SQL Anywhere - Ultra Light
文書バージョン: 17 – 2016-05-11

Ultra Light - WinRT API リファレンス

目次

1 Ultra Light for WinRT API リファレンス

Ultra Light for WinRT には C#、C++、または JavaScript から使用する豊富な API オブジェクトがあります。

次に、よく使用される API オブジェクトの一部を示します。

DatabaseManager

データベースと接続を管理する方法を提供します。

接続

Ultra Light データベースへの接続を表します。Connection オブジェクトは 1 つまたは複数作成できます。

PreparedStatement、ResultSet

動的 SQL 文の作成、クエリの記述、INSERT、UPDATE、DELETE 文の実行、プログラムによるデータベースの結果セットの制御を行います。

ネームスペース

```
UltraLite
```

> **注記**
>
> 主な SQL Anywhere マニュアルをお探しですか。マニュアルをローカルにインストールした場合は、Windows のスタートメニューを使用してアクセスするか (Microsoft Windows)、`C:¥Program Files¥SQL Anywhere 17¥Documentation` にナビゲートします。
>
> また、DocCommentXchange の Web で、主な SQL Anywhere API リファレンスマニュアルにアクセスすることもできます。http://dcx.sap.com

このセクションの内容:

Ultra Light データベースのスキーマを表します。Connection.GetDatabaseSchema を使用して DatabaseSchema オブジェクトを取得します。

FileTransfer クラス [121 ページ]
Mobile Link サーバを通じて、リモートデータベースからファイルを転送します。

IndexSchema クラス [125 ページ]
Ultra Light テーブルのインデックスのスキーマを表します。TableSchem.GetIndexSchema(indexName) を使用して IndexSchema オブジェクトを取得します。

PreparedStatement クラス [131 ページ]
準備された SQL 文を表します。Connection.PrepareStatement は PreparedStatement を返します。

ResultSet クラス [146 ページ]
Ultra Light データベースの結果セットを表します。ResultSet オブジェクトは更新および削除に使用され、PreparedStatement.ExecuteQuery() によって返されます。

ResultSetSchema クラス [211 ページ]
Ultra Light の結果セットのスキーマを表します。ResultSet.GetResultSetSchema または PreparedStatement.GetResultSetSchema は ResultSetSchema を返します。

Table クラス [217 ページ]
Ultra Light データベース内のテーブルを表します。Connection.OpenTable は Table を返します。

TableSchema クラス [290 ページ]
Ultra Light のテーブルのスキーマを表します。DatabaseSchema.GetTables()、GetTableSchema() または Table.GetTableSchema() は TableSchema を返します。

FileTransferObserver(FileTransferStatus^) デリゲート [304 ページ]
ファイル転送のさまざまな段階で呼び出されるデリゲートの定義。

SyncObserver(SyncStatus^) デリゲート [305 ページ]
同期のさまざまな段階で呼び出されるデリゲートの定義。

ValidateCallback(ValidateData^) デリゲート [305 ページ]
検証フィードバックに対して呼び出されるデリゲートの定義。

AuthStatusCode 列挙体 [305 ページ]
Mobile Link 認証ステータスコードを指定します。SyncResult および FileTransferResult には AuthStatusCode 値を返す AuthStatus 項目があります。

ColumnDefaultType 列挙体 [306 ページ]
カラムのデフォルト型を識別します。

ColumnNameType 列挙体 [307 ページ]
結果セットの記述時にカラムの名前を取得する方法を制御する値を指定します。

ColumnSQLType 列挙体 [308 ページ]
カラムの SQL タイプを指定します。CursorSchema.GetColumnSQLType() は ColumnSQLType を返します。

ColumnStorageType 列挙体 [309 ページ]
カラムのホスト変数タイプを指定します。

CursorState 列挙体 [310 ページ]
可能な結果セットまたはカーソルステータスを指定します。Cursor.GetState() は CursorState を返します。

ErrorCodes 列挙体 [311 ページ]

Ultra Light 例外を示す Platform::COMException HResult コードを指定します。

IndexFlag 列挙体 [312 ページ]
インデックスのプロパティを識別するフラグ (ビットフィールド) を指定します。

StreamType 列挙体 [312 ページ]
FileTransfer.Stream プロパティに使用可能な値を指定します。

SyncState 列挙体 [313 ページ]
同期の現在の処理を示します。

TableSyncType 列挙体 [314 ページ]
テーブルの同期タイプを識別します。

ValidateFlags 列挙体 [315 ページ]
検証入力フラグを表します。Connection の ValidateDatabase メソッドおよび DatabaseManager は ValidateFlags をフラグパラメータとして使用します。

ValidateStatusId 列挙体 [316 ページ]
検証進捗状況コールバックの可能なステータス ID を指定します。

FileTransferResult 構造 [318 ページ]
アプリケーションで適切なアクションを実行できるようにするために、ファイル転送結果を格納します。

FileTransferStatus 構造 [318 ページ]
ファイルのアップロードまたはダウンロードの進行中の、ステータスまたは進行状況情報を格納します。

SyncResult 構造 [319 ページ]
アプリケーションで適切なアクションを実行できるようにするために、同期の結果を格納します。Connection.GetSyncResult() は SyncResult を返します。

SyncStatus 構造 [321 ページ]
同期の進行中のステータスまたは進行状況情報を格納します。

ValidateData 構造 [323 ページ]
検証の進行中の検証ステータス情報を格納します。

1.1 Connection クラス

Ultra Light データベースへの接続を表します。

ネームスペース

```
UltraLite
```

構文

```
public ref class  sealed
```

メンバー

Connection のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public unsigned int	CancelGetNotification(String^) [18 ページ]	指定された名前に一致する、すべてのキューに登録されている保留中の取得通知コールをキャンセルします。
public void	ChangeEncryptionKey(String^) [18 ページ]	Ultra Light データベースのデータベース暗号化キーを変更します。
public void	Checkpoint() [19 ページ]	チェックポイント操作を実行し、保留中になっているコミット済みトランザクションをデータベースにフラッシュします。
public void	Close() [19 ページ]	この接続と、残りの関連オブジェクトを破棄します。
public void	Commit() [19 ページ]	現在のトランザクションをコミットします。
public unsigned int	CountUploadRows(String^, unsigned int) [20 ページ]	同期するためにアップロードする必要があるローの数を数えます。
public void	CreateNotificationQueue(String^) [21 ページ]	この接続のイベント通知キューを作成します。
public void	DeclareEvent(String^) [21 ページ]	登録およびトリガできるイベントを宣言します。
public void	DestroyNotificationQueue(String^) [22 ページ]	指定されたイベント通知キューを破棄します。
public void	ExecuteStatement(String^) [22 ページ]	SQL 文の文字列を直接実行します。
public String	GetDatabaseProperty(String^) [23 ページ]	データベースプロパティの値を取得します。
public unsigned int	GetDatabasePropertyInt(String^) [23 ページ]	データベースプロパティの整数値を取得します。
public DatabaseSchema	GetDatabaseSchema() [24 ページ]	データベースのスキーマを問い合わせるために使用されるオブジェクトを返します。
public DateTime	GetLastDownloadTime(String^) [24 ページ]	指定したパブリケーションが最後にダウンロードされた時刻を取得します。
public void	GetLastError(ULSqlCode *, String^*) [25 ページ]	最後の呼び出しに関連付けられているエラー情報をこの Connection に返します。
public uint64	GetLastIdentity() [25 ページ]	@@identity の値を取得します。
public String	GetNotification(String^, unsigned int) [26 ページ]	イベント通知を読み込みます。
public String	GetNotificationParameter(String^, String^) [27 ページ]	GetNotification メソッドによって読み込まれたイベント通知のパラメータを取得します。
public SyncResult	GetSyncResult() [27 ページ]	前回の同期結果を取得します。

変更子とタイプ	メソッド	説明
public uint16	GlobalAutoIncrementUsage() [28 ページ]	グローバルオートインクリメントのデフォルト値を持つすべてのカラムで、デフォルト値が使用されている比率 (%) を取得します。
public void	GrantConnectTo(String^, String^) [28 ページ]	指定されたパスワードを持つ新しいまたは既存のユーザ ID に、Ultra Light データベースへのアクセスを許可します。
public Table	OpenTable(String^, String^) [29 ページ]	テーブルを開きます。
public PreparedStatement	PrepareStatement(String^) [29 ページ]	SQL 文の準備を行います。
public void	RegisterForEvent(String^, String^, String^, bool) [30 ページ]	イベントの通知を受け取るためのキューを登録または登録解除します。
public void	ResetLastDownloadTime(String^) [31 ページ]	アプリケーションが以前にダウンロードされたデータを再同期するように、パブリケーションの最終ダウンロード時間をリセットします。
public void	RevokeConnectFrom(String^) [31 ページ]	Ultra Light データベースからユーザ ID のアクセス権を取り消します。
public void	Rollback() [32 ページ]	現在のトランザクションをロールバックします。
public void	RollbackPartialDownload() [32 ページ]	失敗した同期からの変更をロールバックします。
public unsigned int	SendNotification(String^, String^, String^) [32 ページ]	指定された名前と一致するすべてのキューに通知を送信します。
public void	SetDatabaseOption(String^, String^) [33 ページ]	指定されたデータベースオプションを設定します。
public void	SetDatabaseOptionInt(String^, unsigned int) [33 ページ]	データベースオプションを設定します。
public void	StartSynchronizationDelete() [34 ページ]	この接続の START SYNCHRONIZATION DELETE を設定します。
public void	StopSynchronizationDelete() [34 ページ]	この接続の STOP SYNCHRONIZATION DELETE を設定します。
public void	Synchronize(String^, SyncObserver^) [34 ページ]	Ultra Light アプリケーションで同期を開始します。
public IAsyncActionWithProgress< SyncStatus^>	SynchronizeAsync(String^) [35 ページ]	Ultra Light アプリケーションで同期を開始します。
public unsigned int	TriggerEvent(String^, String^) [35 ページ]	ユーザ定義のイベントをトリガして、登録されたすべてのキューに通知を送信します。
public void	ValidateDatabase(uint16, String^, ValidateCallback^) [36 ページ]	この接続でのデータベースを検証します。
public IAsyncActionWithProgress< ValidateData^>	ValidateDatabaseAsync(uint16flags, String^) [37 ページ]	ValidateDatabase 操作を非同期に実行します。

このセクションの内容:

グローバルオートインクリメントのデフォルト値を持つすべてのカラムで、デフォルト値が使用されている比率 (%) を取得します。

GrantConnectTo(String^, String^) メソッド [28 ページ]
指定されたパスワードを持つ新しいまたは既存のユーザ ID に、Ultra Light データベースへのアクセスを許可します。

OpenTable(String^, String^) メソッド [29 ページ]
テーブルを開きます。

PrepareStatement(String^) メソッド [29 ページ]
SQL 文の準備を行います。

RegisterForEvent(String^, String^, String^, bool) メソッド [30 ページ]
イベントの通知を受け取るためのキューを登録または登録解除します。

ResetLastDownloadTime(String^) メソッド [31 ページ]
アプリケーションが以前にダウンロードされたデータを再同期するように、パブリケーションの最終ダウンロード時間をリセットします。

RevokeConnectFrom(String^) メソッド [31 ページ]
Ultra Light データベースからユーザ ID のアクセス権を取り消します。

Rollback() メソッド [32 ページ]
現在のトランザクションをロールバックします。

RollbackPartialDownload() メソッド [32 ページ]
失敗した同期からの変更をロールバックします。

SendNotification(String^, String^, String^) メソッド [32 ページ]
指定された名前と一致するすべてのキューに通知を送信します。

SetDatabaseOption(String^, String^) メソッド [33 ページ]
指定されたデータベースオプションを設定します。

SetDatabaseOptionInt(String^, unsigned int) メソッド [33 ページ]
データベースオプションを設定します。

StartSynchronizationDelete() メソッド [34 ページ]
この接続の START SYNCHRONIZATION DELETE を設定します。

StopSynchronizationDelete() メソッド [34 ページ]
この接続の STOP SYNCHRONIZATION DELETE を設定します。

Synchronize(String^, SyncObserver^) メソッド [34 ページ]
Ultra Light アプリケーションで同期を開始します。

SynchronizeAsync(String^) メソッド [35 ページ]
Ultra Light アプリケーションで同期を開始します。

TriggerEvent(String^, String^) メソッド [35 ページ]
ユーザ定義のイベントをトリガして、登録されたすべてのキューに通知を送信します。

ValidateDatabase(uint16, String^, ValidateCallback^) メソッド [36 ページ]
この接続でのデータベースを検証します。

ValidateDatabaseAsync(uint16flags, String^) メソッド [37 ページ]
ValidateDatabase 操作を非同期に実行します。

1.1.1 CancelGetNotification(String^) メソッド

指定された名前に一致する、すべてのキューに登録されている保留中の取得通知コールをキャンセルします。

構文

```
public unsigned int CancelGetNotification (queueName)
```

パラメータ

queueName キューの名前。

戻り値

影響を受けるキューの数 (必ずしもブロックされた読み込み数ではありません)。

1.1.2 ChangeEncryptionKey(String^) メソッド

Ultra Light データベースのデータベース暗号化キーを変更します。

構文

```
public void ChangeEncryptionKey (newKey)
```

パラメータ

newKey データベースの新しい暗号化キー。

備考

このメソッドを呼び出すアプリケーションでは、データベースが同期されていること、または信頼できるバックアップコピーが作成されていることを、先に確認しておく必要があります。ChangeEncryptionKey メソッドは完了まで実行する必要のある操作であるため、信頼できるデータベースバックアップが必要です。データベース暗号化キーを変更すると、まずデータベースのすべてのローは古いキーを使用して復号され、次に新しいキーを使用して再度暗号化されて、書き込まれます。この操作は元に

戻せません。暗号化変更処理が完了しなかった場合、データベースは無効な状態のままになり、再度アクセスすることはできなくなります。

1.1.3 Checkpoint() メソッド

チェックポイント操作を実行し、保留中になっているコミット済みトランザクションをデータベースにフラッシュします。

構文

```
public void Checkpoint ()
```

備考

Checkpoint メソッドを呼び出しても、現在のトランザクションすべてがコミットされるわけではありません。このメソッドは、パフォーマンスを向上させるために後回しにされた自動トランザクションチェックポイントとともに (commit_flush 接続パラメータを使用して) 使用されます。

Checkpoint メソッドを使用すると、保留中のコミット済みトランザクションがすべてデータベースの記憶領域に書き込まれることが保証されます。

1.1.4 CloseObject() メソッド

この接続と、残りの関連オブジェクトを破棄します。

構文

```
public void CloseObject ()
```

1.1.5 Commit() メソッド

現在のトランザクションをコミットします。

構文

```
public void Commit ()
```

1.1.6 CountUploadRows(String^, unsigned int) メソッド

同期するためにアップロードする必要があるローの数を数えます。

構文

```
public unsigned int CountUploadRows (pubList, threshold)
```

パラメータ

pubList チェック対象となるパブリケーションのカンマ区切りのリストを含む文字列。空の文字列 (SYNC_ALL 定数) は、非同期とマーク付けされたものを除くすべてのテーブルを表します。アスタリスクのみの文字列 (SYNC_ALL_PUBS 定数) は、いずれかのパブリケーションで参照されているすべてのテーブルを表します。一部のテーブルは、どのパブリケーションの一部でもないため、この値が * の場合は含まれません。

threshold カウントするローの最大数を判断します。メソッドの所要時間を制限します。threshold が 0 の場合、制限はありません (つまり、同期する必要のあるすべてのローをカウントします)。また、threshold が 1 の場合、同期の必要なローがあるかどうかを簡単に判別するために使用できます。

戻り値

指定されたパブリケーションのセットまたはデータベース全体のいずれかで、同期を必要とするローの数。

備考

このメソッドを使用すると、ユーザは同期、または自動バックグラウンド同期が行われるタイミングの決定を求められます。

次の呼び出しでは、データベース全体をチェックして、同期させるローの総数を確認します。

```
count = conn.CountUploadRows( Constants.SYNC_ALL, 0 );
```

次の呼び出しでは、最大 1000 のローに対してパブリケーション PUB1 と PUB2 がチェックされます。

```
count = conn.CountUploadRows( "PUB1,PUB2", 1000 );
```

次の呼び出しでは、パブリケーション PUB1 と PUB2 で同期させる必要のあるローがあるかどうかがチェックされます。

```
anyToSync = conn.CountUploadRows( "PUB1,PUB2", 1 ) != 0;
```

1.1.7 CreateNotificationQueue(String^) メソッド

この接続のイベント通知キューを作成します。

構文

```
public void CreateNotificationQueue (name)
```

パラメータ

name 新しいキューの名前。

備考

キュー名は、接続ごとにスコープされるため、別々の接続で同じ名前を持つキューを作成できます。イベント通知が送信されると、データベース内で一致する名前を持つすべてのキューが、個別のインスタンスの通知を受け取ります。名前では、大文字と小文字が区別されません。RegisterForEvent メソッドを呼び出したときに、キューが指定されていない場合は、接続ごとにデフォルトのキューが作成されます。その名前がすでに存在する場合や有効でない場合は、エラーが発生して呼び出しが失敗します。

1.1.8 DeclareEvent(String^) メソッド

登録およびトリガできるイベントを宣言します。

構文

```
public void DeclareEvent (eventName)
```

パラメータ

eventName 新しいユーザ定義イベントの名前。

備考

Ultra Light では、データベースまたは環境での操作によってトリガされるシステムイベントが事前に定義されています。このメソッドは、ユーザ定義イベントを宣言します。ユーザ定義イベントは、TriggerEvent メソッドでトリガされます。イベント名は、ユニークにする必要があります。名前では、大文字と小文字が区別されません。

1.1.9 DestroyNotificationQueue(String^) メソッド

指定されたイベント通知キューを破棄します。

構文

```
public void DestroyNotificationQueue (name)
```

パラメータ

name 破棄するキューの名前。

備考

キュー内に未読の通知が残っている場合は、警告が通知されます。未読の通知は破棄されます。接続のデフォルトのイベントキューが作成されている場合、接続が閉じると破棄されます。

1.1.10 ExecuteStatement(String^) メソッド

SQL 文の文字列を直接実行します。

構文

```
public void ExecuteStatement (sql)
```

パラメータ

sql 実行する SQL スクリプト。

備考

このメソッドを使用して、SELECT 文を直接実行し、単一の結果を取り出します。

PrepareStatement メソッドを使用して、変数パラメータと共に文を繰り返し実行するか、複数の結果をフェッチします。

1.1.11 GetDatabaseProperty(String^) メソッド

データベースプロパティの値を取得します。

構文

```
public String GetDatabaseProperty (propName)
```

パラメータ

propName 要求されているプロパティの名前。

戻り値

正常に実行された場合は、データベースプロパティの値が格納された文字列。失敗した場合は NULL。

備考

戻り値は静的バッファを指します。静的バッファの内容は、それ以降の Ultra Light の呼び出しによって変更される可能性があるため、値を保存しておきたい場合はその値のコピーを作成してください。

```
String^ charset = conn.GetDatabaseProperty( "CharSet" );
```

1.1.12 GetDatabasePropertyInt(String^) メソッド

データベースプロパティの整数値を取得します。

構文

```
public unsigned int GetDatabasePropertyInt (propName)
```

パラメータ

propName 要求されているプロパティの名前。

戻り値

呼び出しが成功した場合はプロパティの整数値。失敗した場合は 0。

備考

```
unsigned connectionCount = conn.GetDatabasePropertyInt( "ConnCount" );
```

1.1.13 GetDatabaseSchema() メソッド

データベースのスキーマを問い合わせるために使用されるオブジェクトを返します。

構文

```
public DatabaseSchema GetDatabaseSchema ()
```

戻り値

データベースのスキーマを問い合わせるために使用される DatabaseSchema オブジェクト。

1.1.14 GetLastDownloadTime(String^) メソッド

指定したパブリケーションが最後にダウンロードされた時刻を取得します。

構文

```
public DateTime GetLastDownloadTime (publication)
```

パラメータ

publication パブリケーション名。

戻り値

前回のダウンロードの時刻。値 January 1, 1900 は、パブリケーションがまだ同期されていないか、時間がリセットされたことを示します。

備考

次の呼び出しでは、pub1 パブリケーションがダウンロードされた日付と時刻とともに dt 構造体が投入されます。

```
DECL_DATETIME dt;
ok = conn->GetLastDownloadTime( "pub1", &dt );
```

1.1.15 GetLastError(ULSqlCode *, String^*) メソッド

最後の呼び出しに関連付けられているエラー情報をこの Connection に返します。

構文

```
public void GetLastError (errorCode, errorParms)
```

備考

DatabaseManager::GetLastError を参照してください。

1.1.16 GetLastIdentity() メソッド

@@identity の値を取得します。

構文

```
public uint64 GetLastIdentity ()
```

戻り値

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に最後に挿入された値。

備考

この値は、データベースの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に最後に挿入された値です。データベースが停止している場合、この値は記録されません。このため、オートインクリメントの値が挿入される前にこのメソッドを呼び出すと、0 が返されます。

> **注記**
>
> 最後に挿入された値が別の接続の値である可能性があります。

1.1.17 GetNotification(String^, unsigned int) メソッド

イベント通知を読み込みます。

構文

```
public String GetNotification (queueName, waitms)
```

パラメータ

queueName 読み取るキュー。または、デフォルト接続キューの場合は NULL。
waitms 返す前に、待機 (ブロック) する時間 (ミリ秒単位)。

戻り値

読み込まれたイベントの名前。エラーが発生した場合は NULL。

備考

この呼び出しは、通知が受信されるまで、または指定された待機時間が経過するまで呼び出しスレッドをブロックします。

無期限に待機するには、waitms パラメータを UL_READ_WAIT_INFINITE に設定します。

待機をキャンセルするには、指定したキューに別の通知を送信するか、CancelGetNotification メソッドを使用します。

通知を読み込んだ後に GetNotificationParameter メソッドを使用して、追加のパラメータを名前で取得します。

1.1.18 GetNotificationParameter(String^, String^) メソッド

GetNotification メソッドによって読み込まれたイベント通知のパラメータを取得します。

構文

```
public String GetNotificationParameter (queueName, parameterName)
```

パラメータ

queueName 読み取るキュー。デフォルト接続キューの場合は NULL。
parameterName 読み込むパラメータの名前 (または *)。

戻り値

パラメータ値。エラーが発生した場合は NULL。

備考

指定されたキューで最近読み込まれた通知のパラメータのみが使用可能です。パラメータは名前によって取得されます。パラメータ名を * と指定すると、パラメータ文字列全体が取得されます。

1.1.19 GetSyncResult() メソッド

前回の同期結果を取得します。

構文

```
public SyncResult GetSyncResult ()
```

1.1.20 GlobalAutoIncrementUsage() メソッド

グローバルオートインクリメントのデフォルト値を持つすべてのカラムで、デフォルト値が使用されている比率 (%) を取得します。

構文

```
public uint16 GlobalAutoIncrementUsage ()
```

戻り値

グローバルオートインクリメントの値のカウンタによる使用済み比率 (%)。

備考

このデフォルト値を使用するカラムがデータベース内に複数含まれている場合は、すべてのカラムに対してこの値が計算され、最大値が返されます。たとえば、戻り値 99 は、少なくとも 1 つのカラムではデフォルト値が残されているが、少ないことを示します。

1.1.21 GrantConnectTo(String^, String^) メソッド

指定されたパスワードを持つ新しいまたは既存のユーザ ID に、Ultra Light データベースへのアクセスを許可します。

構文

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

パラメータ

uid ユーザ ID を表す String。最大長は 31 文字です。
pwd ユーザ ID のパスワードを表す String。

備考

このメソッドは、既存のユーザ ID を指定したときに、既存のユーザのパスワードを更新します。

1.1.22 OpenTable(String^, String^) メソッド

テーブルを開きます。

構文

```
public Table OpenTable (tableName, indexName)
```

パラメータ

tableName 開くテーブルの名前。

indexName テーブルを開く場合に使用するインデックスの名前。プライマリキーを使用してテーブルを開く場合は NULL、順序付けなしでテーブルを開く場合は空の文字列を渡します。

戻り値

呼び出しが成功した場合は Table オブジェクト。失敗した場合は NULL。

備考

アプリケーションがテーブルを初めて開いたときは、カーソルの位置が最初のローの前に設定されます。

1.1.23 PrepareStatement(String^) メソッド

SQL 文の準備を行います。

構文

```
public PreparedStatement PrepareStatement (sql)
```

パラメータ

sql 準備する SQL 文。

戻り値

成功した場合は PreparedStatement オブジェクト、それ以外の場合は NULL。

1.1.24 RegisterForEvent(String^, String^, String^, bool) メソッド

イベントの通知を受け取るためのキューを登録または登録解除します。

構文

```
public void RegisterForEvent (eventName, objectName, queueName,
register_not_unreg)
```

パラメータ

eventName 登録するシステム定義またはユーザ定義のイベント。
objectName イベントを適用するオブジェクト (テーブル名など)。
queueName NULL は、デフォルトの接続キューの使用を表します。
register_not_unreg 登録する場合は true、登録解除する場合は false を設定します。

備考

キュー名が指定されていない場合は、デフォルトの接続キューが暗黙で指定され、必要に応じて作成されます。特定のシステムイベントでは、そのイベントが適用されるオブジェクト名を指定できます。たとえば、TableModified イベントではテーブル名を指定できます。SendNotification メソッドとは異なり、登録された特定のキューのみイベントの通知を受信します。別の接続に、同じ名前の他のキューがある場合、それらは、同様に明示的に登録されていないかぎり、通知を受信しません。

事前に定義されたシステムイベントは次のとおりです。

- TableModified - テーブル内のローが挿入、更新、または削除されたときにトリガされます。要求の影響を受けるローの数にかかわらず、要求ごとに 1 つの通知が送信されます。object_name パラメータは、モニタするテーブルを指定します。値 "*" は、データベース内のすべてのテーブルを意味します。このイベントには、table_name というパラメータがあり、このパラメータの値は変更されたテーブルの名前です。
- Commit - コミットが完了した後にトリガされます。このイベントにはパラメータはありません。
- SyncComplete - 同期が完了した後にトリガされます。このイベントにはパラメータはありません。

1.1.25 ResetLastDownloadTime(String^) メソッド

アプリケーションが以前にダウンロードされたデータを再同期するように、パブリケーションの最終ダウンロード時間をリセットします。

構文

```
public void ResetLastDownloadTime (pubList)
```

パラメータ

pubList リセットするパブリケーションのカンマ区切りのリストを含む文字列。空の文字列は、非同期とマーク付けされたものを除くすべてのテーブルが含まれることを意味します。アスタリスクのみの文字列 (*) は、すべてのパブリケーションを表します。一部のテーブルは、どのパブリケーションの一部でもないため、この値が * の場合は含まれません。

備考

次のメソッド呼び出しは、すべてのテーブルの最終ダウンロード時間をリセットします。

```
conn.ResetLastDownloadTime( "" );
```

1.1.26 RevokeConnectFrom(String^) メソッド

Ultra Light データベースからユーザ ID のアクセス権を取り消します。

構文

```
public void RevokeConnectFrom (uid)
```

パラメータ

uid データベースアクセスから除外されるユーザ ID。

1.1.27 Rollback() メソッド

現在のトランザクションをロールバックします。

構文

```
public void Rollback ()
```

1.1.28 RollbackPartialDownload() メソッド

失敗した同期からの変更をロールバックします。

構文

```
public void RollbackPartialDownload ()
```

備考

再開可能なダウンロードを使用 (Keep Partial Download オプションを有効にして同期) しているときに、同期のダウンロードフェーズ中に通信エラーが発生すると、Ultra Light は、ダウンロードされた変更を保持するため、同期は、中断した時点から再開可能です)。ダウンロードを再開する必要がない場合は、このメソッドを使用して、この部分的なダウンロードを破棄します。

このメソッドは、再開可能なダウンロードの使用時にのみ効果があります。

1.1.29 SendNotification(String^, String^, String^) メソッド

指定された名前と一致するすべてのキューに通知を送信します。

構文

```
public unsigned int SendNotification (queueName, eventName, parameters)
```

パラメータ

queueName 対象となるキューの名前 (または *)。
eventName 通知の ID。

parameters パラメータのオプションリスト。

戻り値

送信済みの通知の数 (一致するキューの数)。

備考

これに含まれるのは、現在の接続における一致するキューです。この呼び出しはブロックしません。キューの特殊名 "*" を使用すると、すべてのキューに通知が送信されます。指定されたイベント名は、システム定義またはユーザ定義のイベントと対応する必要はありません。読み込まれたイベント名は、通知を識別するために渡され、送信者と受信者に対してしか意味を持たないからです。

parameters の値には、名前=値のペアをセミコロンで区切ったリストを指定します。通知が読み込まれた後、パラメータの値が GetNotificationParameter メソッドによって読み込まれます。

1.1.30 SetDatabaseOption(String^, String^) メソッド

指定されたデータベースオプションを設定します。

構文

```
public void SetDatabaseOption (optName, value)
```

パラメータ

optName 設定されるオプションの名前。
value オプションの新しい値。

1.1.31 SetDatabaseOptionInt(String^, unsigned int) メソッド

データベースオプションを設定します。

構文

```
public void SetDatabaseOptionInt (optName, value)
```

パラメータ

optName 設定されるオプションの名前。
value オプションの新しい値。

1.1.32 StartSynchronizationDelete() メソッド

この接続の START SYNCHRONIZATION DELETE を設定します。

構文

```
public void StartSynchronizationDelete ()
```

1.1.33 StopSynchronizationDelete() メソッド

この接続の STOP SYNCHRONIZATION DELETE を設定します。

構文

```
public void StopSynchronizationDelete ()
```

1.1.34 Synchronize(String^, SyncObserver^) メソッド

Ultra Light アプリケーションで同期を開始します。

構文

```
public void Synchronize (syncParms, observer)
```

パラメータ

syncParms セミコロンで区切った同期プロファイルオプションのリスト。
observer ステータス更新の送信先となる observer コールバック。

備考

このメソッドで、Mobile Link サーバとの同期を開始します。このメソッドは、同期が完了するまで戻りませんが、同期中に、別の接続を使用する追加スレッドがデータベースにアクセスできます。

UI スレッドからこのメソッドを呼び出さないでください。

1.1.35 SynchronizeAsync(String^) メソッド

Ultra Light アプリケーションで同期を開始します。

構文

```
public IAsyncActionWithProgress< SyncStatus^> SynchronizeAsync (syncParms)
```

パラメータ

syncParms セミコロンで区切った同期プロファイルオプションのリスト。

備考

このメソッドで、Mobile Link サーバとの同期を開始します。この方法は非同期です。同期中に、別の接続を使用する追加スレッドがデータベースにアクセスできます。

1.1.36 TriggerEvent(String^, String^) メソッド

ユーザ定義のイベントをトリガして、登録されたすべてのキューに通知を送信します。

構文

```
public unsigned int TriggerEvent (eventName, parameters)
```

パラメータ

eventName トリガするシステム定義またはユーザ定義のイベントの名前。

parameters 名前=値のペアをセミコロンで区切ったパラメータのオプションリスト。

戻り値

送信済みのイベント通知の数。

備考

通知が読み込まれた後、パラメータの値が GetNotificationParameter によって読み込まれます。

1.1.37 ValidateDatabase(uint16, String^, ValidateCallback^) メソッド

この接続でのデータベースを検証します。

構文

```
public void ValidateDatabase (flags, tableName, cb)
```

パラメータ

flags 検証のタイプを制御するフラグ。後述の例を参照してください。
tableName 検証する特定のテーブルまたは NULL。
cb 検証の進行状況の情報を受け取る関数。

備考

このルーチンに渡されるフラグに応じて、テーブル、インデックス、およびデータベースページを検証できます。検証中に情報を受け取るには、コールバック関数を実装し、アドレスをこのルーチンに渡します。検証対象を特定のテーブルに限定するには、テーブル名を tableName パラメータとしてこのメソッドに渡します。

flags パラメータは、次のいずれかの値の組み合わせです。

- ULVF_TABLE
- ULVF_INDEX
- ULVF_DATABASE

- ULVF_EXPRESS または ULVF_FULL_VALIDATE

```
flags = ULVF_TABLE | ULVF_INDEX | ULVF_EXPRESS;
```

1.1.38 ValidateDatabaseAsync(uint16flags, String^) メソッド

ValidateDatabase 操作を非同期に実行します。

構文

```
public IAsyncActionWithProgress< ValidateData^> ValidateDatabaseAsync (,
tableName)
```

戻り値

進捗レポートの待機可能な非同期アクション。

1.2 Constants クラス

Ultra Light WinRT API に使用可能な定数を指定します。

ネームスペース

```
UltraLite
```

構文

```
public ref class  sealed
```

メンバー

Constants のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

プロパティ

変更子とタイプ	プロパティ	説明
public static property int64	BLOB_CONTINUE [38 ページ]	ResultSet.GetChars メソッドまたは ResultSet.GetBytes メソッドを使用してデータを読み込む場合に使用します。
public static property String	SYNC_ALL [39 ページ]	どのパブリケーションにも含まれていないテーブルも含め、データベース内の no sync のマークが付いていないすべてのテーブルを同期させます。
public static property String	SYNC_ALL_PUBS [39 ページ]	パブリケーション内のすべてのテーブルを同期させます。

