

SQL Anywhere - Mobile Link
文書バージョン: 17 - 2016-05-11

Mobile Link - .NET API リファレンス

目次

1	Mobile Link サーバ .NET API リファレンス	6
1.1	DateTimeWithTimeZone クラス.....	7
	DateTimeWithTimeZone コンストラクタ.....	9
	Parse(string) メソッド.....	13
	ToString メソッド.....	13
	DateTime プロパティ.....	15
	Day プロパティ.....	16
	Hour プロパティ.....	16
	Millisecond プロパティ.....	16
	Minute プロパティ.....	17
	Month プロパティ.....	17
	Second プロパティ.....	17
	TimeZoneHour プロパティ.....	18
	TimeZoneMinute プロパティ.....	18
	Year プロパティ.....	18
1.2	DBCommand インタフェース.....	19
	Close() メソッド.....	21
	ExecuteNonQuery() メソッド.....	21
	ExecuteReader() メソッド.....	21
	Prepare() メソッド.....	22
	CommandText プロパティ.....	22
	Parameters プロパティ.....	23
1.3	DBConnection インタフェース.....	23
	Close() メソッド.....	24
	Commit() メソッド.....	25
	CreateCommand() メソッド.....	25
	Rollback() メソッド.....	25
1.4	DBConnectionContext インタフェース.....	26
	GetConnection() メソッド.....	28
	GetDownloadData() メソッド.....	28
	GetProperties() メソッド.....	29
	GetRemoteID() メソッド.....	30
	GetServerContext() メソッド.....	31
	GetVersion() メソッド.....	32

	NetworkData プロパティ	33
1.5	DBParameter クラス	33
	DbType プロパティ	35
	Direction プロパティ	36
	IsNullable プロパティ	36
	ParameterName プロパティ	37
	Precision プロパティ	37
	Scale プロパティ	38
	Size プロパティ	38
	Value プロパティ	39
1.6	DBParameterCollection クラス	39
	DBParameterCollection() コンストラクタ	42
	Add(object) メソッド	42
	Clear() メソッド	43
	Contains メソッド	43
	CopyTo(Array, int) メソッド	45
	GetEnumerator() メソッド	45
	IndexOf メソッド	46
	Insert(int, object) メソッド	48
	Remove(object) メソッド	48
	RemoveAt メソッド	49
	Count プロパティ	51
	IsFixedSize プロパティ	51
	IsReadOnly プロパティ	51
	IsSynchronized プロパティ	52
	SyncRoot プロパティ	52
	this プロパティ	52
1.7	DBRowReader インタフェース	54
	Close() メソッド	55
	NextRow() メソッド	56
	ColumnNames プロパティ	57
	ColumnTypes プロパティ	57
1.8	DownloadData インタフェース	58
	GetDownloadTableByName(string) メソッド	59
	GetDownloadTables() メソッド	59
1.9	DownloadTableData インタフェース	60
	GetDeleteCommand() メソッド	62
	GetLastDownloadTime() メソッド	63
	GetName() メソッド	64

	GetSchemaTable() メソッド	64
	GetUpsertCommand() メソッド	65
1.10	FatalException クラス	66
	FatalException コンストラクタ	66
1.11	LogMessage クラス	69
	LogMessage(MessageType, string, string) コンストラクタ	71
	MessageType 列挙体	71
	Text プロパティ	72
	Type プロパティ	72
	User プロパティ	72
1.12	NetworkData インタフェース	73
	GetHTTPHeaderValue(string) メソッド	75
	GetHTTPHeaderValues(string) メソッド	76
	ClientCertificates プロパティ	77
	HTTPHeaders プロパティ	77
	IsEndToEndEncrypted プロパティ	78
	IsHTTP プロパティ	78
	IsTLS プロパティ	79
1.13	ScriptExecutionException クラス	79
	ScriptExecutionException コンストラクタ	80
1.14	ServerContext インタフェース	83
	getProperties(string, string) メソッド	85
	getPropertiesByVersion(string) メソッド	86
	getPropertySetNames(string) メソッド	86
	GetStartClassInstances() メソッド	87
	MakeConnection() メソッド	88
	Shutdown() メソッド	88
	ErrorListener イベント	89
	InfoListener イベント	89
	ShutdownListener イベント	89
	WarningListener イベント	90
1.15	ServerException クラス	90
	ServerException コンストラクタ	91
1.16	SpatialUtilities クラス	94
	CreateSpatialValue(int, byte[]) メソッド	95
	GetBytes(byte[]) メソッド	96
	GetSRID(byte[]) メソッド	96
	SetSRID(byte[], int) メソッド	97
1.17	SynchronizationException クラス	97

	SynchronizationException コンストラクタ	98
1.18	UpdateDataReader インタフェース	101
	SetNewRowValues() メソッド	102
	SetOldRowValues() メソッド	103
1.19	UploadData インタフェース	103
	GetUploadedTableByName(string) メソッド	104
	GetUploadedTables() メソッド	105
1.20	UploadedTableData インタフェース	106
	GetDeletes() メソッド	107
	GetInserts() メソッド	109
	GetName() メソッド	110
	GetSchemaTable() メソッド	110
	GetUpdates() メソッド	111
1.21	LogCallback(ServerContext, LogMessage) デリゲート	112
1.22	ShutdownCallback(ServerContext) デリゲート	113
1.23	SQLType 列挙体	114
2	このマニュアルの印刷、再生、および再配布	117

1 Mobile Link サーバ .NET API リファレンス

Mobile Link .NET API のトピックでは、インタフェースとクラス、これらに関連するメソッド、プロパティ、コンストラクタについて説明します。これらのクラスを使用するには、`%SQLANY17%\Assembly\%V2`にある `Sap.MobiLink.Script.dll` アセンブリを参照してください。

ネームスペース

```
Sap.MobiLink.Script
```

i 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:

[DateTimeWithTimeZone クラス \[7 ページ\]](#)

タイムゾーンオフセット付きの DateTime を表します。

[DBCommand インタフェース \[19 ページ\]](#)

SQL 文またはデータベースコマンドを表します。

[DBConnection インタフェース \[23 ページ\]](#)

Mobile Link ODBC 接続を表します。

[DBConnectionContext インタフェース \[26 ページ\]](#)

現在のデータベース接続に関する情報を取得します。

[DBParameter クラス \[33 ページ\]](#)

ODBC バウンドパラメータを表します。

[DBParameterCollection クラス \[39 ページ\]](#)

DBParameter のコレクション。

[DBRowReader インタフェース \[54 ページ\]](#)

データベースから読み込まれるローセットを表します。

[DownloadData インタフェース \[58 ページ\]](#)

ダイレクトローハンドリングで使用するすべてのダウンロードデータ操作をカプセル化します。

[DownloadTableData インタフェース \[60 ページ\]](#)

1つのダウンロードテーブルの情報を同期用にカプセル化します。

[FatalException クラス \[66 ページ\]](#)

サーバ側で致命的エラーが内部的に発生したため、直ちに停止する必要があることを Mobile Link に通知します。

[LogMessage クラス \[69 ページ\]](#)

ログに出力されたメッセージに関する情報が含まれています。

[NetworkData インタフェース \[73 ページ\]](#)

同期用のネットワークストリームの情報を含みます。

[ScriptExecutionException クラス \[79 ページ\]](#)

ユーザスクリプトでエラーが発生したことを通知します。

[ServerContext インタフェース \[83 ページ\]](#)

Mobile Link サーバの継続期間中に存在する、すべてのコンテキストをインスタンス化します。

[ServerException クラス \[90 ページ\]](#)

サーバでエラーが発生したため直ちに停止する必要があることを Mobile Link に通知します。

[SpatialUtilities クラス \[94 ページ\]](#)

空間値を操作する静的メソッドのコレクションを表します。

[SynchronizationException クラス \[97 ページ\]](#)

同期例外が発生したこと、現在の同期をロールバックして再開する必要があることを示します。

[UpdateDataReader インタフェース \[101 ページ\]](#)

1つのテーブルの1つのアップロードトランザクションのための更新操作を保持します。

[UploadData インタフェース \[103 ページ\]](#)

ダイレクトローハンドリングで使用するアップロード操作をカプセル化します。

[UploadedTableData インタフェース \[106 ページ\]](#)

1つのアップロードテーブルの情報を同期用にカプセル化します。

[LogCallback\(ServerContext, LogMessage\) デリゲート \[112 ページ\]](#)

Mobile Link サーバがメッセージを出力したときに呼び出されます。

[ShutdownCallback\(ServerContext\) デリゲート \[113 ページ\]](#)

Mobile Link サーバが停止するときに呼び出されます。

[SQLType 列挙体 \[114 ページ\]](#)

可能性のあるすべての ODBC データ型を列挙します。

1.1 DateTimeWithTimeZone クラス

タイムゾーンオフセット付きの DateTime を表します。

構文

Visual Basic

```
Public NotInheritable Class DateTimeWithTimeZone
```

C#

```
public sealed class DateTimeWithTimeZone
```

メンバー

DateTimeWithTimeZone 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。

コンストラクタ

変更子とタイプ	コンストラクタ	説明
public	DateTimeWithTimeZone [9 ページ]	タイムゾーンオフセットが 0 で指定された DateTime と同じ日付と時刻を含む DateTimeWithTimeZone を構成します。

メソッド

変更子とタイプ	メソッド	説明
public static DateTimeWithTimeZone	Parse(string) [13 ページ]	指定された文字列を解析して、新しい DateTimeWithTimeZone を返します。
public override string	ToString [13 ページ]	タイムゾーンオフセットが最後に追加された DateTime.ToString() と同じです。

プロパティ

変更子とタイプ	プロパティ	説明
public DateTime	DateTime [15 ページ]	タイムゾーンオフセットなしでこれに対応する DateTime を取得します。
public int	Day [16 ページ]	DateTime.Day と同じです。
public int	Hour [16 ページ]	DateTime.Hour と同じです。
public int	Millisecond [16 ページ]	DateTime.Millisecond と同じです。
public int	Minute [17 ページ]	DateTime.Minute と同じです。
public int	Month [17 ページ]	DateTime.Month と同じです。
public int	Second [17 ページ]	DateTime.Second と同じです。
public int	TimeZoneHour [18 ページ]	タイムゾーンオフセットの時間の部分を取得します。
public int	TimeZoneMinute [18 ページ]	タイムゾーンオフセットの分の部分を取得します。
public int	Year [18 ページ]	DateTime.Year と同じです。

このセクションの内容:

[DateTimeWithTimeZone コンストラクタ \[9 ページ\]](#)

タイムゾーンオフセットが 0 で指定された DateTime と同じ日付と時刻を含む DateTimeWithTimeZone を構成します。

[Parse\(string\) メソッド \[13 ページ\]](#)

指定された文字列を解析して、新しい DateTimeWithTimeZone を返します。

[ToString メソッド \[13 ページ\]](#)

タイムゾーンオフセットが最後に追加された DateTime.ToString() と同じです。

[DateTime プロパティ \[15 ページ\]](#)

タイムゾーンオフセットなしでこれに対応する DateTime を取得します。

[Day プロパティ \[16 ページ\]](#)

DateTime.Day と同じです。

[Hour プロパティ \[16 ページ\]](#)

DateTime.Hour と同じです。

[Millisecond プロパティ \[16 ページ\]](#)

DateTime.Millisecond と同じです。

[Minute プロパティ \[17 ページ\]](#)

DateTime.Minute と同じです。

[Month プロパティ \[17 ページ\]](#)

DateTime.Month と同じです。

[Second プロパティ \[17 ページ\]](#)

DateTime.Second と同じです。

[TimeZoneHour プロパティ \[18 ページ\]](#)

タイムゾーンオフセットの時間の部分を取得します。

[TimeZoneMinute プロパティ \[18 ページ\]](#)

タイムゾーンオフセットの分の部分を取得します。

[Year プロパティ \[18 ページ\]](#)

DateTime.Year と同じです。

1.1.1 DateTimeWithTimeZone コンストラクタ

タイムゾーンオフセットが 0 で指定された DateTime と同じ日付と時刻を含む DateTimeWithTimeZone を構成します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public	DateTimeWithTimeZone(DateTime) [10 ページ]	タイムゾーンオフセットが 0 で指定された DateTime と同じ日付と時刻を含む DateTimeWithTimeZone を構成します。

変更子とタイプ	オーバーロード名	説明
public	DateTimeWithTimeZone(DateTime, int, int) [11 ページ]	指定のタイムゾーンオフセットで指定された DateTime と同じ日付と時刻を含む DateTimeWithTimeZone を構成します。
public	DateTimeWithTimeZone(int, int, int, int, int, int, int, int, int) [11 ページ]	指定された年、月、日、時間、分、秒、ミリ秒、タイムゾーンの時間、タイムゾーンの分を含む DateTimeWithTimeZone を構成します。

このセクションの内容:

[DateTimeWithTimeZone\(DateTime\) コンストラクタ \[10 ページ\]](#)

タイムゾーンオフセットが 0 で指定された DateTime と同じ日付と時刻を含む DateTimeWithTimeZone を構成します。

[DateTimeWithTimeZone\(DateTime, int, int\) コンストラクタ \[11 ページ\]](#)

指定のタイムゾーンオフセットで指定された DateTime と同じ日付と時刻を含む DateTimeWithTimeZone を構成します。

[DateTimeWithTimeZone\(int, int, int, int, int, int, int, int, int\) コンストラクタ \[11 ページ\]](#)

指定された年、月、日、時間、分、秒、ミリ秒、タイムゾーンの時間、タイムゾーンの分を含む DateTimeWithTimeZone を構成します。

1.1.1.1 DateTimeWithTimeZone(DateTime) コンストラクタ

タイムゾーンオフセットが 0 で指定された DateTime と同じ日付と時刻を含む DateTimeWithTimeZone を構成します。

構文

Microsoft Visual Basic

```
Public Sub DateTimeWithTimeZone (ByVal dt As Date)
```

