTableSchema クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 453
ULTableSchema クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 954
ULTableSchema メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 954
ULTable クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 431
ULTable クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 933
ULTable メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 933
Ultra Light
Visual Studio との統合, 36
Ultra Light.NET
ActiveSync 同期, 33
SQL を使用したデータ操作, 18
Ultra Light 用 SQL Anywhere Explorer, 7
アプリケーションでの同期, 32
暗号化, 17
アーキテクチャ, 5
エラー処理, 30
開発, 13
サポートされるプラットフォーム, 4
スキーマ情報のアクセス, 29
せつめい, 1
チュートリアル, 35
データ操作, 18
データの取得, 20
テーブル API の概要, 22
同期の例, 32
動的 SQL チュートリアル, 43
トランザクション処理, 28
配置, 48
ユーザ認証, 31
利点, 2
Ultra Light.NET 開発の概要
説明, 13
Ultra Light データベース
Ultra Light.NET 情報へのアクセス, 29
Ultra Light.NET 接続, 15
Ultra Light モード
Ultra Light.NET, 23
Ultra Light モードの使用
Ultra Light.NET, 23
Ultra Light 用 SQL Anywhere Explorer
Visual Studio との統合, 9
接続, 9
設定, 10
説明, 7
データベース・オブジェクトの追加, 11
テーブルの操作, 12
Ultra Light 用 SQL Anywhere Explorer の使用
説明, 9
Ultra Light 用 SQL Anywhere Explorer を使用したデータベース・オブジェクトの追加
説明, 11
ULTransaction クラス [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 468
ULTransaction クラス [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 969
ULTransaction メンバ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 969
UnusedEvent イベント [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 135
UnusedEvent イベント [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 603

UpdateBegin メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 452
UpdateBegin メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 953
UpdateCommand プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 167
UpdateCommand プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 653
UpdatedRowSource プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 69
UpdatedRowSource プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 522
Update メソッド [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 176, 177, 178, 179, 350
Update メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 837
UploadOK プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 430
UploadOK プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 932
UploadOnly プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 408
UploadOnly プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 909
UploadUnchangedRows プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 457
UploadUnchangedRows プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 958
UserID プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 133
UserID プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 602, 619
UserName プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 263, 408
UserName プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 740, 909
UTF8Encoding プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 145
UTF8Encoding プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 633

V

Value プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 301
Value プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 788
Version プロパティ [UL .NET 1.0]
iAnywhere.Data.UltraLite ネームスペース, 264, 409
Version プロパティ [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 740, 910
Visual Studio
SQL Anywhere Explorer と Ultra Light との統合, 9
Ultra Light との統合, 36
Visual Studio .NET
Ultra Light.NET Ultra Light データベースへの接続, 14
Ultra Light データベース接続, 9
Ultra Light データベースへのアクセス, 9

W

Windows CE
Ultra Light.NET ターゲット・プラットフォーム, 4
WriteToServer(DataRow[]) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 488
WriteToServer(DataTable, DataRowState) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 490
WriteToServer(DataTable) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 489
WriteToServer(IDataReader) メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 489
WriteToServer メソッド [UL .NET 2.0]
iAnywhere.Data.UltraLite ネームスペース, 488

あ

アイコン
マニュアルで使用, xv
値
Ultra Light.NET アクセス, 23
暗号化
Ultra Light.NET 開発, 17
アーキテクチャ
Ultra Light.NET, 5

い

インストール

 Copyright © 2007, iAnywhere Solutions, Inc.

Ultra Light 用 SQL Anywhere Explorer, 9
インデックス
Ultra Light.NET スキーマ情報, 29

え

エラー
Ultra Light.NET 処理, 30
エラー処理
Ultra Light.NET, 30

お

[オプション] ダイアログ
Ultra Light 用 SQL Anywhere Explorer, 10
オフセット
Ultra Light.NET での相対オフセット, 22
オンライン・マニュアル
PDF, x

か

開発
Ultra Light.NET, 13
開発プラットフォーム
Ultra Light.NET, 4
カラム
Ultra Light.NET 値の変更, 24
管理
Ultra Light.NET トランザクション, 28

き

規則
表記, xii
マニュアルでのファイル名, xiv
キャスト
Ultra Light.NET データ型のキャスト, 24

け

結果セット
Ultra Light.NET ナビゲーション, 20
結果セット・スキーム
Ultra Light.NET , 21
検索
Ultra Light.NET を使用して Ultra Light ローを検索, 24
検索メソッド
Ultra Light.NET , 24
検索モード
Ultra Light.NET , 23