このセクションの内容:

BLOB_CONTINUE プロパティ [38 ページ]
ResultSet.GetChars メソッドまたは ResultSet.GetBytes メソッドを使用してデータを読み込む場合に使用します。

SYNC_ALL プロパティ [39 ページ]
どのパブリケーションにも含まれていないテーブルも含め、データベース内の no sync のマークが付いていないすべてのテーブルを同期させます。

SYNC_ALL_PUBS プロパティ [39 ページ]
パブリケーション内のすべてのテーブルを同期させます。

1.2.1 BLOB_CONTINUE プロパティ

ResultSet.GetChars メソッドまたは ResultSet.GetBytes メソッドを使用してデータを読み込む場合に使用します。

構文

```
public static property int64 BLOB_CONTINUE {get;}
```

備考

この値は、読み込むデータのチャンクが、最後のチャンクが読み込まれた位置から続いている必要があることを示します。

ResultSet::GetChars ResultSet::GetBytes を参照してください。

1.2.2 SYNC_ALL プロパティ

どのパブリケーションにも含まれていないテーブルも含め、データベース内の no sync のマークが付いていないすべてのテーブルを同期させます。

構文

```
public static property String SYNC_ALL {get;}
```

1.2.3 SYNC_ALL_PUBS プロパティ

パブリケーション内のすべてのテーブルを同期させます。

構文

```
public static property String SYNC_ALL_PUBS {get;}
```

1.3 Cursor インタフェース

Ultra Light データベース内のカーソルを表します。

ネームスペース

```
UltraLite
```

構文

```
public interface class
```

メンバー

Cursor のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public void	AfterLast() [44 ページ]	カーソルを最後のローの後に移動します。
public void	AppendBytes [45 ページ]	バイトをカラムに追加します。
public void	AppendChars [46 ページ]	文字列のチャンクをカラムに追加します。
public void	BeforeFirst() [48 ページ]	カーソルを最初のローの前に移動します。
public void	Close() [48 ページ]	このオブジェクトを破棄します。
public void	Delete() [49 ページ]	現在のローを削除し、カーソルを次の有効なローに移動します。
public void	DeleteNamed(String^) [49 ページ]	現在のローを削除し、カーソルを次の有効なローに移動します。
public void	First() [49 ページ]	カーソルを最初のローに移動します。
public int64	GetBinaryLength [50 ページ]	カラムの値のバイナリ長さを取得します。
public bool	GetBool [51 ページ]	カラムから値を boolean としてフェッチします。
public uint8	GetByte [53 ページ]	カラムから値を byte としてフェッチします。
public int64	GetBytes [54 ページ]	カラムからバイナリチャンクを取得します。
public int64	GetChars [56 ページ]	カラムからワイド文字列のチャンクを取得します。
public DateTime	GetDateTime [59 ページ]	カラムから値を DateTime としてフェッチします。
public double	GetDouble [60 ページ]	カラムから値を double としてフェッチします。
public float	GetFloat [62 ページ]	カラムから値を float としてフェッチします。
public Guid	GetGuid [63 ページ]	カラムから値を GUID としてフェッチします。
public int16	GetInt16 [65 ページ]	カラムから値を 16 ビット整数としてフェッチします。
public int	GetInt32 [66 ページ]	カラムから値を整数としてフェッチします。
public int64	GetInt64 [68 ページ]	カラムから値を 64 ビット整数としてフェッチします。
public unsigned int	GetRowCount(unsigned int) [69 ページ]	テーブルのローの数を取得します。
public uint8	GetState() [70 ページ]	カーソルの内部ステータスを取得します。
public String	GetString [70 ページ]	カラムから値を文字列としてフェッチします。
public int64	GetStringLength [72 ページ]	カラムの値の文字列長さを取得します。
public uint16	GetUInt16 [73 ページ]	カラムから値を 16 ビット符号なし整数としてフェッチします。
public unsigned int	GetUInt32 [75 ページ]	カラムから値を 32 ビット符号なし整数としてフェッチします。

変更子とタイプ	メソッド	説明
public uint64	GetUInt64 [76 ページ]	カラムから値を 64 ビット符号なし整数としてフェッチします。
public bool	IsNull [78 ページ]	カラム値が NULL かどうかをチェックします。
public void	Last() [79 ページ]	カーソルを最後のローに移動します。
public bool	Next() [79 ページ]	カーソルをロー 1 つ分進めます。
public bool	Previous() [80 ページ]	カーソルをロー 1 つ分戻します。
public void	Relative(int) [80 ページ]	カーソルを、現在のカーソルの位置から、offset で指定したロー数分移動します。
public void	SetBool [81 ページ]	カラムを boolean 値に設定します。
public void	SetByte [82 ページ]	カラムを byte 値に設定します。
public void	SetBytes [84 ページ]	カラムを binary 値に設定します。
public void	SetDateTime [85 ページ]	カラムを DateTime 値に設定します。
public void	SetDefault [87 ページ]	カラムをそのデフォルト値に設定します。
public void	SetDouble [88 ページ]	カラムを double 値に設定します。
public void	SetFloat [89 ページ]	カラムを float 値に設定します。
public void	SetGuid [91 ページ]	カラムを GUID 値に設定します。
public void	SetInt16 [92 ページ]	カラムを整数値に設定します。
public void	SetInt32 [94 ページ]	カラムを整数値に設定します。
public void	SetInt64 [95 ページ]	カラムを整数値に設定します。
public void	SetNull [97 ページ]	カラムを NULL に設定します。
public void	SetString [98 ページ]	カラムを文字列値に設定します。
public void	SetUInt16 [99 ページ]	カラムを符号なし 16 ビット整数値に設定します。
public void	SetUInt32 [101 ページ]	カラムを符号なし 32 ビット整数値に設定します。
public void	SetUInt64 [102 ページ]	カラムを符号なし 64 ビット整数値に設定します。
public void	Update() [103 ページ]	現在の行を更新します。
public void	UpdateBegin() [104 ページ]	カラムの設定に使用される更新モードを選択します。

備考

このインタフェースは、ResultSet クラスと Table クラスによって実装されます。

このセクションの内容:

AfterLast() メソッド [44 ページ]
カーソルを最後のローの後に移動します。

AppendBytes メソッド [45 ページ]
バイトをカラムに追加します。

AppendChars メソッド [46 ページ]
文字列のチャンクをカラムに追加します。

BeforeFirst() メソッド [48 ページ]
カーソルを最初のローの前に移動します。

CloseObject() メソッド [48 ページ]
このオブジェクトを破棄します。

Delete() メソッド [49 ページ]
現在のローを削除し、カーソルを次の有効なローに移動します。

DeleteNamed(String^) メソッド [49 ページ]
現在のローを削除し、カーソルを次の有効なローに移動します。

First() メソッド [49 ページ]
カーソルを最初のローに移動します。

GetBinaryLength メソッド [50 ページ]
カラムの値のバイナリ長さを取得します。

GetBool メソッド [51 ページ]
カラムから値を boolean としてフェッチします。

GetByte メソッド [53 ページ]
カラムから値を byte としてフェッチします。

GetBytes メソッド [54 ページ]
カラムからバイナリチャンクを取得します。

GetChars メソッド [56 ページ]
カラムからワイド文字列のチャンクを取得します。

GetDateTime メソッド [59 ページ]
カラムから値を DateTime としてフェッチします。

GetDouble メソッド [60 ページ]
カラムから値を double としてフェッチします。

GetFloat メソッド [62 ページ]
カラムから値を float としてフェッチします。

GetGuid メソッド [63 ページ]
カラムから値を GUID としてフェッチします。

GetInt16 メソッド [65 ページ]
カラムから値を 16 ビット整数としてフェッチします。

GetInt32 メソッド [66 ページ]
カラムから値を 32 ビット符号付き整数としてフェッチします。

GetInt64 メソッド [68 ページ]
カラムから値を 64 ビット符号付き整数としてフェッチします。

SetInt16 メソッド [92 ページ]
カラムを 16 ビット符号付き整数値に設定します。

SetInt32 メソッド [94 ページ]
カラムを 32 ビット符号付き整数値に設定します。

SetInt64 メソッド [95 ページ]
カラムを整数値に設定します。

SetNull メソッド [97 ページ]
カラムを NULL に設定します。

SetString メソッド [98 ページ]
カラムを文字列値に設定します。

SetUInt16 メソッド [99 ページ]
カラムを符号なし 16 ビット整数値に設定します。

SetUInt32 メソッド [101 ページ]
カラムを符号なし 32 ビット整数値に設定します。

SetUInt64 メソッド [102 ページ]
カラムを符号なし 64 ビット整数値に設定します。

Update() メソッド [103 ページ]
現在の行を更新します。

UpdateBegin() メソッド [104 ページ]
カラムの設定に使用される更新モードを開始します。

1.3.1 AfterLast() メソッド

カーソルを最後のローの後に移動します。

構文

```
public void AfterLast ()
```

1.3.2 AppendBytes メソッド

バイトをカラムに追加します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	AppendBytes(String^, const Array< uint8 >^, int, int) [45 ページ]	バイトをカラムに追加します。
public void	AppendBytes(uint16, const Array< uint8 >^, int, int) [46 ページ]	バイトをカラムに追加します。

このセクションの内容:

1.3.2.1 AppendBytes(String^, const Array< uint8 >^, int, int) メソッド

バイトをカラムに追加します。

構文

```
public void AppendBytes (cname, value, offset, size)
```

パラメータ

cname カラム名。
value 追加するバイトのチャンク。
offset バイトのチャンク内でのオフセット (開始位置)。
size バイトのチャンクのサイズ (バイト単位)。

備考

AppendBytes メソッド呼び出しにより現時点で書き込み済みのカラムの末尾に、指定のバイトが追加されます。

1.3.2.2 AppendBytes(uint16, const Array< uint8 >^, int, int) メソッド

バイトをカラムに追加します。

構文

```
public void AppendBytes (cid, value, offset, size)
```

パラメータ

cid 0 から始まるカラムの序数。
value 追加するバイトのチャンク。
offset バイトのチャンク内でのオフセット (開始位置)。
size バイトのチャンクのサイズ (バイト単位)。

備考

AppendBytes メソッド呼び出しにより現時点で書き込み済みのカラムの末尾に、指定のバイトが追加されます。

1.3.3 AppendChars メソッド

文字列のチャンクをカラムに追加します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	AppendChars(String^, const Array< wchar_t >^, int, int) [47 ページ]	文字列のチャンクをカラムに追加します。

変更子とタイプ	オーバーロード名	説明
public void	AppendChars(uint16, const Array< wchar_t >^, int, int) [48 ページ]	文字列のチャンクをカラムに追加します。

このセクションの内容:

1.3.3.1 AppendChars(String^, const Array< wchar_t >^, int, int) メソッド

文字列のチャンクをカラムに追加します。

構文

```
public void AppendChars (cname, value, offset, size)
```

パラメータ

cname カラム名。
value 追加する文字列のチャンク。
offset 文字列のチャンク内でのオフセット (開始位置)。
size 文字列のチャンクの文字単位での長さです。

備考

このメソッドは、AppendChars メソッド呼び出しにより現時点で書き込み済みの文字列の末尾に、指定の文字列を追加します。

1.3.3.2 AppendChars(uint16, const Array< wchar_t >^, int, int) メソッド

文字列のチャンクをカラムに追加します。

構文

```
public void AppendChars (cid, value, offset, size)
```

パラメータ

cid 0 から始まるカラムの序数。
value 追加する文字列のチャンク。
offset 文字列のチャンク内でのオフセット (開始位置)。
size 文字列のチャンクの文字単位での長さです。

備考

このメソッドは、AppendChars メソッド呼び出しにより現時点で書き込み済みの文字列の末尾に、指定の文字列を追加します。

1.3.4 BeforeFirst() メソッド

カーソルを最初のローの前に移動します。

構文

```
public void BeforeFirst ()
```

1.3.5 CloseObject() メソッド

このオブジェクトを破棄します。

構文

```
public void CloseObject ()
```

1.3.6 Delete() メソッド

現在のローを削除し、カーソルを次の有効なローに移動します。

構文

```
public void Delete ()
```

1.3.7 DeleteNamed(String^) メソッド

現在のローを削除し、カーソルを次の有効なローに移動します。

構文

```
public void DeleteNamed (tableName)
```

パラメータ

tableName テーブル名またはその相関 (同じテーブル名を共有する複数のカラムがデータベースに存在する場合に必要)。

1.3.8 First() メソッド

カーソルを最初のローに移動します。

構文

```
public void First ()
```

1.3.9 GetBinaryLength メソッド

カラムの値のバイナリ長さを取得します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public int64	GetBinaryLength(String^) [50 ページ]	カラムの値のバイナリ長さを取得します。
public int64	GetBinaryLength(uint16) [51 ページ]	カラムの値のバイナリ長さを取得します。

このセクションの内容:

GetBinaryLength(String^) メソッド [50 ページ]
カラムの値のバイナリ長さを取得します。

GetBinaryLength(uint16) メソッド [51 ページ]
カラムの値のバイナリ長さを取得します。

1.3.9.1 GetBinaryLength(String^) メソッド

カラムの値のバイナリ長さを取得します。

構文

```
public int64 GetBinaryLength (cname)
```

パラメータ

cname カラム名。

戻り値

binary としてのカラムの値の長さ。

1.3.9.2 GetBinaryLength(uint16) メソッド

カラムの値のバイナリ長さを取得します。

構文

```
public int64 GetBinaryLength (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

binary としてのカラムの値の長さ。

1.3.10 GetBool メソッド

カラムから値を boolean としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public bool	GetBool(String^) [52 ページ]	カラムから値を boolean としてフェッチします。
public bool	GetBool(uint16) [52 ページ]	カラムから値を boolean としてフェッチします。

このセクションの内容:

GetBool(String^) メソッド [52 ページ]
カラムから値を boolean としてフェッチします。

GetBool(uint16) メソッド [52 ページ]
カラムから値を boolean としてフェッチします。

1.3.10.1 GetBool(String^) メソッド

カラムから値を boolean としてフェッチします。

構文

```
public bool GetBool (cname)
```

パラメータ

cname カラム名。

戻り値

boolean としてのカラム値。

1.3.10.2 GetBool(uint16) メソッド

カラムから値を boolean としてフェッチします。

構文

```
public bool GetBool (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

boolean としてのカラム値。

1.3.11 GetByte メソッド

カラムから値を byte としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public uint8	GetByte(String^) [53 ページ]	カラムから値を byte としてフェッチします。
public uint8	GetByte(uint16) [54 ページ]	カラムから値を byte としてフェッチします。

このセクションの内容:

GetByte(String^) メソッド [53 ページ]
カラムから値を byte としてフェッチします。

GetByte(uint16) メソッド [54 ページ]
カラムから値を byte としてフェッチします。

1.3.11.1 GetByte(String^) メソッド

カラムから値を byte としてフェッチします。

構文

```
public uint8 GetByte (cname)
```

パラメータ

cname カラム名。

戻り値

byte としてのカラム値。

1.3.11.2 GetByte(uint16) メソッド

カラムから値を byte としてフェッチします。

構文

```
public uint8 GetByte (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

byte としてのカラム値。

1.3.12 GetBytes メソッド

カラムからバイナリチャンクを取得します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public int64	GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) [55 ページ]	カラムからバイナリチャンクを取得します。
public int64	GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) [55 ページ]	カラムからバイナリチャンクを取得します。

このセクションの内容:

GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) メソッド [55 ページ]
カラムからバイナリチャンクを取得します。

GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) メソッド [55 ページ]
カラムからバイナリチャンクを取得します。

1.3.12.1 GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) メソッド

カラムからバイナリチャンクを取得します。

構文

```
public int64 GetBytes (cname, srcOffset, dst, dstOffset, count)
```

パラメータ

cname カラム名。
dst バイトを保持するバッファ。
count バッファのサイズ (バイト単位)。
srcOffset 値内でのオフセット (読み込み開始位置)、または前回の読み込みが終了したところから続行する場合は BLOB_CONTINUE 定数。
dstOffset 宛先のバッファ内でのオフセット (バイトのコピー位置)。

戻り値

宛先のバッファにコピーされたバイト数。dst 値が NULL の場合は、残りのバイト数が返されます。カラムが NULL のときは、dst パラメータに空の文字列が返されます。

備考

0 が返された場合は、値の最後に到達しました。

IsNull メソッドを使用して、NULL と空の文字列を区別してください。

1.3.12.2 GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) メソッド

カラムからバイナリチャンクを取得します。

構文

```
public int64 GetBytes (cid, srcOffset, dst, dstOffset, count)
```

パラメータ

cid 0 から始まるカラムの序数。
dst バイトを保持するバッファ。
count バッファのサイズ (バイト単位)。
srcOffset 値内でのオフセット (読み込み開始位置)、または前回の読み込みが終了したところから続行する場合は BLOB_CONTINUE 定数。
dstOffset 宛先のバッファ内でのオフセット (バイトのコピー位置)。

戻り値

宛先のバッファにコピーされたバイト数。dst 値が NULL の場合、このメソッドは残りのバイト数を返します。カラムが NULL のときは、dst パラメータに空の文字列が返されます。

備考

0 が返された場合は、値の最後に到達しました。

IsNull メソッドを使用して、NULL と空の文字列を区別してください。

1.3.13 GetChars メソッド

カラムからワイド文字列のチャンクを取得します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public int64	GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) [57 ページ]	カラムからワイド文字列のチャンクを取得します。
public int64	GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) [58 ページ]	カラムからワイド文字列のチャンクを取得します。

このセクションの内容:

GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) メソッド [57 ページ]
カラムからワイド文字列のチャンクを取得します。

GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) メソッド [58 ページ]
カラムからワイド文字列のチャンクを取得します。

1.3.13.1 GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) メソッド

カラムからワイド文字列のチャンクを取得します。

構文

```
public int64 GetChars (cname, srcOffset, dst, dstOffset, count)
```

パラメータ

cname カラム名。
dst 文字列のチャンクを保持するバッファ。文字列は、トランケートされても NULL で終了します。
count バッファのサイズ (文字数)。
srcOffset 値内でのオフセット (読み込み開始位置)、または前回の読み込みが終了したところから続行する場合は BLOB_CONTINUE 定数。
dstOffset 宛先のバッファ内でのオフセット (文字のコピー位置)。

戻り値

宛先のバッファにコピーされた文字数 (NULL ターミネータを含まない)。dst 値が NULL の場合、このメソッドは文字列の残りの文字数を返します。カラムが NULL のときは、dst パラメータに空の文字列が返されます。

備考

0 が返された場合は、値の最後に到達しました。

IsNull メソッドを使用して、NULL と空の文字列を区別してください。

1.3.13.2 GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) メソッド

カラムからワイド文字列のチャンクを取得します。

構文

```
public int64 GetChars (cid, srcOffset, dst, dstOffset, count)
```

パラメータ

cid 0 から始まるカラムの序数。
dst 文字列のチャンクを保持するバッファ。文字列は、トランケートされても NULL で終了します。
count バッファのサイズ (文字数)。
srcOffset 値内でのオフセット (読み込み開始位置)、または前回の読み込みが終了したところから続行する場合は BLOB_CONTINUE 定数。
dstOffset 宛先のバッファ内でのオフセット (文字のコピー位置)。

戻り値

宛先のバッファにコピーされた文字数 (NULL ターミネータを含まない)。dst 値が NULL の場合、このメソッドは文字列の残りの文字数を返します。カラムが NULL のときは、dst パラメータに空の文字列が返されます。

備考

0 が返された場合は、値の最後に到達しました。

IsNull メソッドを使用して、NULL と空の文字列を区別してください。

1.3.14 GetDateTime メソッド

カラムから値を DateTime としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public DateTime	GetDateTime(String^) [59 ページ]	カラムから値を DateTime としてフェッチします。
public DateTime	GetDateTime(uint16) [60 ページ]	カラムから値を DateTime としてフェッチします。

このセクションの内容:

GetDateTime(String^) メソッド [59 ページ]
カラムから値を DateTime としてフェッチします。

GetDateTime(uint16) メソッド [60 ページ]
カラムから値を DateTime としてフェッチします。

1.3.14.1 GetDateTime(String^) メソッド

カラムから値を DateTime としてフェッチします。

構文

```
public DateTime GetDateTime (cname)
```

パラメータ

cname カラム名。

戻り値

DateTime 値。

1.3.14.2 GetDateTime(uint16) メソッド

カラムから値を DateTime としてフェッチします。

構文

```
public DateTime GetDateTime (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

DateTime 値。

1.3.15 GetDouble メソッド

カラムから値を double としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public double	GetDouble(String^) [61 ページ]	カラムから値を double としてフェッチします。
public double	GetDouble(uint16) [61 ページ]	カラムから値を double としてフェッチします。

このセクションの内容:

GetDouble(String^) メソッド [61 ページ]
カラムから値を double としてフェッチします。

GetDouble(uint16) メソッド [61 ページ]
カラムから値を double としてフェッチします。

1.3.15.1 GetDouble(String^) メソッド

カラムから値を double としてフェッチします。

構文

```
public double GetDouble (cname)
```

パラメータ

cname カラム名。

戻り値

double としてのカラム値。

1.3.15.2 GetDouble(uint16) メソッド

カラムから値を double としてフェッチします。

構文

```
public double GetDouble (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

double としてのカラム値。

1.3.16 GetFloat メソッド

カラムから値を float としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public float	GetFloat(String^) [62 ページ]	カラムから値を float としてフェッチします。
public float	GetFloat(uint16) [63 ページ]	カラムから値を float としてフェッチします。

このセクションの内容:

GetFloat(String^) メソッド [62 ページ]
カラムから値を float としてフェッチします。

GetFloat(uint16) メソッド [63 ページ]
カラムから値を float としてフェッチします。

1.3.16.1 GetFloat(String^) メソッド

カラムから値を float としてフェッチします。

構文

```
public float GetFloat (cname)
```

パラメータ

cname カラム名。

戻り値

float としてのカラム値。

1.3.16.2 GetFloat(uint16) メソッド

カラムから値を float としてフェッチします。

構文

```
public float GetFloat (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

float としてのカラム値。

1.3.17 GetGuid メソッド

カラムから値を GUID としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public Guid	GetGuid(String^) [64 ページ]	カラムから値を GUID としてフェッチします。
public Guid	GetGuid(uint16) [64 ページ]	カラムから値を GUID としてフェッチします。

このセクションの内容:

GetGuid(String^) メソッド [64 ページ]
カラムから値を GUID としてフェッチします。

GetGuid(uint16) メソッド [64 ページ]
カラムから値を GUID としてフェッチします。

1.3.17.1 GetGuid(String^) メソッド

カラムから値を GUID としてフェッチします。

構文

```
public Guid GetGuid (cname)
```

パラメータ

cname カラム名。

戻り値

GUID 値。

1.3.17.2 GetGuid(uint16) メソッド

カラムから値を GUID としてフェッチします。

構文

```
public Guid GetGuid (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

GUID 値。

1.3.18 GetInt16 メソッド

カラムから値を 16 ビット整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public int16	GetInt16(String^) [65 ページ]	カラムから値を 16 ビット整数としてフェッチします。
public int16	GetInt16(uint16) [66 ページ]	カラムから値を 16 ビット整数としてフェッチします。

このセクションの内容:

1.3.18.1 GetInt16(String^) メソッド

カラムから値を 16 ビット整数としてフェッチします。

構文

```
public int16 GetInt16 (cname)
```

パラメータ

cname カラム名。

戻り値

整数としてのカラム値。

1.3.18.2 GetInt16(uint16) メソッド

カラムから値を 16 ビット整数としてフェッチします。

構文

```
public int16 GetInt16 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

整数としてのカラム値。

1.3.19 GetInt32 メソッド

カラムから値を 32 ビット符号付き整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public int	GetInt32(String^) [67 ページ]	カラムから値を 32 ビット符号付き整数としてフェッチします。
public int	GetInt32(uint16) [67 ページ]	カラムから値を 32 ビット符号付き整数としてフェッチします。

このセクションの内容:

GetInt32(String^) メソッド [67 ページ]
カラムから値を 32 ビット符号付き整数としてフェッチします。

GetInt32(uint16) メソッド [67 ページ]
カラムから値を 32 ビット符号付き整数としてフェッチします。

1.3.19.1 GetInt32(String^) メソッド

カラムから値を 32 ビット符号付き整数としてフェッチします。

構文

```
public int GetInt32 (cname)
```

パラメータ

cname カラム名。

戻り値

32 ビット符号付き整数としてのカラム値。

1.3.19.2 GetInt32(uint16) メソッド

カラムから値を 32 ビット符号付き整数としてフェッチします。

構文

```
public int GetInt32 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

32 ビット符号付き整数としてのカラム値。

1.3.20 GetInt64 メソッド

カラムから値を 64 ビット符号付き整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public int64	GetInt64(String^) [68 ページ]	カラムから値を 64 ビット符号付き整数としてフェッチします。
public int64	GetInt64(uint16) [69 ページ]	カラムから値を 64 ビット符号付き整数としてフェッチします。

このセクションの内容:

1.3.20.1 GetInt64(String^) メソッド

カラムから値を 64 ビット符号付き整数としてフェッチします。

構文

```
public int64 GetInt64 (cname)
```

パラメータ

cname カラム名。

戻り値

64 ビット符号付き整数としてのカラム値。

1.3.20.2 GetInt64(uint16) メソッド

カラムから値を 64 ビット符号付き整数としてフェッチします。

構文

```
public int64 GetInt64 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

64 ビット符号付き整数としてのカラム値。

1.3.21 GetRowCount(unsigned int) メソッド

テーブルのローの数を取得します。

構文

```
public unsigned int GetRowCount (threshold)
```

パラメータ

threshold カウントするローの数の制限。無限を表すには 0 を設定します。

戻り値

テーブル内のローの数。

備考

このメソッドは、次の文を実行することと同義です。

```
SELECT COUNT(*) FROM table
```

1.3.22 GetState() メソッド

カーソルの内部ステータスを取得します。

構文

```
public uint8 GetState ()
```

戻り値

カーソルのステータス

1.3.23 GetString メソッド

カラムから値を文字列としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public String	GetString(String^) [71 ページ]	カラムから値を文字列としてフェッチします。
public String	GetString(uint16) [71 ページ]	カラムから値を文字列としてフェッチします。

このセクションの内容:

GetString(String^) メソッド [71 ページ]
カラムから値を文字列としてフェッチします。

GetString(uint16) メソッド [71 ページ]
カラムから値を文字列としてフェッチします。

1.3.23.1 GetString(String^) メソッド

カラムから値を文字列としてフェッチします。

構文

```
public String GetString (cname)
```

パラメータ

cname カラム名。

戻り値

文字列値。

1.3.23.2 GetString(uint16) メソッド

カラムから値を文字列としてフェッチします。

構文

```
public String GetString (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

文字列値。

1.3.24 GetStringLength メソッド

カラムの値の文字列長さを取得します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public int64	GetStringLength(String^) [72 ページ]	カラムの値の文字列長さを取得します。
public int64	GetStringLength(uint16) [73 ページ]	カラムの値の文字列長さを取得します。

このセクションの内容:

GetStringLength(String^) メソッド [72 ページ]
カラムの値の文字列長さを取得します。

GetStringLength(uint16) メソッド [73 ページ]
カラムの値の文字列長さを取得します。

1.3.24.1 GetStringLength(String^) メソッド

カラムの値の文字列長さを取得します。

構文

```
public int64 GetStringLength (cname)
```

パラメータ

cname カラム名。

戻り値

文字列型のカラムの値の文字数。値が NULL またはエラーが発生した場合は 0 が返されます。

1.3.24.2 GetStringLength(uint16) メソッド

カラムの値の文字列長さを取得します。

構文

```
public int64 GetStringLength (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

文字列型のカラムの値の文字数。

1.3.25 GetUInt16 メソッド

カラムから値を 16 ビット符号なし整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public uint16	GetUInt16(String^) [74 ページ]	カラムから値を 16 ビット符号なし整数としてフェッチします。
public uint16	GetUInt16(uint16) [74 ページ]	カラムから値を 16 ビット符号なし整数としてフェッチします。

このセクションの内容:

GetUInt16(String^) メソッド [74 ページ]
カラムから値を 16 ビット符号なし整数としてフェッチします。

GetUInt16(uint16) メソッド [74 ページ]
カラムから値を 16 ビット符号なし整数としてフェッチします。

1.3.25.1 GetUInt16(String^) メソッド

カラムから値を 16 ビット符号なし整数としてフェッチします。

構文

```
public uint16 GetUInt16 (cname)
```

パラメータ

cname カラム名。

戻り値

16 ビット符号なし整数としてのカラム値。

1.3.25.2 GetUInt16(uint16) メソッド

カラムから値を 16 ビット符号なし整数としてフェッチします。

構文

```
public uint16 GetUInt16 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

16 ビット符号なし整数としてのカラム値。

1.3.26 GetUInt32 メソッド

カラムから値を 32 ビット符号なし整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public unsigned int	GetUInt32(String^) [75 ページ]	カラムから値を 32 ビット符号なし整数としてフェッチします。
public unsigned int	GetUInt32(uint16) [76 ページ]	カラムから値を 32 ビット符号なし整数としてフェッチします。

このセクションの内容:

1.3.26.1 GetUInt32(String^) メソッド

カラムから値を 32 ビット符号なし整数としてフェッチします。

構文

```
public unsigned int GetUInt32 (cname)
```

パラメータ

cname カラム名。

戻り値

32 ビット符号なし整数としてのカラム値。

1.3.26.2 GetUInt32(uint16) メソッド

カラムから値を 32 ビット符号なし整数としてフェッチします。

構文

```
public unsigned int GetUInt32 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

32 ビット符号なし整数としてのカラム値。

1.3.27 GetUInt64 メソッド

カラムから値を 64 ビット符号なし整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public uint64	GetUInt64(String^) [77 ページ]	カラムから値を 64 ビット符号なし整数としてフェッチします。
public uint64	GetUInt64(uint16) [77 ページ]	カラムから値を 64 ビット符号なし整数としてフェッチします。

このセクションの内容:

GetUInt64(String^) メソッド [77 ページ]
カラムから値を 64 ビット符号なし整数としてフェッチします。

GetUInt64(uint16) メソッド [77 ページ]
カラムから値を 64 ビット符号なし整数としてフェッチします。

1.3.27.1 GetUInt64(String^) メソッド

カラムから値を 64 ビット符号なし整数としてフェッチします。

構文

```
public uint64 GetUInt64 (cname)
```

パラメータ

cname カラム名。

戻り値

64 ビット符号なし整数としてのカラム値。

1.3.27.2 GetUInt64(uint16) メソッド

カラムから値を 64 ビット符号なし整数としてフェッチします。

構文

```
public uint64 GetUInt64 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

64 ビット符号なし整数としてのカラム値。

1.3.28 IsNull メソッド

カラム値が NULL かどうかをチェックします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public bool	IsNull(String^) [78 ページ]	カラム値が NULL かどうかをチェックします。
public bool	IsNull(uint16) [79 ページ]	カラム値が NULL かどうかをチェックします。

このセクションの内容:

1.3.28.1 IsNull(String^) メソッド

カラム値が NULL かどうかをチェックします。

構文

```
public bool IsNull (cname)
```

パラメータ

cname カラム名。

戻り値

カラム値が NULL の場合は true。

1.3.28.2 IsNull(uint16) メソッド

カラム値が NULL かどうかをチェックします。

構文

```
public bool IsNull (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラム値が NULL の場合は true。

1.3.29 Last() メソッド

カーソルを最後のローに移動します。

構文

```
public void Last ()
```

1.3.30 Next() メソッド

カーソルをロー 1 つ分進めます。

構文

```
public bool Next ()
```

戻り値

カーソルが正常に進められる場合は、true。カーソルが次のローに正常に進められる場合は、引き続きエラーが通知される可能性があります。たとえば SELECT 式の評価中に変換エラーが発生する可能性があります。この場合、カラム値を取得するときにもエラーが返されます。カーソルを進められなかった場合は、false が返されます。たとえば、次のローが存在しないなどです。この場合、移動後のカーソル位置は最後のローの後ろに設定されます。

1.3.31 Previous() メソッド

カーソルをロー 1 つ分戻します。

構文

```
public bool Previous ()
```

戻り値

カーソルをロー 1 つ分戻せた場合は、true。カーソルを戻せなかった場合は、false。移動後のカーソル位置は、最初のローの前に設定されます。

1.3.32 Relative(int) メソッド

カーソルを、現在のカーソルの位置から、offset で指定したロー数分移動します。

構文

```
public void Relative (offset)
```

パラメータ

offset 移動するローの数。

1.3.33 SetBool メソッド

カラムを boolean 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetBool(String^, bool) [81 ページ]	カラムを boolean 値に設定します。
public void	SetBool(uint16, bool) [82 ページ]	カラムを boolean 値に設定します。

このセクションの内容:

1.3.33.1 SetBool(String^, bool) メソッド

カラムを boolean 値に設定します。

構文

```
public void SetBool (cname, value)
```

パラメータ

cname カラム名。
value boolean 値。

1.3.33.2 SetBool(uint16, bool) メソッド

カラムを boolean 値に設定します。

構文

```
public void SetBool (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value boolean 値。

1.3.34 SetByte メソッド

カラムを byte 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetByte(String^, uint8) [83 ページ]	カラムを byte 値に設定します。
public void	SetByte(uint16, uint8) [83 ページ]	カラムを byte 値に設定します。

このセクションの内容:

SetByte(String^, uint8) メソッド [83 ページ]
カラムを byte 値に設定します。

SetByte(uint16, uint8) メソッド [83 ページ]
カラムを byte 値に設定します。

1.3.34.1 SetByte(String^, uint8) メソッド

カラムを byte 値に設定します。

構文

```
public void SetByte (cname, value)
```

パラメータ

cname カラム名。
value byte 値。

1.3.34.2 SetByte(uint16, uint8) メソッド

カラムを byte 値に設定します。

構文

```
public void SetByte (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value byte 値。

1.3.35 SetBytes メソッド

カラムを BINARY 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetBytes(String^, const Array< uint8 >^, int) [84 ページ]	カラムを BINARY 値に設定します。
public void	SetBytes(uint16, const Array< uint8 >^, int) [85 ページ]	カラムを BINARY 値に設定します。

このセクションの内容:

1.3.35.1 SetBytes(String^, const Array< uint8 >^, int) メソッド

カラムを BINARY 値に設定します。

構文

```
public void SetBytes (cname, value, length)
```

パラメータ

cname カラム名。
value バイナリ値。NULL を渡すことは、SetNull メソッドを呼び出すことと同じです。
length byte 配列値から取得するバイト単位の長さ。

1.3.35.2 SetBytes(uint16, const Array< uint8 >^, int) メソッド

カラムを BINARY 値に設定します。

構文

```
public void SetBytes (cid, value, length)
```

パラメータ

cid 0 から始まるカラムの序数。
value バイナリ値。NULL を渡すことは、SetNull メソッドを呼び出すことと同じです。
length byte 配列値から取得するバイト単位の長さ。

1.3.36 SetDateTime メソッド

カラムを DateTime 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetDateTime(String^, DateTime) [86 ページ]	カラムを DateTime 値に設定します。
public void	SetDateTime(uint16, DateTime) [86 ページ]	カラムを DateTime 値に設定します。

このセクションの内容:

SetDateTime(String^, DateTime) メソッド [86 ページ]
カラムを DateTime 値に設定します。

SetDateTime(uint16, DateTime) メソッド [86 ページ]
カラムを DateTime 値に設定します。

1.3.36.1 SetDateTime(String^, DateTime) メソッド

カラムを DateTime 値に設定します。

構文

```
public void SetDateTime (cname, value)
```

パラメータ

cname カラム名。
value DateTime 値。

1.3.36.2 SetDateTime(uint16, DateTime) メソッド

カラムを DateTime 値に設定します。

構文

```
public void SetDateTime (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value DateTime 値。

1.3.37 SetDefault メソッド

カラムをそのデフォルト値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetDefault(String^) [87 ページ]	カラムをそのデフォルト値に設定します。
public void	SetDefault(uint16) [87 ページ]	カラムをそのデフォルト値に設定します。

このセクションの内容:

SetDefault(String^) メソッド [87 ページ]
カラムをそのデフォルト値に設定します。

SetDefault(uint16) メソッド [87 ページ]
カラムをそのデフォルト値に設定します。

1.3.37.1 SetDefault(String^) メソッド

カラムをそのデフォルト値に設定します。

構文

```
public void SetDefault (cname)
```

パラメータ

cname カラム名。

1.3.37.2 SetDefault(uint16) メソッド

カラムをそのデフォルト値に設定します。

構文

```
public void SetDefault (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

1.3.38 SetDouble メソッド

カラムを double 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetDouble(String^, double) [88 ページ]	カラムを double 値に設定します。
public void	SetDouble(uint16, double) [89 ページ]	カラムを double 値に設定します。

このセクションの内容:

SetDouble(String^, double) メソッド [88 ページ]
カラムを double 値に設定します。

SetDouble(uint16, double) メソッド [89 ページ]
カラムを double 値に設定します。

1.3.38.1 SetDouble(String^, double) メソッド

カラムを double 値に設定します。

構文

```
public void SetDouble (cname, value)
```

パラメータ

cname カラム名。
value double 値。

1.3.38.2 SetDouble(uint16, double) メソッド

カラムを double 値に設定します。

構文

```
public void SetDouble (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value double 値。

1.3.39 SetFloat メソッド

カラムを float 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetFloat(String^, float) [90 ページ]	カラムを float 値に設定します。
public void	SetFloat(uint16, float) [90 ページ]	カラムを float 値に設定します。

このセクションの内容:

SetFloat(String^, float) メソッド [90 ページ]
カラムを float 値に設定します。

SetFloat(uint16, float) メソッド [90 ページ]
カラムを float 値に設定します。

1.3.39.1 SetFloat(String^, float) メソッド

カラムを float 値に設定します。

構文

```
public void SetFloat (cname, value)
```

パラメータ

cname カラム名。
value float 値。

1.3.39.2 SetFloat(uint16, float) メソッド

カラムを float 値に設定します。

構文

```
public void SetFloat (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value float 値。

1.3.40 SetGuid メソッド

カラムを GUID 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetGuid(uint16, Guid) [91 ページ]	カラムを GUID 値に設定します。
public void	SetGuid(uint16, Guid) [92 ページ]	カラムを GUID 値に設定します。

このセクションの内容:

1.3.40.1 SetGuid(uint16, Guid) メソッド

カラムを GUID 値に設定します。

構文

```
public void SetGuid (cname, value)
```

パラメータ

cname カラム名。
value GUID 値。

1.3.40.2 SetGuid(uint16, Guid) メソッド

カラムを GUID 値に設定します。

構文

```
public void SetGuid (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value GUID 値。

1.3.41 SetInt16 メソッド

カラムを 16 ビット符号付き整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetInt16(String^, int16) [93 ページ]	カラムを 16 ビット符号付き整数値に設定します。
public void	SetInt16(uint16, int16) [93 ページ]	カラムを 16 ビット符号付き整数値に設定します。

このセクションの内容:

SetInt16(String^, int16) メソッド [93 ページ]
カラムを 16 ビット符号付き整数値に設定します。

SetInt16(uint16, int16) メソッド [93 ページ]
カラムを 16 ビット符号付き整数値に設定します。

1.3.41.1 SetInt16(String^, int16) メソッド

カラムを 16 ビット符号付き整数値に設定します。

構文

```
public void SetInt16 (cname, value)
```

パラメータ

cname カラム名。
value 16 ビット符号付き整数値。

1.3.41.2 SetInt16(uint16, int16) メソッド

カラムを 16 ビット符号付き整数値に設定します。

構文

```
public void SetInt16 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 16 ビット符号付き整数値。

1.3.42 SetInt32 メソッド

カラムを 32 ビット符号付き整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetInt32(String^, int) [94 ページ]	カラムを 32 ビット符号付き整数値に設定します。
public void	SetInt32(uint16, int) [95 ページ]	カラムを 32 ビット符号付き整数値に設定します。

このセクションの内容:

SetInt32(String^, int) メソッド [94 ページ]
カラムを 32 ビット符号付き整数値に設定します。

SetInt32(uint16, int) メソッド [95 ページ]
カラムを整数値に設定します。

1.3.42.1 SetInt32(String^, int) メソッド

カラムを 32 ビット符号付き整数値に設定します。

構文

```
public void SetInt32 (cname, value)
```

パラメータ

cname カラム名。
value 32 ビット符号付き整数値。

1.3.42.2 SetInt32(uint16, int) メソッド

カラムを整数値に設定します。

構文

```
public void SetInt32 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号付き整数値。

1.3.43 SetInt64 メソッド

カラムを整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetInt64(String^, int64) [96 ページ]	カラムを整数値に設定します。
public void	SetInt64(uint16, int64) [96 ページ]	カラムを整数値に設定します。

このセクションの内容:

SetInt64(String^, int64) メソッド [96 ページ]
カラムを整数値に設定します。

SetInt64(uint16, int64) メソッド [96 ページ]
カラムを整数値に設定します。

1.3.43.1 SetInt64(String^, int64) メソッド

カラムを整数値に設定します。

構文

```
public void SetInt64 (cname, value)
```

パラメータ

cname カラム名。
value 符号付き整数値。

1.3.43.2 SetInt64(uint16, int64) メソッド

カラムを整数値に設定します。

構文

```
public void SetInt64 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号付き整数値。

1.3.44 SetNull メソッド

カラムを NULL に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetNull(String^) [97 ページ]	カラムを NULL に設定します。
public void	SetNull(uint16) [97 ページ]	カラムを NULL に設定します。

このセクションの内容:

SetNull(String^) メソッド [97 ページ]
カラムを NULL に設定します。

SetNull(uint16) メソッド [97 ページ]
カラムを NULL に設定します。

1.3.44.1 SetNull(String^) メソッド

カラムを NULL に設定します。

構文

```
public void SetNull (cname)
```

パラメータ

cname カラム名。

1.3.44.2 SetNull(uint16) メソッド

カラムを NULL に設定します。

構文

```
public void SetNull (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

1.3.45 SetString メソッド

カラムを文字列値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetString(String^, String^) [98 ページ]	カラムを文字列値に設定します。
public void	SetString(uint16, String^) [99 ページ]	カラムを文字列値に設定します。

このセクションの内容:

SetString(String^, String^) メソッド [98 ページ]
カラムを文字列値に設定します。

SetString(uint16, String^) メソッド [99 ページ]
カラムを文字列値に設定します。

1.3.45.1 SetString(String^, String^) メソッド

カラムを文字列値に設定します。

構文

```
public void SetString (cname, value)
```

パラメータ

cname カラム名。
value 文字列値。

1.3.45.2 SetString(uint16, String^) メソッド

カラムを文字列値に設定します。

構文

```
public void SetString (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 文字列値。

1.3.46 SetUInt16 メソッド

カラムを符号なし 16 ビット整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetUInt16(String^, uint16) [100 ページ]	カラムを符号なし 16 ビット整数値に設定します。
public void	SetUInt16(uint16, uint16) [100 ページ]	カラムを符号なし 16 ビット整数値に設定します。

このセクションの内容:

SetUInt16(String^, uint16) メソッド [100 ページ]
カラムを符号なし 16 ビット整数値に設定します。

SetUInt16(uint16, uint16) メソッド [100 ページ]
カラムを符号なし 16 ビット整数値に設定します。

1.3.46.1 SetUInt16(String^, uint16) メソッド

カラムを符号なし 16 ビット整数値に設定します。

構文

```
public void SetUInt16 (cname, value)
```

パラメータ

cname カラム名。
value 符号なし整数値。

1.3.46.2 SetUInt16(uint16, uint16) メソッド

カラムを符号なし 16 ビット整数値に設定します。

構文

```
public void SetUInt16 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号なし整数値。

1.3.47 SetUInt32 メソッド

カラムを符号なし 32 ビット整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetUInt32(String^, unsigned int) [101 ページ]	カラムを符号なし 32 ビット整数値に設定します。
public void	SetUInt32(uint16, unsigned int) [102 ページ]	カラムを符号なし 32 ビット整数値に設定します。

このセクションの内容:

1.3.47.1 SetUInt32(String^, unsigned int) メソッド

カラムを符号なし 32 ビット整数値に設定します。

構文

```
public void SetUInt32 (cname, value)
```

パラメータ

cname カラム名。

value 符号なし整数値。

1.3.47.2 SetUInt32(uint16, unsigned int) メソッド

カラムを符号なし 32 ビット整数値に設定します。

構文

```
public void SetUInt32 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号なし整数値。

1.3.48 SetUInt64 メソッド

カラムを符号なし 64 ビット整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	SetUInt64(String^, uint64) [103 ページ]	カラムを符号なし 64 ビット整数値に設定します。
public void	SetUInt64(uint16, uint64) [103 ページ]	カラムを符号なし 64 ビット整数値に設定します。

このセクションの内容:

SetUInt64(String^, uint64) メソッド [103 ページ]
カラムを符号なし 64 ビット整数値に設定します。

SetUInt64(uint16, uint64) メソッド [103 ページ]
カラムを符号なし 64 ビット整数値に設定します。

1.3.48.1 SetUInt64(String^, uint64) メソッド

カラムを符号なし 64 ビット整数値に設定します。

構文

```
public void SetUInt64 (cname, value)
```

パラメータ

cname カラム名。
value 符号なし整数値。

1.3.48.2 SetUInt64(uint16, uint64) メソッド

カラムを符号なし 64 ビット整数値に設定します。

構文

```
public void SetUInt64 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号なし整数値。

1.3.49 Update() メソッド

現在の行を更新します。

構文

```
public void Update ()
```

1.3.50 UpdateBegin() メソッド

カラムの設定に使用される更新モードを開始します。

構文

```
public void UpdateBegin ()
```

備考

Ultra Light が更新モードの場合、プライマリキー内のカラムは修正できません。ULResultSet.Update() は UpdateBegin() がコールされた後にローを更新します。Connection.Commit() または Rollback() はトランザクションをコミットまたはロールバックします。

1.4 CursorSchema インタフェース

ResultSetSchema と TableSchema に共通のインタフェース。

ネームスペース

```
UltraLite
```

構文

```
public interface class
```

メンバー

CursorSchema のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public uint16	GetColumnCount() [106 ページ]	結果セットまたはテーブル内のカラム数を取得します。

変更子とタイプ	メソッド	説明
public uint16	GetColumnID(String^) [106 ページ]	0 から始まるカラム ID を名前から取得します。
public String	GetColumnName(uint16, uint16type) [106 ページ]	0 から始まる ID を指定してカラムの名前を取得します。
public uint16	GetColumnPrecision(uint16) [107 ページ]	数値カラムの精度を取得します。
public uint16	GetColumnScale(uint16) [108 ページ]	数値カラムの位取りを取得します。
public int	GetColumnSize(uint16) [108 ページ]	カラムのサイズを取得します。
public uint16	GetColumnSQLType(uint16) [109 ページ]	カラムの SQL の型を取得します。
public uint16	GetColumnType(uint16) [109 ページ]	カラムの記憶タイプまたはホスト変数の型を取得します。
public bool	IsAliased(uint16) [110 ページ]	結果セット内のカラムにエイリアスが付与されているかどうかを示します。

このセクションの内容:

GetColumnCount() メソッド [106 ページ]
結果セットまたはテーブル内のカラム数を取得します。

GetColumnID(String^) メソッド [106 ページ]
0 から始まるカラム ID を名前から取得します。

GetColumnName(uint16, uint16) メソッド [106 ページ]
0 から始まる ID を指定してカラムの名前を取得します。

GetColumnPrecision(uint16) メソッド [107 ページ]
数値カラムの精度を取得します。

GetColumnScale(uint16) メソッド [108 ページ]
数値カラムの位取りを取得します。

GetColumnSize(uint16) メソッド [108 ページ]
カラムのサイズを取得します。

GetColumnSQLType(uint16) メソッド [109 ページ]
カラムの SQL の型を取得します。

GetColumnType(uint16) メソッド [109 ページ]
カラムの記憶タイプまたはホスト変数の型を取得します。

IsAliased(uint16) メソッド [110 ページ]
結果セット内のカラムにエイリアスが付与されているかどうかを示します。

1.4.1 GetColumnCount() メソッド

結果セットまたはテーブル内のカラム数を取得します。

構文

```
public uint16 GetColumnCount ()
```

戻り値

結果セットまたはテーブル内のカラム数。

1.4.2 GetColumnID(String^) メソッド

0 から始まるカラム ID を名前から取得します。

構文

```
public uint16 GetColumnID (columnName)
```

パラメータ

columnName カラムの名前。

戻り値

カラム ID。

1.4.3 GetColumnName(uint16, uint16) メソッド

0 から始まる ID を指定してカラムの名前を取得します。

構文

```
public String GetColumnName (cid, type)
```

パラメータ

cid 0 から始まるカラムの序数。
type カラム名の必要な型。

戻り値

存在する場合は、カラム名を格納する文字列。存在しない場合は、エラーがスローされます。

備考

選択した型、および SELECT 文でのカラムの宣言方法によっては、カラム名が [table-name].[column-name] という形式で返されることがあります。

type パラメータは、どの型のカラム名を返すかを指定します。

1.4.4 GetColumnPrecision(uint16) メソッド

数値カラムの精度を取得します。

構文

```
public uint16 GetColumnPrecision (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

数値カラムの精度。

1.4.5 GetColumnScale(uint16) メソッド

数値カラムの位取りを取得します。

構文

```
public uint16 GetColumnScale (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

数値カラムの位取り。

1.4.6 GetColumnSize(uint16) メソッド

カラムのサイズを取得します。

構文

```
public int GetColumnSize (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

固定長 CHAR または BINARY カラムのバイト単位の長さ、または GEOMETRY タイプカラム..

1.4.7 GetColumnSQLType(uint16) メソッド

カラムの SQL の型を取得します。

構文

```
public uint16 GetColumnSQLType (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラムが存在しない場合は、SQLTYPE_BAD_INDEX。

1.4.8 GetColumnType(uint16) メソッド

カラムの記憶タイプまたはホスト変数の型を取得します。

構文

```
public uint16 GetColumnType (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラムが存在しない場合は、TYPE_BAD_INDEX。

1.4.9 IsAliased(uint16) メソッド

結果セット内のカラムにエイリアスが付与されているかどうかを示します。

構文

```
public bool IsAliased (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラムのエイリアスが使用されている場合は true、そうでない場合は false。

1.5 DatabaseManager クラス

接続とデータベースを管理します。

ネームスペース

```
UltraLite
```

構文

```
public ref class  sealed
```

メンバー

DatabaseManager のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public static Connection	CreateDatabase(String^, String^) [112 ページ]	新しいデータベースを作成します。
public static FileTransfer	CreateFileTransfer(String^, uint16, String^, String^) [113 ページ]	FileTransfer オブジェクトを作成します。
public static void	DropDatabase(String^) [113 ページ]	現在実行していない既存のデータベースを消去します。
public static void	Fini() [114 ページ]	Ultra Light ランタイムをファイナライズします。
public static void	GetLastError(ULSqlCode *, String^*) [114 ページ]	最後の呼び出しに関連付けられているエラー情報をこの DatabaseManager に返します。
public static bool	Init() [115 ページ]	Ultra Light ランタイムを初期化します。
public static Connection	OpenConnection(String^) [115 ページ]	既存のデータベースへの新しい接続を開きます。
public static void	ValidateDatabase(String^, uint16, ValidateCallback^) [116 ページ]	データベースで低レベルのインデックス検証を実行します。
public static IAsyncActionWithProgress< ValidateData^>	ValidateDatabaseAsync(String^, uint16flags) [117 ページ]	ValidateDatabase 操作を非同期に実行します。

備考

スレッド対応環境で Init メソッドを呼び出してから、他の呼び出しを行う必要があります。終了したら、同様にスレッド対応環境で Fini メソッドを呼び出してください。

> **注記**
>
> このクラスは静的です。このクラスのインスタンスを作成しないでください。

このセクションの内容:

CreateDatabase(String^, String^) メソッド [112 ページ]
: 新しいデータベースを作成します。

CreateFileTransfer(String^, uint16, String^, String^) メソッド [113 ページ]
: FileTransfer オブジェクトを作成します。

DropDatabase(String^) メソッド [113 ページ]
: 現在実行していない既存のデータベースを消去します。

Fini() メソッド [114 ページ]
: Ultra Light ランタイムをファイナライズします。

GetLastError(ULSqlCode *, String^*) メソッド [114 ページ]
: 最後の呼び出しに関連付けられているエラー情報をこの DatabaseManager に返します。

Init() メソッド [115 ページ]
Ultra Light ランタイムを初期化します。

OpenConnection(String^) メソッド [115 ページ]
既存のデータベースへの新しい接続を開きます。

ValidateDatabase(String^, uint16, ValidateCallback^) メソッド [116 ページ]
データベースで低レベルのインデックス検証を実行します。

ValidateDatabaseAsync(String^, uint16flags) メソッド [117 ページ]
ValidateDatabase 操作を非同期に実行します。

1.5.1 CreateDatabase(String^, String^) メソッド

新しいデータベースを作成します。

構文

```
public static Connection CreateDatabase (connParms, createParms)
```

パラメータ

connParms セミコロンで区切った接続パラメータ文字列で、キーワード=値のペアで設定されます。接続文字列には、データベースの名前を含める必要があります。ここに含まれるパラメータは、データベースの接続時に指定されるパラメータセットと同じです。
createParms データベース作成パラメータをセミコロンで区切った文字列。キーワードと値のペアとして設定されます。例: page_size=2048;obfuscate=yes。

備考

2 セットのパラメータで指定される情報を使用してデータベースが作成されます。

connParms パラメータは、ファイル名や暗号化キーなど、データベースへのアクセス時に必ず適用される一連の標準接続パラメータです。

createParms パラメータは、チェックサムレベル、ページサイズ、照合、時刻と日付の形式など、データベースの作成時にのみ意味を持つ一連のパラメータです。

例

次のコードは、CreateDatabase メソッドを使用して、ファイル `mydb.udb` に Ultra Light データベースを作成する方法を示します。

```
String mypath = ApplicationData.Current.LocalFolder.Path;
            String connParms = "dbf=" + mypath + "¥¥mydb.udb";
            Connection conn = null;
            try {
            conn = DatabaseManager.CreateDatabase(connParms, "checksum_level=2");
            } catch( Exception ex ) {
            // report error
            }
```

1.5.2 CreateFileTransfer(String^, uint16, String^, String^) メソッド

FileTransfer オブジェクトを作成します。

構文

```
public static FileTransfer CreateFileTransfer (fileName, stream, userName,
version)
```

パラメータ

fileName 転送するサーバファイルの名前。このパラメータにパス情報を含めることはできません。
stream StreamType 列挙体に定義されている定数の 1 つ。通信ストリームタイプの識別に使用されます。
userName Mobile Link ユーザ名
version Mobile Link スクリプトのバージョン。

1.5.3 DropDatabase(String^) メソッド

現在実行していない既存のデータベースを消去します。

構文

```
public static void DropDatabase (parms)
```

パラメータ

parms データベース識別パラメータ (接続文字列)。

1.5.4 Fini() メソッド

Ultra Light ランタイムをファイナライズします。

構文

```
public static void Fini ()
```

備考

このメソッドは、アプリケーションの終了時に、単一のスレッドで 1 度だけ呼び出す必要があります。このメソッドはスレッド対応ではありません。

1.5.5 GetLastError(ULSqlCode *, String^*) メソッド

最後の呼び出しに関連付けられているエラー情報をこの DatabaseManager に返します。

構文

```
public static void GetLastError (errorCode, errorParms)
```

パラメータ

errorCode 返された Ultra Light SQL エラーコードを保持する出力パラメータ。
errorParms 返されたエラーパラメータの文字列、またはエラーパラメータが存在しない場合は空の文字列を保持する出力パラメータ。エラーパラメータはエラーメッセージにおける %n パラメータの値のカンマ区切りリストです。

備考

Connection::GetLastError を参照してください。

1.5.6 Init() メソッド

Ultra Light ランタイムを初期化します。

構文

```
public static bool Init ()
```

戻り値

成功した場合は true、失敗した場合は false。このメソッドは、メソッドが複数回呼び出された場合に false を返します。

備考

このメソッドは、その他の呼び出しを行う前に、単一のスレッドで 1 度だけ呼び出す必要があります。このメソッドはスレッド対応ではありません。

通常、メモリが使用可能であるかぎり、このメソッドは失敗しません。

1.5.7 OpenConnection(String^) メソッド

既存のデータベースへの新しい接続を開きます。

構文

```
public static Connection OpenConnection (connParms)
```

パラメータ

connParms 接続文字列。

備考

接続文字列は、どのデータベースに接続するかを示す、セミコロンで区切られた option=value 接続パラメータと、接続に使用するオプションのセットです。たとえば、暗号化キーを安全に取得した後に得られる接続文字列は、"DBF=mydb.udb;DBKEY=iyntTZId9OEa#&G" のようになります。

次は発生する可能性のあるエラーのリストです (GetLastError から取得できます)。

- SQLE_INVALID_PARSE_PARAMETER - connParms が正しくフォーマットされていません。
- SQLE_UNRECOGNIZED_OPTION - 接続オプション名のスペルを間違えた可能性があります。
- SQLE_INVALID_OPTION_VALUE - 接続オプション値が正しく指定されていません。
- SQLE_ULTRALITE_DATABASE_NOT_FOUND - 指定されたデータベースが見つかりませんでした。
- SQLE_INVALID_LOGON - 無効なユーザ ID または間違ったパスワードを入力しました。
- SQLE_TOO_MANY_CONNECTIONS - 同時データベース接続の最大数を超えました。

1.5.8 ValidateDatabase(String^, uint16, ValidateCallback^) メソッド

データベースで低レベルのインデックス検証を実行します。

構文

```
public static void ValidateDatabase (connParms, flags, cb)
```

パラメータ

connParms データベースへの接続に使用されるパラメータ。
flags 検証のタイプを制御するフラグ。次の例を参照してください。
cb 検証の進行状況の情報を受け取る関数。

備考

flags パラメータは、次のいずれかの値の組み合わせです。

- ULVF_TABLE
- ULVF_INDEX
- ULVF_DATABASE
- ULVF_EXPRESS または ULVF_FULL_VALIDATE

```
flags = ULVF_TABLE | ULVF_INDEX | ULVF_EXPRESS;
```

1.5.9 ValidateDatabaseAsync(String^, uint16flags) メソッド

ValidateDatabase 操作を非同期に実行します。

構文

```
public static IAsyncActionWithProgress< ValidateData^> ValidateDatabaseAsync
(connParms, )
```

パラメータ

connParms データベースへの接続に使用されるパラメータ。
flags 検証タイプの制御を行います。

戻り値

進捗レポートの待機可能な非同期アクション。

1.6 DatabaseSchema クラス

Ultra Light データベースのスキーマを表します。Connection.GetDatabaseSchema を使用して DatabaseSchema オブジェクトを取得します。

ネームスペース

```
UltraLite
```

構文

```
public ref class  sealed
```

メンバー

DatabaseSchema のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public void	Close() [118 ページ]	このオブジェクトを破棄します。
public uint8	GetPublicationCount() [119 ページ]	データベース内のパブリケーション数を取得します。
public IIterator< String^>	GetPublications() [119 ページ]	データベース内のすべてのパブリケーション (名前) の集まりに対する反復子を取得します。
public uint16	GetTableCount() [119 ページ]	データベース内のテーブルの数を返します。
public IIterator< TableSchema^>	GetTables() [120 ページ]	データベース内のすべてのテーブル (スキーマ) の集まりに対する反復子を取得します。
public TableSchema	GetTableSchema(String^) [120 ページ]	指定したテーブルのスキーマを返します。

このセクションの内容:

CloseObject() メソッド [118 ページ]
このオブジェクトを破棄します。

GetPublicationCount() メソッド [119 ページ]
データベース内のパブリケーション数を取得します。

GetPublications() メソッド [119 ページ]
データベース内のすべてのパブリケーション (名前) の集まりに対する反復子を取得します。

GetTableCount() メソッド [119 ページ]
データベース内のテーブルの数を返します。

GetTables() メソッド [120 ページ]
データベース内のすべてのテーブル (スキーマ) の集まりに対する反復子を取得します。

GetTableSchema(String^) メソッド [120 ページ]
指定したテーブルのスキーマを返します。

1.6.1 CloseObject() メソッド

このオブジェクトを破棄します。

構文

```
public void CloseObject ()
```

1.6.2 GetPublicationCount() メソッド

データベース内のパブリケーション数を取得します。

構文

```
public uint8 GetPublicationCount ()
```

戻り値

データベース内のパブリケーションの数です。

備考

パブリケーション ID の範囲は 1 から、このメソッドが返す数字までです。

1.6.3 GetPublications() メソッド

データベース内のすべてのパブリケーション (名前) の集まりに対する反復子を取得します。

構文

```
public IIterator< String^> GetPublications ()
```

戻り値

String オブジェクトの集まりに対する反復子。

1.6.4 GetTableCount() メソッド

データベース内のテーブルの数を返します。

構文

```
public uint16 GetTableCount ()
```

戻り値

テーブルの数を表す整数。

1.6.5 GetTables() メソッド

データベース内のすべてのテーブル (スキーマ) の集まりに対する反復子を取得します。

構文

```
public IIterator< TableSchema^> GetTables ()
```

戻り値

TableSchema オブジェクトの集まりに対する反復子。

1.6.6 GetTableSchema(String^) メソッド

指定したテーブルのスキーマを返します。

構文

```
public TableSchema GetTableSchema (tableName)
```

パラメータ

tableName テーブル名。

戻り値

特定のテーブルの場合は TableSchema オブジェクト。それ以外で、テーブルが存在しない場合は UL_NULL。

1.7 FileTransfer クラス

Mobile Link サーバを通じて、リモートデータベースからファイルを転送します。

ネームスペース

```
UltraLite
```

構文

```
public ref class  sealed
```

メンバー

FileTransfer のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public property String	FileName	ダウンロードするファイルの名前を指定します。
public property String	LocalPath	ファイルをダウンロードする場所を指定します。
public property String	LocalFileName	ダウンロードファイルのローカル名を指定します。
public property String	EncryptionKey	互換性のあるファイルの暗号化に使用する暗号化キー。 このプロパティは、今後の使用のために予約されています。このパラメータが NULL ではない場合、ダウンロードファイルは、暗号化と互換性のあるファイルである必要があり、デバイス上のファイルシステムに格納されるときに暗号化されます。このパラメータが NULL (デフォルト) の場合、ダウンロードファイルは通常どおり処理されます。
public property uint16	Stream	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。
public property String	StreamParms	同期ストリームの設定パラメータを指定します。

変更子とタイプ	変数	説明
public property String	UserName	Mobile Link サーバが Mobile Link クライアントを識別するユーザ名を指定します。
public property String	パスワード	UserName 値で指定されたユーザの Mobile Link パスワードを指定します。パスワードが設定されていないと、"" がデフォルトに使用されます。
public property String	RemoteKey	Mobile Link サーバが Mobile Link クライアントをユニークに識別するキーを指定します。
public property String	バージョン	使用する同期スクリプトを指定します。
public property bool	ResumePartialDownload	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。
public property Array< String^>	AuthenticationParms	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。

メソッド

変更子とタイプ	メソッド	説明
public bool	DownloadFile(FileTransferObserver^) [123 ページ]	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。
public IAsyncOperationWithProgress< bool, FileTransferStatus^>	DownloadFileAsync() [124 ページ]	DownloadFile 操作を非同期に実行します。
public FileTransferResult	GetResult() [124 ページ]	ファイル転送の結果を返します。
public bool	UploadFile(FileTransferObserver^) [124 ページ]	このオブジェクトのプロパティで指定されたファイルをアップロードします。
public IAsyncOperationWithProgress< bool, FileTransferStatus^>	UploadFileAsync() [125 ページ]	UploadFile 操作を非同期に実行します。

備考

ファイル転送を実行するデータベース接続は不要です。ファイルを転送するには、FileTransfer.FileName、FileTransfer.Stream、FileTransfer.UserName、FileTransfer.Version の値を設定します。DatabaseManager.CreateFileTransfer() を使用して FileTransfer オブジェクトを登録します。

このセクションの内容:

DownloadFile(FileTransferObserver^) メソッド [123 ページ]
FileTransfer.GetResult 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。

DownloadFileAsync() メソッド [124 ページ]
DownloadFile 操作を非同期に実行します。

GetResult() メソッド [124 ページ]

ファイル転送の結果を返します。

UploadFile(FileTransferObserver^) メソッド [124 ページ]
このオブジェクトのプロパティで指定されたファイルをアップロードします。

UploadFileAsync() メソッド [125 ページ]
UploadFile 操作を非同期に実行します。

1.7.1 DownloadFile(FileTransferObserver^) メソッド

FileTransfer.GetResult 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

```
public bool DownloadFile (observer)
```

パラメータ

observer ステータス更新の送信先となる observer コールバック。

戻り値

成功した場合は true、それ以外の場合は false (理由については FileTransferResult.streamErrorCode の値とその他のステータスに関するプロパティを確認)。

備考

FileTransfer.FileName 値で指定されたファイルが、Mobile Link サーバによって FileTransfer.LocalPath 値にダウンロードされます。このとき、FileTransfer.Stream、FileTransfer.UserName、FileTransfer.Password、および FileTransfer.Version の各値が使用されます。ファイルが破損することを防ぐため、Ultra Light はテンポラリファイルにダウンロードし、ダウンロードが完了したときのみローカルファイルを置換します。ダウンロードに影響を及ぼすその他のプロパティは、FileTransfer.LocalFileName、FileTransfer.AuthenticationParms、FileTransfer.ResumePartialDownload です。結果の詳細なステータスは、このオブジェクトの GetResult メソッドによってレポートされます。

1.7.2 DownloadFileAsync() メソッド

DownloadFile 操作を非同期に実行します。

構文