C#

```
public DateTimeWithTimeZone (DateTime dt)
```

1.1.1.2 DateTimeWithTimeZone(DateTime, int, int) コンストラクタ

指定のタイムゾーンオフセットで指定された DateTime と同じ日付と時刻を含む DateTimeWithTimeZone を構成します。

構文

Microsoft Visual Basic

```
Public Sub DateTimeWithTimeZone (  
    ByVal dt As Date,  
    ByVal tz_hour As Integer,  
    ByVal tz_minute As Integer  
)
```

C#

```
public DateTimeWithTimeZone (  
    DateTime dt,  
    int tz_hour,  
    int tz_minute  
)
```

パラメータ

dt 日付と時刻の部分。

tz_hour タイムゾーンオフセットの時間 (-12 ~ 14)

tz_minute タイムゾーンオフセットの分 (-59 ~ 59)。 **tz_hour** が正でない場合にのみ、負の値を指定できます。

例外

System.ArgumentOutOfRangeException **tz_hour** または **tz_minute** が範囲外の場合にスローされます。

1.1.1.3 DateTimeWithTimeZone(int, int, int, int, int, int, int, int, int) コンストラクタ

指定された年、月、日、時間、分、秒、ミリ秒、タイムゾーンの時間、タイムゾーンの分を含む DateTimeWithTimeZone を構成します。

構文

Microsoft Visual Basic

```
Public Sub DateTimeWithTimeZone (  

```

```
ByVal year As Integer,  
ByVal month As Integer,  
ByVal day As Integer,  
ByVal hour As Integer,  
ByVal minute As Integer,  
ByVal second As Integer,  
ByVal millisecond As Integer,  
ByVal tz_hour As Integer,  
ByVal tz_minute As Integer  
)
```

C#

```
public DateTimeWithTimeZone (  
    int year,  
    int month,  
    int day,  
    int hour,  
    int minute,  
    int second,  
    int millisecond,  
    int tz_hour,  
    int tz_minute  
)
```

パラメータ

year 年 (1 ~ 9999)

month 月 (1 ~ 12)

day 日 (1 ~ *month* の日数)

hour 時間 (0 ~ 23)

minute 分 (0 ~ 59)

second 秒 (0 ~ 59)

millisecond ミリ秒 (1 ~ 999)

tz_hour タイムゾーンオフセットの時間 (-12 ~ 14)

tz_minute タイムゾーンオフセットの分 (-59 ~ 59)。*tz_hour* が正でない場合にのみ、負の値を指定できます。

例外

System.ArgumentOutOfRangeException いずれかのパラメータが範囲外の場合にスローされます。

1.1.2 Parse(string) メソッド

指定された文字列を解析して、新しい DateTimeWithTimeZone を返します。

構文

Visual Basic

```
Public Shared Function Parse (ByVal val As String) As DateTimeWithTimeZone
```

C#

```
public static DateTimeWithTimeZone Parse (string val)
```

パラメータ

val "yyyy-MM-dd HH:mm:ss.ffffff SHH:mm" 形式の文字列で、S はタイムゾーンの時間オフセットの符号を表し、2 つ目の HH:mm はタイムゾーンオフセットを表します。時刻の小数部分は省略できます。タイムゾーンオフセットは省略できます。タイムゾーンオフセットを指定する場合、符号は省略できます。

例外

System.FormatException 指定された文字列がそのフォーマットと一致しない場合にスローされます。

1.1.3 ToString メソッド

タイムゾーンオフセットが最後に追加された DateTime.ToString() と同じです。

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 override string	ToString() [14 ページ]	タイムゾーンオフセットが最後に追加された DateTime.ToString() と同じです。
public string	ToString(IFormatProvider) [14 ページ]	タイムゾーンオフセットが最後に追加された DateTime.ToString(provider) と同じです。
public string	ToString(string) [15 ページ]	タイムゾーンオフセットが最後に追加された DateTime.ToString(format) と同じです。

変更子とタイプ	オーバーロード名	説明
public string	ToString(string, IFormatProvider) [15 ページ]	タイムゾーンオフセットが最後に追加された DateTime.ToString(format, provider) と同じです。

このセクションの内容:

[ToString\(\) メソッド \[14 ページ\]](#)

タイムゾーンオフセットが最後に追加された DateTime.ToString() と同じです。

[ToString\(IFormatProvider\) メソッド \[14 ページ\]](#)

タイムゾーンオフセットが最後に追加された DateTime.ToString(provider) と同じです。

[ToString\(string\) メソッド \[15 ページ\]](#)

タイムゾーンオフセットが最後に追加された DateTime.ToString(format) と同じです。

[ToString\(string, IFormatProvider\) メソッド \[15 ページ\]](#)

タイムゾーンオフセットが最後に追加された DateTime.ToString(format, provider) と同じです。

1.1.3.1 ToString() メソッド

タイムゾーンオフセットが最後に追加された DateTime.ToString() と同じです。

構文

Visual Basic

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

C#

```
public override string ToString ()
```

1.1.3.2 ToString(IFormatProvider) メソッド

タイムゾーンオフセットが最後に追加された DateTime.ToString(provider) と同じです。

構文

Visual Basic

```
Public Function ToString (ByVal provider As IFormatProvider) As String
```

C#

```
public string ToString (IFormatProvider provider)
```

1.1.3.3 ToString(string) メソッド

タイムゾーンオフセットが最後に追加された DateTime.ToString(format) と同じです。

構文

Visual Basic

```
Public Function ToString (ByVal format As String) As String
```

C#

```
public string ToString (string format)
```

1.1.3.4 ToString(string, IFormatProvider) メソッド

タイムゾーンオフセットが最後に追加された DateTime.ToString(format, provider) と同じです。

構文

Visual Basic

```
Public Function ToString (
    ByVal format As String,
    ByVal provider As IFormatProvider
) As String
```

C#

```
public string ToString (
    string format,
    IFormatProvider provider
)
```

1.1.4 DateTime プロパティ

タイムゾーンオフセットなしでこれに対応する DateTime を取得します。

構文

Microsoft Visual Basic

```
Public Property DateTime As Date
```

C#

```
public DateTime DateTime {get;set;}
```

1.1.5 Day プロパティ

DateTime.Day と同じです。

構文

Visual Basic

```
Public ReadOnly Property Day As Integer
```

C#

```
public int Day {get;}
```

1.1.6 Hour プロパティ

DateTime.Hour と同じです。

構文

Visual Basic

```
Public ReadOnly Property Hour As Integer
```

C#

```
public int Hour {get;}
```

1.1.7 Millisecond プロパティ

DateTime.Millisecond と同じです。

構文

Visual Basic

```
Public ReadOnly Property Millisecond As Integer
```

C#

```
public int Millisecond {get;}
```

1.1.8 Minute プロパティ

DateTime.Minute と同じです。

構文

Visual Basic

```
Public ReadOnly Property Minute As Integer
```

C#

```
public int Minute {get;}
```

1.1.9 Month プロパティ

DateTime.Month と同じです。

構文

Visual Basic

```
Public ReadOnly Property Month As Integer
```

C#

```
public int Month {get;}
```

1.1.10 Second プロパティ

DateTime.Second と同じです。

構文

Visual Basic

```
Public ReadOnly Property Second As Integer
```

C#

```
public int Second {get;}
```

1.1.11 TimeZoneHour プロパティ

タイムゾーンオフセットの時間の部分を取得します。

構文

Visual Basic

```
Public Property TimeZoneHour As Integer
```

C#

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

1.1.12 TimeZoneMinute プロパティ

タイムゾーンオフセットの分の部分を取得します。

構文

Visual Basic

```
Public Property TimeZoneMinute As Integer
```

C#

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

1.1.13 Year プロパティ

DateTime.Year と同じです。

構文

Visual Basic

```
Public ReadOnly Property Year As Integer
```

C#

```
public int Year {get;}
```

1.2 DBCommand インタフェース

SQL 文またはデータベースコマンドを表します。

構文

Visual Basic

```
Public Interface DBCommand
```

C#

```
public interface DBCommand
```

メンバー

DBCommand 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() [21 ページ]	現在の SQL 文またはコマンドを終了します。
public int	ExecuteNonQuery() [21 ページ]	non-query の文を実行します。
public DBRowReader	ExecuteReader() [21 ページ]	結果セットを返すクエリ文を実行します。
public void	Prepare() [22 ページ]	CommandText に格納されている SQL 文の実行を準備します。

プロパティ

変更子とタイプ	プロパティ	説明
public string	CommandText [22 ページ]	実行される SQL 文。
public DBParameterCollection	Parameters [23 ページ]	この DBCommand の DBParameterCollection を取得します。

備考

DBCommand は更新またはクエリを表すことができます。

例

次の C# コードは DBCommand インタフェースを使用して次の 2 つのクエリを実行します。

```
DBCommand stmt = conn.CreateCommand();
```

```

stmt.CommandText = "SELECT t1a1, t1a2 FROM table1 ";
DBRowReader rs = stmt.ExecuteReader();
printResultSet( rs );
rs.Close();
stmt.CommandText = "SELECT t2a1 FROM table2 ";
rs = stmt.ExecuteReader();
printResultSet( rs );
rs.Close();
stmt.Close();

```

次の C# コードは DBCommand インタフェースを使用してパラメータを指定して更新を実行します。

```

public void prepare_for_download(DateTime last_download,
    String ml_username)
{
    DBCommand cstmt = conn.CreateCommand();
    cstmt.CommandText = "CALL myProc( ?,?,? )";
    cstmt.Prepare();
    DBParameter param = new DBParameter();
    param.DbType      = SQLType.SQL_CHAR;
    param.Value       = "10000";
    cstmt.Parameters.Add(param);
    param              = new DBParameter();
    param.DbType      = SQLType.SQL_INTEGER;
    param.Value       = 20000;
    cstmt.Parameters.Add(param);
    param              = new DBParameter();
    param.DbType      = SQLType.SQL_DECIMAL;
    param.Precision   = 5;
    param.Value       = new Decimal(30000);
    cstmt.Parameters.Add(param);
    // Execute update
    DBRowReader rset = cstmt.ExecuteNonQuery();
    cstmt.Close();
}

```

このセクションの内容:

[Close\(\) メソッド \[21 ページ\]](#)

現在の SQL 文またはコマンドを終了します。

[ExecuteNonQuery\(\) メソッド \[21 ページ\]](#)

non-query の文を実行します。

[ExecuteReader\(\) メソッド \[21 ページ\]](#)

結果セットを返すクエリ文を実行します。

[Prepare\(\) メソッド \[22 ページ\]](#)

CommandText に格納されている SQL 文の実行を準備します。

[CommandText プロパティ \[22 ページ\]](#)

実行される SQL 文。

[Parameters プロパティ \[23 ページ\]](#)

この DBCommand の DBParameterCollection を取得します。

1.2.1 Close() メソッド

現在の SQL 文またはコマンドを終了します。

構文

Visual Basic

```
Public Sub Close ()
```

C#

```
public void Close ()
```

1.2.2 ExecuteNonQuery() メソッド

non-query の文を実行します。

構文

Visual Basic

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

C#

```
public int ExecuteNonQuery ()
```

戻り値

データベース内で SQL 文の影響を受けるローの数

1.2.3 ExecuteReader() メソッド

結果セットを返すクエリ文を実行します。

構文

Visual Basic

```
Public Function ExecuteReader () As DBRowReader
```

C#

```
public DBRowReader ExecuteReader ()
```

戻り値

SQL 文が返す結果を取得するための DBRowReader

1.2.4 Prepare() メソッド

CommandText に格納されている SQL 文の実行を準備します。

構文

Visual Basic

```
Public Sub Prepare ()
```

C#

```
public void Prepare ()
```

1.2.5 CommandText プロパティ

実行される SQL 文。

構文

Visual Basic

```
Public Property CommandText As String
```

C#

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

1.2.6 Parameters プロパティ

この DBCommand の DBParameterCollection を取得します。

構文

Visual Basic

```
Public ReadOnly Property Parameters As DBParameterCollection
```

C#

```
public DBParameterCollection Parameters {get;}
```

戻り値

要求された Parameter コレクション

関連情報

[DBParameterCollection クラス \[39 ページ\]](#)

1.3 DBConnection インタフェース

Mobile Link ODBC 接続を表します。

構文

Visual Basic

```
Public Interface DBConnection
```

C#

```
public interface DBConnection
```

メンバー

DBConnection 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() [24 ページ]	現在の接続を閉じます。
public void	Commit() [25 ページ]	現在のトランザクションをコミットします。
public DBCommand	CreateCommand() [25 ページ]	この接続で SQL 文またはコマンドを作成します。
public void	Rollback() [25 ページ]	現在のトランザクションをロールバックします。

備考

このインターフェースにより、ユーザが作成した同期ロジックで Mobile Link によって確立された ODBC 接続にアクセスできません。

このセクションの内容:

[Close\(\) メソッド \[24 ページ\]](#)

現在の接続を閉じます。

[Commit\(\) メソッド \[25 ページ\]](#)

現在のトランザクションをコミットします。

[CreateCommand\(\) メソッド \[25 ページ\]](#)

この接続で SQL 文またはコマンドを作成します。

[Rollback\(\) メソッド \[25 ページ\]](#)