こ

更新
Ultra Light.NET テーブル・ロー, 25
更新モード
Ultra Light.NET , 23
コピー
Visual Studio .NET での Ultra Light データベース・オブジェクト, 11
コミット
Ultra Light.NET トランザクション, 28
コミット・メソッド
Ultra Light.NET トランザクション, 28

さ

削除
Ultra Light.NET を使用して Ultra Light テーブル・ローを削除, 27
サポート
ニュースグループ, xvii
サポートされるプラットフォーム
Ultra Light.NET, 4

し

詳細情報の検索／フィードバックの提供
テクニカル・サポート, xvii

す

スキーマ
Ultra Light.NET アクセス, 29
スキーマ情報のアクセス
Ultra Light.NET 説明, 29
スクロール
Ultra Light.NET テーブル API, 22
スレッド
Ultra Light.NET マルチスレッド・アプリケーション, 15

せ

接続
Ultra Light.NET チュートリアル, 41
Ultra Light.NET データベース, 15
Ultra Light 用 SQL Anywhere Explorer, 9
選択

Copyright © 2007, iAnywhere Solutions, Inc.

Ultra Light.NET ロー, 20

そ

相対オフセット
Ultra Light.NET テーブル API, 22
挿入
Ultra Light.NET テーブル・ロー, 26
挿入モード
Ultra Light.NET , 23

た

ターゲット・プラットフォーム
Ultra Light.NET, 4

ち

チュートリアル
Ultra Light.NET の C#, 35
Ultra Light.NET の Visual Basic, 35

て

テクニカル・サポート
ニュースグループ, xvii
デベロッパー・コミュニティ
ニュースグループ, xvii
データ・アクセス
Ultra Light.NET テーブル API, 22
データ型
Ultra Light.NET API アクセス, 23
Ultra Light.NET 変換, 24
データ操作
SQL を使用した Ultra Light.NET の操作, 18
Ultra Light.NET テーブル API, 22
データベース・スキーマ
Ultra Light.NET アクセス, 29
データベース・テーブルからのデータ選択
Ultra Light.NET, 20
テーブル
Ultra Light.NET スキーマ情報, 29
テーブル API
Ultra Light.NET の概要, 22

と

同期
Ultra Light.NET, 32
Ultra Light.NET C# の例, 32
Ultra Light.NET の ActiveSync, 33
動的 SQL
Ultra Light.NET チュートリアル, 43
トラブルシューティング
Ultra Light.NET 開発チェックリスト, 48
Ultra Light.NET でのエラー処理, 30
ニュースグループ, xvii
トランザクション
Ultra Light.NET 管理, 28
トランザクション処理
Ultra Light.NET 管理, 28

な

ナビゲーション
Ultra Light.NET テーブル API, 22
難読化
Ultra Light.NET 開発, 17

に

ニュースグループ
テクニカル・サポート, xvii

は

配置
Ultra Light.NET, 48
バグ
フィードバックの提供, xvii
パスワード
Ultra Light.NET 認証, 31
パブリケーション
Ultra Light.NET スキーマ情報, 29

ひ

表記
規則, xii

ふ

フィードバック
提供, xvii
マニュアル, xvii
プラットフォーム
Ultra Light.NET サポート, 4

へ

ヘルプ
テクニカル・サポート, xvii
ヘルプへのアクセス

 Copyright © 2007, iAnywhere Solutions, Inc.

テクニカル・サポート, xvii

ま

マニュアル
SQL Anywhere, x
マルチスレッド・アプリケーション
Ultra Light.NET の概要, 15

も

モード
Ultra Light.NET , 23

ゆ

ユーザ認証
Ultra Light.NET, 31
ユーザの認証
Ultra Light.NET , 31

り

利点
Ultra Light.NET, 2

る

ルックアップ・メソッド
Ultra Light.NET , 24
ルックアップ・モード
Ultra Light.NET , 23

ろ

ロック
Ultra Light.NET キーワード, 33
ロー
Ultra Light.NET テーブル・アクセス、現在の ロー, 23
Ultra Light.NET テーブル・ナビゲーション, 22
Ultra Light.NET を使用して Ultra Light ローを更新, 25
Ultra Light.NET を使用して Ultra Light ローを削除, 27
Ultra Light.NET を使用して Ultra Light ローを挿入, 26
ロールバック
Ultra Light.NET トランザクション, 28
ロールバック・メソッド
Ultra Light.NET トランザクション, 28

Copyright © 2007, iAnywhere Solutions, Inc.