```
public IAsyncOperationWithProgress< bool, FileTransferStatus^> DownloadFileAsync
()
```

1.7.3 GetResult() メソッド

ファイル転送の結果を返します。

構文

```
public FileTransferResult GetResult ()
```

1.7.4 UploadFile(FileTransferObserver^) メソッド

このオブジェクトのプロパティで指定されたファイルをアップロードします。

構文

```
public bool UploadFile (observer)
```

パラメータ

observer ステータス更新の送信先となる observer コールバック。

戻り値

成功した場合は true、それ以外の場合は false (理由については FileTransfer.streamErrorCode の値とその他のステータスに関するプロパティを確認)。

備考

FileTransfer.FileName 値で指定されたファイルが、FileTransfer.LocalPath 値から Mobile Link サーバにアップロードされます。このとき、FileTransfer.Stream、FileTransfer.UserName、FileTransfer.Password、および FileTransfer.Version の各値が使用されます。結果の詳細なステータスは、このオブジェクトの GetResult メソッドによってレポートされます。

1.7.5 UploadFileAsync() メソッド

UploadFile 操作を非同期に実行します。

構文

```
public IAsyncOperationWithProgress< bool, FileTransferStatus^> UploadFileAsync ()
```

1.8 IndexSchema クラス

Ultra Light テーブルのインデックスのスキーマを表します。TableSchem.GetIndexSchema(indexName) を使用して IndexSchema オブジェクトを取得します。

ネームスペース

```
UltraLite
```

構文

```
public ref class  sealed
```

メンバー

IndexSchema のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public void	Close() [127 ページ]	このオブジェクトを破棄します。

変更子とタイプ	メソッド	説明
public uint16	GetColumnCount() [127 ページ]	インデックス内のカラム数を取得します。
public String	GetColumnName(uint16) [127 ページ]	インデックス内のカラムの位置を指定して、カラムの名前を取得します。
public uint16	GetIndexColumnID(String^) [128 ページ]	0 から始まるインデックスカラム ID を名前から取得します。
public uint16	GetIndexFlags() [128 ページ]	インデックスプロパティフラグのビットフィールドを取得します。
public String	GetName() [128 ページ]	インデックスの名前を取得します。
public String	GetReferencedIndexName() [129 ページ]	関連付けられているプライマリインデックスの名前を取得します。
public String	GetReferencedTableName() [129 ページ]	関連付けられているプライマリテーブルの名前を取得します。
public String	GetTableName() [130 ページ]	このインデックスが含まれるテーブルの名前を取得します。
public bool	IsColumnDescending(uint16) [130 ページ]	カラムが降順かどうかを調べます。

このセクションの内容:

CloseObject() メソッド [127 ページ]
: このオブジェクトを破棄します。

GetColumnCount() メソッド [127 ページ]
: インデックス内のカラム数を取得します。

GetColumnName(uint16) メソッド [127 ページ]
: インデックス内のカラムの位置を指定して、カラムの名前を取得します。

GetIndexColumnID(String^) メソッド [128 ページ]
: 0 から始まるインデックスカラム ID を名前から取得します。

GetIndexFlags() メソッド [128 ページ]
: インデックスプロパティフラグのビットフィールドを取得します。

GetName() メソッド [128 ページ]
: インデックスの名前を取得します。

GetReferencedIndexName() メソッド [129 ページ]
: 関連付けられているプライマリインデックスの名前を取得します。

GetReferencedTableName() メソッド [129 ページ]
: 関連付けられているプライマリテーブルの名前を取得します。

GetTableName() メソッド [130 ページ]
: このインデックスが含まれるテーブルの名前を取得します。

IsColumnDescending(uint16) メソッド [130 ページ]
: カラムが降順かどうかを調べます。

1.8.1 CloseObject() メソッド

このオブジェクトを破棄します。

構文

```
public void CloseObject ()
```

1.8.2 GetColumnCount() メソッド

インデックス内のカラム数を取得します。

構文

```
public uint16 GetColumnCount ()
```

戻り値

インデックス内のカラム数。

1.8.3 GetColumnName(uint16) メソッド

インデックス内のカラムの位置を指定して、カラムの名前を取得します。

構文

```
public String GetColumnName (col_id_in_index)
```

パラメータ

col_id_in_index インデックス内のカラムの位置を示す 0 から始まる順序数。

戻り値

カラム名。

1.8.4 GetIndexColumnID(String^) メソッド

0 から始まるインデックスカラム ID を名前から取得します。

構文

```
public uint16 GetIndexColumnID (columnName)
```

パラメータ

columnName カラムの名前。

戻り値

カラム ID。

1.8.5 GetIndexFlags() メソッド

インデックスプロパティフラグのビットフィールドを取得します。

構文

```
public uint16 GetIndexFlags ()
```

戻り値

IndexFlag

1.8.6 GetName() メソッド

インデックスの名前を取得します。

構文

```
public String GetName ()
```

戻り値

インデックスの名前。

1.8.7 GetReferencedIndexName() メソッド

関連付けられているプライマリインデックスの名前を取得します。

構文

```
public String GetReferencedIndexName ()
```

戻り値

参照先インデックスの名前。

備考

このメソッドは、外部キーにのみ適用されます。

1.8.8 GetReferencedTableName() メソッド

関連付けられているプライマリテーブルの名前を取得します。

構文

```
public String GetReferencedTableName ()
```

戻り値

参照先テーブルの名前。

備考

このメソッドは、外部キーにのみ適用されます。

1.8.9 GetTableName() メソッド

このインデックスが含まれるテーブルの名前を取得します。

構文

```
public String GetTableName ()
```

戻り値

このインデックスが含まれるテーブルの名前。

1.8.10 IsColumnDescending(uint16) メソッド

カラムが降順かどうかを調べます。

構文

```
public bool IsColumnDescending (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラムが降順の場合は true、昇順の場合は false。

1.9 PreparedStatement クラス

準備された SQL 文を表します。Connection.PrepareStatement は PreparedStatement を返します。

ネームスペース

```
UltraLite
```

構文

```
public ref class  sealed
```

メンバー

PreparedStatement のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public void	AppendParameterByteChunk(uint16, const Array< uint8 >^, int) [134 ページ]	複数のチャンクに分解される、サイズの大きい Binary パラメータを設定します。
public void	AppendParameterStringChunk(uint16, String^) [135 ページ]	複数のチャンクに分解される、サイズの大きい文字列パラメータを設定します。
public void	Close() [135 ページ]	このオブジェクトを破棄します。
public ResultSet	ExecuteQuery() [135 ページ]	SQL SELECT 文をクエリとして実行します。
public IAsyncOperation< ResultSet^>	ExecuteQueryAsync() [136 ページ]	ExecuteQuery 操作を非同期に実行します。
public void	ExecuteStatement() [136 ページ]	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。
public IAsyncAction	ExecuteStatementAsync() [136 ページ]	ExecuteStatement 操作を非同期に実行します。
public uint16	GetParameterCount() [137 ページ]	この文の入力パラメータの数を取得します。
public uint16	GetParameterID(String^) [137 ページ]	0 から始まるパラメータ名の順序数を取得します。
public uint16	GetParameterType(uint16) [138 ページ]	パラメータの記憶タイプまたはホスト変数の型を取得します。

変更子とタイプ	メソッド	説明
public String	GetPlan() [138 ページ]	クエリ実行プランのテキストベースの記述を取得します。
public ResultSetSchema	GetResultSetSchema() [139 ページ]	結果セットのスキーマを取得します。
public int	GetRowsAffectedCount() [139 ページ]	最後の文の影響を受けるローの数を取得します。
public bool	HasResultSet() [140 ページ]	SQL 文に結果セットがあるかどうかを調べます。
public void	SetParameterBool(uint16, bool) [140 ページ]	パラメータを boolean 値に設定します。
public void	SetParameterByte(uint16, uint8) [140 ページ]	パラメータを 8 ビット整数値に設定します。
public void	SetParameterBytes(uint16, const Array< uint8 >^, int) [141 ページ]	パラメータを byte 配列値に設定します。
public void	SetParameterDateTime(uint16, DateTime) [141 ページ]	パラメータを DateTime 値に設定します。
public void	SetParameterDouble(uint16, double) [142 ページ]	パラメータを double 値に設定します。
public void	SetParameterFloat(uint16, float) [142 ページ]	パラメータを float 値に設定します。
public void	SetParameterGuid(uint16, Guid) [142 ページ]	パラメータを GUID 値に設定します。
public void	SetParameterInt16(uint16, int16) [143 ページ]	パラメータを 16 ビット整数値に設定します。
public void	SetParameterInt32(uint16, int) [143 ページ]	パラメータを 32 ビット整数値に設定します。
public void	SetParameterInt64(uint16, int64) [144 ページ]	パラメータを 64 ビット整数値に設定します。
public void	SetParameterNull(uint16) [144 ページ]	パラメータを NULL に設定します。
public void	SetParameterString(uint16, String^) [144 ページ]	パラメータを文字列値に設定します。
public void	SetParameterUInt16(uint16, uint16) [145 ページ]	パラメータを符号なし 16 ビット整数値に設定します。
public void	SetParameterUInt32(uint16, unsigned int) [145 ページ]	パラメータを符号なし 32 ビット整数値に設定します。
public void	SetParameterUInt64(uint16, uint64) [146 ページ]	パラメータを符号なし 64 ビット整数値に設定します。

このセクションの内容:

AppendParameterByteChunk(uint16, const Array< uint8 >^, int) メソッド [134 ページ]
複数のチャンクに分解される、サイズの大きい Binary パラメータを設定します。

SetParameterInt16(uint16, int16) メソッド [143 ページ]
パラメータを 16 ビット整数値に設定します。

SetParameterInt32(uint16, int) メソッド [143 ページ]
パラメータを 32 ビット整数値に設定します。

SetParameterInt64(uint16, int64) メソッド [144 ページ]
パラメータを 64 ビット整数値に設定します。

SetParameterNull(uint16) メソッド [144 ページ]
パラメータを NULL に設定します。

SetParameterString(uint16, String^) メソッド [144 ページ]
パラメータを文字列値に設定します。

SetParameterUInt16(uint16, uint16) メソッド [145 ページ]
パラメータを符号なし 16 ビット整数値に設定します。

SetParameterUInt32(uint16, unsigned int) メソッド [145 ページ]
パラメータを符号なし 32 ビット整数値に設定します。

SetParameterUInt64(uint16, uint64) メソッド [146 ページ]
パラメータを符号なし 64 ビット整数値に設定します。

1.9.1 AppendParameterByteChunk(uint16, const Array< uint8 >^, int) メソッド

複数のチャンクに分解される、サイズの大きい Binary パラメータを設定します。

構文

```
public void AppendParameterByteChunk (pid, value, valueSize)
```

パラメータ

pid パラメータの、0 から始まる序数。
value 追加するバイトのチャンク。
valueSize バッファのサイズ。

1.9.2 AppendParameterStringChunk(uint16, String^) メソッド

複数のチャンクに分解される、サイズの大きい文字列パラメータを設定します。

構文

```
public void AppendParameterStringChunk (pid, value)
```

パラメータ

pid パラメータの、0 から始まる序数。
value 追加する文字列のチャンク。

1.9.3 CloseObject() メソッド

このオブジェクトを破棄します。

構文

```
public void CloseObject ()
```

1.9.4 ExecuteQuery() メソッド

結果セットを返す準備された文を実行します。

構文

```
public ResultSet ExecuteQuery ()
```

戻り値

クエリの結果 (ローのセット) を含む ResultSet オブジェクト。

1.9.5 ExecuteQueryAsync() メソッド

ExecuteQuery 操作を非同期に実行します。

構文

```
public IAsyncOperation< ResultSet^> ExecuteQueryAsync ()
```

戻り値

ResultSet の待機可能な非同期操作。

備考

結果セットを返す準備された文を実行します。

1.9.6 ExecuteStatement() 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準備された文を実行します。

構文

```
public void ExecuteStatement ()
```

1.9.7 ExecuteStatementAsync() メソッド

ExecuteStatement 操作を非同期に実行します。

構文

```
public IAsyncAction ExecuteStatementAsync ()
```

戻り値

待機可能な非同期アクション。

備考

SQL INSERT 文、DELETE 文、UPDATE 文のように、結果セットを返さない準備された文を実行します。

1.9.8 GetParameterCount() メソッド

準備した文の入力パラメータの数を取得します。

構文

```
public uint16 GetParameterCount ()
```

戻り値

準備した文の入力パラメータの数。

1.9.9 GetParameterID(String^) メソッド

0 から始まるパラメータ名の順序数を取得します。

構文

```
public uint16 GetParameterID (name)
```

パラメータ

name ホスト変数名。

戻り値

0 から始まるパラメータ名の順序数。

1.9.10 GetParameterType(uint16) メソッド

パラメータの記憶タイプまたはホスト変数の型を取得します。

構文

```
public uint16 GetParameterType (pid)
```

パラメータ

pid パラメータの、0 から始まる序数。

戻り値

指定したパラメータのタイプ。

1.9.11 GetPlan() メソッド

クエリ実行プランのテキストベースの記述を取得します。

構文

```
public String GetPlan ()
```

戻り値

プランテキスト

備考

このメソッドは、主に開発中の使用を目的とします。

プランがない場合は、空の文字列を返します。準備された文が SQL クエリの場合には、プランが存在します。

関連するクエリの実行前にプランが取得された場合は、クエリの実行に使用される操作がプランに表示されます。また、クエリの実行後にプランが取得された場合は、各操作で生成されるロー数も表示されます。このプランを使用して、クエリの実行に関する理解を深めることができます。

1.9.12 GetResultSetSchema() メソッド

結果セットのスキーマを取得します。

構文

```
public ResultSetSchema GetResultSetSchema ()
```

戻り値

結果セットのスキーマに関する情報を取得するために使用できる ResultSetSchema オブジェクト。

1.9.13 HasResultSet() メソッド

最後の文の影響を受けるローの数を取得します。

構文

```
public int GetRowsAffectedCount ()
```

戻り値

最後の文の影響を受けるローの数。この情報を使用できない場合 (たとえば、文によってデータではなくスキーマが変更される場合)、戻り値は -1 になります。

1.9.14 HasResultSet() メソッド

準備された SQL 文に結果セットがあるかどうかを調べます。

構文

```
public bool HasResultSet ()
```

戻り値

この文が実行されたときに結果セットが生成される場合は true。それ以外で、結果セットが生成されない場合は false。

1.9.15 SetParameterBool(uint16, bool) メソッド

パラメータを boolean 値に設定します。

構文

```
public void SetParameterBool (pid, value)
```

パラメータ

pid パラメータの、0 から始まる序数。
value boolean 値。

1.9.16 SetParameterByte(uint16, uint8) メソッド

パラメータを 8 ビット整数値に設定します。

構文

```
public void SetParameterByte (pid, value)
```

パラメータ

pid パラメータの、0 から始まる序数。
value 整数値。

1.9.17 SetParameterBytes(uint16, const Array< uint8 >^, int) メソッド

パラメータを byte 配列値に設定します。

構文

```
public void SetParameterBytes (pid, value, length)
```

パラメータ

pid パラメータの、0 から始まる序数。
value byte 配列値。
length byte 配列値から取得するバイト単位の長さ。

1.9.18 SetParameterDateTime(uint16, DateTime) メソッド

パラメータを DateTime 値に設定します。

構文

```
public void SetParameterDateTime (pid, value)
```

パラメータ

pid パラメータの、0 から始まる序数。
value DateTime 値。

1.9.19 SetParameterDouble(uint16, double) メソッド

パラメータを double 値に設定します。

構文

```
public void SetParameterDouble (pid, value)
```

パラメータ

pid パラメータの、0 から始まる序数。
value double 値。

1.9.20 SetParameterFloat(uint16, float) メソッド

パラメータを float 値に設定します。

構文

```
public void SetParameterFloat (pid, value)
```

パラメータ

pid パラメータの、0 から始まる序数。
value float 値。

1.9.21 SetParameterGuid(uint16, Guid) メソッド

パラメータを GUID 値に設定します。

構文

```
public void SetParameterGuid (pid, value)
```

パラメータ

pid パラメータの、0 から始まる序数。
value GUID 値。

1.9.22 SetParameterInt16(uint16, int16) メソッド

パラメータを 16 ビット整数値に設定します。

構文

```
public void SetParameterInt16 (pid, value)
```

パラメータ

pid パラメータの、0 から始まる序数。
value 整数値。

1.9.23 SetParameterInt32(uint16, int) メソッド

パラメータを 32 ビット整数値に設定します。

構文

```
public void SetParameterInt32 (pid, value)
```

パラメータ

pid パラメータの、0 から始まる序数。
value 整数値。

1.9.24 SetParameterInt64(uint16, int64) メソッド

パラメータを 64 ビット整数値に設定します。

構文

```
public void SetParameterInt64 (pid, value)
```

パラメータ

pid パラメータの、0 から始まる序数。
value 整数値。

1.9.25 SetParameterNull(uint16) メソッド

パラメータを NULL に設定します。

構文

```
public void SetParameterNull (pid)
```

パラメータ

pid パラメータの、0 から始まる序数。

1.9.26 SetParameterString(uint16, String^) メソッド

パラメータを文字列値に設定します。

構文

```
public void SetParameterString (pid, value)
```

パラメータ

pid パラメータの、0 から始まる序数。
value 文字列値。このパラメータの長さが 32 KB を超える場合は、SQLE_INVALID_PARAMETER に設定されます。サイズの大きい文字列の場合は、代わりに AppendParameterStringChunk メソッドが呼び出されます。

1.9.27 SetParameterUInt16(uint16, uint16) メソッド

パラメータを符号なし 16 ビット整数値に設定します。

構文

```
public void SetParameterUInt16 (pid, value)
```

パラメータ

pid パラメータの、0 から始まる序数。
value 整数値。

1.9.28 SetParameterUInt32(uint16, unsigned int) メソッド

パラメータを符号なし 32 ビット整数値に設定します。

構文

```
public void SetParameterUInt32 (pid, value)
```

パラメータ

pid パラメータの、0 から始まる序数。
value 整数値。

1.9.29 SetParameterUInt64(uint16, uint64) メソッド

パラメータを符号なし 64 ビット整数値に設定します。

構文

```
public void SetParameterUInt64 (pid, value)
```

パラメータ

pid パラメータの、0 から始まる序数。
value 整数値。

1.10 ResultSet クラス

Ultra Light データベースの結果セットを表します。ResultSet オブジェクトは更新および削除に使用され、PreparedStatement.ExecuteQuery() によって返されます。

ネームスペース

```
UltraLite
```

構文

```
public ref class  sealed  : Cursor
```

メンバー

ResultSet のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public virtual void	AfterLast() [151 ページ]	カーソルを最後のローの後に移動します。

変更子とタイプ	メソッド	説明
public virtual void	AppendBytes [151 ページ]	バイトをカラムに追加します。
public virtual void	AppendChars [153 ページ]	文字列のチャンクをカラムに追加します。
public virtual void	BeforeFirst() [155 ページ]	カーソルを最初のローの前に移動します。
public virtual void	Close() [155 ページ]	このオブジェクトを破棄します。
public virtual void	Delete() [155 ページ]	現在のローを削除し、カーソルを次の有効なローに移動します。
public virtual void	DeleteNamed(String^) [156 ページ]	現在のローを削除し、カーソルを次の有効なローに移動します。
public virtual void	First() [156 ページ]	カーソルを最初のローに移動します。
public virtual int64	GetBinaryLength [156 ページ]	カラムの値のバイナリ長さを取得します。
public virtual bool	GetBool [158 ページ]	カラムから値を boolean としてフェッチします。
public virtual uint8	GetByte [159 ページ]	カラムから値を byte としてフェッチします。
public virtual int64	GetBytes [161 ページ]	カラムからバイナリチャンクを取得します。
public virtual int64	GetChars [163 ページ]	カラムからワイド文字列のチャンクを取得します。
public virtual DateTime	GetDateTime [165 ページ]	カラムから値を DateTime としてフェッチします。
public virtual double	GetDouble [167 ページ]	カラムから値を double としてフェッチします。
public virtual float	GetFloat [168 ページ]	カラムから値を float としてフェッチします。
public virtual Guid	GetGuid [170 ページ]	カラムから値を GUID としてフェッチします。
public virtual int16	GetInt16 [171 ページ]	カラムから値を 16 ビット整数としてフェッチします。
public virtual int	GetInt32 [173 ページ]	カラムから値を整数としてフェッチします。
public virtual int64	GetInt64 [174 ページ]	カラムから値を 64 ビット整数としてフェッチします。
public ResultSetSchema	GetResultSetSchema() [176 ページ]	結果セットに関する情報を取得するために使用できるオブジェクトを返します。
public virtual unsigned int	GetRowCount(unsigned int) [176 ページ]	テーブルのローの数を取得します。
public virtual uint8	GetState() [177 ページ]	カーソルの内部ステータスを取得します。
public virtual String	GetString [177 ページ]	カラムから値を文字列としてフェッチします。
public virtual int64	GetStringLength [179 ページ]	カラムの値の文字列長さを取得します。
public virtual uint16	GetUInt16 [180 ページ]	カラムから値を 16 ビット符号なし整数としてフェッチします。
public virtual unsigned int	GetUInt32 [182 ページ]	カラムから値を 32 ビット符号なし整数としてフェッチします。

変更子とタイプ	メソッド	説明
public virtual uint64	GetUInt64 [183 ページ]	カラムから値を 64 ビット符号なし整数としてフェッチします。
public virtual bool	IsNull [185 ページ]	カラム値が NULL かどうかをチェックします。
public virtual void	Last() [186 ページ]	カーソルを最後のローに移動します。
public virtual bool	Next() [186 ページ]	カーソルをロー 1 つ分進めます。
public virtual bool	Previous() [187 ページ]	カーソルをロー 1 つ分戻します。
public virtual void	Relative(int) [187 ページ]	カーソルを、現在のカーソルの位置から、offset で指定したロー数分移動します。
public virtual void	SetBool [188 ページ]	カラムを boolean 値に設定します。
public virtual void	SetByte [189 ページ]	カラムを byte 値に設定します。
public virtual void	SetBytes [191 ページ]	カラムを binary 値に設定します。
public virtual void	SetDateTime [192 ページ]	カラムを DateTime 値に設定します。
public virtual void	SetDefault [194 ページ]	カラムをそのデフォルト値に設定します。
public virtual void	SetDouble [195 ページ]	カラムを double 値に設定します。
public virtual void	SetFloat [196 ページ]	カラムを float 値に設定します。
public virtual void	SetGuid [198 ページ]	カラムを GUID 値に設定します。
public virtual void	SetInt16 [199 ページ]	カラムを整数値に設定します。
public virtual void	SetInt32 [201 ページ]	カラムを整数値に設定します。
public virtual void	SetInt64 [202 ページ]	カラムを整数値に設定します。
public virtual void	SetNull [204 ページ]	カラムを NULL に設定します。
public virtual void	SetString [205 ページ]	カラムを文字列値に設定します。
public virtual void	SetUInt16 [206 ページ]	カラムを符号なし 16 ビット整数値に設定します。
public virtual void	SetUInt32 [208 ページ]	カラムを符号なし 32 ビット整数値に設定します。
public virtual void	SetUInt64 [209 ページ]	カラムを符号なし 64 ビット整数値に設定します。
public virtual void	Update() [210 ページ]	現在の行を更新します。
public virtual void	UpdateBegin() [211 ページ]	カラムの設定に使用される更新モードを選択します。

このセクションの内容:

AfterLast() メソッド [151 ページ]
カーソルを最後のローの後に移動します。

AppendBytes メソッド [151 ページ]
バイトをカラムに追加します。

AppendChars メソッド [153 ページ]
文字列のチャンクをカラムに追加します。

BeforeFirst() メソッド [155 ページ]
カーソルを最初のローの前に移動します。

CloseObject() メソッド [155 ページ]
このオブジェクトを破棄します。

Delete() メソッド [155 ページ]
現在のローを削除し、カーソルを次の有効なローに移動します。

DeleteNamed(String^) メソッド [156 ページ]
現在のローを削除し、カーソルを次の有効なローに移動します。

First() メソッド [156 ページ]
カーソルを最初のローに移動します。

GetBinaryLength メソッド [156 ページ]
カラムの値のバイナリ長さを取得します。

GetBool メソッド [158 ページ]
カラムから値を boolean としてフェッチします。

GetByte メソッド [159 ページ]
カラムから値を byte としてフェッチします。

GetBytes メソッド [161 ページ]
カラムからバイナリチャンクを取得します。

GetChars メソッド [163 ページ]
カラムからワイド文字列のチャンクを取得します。

GetDateTime メソッド [165 ページ]
カラムから値を DateTime としてフェッチします。

GetDouble メソッド [167 ページ]
カラムから値を double としてフェッチします。

GetFloat メソッド [168 ページ]
カラムから値を float としてフェッチします。

GetGuid メソッド [170 ページ]
カラムから値を GUID としてフェッチします。

GetInt16 メソッド [171 ページ]
カラムから値を 16 ビット整数としてフェッチします。

GetInt32 メソッド [173 ページ]
カラムから値を整数としてフェッチします。

GetInt64 メソッド [174 ページ]
カラムから値を 64 ビット整数としてフェッチします。

GetResultSetSchema() メソッド [176 ページ]
結果セットに関する情報を取得するために使用できるオブジェクトを返します。

GetRowCount(unsigned int) メソッド [176 ページ]
テーブルのローの数を取得します。

GetState() メソッド [177 ページ]
カーソルの内部ステータスを取得します。

GetString メソッド [177 ページ]
カラムから値を文字列としてフェッチします。

GetStringLength メソッド [179 ページ]
カラムの値の文字列長さを取得します。

GetUInt16 メソッド [180 ページ]
カラムから値を 16 ビット符号なし整数としてフェッチします。

GetUInt32 メソッド [182 ページ]
カラムから値を 32 ビット符号なし整数としてフェッチします。

GetUInt64 メソッド [183 ページ]
カラムから値を 64 ビット符号なし整数としてフェッチします。

IsNull メソッド [185 ページ]
カラム値が NULL かどうかをチェックします。

Last() メソッド [186 ページ]
カーソルを最後のローに移動します。

Next() メソッド [186 ページ]
カーソルをロー 1 つ分進めます。

Previous() メソッド [187 ページ]
カーソルをロー 1 つ分戻します。

Relative(int) メソッド [187 ページ]
カーソルを、現在のカーソルの位置から、offset で指定したロー数分移動します。

SetBool メソッド [188 ページ]
カラムを boolean 値に設定します。

SetByte メソッド [189 ページ]
カラムを byte 値に設定します。

SetBytes メソッド [191 ページ]
カラムを binary 値に設定します。

SetDateTime メソッド [192 ページ]
カラムを DateTime 値に設定します。

SetDefault メソッド [194 ページ]
カラムをそのデフォルト値に設定します。

SetDouble メソッド [195 ページ]
カラムを double 値に設定します。

SetFloat メソッド [196 ページ]
カラムを float 値に設定します。

SetGuid メソッド [198 ページ]
カラムを GUID 値に設定します。

SetInt16 メソッド [199 ページ]
カラムを整数値に設定します。

SetInt32 メソッド [201 ページ]
カラムを整数値に設定します。

SetInt64 メソッド [202 ページ]
カラムを整数値に設定します。

SetNull メソッド [204 ページ]
カラムを NULL に設定します。

SetString メソッド [205 ページ]
カラムを文字列値に設定します。

SetUInt16 メソッド [206 ページ]
カラムを符号なし 16 ビット整数値に設定します。

SetUInt32 メソッド [208 ページ]
カラムを符号なし 32 ビット整数値に設定します。

SetUInt64 メソッド [209 ページ]
カラムを符号なし 64 ビット整数値に設定します。

Update() メソッド [210 ページ]
現在の行を更新します。

UpdateBegin() メソッド [211 ページ]
カラムの設定に使用される更新モードを選択します。

1.10.1 AfterLast() メソッド

カーソルを最後のローの後に移動します。

構文

```
public virtual void AfterLast ()
```

1.10.2 AppendBytes メソッド

バイトをカラムに追加します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	AppendBytes(String^, const Array< uint8 >^, int, int) [152 ページ]	バイトをカラムに追加します。

変更子とタイプ	オーバーロード名	説明
public virtual void	AppendBytes(uint16, const Array< uint8 >^, int, int) [153 ページ]	バイトをカラムに追加します。

このセクションの内容:

AppendBytes(String^, const Array< uint8 >^, int, int) メソッド [152 ページ]
バイトをカラムに追加します。

AppendBytes(uint16, const Array< uint8 >^, int, int) メソッド [153 ページ]
バイトをカラムに追加します。

1.10.2.1 AppendBytes(String^, const Array< uint8 >^, int, int) メソッド

バイトをカラムに追加します。

構文

```
public virtual void AppendBytes (cname, value, offset, size)
```

パラメータ

cname カラム名。
value 追加するバイトのチャンク。
offset バイトのチャンク内でのオフセット (開始位置)。
size バイトのチャンクのサイズ (バイト単位)。

備考

AppendBytes メソッド呼び出しにより現時点で書き込み済みのカラムの末尾に、指定のバイトが追加されます。

1.10.2.2 AppendBytes(uint16, const Array< uint8 >^, int, int) メソッド

バイトをカラムに追加します。

構文

```
public virtual void AppendBytes (cid, value, offset, size)
```

パラメータ

cid 0 から始まるカラムの序数。
value 追加するバイトのチャンク。
offset バイトのチャンク内でのオフセット (開始位置)。
size バイトのチャンクのサイズ (バイト単位)。

備考

AppendBytes メソッド呼び出しにより現時点で書き込み済みのカラムの末尾に、指定のバイトが追加されます。

1.10.3 AppendChars メソッド

文字列のチャンクをカラムに追加します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	AppendChars(String^, const Array< wchar_t >^, int, int) [154 ページ]	文字列のチャンクをカラムに追加します。
public virtual void	AppendChars(uint16, const Array< wchar_t >^, int, int) [154 ページ]	文字列のチャンクをカラムに追加します。

このセクションの内容:

AppendChars(String^, const Array< wchar_t >^, int, int) メソッド [154 ページ]
文字列のチャンクをカラムに追加します。

AppendChars(uint16, const Array< wchar_t >^, int, int) メソッド [154 ページ]
文字列のチャンクをカラムに追加します。

1.10.3.1 AppendChars(String^, const Array< wchar_t >^, int, int) メソッド

文字列のチャンクをカラムに追加します。

構文

```
public virtual void AppendChars (cname, value, offset, size)
```

パラメータ

cname カラム名。
value 追加する文字列のチャンク。
offset 文字列のチャンク内でのオフセット (開始位置)。
size 文字列のチャンクの文字単位での長さです。

備考

このメソッドは、AppendChars メソッド呼び出しにより現時点で書き込み済みの文字列の末尾に、指定の文字列を追加します。

1.10.3.2 AppendChars(uint16, const Array< wchar_t >^, int, int) メソッド

文字列のチャンクをカラムに追加します。

構文

```
public virtual void AppendChars (cid, value, offset, size)
```

パラメータ

cid 0 から始まるカラムの序数。
value 追加する文字列のチャンク。
offset 文字列のチャンク内でのオフセット (開始位置)。
size 文字列のチャンクの文字単位での長さです。

備考

このメソッドは、AppendChars メソッド呼び出しにより現時点で書き込み済みの文字列の末尾に、指定の文字列を追加します。

1.10.4 BeforeFirst() メソッド

カーソルを最初のローの前に移動します。

構文

```
public virtual void BeforeFirst ()
```

1.10.5 CloseObject() メソッド

このオブジェクトを破棄します。

構文

```
public virtual void CloseObject ()
```

1.10.6 Delete() メソッド

現在のローを削除し、カーソルを次の有効なローに移動します。

構文

```
public virtual void Delete ()
```

1.10.7 DeleteNamed(String^) メソッド

現在のローを削除し、カーソルを次の有効なローに移動します。

構文

```
public virtual void DeleteNamed (tableName)
```

パラメータ

tableName テーブル名またはその相関 (同じテーブル名を共有する複数のカラムがデータベースに存在する場合に必要)。

1.10.8 First() メソッド

カーソルを最初のローに移動します。

構文

```
public virtual void First ()
```

1.10.9 GetBinaryLength メソッド

カラムの値のバイナリ長さを取得します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual int64	GetBinaryLength(String^) [157 ページ]	カラムの値のバイナリ長さを取得します。
public virtual int64	GetBinaryLength(uint16) [157 ページ]	カラムの値のバイナリ長さを取得します。

このセクションの内容:

GetBinaryLength(String^) メソッド [157 ページ]
カラムの値のバイナリ長さを取得します。

GetBinaryLength(uint16) メソッド [157 ページ]