現在のトランザクションをロールバックします。

1.3.1 Close() メソッド

現在の接続を閉じます。

構文

Visual Basic

```
Public Sub Close ()
```

C#

```
public void Close ()
```

1.3.2 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

Visual Basic

```
Public Sub Commit ()
```

C#

```
public void Commit ()
```

1.3.3 CreateCommand() メソッド

この接続で SQL 文またはコマンドを作成します。

構文

Visual Basic

```
Public Function CreateCommand () As DBCommand
```

C#

```
public DBCommand CreateCommand ()
```

戻り値

新しく生成された DBCommand

1.3.4 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

Visual Basic

```
Public Sub Rollback ()
```

C#

```
public void Rollback ()
```

1.4 DBConnectionContext インタフェース

現在のデータベース接続に関する情報を取得します。

構文

Visual Basic

```
Public Interface DBConnectionContext
```

C#

```
public interface DBConnectionContext
```

メンバー

DBConnectionContext 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 DBConnection	GetConnection() [28 ページ]	Mobile Link 統合データベースへの既存の接続を返します。
public DownloadData	GetDownloadData() [28 ページ]	現在の同期に対する DownloadData を返します。
public NameValueCollection	GetProperties() [29 ページ]	この接続のスクリプトバージョンに基づいて、プロパティのコレクションを返します。
public string	GetRemoteID() [30 ページ]	この接続で現在同期中のデータベースのリモート ID を返します。
public ServerContext	GetServerContext() [31 ページ]	現在のサーバコンテキストを返します。
public string	GetVersion() [32 ページ]	この接続のバージョン文字列を返します。

プロパティ

変更子とタイプ	プロパティ	説明
public NetworkData	NetworkData [33 ページ]	同期用のネットワークストリームの情報を返します。

備考

このインタフェースは、スクリプトを含むクラスのコンストラクタに渡されます。コンテキストがバックグラウンドスレッドに必要な場合や、接続期間を超えて必要な場合は、ServerContext を使用してください。

コンストラクタの詳細については、コンストラクタを参照してください。

i 注記

DBConnectionContext インスタンスは、.NET コードに呼び出すスレッド以外で使わないでください。

例

次の例は、同期スクリプトで使用するクラスレベルの DBConnectionContext インスタンスを作成する方法を示します。DBConnectionContext の getConnection メソッドは、Mobile Link 統合データベースとの現在の接続を表す DBConnection インスタンスを取得します。

```
using Sap.MobiLink.Script;
using System.Data;
public class OrderProcessor {
    DBConnectionContext _cc;
    public OrderProcessor( DBConnectionContext cc ) {
        _cc = cc;
    }
    // The method used for the handle_DownloadData event.
    public void HandleEvent() {
        DBConnection my_connection = _cc.GetConnection();
        // ...
    }
    // ...
}
```

このセクションの内容:

[GetConnection\(\) メソッド \[28 ページ\]](#)

Mobile Link 統合データベースへの既存の接続を返します。

[GetDownloadData\(\) メソッド \[28 ページ\]](#)

現在の同期に対する DownloadData を返します。

[GetProperties\(\) メソッド \[29 ページ\]](#)

この接続のスクリプトバージョンに基づいて、プロパティのコレクションを返します。

[GetRemoteID\(\) メソッド \[30 ページ\]](#)

この接続で現在同期中のデータベースのリモート ID を返します。

[GetServerContext\(\) メソッド \[31 ページ\]](#)

現在のサーバコンテキストを返します。

[GetVersion\(\) メソッド \[32 ページ\]](#)

この接続のバージョン文字列を返します。

[NetworkData プロパティ \[33 ページ\]](#)

同期用のネットワークストリームの情報を返します。

1.4.1 GetConnection() メソッド

Mobile Link 統合データベースへの既存の接続を返します。

構文

Visual Basic

```
Public Function GetConnection () As DBConnection
```

C#

```
public DBConnection GetConnection ()
```

戻り値

統合データベースへの現在の接続。この接続は、基本となる Mobile Link 接続の期間中にのみ有効です。

備考

Mobile Link が SQL スクリプトの実行に使用するものと同じ接続です。Mobile Link サーバのこの接続での使用に影響する方法でコミット、終了、または変更しないでください。

接続に対して end_connection イベントが呼び出された後は、その接続を使用しないでください。

フルアクセス権を持つサーバ接続が必要な場合は、MakeConnection メソッドを使用します。

関連情報

[MakeConnection\(\) メソッド \[88 ページ\]](#)

1.4.2 GetDownloadData() メソッド

現在の同期に対する DownloadData を返します。

構文

Visual Basic

```
Public Function GetDownloadData () As DownloadData
```

C#

```
public DownloadData GetDownloadData ()
```

戻り値

現在の同期に対する DownloadData、同期がアップロード専用の場合は NULL

備考

ダイレクトローハンドリング用のダウンロードを作成する場合は、DownloadData インスタンスを使用してください。

例

次の例は、_cc という DBConnectionContext インスタンスが作成済みであることを前提としています。

```
// The method used for the handle_DownloadData event.
public void HandleDownload() {
    // Get the DownloadData for the current synchronization.
    DownloadData my_dd = _cc.GetDownloadData();
    // Get an array of tables to set download operations.
    DownloadTableData[] download_tables = my_dd.GetDownloadTables();
    // Get the first table in the DownloadTableData array.
    DownloadTableData my_download_table = download_tables[0];
    // ...
}
```

1.4.3 GetProperties() メソッド

この接続のスクリプトバージョンに基づいて、プロパティのコレクションを返します。

構文

Visual Basic

```
Public Function GetProperties () As NameValueCollection
```

C#

```
public NameValueCollection GetProperties ()
```

戻り値

現在のスクリプトバージョンのプロパティ

備考

プロパティは、ml_property テーブルに格納されます。詳細については、ml_add_property システムプロシージャを参照してください。

例

次の例は、DBConnectionContext のプロパティを出力する方法を示します。この例は、_cc という DBConnectionContext インスタンスがあることを前提としています。

```
// The method used to output the connection properties.
public void OutputProperties() {
    // output the Properties for the current synchronization
    NameValueCollection properties = _cc.GetProperties();
    System.Console.WriteLine(properties.ToString());
}
```

1.4.4 GetRemoteID() メソッド

この接続で現在同期中のデータベースのリモート ID を返します。

構文

Visual Basic

```
Public Function GetRemoteID () As String
```

C#

```
public string GetRemoteID ()
```

戻り値

リモート ID。

備考

リモート ID の詳細については、リモート ID を参照してください。

例

次の例は、DBConnectionContext のリモート ID を出力する方法を示します。

```
// The method used to output the remote ID.
public void OutputRemoteID() {
    // output the Remote ID for the current synchronization
    string remoteID = _cc.GetRemoteID();
    System.Console.WriteLine(remoteID);
}
```

1.4.5 GetServerContext() メソッド

現在のサーバコンテキストを返します。

構文

Visual Basic

```
Public Function GetServerContext () As ServerContext
```

C#

```
public ServerContext GetServerContext ()
```

戻り値

この Mobile Link サーバの ServerContext

備考

このメソッドは、新しい接続を作成したり、ブートクラスと対話したりするために使用できます。

例

次の例は、DBConnectionContext の ServerContext インスタンスを取得して、サーバを停止する方法を示します。

```
// A method that uses an instance of the ServerContext to shut down the server
public void ShutDownServer() {
    ServerContext context = _cc.GetServerContext();
    context.Shutdown();
}
```

```
}
```

1.4.6 GetVersion() メソッド

この接続のバージョン文字列を返します。

構文

Visual Basic

```
Public Function GetVersion () As String
```

C#

```
public string GetVersion ()
```

戻り値

スクリプトバージョン名

備考

プロパティは、ml_property テーブルに格納されます。詳細については、ml_add_property システムプロシーダを参照してください。

例

次の例は、スクリプトバージョンを取得して、決定を行うために使用する方法を示します。

```
public void MyEvent() {  
    // ...  
    string version = _cc.GetVersion();  
    switch( version ) {  
        case "My Version 1":  
            // ...  
            break;  
        case "My Version 2":  
            // ...  
            break;  
    }  
}
```

1.4.7 NetworkData プロパティ

同期用のネットワークストリームの情報を返します。

構文

Visual Basic

```
Public ReadOnly Property NetworkData As NetworkData
```

C#

```
public NetworkData NetworkData {get;}
```

戻り値

要求に使用されたネットワークストリームの情報、コレクションが有効でない場合は NULL

備考

この方法は、クライアント側証明書と HTTP ヘッダを使用するエンタープライズ内の別のサーバを認証する際に便利です。

ネットワークストリームのデータの収集を有効にするには、-x スイッチに collect_network_data=1 を追加します。このオプションにより、同期ごとのデータの保存時に、メモリのオーバーヘッドが増えます。

関連情報

[NetworkData インタフェース \[73 ページ\]](#)

1.5 DBParameter クラス

ODBC バウンドパラメータを表します。

構文

Visual Basic

```
Public Class DBParameter
```

C#

```
public class DBParameter
```

メンバー

DBParameter 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

変数とタイプ	変数	説明
public bool	HasChanged	パラメータが作成後に変更されたかどうかを返します。

プロパティ

変数とタイプ	プロパティ	説明
public SQLType	DbType [35 ページ]	このパラメータの SQLType。
public ParameterDirection	Direction [36 ページ]	このパラメータの入出力方向。
public bool	IsNullable [36 ページ]	値に NULL を指定できる場合は true、そうでない場合は false。
public string	ParameterName [37 ページ]	このパラメータの名前。
public uint	Precision [37 ページ]	このパラメータの 10 進数精度。
public short	Scale [38 ページ]	このパラメータの解決可能な桁数。
public uint	Size [38 ページ]	このパラメータのサイズ (バイト単位)。
public object	Value [39 ページ]	このパラメータの値。

備考

パラメータを指定してコマンドを実行するには、このクラスが必要です。すべてのパラメータを指定してからコマンドを実行してください。

例

次の C# コードは DBCommand インタフェースを使用してパラメータを指定して更新を実行します。

```
using( DBCommand cstmt = conn.CreateCommand() ) {  
    DBCommand cstmt = conn.CreateCommand();  
    cstmt.CommandText = "call myProc( ?, ?, ? )";  
    cstmt.Prepare();  
    DBParameter param = new DBParameter();  
    param.DbType      = SQLType.SQL_CHAR;  
    param.Value       = "10000";  
    cstmt.Parameters.Add( param );  
    param             = new DBParameter();  
}
```

```

param.DbType      = SQLType.SQL_INTEGER;
param.Value       = 20000;
cstmt.Parameters.Add( param );
param             = new DBParameter();
param.DbType      = SQLType.SQL_DECIMAL;
param.Precision   = 5;
param.Value       = new Decimal( 30000 );
cstmt.Parameters.Add( param );
// Execute update
DBRowReader rset = cstmt.ExecuteNonQuery();
cstmt.Close();
}

```

このセクションの内容:

[DbType プロパティ \[35 ページ\]](#)

このパラメータの SQLType。

[Direction プロパティ \[36 ページ\]](#)

このパラメータの入出力方向。

[IsNullable プロパティ \[36 ページ\]](#)

値に NULL を指定できる場合は true、そうでない場合は false。

[ParameterName プロパティ \[37 ページ\]](#)

このパラメータの名前。

[Precision プロパティ \[37 ページ\]](#)

このパラメータの 10 進数精度。

[Scale プロパティ \[38 ページ\]](#)

このパラメータの解決可能な桁数。

[Size プロパティ \[38 ページ\]](#)

このパラメータのサイズ (バイト単位)。

[Value プロパティ \[39 ページ\]](#)

このパラメータの値。

1.5.1 DbType プロパティ

このパラメータの SQLType。

構文

Visual Basic

```
Public Property DbType As SQLType
```

C#

```
public SQLType DbType {get;set;}
```

備考

デフォルト値は `SQLType.SQL_TYPE_NULL` です。

1.5.2 Direction プロパティ

このパラメータの入出力方向。

≡ 構文

Visual Basic

```
Public Property Direction As ParameterDirection
```

C#

```
public ParameterDirection Direction {get;set;}
```

備考

デフォルト値は `ParameterDirection.Input` です。

1.5.3 IsNullable プロパティ

値に NULL を指定できる場合は `true`、そうでない場合は `false`。

≡ 構文

Visual Basic

```
Public Property IsNullable As Boolean
```

C#

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

備考

デフォルト値は `false` です。

1.5.4 ParameterName プロパティ

このパラメータの名前。

構文

Visual Basic

```
Public Property ParameterName As String
```

C#

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

備考

デフォルト値は NULL です。

1.5.5 Precision プロパティ

このパラメータの 10 進数精度。

構文

Visual Basic

```
Public Property Precision As UInteger
```

C#

```
public uint Precision {get;set;}
```

備考

このプロパティは、SQLType.SQL_NUMERIC パラメータと SQLType.SQL_DECIMAL パラメータにのみ使用します。

デフォルト値は 0 です。

1.5.6 Scale プロパティ

このパラメータの解決可能な桁数。

構文

Visual Basic

```
Public Property Scale As Short
```

C#

```
public short Scale {get;set;}
```

備考

このプロパティは、SQLType.SQL_NUMERIC パラメータと SQLType.SQL_DECIMAL パラメータにのみ使用します。

デフォルト値は 0 です。

1.5.7 Size プロパティ

このパラメータのサイズ (バイト単位)。

構文

Visual Basic

```
Public Property Size As UInteger
```

C#

```
public uint Size {get;set;}
```

備考

デフォルト値は DbType から推定されます。

1.5.8 Value プロパティ

このパラメータの値。

構文

Visual Basic

```
Public Property Value As Object
```

C#

```
public object Value {get;set;}
```

備考

デフォルト値は NULL です。

1.6 DBParameterCollection クラス

DBParameter のコレクション。

構文

Visual Basic

```
Public Class DBParameterCollection Implements  
System.Data.IDataParameterCollection, System.Collections.IList,  
System.Collections.ICollection, System.Collections.IEnumerable
```

C#

```
public class DBParameterCollection : System.Data.IDataParameterCollection,  
System.Collections.IList, System.Collections.ICollection,  
System.Collections.IEnumerable
```

メンバー

DBParameterCollection のすべてのメンバー (継承されたメンバーも含まれます) を次に示します。

コンストラクタ

変更子とタイプ	コンストラクタ	説明
public	DBParameterCollection() [42 ページ]	DBParameter の空リストを作成します。

メソッド

変更子とタイプ	メソッド	説明
public int	Add(object) [42 ページ]	指定したパラメータをコレクションに追加します。
public void	Clear() [43 ページ]	すべてのパラメータをコレクションから削除します。
public bool	Contains [43 ページ]	指定した DBParameter がコレクションに含まれる場合は true を返します。
public void	CopyTo(Array, int) [45 ページ]	コレクションの内容を、指定したインデックスで始まる特定の配列にコピーします。
public IEnumerator	GetEnumerator() [45 ページ]	コレクションの列挙子を返します。
public int	IndexOf [46 ページ]	コレクション内の指定された DBParameter のインデックスを返します。
public void	Insert(int, object) [48 ページ]	指定した DBParameter を、コレクション内の指定したインデックス位置に挿入します。
public void	Remove(object) [48 ページ]	指定した DBParameter をコレクションから削除します。
public void	RemoveAt [49 ページ]	コレクション内の指定したインデックス位置の DBParameter を削除します。

プロパティ

変更子とタイプ	プロパティ	説明
public int	Count [51 ページ]	コレクション内のパラメータ数。
public bool	IsFixedSize [51 ページ]	false を返します。
public bool	IsReadOnly [51 ページ]	false を返します。
public bool	IsSynchronized [52 ページ]	false を返します。
public object	SyncRoot [52 ページ]	DBParameterCollection へのアクセスを同期するために使用します。
public object	this [52 ページ]	コレクション内の指定したインデックスを持つ DBParameter を取得または設定します。

備考

DBCommand で DBParameterCollection を作成した時点では、DBParameterCollection は空です。DBCommand を実行する前に、適切なパラメータを指定してください。

このセクションの内容:

[DBParameterCollection\(\) コンストラクタ \[42 ページ\]](#)

DBParameter の空リストを作成します。

[Add\(object\) メソッド \[42 ページ\]](#)

指定したパラメータをコレクションに追加します。

[Clear\(\) メソッド \[43 ページ\]](#)

すべてのパラメータをコレクションから削除します。

[Contains メソッド \[43 ページ\]](#)

指定した DBParameter がコレクションに含まれる場合は true を返します。

[CopyTo\(Array, int\) メソッド \[45 ページ\]](#)

コレクションの内容を、指定したインデックスで始まる特定の配列にコピーします。

[GetEnumerator\(\) メソッド \[45 ページ\]](#)

コレクションの列挙子を返します。

[IndexOf メソッド \[46 ページ\]](#)

コレクション内の指定された DBParameter のインデックスを返します。

[Insert\(int, object\) メソッド \[48 ページ\]](#)

指定した DBParameter を、コレクション内の指定したインデックス位置に挿入します。

[Remove\(object\) メソッド \[48 ページ\]](#)

指定した DBParameter をコレクションから削除します。

[RemoveAt メソッド \[49 ページ\]](#)

コレクション内の指定したインデックス位置の DBParameter を削除します。

[Count プロパティ \[51 ページ\]](#)

コレクション内のパラメータ数。

[IsFixedSize プロパティ \[51 ページ\]](#)

false を返します。

[IsReadOnly プロパティ \[51 ページ\]](#)

false を返します。

[IsSynchronized プロパティ \[52 ページ\]](#)

false を返します。

[SyncRoot プロパティ \[52 ページ\]](#)

DBParameterCollection へのアクセスを同期するために使用します。

[this プロパティ \[52 ページ\]](#)

コレクション内の指定したインデックスを持つ DBParameter を取得または設定します。

1.6.1 DBParameterCollection() コンストラクタ

DBParameter の空リストを作成します。

構文

Visual Basic

```
Public Sub DBParameterCollection ()
```

C#

```
public DBParameterCollection ()
```

1.6.2 Add(object) メソッド

指定したパラメータをコレクションに追加します。

構文

Visual Basic

```
Public Function Add (ByVal value As Object) As Integer
```

C#

```
public int Add (object value)
```

パラメータ

value コレクションに追加される DBParameter オブジェクト

戻り値

コレクションに追加されたパラメータのインデックス

関連情報

[DBParameter クラス \[33 ページ\]](#)

1.6.3 Clear() メソッド

すべてのパラメータをコレクションから削除します。

構文

Visual Basic

```
Public Sub Clear ()
```

C#

```
public void Clear ()
```

1.6.4 Contains メソッド

指定した DBParameter がコレクションに含まれる場合は true を返します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public bool	Contains(object) [44 ページ]	指定した DBParameter がコレクションに含まれる場合は true を返します。
public bool	Contains(string) [44 ページ]	指定した名前のパラメータがコレクションに含まれているかどうかをチェックします。

このセクションの内容:

[Contains\(object\) メソッド \[44 ページ\]](#)

指定した DBParameter がコレクションに含まれる場合は true を返します。

[Contains\(string\) メソッド \[44 ページ\]](#)

指定した名前のパラメータがコレクションに含まれているかどうかをチェックします。

1.6.4.1 Contains(object) メソッド

指定した DBParameter がコレクションに含まれる場合は true を返します。

構文

Visual Basic

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

C#

```
public bool Contains (object value)
```

パラメータ

value チェック対象の DBParameter オブジェクト

戻り値

このコレクションに DBParameter が含まれる場合は true、そうでない場合は false

関連情報

[DBParameter クラス \[33 ページ\]](#)

1.6.4.2 Contains(string) メソッド

指定した名前のパラメータがコレクションに含まれているかどうかをチェックします。

構文

Visual Basic

```
Public Function Contains (ByVal parameterName As String) As Boolean
```

C#

```
public bool Contains (string parameterName)
```

パラメータ

parameterName 確認するパラメータの名前

戻り値

このコレクションに指定したパラメータが含まれる場合は true、そうでない場合は false

1.6.5 CopyTo(Array, int) メソッド

コレクションの内容を、指定したインデックスで始まる特定の配列にコピーします。

構文

Visual Basic

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

C#

```
public void CopyTo (  
    Array array,  
    int index  
)
```

パラメータ

array コレクションの内容のコピー先となる配列

index 配列の内容のコレクションへのコピーを開始する、配列内のインデックス

1.6.6 GetEnumerator() メソッド

コレクションの列挙子を返します。

構文

Visual Basic

```
Public Function GetEnumerator () As System.Collections.IEnumerator
```

C#

```
public IEnumerator GetEnumerator ()
```

戻り値

コレクションの列挙子

1.6.7 IndexOf メソッド

コレクション内の指定された DBParameter のインデックスを返します。

オーバードリスト

変数とタイプ	オーバーロード名	説明
public int	IndexOf(object) [46 ページ]	コレクション内の指定された DBParameter のインデックスを返します。
public int	IndexOf(string) [47 ページ]	指定した名前を持つパラメータのコレクション内のインデックスを返します。

このセクションの内容:

[IndexOf\(object\) メソッド \[46 ページ\]](#)

コレクション内の指定された DBParameter のインデックスを返します。

[IndexOf\(string\) メソッド \[47 ページ\]](#)

指定した名前を持つパラメータのコレクション内のインデックスを返します。

1.6.7.1 IndexOf(object) メソッド

コレクション内の指定された DBParameter のインデックスを返します。

 構文

Visual Basic

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

C#

```
public int IndexOf (object value)
```

パラメータ

value 検索する DBParameter オブジェクト

戻り値

コレクション内の DBParameter のインデックス

関連情報

[DBParameter クラス \[33 ページ\]](#)

1.6.7.2 IndexOf(string) メソッド

指定した名前を持つパラメータのコレクション内のインデックスを返します。

構文

Visual Basic

```
Public Function IndexOf (ByVal parameterName As String) As Integer
```

C#

```
public int IndexOf (string parameterName)
```

パラメータ

parameterName 検索するパラメータの名前

戻り値

パラメータのインデックス、指定した名前のパラメータがない場合は -1

1.6.8 Insert(int, object) メソッド

指定した DBParameter を、コレクション内の指定したインデックス位置に挿入します。

構文

Visual Basic

```
Public Sub Insert (  
    ByVal index As Integer,  
    ByVal value As Object  
)
```

C#

```
public void Insert (  
    int index,  
    object value  
)
```

パラメータ

value 挿入する DBParameter オブジェクト

index 値を挿入するインデックス位置

関連情報

[DBParameter クラス \[33 ページ\]](#)

1.6.9 Remove(object) メソッド

指定した DBParameter をコレクションから削除します。

構文

Visual Basic

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

C#

```
public void Remove (object value)
```

パラメータ

value 削除する DBParameter

関連情報

[DBParameter クラス \[33 ページ\]](#)

1.6.10 RemoveAt メソッド

コレクション内の指定したインデックス位置の DBParameter を削除します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public void	RemoveAt(int) [50 ページ]	コレクション内の指定したインデックス位置の DBParameter を削除します。
public void	RemoveAt(string) [50 ページ]	指定した名前のパラメータをコレクションから削除します。

このセクションの内容:

[RemoveAt\(int\) メソッド \[50 ページ\]](#)

コレクション内の指定したインデックス位置の DBParameter を削除します。

[RemoveAt\(string\) メソッド \[50 ページ\]](#)

指定した名前のパラメータをコレクションから削除します。

1.6.10.1 RemoveAt(int) メソッド

コレクション内の指定したインデックス位置の DBParameter を削除します。

構文

Visual Basic

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

C#

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

パラメータ

index 削除する DBParameter のインデックス

関連情報

[DBParameter クラス \[33 ページ\]](#)

1.6.10.2 RemoveAt(string) メソッド

指定した名前のパラメータをコレクションから削除します。

構文

Visual Basic

```
Public Sub RemoveAt (ByVal parameterName As String)
```

C#

```
public void RemoveAt (string parameterName)
```

パラメータ

parameterName 削除するパラメータの名前

1.6.11 Count プロパティ

コレクション内のパラメータ数。

構文

Visual Basic

```
Public ReadOnly Property Count As Integer
```

C#

```
public int Count {get;}
```

1.6.12 IsFixedSize プロパティ

false を返します。

構文

Visual Basic

```
Public ReadOnly Property IsFixedSize As Boolean
```

C#

```
public bool IsFixedSize {get;}
```

1.6.13 IsReadOnly プロパティ

false を返します。

構文

Visual Basic

```
Public ReadOnly Property IsReadOnly As Boolean
```

C#

```
public bool IsReadOnly {get;}
```

1.6.14 IsSynchronized プロパティ

false を返します。

構文

Visual Basic

```
Public ReadOnly Property IsSynchronized As Boolean
```

C#

```
public bool IsSynchronized {get;}
```

1.6.15 SyncRoot プロパティ

DBParameterCollection へのアクセスを同期するために使用します。

構文

Visual Basic

```
Public ReadOnly Property SyncRoot As Object
```

C#

```
public object SyncRoot {get;}
```

1.6.16 this プロパティ

コレクション内の指定したインデックスを持つ DBParameter を取得または設定します。

オーバーロードリスト

変数とタイプ	オーバーロード名	説明
public object	this[int index] [53 ページ]	コレクション内の指定したインデックスを持つ DBParameter を取得または設定します。
public object	this[string parameterName] [53 ページ]	コレクション内の指定した名前を持つ DBParameter を取得または設定します。

このセクションの内容:

[this\[int index\] プロパティ \[53 ページ\]](#)

コレクション内の指定したインデックスを持つ DBParameter を取得または設定します。

[this\[string parameterName\] プロパティ \[53 ページ\]](#)

コレクション内の指定した名前を持つ DBParameter を取得または設定します。

1.6.16.1 this[int index] プロパティ

コレクション内の指定したインデックスを持つ DBParameter を取得または設定します。

構文

Visual Basic

```
Public Property Item (ByVal indexAs Integer) As Object
```

C#

```
public object this[int index] {get;set;}
```

戻り値

コレクション内の指定したインデックスを持つ this

関連情報

[DBParameter クラス \[33 ページ\]](#)

1.6.16.2 this[string parameterName] プロパティ

コレクション内の指定した名前を持つ DBParameter を取得または設定します。

構文

Visual Basic

```
Public Property Item (ByVal parameterNameAs String) As Object
```

C#

```
public object this[string parameterName] {get;set;}
```

戻り値

コレクション内の指定した名前を持つ this

関連情報

[DBParameter クラス \[33 ページ\]](#)

1.7 DBRowReader インタフェース

データベースから読み込まれるローセットを表します。

構文

Visual Basic

```
Public Interface DBRowReader
```

C#

```
public interface DBRowReader
```

メンバー

DBRowReader 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() [55 ページ]	この MLDBRowReader で使用されるすべてのリソースをクリーンアップします。
public object[]	NextRow() [56 ページ]	結果セット内の次のローを取得して返します。

プロパティ

変更子とタイプ	プロパティ	説明
public string[]	ColumnNames [57 ページ]	結果セットに含まれるすべてのカラムの名前を取得します。
public SQLType[]	ColumnTypes [57 ページ]	結果セットに含まれるすべてのカラムの型を取得します。

備考

ExecuteReader メソッドを実行すると、DBRowReader が作成されます。

例

次の C# コードは、結果セット内で所定の DBRowReader で表されるローの関数を呼び出します。

```
DBCommand stmt = conn.CreateCommand();
stmt.CommandText = "select intCol, strCol from table1 ";
DBRowReader rs = stmt.ExecuteReader();
object[] values = rset.NextRow();
while( values != null ) {
    handleRow( (int)values[0], (String)values[1] );
    values = rset.NextRow();
}
rset.Close();
stmt.Close();
```

このセクションの内容:

[Close\(\) メソッド \[55 ページ\]](#)

この MLDBRowReader で使用されるすべてのリソースをクリーンアップします。

[NextRow\(\) メソッド \[56 ページ\]](#)

結果セット内の次のローを取得して返します。

[ColumnNames プロパティ \[57 ページ\]](#)