カラムの値のバイナリ長さを取得します。

1.10.9.1 GetBinaryLength(String^) メソッド

カラムの値のバイナリ長さを取得します。

構文

```
public virtual int64 GetBinaryLength (cname)
```

パラメータ

cname カラム名。

戻り値

binary としてのカラムの値の長さ。

1.10.9.2 GetBinaryLength(uint16) メソッド

カラムの値のバイナリ長さを取得します。

構文

```
public virtual int64 GetBinaryLength (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

binary としてのカラムの値の長さ。

1.10.10 GetBool メソッド

カラムから値を boolean としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual bool	GetBool(String^) [158 ページ]	カラムから値を boolean としてフェッチします。
public virtual bool	GetBool(uint16) [159 ページ]	カラムから値を boolean としてフェッチします。

このセクションの内容:

GetBool(String^) メソッド [158 ページ]
カラムから値を boolean としてフェッチします。

GetBool(uint16) メソッド [159 ページ]
カラムから値を boolean としてフェッチします。

1.10.10.1 GetBool(String^) メソッド

カラムから値を boolean としてフェッチします。

構文

```
public virtual bool GetBool (cname)
```

パラメータ

cname カラム名。

戻り値

boolean としてのカラム値。

1.10.10.2 GetBool(uint16) メソッド

カラムから値を boolean としてフェッチします。

構文

```
public virtual bool GetBool (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

boolean としてのカラム値。

1.10.11 GetByte メソッド

カラムから値を byte としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual uint8	GetByte(String^) [160 ページ]	カラムから値を byte としてフェッチします。
public virtual uint8	GetByte(uint16) [160 ページ]	カラムから値を byte としてフェッチします。

このセクションの内容:

GetByte(String^) メソッド [160 ページ]
カラムから値を byte としてフェッチします。

GetByte(uint16) メソッド [160 ページ]
カラムから値を byte としてフェッチします。

1.10.11.1 GetByte(String^) メソッド

カラムから値を byte としてフェッチします。

構文

```
public virtual uint8 GetByte (cname)
```

パラメータ

cname カラム名。

戻り値

byte としてのカラム値。

1.10.11.2 GetByte(uint16) メソッド

カラムから値を byte としてフェッチします。

構文

```
public virtual uint8 GetByte (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

byte としてのカラム値。

1.10.12 GetBytes メソッド

カラムからバイナリチャンクを取得します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual int64	GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) [161 ページ]	カラムからバイナリチャンクを取得します。
public virtual int64	GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) [162 ページ]	カラムからバイナリチャンクを取得します。

このセクションの内容:

GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) メソッド [161 ページ]
カラムからバイナリチャンクを取得します。

GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) メソッド [162 ページ]
カラムからバイナリチャンクを取得します。

1.10.12.1 GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) メソッド

カラムからバイナリチャンクを取得します。

構文

```
public virtual int64 GetBytes (cname, srcOffset, dst, dstOffset, count)
```

パラメータ

cname カラム名。
dst バイトを保持するバッファ。
count バッファのサイズ (バイト単位)。
srcOffset 値内でのオフセット (読み込み開始位置)、または前回の読み込みが終了したところから続行する場合は BLOB_CONTINUE 定数。
dstOffset 宛先のバッファ内でのオフセット (バイトのコピー位置)。

戻り値

宛先のバッファにコピーされたバイト数。dst 値が NULL の場合は、残りのバイト数が返されます。カラムが NULL のときは、dst パラメータに空の文字列が返されます。

備考

0 が返された場合は、値の最後に到達しました。

IsNull メソッドを使用して、NULL と空の文字列を区別してください。

1.10.12.2 GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) メソッド

カラムからバイナリチャンクを取得します。

構文

```
public virtual int64 GetBytes (cid, srcOffset, dst, dstOffset, count)
```

パラメータ

cid 0 から始まるカラムの序数。
dst バイトを保持するバッファ。
count バッファのサイズ (バイト単位)。
srcOffset 値内でのオフセット (読み込み開始位置)、または前回の読み込みが終了したところから続行する場合は BLOB_CONTINUE 定数。
dstOffset 宛先のバッファ内でのオフセット (バイトのコピー位置)。

戻り値

宛先のバッファにコピーされたバイト数。dst 値が NULL の場合、このメソッドは残りのバイト数を返します。カラムが NULL のときは、dst パラメータに空の文字列が返されます。

備考

0 が返された場合は、値の最後に到達しました。

IsNull メソッドを使用して、NULL と空の文字列を区別してください。

1.10.13 GetChars メソッド

カラムからワイド文字列のチャンクを取得します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual int64	GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) [163 ページ]	カラムからワイド文字列のチャンクを取得します。
public virtual int64	GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) [164 ページ]	カラムからワイド文字列のチャンクを取得します。

このセクションの内容:

GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) メソッド [163 ページ]
カラムからワイド文字列のチャンクを取得します。

GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) メソッド [164 ページ]
カラムからワイド文字列のチャンクを取得します。

1.10.13.1 GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) メソッド

カラムからワイド文字列のチャンクを取得します。

構文

```
public virtual int64 GetChars (cname, srcOffset, dst, dstOffset, count)
```

パラメータ

cname カラム名。
dst 文字列のチャンクを保持するバッファ。文字列は、トランケートされても NULL で終了します。
count バッファのサイズ (文字数)。
srcOffset 値内でのオフセット (読み込み開始位置)、または前回の読み込みが終了したところから続行する場合は BLOB_CONTINUE 定数。
dstOffset 宛先のバッファ内でのオフセット (文字のコピー位置)。

戻り値

宛先のバッファにコピーされた文字数 (NULL ターミネータを含まない)。dst 値が NULL の場合、このメソッドは文字列の残りの文字数を返します。カラムが NULL のときは、dst パラメータに空の文字列が返されます。

備考

0 が返された場合は、値の最後に到達しました。

IsNull メソッドを使用して、NULL と空の文字列を区別してください。

1.10.13.2 GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) メソッド

カラムからワイド文字列のチャンクを取得します。

構文

```
public virtual int64 GetChars (cid, srcOffset, dst, dstOffset, count)
```

パラメータ

cid 0 から始まるカラムの序数。
dst 文字列のチャンクを保持するバッファ。文字列は、トランケートされても NULL で終了します。
count バッファのサイズ (文字数)。
srcOffset 値内でのオフセット (読み込み開始位置)、または前回の読み込みが終了したところから続行する場合は BLOB_CONTINUE 定数。
dstOffset 宛先のバッファ内でのオフセット (文字のコピー位置)。

戻り値

宛先のバッファにコピーされた文字数 (NULL ターミネータを含まない)。dst 値が NULL の場合、このメソッドは文字列の残りの文字数を返します。カラムが NULL のときは、dst パラメータに空の文字列が返されます。

備考

0 が返された場合は、値の最後に到達しました。

IsNull メソッドを使用して、NULL と空の文字列を区別してください。

1.10.14 GetDateTime メソッド

カラムから値を DateTime としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual DateTime	GetDateTime(String^) [165 ページ]	カラムから値を DateTime としてフェッチします。
public virtual DateTime	GetDateTime(uint16) [166 ページ]	カラムから値を DateTime としてフェッチします。

このセクションの内容:

GetDateTime(String^) メソッド [165 ページ]
カラムから値を DateTime としてフェッチします。

GetDateTime(uint16) メソッド [166 ページ]
カラムから値を DateTime としてフェッチします。

1.10.14.1 GetDateTime(String^) メソッド

カラムから値を DateTime としてフェッチします。

構文

```
public virtual DateTime GetDateTime (cname)
```

パラメータ

cname カラム名。

戻り値

DateTime 値。

1.10.14.2 GetDateTime(uint16) メソッド

カラムから値を DateTime としてフェッチします。

構文

```
public virtual DateTime GetDateTime (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

DateTime 値。

1.10.15 GetDouble メソッド

カラムから値を double としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual double	GetDouble(String^) [167 ページ]	カラムから値を double としてフェッチします。
public virtual double	GetDouble(uint16) [168 ページ]	カラムから値を double としてフェッチします。

このセクションの内容:

GetDouble(String^) メソッド [167 ページ]
カラムから値を double としてフェッチします。

GetDouble(uint16) メソッド [168 ページ]
カラムから値を double としてフェッチします。

1.10.15.1 GetDouble(String^) メソッド

カラムから値を double としてフェッチします。

構文

```
public virtual double GetDouble (cname)
```

パラメータ

cname カラム名。

戻り値

double としてのカラム値。

1.10.15.2 GetDouble(uint16) メソッド

カラムから値を double としてフェッチします。

構文

```
public virtual double GetDouble (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

double としてのカラム値。

1.10.16 GetFloat メソッド

カラムから値を float としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual float	GetFloat(String^) [169 ページ]	カラムから値を float としてフェッチします。
public virtual float	GetFloat(uint16) [169 ページ]	カラムから値を float としてフェッチします。

このセクションの内容:

GetFloat(String^) メソッド [169 ページ]
カラムから値を float としてフェッチします。

GetFloat(uint16) メソッド [169 ページ]
カラムから値を float としてフェッチします。

1.10.16.1 GetFloat(String^) メソッド

カラムから値を float としてフェッチします。

構文

```
public virtual float GetFloat (cname)
```

パラメータ

cname カラム名。

戻り値

float としてのカラム値。

1.10.16.2 GetFloat(uint16) メソッド

カラムから値を float としてフェッチします。

構文

```
public virtual float GetFloat (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

float としてのカラム値。

1.10.17 GetGuid メソッド

カラムから値を GUID としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual Guid	GetGuid(String^) [170 ページ]	カラムから値を GUID としてフェッチします。
public virtual Guid	GetGuid(uint16) [171 ページ]	カラムから値を GUID としてフェッチします。

このセクションの内容:

GetGuid(String^) メソッド [170 ページ]
カラムから値を GUID としてフェッチします。

GetGuid(uint16) メソッド [171 ページ]
カラムから値を GUID としてフェッチします。

1.10.17.1 GetGuid(String^) メソッド

カラムから値を GUID としてフェッチします。

構文

```
public virtual Guid GetGuid (cname)
```

パラメータ

cname カラム名。

戻り値

GUID 値。

1.10.17.2 GetGuid(uint16) メソッド

カラムから値を GUID としてフェッチします。

構文

```
public virtual Guid GetGuid (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

GUID 値。

1.10.18 GetInt16 メソッド

カラムから値を 16 ビット整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual int16	GetInt16(String^) [172 ページ]	カラムから値を 16 ビット整数としてフェッチします。
public virtual int16	GetInt16(uint16) [172 ページ]	カラムから値を 16 ビット整数としてフェッチします。

このセクションの内容:

GetInt16(String^) メソッド [172 ページ]
カラムから値を 16 ビット整数としてフェッチします。

GetInt16(uint16) メソッド [172 ページ]
カラムから値を 16 ビット整数としてフェッチします。

1.10.18.1 GetInt16(String^) メソッド

カラムから値を 16 ビット整数としてフェッチします。

構文

```
public virtual int16 GetInt16 (cname)
```

パラメータ

cname カラム名。

戻り値

整数としてのカラム値。

1.10.18.2 GetInt16(uint16) メソッド

カラムから値を 16 ビット整数としてフェッチします。

構文

```
public virtual int16 GetInt16 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

整数としてのカラム値。

1.10.19 GetInt32 メソッド

カラムから値を整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual int	GetInt32(String^) [173 ページ]	カラムから値を整数としてフェッチします。
public virtual int	GetInt32(uint16) [174 ページ]	カラムから値を整数としてフェッチします。

このセクションの内容:

GetInt32(String^) メソッド [173 ページ]
カラムから値を整数としてフェッチします。

GetInt32(uint16) メソッド [174 ページ]
カラムから値を整数としてフェッチします。

1.10.19.1 GetInt32(String^) メソッド

カラムから値を整数としてフェッチします。

構文

```
public virtual int GetInt32 (cname)
```

パラメータ

cname カラム名。

戻り値

整数としてのカラム値。

1.10.19.2 GetInt32(uint16) メソッド

カラムから値を整数としてフェッチします。

構文

```
public virtual int GetInt32 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

整数としてのカラム値。

1.10.20 GetInt64 メソッド

カラムから値を 64 ビット整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual int64	GetInt64(String^) [175 ページ]	カラムから値を 64 ビット整数としてフェッチします。
public virtual int64	GetInt64(uint16) [175 ページ]	カラムから値を 64 ビット整数としてフェッチします。

このセクションの内容:

GetInt64(String^) メソッド [175 ページ]
カラムから値を 64 ビット整数としてフェッチします。

GetInt64(uint16) メソッド [175 ページ]
カラムから値を 64 ビット整数としてフェッチします。

1.10.20.1 GetInt64(String^) メソッド

カラムから値を 64 ビット整数としてフェッチします。

構文

```
public virtual int64 GetInt64 (cname)
```

パラメータ

cname カラム名。

戻り値

整数としてのカラム値。

1.10.20.2 GetInt64(uint16) メソッド

カラムから値を 64 ビット整数としてフェッチします。

構文

```
public virtual int64 GetInt64 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

整数としてのカラム値。

1.10.21 GetResultSetSchema() メソッド

結果セットに関する情報を取得するために使用できるオブジェクトを返します。

構文

```
public ResultSetSchema GetResultSetSchema ()
```

戻り値

結果セットに関する情報を取得するために使用できる ResultSetSchema オブジェクト。

1.10.22 GetRowCount(unsigned int) メソッド

テーブルのローの数を取得します。

構文

```
public virtual unsigned int GetRowCount (threshold)
```

パラメータ

threshold カウントするローの数の制限。無限を表すには 0 を設定します。

戻り値

テーブル内のローの数。

備考

このメソッドは、次の文を実行することと同義です。

```
SELECT COUNT(*) FROM table
```

1.10.23 GetState() メソッド

カーソルの内部ステータスを取得します。

構文

```
public virtual uint8 GetState ()
```

戻り値

カーソルのステータス

1.10.24 GetString メソッド

カラムから値を文字列としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual String	GetString(String^) [177 ページ]	カラムから値を文字列としてフェッチします。
public virtual String	GetString(uint16) [178 ページ]	カラムから値を文字列としてフェッチします。

このセクションの内容:

GetString(String^) メソッド [177 ページ]
カラムから値を文字列としてフェッチします。

GetString(uint16) メソッド [178 ページ]
カラムから値を文字列としてフェッチします。

1.10.24.1 GetString(String^) メソッド

カラムから値を文字列としてフェッチします。

構文

```
public virtual String GetString (cname)
```

パラメータ

cname カラム名。

戻り値

文字列値。

1.10.24.2 GetString(uint16) メソッド

カラムから値を文字列としてフェッチします。

構文

```
public virtual String GetString (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

文字列値。

1.10.25 GetStringLength メソッド

カラムの値の文字列長さを取得します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual int64	GetStringLength(String^) [179 ページ]	カラムの値の文字列長さを取得します。
public virtual int64	GetStringLength(uint16) [180 ページ]	カラムの値の文字列長さを取得します。

このセクションの内容:

GetStringLength(String^) メソッド [179 ページ]
カラムの値の文字列長さを取得します。

GetStringLength(uint16) メソッド [180 ページ]
カラムの値の文字列長さを取得します。

1.10.25.1 GetStringLength(String^) メソッド

カラムの値の文字列長さを取得します。

構文

```
public virtual int64 GetStringLength (cname)
```

パラメータ

cname カラム名。

戻り値

文字列型のカラムの値の文字数。

1.10.25.2 GetStringLength(uint16) メソッド

カラムの値の文字列長さを取得します。

構文

```
public virtual int64 GetStringLength (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

文字列型のカラムの値の文字数。

1.10.26 GetUInt16 メソッド

カラムから値を 16 ビット符号なし整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual uint16	GetUInt16(String^) [181 ページ]	カラムから値を 16 ビット符号なし整数としてフェッチします。
public virtual uint16	GetUInt16(uint16) [181 ページ]	カラムから値を 16 ビット符号なし整数としてフェッチします。

このセクションの内容:

GetUInt16(String^) メソッド [181 ページ]
カラムから値を 16 ビット符号なし整数としてフェッチします。

GetUInt16(uint16) メソッド [181 ページ]
カラムから値を 16 ビット符号なし整数としてフェッチします。

1.10.26.1 GetUInt16(String^) メソッド

カラムから値を 16 ビット符号なし整数としてフェッチします。

構文

```
public virtual uint16 GetUInt16 (cname)
```

パラメータ

cname カラム名。

戻り値

符号なし整数としてのカラム値。

1.10.26.2 GetUInt16(uint16) メソッド

カラムから値を 16 ビット符号なし整数としてフェッチします。

構文

```
public virtual uint16 GetUInt16 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

符号なし整数としてのカラム値。

1.10.27 GetUInt32 メソッド

カラムから値を 32 ビット符号なし整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual unsigned int	GetUInt32(String^) [182 ページ]	カラムから値を 32 ビット符号なし整数としてフェッチします。
public virtual unsigned int	GetUInt32(uint16) [183 ページ]	カラムから値を 32 ビット符号なし整数としてフェッチします。

このセクションの内容:

1.10.27.1 GetUInt32(String^) メソッド

カラムから値を 32 ビット符号なし整数としてフェッチします。

構文

```
public virtual unsigned int GetUInt32 (cname)
```

パラメータ

cname カラム名。

戻り値

符号なし整数としてのカラム値。

1.10.27.2 GetUInt32(uint16) メソッド

カラムから値を 32 ビット符号なし整数としてフェッチします。

構文

```
public virtual unsigned int GetUInt32 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

符号なし整数としてのカラム値。

1.10.28 GetUInt64 メソッド

カラムから値を 64 ビット符号なし整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual uint64	GetUInt64(String^) [184 ページ]	カラムから値を 64 ビット符号なし整数としてフェッチします。
public virtual uint64	GetUInt64(uint16) [184 ページ]	カラムから値を 64 ビット符号なし整数としてフェッチします。

このセクションの内容:

GetUInt64(String^) メソッド [184 ページ]
カラムから値を 64 ビット符号なし整数としてフェッチします。

GetUInt64(uint16) メソッド [184 ページ]
カラムから値を 64 ビット符号なし整数としてフェッチします。

1.10.28.1 GetUInt64(String^) メソッド

カラムから値を 64 ビット符号なし整数としてフェッチします。

構文

```
public virtual uint64 GetUInt64 (cname)
```

パラメータ

cname カラム名。

戻り値

符号なし整数としてのカラム値。

1.10.28.2 GetUInt64(uint16) メソッド

カラムから値を 64 ビット符号なし整数としてフェッチします。

構文

```
public virtual uint64 GetUInt64 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

符号なし整数としてのカラム値。

1.10.29 IsNull メソッド

カラム値が NULL かどうかをチェックします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual bool	IsNull(String^) [185 ページ]	カラム値が NULL かどうかをチェックします。
public virtual bool	IsNull(uint16) [186 ページ]	カラム値が NULL かどうかをチェックします。

このセクションの内容:

1.10.29.1 IsNull(String^) メソッド

カラム値が NULL かどうかをチェックします。

構文

```
public virtual bool IsNull (cname)
```

パラメータ

cname カラム名。

戻り値

カラム値が NULL の場合は true。

1.10.29.2 IsNull(uint16) メソッド

カラム値が NULL かどうかをチェックします。

構文

```
public virtual bool IsNull (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラム値が NULL の場合は true。

1.10.30 Last() メソッド

カーソルを最後のローに移動します。

構文

```
public virtual void Last ()
```

1.10.31 Next() メソッド

カーソルをロー 1 つ分進めます。

構文

```
public virtual bool Next ()
```

戻り値

カーソルが正常に進められる場合は、true。カーソルが次のローに正常に進められる場合は、引き続きエラーが通知される可能性があります。たとえば SELECT 式の評価中に変換エラーが発生する可能性があります。この場合、カラム値を取得するときにもエラーが返されます。カーソルを進められなかった場合は、false が返されます。たとえば、次のローが存在しないなどです。この場合、移動後のカーソル位置は最後のローの後ろに設定されます。

1.10.32 Previous() メソッド

カーソルをロー 1 つ分戻します。

構文

```
public virtual bool Previous ()
```

戻り値

カーソルをロー 1 つ分戻せた場合は、true。カーソルを戻せなかった場合は、false。移動後のカーソル位置は、最初のローの前に設定されます。

1.10.33 Relative(int) メソッド

カーソルを、現在のカーソルの位置から、offset で指定したロー数分移動します。

構文

```
public virtual void Relative (offset)
```

パラメータ

offset 移動するローの数。

1.10.34 SetBool メソッド

カラムを boolean 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetBool(String^, bool) [188 ページ]	カラムを boolean 値に設定します。
public virtual void	SetBool(uint16, bool) [189 ページ]	カラムを boolean 値に設定します。

このセクションの内容:

1.10.34.1 SetBool(String^, bool) メソッド

カラムを boolean 値に設定します。

構文

```
public virtual void SetBool (cname, value)
```

パラメータ

cname カラム名。
value boolean 値。

1.10.34.2 SetBool(uint16, bool) メソッド

カラムを boolean 値に設定します。

構文

```
public virtual void SetBool (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value boolean 値。

1.10.35 SetByte メソッド

カラムを byte 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetByte(String^, uint8) [190 ページ]	カラムを byte 値に設定します。
public virtual void	SetByte(uint16, uint8) [190 ページ]	カラムを byte 値に設定します。

このセクションの内容:

SetByte(String^, uint8) メソッド [190 ページ]
カラムを byte 値に設定します。

SetByte(uint16, uint8) メソッド [190 ページ]
カラムを byte 値に設定します。

1.10.35.1 SetByte(String^, uint8) メソッド

カラムを byte 値に設定します。

構文

```
public virtual void SetByte (cname, value)
```

パラメータ

cname カラム名。
value byte 値。

1.10.35.2 SetByte(uint16, uint8) メソッド

カラムを byte 値に設定します。

構文

```
public virtual void SetByte (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value byte 値。

1.10.36 SetBytes メソッド

カラムを binary 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetBytes(String^, const Array< uint8 >^, int) [191 ページ]	カラムを binary 値に設定します。
public virtual void	SetBytes(uint16, const Array< uint8 >^, int) [192 ページ]	カラムを binary 値に設定します。

このセクションの内容:

SetBytes(String^, const Array< uint8 >^, int) メソッド [191 ページ]
カラムを binary 値に設定します。

SetBytes(uint16, const Array< uint8 >^, int) メソッド [192 ページ]
カラムを binary 値に設定します。

1.10.36.1 SetBytes(String^, const Array< uint8 >^, int) メソッド

カラムを binary 値に設定します。

構文

```
public virtual void SetBytes (cname, value, length)
```

パラメータ

cname カラム名。
value バイナリ値。NULL を渡すことは、SetNull メソッドを呼び出すことと同じです。
length byte 配列値から取得するバイト単位の長さ。

1.10.36.2 SetBytes(uint16, const Array< uint8 >^, int) メソッド

カラムを binary 値に設定します。

構文

```
public virtual void SetBytes (cid, value, length)
```

パラメータ

cid 0 から始まるカラムの序数。
value バイナリ値。NULL を渡すことは、SetNull メソッドを呼び出すことと同じです。
length byte 配列値から取得するバイト単位の長さ。

1.10.37 SetDateTime メソッド

カラムを DateTime 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetDateTime(String^, DateTime) [193 ページ]	カラムを DateTime 値に設定します。
public virtual void	SetDateTime(uint16, DateTime) [193 ページ]	カラムを DateTime 値に設定します。

このセクションの内容:

SetDateTime(String^, DateTime) メソッド [193 ページ]
カラムを DateTime 値に設定します。

SetDateTime(uint16, DateTime) メソッド [193 ページ]
カラムを DateTime 値に設定します。

1.10.37.1 SetDateTime(String^, DateTime) メソッド

カラムを DateTime 値に設定します。

構文

```
public virtual void SetDateTime (cname, value)
```

パラメータ

cname カラム名。
value DateTime 値。

1.10.37.2 SetDateTime(uint16, DateTime) メソッド

カラムを DateTime 値に設定します。

構文

```
public virtual void SetDateTime (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value DateTime 値。

1.10.38 SetDefault メソッド

カラムをそのデフォルト値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetDefault(String^) [194 ページ]	カラムをそのデフォルト値に設定します。
public virtual void	SetDefault(uint16) [194 ページ]	カラムをそのデフォルト値に設定します。

このセクションの内容:

SetDefault(String^) メソッド [194 ページ]
カラムをそのデフォルト値に設定します。

SetDefault(uint16) メソッド [194 ページ]
カラムをそのデフォルト値に設定します。

1.10.38.1 SetDefault(String^) メソッド

カラムをそのデフォルト値に設定します。

構文

```
public virtual void SetDefault (cname)
```

パラメータ

cname カラム名。

1.10.38.2 SetDefault(uint16) メソッド

カラムをそのデフォルト値に設定します。

構文

```
public virtual void SetDefault (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

1.10.39 SetDouble メソッド

カラムを double 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetDouble(String^, double) [195 ページ]	カラムを double 値に設定します。
public virtual void	SetDouble(uint16, double) [196 ページ]	カラムを double 値に設定します。

このセクションの内容:

SetDouble(String^, double) メソッド [195 ページ]
カラムを double 値に設定します。

SetDouble(uint16, double) メソッド [196 ページ]
カラムを double 値に設定します。

1.10.39.1 SetDouble(String^, double) メソッド

カラムを double 値に設定します。

構文

```
public virtual void SetDouble (cname, value)
```

パラメータ

cname カラム名。
value double 値。

1.10.39.2 SetDouble(uint16, double) メソッド

カラムを double 値に設定します。

構文

```
public virtual void SetDouble (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value double 値。

1.10.40 SetFloat メソッド

カラムを float 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetFloat(String^, float) [197 ページ]	カラムを float 値に設定します。
public virtual void	SetFloat(uint16, float) [197 ページ]	カラムを float 値に設定します。

このセクションの内容:

SetFloat(String^, float) メソッド [197 ページ]
カラムを float 値に設定します。

SetFloat(uint16, float) メソッド [197 ページ]
カラムを float 値に設定します。

1.10.40.1 SetFloat(String^, float) メソッド

カラムを float 値に設定します。

構文

```
public virtual void SetFloat (cname, value)
```

パラメータ

cname カラム名。
value float 値。

1.10.40.2 SetFloat(uint16, float) メソッド

カラムを float 値に設定します。

構文

```
public virtual void SetFloat (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value float 値。

1.10.41 SetGuid メソッド

カラムを GUID 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetGuid(uint16, Guid) [198 ページ]	カラムを GUID 値に設定します。
public virtual void	SetGuid(uint16, Guid) [199 ページ]	カラムを GUID 値に設定します。

このセクションの内容:

SetGuid(uint16, Guid) メソッド [198 ページ]
カラムを GUID 値に設定します。

SetGuid(uint16, Guid) メソッド [199 ページ]
カラムを GUID 値に設定します。

1.10.41.1 SetGuid(uint16, Guid) メソッド

カラムを GUID 値に設定します。

構文

```
public virtual void SetGuid (cname, value)
```

パラメータ

cname カラム名。
value GUID 値。

1.10.41.2 SetGuid(uint16, Guid) メソッド

カラムを GUID 値に設定します。

構文

```
public virtual void SetGuid (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value GUID 値。

1.10.42 SetInt16 メソッド

カラムを整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetInt16(String^, int16) [200 ページ]	カラムを整数値に設定します。
public virtual void	SetInt16(uint16, int16) [200 ページ]	カラムを整数値に設定します。

このセクションの内容:

SetInt16(String^, int16) メソッド [200 ページ]
カラムを整数値に設定します。

SetInt16(uint16, int16) メソッド [200 ページ]
カラムを整数値に設定します。

1.10.42.1 SetInt16(String^, int16) メソッド

カラムを整数値に設定します。

構文

```
public virtual void SetInt16 (cname, value)
```

パラメータ

cname カラム名。
value 符号付き整数値。

1.10.42.2 SetInt16(uint16, int16) メソッド

カラムを整数値に設定します。

構文

```
public virtual void SetInt16 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号付き整数値。

1.10.43 SetInt32 メソッド

カラムを整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetInt32(String^, int) [201 ページ]	カラムを整数値に設定します。
public virtual void	SetInt32(uint16, int) [202 ページ]	カラムを整数値に設定します。

このセクションの内容:

SetInt32(String^, int) メソッド [201 ページ]
カラムを整数値に設定します。

SetInt32(uint16, int) メソッド [202 ページ]
カラムを整数値に設定します。

1.10.43.1 SetInt32(String^, int) メソッド

カラムを整数値に設定します。

構文

```
public virtual void SetInt32 (cname, value)
```

パラメータ

cname カラム名。
value 符号付き整数値。

1.10.43.2 SetInt32(uint16, int) メソッド

カラムを整数値に設定します。

構文

```
public virtual void SetInt32 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号付き整数値。

1.10.44 SetInt64 メソッド

カラムを整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetInt64(String^, int64) [203 ページ]	カラムを整数値に設定します。
public virtual void	SetInt64(uint16, int64) [203 ページ]	カラムを整数値に設定します。

このセクションの内容:

SetInt64(String^, int64) メソッド [203 ページ]
カラムを整数値に設定します。

SetInt64(uint16, int64) メソッド [203 ページ]
カラムを整数値に設定します。

1.10.44.1 SetInt64(String^, int64) メソッド

カラムを整数値に設定します。

構文

```
public virtual void SetInt64 (cname, value)
```

パラメータ

cname カラム名。
value 符号付き整数値。

1.10.44.2 SetInt64(uint16, int64) メソッド

カラムを整数値に設定します。

構文

```
public virtual void SetInt64 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号付き整数値。

1.10.45 SetNull メソッド

カラムを NULL に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetNull(String^) [204 ページ]	カラムを NULL に設定します。
public virtual void	SetNull(uint16) [204 ページ]	カラムを NULL に設定します。

このセクションの内容:

SetNull(String^) メソッド [204 ページ]
カラムを NULL に設定します。

SetNull(uint16) メソッド [204 ページ]
カラムを NULL に設定します。

1.10.45.1 SetNull(String^) メソッド

カラムを NULL に設定します。

構文

```
public virtual void SetNull (cname)
```

パラメータ

cname カラム名。

1.10.45.2 SetNull(uint16) メソッド

カラムを NULL に設定します。

構文

```
public virtual void SetNull (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

1.10.46 SetString メソッド

カラムを文字列値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetString(String^, String^) [205 ページ]	カラムを文字列値に設定します。
public virtual void	SetString(uint16, String^) [206 ページ]	カラムを文字列値に設定します。

このセクションの内容:

SetString(String^, String^) メソッド [205 ページ]
カラムを文字列値に設定します。

SetString(uint16, String^) メソッド [206 ページ]
カラムを文字列値に設定します。

1.10.46.1 SetString(String^, String^) メソッド

カラムを文字列値に設定します。

構文

```
public virtual void SetString (cname, value)
```

パラメータ

cname カラム名。
value 文字列値。

1.10.46.2 SetString(uint16, String^) メソッド

カラムを文字列値に設定します。

構文

```
public virtual void SetString (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 文字列値。

1.10.47 SetUInt16 メソッド

カラムを符号なし 16 ビット整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetUInt16(String^, uint16) [207 ページ]	カラムを符号なし 16 ビット整数値に設定します。
public virtual void	SetUInt16(uint16, uint16) [207 ページ]	カラムを符号なし 16 ビット整数値に設定します。

このセクションの内容:

1.10.47.1 SetUInt16(String^, uint16) メソッド

カラムを符号なし 16 ビット整数値に設定します。

構文

```
public virtual void SetUInt16 (cname, value)
```

パラメータ

cname カラム名。
value 符号なし整数値。

1.10.47.2 SetUInt16(uint16, uint16) メソッド

カラムを符号なし 16 ビット整数値に設定します。

構文

```
public virtual void SetUInt16 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号なし整数値。

1.10.48 SetUInt32 メソッド

カラムを符号なし 32 ビット整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetUInt32(String^, unsigned int) [208 ページ]	カラムを符号なし 32 ビット整数値に設定します。
public virtual void	SetUInt32(uint16, unsigned int) [209 ページ]	カラムを符号なし 32 ビット整数値に設定します。

このセクションの内容:

1.10.48.1 SetUInt32(String^, unsigned int) メソッド

カラムを符号なし 32 ビット整数値に設定します。

構文

```
public virtual void SetUInt32 (cname, value)
```

パラメータ

cname カラム名。
value 符号なし整数値。

1.10.48.2 SetUInt32(uint16, unsigned int) メソッド

カラムを符号なし 32 ビット整数値に設定します。

構文

```
public virtual void SetUInt32 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号なし整数値。

1.10.49 SetUInt64 メソッド

カラムを符号なし 64 ビット整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetUInt64(String^, uint64) [210 ページ]	カラムを符号なし 64 ビット整数値に設定します。
public virtual void	SetUInt64(uint16, uint64) [210 ページ]	カラムを符号なし 64 ビット整数値に設定します。

このセクションの内容:

SetUInt64(String^, uint64) メソッド [210 ページ]
カラムを符号なし 64 ビット整数値に設定します。

SetUInt64(uint16, uint64) メソッド [210 ページ]
カラムを符号なし 64 ビット整数値に設定します。

1.10.49.1 SetUInt64(String^, uint64) メソッド

カラムを符号なし 64 ビット整数値に設定します。

構文

```
public virtual void SetUInt64 (cname, value)
```

パラメータ

cname カラム名。
value 符号なし整数値。

1.10.49.2 SetUInt64(uint16, uint64) メソッド

カラムを符号なし 64 ビット整数値に設定します。

構文

```
public virtual void SetUInt64 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号なし整数値。

1.10.50 Update() メソッド

現在の行を更新します。

構文

```
public virtual void Update ()
```

1.10.51 UpdateBegin() メソッド

カラムの設定に使用される更新モードを選択します。

構文

```
public virtual void UpdateBegin ()
```

備考

Ultra Light が更新モードの場合、プライマリキー内のカラムは修正できません。

1.11 ResultSetSchema クラス

Ultra Light の結果セットのスキーマを表します。ResultSet.GetResultSetSchema または PreparedStatement.GetResultSetSchema は ResultSetSchema を返します。

ネームスペース

```
UltraLite
```

構文

```
public ref class  sealed  :  CursorSchema
```

メンバー

ResultSetSchema のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public virtual uint16	GetColumnCount() [213 ページ]	結果セットまたはテーブル内のカラム数を取得します。

変更子とタイプ	メソッド	説明
public virtual uint16	GetColumnID(String^) [213 ページ]	0 から始まるカラム ID を名前から取得します。
public virtual String	GetColumnName(uint16, uint16type) [213 ページ]	0 から始まる ID を指定してカラムの名前を取得します。
public virtual uint16	GetColumnPrecision(uint16) [214 ページ]	数値カラムの精度を取得します。
public virtual uint16	GetColumnScale(uint16) [215 ページ]	数値カラムの位取りを取得します。
public virtual int	GetColumnSize(uint16) [215 ページ]	カラムのサイズを取得します。
public virtual uint16	GetColumnSQLType(uint16) [216 ページ]	カラムの SQL の型を取得します。
public virtual uint16	GetColumnType(uint16) [216 ページ]	カラムの記憶タイプまたはホスト変数の型を取得します。
public virtual bool	IsAliased(uint16) [217 ページ]	結果セット内のカラムにエイリアスが付与されているかどうかを示します。

このセクションの内容:

GetColumnCount() メソッド [213 ページ]
結果セットまたはテーブル内のカラム数を取得します。

GetColumnID(String^) メソッド [213 ページ]
0 から始まるカラム ID を名前から取得します。

GetColumnName(uint16, uint16type) メソッド [213 ページ]
0 から始まる ID を指定してカラムの名前を取得します。

GetColumnPrecision(uint16) メソッド [214 ページ]
数値カラムの精度を取得します。

GetColumnScale(uint16) メソッド [215 ページ]
数値カラムの位取りを取得します。

GetColumnSize(uint16) メソッド [215 ページ]
カラムのサイズを取得します。

GetColumnSQLType(uint16) メソッド [216 ページ]
カラムの SQL の型を取得します。

GetColumnType(uint16) メソッド [216 ページ]
カラムの記憶タイプまたはホスト変数の型を取得します。

IsAliased(uint16) メソッド [217 ページ]
結果セット内のカラムにエイリアスが付与されているかどうかを示します。

1.11.1 GetColumnCount() メソッド

結果セットまたはテーブル内のカラム数を取得します。

構文

```
public virtual uint16 GetColumnCount ()
```

戻り値

結果セットまたはテーブル内のカラム数。

1.11.2 GetColumnID(String^) メソッド

0 から始まるカラム ID を名前から取得します。

構文

```
public virtual uint16 GetColumnID (columnName)
```

パラメータ

columnName カラムの名前。

戻り値

カラム ID。

1.11.3 GetColumnName(uint16, uint16type) メソッド

0 から始まる ID を指定してカラムの名前を取得します。

構文

```
public virtual String GetColumnName (cid, type)
```

パラメータ

cid 0 から始まるカラムの序数。
type カラム名の必要な型。

戻り値

存在する場合は、カラム名を格納する文字列。

備考

選択した型、および SELECT 文でのカラムの宣言方法によっては、カラム名が [table-name].[column-name] という形式で返されることがあります。

type パラメータを使用して、どの型のカラム名を返すかを指定します。

1.11.4 GetColumnPrecision(uint16) メソッド

数値カラムの精度を取得します。

構文

```
public virtual uint16 GetColumnPrecision (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

精度

1.11.5 GetColumnScale(uint16) メソッド

数値カラムの位取りを取得します。

構文

```
public virtual uint16 GetColumnScale (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

位取り。

1.11.6 GetColumnSize(uint16) メソッド

カラムのサイズを取得します。

構文

```
public virtual int GetColumnSize (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラムサイズ。

1.11.7 GetColumnSQLType(uint16) メソッド

カラムの SQL の型を取得します。

構文

```
public virtual uint16 GetColumnSQLType (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラムが存在しない場合は、SQLTYPE_BAD_INDEX。

1.11.8 GetColumnType(uint16) メソッド

カラムの記憶タイプまたはホスト変数の型を取得します。

構文

```
public virtual uint16 GetColumnType (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラムが存在しない場合は、TYPE_BAD_INDEX。

1.11.9 IsAliased(uint16) メソッド

結果セット内のカラムにエイリアスが付与されているかどうかを示します。

構文

```
public virtual bool IsAliased (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラムのエイリアスが使用されている場合は true、そうでない場合は false。

1.12 Table クラス

Ultra Light データベース内のテーブルを表します。Connection.OpenTable は Table を返します。

ネームスペース

```
UltraLite
```

構文

```
public ref class  sealed  : Cursor
```

メンバー

Table のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public virtual void	AfterLast() [224 ページ]	カーソルを最後のローの後に移動します。
public virtual void	AppendBytes [224 ページ]	バイトをカラムに追加します。
public virtual void	AppendChars [226 ページ]	文字列のチャンクをカラムに追加します。
public virtual void	BeforeFirst() [227 ページ]	カーソルを最初のローの前に移動します。
public virtual void	Close() [228 ページ]	このオブジェクトを破棄します。
public virtual void	Delete() [228 ページ]	現在のローを削除し、カーソルを次の有効なローに移動します。
public void	DeleteAllRows() [228 ページ]	テーブルからすべてのローを削除します。
public virtual void	DeleteNamed(String^) [229 ページ]	現在のローを削除し、カーソルを次の有効なローに移動します。
public bool	Find(uint16) [229 ページ]	現在のインデックスに基づいて、テーブルを順方向にスキャンして完全一致のルックアップを実行します。
public void	FindBegin() [230 ページ]	検索モードを開始することで、テーブルで新規に検索呼び出しを実行する準備を行います。
public bool	FindFirst(uint16) [230 ページ]	現在のインデックスに基づいて、テーブルを順方向にスキャンして完全一致のルックアップを実行します。
public bool	FindLast(uint16) [231 ページ]	現在のインデックスに基づいて、テーブルを逆方向にスキャンして完全一致のルックアップを実行します。
public bool	FindNext(uint16) [231 ページ]	インデックスに完全に一致する次のローを取得します。
public bool	FindPrevious(uint16) [232 ページ]	インデックスに完全に一致する前のローを取得します。
public virtual void	First() [232 ページ]	カーソルを最初のローに移動します。
public virtual int64	GetBinaryLength [233 ページ]	カラムの値のバイナリ長さを取得します。
public virtual bool	GetBool [234 ページ]	カラムから値を boolean としてフェッチします。
public virtual uint8	GetByte [236 ページ]	カラムから値を byte としてフェッチします。
public virtual int64	GetBytes [237 ページ]	カラムからバイナリチャンクを取得します。
public virtual int64	GetChars [239 ページ]	カラムからワイド文字列のチャンクを取得します。
public virtual DateTime	GetDateTime [242 ページ]	カラムから値を DateTime としてフェッチします。
public virtual double	GetDouble [243 ページ]	カラムから値を double としてフェッチします。
public virtual float	GetFloat [245 ページ]	カラムから値を float としてフェッチします。
public virtual Guid	GetGuid [246 ページ]	カラムから値を GUID としてフェッチします。

変更子とタイプ	メソッド	説明
public virtual int16	GetInt16 [248 ページ]	カラムから値を 16 ビット整数としてフェッチします。
public virtual int	GetInt32 [249 ページ]	カラムから値を整数としてフェッチします。
public virtual int64	GetInt64 [251 ページ]	カラムから値を 64 ビット整数としてフェッチします。
public virtual unsigned int	GetRowCount(unsigned int) [252 ページ]	テーブルのローの数を取得します。
public virtual uint8	GetState() [253 ページ]	カーソルの内部ステータスを取得します。
public virtual String	GetString [253 ページ]	カラムから値を文字列としてフェッチします。
public virtual int64	GetStringLength [255 ページ]	カラムの値の文字列長さを取得します。
public TableSchema	GetTableSchema() [256 ページ]	テーブルに関するスキーマ情報の取得に使用できる TableSchema オブジェクトを返します。
public virtual uint16	GetUInt16 [257 ページ]	カラムから値を 16 ビット符号なし整数としてフェッチします。
public virtual unsigned int	GetUInt32 [258 ページ]	カラムから値を 32 ビット符号なし整数としてフェッチします。
public virtual uint64	GetUInt64 [260 ページ]	カラムから値を 64 ビット符号なし整数としてフェッチします。
public void	Insert() [261 ページ]	新しいローをテーブルに挿入します。
public void	InsertBegin() [261 ページ]	設定されたカラムに対する挿入モードを選択します。
public virtual bool	IsNull [262 ページ]	カラム値が NULL かどうかをチェックします。
public virtual void	Last() [263 ページ]	カーソルを最後のローに移動します。
public bool	Lookup(uint16) [263 ページ]	現在のインデックスに基づいて、テーブルを順方向にスキャンしてルックアップを実行します。
public bool	LookupBackward(uint16) [264 ページ]	現在のインデックスに基づいて、テーブルを逆方向にスキャンしてルックアップを実行します。
public void	LookupBegin() [265 ページ]	テーブルで新規に検索を実行する準備を行います。
public bool	LookupForward(uint16) [265 ページ]	現在のインデックスに基づいて、テーブルを順方向にスキャンしてルックアップを実行します。
public virtual bool	Next() [266 ページ]	カーソルをロー 1 つ分進めます。
public virtual bool	Previous() [266 ページ]	カーソルをロー 1 つ分戻します。
public virtual void	Relative(int) [267 ページ]	カーソルを、現在のカーソルの位置から、offset で指定したロー数分移動します。
public virtual void	SetBool [267 ページ]	カラムを boolean 値に設定します。

変更子とタイプ	メソッド	説明
public virtual void	SetByte [268 ページ]	カラムを byte 値に設定します。
public virtual void	SetBytes [270 ページ]	カラムを binary 値に設定します。
public virtual void	SetDateTime [271 ページ]	カラムを DateTime 値に設定します。
public virtual void	SetDefault [273 ページ]	カラムをそのデフォルト値に設定します。
public virtual void	SetDouble [274 ページ]	カラムを double 値に設定します。
public virtual void	SetFloat [275 ページ]	カラムを float 値に設定します。
public virtual void	SetGuid [277 ページ]	カラムを GUID 値に設定します。
public virtual void	SetInt16 [278 ページ]	カラムを整数値に設定します。
public virtual void	SetInt32 [280 ページ]	カラムを整数値に設定します。
public virtual void	SetInt64 [281 ページ]	カラムを整数値に設定します。
public virtual void	SetNull [283 ページ]	カラムを NULL に設定します。
public virtual void	SetString [284 ページ]	カラムを文字列値に設定します。
public virtual void	SetUInt16 [285 ページ]	カラムを符号なし 16 ビット整数値に設定します。
public virtual void	SetUInt32 [287 ページ]	カラムを符号なし 32 ビット整数値に設定します。
public virtual void	SetUInt64 [288 ページ]	カラムを符号なし 64 ビット整数値に設定します。
public void	TruncateTable() [289 ページ]	テーブルをトランケートし、STOP SYNCHRONIZATION DELETE を一時的にアクティブにします。
public virtual void	Update() [290 ページ]	現在の行を更新します。
public virtual void	UpdateBegin() [290 ページ]	カラムの設定に使用される更新モードを選択します。

このセクションの内容:

現在のローを削除し、カーソルを次の有効なローに移動します。

DeleteAllRows() メソッド [228 ページ]
テーブルからすべてのローを削除します。

DeleteNamed(String^) メソッド [229 ページ]
現在のローを削除し、カーソルを次の有効なローに移動します。

Find(uint16) メソッド [229 ページ]
現在のインデックスに基づいて、テーブルを順方向にスキャンして完全一致のルックアップを実行します。

FindBegin() メソッド [230 ページ]
検索モードを開始することで、テーブルで新規に検索呼び出しを実行する準備を行います。

FindFirst(uint16) メソッド [230 ページ]
現在のインデックスに基づいて、テーブルを順方向にスキャンして完全一致のルックアップを実行します。

FindLast(uint16) メソッド [231 ページ]
現在のインデックスに基づいて、テーブルを逆方向にスキャンして完全一致のルックアップを実行します。

FindNext(uint16) メソッド [231 ページ]
インデックスに完全に一致する次のローを取得します。

FindPrevious(uint16) メソッド [232 ページ]
インデックスに完全に一致する前のローを取得します。

First() メソッド [232 ページ]
カーソルを最初のローに移動します。

GetBinaryLength メソッド [233 ページ]
カラムの値のバイナリ長さを取得します。

GetBool メソッド [234 ページ]
カラムから値を boolean としてフェッチします。

GetByte メソッド [236 ページ]
カラムから値を byte としてフェッチします。

GetBytes メソッド [237 ページ]
カラムからバイナリチャンクを取得します。

GetChars メソッド [239 ページ]
カラムからワイド文字列のチャンクを取得します。

GetDateTime メソッド [242 ページ]
カラムから値を DateTime としてフェッチします。

GetDouble メソッド [243 ページ]
カラムから値を double としてフェッチします。

GetFloat メソッド [245 ページ]
カラムから値を float としてフェッチします。

GetGuid メソッド [246 ページ]
カラムから値を GUID としてフェッチします。

GetInt16 メソッド [248 ページ]
カラムから値を 16 ビット整数としてフェッチします。

GetInt32 メソッド [249 ページ]

カラムから値を整数としてフェッチします。

GetInt64 メソッド [251 ページ]
カラムから値を 64 ビット整数としてフェッチします。

GetRowCount(unsigned int) メソッド [252 ページ]
テーブルのローの数を取得します。

GetState() メソッド [253 ページ]
カーソルの内部ステータスを取得します。

GetString メソッド [253 ページ]
カラムから値を文字列としてフェッチします。

GetStringLength メソッド [255 ページ]
カラムの値の文字列長さを取得します。

GetTableSchema メソッド [256 ページ]
テーブルに関するスキーマ情報の取得に使用できる TableSchema オブジェクトを返します。

GetUInt16 メソッド [257 ページ]
カラムから値を 16 ビット符号なし整数としてフェッチします。

GetUInt32 メソッド [258 ページ]
カラムから値を 32 ビット符号なし整数としてフェッチします。

GetUInt64 メソッド [260 ページ]
カラムから値を 64 ビット符号なし整数としてフェッチします。

Insert() メソッド [261 ページ]
新しいローをテーブルに挿入します。

InsertBegin() メソッド [261 ページ]
設定されたカラムに対する挿入モードを選択します。

IsNull メソッド [262 ページ]
カラム値が NULL かどうかをチェックします。

Last() メソッド [263 ページ]
カーソルを最後のローに移動します。

Lookup(uint16) メソッド [263 ページ]
現在のインデックスに基づいて、テーブルを順方向にスキャンしてルックアップを実行します。

LookupBackward(uint16) メソッド [264 ページ]
現在のインデックスに基づいて、テーブルを逆方向にスキャンしてルックアップを実行します。

LookupBegin() メソッド [265 ページ]
テーブルで新規に検索を実行する準備を行います。

LookupForward(uint16) メソッド [265 ページ]
現在のインデックスに基づいて、テーブルを順方向にスキャンしてルックアップを実行します。

Next() メソッド [266 ページ]
カーソルをロー 1 つ分進めます。

Previous() メソッド [266 ページ]
カーソルをロー 1 つ分戻します。

Relative(int) メソッド [267 ページ]

カーソルを、現在のカーソルの位置から、offset で指定したロー数分移動します。

SetBool メソッド [267 ページ]
カラムを boolean 値に設定します。

SetByte メソッド [268 ページ]
カラムを byte 値に設定します。

SetBytes メソッド [270 ページ]
カラムを binary 値に設定します。

SetDateTime メソッド [271 ページ]
カラムを DateTime 値に設定します。

SetDefault メソッド [273 ページ]
カラムをそのデフォルト値に設定します。

SetDouble メソッド [274 ページ]
カラムを double 値に設定します。

SetFloat メソッド [275 ページ]
カラムを float 値に設定します。

SetGuid メソッド [277 ページ]
カラムを GUID 値に設定します。

SetInt16 メソッド [278 ページ]
カラムを整数値に設定します。

SetInt32 メソッド [280 ページ]
カラムを整数値に設定します。

SetInt64 メソッド [281 ページ]
カラムを整数値に設定します。

SetNull メソッド [283 ページ]
カラムを NULL に設定します。

SetString メソッド [284 ページ]
カラムを文字列値に設定します。

SetUInt16 メソッド [285 ページ]
カラムを符号なし 16 ビット整数値に設定します。

SetUInt32 メソッド [287 ページ]
カラムを符号なし 32 ビット整数値に設定します。

SetUInt64 メソッド [288 ページ]
カラムを符号なし 64 ビット整数値に設定します。

TruncateTable() メソッド [289 ページ]
テーブルをトランケートし、STOP SYNCHRONIZATION DELETE を一時的にアクティブにします。

Update() メソッド [290 ページ]
現在の行を更新します。

UpdateBegin() メソッド [290 ページ]
カラムの設定に使用される更新モードを選択します。

1.12.1 AfterLast() メソッド

カーソルを最後のローの後に移動します。

構文

```
public virtual void AfterLast ()
```

1.12.2 AppendBytes メソッド

バイトをカラムに追加します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	AppendBytes(String^, const Array< uint8 >^, int, int) [224 ページ]	バイトをカラムに追加します。
public virtual void	AppendBytes(uint16, const Array< uint8 >^, int, int) [225 ページ]	バイトをカラムに追加します。

このセクションの内容:

AppendBytes(String^, const Array< uint8 >^, int, int) メソッド [224 ページ]
バイトをカラムに追加します。

AppendBytes(uint16, const Array< uint8 >^, int, int) メソッド [225 ページ]
バイトをカラムに追加します。

1.12.2.1 AppendBytes(String^, const Array< uint8 >^, int, int) メソッド

バイトをカラムに追加します。

構文

```
public virtual void AppendBytes (cname, value, offset, size)
```

パラメータ

cname カラム名。
value 追加するバイトのチャンク。
offset バイトのチャンク内でのオフセット (開始位置)。
size バイトのチャンクのサイズ (バイト単位)。

備考

AppendBytes メソッド呼び出しにより現時点で書き込み済みのカラムの末尾に、指定のバイトが追加されます。

1.12.2.2 AppendBytes(uint16, const Array< uint8 >^, int, int) メソッド

バイトをカラムに追加します。

構文

```
public virtual void AppendBytes (cid, value, offset, size)
```

パラメータ

cid 0 から始まるカラムの序数。
value 追加するバイトのチャンク。
offset バイトのチャンク内でのオフセット (開始位置)。
size バイトのチャンクのサイズ (バイト単位)。

備考

AppendBytes メソッド呼び出しにより現時点で書き込み済みのカラムの末尾に、指定のバイトが追加されます。

1.12.3 AppendChars メソッド

文字列のチャンクをカラムに追加します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	AppendChars(String^, const Array< wchar_t >^, int, int) [226 ページ]	文字列のチャンクをカラムに追加します。
public virtual void	AppendChars(uint16, const Array< wchar_t >^, int, int) [227 ページ]	文字列のチャンクをカラムに追加します。

このセクションの内容:

AppendChars(String^, const Array< wchar_t >^, int, int) メソッド [226 ページ]
文字列のチャンクをカラムに追加します。

AppendChars(uint16, const Array< wchar_t >^, int, int) メソッド [227 ページ]
文字列のチャンクをカラムに追加します。

1.12.3.1 AppendChars(String^, const Array< wchar_t >^, int, int) メソッド

文字列のチャンクをカラムに追加します。

構文

```
public virtual void AppendChars (cname, value, offset, size)
```

パラメータ

cname カラム名。
value 追加する文字列のチャンク。
offset 文字列のチャンク内でのオフセット (開始位置)。
size 文字列のチャンクの文字単位での長さです。

備考

このメソッドは、AppendChars メソッド呼び出しにより現時点で書き込み済みの文字列の末尾に、指定の文字列を追加します。

1.12.3.2 AppendChars(uint16, const Array< wchar_t >^, int, int) メソッド

文字列のチャンクをカラムに追加します。

構文

```
public virtual void AppendChars (cid, value, offset, size)
```

パラメータ

cid 0 から始まるカラムの序数。
value 追加する文字列のチャンク。
offset 文字列のチャンク内でのオフセット (開始位置)。
size 文字列のチャンクの文字単位での長さです。

備考

このメソッドは、AppendChars メソッド呼び出しにより現時点で書き込み済みの文字列の末尾に、指定の文字列を追加します。

1.12.4 BeforeFirst() メソッド

カーソルを最初のローの前に移動します。

構文

```
public virtual void BeforeFirst ()
```

1.12.5 CloseObject() メソッド

このオブジェクトを破棄します。

構文

```
public virtual void CloseObject ()
```

1.12.6 Delete() メソッド

現在のローを削除し、カーソルを次の有効なローに移動します。

構文

```
public virtual void Delete ()
```

1.12.7 DeleteAllRows() メソッド

テーブルからすべてのローを削除します。

構文

```
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必要があることがあります。接続で Connection.StopSynchronizationDelete を呼び出すと、削除されたローが同期されません。

> **注記**
>
> 別の接続からのコミットされていない挿入は削除されません。このような挿入は、他の接続が DeleteAllRows メソッドを呼び出した後にロールバックを実行した場合にも削除されません。

インデックスを使用しないでこのテーブルを開いた場合、テーブルは読み込み専用とみなされ、データを削除できません。

1.12.8 DeleteNamed(String^) メソッド

現在のローを削除し、カーソルを次の有効なローに移動します。

構文

```
public virtual void DeleteNamed (tableName)
```

パラメータ

tableName 削除元のテーブル名またはその相関。

1.12.9 Find(uint16) メソッド

現在のインデックスに基づいて、テーブルを順方向にスキャンして完全一致のルックアップを実行します。

構文

```
public bool Find (ncols)
```

パラメータ

ncols 複合インデックスの場合に、検索中に使用するカラムの数。

戻り値

インデックスの値に一致するローがない場合は、カーソルの位置が最後のローの後ろに設定され、false が返されます。

備考

検索する値を指定するには、インデックスのカラムごとに値を設定します。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。

1.12.10 FindBegin() メソッド

検索モードを開始することで、テーブルで新規に検索呼び出しを実行する準備を行います。

構文

```
public void FindBegin ()
```

備考

テーブルを開くのに使用されたインデックス内のカラムのみを設定できます。インデックスを使用しないでテーブルを開いた場合は、このメソッドを呼び出せません。

1.12.11 FindFirst(uint16) メソッド

現在のインデックスに基づいて、テーブルを順方向にスキャンして完全一致のルックアップを実行します。

構文

```
public bool FindFirst (ncols)
```

パラメータ

ncols 複合インデックスの場合に、検索中に使用するカラムの数。

戻り値

インデックスの値に一致するローがない場合は、カーソルの位置が最後のローの後ろに設定され、false が返されます。

備考

検索する値を指定するには、インデックスのカラムごとに値を設定します。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。

1.12.12 FindLast(uint16) メソッド

現在のインデックスに基づいて、テーブルを逆方向にスキャンして完全一致のルックアップを実行します。

構文

```
public bool FindLast (ncols)
```

パラメータ

ncols 複合インデックスの場合に、検索中に使用するカラムの数。

戻り値

インデックスの値に一致するローがない場合は、カーソルの位置が最初のローの前に設定され、false が返されます。

備考

検索する値を指定するには、インデックスのカラムごとに値を設定します。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。

1.12.13 FindNext(uint16) メソッド

インデックスに完全に一致する次のローを取得します。

構文

```
public bool FindNext (ncols)
```

パラメータ

ncols 複合インデックスの場合に、検索中に使用するカラムの数。

戻り値

それ以上インデックスに一致するローがない場合は、false。この場合、カーソルは最後のローの後ろに配置されます。

1.12.14 FindPrevious(uint16) メソッド

インデックスに完全に一致する前のローを取得します。

構文

```
public bool FindPrevious (ncols)
```

パラメータ

ncols 複合インデックスの場合に、検索中に使用するカラムの数。

戻り値

それ以上インデックスに一致するローがない場合は、false。この場合、カーソルは最初のローの前に配置されます。

1.12.15 First() メソッド

カーソルを最初のローに移動します。

構文

```
public virtual void First ()
```

1.12.16 GetBinaryLength メソッド

カラムの値のバイナリ長さを取得します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual int64	GetBinaryLength(String^) [233 ページ]	カラムの値のバイナリ長さを取得します。
public virtual int64	GetBinaryLength(uint16) [234 ページ]	カラムの値のバイナリ長さを取得します。

このセクションの内容:

GetBinaryLength(String^) メソッド [233 ページ]
カラムの値のバイナリ長さを取得します。

GetBinaryLength(uint16) メソッド [234 ページ]
カラムの値のバイナリ長さを取得します。

1.12.16.1 GetBinaryLength(String^) メソッド

カラムの値のバイナリ長さを取得します。

構文

```
public virtual int64 GetBinaryLength (cname)
```

パラメータ

cname カラム名。

戻り値

binary としてのカラムの値の長さ。

1.12.16.2 GetBinaryLength(uint16) メソッド

カラムの値のバイナリ長さを取得します。

構文

```
public virtual int64 GetBinaryLength (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

binary としてのカラムの値の長さ。

1.12.17 GetBool メソッド

カラムから値を boolean としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual bool	GetBool(String^) [235 ページ]	カラムから値を boolean としてフェッチします。
public virtual bool	GetBool(uint16) [235 ページ]	カラムから値を boolean としてフェッチします。

このセクションの内容:

GetBool(String^) メソッド [235 ページ]
カラムから値を boolean としてフェッチします。

GetBool(uint16) メソッド [235 ページ]
カラムから値を boolean としてフェッチします。

1.12.17.1 GetBool(String^) メソッド

カラムから値を boolean としてフェッチします。

構文

```
public virtual bool GetBool (cname)
```

パラメータ

cname カラム名。

戻り値

boolean としてのカラム値。

1.12.17.2 GetBool(uint16) メソッド

カラムから値を boolean としてフェッチします。

構文

```
public virtual bool GetBool (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

boolean としてのカラム値。

1.12.18 GetByte メソッド

カラムから値を byte としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual uint8	GetByte(String^) [236 ページ]	カラムから値を byte としてフェッチします。
public virtual uint8	GetByte(uint16) [237 ページ]	カラムから値を byte としてフェッチします。

このセクションの内容:

GetByte(String^) メソッド [236 ページ]
カラムから値を byte としてフェッチします。

GetByte(uint16) メソッド [237 ページ]
カラムから値を byte としてフェッチします。

1.12.18.1 GetByte(String^) メソッド

カラムから値を byte としてフェッチします。

構文

```
public virtual uint8 GetByte (cname)
```

パラメータ

cname カラム名。

戻り値

byte としてのカラム値。

1.12.18.2 GetByte(uint16) メソッド

カラムから値を byte としてフェッチします。

構文

```
public virtual uint8 GetByte (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

byte としてのカラム値。

1.12.19 GetBytes メソッド

カラムからバイナリチャンクを取得します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual int64	GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) [238 ページ]	カラムからバイナリチャンクを取得します。
public virtual int64	GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) [238 ページ]	カラムからバイナリチャンクを取得します。

このセクションの内容:

GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) メソッド [238 ページ]
カラムからバイナリチャンクを取得します。

GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) メソッド [238 ページ]
カラムからバイナリチャンクを取得します。

1.12.19.1 GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) メソッド

カラムからバイナリチャンクを取得します。

構文

```
public virtual int64 GetBytes (cname, srcOffset, dst, dstOffset, count)
```

パラメータ

cname カラム名。
dst バイトを保持するバッファ。
count バッファのサイズ (バイト単位)。
srcOffset 値内でのオフセット (読み込み開始位置)、または前回の読み込みが終了したところから続行する場合は BLOB_CONTINUE 定数。
dstOffset 宛先のバッファ内でのオフセット (バイトのコピー位置)。

戻り値

宛先のバッファにコピーされたバイト数。dst 値が NULL の場合は、残りのバイト数が返されます。カラムが NULL のときは、dst パラメータに空の文字列が返されます。

備考

0 が返された場合は、値の最後に到達しました。

IsNull メソッドを使用して、NULL と空の文字列を区別してください。

1.12.19.2 GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) メソッド

カラムからバイナリチャンクを取得します。

構文

```
public virtual int64 GetBytes (cid, srcOffset, dst, dstOffset, count)
```

パラメータ

cid 0 から始まるカラムの序数。
dst バイトを保持するバッファ。
count バッファのサイズ (バイト単位)。
srcOffset 値内でのオフセット (読み込み開始位置)、または前回の読み込みが終了したところから続行する場合は BLOB_CONTINUE 定数。
dstOffset 宛先のバッファ内でのオフセット (バイトのコピー位置)。

戻り値

宛先のバッファにコピーされたバイト数。dst 値が NULL の場合、このメソッドは残りのバイト数を返します。カラムが NULL のときは、dst パラメータに空の文字列が返されます。

備考

0 が返された場合は、値の最後に到達しました。

IsNull メソッドを使用して、NULL と空の文字列を区別してください。

1.12.20 GetChars メソッド

カラムからワイド文字列のチャンクを取得します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual int64	GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) [240 ページ]	カラムからワイド文字列のチャンクを取得します。
public virtual int64	GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) [241 ページ]	カラムからワイド文字列のチャンクを取得します。

このセクションの内容:

GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) メソッド [240 ページ]
カラムからワイド文字列のチャンクを取得します。

GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) メソッド [241 ページ]
カラムからワイド文字列のチャンクを取得します。

1.12.20.1 GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) メソッド

カラムからワイド文字列のチャンクを取得します。

構文

```
public virtual int64 GetChars (cname, srcOffset, dst, dstOffset, count)
```

パラメータ

cname カラム名。
dst 文字列のチャンクを保持するバッファ。文字列は、トランケートされても NULL で終了します。
count バッファのサイズ (文字数)。
srcOffset 値内でのオフセット (読み込み開始位置)、または前回の読み込みが終了したところから続行する場合は BLOB_CONTINUE 定数。
dstOffset 宛先のバッファ内でのオフセット (文字のコピー位置)。

戻り値

宛先のバッファにコピーされた文字数 (NULL ターミネータを含まない)。dst 値が NULL の場合、このメソッドは文字列の残りの文字数を返します。カラムが NULL のときは、dst パラメータに空の文字列が返されます。

備考

0 が返された場合は、値の最後に到達しました。

IsNull メソッドを使用して、NULL と空の文字列を区別してください。

1.12.20.2 GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) メソッド

カラムからワイド文字列のチャンクを取得します。

構文

```
public virtual int64 GetChars (cid, srcOffset, dst, dstOffset, count)
```

パラメータ

cid 0 から始まるカラムの序数。
dst 文字列のチャンクを保持するバッファ。文字列は、トランケートされても NULL で終了します。
count バッファのサイズ (文字数)。
srcOffset 値内でのオフセット (読み込み開始位置)、または前回の読み込みが終了したところから続行する場合は BLOB_CONTINUE 定数。
dstOffset 宛先のバッファ内でのオフセット (文字のコピー位置)。

戻り値

宛先のバッファにコピーされた文字数 (NULL ターミネータを含まない)。dst 値が NULL の場合、このメソッドは文字列の残りの文字数を返します。カラムが NULL のときは、dst パラメータに空の文字列が返されます。

備考

0 が返された場合は、値の最後に到達しました。

IsNull メソッドを使用して、NULL と空の文字列を区別してください。

1.12.21 GetDateTime メソッド

カラムから値を DateTime としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual DateTime	GetDateTime(String^) [242 ページ]	カラムから値を DateTime としてフェッチします。
public virtual DateTime	GetDateTime(uint16) [243 ページ]	カラムから値を DateTime としてフェッチします。

このセクションの内容:

GetDateTime(String^) メソッド [242 ページ]
カラムから値を DateTime としてフェッチします。

GetDateTime(uint16) メソッド [243 ページ]
カラムから値を DateTime としてフェッチします。

1.12.21.1 GetDateTime(String^) メソッド

カラムから値を DateTime としてフェッチします。

構文