結果セットに含まれるすべてのカラムの名前を取得します。

[ColumnTypes プロパティ \[57 ページ\]](#)

結果セットに含まれるすべてのカラムの型を取得します。

関連情報

[ExecuteReader\(\) メソッド \[21 ページ\]](#)

1.7.1 Close() メソッド

この MLDBRowReader で使用されるすべてのリソースをクリーンアップします。

構文

Visual Basic

```
Public Sub Close ()
```

C#

```
public void Close ()
```

備考

このメソッドを呼び出した後は、この MLDBRowReader を再び使用することはできません。

1.7.2 NextRow() メソッド

結果セット内の次のローを取得して返します。

構文

Visual Basic

```
Public Function NextRow () As Object()
```

C#

```
public object[] NextRow ()
```

戻り値

結果セット内の次のローの値、この結果セットにローがこれ以上ない場合は NULL

関連情報

[SQLType 列挙体 \[114 ページ\]](#)

1.7.3 ColumnNames プロパティ

結果セットに含まれるすべてのカラムの名前を取得します。

構文

Visual Basic

```
Public ReadOnly Property ColumnNames As String()
```

C#

```
public string[] ColumnNames {get;}
```

備考

値は、結果セット内のカラム名に相当する文字列の配列です。

1.7.4 ColumnTypes プロパティ

結果セットに含まれるすべてのカラムの型を取得します。

構文

Visual Basic

```
Public ReadOnly Property ColumnTypes As SQLType()
```

C#

```
public SQLType[] ColumnTypes {get;}
```

備考

値は、結果セット内のカラムの型に対応する SQLTypes の配列です。

1.8 DownloadData インタフェース

ダイレクトローハンドリングで使用するすべてのダウンロードデータ操作をカプセル化します。

構文

Visual Basic

```
Public Interface DownloadData
```

C#

```
public interface DownloadData
```

メンバー

DownloadData 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

変数とタイプ	メソッド	説明
public DownloadTableData	GetDownloadTableByName(string) [59 ページ]	現在の同期に使用する名前付きダウンロードテーブルを取得します。
public DownloadTableData[]	GetDownloadTables() [59 ページ]	現在の同期のダウンロードのすべてのテーブルを含む配列を取得します。

備考

DownloadData のインスタンスを取得するには、GetDownloadData メソッドを使用します。DownloadTableData のインスタンスを返すには、GetDownloadTables メソッドと GetDownloadTableByName メソッドを使用します。

ダウンロードデータは DBConnectionContext を使用して取得できます。begin_sync イベントの前にダウンロードデータにアクセスしたり、ダウンロード専用同期で DownloadData にアクセスしたりすることはできません。

ダイレクトローハンドリングの詳細については、handle_DownloadData 接続イベントとダイレクトローハンドリングを参照してください。

このセクションの内容:

[GetDownloadTableByName\(string\) メソッド \[59 ページ\]](#)

現在の同期に使用する名前付きダウンロードテーブルを取得します。

[GetDownloadTables\(\) メソッド \[59 ページ\]](#)

現在の同期のダウンロードのすべてのテーブルを含む配列を取得します。

関連情報

[DownloadTableData インタフェース \[60 ページ\]](#)

[GetDownloadData\(\) メソッド \[28 ページ\]](#)

1.8.1 GetDownloadTableByName(string) メソッド

現在の同期に使用する名前付きダウンロードテーブルを取得します。

構文

Visual Basic

```
Public Function GetDownloadTableByName (ByVal table_name As String) As DownloadTableData
```

C#

```
public DownloadTableData GetDownloadTableByName (string table_name)
```

パラメータ

table_name ダウンロードデータの取得先テーブルの名前

戻り値

指定したテーブル名のダウンロードデータ、見つからなかった場合は NULL

1.8.2 GetDownloadTables() メソッド

現在の同期のダウンロードのすべてのテーブルを含む配列を取得します。

構文

Visual Basic

```
Public Function GetDownloadTables () As DownloadTableData()
```

C#

```
public DownloadTableData[] GetDownloadTables ()
```

戻り値

ダウンロードテーブルデータの配列。配列内でのテーブルの順序は、リモートのアップロード順と同じです。

備考

このテーブルに対して実行された操作はリモートデータベースに送信されます。

例

次の例では `GetDownloadTables` メソッドを使用して、現在の同期の `DownloadTableData` オブジェクトの配列を取得します。この例は、`_cc` という `DBConnectionContext` インスタンスがあることを前提としています。

```
// The method used for the handle_DownloadData event.
public void HandleDownload() {
    // Get the DownloadData for the current synchronization.
    DownloadData my_dd = _cc.GetDownloadData();
    // Get an array of tables to set download operations.
    DownloadTableData[] download_tables = my_dd.GetDownloadTables();
    // Get the first table in the DownloadTableData array.
    DownloadTableData my_download_table = download_tables[0];
    // ...
}
```

1.9 DownloadTableData インタフェース

1つのダウンロードテーブルの情報を同期用にカプセル化します。

構文

Visual Basic

```
Public Interface DownloadTableData
```

C#

```
public interface DownloadTableData
```

メンバー

`DownloadTableData` 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

変数とタイプ	メソッド	説明
public IDbCommand	GetDeleteCommand() [62 ページ]	ユーザが削除操作をダウンロードデータ操作に追加できるようにするコマンドを取得します。
public DateTime	GetLastDownloadTime() [63 ページ]	このテーブルの最終ダウンロード時刻を返します。
public string	GetName() [64 ページ]	このインスタンスのテーブル名を取得します。
public DataTable	GetSchemaTable() [64 ページ]	このダウンロードテーブルのメタデータを記述する DataTable を取得します。
public IDbCommand	GetUpsertCommand() [65 ページ]	ユーザがアップサート (更新または挿入) をダウンロードデータ操作に追加できるようにするコマンドを取得します。

備考

このインターフェースを使用して、同期クライアントサイトにダウンロードされるデータ操作を設定します。

例

次のテーブルがあるとします。

```
CREATE TABLE remoteOrders (
  pk INT NOT NULL,
  col1 VARCHAR(200),
  PRIMARY KEY (pk)
);
```

次の例では `GetDownloadTableByName` メソッドを使用して、`remoteOrders` テーブルを表す `DownloadTableData` インスタンスを返します。

```
// The method used for the handle_DownloadData event
public void HandleDownload() {
  // _cc is a DBConnectionContext instance.
  // Get the DownloadData for the current synchronization.
  DownloadData my_dd = _cc.GetDownloadData();
  // Get the DownloadTableData for the remoteOrders table.
  DownloadTableData td = my_dd.GetDownloadTableByName("remoteOrders");
  // User defined-methods to set download operations.
  SetDownloadUpserts(td);
  SetDownloadDeletes(td);
  // ...
}
```

この例では、`SetDownloadInserts` メソッドは `GetUpsertCommand` を使用して、挿入または更新するローのコマンドを取得します。IDbCommand は、リモートデータベースに挿入する値の設定先となるパラメータを保持します。

```
void SetDownloadInserts(DownloadTableData td) {
  IDbCommand upsert_cmd = td.GetUpsertCommand();
  IDataParameterCollection parameters = upsert_cmd.Parameters;
  // The following method calls are the same as the following SQL statement:
  // INSERT INTO remoteOrders(pk, col1) values(2300, "truck");
```

```

((IDataParameter) (parameters[0])).Value = (Int32) 2300;
((IDataParameter) (parameters[1])).Value = (String) "truck";
if (upsert_cmd.ExecuteNonQuery() > 0) {
    // Insert was not filtered.
}
else {
    // Insert was filtered because it was uploaded
    // in the same synchronization.
}
}
}

```

次のメソッドは DownloadTableData.GetDeleteCommand を使用して、削除するローのコマンドを取得します。

```

void SetDownloadDeletes(DownloadTableData td) {
    IDbCommand delete_cmd = t2 download_dd.GetDeleteCommand();
    // The following method calls are the same as the following SQL statement:
    // DELETE FROM remoteOrders where pk = 2300;
    IDataParameterCollection parameters = delete_cmd.Parameters;
    ((IDataParameter) (parameters[0])).Value = (Int32) 2300;
    delete_cmd.ExecuteNonQuery();
}

```

このセクションの内容:

[GetDeleteCommand\(\) メソッド \[62 ページ\]](#)

ユーザが削除操作をダウンロードデータ操作に追加できるようにするコマンドを取得します。

[GetLastDownloadTime\(\) メソッド \[63 ページ\]](#)

このテーブルの最終ダウンロード時刻を返します。

[GetName\(\) メソッド \[64 ページ\]](#)

このインスタンスのテーブル名を取得します。

[GetSchemaTable\(\) メソッド \[64 ページ\]](#)

このダウンロードテーブルのメタデータを記述する DataTable を取得します。

[GetUpsertCommand\(\) メソッド \[65 ページ\]](#)

ユーザがアップサート (更新または挿入) をダウンロードデータ操作に追加できるようにするコマンドを取得します。

1.9.1 GetDeleteCommand() メソッド

ユーザが削除操作をダウンロードデータ操作に追加できるようにするコマンドを取得します。

構文

Visual Basic

```
Public Function GetDeleteCommand () As IDbCommand
```

C#

```
public IDbCommand GetDeleteCommand ()
```

戻り値

ダウンロードでの削除に使用するコマンド

備考

返されるコマンドには、テーブルのプライマリキーカラムと同じ数のパラメータがあります。削除をダウンロードに含めるには、プライマリキーカラムのカラム値を設定し、文を ExecuteNonQuery を使用して実行する必要があります。コマンドの ExecuteNonQuery は、削除操作がフィルタされた場合は 0 を返し、挿入がフィルタされていない場合は 1 を返します。

ローを削除するには、ダウンロード削除操作対象のすべてのプライマリキー値を設定してください。リモートテーブルをトランケートするには、すべてのプライマリキーカラムを NULL に設定します。

関連情報

[DownloadTableData インタフェース \[60 ページ\]](#)

1.9.2 GetLastDownloadTime() メソッド

このテーブルの最終ダウンロード時刻を返します。

構文

Visual Basic

```
Public Function GetLastDownloadTime () As Date
```

C#

```
public DateTime GetLastDownloadTime ()
```

戻り値

このテーブルの最後のダウンロード時刻

備考

これは、テーブルごとのダウンロードイベントの多くで渡される最終ダウンロード時刻と同じです。

最終ダウンロード時刻は、特定の同期に対してテーブルダウンロードデータを生成する場合に便利です。

1.9.3 GetName() メソッド

このインスタンスのテーブル名を取得します。

構文

Visual Basic

```
Public Function GetName () As String
```

C#

```
public string GetName ()
```

戻り値

このインスタンスのテーブル名。

備考

これはユーティリティ関数です。テーブル名には、このインスタンスのスキーマを使用してもアクセスできます。

1.9.4 GetSchemaTable() メソッド

このダウンロードテーブルのメタデータを記述する DataTable を取得します。

構文

Visual Basic

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

C#

```
public DataTable GetSchemaTable ()
```

戻り値

カラムメタデータが記述された DataTable。

備考

DataTable にカラム名情報を含める場合は、カラム名を送信するためのクライアントオプションを指定してください。送信カラム名は、デフォルトで指定されます。

1.9.5 GetUpsertCommand() メソッド

ユーザがアップサート (更新または挿入) をダウンロードデータ操作に追加できるようにするコマンドを取得します。

構文

Visual Basic

```
Public Function GetUpsertCommand () As IDbCommand
```

C#

```
public IDbCommand GetUpsertCommand ()
```

戻り値

ダウンロードの挿入/更新に使用するコマンド

備考

返されるコマンドには、このテーブルのカラムと同じ数のパラメータがあります。挿入/更新をダウンロードに含めるには、挿入するカラム値を設定し、文を ExecuteNonQuery を使用して実行する必要があります。コマンドの ExecuteNonQuery は、挿入操作がフィルタされた場合は 0 を返し、挿入がフィルタされていない場合は 1 を返します。

コマンドに対するパラメータの追加や削除はできません。できるのは値の設定だけです。

関連情報

[DownloadTableData インタフェース \[60 ページ\]](#)

1.10 FatalException クラス

サーバ側で致命的エラーが内部的に発生したため、直ちに停止する必要があることを Mobile Link に通知します。

構文

Visual Basic

```
Public Class FatalException Inherits System.ApplicationException
```

C#

```
public class FatalException : System.ApplicationException
```

メンバー

FatalException のすべてのメンバー (継承されたメンバーも含まれます) を次に示します。

コンストラクタ

変更子とタイプ	コンストラクタ	説明
public	FatalException [66 ページ]	デフォルトのメッセージを持つ FatalException を新規に作成します。

このセクションの内容:

[FatalException コンストラクタ \[66 ページ\]](#)

デフォルトのメッセージを持つ FatalException を新規に作成します。

1.10.1 FatalException コンストラクタ

デフォルトのメッセージを持つ FatalException を新規に作成します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public	FatalException() [67 ページ]	デフォルトのメッセージを持つ FatalException を新規に作成します。

変数とタイプ	オーバーロード名	説明
protected	FatalException(SerializationInfo, StreamingContext) [68 ページ]	特定の SerializationInfo オブジェクトおよび StreamingContext オブジェクトから新しい FatalException を作成します。
public	FatalException(string) [68 ページ]	指定されたメッセージを持つ FatalException を新規に作成します。
public	FatalException(string, Exception) [69 ページ]	指定されたメッセージを持ち、このエラーの原因となった特定の内部例外を含む FatalException を新規に作成します。

このセクションの内容:

[FatalException\(\) コンストラクタ \[67 ページ\]](#)

デフォルトのメッセージを持つ FatalException を新規に作成します。

[FatalException\(SerializationInfo, StreamingContext\) コンストラクタ \[68 ページ\]](#)

特定の SerializationInfo オブジェクトおよび StreamingContext オブジェクトから新しい FatalException を作成します。

[FatalException\(string\) コンストラクタ \[68 ページ\]](#)

指定されたメッセージを持つ FatalException を新規に作成します。

[FatalException\(string, Exception\) コンストラクタ \[69 ページ\]](#)

指定されたメッセージを持ち、このエラーの原因となった特定の内部例外を含む FatalException を新規に作成します。

1.10.1.1 FatalException() コンストラクタ

デフォルトのメッセージを持つ FatalException を新規に作成します。

構文

Visual Basic

```
Public Sub FatalException ()
```

C#

```
public FatalException ()
```

1.10.1.2 FatalException(SerializationInfo, StreamingContext) コンストラクタ

特定の SerializationInfo オブジェクトおよび StreamingContext オブジェクトから新しい FatalException を作成します。

構文

Visual Basic

```
Protected Sub FatalException (  
    ByVal info As SerializationInfo,  
    ByVal context As StreamingContext  
)
```

C#

```
protected FatalException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

パラメータ

info この FatalException の構成に使用する SerializationInfo オブジェクト。

context この FatalException の構成に使用する StreamingContext オブジェクト。

1.10.1.3 FatalException(string) コンストラクタ

指定されたメッセージを持つ FatalException を新規に作成します。

構文

Visual Basic

```
Public Sub FatalException (ByVal message As String)
```

C#

```
public FatalException (string message)
```

パラメータ

message この FatalException のメッセージ。

1.10.1.4 FatalException(string, Exception) コンストラクタ

指定されたメッセージを持ち、このエラーの原因となった特定の内部例外を含む FatalException を新規に作成します。

構文

Visual Basic

```
Public Sub FatalException (  
    ByVal message As String,  
    ByVal ie As Exception  
)
```

C#

```
public FatalException (  
    string message,  
    Exception ie  
)
```

パラメータ

message この FatalException のメッセージ。

ie この FatalException の原因となった例外

1.11 LogMessage クラス

ログに出力されたメッセージに関する情報が含まれています。

構文

Microsoft Visual Basic

```
Public Class LogMessage
```

C#

```
public class LogMessage
```

メンバー

LogMessage 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。

列挙体

変更子とタイプ	列挙体	説明
Public	MessageType [71 ページ]	可能性のあるすべての LogMessage タイプを列挙します。

コンストラクタ

変更子とタイプ	コンストラクタ	説明
public	LogMessage(MessageType, string, string) [71 ページ]	指定された属性で LogMessage を作成します。

プロパティ

変更子とタイプ	プロパティ	説明
public string	テキスト [72 ページ]	メッセージの本文。
public MessageType	タイプ [72 ページ]	このインスタンスが表すログメッセージのタイプ。
public string	ユーザ [72 ページ]	このメッセージがログに記録されるユーザ。

備考

このクラスのインスタンスは、LogCallback に渡されます。

このセクションの内容:

[LogMessage\(MessageType, string, string\) コンストラクタ \[71 ページ\]](#)

指定された属性で LogMessage を作成します。

[MessageType 列挙体 \[71 ページ\]](#)

可能性のあるすべての LogMessage タイプを列挙します。

[Text プロパティ \[72 ページ\]](#)

メッセージの本文。

[Type プロパティ \[72 ページ\]](#)

このインスタンスが表すログメッセージのタイプ。

[User プロパティ \[72 ページ\]](#)

このメッセージがログに記録されるユーザ。

関連情報

[LogCallback\(ServerContext, LogMessage\) デリゲート \[112 ページ\]](#)

1.11.1 LogMessage(MessageType, string, string) コンストラクタ

指定された属性で LogMessage を作成します。

構文

Visual Basic

```
Public Sub LogMessage (  
    ByVal type As MessageType,  
    ByVal user As String,  
    ByVal text As String  
)
```

C#

```
public LogMessage (  
    MessageType type,  
    string user,  
    string text  
)
```

1.11.2 MessageType 列挙体

可能性のあるすべての LogMessage タイプを列挙します。

構文

Visual Basic

```
Public Enum MessageType
```

C#

```
enum MessageType
```

メンバー

メンバー名	説明
ERROR	ログメッセージはエラーです。
WARNING	ログメッセージは警告です。
INFO	ログ情報メッセージです。

1.11.3 Text プロパティ

メッセージの本文。

構文

Visual Basic

```
Public ReadOnly Property Text As String
```

C#

```
public string Text {get;}
```

1.11.4 Type プロパティ

このインスタンスが表すログメッセージのタイプ。

構文

Visual Basic

```
Public ReadOnly Property Type As MessageType
```

C#

```
public MessageType Type {get;}
```

1.11.5 User プロパティ

このメッセージがログに記録されるユーザ。

構文

Visual Basic

```
Public ReadOnly Property User As String
```

C#

```
public string User {get;}
```

備考

このプロパティには NULL を指定できます。

1.12 NetworkData インタフェース

同期用のネットワークストリームの情報を含みます。

構文

Visual Basic

```
Public Interface NetworkData
```

C#

```
public interface NetworkData
```

メンバー

NetworkData 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

変数とタイプ	メソッド	説明
public string	GetHTTPHeaderValue(string) [75 ページ]	指定した名前前でサーバが受信した最終ヘッダ値を返します。
public IList< string >	GetHTTPHeaderValues(string) [76 ページ]	指定した名前と関連付けられている、サーバが受信したすべてのヘッダ値を返します。

プロパティ

変数とタイプ	プロパティ	説明
public X509Certificate2Collection	ClientCertificates [77 ページ]	クライアントから送信されたすべての証明書を含む X509Certificate2Collection を返します。
public IDictionary< string, IList< string > >	HTTPHeaders [77 ページ]	ヘッダ名をヘッダ値のリストとマップする辞書を返します。
public bool	IsEndToEndEncrypted [78 ページ]	同期がエンドツーエンドで暗号化されるか確認します。
public bool	IsHTTP [78 ページ]	同期で HTTP または HTTPS のどちらが使用されるかを確認します。

変更子とタイプ	プロパティ	説明
public bool	IsTLS [79 ページ]	同期で TLS が使用されるかを確認します。

備考

このインターフェースは、クライアント側証明書と HTTP ヘッダを使用するエンタープライズ内の別のサーバを認証する際に便利です。

ネットワークストリームのデータの収集を有効にするには、-x スイッチに collect_network_data=1 を追加します。このオプションにより、同期ごとのデータの保存時に、メモリのオーバーヘッドが増えます。TLS または HTTPS をクライアント側証明書で使用する場合に、trusted_certificates=<certificate file> を追加すると、サーバは TLS ハンドシェイクの間に証明書を送信するようクライアントに要求します。これにより、時間とネットワークコストがかかるようになります。

NetworkData オブジェクトは、DBConnectionContext インターフェースの NetworkData メソッドを呼び出して取得できます。HTTP または HTTPS を使用する場合、これには、認証スクリプトが呼び出される前にサーバが受信した、最終の HTTP 要求のヘッダデータが含まれます。

例

次に、DBConnectionContext オブジェクトから NetworkData オブジェクトを取得し、データを出力する例を示します。

```
using Sap.MobiLink.Script;
using System.Collections.Generic;
using System.Security.Cryptography.X509Certificates;
public class OrderProcessor {
    DBConnectionContext _cc;
    public OrderProcessor( DBConnectionContext cc ) {
        _cc = cc;
    }
    public void AuthUser() {
        NetworkData nd = _cc.NetworkData;
        if( nd != null ) {
            if( nd.IsHTTP ) {
                PrintLn( "http" );
                string user_agent = nd.GetHTTPHeaderValue( "user-agent" );
                PrintLn( " user-agent: " + user_agent.Substring( 0,
user_agent.IndexOf( '/' ) ) );
            } else {
                PrintLn( "no http" );
            }
            if( nd.IsTLS ) {
                PrintLn( "tls" );
                X509Certificate2Collection certs = nd.ClientCertificates;
                if( certs != null ) {
                    PrintLn( " client-side cert:" );
                    int n = 1;
                    foreach( X509Certificate2 x509 in certs ) {
                        PrintLn( " cert " + n++ );
                        PrintLn( " Subject: " + x509.SubjectName.Name );
                        PrintLn( " Issuer: " + x509.IssuerName.Name );
                    }
                } else {
                    PrintLn( " no client cert" );
                }
            } else {
                PrintLn( "no tls" );
            }
        }
    }
}
```

```

        if( nd.IsEndToEndEncrypted ) {
            PrintLn( "e2ee" );
        } else {
            PrintLn( "no e2ee" );
        }
    } else {
        PrintLn( "NULL networkdata" );
    }
}
}
}

```

このセクションの内容:

[GetHTTPHeaderValue\(string\) メソッド \[75 ページ\]](#)

指定した名前でサーバが受信した最終ヘッダ値を返します。

[GetHTTPHeaderValues\(string\) メソッド \[76 ページ\]](#)

指定した名前と関連付けられている、サーバが受信したすべてのヘッダ値を返します。

[ClientCertificates プロパティ \[77 ページ\]](#)

クライアントから送信されたすべての証明書を含む X509Certificate2Collection を返します。

[HTTPHeaders プロパティ \[77 ページ\]](#)

ヘッダ名をヘッダ値のリストとマップする辞書を返します。

[IsEndToEndEncrypted プロパティ \[78 ページ\]](#)

同期がエンドツーエンドで暗号化されるか確認します。

[IsHTTP プロパティ \[78 ページ\]](#)

同期で HTTP または HTTPS のどちらが使用されるかを確認します。

[IsTLS プロパティ \[79 ページ\]](#)

同期で TLS が使用されるかを確認します。

1.12.1 GetHTTPHeaderValue(string) メソッド

指定した名前でサーバが受信した最終ヘッダ値を返します。

構文

Visual Basic

```
Public Function GetHTTPHeaderValue (ByVal name As String) As String
```

C#

```
public string GetHTTPHeaderValue (string name)
```

パラメータ

name 値を返すヘッダの名前

戻り値

指定したヘッダ名に関連付けられている最終ヘッダ値。

関連情報

[GetHTTPHeaderValues\(string\) メソッド \[76 ページ\]](#)

[HTTPHeaders プロパティ \[77 ページ\]](#)

1.12.2 GetHTTPHeaderValues(string) メソッド

指定した名前と関連付けられている、サーバが受信したすべてのヘッダ値を返します。

構文

Visual Basic

```
Public Function GetHTTPHeaderValues (ByVal name As String) As IList< string >
```

C#

```
public IList< string > GetHTTPHeaderValues (string name)
```

パラメータ

name 値を返すヘッダの名前

戻り値

指定したヘッダ名に関連付けられているヘッダ値。

関連情報

[GetHTTPHeaderValue\(string\) メソッド \[75 ページ\]](#)

[HTTPHeaders プロパティ \[77 ページ\]](#)

1.12.3 ClientCertificates プロパティ

クライアントから送信されたすべての証明書を含む X509Certificate2Collection を返します。

構文

Visual Basic

```
Public ReadOnly Property ClientCertificates As X509Certificate2Collection
```

C#

```
public X509Certificate2Collection ClientCertificates {get;}
```

戻り値

クライアントを識別する X509 証明書を含む X509Certificate2Collection。または、そのような証明書が提供されなかった場合は NULL。

備考

この関数によって NULL 以外の値が返されるのは、isTLS() が true であり、クライアントが「identity」ストリームパラメータを使用して証明書を提供し、サーバ上で trusted_certificates オプションが設定される場合のみです。NULL 以外の CertPath には、自己署名証明書からピアの証明書の順の証明書が含まれます。

1.12.4 HTTPHeaders プロパティ

ヘッダ名をヘッダ値のリストとマップする辞書を返します。

構文

Visual Basic

```
Public ReadOnly Property HTTPHeaders As IDictionary< string, IList< string > >
```

C#

```
public IDictionary< string, IList< string > > HTTPHeaders {get;}
```

戻り値

ヘッダ名と値のペアの辞書。

関連情報

[GetHTTPHeaderValue\(string\) メソッド \[75 ページ\]](#)

[GetHTTPHeaderValues\(string\) メソッド \[76 ページ\]](#)

1.12.5 IsEndToEndEncrypted プロパティ

同期がエンドツーエンドで暗号化されるか確認します。

構文

Visual Basic

```
Public ReadOnly Property IsEndToEndEncrypted As Boolean
```

C#

```
public bool IsEndToEndEncrypted {get;}
```

1.12.6 IsHTTP プロパティ

同期で HTTP または HTTPS のどちらが使用されるかを確認します。

構文

Visual Basic

```
Public ReadOnly Property IsHTTP As Boolean
```

C#

```
public bool IsHTTP {get;}
```

戻り値

同期で HTTP または HTTPS が使用される場合は true、それ以外の場合は false を返します。

1.12.7 IsTLS プロパティ

同期で TLS が使用されるかを確認します。

構文

Visual Basic

```
Public ReadOnly Property IsTLS As Boolean
```

C#

```
public bool IsTLS {get;}
```

戻り値

同期で TLS が使用される場合は true、それ以外の場合は false を返します。

1.13 ScriptExecutionException クラス

ユーザスクリプトでエラーが発生したことを通知します。

構文

Visual Basic

```
Public Class ScriptExecutionException Inherits System.ApplicationException
```

C#