```
public virtual DateTime GetDateTime (cname)
```

パラメータ

cname カラム名。

戻り値

DateTime 値。

1.12.21.2 GetDateTime(uint16) メソッド

カラムから値を DateTime としてフェッチします。

構文

```
public virtual DateTime GetDateTime (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

DateTime 値。

1.12.22 GetDouble メソッド

カラムから値を double としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual double	GetDouble(String^) [244 ページ]	カラムから値を double としてフェッチします。
public virtual double	GetDouble(uint16) [244 ページ]	カラムから値を double としてフェッチします。

このセクションの内容:

GetDouble(String^) メソッド [244 ページ]
カラムから値を double としてフェッチします。

GetDouble(uint16) メソッド [244 ページ]
カラムから値を double としてフェッチします。

1.12.22.1 GetDouble(String^) メソッド

カラムから値を double としてフェッチします。

構文

```
public virtual double GetDouble (cname)
```

パラメータ

cname カラム名。

戻り値

double としてのカラム値。

1.12.22.2 GetDouble(uint16) メソッド

カラムから値を double としてフェッチします。

構文

```
public virtual double GetDouble (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

double としてのカラム値。

1.12.23 GetFloat メソッド

カラムから値を float としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual float	GetFloat(String^) [245 ページ]	カラムから値を float としてフェッチします。
public virtual float	GetFloat(uint16) [246 ページ]	カラムから値を float としてフェッチします。

このセクションの内容:

GetFloat(String^) メソッド [245 ページ]
カラムから値を float としてフェッチします。

GetFloat(uint16) メソッド [246 ページ]
カラムから値を float としてフェッチします。

1.12.23.1 GetFloat(String^) メソッド

カラムから値を float としてフェッチします。

構文

```
public virtual float GetFloat (cname)
```

パラメータ

cname カラム名。

戻り値

float としてのカラム値。

1.12.23.2 GetFloat(uint16) メソッド

カラムから値を float としてフェッチします。

構文

```
public virtual float GetFloat (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

float としてのカラム値。

1.12.24 GetGuid メソッド

カラムから値を GUID としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual Guid	GetGuid(String^) [247 ページ]	カラムから値を GUID としてフェッチします。
public virtual Guid	GetGuid(uint16) [247 ページ]	カラムから値を GUID としてフェッチします。

このセクションの内容:

GetGuid(String^) メソッド [247 ページ]
カラムから値を GUID としてフェッチします。

GetGuid(uint16) メソッド [247 ページ]
カラムから値を GUID としてフェッチします。

1.12.24.1 GetGuid(String^) メソッド

カラムから値を GUID としてフェッチします。

構文

```
public virtual Guid GetGuid (cname)
```

パラメータ

cname カラム名。

戻り値

GUID 値。

1.12.24.2 GetGuid(uint16) メソッド

カラムから値を GUID としてフェッチします。

構文

```
public virtual Guid GetGuid (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

GUID 値。

1.12.25 GetInt16 メソッド

カラムから値を 16 ビット整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual int16	GetInt16(String^) [248 ページ]	カラムから値を 16 ビット整数としてフェッチします。
public virtual int16	GetInt16(uint16) [249 ページ]	カラムから値を 16 ビット整数としてフェッチします。

このセクションの内容:

1.12.25.1 GetInt16(String^) メソッド

カラムから値を 16 ビット整数としてフェッチします。

構文

```
public virtual int16 GetInt16 (cname)
```

パラメータ

cname カラム名。

戻り値

整数としてのカラム値。

1.12.25.2 GetInt16(uint16) メソッド

カラムから値を 16 ビット整数としてフェッチします。

構文

```
public virtual int16 GetInt16 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

整数としてのカラム値。

1.12.26 GetInt32 メソッド

カラムから値を整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual int	GetInt32(String^) [250 ページ]	カラムから値を整数としてフェッチします。
public virtual int	GetInt32(uint16) [250 ページ]	カラムから値を整数としてフェッチします。

このセクションの内容:

GetInt32(String^) メソッド [250 ページ]
カラムから値を整数としてフェッチします。

GetInt32(uint16) メソッド [250 ページ]
カラムから値を整数としてフェッチします。

1.12.26.1 GetInt32(String^) メソッド

カラムから値を整数としてフェッチします。

構文

```
public virtual int GetInt32 (cname)
```

パラメータ

cname カラム名。

戻り値

整数としてのカラム値。

1.12.26.2 GetInt32(uint16) メソッド

カラムから値を整数としてフェッチします。

構文

```
public virtual int GetInt32 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

整数としてのカラム値。

1.12.27 GetInt64 メソッド

カラムから値を 64 ビット整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual int64	GetInt64(String^) [251 ページ]	カラムから値を 64 ビット整数としてフェッチします。
public virtual int64	GetInt64(uint16) [252 ページ]	カラムから値を 64 ビット整数としてフェッチします。

このセクションの内容:

1.12.27.1 GetInt64(String^) メソッド

カラムから値を 64 ビット整数としてフェッチします。

構文

```
public virtual int64 GetInt64 (cname)
```

パラメータ

cname カラム名。

戻り値

整数としてのカラム値。

1.12.27.2 GetInt64(uint16) メソッド

カラムから値を 64 ビット整数としてフェッチします。

構文

```
public virtual int64 GetInt64 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

整数としてのカラム値。

1.12.28 GetRowCount(unsigned int) メソッド

テーブルのローの数を取得します。

構文

```
public virtual unsigned int GetRowCount (threshold)
```

パラメータ

threshold カウントするローの数の制限。無限を表すには 0 を設定します。

戻り値

テーブル内のローの数。

備考

このメソッドは、次の文を実行することと同義です。

```
SELECT COUNT(*) FROM table
```

1.12.29 GetState() メソッド

カーソルの内部ステータスを取得します。

構文

```
public virtual uint8 GetState ()
```

戻り値

カーソルのステータス

1.12.30 GetString メソッド

カラムから値を文字列としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual String	GetString(String^) [254 ページ]	カラムから値を文字列としてフェッチします。
public virtual String	GetString(uint16) [254 ページ]	カラムから値を文字列としてフェッチします。

このセクションの内容:

GetString(String^) メソッド [254 ページ]
カラムから値を文字列としてフェッチします。

GetString(uint16) メソッド [254 ページ]
カラムから値を文字列としてフェッチします。

1.12.30.1 GetString(String^) メソッド

カラムから値を文字列としてフェッチします。

構文

```
public virtual String GetString (cname)
```

パラメータ

cname カラム名。

戻り値

文字列値。

1.12.30.2 GetString(uint16) メソッド

カラムから値を文字列としてフェッチします。

構文

```
public virtual String GetString (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

文字列値。

1.12.31 GetStringLength メソッド

カラムの値の文字列長さを取得します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual int64	GetStringLength(String^) [255 ページ]	カラムの値の文字列長さを取得します。
public virtual int64	GetStringLength(uint16) [256 ページ]	カラムの値の文字列長さを取得します。

このセクションの内容:

GetStringLength(String^) メソッド [255 ページ]
カラムの値の文字列長さを取得します。

GetStringLength(uint16) メソッド [256 ページ]
カラムの値の文字列長さを取得します。

1.12.31.1 GetStringLength(String^) メソッド

カラムの値の文字列長さを取得します。

構文

```
public virtual int64 GetStringLength (cname)
```

パラメータ

cname カラム名。

戻り値

文字列型のカラムの値の文字数。

1.12.31.2 GetStringLength(uint16) メソッド

カラムの値の文字列長さを取得します。

構文

```
public virtual int64 GetStringLength (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

文字列型のカラムの値の文字数。

1.12.32 GetTableSchema メソッド

テーブルに関するスキーマ情報の取得に使用できる TableSchema オブジェクトを返します。

構文

```
public TableSchema GetTableSchema ()
```

戻り値

テーブルに関するスキーマ情報の取得に使用できる TableSchema オブジェクト。

1.12.33 GetUInt16 メソッド

カラムから値を 16 ビット符号なし整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual uint16	GetUInt16(String^) [257 ページ]	カラムから値を 16 ビット符号なし整数としてフェッチします。
public virtual uint16	GetUInt16(uint16) [258 ページ]	カラムから値を 16 ビット符号なし整数としてフェッチします。

このセクションの内容:

1.12.33.1 GetUInt16(String^) メソッド

カラムから値を 16 ビット符号なし整数としてフェッチします。

構文

```
public virtual uint16 GetUInt16 (cname)
```

パラメータ

cname カラム名。

戻り値

符号なし整数としてのカラム値。

1.12.33.2 GetUInt16(uint16) メソッド

カラムから値を 16 ビット符号なし整数としてフェッチします。

構文

```
public virtual uint16 GetUInt16 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

符号なし整数としてのカラム値。

1.12.34 GetUInt32 メソッド

カラムから値を 32 ビット符号なし整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual unsigned int	GetUInt32(String^) [259 ページ]	カラムから値を 32 ビット符号なし整数としてフェッチします。
public virtual unsigned int	GetUInt32(uint16) [259 ページ]	カラムから値を 32 ビット符号なし整数としてフェッチします。

このセクションの内容:

GetUInt32(String^) メソッド [259 ページ]
カラムから値を 32 ビット符号なし整数としてフェッチします。

GetUInt32(uint16) メソッド [259 ページ]
カラムから値を 32 ビット符号なし整数としてフェッチします。

1.12.34.1 GetUInt32(String^) メソッド

カラムから値を 32 ビット符号なし整数としてフェッチします。

構文

```
public virtual unsigned int GetUInt32 (cname)
```

パラメータ

cname カラム名。

戻り値

符号なし整数としてのカラム値。

1.12.34.2 GetUInt32(uint16) メソッド

カラムから値を 32 ビット符号なし整数としてフェッチします。

構文

```
public virtual unsigned int GetUInt32 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

符号なし整数としてのカラム値。

1.12.35 GetUInt64 メソッド

カラムから値を 64 ビット符号なし整数としてフェッチします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual uint64	GetUInt64(String^) [260 ページ]	カラムから値を 64 ビット符号なし整数としてフェッチします。
public virtual uint64	GetUInt64(uint16) [261 ページ]	カラムから値を 64 ビット符号なし整数としてフェッチします。

このセクションの内容:

1.12.35.1 GetUInt64(String^) メソッド

カラムから値を 64 ビット符号なし整数としてフェッチします。

構文

```
public virtual uint64 GetUInt64 (cname)
```

パラメータ

cname カラム名。

戻り値

符号なし整数としてのカラム値。

1.12.35.2 GetUInt64(uint16) メソッド

カラムから値を 64 ビット符号なし整数としてフェッチします。

構文

```
public virtual uint64 GetUInt64 (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

符号なし整数としてのカラム値。

1.12.36 Insert() メソッド

新しいローをテーブルに挿入します。

構文

```
public void Insert ()
```

1.12.37 InsertBegin() メソッド

設定されたカラムに対する挿入モードを選択します。

構文

```
public void InsertBegin ()
```

備考

Set メソッド呼び出しによって代わりの値が指定されない場合、すべてのカラムは挿入時に初期値に設定されます。

1.12.38 IsNull メソッド

カラム値が NULL かどうかをチェックします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual bool	IsNull(String^) [262 ページ]	カラム値が NULL かどうかをチェックします。
public virtual bool	IsNull(uint16) [263 ページ]	カラム値が NULL かどうかをチェックします。

このセクションの内容:

1.12.38.1 IsNull(String^) メソッド

カラム値が NULL かどうかをチェックします。

構文

```
public virtual bool IsNull (cname)
```

パラメータ

cname カラム名。

戻り値

カラム値が NULL の場合は true。

1.12.38.2 IsNull(uint16) メソッド

カラム値が NULL かどうかをチェックします。

構文

```
public virtual bool IsNull (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラム値が NULL の場合は true。

1.12.39 Last() メソッド

カーソルを最後のローに移動します。

構文

```
public virtual void Last ()
```

1.12.40 Lookup(uint16) メソッド

現在のインデックスに基づいて、テーブルを順方向にスキャンしてルックアップを実行します。

構文

```
public bool Lookup (ncols)
```

パラメータ

ncols 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。

戻り値

ルックアップ後のカーソル位置が最後のローの後ろに設定された場合は false。

備考

検索する値を指定するには、インデックスのカラムごとに値を設定します。カーソルは、インデックスの値に一致するか、それより少ない値の最後のローで停止します。複合インデックスの場合、ncols パラメータはルックアップで使用するカラムの数を指定します。

1.12.41 LookupBackward(uint16) メソッド

現在のインデックスに基づいて、テーブルを逆方向にスキャンしてルックアップを実行します。

構文

```
public bool LookupBackward (ncols)
```

パラメータ

ncols 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。

戻り値

ルックアップ後のカーソル位置が最初のローの前に設定された場合は false。

備考

検索する値を指定するには、インデックスのカラムごとに値を設定します。カーソルは、インデックスの値に一致するか、それより少ない値の最後のローで停止します。複合インデックスの場合、ncols パラメータはルックアップで使用するカラムの数を指定します。

1.12.42 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。

構文

```
public void LookupBegin ()
```

備考

テーブルを開くのに使用されたインデックス内のカラムのみを設定できます。インデックスを使用しないでテーブルを開いた場合は、このメソッドを呼び出せません。

1.12.43 LookupForward(uint16) メソッド

現在のインデックスに基づいて、テーブルを順方向にスキャンしてルックアップを実行します。

構文

```
public bool LookupForward (ncols)
```

パラメータ

ncols 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。

戻り値

ルックアップ後のカーソル位置が最後のローの後ろに設定された場合は false。

備考

検索する値を指定するには、インデックスのカラムごとに値を設定します。カーソルは、インデックスの値に一致するか、それより少ない値の最後のローで停止します。複合インデックスの場合、ncols パラメータはルックアップで使用するカラムの数を指定します。

1.12.44 Next() メソッド

カーソルをロー 1 つ分進めます。

構文

```
public virtual bool Next ()
```

戻り値

カーソルが正常に進められる場合は、true。カーソルが次のローに正常に進められる場合は、引き続きエラーが通知される可能性があります。たとえば SELECT 式の評価中に変換エラーが発生する可能性があります。この場合、カラム値を取得するときにもエラーが返されます。カーソルを進められなかった場合は、false が返されます。たとえば、次のローが存在しないなどです。この場合、移動後のカーソル位置は最後のローの後ろに設定されます。

1.12.45 Previous() メソッド

カーソルをロー 1 つ分戻します。

構文

```
public virtual bool Previous ()
```

戻り値

カーソルをロー 1 つ分戻せた場合は、true。カーソルを戻せなかった場合は、false。移動後のカーソル位置は、最初のローの前に設定されます。

1.12.46 Relative(int) メソッド

カーソルを、現在のカーソルの位置から、offset で指定したロー数分移動します。

構文

```
public virtual void Relative (offset)
```

パラメータ

offset 移動するローの数。

1.12.47 SetBool メソッド

カラムを boolean 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetBool(String^, bool) [267 ページ]	カラムを boolean 値に設定します。
public virtual void	SetBool(uint16, bool) [268 ページ]	カラムを boolean 値に設定します。

このセクションの内容:

SetBool(String^, bool) メソッド [267 ページ]
カラムを boolean 値に設定します。

SetBool(uint16, bool) メソッド [268 ページ]
カラムを boolean 値に設定します。

1.12.47.1 SetBool(String^, bool) メソッド

カラムを boolean 値に設定します。

構文

```
public virtual void SetBool (cname, value)
```

パラメータ

cname カラム名。
value boolean 値。

1.12.47.2 SetBool(uint16, bool) メソッド

カラムを boolean 値に設定します。

構文

```
public virtual void SetBool (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value boolean 値。

1.12.48 SetByte メソッド

カラムを byte 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetByte(String^, uint8) [269 ページ]	カラムを byte 値に設定します。
public virtual void	SetByte(uint16, uint8) [269 ページ]	カラムを byte 値に設定します。

このセクションの内容:

SetByte(String^, uint8) メソッド [269 ページ]
カラムを byte 値に設定します。

SetByte(uint16, uint8) メソッド [269 ページ]
カラムを byte 値に設定します。

1.12.48.1 SetByte(String^, uint8) メソッド

カラムを byte 値に設定します。

構文

```
public virtual void SetByte (cname, value)
```

パラメータ

cname カラム名。
value byte 値。

1.12.48.2 SetByte(uint16, uint8) メソッド

カラムを byte 値に設定します。

構文

```
public virtual void SetByte (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value byte 値。

1.12.49 SetBytes メソッド

カラムを binary 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetBytes(String^, const Array< uint8 >^, int) [270 ページ]	カラムを binary 値に設定します。
public virtual void	SetBytes(uint16, const Array< uint8 >^, int) [271 ページ]	カラムを binary 値に設定します。

このセクションの内容:

SetBytes(String^, const Array< uint8 >^, int) メソッド [270 ページ]
カラムを binary 値に設定します。

SetBytes(uint16, const Array< uint8 >^, int) メソッド [271 ページ]
カラムを binary 値に設定します。

1.12.49.1 SetBytes(String^, const Array< uint8 >^, int) メソッド

カラムを binary 値に設定します。

構文

```
public virtual void SetBytes (cname, value, length)
```

パラメータ

cname カラム名。
value バイナリ値。NULL を渡すことは、SetNull メソッドを呼び出すことと同じです。
length byte 配列値から取得するバイト単位の長さ。

1.12.49.2 SetBytes(uint16, const Array< uint8 >^, int) メソッド

カラムを binary 値に設定します。

構文

```
public virtual void SetBytes (cid, value, length)
```

パラメータ

cid 0 から始まるカラムの序数。
value バイナリ値。NULL を渡すことは、SetNull メソッドを呼び出すことと同じです。
length byte 配列値から取得するバイト単位の長さ。

1.12.50 SetDateTime メソッド

カラムを DateTime 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetDateTime(String^, DateTime) [272 ページ]	カラムを DateTime 値に設定します。
public virtual void	SetDateTime(uint16, DateTime) [272 ページ]	カラムを DateTime 値に設定します。

このセクションの内容:

SetDateTime(String^, DateTime) メソッド [272 ページ]
カラムを DateTime 値に設定します。

SetDateTime(uint16, DateTime) メソッド [272 ページ]
カラムを DateTime 値に設定します。

1.12.50.1 SetDateTime(String^, DateTime) メソッド

カラムを DateTime 値に設定します。

構文

```
public virtual void SetDateTime (cname, value)
```

パラメータ

cname カラム名。
value DateTime 値。

1.12.50.2 SetDateTime(uint16, DateTime) メソッド

カラムを DateTime 値に設定します。

構文

```
public virtual void SetDateTime (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value DateTime 値。

1.12.51 SetDefault メソッド

カラムをそのデフォルト値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetDefault(String^) [273 ページ]	カラムをそのデフォルト値に設定します。
public virtual void	SetDefault(uint16) [273 ページ]	カラムをそのデフォルト値に設定します。

このセクションの内容:

1.12.51.1 SetDefault(String^) メソッド

カラムをそのデフォルト値に設定します。

構文

```
public virtual void SetDefault (cname)
```

パラメータ

cname カラム名。

1.12.51.2 SetDefault(uint16) メソッド

カラムをそのデフォルト値に設定します。

構文

```
public virtual void SetDefault (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

1.12.52 SetDouble メソッド

カラムを double 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetDouble(String^, double) [274 ページ]	カラムを double 値に設定します。
public virtual void	SetDouble(uint16, double) [275 ページ]	カラムを double 値に設定します。

このセクションの内容:

SetDouble(String^, double) メソッド [274 ページ]
カラムを double 値に設定します。

SetDouble(uint16, double) メソッド [275 ページ]
カラムを double 値に設定します。

1.12.52.1 SetDouble(String^, double) メソッド

カラムを double 値に設定します。

構文

```
public virtual void SetDouble (cname, value)
```

パラメータ

cname カラム名。
value double 値。

1.12.52.2 SetDouble(uint16, double) メソッド

カラムを double 値に設定します。

構文

```
public virtual void SetDouble (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value double 値。

1.12.53 SetFloat メソッド

カラムを float 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetFloat(String^, float) [276 ページ]	カラムを float 値に設定します。
public virtual void	SetFloat(uint16, float) [276 ページ]	カラムを float 値に設定します。

このセクションの内容:

SetFloat(String^, float) メソッド [276 ページ]
カラムを float 値に設定します。

SetFloat(uint16, float) メソッド [276 ページ]
カラムを float 値に設定します。

1.12.53.1 SetFloat(String^, float) メソッド

カラムを float 値に設定します。

構文

```
public virtual void SetFloat (cname, value)
```

パラメータ

cname カラム名。
value float 値。

1.12.53.2 SetFloat(uint16, float) メソッド

カラムを float 値に設定します。

構文

```
public virtual void SetFloat (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value float 値。

1.12.54 SetGuid メソッド

カラムを GUID 値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetGuid(uint16, Guid) [277 ページ]	カラムを GUID 値に設定します。
public virtual void	SetGuid(uint16, Guid) [278 ページ]	カラムを GUID 値に設定します。

このセクションの内容:

1.12.54.1 SetGuid(uint16, Guid) メソッド

カラムを GUID 値に設定します。

構文

```
public virtual void SetGuid (cname, value)
```

パラメータ

cname カラム名。
value GUID 値。

1.12.54.2 SetGuid(uint16, Guid) メソッド

カラムを GUID 値に設定します。

構文

```
public virtual void SetGuid (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value GUID 値。

1.12.55 SetInt16 メソッド

カラムを整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetInt16(String^, int16) [279 ページ]	カラムを整数値に設定します。
public virtual void	SetInt16(uint16, int16) [279 ページ]	カラムを整数値に設定します。

このセクションの内容:

SetInt16(String^, int16) メソッド [279 ページ]
カラムを整数値に設定します。

SetInt16(uint16, int16) メソッド [279 ページ]
カラムを整数値に設定します。

1.12.55.1 SetInt16(String^, int16) メソッド

カラムを整数値に設定します。

構文

```
public virtual void SetInt16 (cname, value)
```

パラメータ

cname カラム名。
value 符号付き整数値。

1.12.55.2 SetInt16(uint16, int16) メソッド

カラムを整数値に設定します。

構文

```
public virtual void SetInt16 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号付き整数値。

1.12.56 SetInt32 メソッド

カラムを整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetInt32(String^, int) [280 ページ]	カラムを整数値に設定します。
public virtual void	SetInt32(uint16, int) [281 ページ]	カラムを整数値に設定します。

このセクションの内容:

1.12.56.1 SetInt32(String^, int) メソッド

カラムを整数値に設定します。

構文

```
public virtual void SetInt32 (cname, value)
```

パラメータ

cname カラム名。
value 符号付き整数値。

1.12.56.2 SetInt32(uint16, int) メソッド

カラムを整数値に設定します。

構文

```
public virtual void SetInt32 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号付き整数値。

1.12.57 SetInt64 メソッド

カラムを整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetInt64(String^, int64) [282 ページ]	カラムを整数値に設定します。
public virtual void	SetInt64(uint16, int64) [282 ページ]	カラムを整数値に設定します。

このセクションの内容:

SetInt64(String^, int64) メソッド [282 ページ]
カラムを整数値に設定します。

SetInt64(uint16, int64) メソッド [282 ページ]
カラムを整数値に設定します。

1.12.57.1 SetInt64(String^, int64) メソッド

カラムを整数値に設定します。

構文

```
public virtual void SetInt64 (cname, value)
```

パラメータ

cname カラム名。
value 符号付き整数値。

1.12.57.2 SetInt64(uint16, int64) メソッド

カラムを整数値に設定します。

構文

```
public virtual void SetInt64 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号付き整数値。

1.12.58 SetNull メソッド

カラムを NULL に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetNull(String^) [283 ページ]	カラムを NULL に設定します。
public virtual void	SetNull(uint16) [283 ページ]	カラムを NULL に設定します。

このセクションの内容:

1.12.58.1 SetNull(String^) メソッド

カラムを NULL に設定します。

構文

```
public virtual void SetNull (cname)
```

パラメータ

cname カラム名。

1.12.58.2 SetNull(uint16) メソッド

カラムを NULL に設定します。

構文

```
public virtual void SetNull (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

1.12.59 SetString メソッド

カラムを文字列値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetString(String^, String^) [284 ページ]	カラムを文字列値に設定します。
public virtual void	SetString(uint16, String^) [285 ページ]	カラムを文字列値に設定します。

このセクションの内容:

SetString(String^, String^) メソッド [284 ページ]
カラムを文字列値に設定します。

SetString(uint16, String^) メソッド [285 ページ]
カラムを文字列値に設定します。

1.12.59.1 SetString(String^, String^) メソッド

カラムを文字列値に設定します。

構文

```
public virtual void SetString (cname, value)
```

パラメータ

cname カラム名。
value 文字列値。

1.12.59.2 SetString(uint16, String^) メソッド

カラムを文字列値に設定します。

構文

```
public virtual void SetString (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 文字列値。

1.12.60 SetUInt16 メソッド

カラムを符号なし 16 ビット整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetUInt16(String^, uint16) [286 ページ]	カラムを符号なし 16 ビット整数値に設定します。
public virtual void	SetUInt16(uint16, uint16) [286 ページ]	カラムを符号なし 16 ビット整数値に設定します。

このセクションの内容:

SetUInt16(String^, uint16) メソッド [286 ページ]
カラムを符号なし 16 ビット整数値に設定します。

SetUInt16(uint16, uint16) メソッド [286 ページ]
カラムを符号なし 16 ビット整数値に設定します。

1.12.60.1 SetUInt16(String^, uint16) メソッド

カラムを符号なし 16 ビット整数値に設定します。

構文

```
public virtual void SetUInt16 (cname, value)
```

パラメータ

cname カラム名。
value 符号なし整数値。

1.12.60.2 SetUInt16(uint16, uint16) メソッド

カラムを符号なし 16 ビット整数値に設定します。

構文

```
public virtual void SetUInt16 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号なし整数値。

1.12.61 SetUInt32 メソッド

カラムを符号なし 32 ビット整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetUInt32(String^, unsigned int) [287 ページ]	カラムを符号なし 32 ビット整数値に設定します。
public virtual void	SetUInt32(uint16, unsigned int) [288 ページ]	カラムを符号なし 32 ビット整数値に設定します。

このセクションの内容:

1.12.61.1 SetUInt32(String^, unsigned int) メソッド

カラムを符号なし 32 ビット整数値に設定します。

構文

```
public virtual void SetUInt32 (cname, value)
```

パラメータ

cname カラム名。

value 符号なし整数値。

1.12.61.2 SetUInt32(uint16, unsigned int) メソッド

カラムを符号なし 32 ビット整数値に設定します。

構文

```
public virtual void SetUInt32 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号なし整数値。

1.12.62 SetUInt64 メソッド

カラムを符号なし 64 ビット整数値に設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public virtual void	SetUInt64(String^, uint64) [289 ページ]	カラムを符号なし 64 ビット整数値に設定します。
public virtual void	SetUInt64(uint16, uint64) [289 ページ]	カラムを符号なし 64 ビット整数値に設定します。

このセクションの内容:

SetUInt64(String^, uint64) メソッド [289 ページ]
カラムを符号なし 64 ビット整数値に設定します。

SetUInt64(uint16, uint64) メソッド [289 ページ]
カラムを符号なし 64 ビット整数値に設定します。

1.12.62.1 SetUInt64(String^, uint64) メソッド

カラムを符号なし 64 ビット整数値に設定します。

構文

```
public virtual void SetUInt64 (cname, value)
```

パラメータ

cname カラム名。
value 符号なし整数値。

1.12.62.2 SetUInt64(uint16, uint64) メソッド

カラムを符号なし 64 ビット整数値に設定します。

構文

```
public virtual void SetUInt64 (cid, value)
```

パラメータ

cid 0 から始まるカラムの序数。
value 符号なし整数値。

1.12.63 TruncateTable() メソッド

テーブルをトランケートし、STOP SYNCHRONIZATION DELETE を一時的にアクティブにします。

構文

```
public void TruncateTable ()
```

1.12.64 Update() メソッド

現在の行を更新します。

構文

```
public virtual void Update ()
```

1.12.65 UpdateBegin() メソッド

カラムの設定に使用される更新モードを選択します。

構文

```
public virtual void UpdateBegin ()
```

備考

Ultra Light が更新モードの場合、プライマリキー内のカラムは修正できません。Connection.Commit() または Rollback() を使用してトランザクションをコミットまたはロールバックします。

1.13 TableSchema クラス

Ultra Light のテーブルのスキーマを表します。DatabaseSchema.GetTables()、GetTableSchema() または Table.GetTableSchema() は TableSchema を返します。

ネームスペース

```
UltraLite
```

構文

```
public ref class  sealed  : CursorSchema
```

メンバー

TableSchema のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public void	Close() [293 ページ]	このオブジェクトを破棄します。
public virtual uint16	GetColumnCount() [293 ページ]	結果セットまたはテーブル内のカラム数を取得します。
public String	GetColumnDefault(uint16) [294 ページ]	カラムのデフォルト値が存在する場合は取得します。
public uint16	GetColumnDefaultType(uint16) [294 ページ]	カラムのデフォルトの型を取得します。
public virtual uint16	GetColumnID(String^) [295 ページ]	0 から始まるカラム ID を名前から取得します。
public virtual String	GetColumnName(uint16, uint16type) [295 ページ]	0 から始まる ID を指定してカラムの名前を取得します。
public virtual uint16	GetColumnPrecision(uint16) [296 ページ]	数値カラムの精度を取得します。
public virtual uint16	GetColumnScale(uint16) [296 ページ]	数値カラムの位取りを取得します。
public virtual int	GetColumnSize(uint16) [297 ページ]	カラムのサイズを取得します。
public virtual uint16	GetColumnSQLType(uint16) [297 ページ]	カラムの SQL の型を取得します。
public virtual uint16	GetColumnType(uint16) [298 ページ]	カラムの記憶タイプまたはホスト変数の型を取得します。
public uint64	GetGlobalAutoincPartitionSize(uint16) [298 ページ]	分割サイズを取得します。
public uint16	GetIndexCount() [299 ページ]	テーブル内のインデックス数を取得します。
public IIterator< IndexSchema^>	GetIndexes() [299 ページ]	テーブル内のすべてのインデックス (スキーマ) の集まりに対する反復子を取得します。
public IndexSchema	GetIndexSchema(String^) [300 ページ]	名前を指定すると、インデックスのスキーマを取得します。
public String	GetName() [300 ページ]	テーブルの名前を取得します。
public String	GetOptimalIndex(uint16) [301 ページ]	カラム値を検索するのに最適なインデックスを特定します。
public IndexSchema	GetPrimaryKey() [301 ページ]	テーブルのプライマリキーを取得します。
public String	GetPublicationPredicate(String^) [301 ページ]	文字列としてのパブリケーション述部を取得します。
public uint16	GetTableSyncType() [302 ページ]	テーブルの同期タイプを取得します。
public bool	InPublication(String^) [302 ページ]	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。

変更子とタイプ	メソッド	説明
public virtual bool	IsAliased(uint16) [303 ページ]	結果セット内のカラムにエイリアスが付与されているかどうかを示します。
public bool	IsColumnInIndex(uint16, String^) [303 ページ]	カラムが、指定されたインデックスに含まれているかどうかをチェックします。
public bool	IsColumnNullable(uint16) [304 ページ]	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。

このセクションの内容:

CloseObject() メソッド [293 ページ]
このオブジェクトを破棄します。

GetColumnCount() メソッド [293 ページ]
結果セットまたはテーブル内のカラム数を取得します。

GetColumnDefault(uint16) メソッド [294 ページ]
カラムのデフォルト値が存在する場合は取得します。

GetColumnDefaultType(uint16) メソッド [294 ページ]
カラムのデフォルトの型を取得します。

GetColumnID(String^) メソッド [295 ページ]
0 から始まるカラム ID を名前から取得します。

GetColumnName(uint16, uint16type) メソッド [295 ページ]
0 から始まる ID を指定してカラムの名前を取得します。

GetColumnPrecision(uint16) メソッド [296 ページ]
数値カラムの精度を取得します。

GetColumnScale(uint16) メソッド [296 ページ]
数値カラムの位取りを取得します。

GetColumnSize(uint16) メソッド [297 ページ]
カラムのサイズを取得します。

GetColumnSQLType(uint16) メソッド [297 ページ]
カラムの SQL の型を取得します。

GetColumnType(uint16) メソッド [298 ページ]
カラムの記憶タイプまたはホスト変数の型を取得します。

GetGlobalAutoincPartitionSize(uint16) メソッド [298 ページ]
分割サイズを取得します。

GetIndexCount() メソッド [299 ページ]
テーブル内のインデックス数を取得します。

GetIndexes() メソッド [299 ページ]
テーブル内のすべてのインデックス (スキーマ) の集まりに対する反復子を取得します。

GetIndexSchema(String^) メソッド [300 ページ]
名前を指定すると、インデックスのスキーマを取得します。

GetName() メソッド [300 ページ]
テーブルの名前を取得します。

GetOptimalIndex(uint16) メソッド [301 ページ]
カラム値を検索するのに最適なインデックスを特定します。

GetPrimaryKey() メソッド [301 ページ]
テーブルのプライマリキーを取得します。

GetPublicationPredicate(String^) メソッド [301 ページ]
文字列としてのパブリケーション述部を取得します。

GetTableSyncType() メソッド [302 ページ]
テーブルの同期タイプを取得します。

InPublication(String^) メソッド [302 ページ]
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。