```
public class ScriptExecutionException : System.ApplicationException
```

メンバー

ScriptExecutionException 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。

コンストラクタ

変更子とタイプ	コンストラクタ	説明
public	ScriptExecutionException [80 ページ]	デフォルトのメッセージを持つ ScriptExecutionException を新規に作成します。

備考

この例外または SynchronizationException を除く例外の派生が発行されると、Mobile Link サーバは停止します。

このセクションの内容:

[ScriptExecutionException コンストラクタ \[80 ページ\]](#)

デフォルトのメッセージを持つ ScriptExecutionException を新規に作成します。

1.13.1 ScriptExecutionException コンストラクタ

デフォルトのメッセージを持つ ScriptExecutionException を新規に作成します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public	ScriptExecutionException() [81 ページ]	デフォルトのメッセージを持つ ScriptExecutionException を新規に作成します。
protected	ScriptExecutionException(SerializationInfo, StreamingContext) [81 ページ]	特定の SerializationInfo オブジェクトおよび StreamingContext オブジェクトから新しい ScriptExecutionException を作成します。
public	ScriptExecutionException(string) [82 ページ]	指定されたメッセージを持つ ScriptExecutionException を新規に作成します。
public	ScriptExecutionException(string, Exception) [82 ページ]	指定されたメッセージを持ち、このエラーの原因となった特定の内部例外を含む ScriptExecutionException を新規に作成します。

このセクションの内容:

[ScriptExecutionException\(\) コンストラクタ \[81 ページ\]](#)

デフォルトのメッセージを持つ ScriptExecutionException を新規に作成します。

[ScriptExecutionException\(SerializationInfo, StreamingContext\) コンストラクタ \[81 ページ\]](#)

特定の `SerializationInfo` オブジェクトおよび `StreamingContext` オブジェクトから新しい `ScriptExecutionException` を作成します。

[ScriptExecutionException\(string\) コンストラクタ \[82 ページ\]](#)

指定されたメッセージを持つ `ScriptExecutionException` を新規に作成します。

[ScriptExecutionException\(string, Exception\) コンストラクタ \[82 ページ\]](#)

指定されたメッセージを持ち、このエラーの原因となった特定の内部例外を含む `ScriptExecutionException` を新規に作成します。

1.13.1.1 ScriptExecutionException() コンストラクタ

デフォルトのメッセージを持つ `ScriptExecutionException` を新規に作成します。

構文

Visual Basic

```
Public Sub ScriptExecutionException ()
```

C#

```
public ScriptExecutionException ()
```

1.13.1.2 ScriptExecutionException(SerializationInfo, StreamingContext) コンストラクタ

特定の `SerializationInfo` オブジェクトおよび `StreamingContext` オブジェクトから新しい `ScriptExecutionException` を作成します。

構文

Visual Basic

```
Protected Sub ScriptExecutionException (  
    ByVal info As SerializationInfo,  
    ByVal context As StreamingContext  
)
```

C#

```
protected ScriptExecutionException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

パラメータ

info この `ScriptExecutionException` を構成する `SerializationInfo` オブジェクト。

context この `ScriptExecutionException` を構成する `StreamingContext` オブジェクト。

1.13.1.3 `ScriptExecutionException(string)` コンストラクタ

指定されたメッセージを持つ `ScriptExecutionException` を新規に作成します。

構文

Visual Basic

```
Public Sub ScriptExecutionException (ByVal message As String)
```

C#

```
public ScriptExecutionException (string message)
```

パラメータ

message この `ScriptExecutionException` のメッセージ。

1.13.1.4 `ScriptExecutionException(string, Exception)` コンストラクタ

指定されたメッセージを持ち、このエラーの原因となった特定の内部例外を含む `ScriptExecutionException` を新規に作成します。

構文

Visual Basic

```
Public Sub ScriptExecutionException (  
    ByVal message As String,  
    ByVal ie As Exception  
)
```

C#

```
public ScriptExecutionException (  
    string message,  
    Exception ie  
)
```

パラメータ

message この ScriptExecutionException のメッセージ。

ie この ScriptExecutionException の原因となった例外

1.14 ServerContext インタフェース

Mobile Link サーバの継続期間中に存在する、すべてのコンテキストをインスタンス化します。

構文

Visual Basic

```
Public Interface ServerContext
```

C#

```
public interface ServerContext
```

メンバー

ServerContext 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

変数とタイプ	メソッド	説明
public NameValueCollection	getProperties(string, string) [85 ページ]	指定したコンポーネントとセットのプロパティのコレクションを返します。
public NameValueCollection	getPropertiesByVersion(string) [86 ページ]	指定したスクリプトバージョンのプロパティのコレクションを返します。
public StringCollection	getPropertySetNames(string) [86 ページ]	指定したコンポーネントのセット名のコレクションを返します。
public object[]	GetStartClassInstances() [87 ページ]	このドメインにロードされたすべての起動クラスを返します。
public DBConnection	MakeConnection() [88 ページ]	新しいデータベース接続を作成します。
public void	Shutdown() [88 ページ]	Mobile Link サーバがソフトシャットダウンを実行するようにします。

イベント

変更子とタイプ	イベント	説明
public LogCallback	ErrorListener [89 ページ]	Mobile Link サーバがエラーを出力したときにトリガされます。
public LogCallback	InfoListener [89 ページ]	Mobile Link サーバが情報を出力したときにトリガされます。
public ShutdownCallback	ShutdownListener [89 ページ]	Mobile Link サーバが停止するときにトリガされます。
public LogCallback	WarningListener [90 ページ]	Mobile Link サーバが警告を出力したときにトリガされます。

備考

このコンテキストを静的なデータとして保持し、バックグラウンドスレッドで使用できます。Mobile Link で起動される .NET CLR の継続期間中は有効です。

ServerContext インスタンスにアクセスするには、GetServerContext メソッドを使用します。このインスタンスは、起動クラスのコンストラクタに渡されます。

このセクションの内容:

[getProperties\(string, string\) メソッド \[85 ページ\]](#)

指定したコンポーネントとセットのプロパティのコレクションを返します。

[getPropertiesByVersion\(string\) メソッド \[86 ページ\]](#)

指定したスクリプトバージョンのプロパティのコレクションを返します。

[getPropertySetNames\(string\) メソッド \[86 ページ\]](#)

指定したコンポーネントのセット名のコレクションを返します。

[GetStartClassInstances\(\) メソッド \[87 ページ\]](#)

このドメインにロードされたすべての起動クラスを返します。

[MakeConnection\(\) メソッド \[88 ページ\]](#)

新しいデータベース接続を作成します。

[Shutdown\(\) メソッド \[88 ページ\]](#)

Mobile Link サーバがソフトシャットダウンを実行するようにします。

[ErrorListener イベント \[89 ページ\]](#)

Mobile Link サーバがエラーを出力したときにトリガされます。

[InfoListener イベント \[89 ページ\]](#)

Mobile Link サーバが情報を出力したときにトリガされます。

[ShutdownListener イベント \[89 ページ\]](#)

Mobile Link サーバが停止するときにトリガされます。

[WarningListener イベント \[90 ページ\]](#)

Mobile Link サーバが警告を出力したときにトリガされます。

関連情報

[GetServerContext\(\) メソッド \[31 ページ\]](#)

1.14.1 getProperties(string, string) メソッド

指定したコンポーネントとセットのプロパティのコレクションを返します。

構文

Visual Basic

```
Public Function getProperties (  
    ByVal component As String,  
    ByVal set As String  
) As NameValueCollection
```

C#

```
public NameValueCollection getProperties (  
    string component,  
    string set  
)
```

パラメータ

component ml_property テーブルの component_name カラムを表します。

set ml_property テーブルの property_set_name カラムを表します。

戻り値

指定したコンポーネント/セットのプロパティ

備考

プロパティは、ml_property テーブルに格納されます。詳細については、ml_add_property システムプロシージャを参照してください。

1.14.2 getPropertiesByVersion(string) メソッド

指定したスクリプトバージョンのプロパティのコレクションを返します。

構文

Visual Basic

```
Public Function getPropertiesByVersion (ByVal script_version As String) As  
    NameValueCollection
```

C#

```
public NameValueCollection getPropertiesByVersion (string script_version)
```

パラメータ

script_version 関連するプロパティを返すスクリプトバージョン

戻り値

指定したスクリプトバージョンのプロパティ

備考

プロパティは、ml_property テーブルに格納されます。詳細については、ml_add_property システムプロシージャを参照してください。

1.14.3 getPropertySetNames(string) メソッド

指定したコンポーネントのセット名のコレクションを返します。

構文

Visual Basic

```
Public Function getPropertySetNames (ByVal component As String) As  
    StringCollection
```

C#

```
public StringCollection getPropertySetNames (string component)
```

パラメータ

component ml_property テーブルの component_name カラムを表します。

戻り値

指定したコンポーネントのセット名のコレクション

備考

プロパティは、ml_property テーブルに格納されます。詳細については、ml_add_property システムプロシージャを参照してください。

1.14.4 GetStartClassInstances() メソッド

このドメインにロードされたすべての起動クラスを返します。

構文

Visual Basic

```
Public Function GetStartClassInstances () As Object()
```

C#

```
public object[] GetStartClassInstances ()
```

戻り値

サーバ起動時に構築されたすべての起動クラスの配列。起動クラスがない場合、配列の長さは 0 です。

備考

ユーザ定義起動クラスの詳細については、"ユーザ定義起動クラス" を参照してください。

例

次の例は、起動クラスの検索方法を示します。

```
void FindStartClass( ServerContext sc, string name )
{
    object[] startClasses = sc.GetStartClassInstances();
    foreach( object obj in startClasses ) {
        if( obj is MyClass ) {
            // Execute some code.....
        }
    }
}
```

1.14.5 MakeConnection() メソッド

新しいデータベース接続を作成します。

構文

Visual Basic

```
Public Function MakeConnection () As DBConnection
```

C#

```
public DBConnection MakeConnection ()
```

戻り値

新しい接続

1.14.6 Shutdown() メソッド

Mobile Link サーバがソフトシャットダウンを実行するようにします。

構文

Visual Basic

```
Public Sub Shutdown ()
```

C#

```
public void Shutdown ()
```

1.14.7 ErrorListener イベント

Mobile Link サーバがエラーを出力したときにトリガされます。

構文

Visual Basic

```
Public Event ErrorListener As LogCallback
```

C#

```
public LogCallback ErrorListener;
```

1.14.8 InfoListener イベント

Mobile Link サーバが情報を出力したときにトリガされます。

構文

Visual Basic

```
Public Event InfoListener As LogCallback
```

C#

```
public LogCallback InfoListener;
```

1.14.9 ShutdownListener イベント

Mobile Link サーバが停止するときにトリガされます。

構文

Visual Basic

```
Public Event ShutdownListener As ShutdownCallback
```

C#

```
public ShutdownCallback ShutdownListener;
```

1.14.10 WarningListener イベント

Mobile Link サーバが警告を出力したときにトリガされます。

構文

Visual Basic

```
Public Event WarningListener As LogCallback
```

C#

```
public LogCallback WarningListener;
```

1.15 ServerException クラス

サーバでエラーが発生したため直ちに停止する必要があることを Mobile Link に通知します。

構文

Visual Basic

```
Public Class ServerException Inherits ScriptExecutionException
```

C#

```
public class ServerException : ScriptExecutionException
```

メンバー

ServerException 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。

コンストラクタ

変更子とタイプ	コンストラクタ	説明
public	ServerException [91 ページ]	デフォルトのメッセージを持つ ServerException を新規に作成します。

ScriptExecutionException から継承されたメンバー

変更子とタイプ	メンバー	説明
public	ScriptExecutionException() [81 ページ]	デフォルトのメッセージを持つ ScriptExecutionException を新規に作成します。

変数とタイプ	メンバー	説明
public	ScriptExecutionException(string) [82 ページ]	指定されたメッセージを持つ ScriptExecutionException を新規に作成します。
protected	ScriptExecutionException(SerializationInfo, StreamingContext) [81 ページ]	特定の SerializationInfo オブジェクトおよび StreamingContext オブジェクトから新しい ScriptExecutionException を作成します。
public	ScriptExecutionException(string, Exception) [82 ページ]	指定されたメッセージを持ち、このエラーの原因となった特定の内部例外を含む ScriptExecutionException を新規に作成します。

このセクションの内容:

[ServerException コンストラクタ \[91 ページ\]](#)

デフォルトのメッセージを持つ ServerException を新規に作成します。

1.15.1 ServerException コンストラクタ

デフォルトのメッセージを持つ ServerException を新規に作成します。

オーバーロードリスト

変数とタイプ	オーバーロード名	説明
public	ServerException() [92 ページ]	デフォルトのメッセージを持つ ServerException を新規に作成します。
protected	ServerException(SerializationInfo, StreamingContext) [92 ページ]	特定の SerializationInfo オブジェクトおよび StreamingContext オブジェクトから新しい ServerException を作成します。
public	ServerException(string) [93 ページ]	指定されたメッセージを持つ ServerException を新規に作成します。
public	ServerException(string, Exception) [93 ページ]	指定されたメッセージを持ち、このエラーの原因となった特定の内部例外を含む ServerException を新規に作成します。

このセクションの内容:

[ServerException\(\) コンストラクタ \[92 ページ\]](#)

デフォルトのメッセージを持つ ServerException を新規に作成します。

[ServerException\(SerializationInfo, StreamingContext\) コンストラクタ \[92 ページ\]](#)

特定の `SerializationInfo` オブジェクトおよび `StreamingContext` オブジェクトから新しい `ServerException` を作成します。

[ServerException\(string\) コンストラクタ \[93 ページ\]](#)

指定されたメッセージを持つ `ServerException` を新規に作成します。

[ServerException\(string, Exception\) コンストラクタ \[93 ページ\]](#)

指定されたメッセージを持ち、このエラーの原因となった特定の内部例外を含む `ServerException` を新規に作成します。

1.15.1.1 ServerException() コンストラクタ

デフォルトのメッセージを持つ `ServerException` を新規に作成します。

構文

Visual Basic

```
Public Sub ServerException ()
```

C#

```
public ServerException ()
```

1.15.1.2 ServerException(SerializationInfo, StreamingContext) コンストラクタ

特定の `SerializationInfo` オブジェクトおよび `StreamingContext` オブジェクトから新しい `ServerException` を作成します。