IsAliased(uint16) メソッド [303 ページ]
結果セット内のカラムにエイリアスが付与されているかどうかを示します。

IsColumnInIndex(uint16, String^) メソッド [303 ページ]
カラムが、指定されたインデックスに含まれているかどうかをチェックします。

IsColumnNullable(uint16) メソッド [304 ページ]
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。

1.13.1 CloseObject() メソッド

このオブジェクトを破棄します。

構文

```
public void CloseObject ()
```

1.13.2 GetColumnCount() メソッド

結果セットまたはテーブル内のカラム数を取得します。

構文

```
public virtual uint16 GetColumnCount ()
```

戻り値

結果セットまたはテーブル内のカラム数。

1.13.3 GetColumnDefault(uint16) メソッド

カラムのデフォルト値が存在する場合は取得します。

構文

```
public String GetColumnDefault (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

デフォルト値。カラムにデフォルト値がない場合は、空の文字列を返します。

1.13.4 GetColumnDefaultType(uint16) メソッド

カラムのデフォルトの型を取得します。

構文

```
public uint16 GetColumnDefaultType (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラムのデフォルトの型。

1.13.5 GetColumnID(String^) メソッド

0 から始まるカラム ID を名前から取得します。

構文

```
public virtual uint16 GetColumnID (columnName)
```

パラメータ

columnName カラムの名前。

戻り値

カラム ID。

1.13.6 GetColumnName(uint16, uint16type) メソッド

0 から始まる ID を指定してカラムの名前を取得します。

構文

```
public virtual String GetColumnName (cid, type)
```

パラメータ

cid 0 から始まるカラムの序数。
type カラム名の必要な型。

戻り値

存在する場合は、カラム名を格納する文字列。存在しない場合は、エラーがスローされます。

備考

選択した型、および SELECT 文でのカラムの宣言方法によっては、カラム名が [table-name].[column-name] という形式で返されることがあります。

type パラメータは、どの型のカラム名を返すかを指定します。

1.13.7 GetColumnPrecision(uint16) メソッド

数値カラムの精度を取得します。

構文

```
public virtual uint16 GetColumnPrecision (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

数値カラムの精度。

1.13.8 GetColumnScale(uint16) メソッド

数値カラムの位取りを取得します。

構文

```
public virtual uint16 GetColumnScale (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

数値カラムの位取り。

1.13.9 GetColumnSize(uint16) メソッド

カラムのサイズを取得します。

構文

```
public virtual int GetColumnSize (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラムサイズ。

1.13.10 GetColumnSQLType(uint16) メソッド

カラムの SQL の型を取得します。

構文

```
public virtual uint16 GetColumnSQLType (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

ColumnSQLType 列挙体または列が存在しない場合は SQLTYPE_BAD_INDEX からの値。

1.13.11 GetColumnType(uint16) メソッド

カラムの記憶タイプまたはホスト変数の型を取得します。

構文

```
public virtual uint16 GetColumnType (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラムが存在しない場合は、TYPE_BAD_INDEX。

1.13.12 GetGlobalAutoincPartitionSize(uint16) メソッド

分割サイズを取得します。

構文

```
public uint64 GetGlobalAutoincPartitionSize (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラムの分割サイズ。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。

1.13.13 GetIndexCount() メソッド

テーブル内のインデックス数を取得します。

構文

```
public uint16 GetIndexCount ()
```

戻り値

テーブル内のインデックス数。

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再表示します。

1.13.14 GetIndexes() メソッド

テーブル内のすべてのインデックス (スキーマ) の集まりに対する反復子を取得します。

構文

```
public IIterator< IndexSchema^> GetIndexes ()
```

戻り値

IndexSchema オブジェクトの集まりに対する反復子。

1.13.15 GetIndexSchema(String^) メソッド

名前を指定すると、インデックスのスキーマを取得します。

構文

```
public IndexSchema GetIndexSchema (indexName)
```

パラメータ

indexName インデックスの名前。

戻り値

指定されたインデックスの IndexSchema オブジェクト、または、このオブジェクトが存在しない場合は NULL。

1.13.16 GetName() メソッド

テーブルの名前を取得します。

構文

```
public String GetName ()
```

戻り値

テーブル名。

1.13.17 GetOptimalIndex(uint16) メソッド

カラム値を検索するのに最適なインデックスを特定します。

構文

```
public String GetOptimalIndex (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

インデックスの名前、またはカラムがインデックス付けされていない場合は NULL。

1.13.18 GetPrimaryKey() メソッド

テーブルのプライマリキーを取得します。

構文

```
public IndexSchema GetPrimaryKey ()
```

戻り値

テーブルのプライマリキーの IndexSchema オブジェクト。

1.13.19 GetPublicationPredicate(String^) メソッド

文字列としてのパブリケーション述部を取得します。

構文

```
public String GetPublicationPredicate (pubName)
```

パラメータ

pubName 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のパブリケーション述部文字列。

1.13.20 GetTableSyncType() メソッド

テーブルの同期タイプを取得します。

構文

```
public uint16 GetTableSyncType ()
```

戻り値

テーブル同期タイプ。

備考

このメソッドは、テーブルが同期に参加する方法、および CREATE TABLE 文の SYNCHRONIZE 制約句によってテーブルを作成するときの定義方法を示します。

1.13.21 InPublication(String^) 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

```
public bool InPublication (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に含まれている場合は true、含まれていない場合は false。

1.13.22 IsAliased(uint16) メソッド

結果セット内のカラムにエイリアスが付与されているかどうかを示します。

構文

```
public virtual bool IsAliased (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラムのエイリアスが使用されている場合は true、そうでない場合は false。

1.13.23 IsColumnInIndex(uint16, String^) メソッド

カラムが、指定されたインデックスに含まれているかどうかをチェックします。

構文

```
public bool IsColumnInIndex (cid, indexName)
```

パラメータ

cid 0 から始まるカラムの序数。
indexName インデックスの名前。

戻り値

カラムがインデックスに含まれている場合は true、含まれていない場合は false。

1.13.24 IsColumnNullable(uint16) 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

```
public bool IsColumnNullable (cid)
```

パラメータ

cid 0 から始まるカラムの序数。

戻り値

カラムが NULL 入力可能である場合は true、そうでない場合は false を返します。

1.14 FileTransferObserver(FileTransferStatus^) デリゲート

ファイル転送のさまざまな段階で呼び出されるデリゲートの定義。

構文

```
public delegate bool FileTransferObserver (status);
```

1.15 SyncObserver(SyncStatus^) デリゲート

同期のさまざまな段階で呼び出されるデリゲートの定義。

構文

```
public delegate bool SyncObserver (status);
```

1.16 ValidateCallback(ValidateData^) デリゲート

検証フィードバックに対して呼び出されるデリゲートの定義。

構文

```
public delegate bool ValidateCallback (vdata);
```

1.17 AuthStatusCode 列挙体

Mobile Link 認証ステータスコードを指定します。SyncResult および FileTransferResult には AuthStatusCode 値を返す AuthStatus 項目があります。

構文

```
enum AuthStatusCode
```

メンバー

メンバー名	説明	値
UNKNOWN	認証ステータスが不明です。接続がまだ同期していない可能性があります。	0
VALID	ユーザ ID とパスワードは、同期時には有効でした。	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。	2

メンバー名	説明	値
EXPIRED	認証に失敗しました。ユーザ ID またはパスワードの有効期限が切れています。	3
INVALID	認証に失敗しました。不正なユーザ ID またはパスワードです。	4
IN_USE	認証に失敗しました。ユーザ ID はすでに使用されています。	5

1.18 ColumnDefaultType 列挙体

カラムのデフォルト型を識別します。

構文

```
enum ColumnDefaultType
```

メンバー

メンバー名	説明
column_default_none	カラムにデフォルト値はありません。
column_default_autoincrement	カラムのデフォルトは AUTOINCREMENT です。
column_default_global_autoincrement	カラムのデフォルトは GLOBAL AUTOINCREMENT です。
column_default_current_timestamp	カラムのデフォルトは CURRENT TIMESTAMP です。
column_default_current_utc_timestamp	カラムのデフォルトは CURRENT UTC TIMESTAMP です。
column_default_current_time	カラムのデフォルトは CURRENT TIME です。
column_default_current_date	カラムのデフォルトは CURRENT DATE です。
column_default_newid	カラムのデフォルトは NEWID() です。
column_default_other	カラムのデフォルトはユーザ指定の定数です。

戻り値

TableSchema::GetColumnDefaultType

1.19 ColumnNameType 列挙体

結果セットの記述時にカラムの名前を取得する方法を制御する値を指定します。

構文

```
enum ColumnNameType
```

メンバー

メンバー名	説明
name_type_sql	SELECT 文の場合は、エイリアスまたは相関名を返します。 テーブルの場合は、カラム名を返します。
name_type_sql_column_only	SELECT 文の場合は、エイリアスまたは相関名を返し、指定されたテーブルの名前を除外します。 テーブルの場合は、カラム名を返します。
name_type_base_table	確認できる場合は、基本となるテーブルの名前を返します。 このテーブルがデータベーススキーマに存在しない場合は、空の文字列を返します。
name_type_base_column	確認できる場合は、基本となるカラムの名前を返します。 このカラムがデータベーススキーマに存在しない場合は、空の文字列を返します。
name_type_qualified	ULResultSetSchema.GetColumnName メソッドと一緒に使用したときに、基本となる修飾カラム名を確認できる場合は、このカラム名を返します。 返される名前は、次の値のいずれかであり、この順序で確認されます。 1. 表現される相関テーブル 2. 表現されるテーブルのカラムの名前 3. カラムのエイリアス名 4. 空の文字列

メンバー名	説明
name_type_base	GetColumnName メソッドと一緒に使用したときにテーブルの名前で修飾されたカラムの名前が返されることを示します。 取得されるカラム名がクエリのベーステーブルに関連付けられている場合は、ベーステーブル名がカラムの修飾子として使用されます (つまり、base_table_name.column_name 値が返されます)。取得されるカラム名がクエリの相関テーブルのカラムを表している場合は、相関名がカラムの修飾子として使用されます (つまり、correl_table_name.col_name 値が返されます)。カラムにエイリアスがある場合は、エイリアスを使用しているカラムの修飾名が返されますが、エイリアスは、修飾名の一部ではありません。それ以外の場合は、空の文字列が返されます。

備考

ResultSetSchema::GetColumnName

1.20 ColumnSQLType 列挙体

カラムの SQL タイプを指定します。CursorSchema.GetColumnSQLType() は ColumnSQLType を返します。

構文

```
enum ColumnSQLType
```

メンバー

メンバー名	説明
SQLTYPE_BAD_INDEX	指定されたインデックス位置のカラムが存在しないことを表します。
SQLTYPE_S_LONG	カラムに、signed long が含まれることを表します。
SQLTYPE_U_LONG	カラムに、unsigned long が含まれることを表します。
SQLTYPE_S_SHORT	カラムに、signed short が含まれることを表します。
SQLTYPE_U_SHORT	カラムに、unsigned short が含まれることを表します。
SQLTYPE_S_BIG	カラムに、64 ビット符号付き整数が含まれることを表します。
SQLTYPE_U_BIG	カラムに、64 ビット符号なし整数が含まれることを表します。

メンバー名	説明
SQLTYPE_TINY	カラムに、8ビット符号なし整数が含まれることを表します。
SQLTYPE_BIT	カラムに、1ビットフラグが含まれることを表します。
SQLTYPE_TIMESTAMP	カラムに、タイムスタンプ情報が含まれることを表します。
SQLTYPE_DATE	カラムに、日付情報が含まれることを表します。
SQLTYPE_TIME	カラムに、時刻の情報が含まれることを表します。
SQLTYPE_DOUBLE	カラムに、倍精度の浮動小数点数 (8 バイト) が含まれることを表します。
SQLTYPE_REAL	カラムに、単精度の浮動小数点数 (4 バイト) が含まれることを表します。
SQLTYPE_NUMERIC	カラムに、精度と桁数が指定された正確な数値データが含まれることを表します。
SQLTYPE_BINARY	カラムに、指定された最大長のバイナリデータが含まれることを表します。
SQLTYPE_CHAR	カラムに、指定された長さの文字列データが含まれることを表します。
SQLTYPE_LONGVARCHAR	カラムに、可変長の文字列データが含まれることを表します。
SQLTYPE_LONGBINARY	カラムに、可変長のバイナリデータが含まれることを表します。
SQLTYPE_UUID	カラムに UUID が含まれることを表します。
SQLTYPE_ST_GEOMETRY	カラムに、ポイント形式の空間データが含まれることを表します。
SQLTYPE_TIMESTAMP_WITH_TIME_ZONE	カラムに、タイムスタンプ情報とタイムゾーン情報が含まれることを表します。

備考

値は SQL カラム型に対応します。

1.21 ColumnStorageType 列挙体

カラムのホスト変数タイプを指定します。

構文

```
enum ColumnStorageType
```

メンバー

メンバー名	説明
TYPE_BAD_INDEX	無効な値を表します。
TYPE_S_LONG	ul_s_long (32 ビット符号付き整数) を表します。
TYPE_U_LONG	ul_u_long (32 ビット符号なし整数) を表します。
TYPE_S_SHORT	ul_s_short (16 ビット符号付き整数) を表します。
TYPE_U_SHORT	ul_u_short (16 ビット符号なし整数) を表します。
TYPE_S_BIG	ul_s_big (64 ビット符号付き整数) を表します。
TYPE_U_BIG	ul_u_big (64 ビット符号なし整数) を表します。
TYPE_TINY	ul_u_byte (8 ビット符号なし) を表します。
TYPE_BIT	ul_byte (8 ビット符号なし、1 ビット使用) を表します。
TYPE_DOUBLE	ul_double (double) を表します。
TYPE_REAL	ul_real (float) を表します。
TYPE_BINARY	ul_binary (2 バイト長の後にバイト配列) を表します。
TYPE_TIMESTAMP_STRUCT	DECL_DATETIME を表します。
TYPE_TCHAR	文字配列 (文字列バッファ) を表します。
TYPE_CHAR	char 配列 (文字列バッファ) を表します。
TYPE_WCHAR	ul_wchar (UTF16) 配列を表します。
TYPE_GUID	GUID 構造体を表します。

備考

これらの値は、カラムに必要なホスト変数の型を識別し、Ultra Light で値をフェッチする方法を示します。
CursorSchema.GetColumnType() は ColumnStorageType を返します。

1.22 CursorState 列挙体

可能な結果セットまたはカーソルステータスを指定します。Cursor.GetState() は CursorState を返します。

構文

```
enum CursorState
```

メンバー

メンバー名	説明
RS_STATE_ERROR	エラー。
RS_STATE_UNPREPARED	準備されていません。
RS_STATE_ON_ROW	有効なローの上。
RS_STATE_BEFORE_FIRST	最初のローの前。
RS_STATE_AFTER_LAST	最後のローの後。
RS_STATE_COMPLETED	閉じています。

1.23 ErrorCodes 列挙体

Ultra Light 例外を示す Platform::COMException HResult コードを指定します。

構文

```
enum ErrorCodes
```

メンバー

メンバー名	説明	値
E_ULTRALITE_ERROR	Ultra Light エラーを示す HResult コード。	(-10000)

備考

Ultra Light 固有のエラー情報を取得するには、GetLastError メソッドを使用します。

DatabaseManager::GetLastError Connection::GetLastError

1.24 IndexFlag 列挙体

インデックスのプロパティを識別するフラグ (ビットフィールド) を指定します。

構文

```
enum IndexFlag
```

メンバー

メンバー名	説明	値
index_flag_primary_key	インデックスがプライマリキーであることを表します。	0x0001
index_flag_unique_key	インデックスがプライマリキーであるか、一意性制約に対して作成されたインデックスである (NULL は許可されない) ことを表します。	0x0002
index_flag_unique_index	インデックスが UNIQUE フラグで作成された (またはプライマリキーである) ことを表します。	0x0004
index_flag_foreign_key	インデックスが外部キーであることを表します。	0x0010
index_flag_foreign_key_nullable	外部キーが NULL を許可することを表します。	0x0020
index_flag_foreign_key_check_on_commit	参照整合性チェックがコミット時に (挿入時や更新時ではなく) 実行されることを表します。	0x0040

備考

IndexSchema::GetIndexFlags

1.25 StreamType 列挙体

FileTransfer.Stream プロパティに使用可能な値を指定します。

構文

```
enum StreamType
```

メンバー

メンバー名	説明
TCPIP	TCP/IP ストリームタイプを表します。
HTTP	HTTP ストリームタイプを表します。
HTTPS	HTTPS ストリームタイプを表します。
TLS	TLS ストリームタイプを表します。

1.26 SyncState 列挙体

同期の現在の処理を示します。

構文

```
enum SyncState
```

メンバー

メンバー名	説明
STATE_STARTING	同期を開始しています。初期パラメータの検証が完了し、同期の結果が保存されます。
STATE_CONNECTING	Mobile Link サーバに接続しています。
STATE_RESUMING_DOWNLOAD	部分ダウンロードを再開しようとしたときに入るオプションの状態。 成功すると、同期が STATE_RECEIVING_TABLE 状態に進みます。再開できない場合は、STATE_ERROR になります。
STATE_SENDING_HEADER	同期接続が確立されました。初期データを送信しようとしています。
STATE_SENDING_CHECK_SYNC_REQUEST	テーブルを送信しようとしています。 前回のアップロードの状態が不明なため、ステータスをチェックする要求が送信されます。
STATE_WAITING_FOR_CHECK_SYNC_RESPONSE	サーバが同期チェック要求に応答するのを待機しています。
STATE_PROCESSING_CHECK_SYNC_RESPONSE	同期チェック要求への応答が受け取られ、処理されています。
STATE_SENDING_TABLE	テーブルを送信しようとしています。
STATE_SENDING_DATA	スキーマ情報またはローデータが送信されています。

メンバー名	説明
STATE_FINISHING_UPLOAD	アップロード処理が完了しました。処理情報をコミットしようとしています。
STATE_WAITING_FOR_UPLOAD_ACK	サーバがアップロードの受信を確認するのを待機しています。
STATE_PROCESSING_UPLOAD_ACK	サーバがアップロードの受信を確認しました。
STATE_WAITING_FOR_DOWNLOAD	サーバがダウンロードの送信を開始するのを待機しています。
STATE_RECEIVING_TABLE	テーブルを受信しようとしています。
STATE_RECEIVING_DATA	最後に識別されたテーブルのデータを受信しています。
STATE_COMMITTING_DOWNLOAD	ダウンロード処理が完了しました。ダウンロードされたローをコミットしようとしています。
STATE_ROLLING_BACK_DOWNLOAD	ダウンロード中にエラーが発生し、ダウンロードがロールバックされています。
STATE_SENDING_DOWNLOAD_ACK	ダウンロード完了の確認を送信しています。
STATE_DISCONNECTING	Mobile Link サーバとの接続を切断しようとしています。
STATE_DONE	同期は正常に完了しました。
STATE_ERROR	同期は完了しましたが、エラーが発生しました。

備考

一覧の順序は同期ステータスが発生する順序とは一致しません。SyncStatus.State には SyncState が含まれます。

1.27 TableSyncType 列挙体

テーブルの同期タイプを識別します。

構文

```
enum TableSyncType
```

メンバー

メンバー名	説明
table_sync_on	変更されたすべてのローが同期される (デフォルトの動作) ことを示します。 このイニシャライザは、CREATE TABLE 文の SYNCHRONIZE ON 句に対応します。
table_sync_off	テーブルが同期されないことを示します。 このイニシャライザは、CREATE TABLE 文の SYNCHRONIZE OFF 句に対応します。
table_sync_upload_all_rows	未変更のローを含め、常にすべてのローがアップロードされることを示します。 このイニシャライザは、CREATE TABLE 文の SYNCHRONIZE ALL 句に対応します。
table_sync_download_only	変更がアップロードされないことを示します。 このイニシャライザは、CREATE TABLE 文の SYNCHRONIZE DOWNLOAD 句に対応します。

備考

TableSchema::GetTableSyncType

1.28 ValidateFlags 列挙体

検証入力フラグを表します。Connection の ValidateDatabase メソッドおよび DatabaseManager は ValidateFlags をフラグパラメータとして使用します。

構文

```
enum ValidateFlags
```

メンバー

メンバー名	説明	値
VF_TABLE	テーブルを検証します。 テーブルとインデックスのローカウントが一致しているかどうかをチェックします。	0x0001
VF_INDEX	インデックスを検証します。 インデックスの整合性をチェックします。	0x0002
VF_DATABASE	データベースを検証します。 ページのチェックサムやその他のチェックを使用してデータベースページを検証します。	0x0004
VF_EXPRESS	完全度は低いが、より高速な検証を実行します。 このフラグは他のフラグと組み合わせて使用します。	0x8000
VF_FULL_VALIDATE	データベース上ですべてのタイプの検証を実行します。	(ULVF_TABLE|ULVF_INDEX|ULVF_DATABASE)

1.29 ValidateStatusId 列挙体

検証進捗状況コールバックの可能なステータス ID を指定します。

構文

```
enum ValidateStatusId
```

メンバー

メンバー名	説明	値
VALID_NO_ERROR	エラーは発生しませんでした。	0
VALID_START	検証を開始します。	1
VALID_END	検証を終了します。 parm1 は、成功または失敗を示す結果の sqlcode を追跡します。	2

メンバー名	説明	値
VALID_CHECKING_PAGE	データベースページのチェック中、定期的にステータスメッセージを送信します。 Parm1 は、ページに関連付けられている数字を追跡します。順序は定義されていません。	10
VALID_CHECKING_TABLE	テーブルをチェックしています。 parm1 は、テーブル名を追跡します。	20
VALID_DUPLICATE_INDEX	インデックスをチェックしています。 parm1 はテーブル名を格納し、parm2 はインデックス名を格納します。	32
VALID_DATABASE_ERROR	データベースにアクセスするときにエラーが発生しました。 詳細については、SQLCODE を参照してください。	100
VALID_STARTUP_ERROR	データベースの起動中にエラーが発生しました。 (低レベルアクセスの場合)	101
VALID_CORRUPT_PAGE_TABLE	ページテーブルが破損しています。	110
VALID_FAILED_CHECKSUM	ページチェックサムが失敗しました。 Parm1 は、ページに関連付けられている数字を追跡します。	111
VALID_CORRUPT_PAGE	ページが破損しています。 Parm1 は、ページに関連付けられている数字を追跡します。	112
VALID_ROWCOUNT_MISMATCH	インデックス内のローの数がテーブルのロー数と異なります。 parm1 はテーブル名を追跡し、parm2 はインデックス名を追跡します。	120
VALID_BAD_ROWID	インデックス内に無効なロー識別子があります。 parm1 はテーブル名を追跡し、parm2 はインデックス名を追跡します。	121

1.30 FileTransferResult 構造

アプリケーションで適切なアクションを実行できるようにするために、ファイル転送結果を格納します。

構文

```
typedef struct  sealed
```

メンバー

FileTransferResult のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public property uint16	AuthCode	Mobile Link サーバ上の authenticate_file_transfer スクリプト (オプション) のリターンコードが格納されます。
public property int16	AuthStatus	Mobile Link イベントの認証パラメータにパラメータを渡します。
public property int64	AuthValue	カスタム Mobile Link のユーザ認証スクリプトの結果をレポートします。 Mobile Link サーバが、この情報をクライアントに提供します。
public property uint16	TransferredFile	ファイルが正常に転送された場合は 1 を返し、エラーが発生した場合は 0 を返します。
public property int16	StreamErrorCode	特定のストリームエラーを表します。 `%SQLANY17%¥¥SDK¥¥Include¥¥sserror.h`
public property int32	StreamErrorSystem	システム固有のエラーコードを表します。

1.31 FileTransferStatus 構造

ファイルのアップロードまたはダウンロードの進行中の、ステータスまたは進行状況情報を格納します。

構文

```
typedef struct  sealed
```

メンバー

FileTransferStatus のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public property uint64	FileSize	ダウンロード中のファイルの合計サイズ (バイト単位) を示します。
public property uint64	BytesReceived	現時点でダウンロード済みのファイルのサイズを示します。再開されたダウンロードの場合は、以前の同期も含みます。
public property uint64	ResumedAtSize	現在のダウンロードの開始点を示します。
public property int	Flags	ファイル転送ステータスについての追加情報を示します。 MLFileDownload メソッドがネットワーク呼び出しをブロックしており、observer メソッドが前回呼び出されたときからダウンロードステータスが変わっていない場合は、MLFT_STATUS_FLAG_IS_BLOCKING が設定されます。

備考

FileTransfer.UploadFile() および DownloadFile() の FileTransferObserver デリゲートに FileTransferStatus オブジェクトが渡されます。また、FileTransfer.UploadFileAsync() および DownloadFileAsync() は進捗状況コールバックで FileTransferStatus オブジェクトを提供します。

1.32 SyncResult 構造

アプリケーションで適切なアクションを実行できるようにするために、同期の結果を格納します。
Connection.GetSyncResult() は SyncResult を返します。

構文

```
typedef struct  sealed
```

メンバー

SyncResult のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public property ULSqlCode	ErrorCode	SQLCODE 値を示します。
public property String	ErrorParms	エラーパラメータを示します。
public property int	SqlCount	SQLCOUNT 値を示します。
public property int16	StreamErrorCode	特定のストリームエラーを示します。 想定される値については、ss_error_code 列挙を参照してください。
public property int32	StreamErrorSystem	システム固有のエラーコードを示します。 エラーコードの詳細については、プラットフォームのマニュアルを参照してください。
public property String	StreamErrorParms	stream_error_code 値のための文字列を示します。利用可能な場合は、追加情報が含まれます。
public property bool	UploadOK	アップロードが成功した場合は true、それ以外の場合は false を返します。
public property bool	IgnoredRows	アップロードされたローが無視された場合は true、それ以外の場合は false を返します。
public property int16	AuthStatus	同期認証ステータスを示します。
public property int64	AuthValue	Mobile Link サーバが auth_status の結果を判断するために使用する値を示します。
public property bool	PartialDownloadRetained	部分的なダウンロードが保持されたことを通知する値を示します。 keep_partial_download を参照してください。
public property DateTime	Timestamp	最後の同期の時刻と日付を示します。
public property int	SentBytes	現在までに送信されたバイト数を示します。
public property int	SentInserts	現在までに挿入されたローの数を示します。
public property int	SentUpdates	現在までに送信された更新済みのローの数を示します。
public property int	SentDeletes	現在までに送信された削除済みのローの数を示します。
public property int	RecvBytes	現在までに受信されたバイト数を示します。
public property int	RecvInserts	現在までに挿入されたローの数を示します。
public property int	RecvUpdates	現在までに受信したローのうち更新されたものの数を示します。

変更子とタイプ	変数	説明
public property int	IgnoredUpdates	現在のダウンロードで受信されたローのうちテーブルにすでに存在しているものの数を示します。
public property int	RecvDeletes	現在までに受信したローのうち削除されたものの数を示します。
public property int	IgnoredDeletes	現在のダウンロードで受信されたローのうちテーブルに存在しないものの数を示します。
public property int	TruncateDeletes	トランケート操作によって削除されたローの数を示します。

1.33 SyncStatus 構造

同期の進行中のステータスまたは進行状況情報を格納します。

構文

```
typedef struct  sealed
```

メンバー

SyncStatus のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public property short	ステータス	SyncState 列挙体のサポートされている多くのステータスの 1 つを示します。
public property short	TableCount	データベース内のテーブルの数を示します。
public property short	TableIndex	1 ～ TableCount の範囲のインデックスを示します。
public property short	TableID	現在アップロードまたはダウンロードされているテーブルの ID を示します (1 から始まります)。 同期されないテーブルがある場合には、この番号で値がスキップされることがあります。また、番号が必ず増加するとはかぎりません。
public property String	TableName	テーブル ID に対応するテーブル名を示します。

変更子とタイプ	変数	説明
public property int	SentBytes	現在までに送信されたバイト数を示します。
public property int	SentInserts	現在までに挿入されたローの数を示します。
public property int	SentUpdates	現在までに送信された更新済みのローの数を示します。
public property int	SentDeletes	現在までに送信された削除済みのローの数を示します。
public property int	RecvBytes	現在までに受信されたバイト数を示します。
public property int	RecvInserts	現在までに挿入されたローの数を示します。
public property int	RecvUpdates	現在までに受信したローのうち更新されたものの数を示します。
public property int	IgnoredUpdates	現在のダウンロードで受信されたローのうちテーブルにすでに存在しているものの数を示します。
public property int	RecvDeletes	現在までに受信したローのうち削除されたものの数を示します。
public property int	IgnoredDeletes	現在のダウンロードで受信されたローのうちテーブルに存在しないものの数を示します。
public property int	TruncateDeletes	トランケート操作によって削除されたローの数を示します。
public property unsigned short	Flags	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示します。
public property int	CurrentDownloadRowCount	これまでにダウンロードされたローの数を示します。 この数には、recvInserts、recvUpdates、または recvDeletes に含まれていない、重複するローが含まれます。
public property int	TotalDownloadRowCount	ダウンロードで受信するローの合計数を示します。 この数には、recvInserts、recvUpdates、または recvDeletes に含まれていない、重複するローが含まれます。 このフィールドは、同期によって最初のテーブルの STATE_RECEIVING_TABLE ステータスが入力されるまで、設定されません。

備考

Connection.Synchronize() の SyncObserver デリゲートに SyncStatus オブジェクトが渡されます。また、Connection.SynchronizeAsync() は進捗状況コールバックで SyncStatus オブジェクトを提供します。

1.34 ValidateData 構造

検証の進行中の検証ステータス情報を格納します。

構文

```
typedef struct  sealed
```

メンバー

ValidateData のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public property uint8	StatusId	検証プロセスでレポートされる対象を示します。
public property Array< String^>	Parms	この検証ステータスのパラメータを表します。

備考

Connection.Synchronize() の SyncObserver デリゲートに SyncStatus オブジェクトが渡されます。また、Connection.SynchronizeAsync() は進捗状況コールバックで SyncStatus オブジェクトを提供します。

2 このマニュアルの印刷、再生、および再配布

次の条件に従うかぎり、このマニュアルの全部または一部を使用、印刷、再生、配布することができます。

1. ここに示したものとそれ以外のすべての版権と商標の表示をすべてのコピーに含めること。
2. マニュアルに変更を加えないこと。
3. SAP 以外の人間がマニュアルの著者または情報源であるかのように示す一切の行為をしないこと。

ここに記載された情報は事前の通知なしに変更されることがあります。

重要免責事項および法的情報

コードサンプル

この文書に含まれるソフトウェアコード及び / 又はコードライン / 文字列 (「コード」) はすべてサンプルとしてのみ提供されるものであり、本稼動システム環境で使用することが目的ではありません。「コード」は、特定のコードの構文及び表現規則を分かりやすく説明及び視覚化することのみを目的としています。SAP は、この文書に記載される「コード」の正確性及び完全性の保証を行いません。更に、SAP は、「コード」の使用により発生したエラー又は損害が SAP の故意又は重大な過失が原因で発生させたものでない限り、そのエラー又は損害に対して一切責任を負いません。

アクセシビリティ

この SAP 文書に含まれる情報は、公開日現在のアクセシビリティ基準に関する SAP の最新の見解を表明するものであり、ソフトウェア製品のアクセシビリティ機能の確実な提供方法に関する拘束力のあるガイドラインとして意図されるものではありません。SAP は、この文書に関する一切の責任を明確に放棄するものです。ただし、この免責事項は、SAP の意図的な違法行為または重大な過失による場合は、適用されません。さらに、この文書により SAP の直接的または間接的な契約上の義務が発生することは一切ありません。

ジェンダーニュートラルな表現

SAP 文書では、可能な限りジェンダーニュートラルな表現を使用しています。文脈により、文書の読者は「あなた」と直接的な呼ばれ方をされたり、ジェンダーニュートラルな名詞 (例: 「販売員」又は「勤務日数」) で表現されます。ただし、男女両方を指すとき、三人称単数形の使用が避けられない又はジェンダーニュートラルな名詞が存在しない場合、SAP はその名詞又は代名詞の男性形を使用する権利を有します。これは、文書を分かりやすくするためです。

インターネットハイパーリンク

SAP 文書にはインターネットへのハイパーリンクが含まれる場合があります。これらのハイパーリンクは、関連情報を見いだすヒントを提供することが目的です。SAP は、この関連情報の可用性や正確性又はこの情報が特定の目的に役立つことの保証は行いません。SAP は、関連情報の使用により発生した損害が、SAP の重大な過失又は意図的な違法行為が原因で発生したものでない限り、その損害に対して一切責任を負いません。すべてのリンクは、透明性を目的に分類されています (http://help.sap.com/disclaimer を参照)。

go.sap.com/registration/contact.html