構文

Visual Basic

```
Protected Sub ServerException (  
    ByVal info As SerializationInfo,  
    ByVal context As StreamingContext  
)
```

C#

```
protected ServerException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

パラメータ

info この `ServerException` を構成する `SerializationInfo` オブジェクト。

context この `ServerException` を構成する `StreamingContext` オブジェクト。

1.15.1.3 `ServerException(string)` コンストラクタ

指定されたメッセージを持つ `ServerException` を新規に作成します。

構文

Visual Basic

```
Public Sub ServerException (ByVal message As String)
```

C#

```
public ServerException (string message)
```

パラメータ

message この `ServerException` のメッセージ。

1.15.1.4 `ServerException(string, Exception)` コンストラクタ

指定されたメッセージを持ち、このエラーの原因となった特定の内部例外を含む `ServerException` を新規に作成します。

構文

Visual Basic

```
Public Sub ServerException (  
    ByVal message As String,  
    ByVal ie As Exception  
)
```

C#

```
public ServerException (  
    string message,  
    Exception ie  
)
```

パラメータ

message この ServerException のメッセージ。

ie この ServerException の原因となった例外

1.16 SpatialUtilities クラス

空間値を操作する静的メソッドのコレクションを表します。

構文

Visual Basic

```
Public NotInheritable Class SpatialUtilities
```

C#

```
public sealed class SpatialUtilities
```

メンバー

SpatialUtilities 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

変数とタイプ	メソッド	説明
public static byte[]	CreateSpatialValue(int, byte[]) [95 ページ]	ダウンロード用にフォーマットされた空間値を含む新しいバイト配列を返します。最初の 4 バイトにはリトルエンディアンで指定した SRID が含まれ、残りのバイトには指定したバイト配列に渡された空間データが含まれます。
public static byte[]	GetBytes(byte[]) [96 ページ]	指定したバイト配列と同じ空間データを含む新しいバイト配列を返しますが、SRID は削除されています。
public static int	GetSRID(byte[]) [96 ページ]	指定した空間値の SRID を返します。
public static void	SetSRID(byte[], int) [97 ページ]	指定した SRID を、指定したバイト配列の最初の 4 バイトに格納します。

このセクションの内容:

[CreateSpatialValue\(int, byte\[\]\) メソッド \[95 ページ\]](#)

ダウンロード用にフォーマットされた空間値を含む新しいバイト配列を返します。最初の 4 バイトにはリトルエンディアンで指定した SRID が含まれ、残りのバイトには指定したバイト配列に渡された空間データが含まれます。

[GetBytes\(byte\[\]\) メソッド \[96 ページ\]](#)

指定したバイト配列と同じ空間データを含む新しいバイト配列を返しますが、SRID は削除されています。

[GetSRID\(byte\[\]\) メソッド \[96 ページ\]](#)

指定した空間値の SRID を返します。

[SetSRID\(byte\[\], int\) メソッド \[97 ページ\]](#)

指定した SRID を、指定したバイト配列の最初の 4 バイトに格納します。

1.16.1 CreateSpatialValue(int, byte[]) メソッド

ダウンロード用にフォーマットされた空間値を含む新しいバイト配列を返します。最初の 4 バイトにはリトルエンディアンで指定した SRID が含まれ、残りのバイトには指定したバイト配列に渡された空間データが含まれます。

構文

Visual Basic

```
Public Shared Function CreateSpatialValue (  
    ByVal srid As Integer,  
    ByVal spatial_value As Byte()  
) As Byte()
```

C#

```
public static byte[] CreateSpatialValue (  
    int srid,  
    byte[] spatial_value  
)
```

パラメータ

srid SRID。

spatial_value 空間データ。

戻り値

ダウンロード用にフォーマットされた空間値。

1.16.2 GetBytes(byte[]) メソッド

指定したバイト配列と同じ空間データを含む新しいバイト配列を返しますが、SRID は削除されています。

構文

Visual Basic

```
Public Shared Function GetBytes (ByVal spatial_value As Byte()) As Byte()
```

C#

```
public static byte[] GetBytes (byte[] spatial_value)
```

パラメータ

spatial_value SRID を削除する必要がある空間値

戻り値

新しいバイト配列

1.16.3 GetSRID(byte[]) メソッド

指定した空間値の SRID を返します。

構文

Visual Basic

```
Public Shared Function GetSRID (ByVal spatial_value As Byte()) As Integer
```

C#

```
public static int GetSRID (byte[] spatial_value)
```

パラメータ

spatial_value アップロードされた値。最初の 4 バイトには、リトルエンディアンでエンコードされた SRID が含まれている必要があります。

戻り値

SRID。

1.16.4 SetSRID(byte[], int) メソッド

指定した SRID を、指定したバイト配列の最初の 4 バイトに格納します。

構文

Visual Basic

```
Public Shared Sub SetSRID (  
    ByVal spatial_value As Byte(),  
    ByVal srid As Integer  
)
```

C#

```
public static void SetSRID (  
    byte[] spatial_value,  
    int srid  
)
```

パラメータ

spatial_value SRID を格納する配列。

srid 格納する SRID。

1.17 SynchronizationException クラス

同期例外が発生したことで、現在の同期をロールバックして再開する必要があることを示します。

構文

Visual Basic

```
Public Class SynchronizationException Inherits ScriptExecutionException
```

C#

```
public class SynchronizationException : ScriptExecutionException
```

メンバー

SynchronizationException 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。

コンストラクタ

変更子とタイプ	コンストラクタ	説明
public	SynchronizationException [98 ページ]	デフォルトのメッセージを持つ SynchronizationException を新規に作成します。

ScriptExecutionException から継承されたメンバー

変更子とタイプ	メンバー	説明
public	ScriptExecutionException() [81 ページ]	デフォルトのメッセージを持つ ScriptExecutionException を新規に作成します。
public	ScriptExecutionException(string) [82 ページ]	指定されたメッセージを持つ ScriptExecutionException を新規に作成します。
protected	ScriptExecutionException(SerializationInfo, StreamingContext) [81 ページ]	特定の SerializationInfo オブジェクトおよび StreamingContext オブジェクトから新しい ScriptExecutionException を作成します。
public	ScriptExecutionException(string, Exception) [82 ページ]	指定されたメッセージを持ち、このエラーの原因となった特定の内部例外を含む ScriptExecutionException を新規に作成します。

このセクションの内容:

[SynchronizationException コンストラクタ](#) [98 ページ]

デフォルトのメッセージを持つ SynchronizationException を新規に作成します。

1.17.1 SynchronizationException コンストラクタ

デフォルトのメッセージを持つ SynchronizationException を新規に作成します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public	SynchronizationException() [99 ページ]	デフォルトのメッセージを持つ SynchronizationException を新規に作成します。

変数とタイプ	オーバーロード名	説明
protected	SynchronizationException(SerializationInfo, StreamingContext) [100 ページ]	特定の SerializationInfo オブジェクトおよび StreamingContext オブジェクトから新しい SynchronizationException を作成します。
public	SynchronizationException(string) [100 ページ]	指定されたメッセージを持つ SynchronizationException を新規に作成します。
public	SynchronizationException(string, Exception) [101 ページ]	指定されたメッセージを持ち、このエラーの原因となった特定の内部例外を含む SynchronizationException を新規に作成します。

このセクションの内容:

[SynchronizationException\(\) コンストラクタ \[99 ページ\]](#)

デフォルトのメッセージを持つ SynchronizationException を新規に作成します。

[SynchronizationException\(SerializationInfo, StreamingContext\) コンストラクタ \[100 ページ\]](#)

特定の SerializationInfo オブジェクトおよび StreamingContext オブジェクトから新しい SynchronizationException を作成します。

[SynchronizationException\(string\) コンストラクタ \[100 ページ\]](#)

指定されたメッセージを持つ SynchronizationException を新規に作成します。

[SynchronizationException\(string, Exception\) コンストラクタ \[101 ページ\]](#)

指定されたメッセージを持ち、このエラーの原因となった特定の内部例外を含む SynchronizationException を新規に作成します。

1.17.1.1 SynchronizationException() コンストラクタ

デフォルトのメッセージを持つ SynchronizationException を新規に作成します。

構文

Visual Basic

```
Public Sub SynchronizationException ()
```

C#

```
public SynchronizationException ()
```

1.17.1.2 SynchronizationException(SerializationInfo, StreamingContext) コンストラクタ

特定の SerializationInfo オブジェクトおよび StreamingContext オブジェクトから新しい SynchronizationException を作成します。

構文

Visual Basic

```
Protected Sub SynchronizationException (  
    ByVal info As SerializationInfo,  
    ByVal context As StreamingContext  
)
```

C#

```
protected SynchronizationException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

パラメータ

info この SynchronizationException を構成する SerializationInfo オブジェクト。

context この SynchronizationException を構成する StreamingContext オブジェクト。

1.17.1.3 SynchronizationException(string) コンストラクタ

指定されたメッセージを持つ SynchronizationException を新規に作成します。

構文

Visual Basic

```
Public Sub SynchronizationException (ByVal message As String)
```

C#

```
public SynchronizationException (string message)
```

パラメータ

message この SynchronizationException のメッセージ。

1.17.1.4 SynchronizationException(string, Exception) コンストラクタ

指定されたメッセージを持ち、このエラーの原因となった特定の内部例外を含む SynchronizationException を新規に作成します。

構文

Visual Basic

```
Public Sub SynchronizationException (  
    ByVal message As String,  
    ByVal ie As Exception  
)
```

C#

```
public SynchronizationException (  
    string message,  
    Exception ie  
)
```

パラメータ

message この SynchronizationException のメッセージ。
ie この SynchronizationException の原因となった例外

1.18 UpdateDataReader インタフェース

1つのテーブルの1つのアップロードトランザクションのための更新操作を保持します。

構文

Visual Basic

```
Public Interface UpdateDataReader Implements System.Data.IDataReader
```

C#

```
public interface UpdateDataReader : System.Data.IDataReader
```

メンバー

UpdateDataReader のすべてのメンバー (継承されたメンバーも含まれます) を次に示します。

メソッド

変数とタイプ	メソッド	説明
public void	SetNewRowValues() [102 ページ]	新しいカラム値 (更新後のロー) を返すように、この DataReader のモードを設定します。
public void	SetOldRowValues() [103 ページ]	古いカラム値 (更新前のロー) を返すように、この DataReader のモードを設定します。

備考

DataReader のモードを変更することで、新しいローと古いローにアクセスできます。そのほかについては、このインターフェースを通常の DataReader と同様に使用できます。

このセクションの内容:

[SetNewRowValues\(\) メソッド \[102 ページ\]](#)

新しいカラム値 (更新後のロー) を返すように、この DataReader のモードを設定します。

[SetOldRowValues\(\) メソッド \[103 ページ\]](#)

古いカラム値 (更新前のロー) を返すように、この DataReader のモードを設定します。

1.18.1 SetNewRowValues() メソッド

新しいカラム値 (更新後のロー) を返すように、この DataReader のモードを設定します。

構文

Visual Basic

```
Public Sub SetNewRowValues ()
```

C#

```
public void SetNewRowValues ()
```

備考

これがデフォルトモードです。

1.18.2 SetOldRowValues() メソッド

古いカラム値 (更新前のロー) を返すように、この DataReader のモードを設定します。

構文

Visual Basic

```
Public Sub SetOldRowValues ()
```

C#

```
public void SetOldRowValues ()
```

1.19 UploadData インタフェース

ディレクトリハンドリングで使用するアップロード操作をカプセル化します。

構文

Visual Basic

```
Public Interface UploadData
```

C#

```
public interface UploadData
```

メンバー

UploadData 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

変数とタイプ	メソッド	説明
public UploadedTableData	GetUploadedTableByName(string) [104 ページ]	アップロードトランザクションの名前付きアップロードテーブルデータを取得します。
public UploadedTableData[]	GetUploadedTables() [105 ページ]	現在のアップロードトランザクションのすべてのアップロードテーブルデータを含む配列を取得します。

備考

アップロードトランザクションには、ロー操作が格納されたテーブルのセットが含まれています。単一のアップロードトランザクションを表す UploadData インスタンスが、handle_UploadData 同期イベントに渡されます。

i 注記

ダイレクトローハンドリングのアップロード操作は、handle_UploadData イベントに対して登録したメソッドで処理してください。登録されたメソッドを呼び出した後、UploadData は破棄されます。後続のイベントで使用するために新しい UploadData インスタンスを作成しないでください。

UploadedTableData インスタンスを取得するには、UploadData.GetUploadedTables メソッドまたは UploadData.GetUploadedTableByName メソッドを使用します。

リモートデータベースがトランザクションアップロードまたはインクリメンタルアップロードを使用している場合を除き、同期の UploadData は 1 つです。

例

handle_UploadData 接続イベントを参照してください。

このセクションの内容:

[GetUploadedTableByName\(string\) メソッド \[104 ページ\]](#)

アップロードトランザクションの名前付きアップロードテーブルデータを取得します。

[GetUploadedTables\(\) メソッド \[105 ページ\]](#)

現在のアップロードトランザクションのすべてのアップロードテーブルデータを含む配列を取得します。

1.19.1 GetUploadedTableByName(string) メソッド

アップロードトランザクションの名前付きアップロードテーブルデータを取得します。

構文

Visual Basic

```
Public Function GetUploadedTableByName (ByVal table_name As String) As  
    UploadedTableData
```

C#

```
public UploadedTableData GetUploadedTableByName (string table_name)
```

パラメータ

table_name アップロードしたデータの取得先テーブルの名前

戻り値

指定したテーブル名のダウンロードしたデータ、見つからなかった場合は NULL

例

handle_UploadData 同期イベントに対して HandleUpload というメソッドを使用すると仮定した場合、次の例は、GetUploadedTableByName メソッドを使用して、remoteOrders テーブルの UploadedTableData インスタンスを返します。

```
// The method used for the handle_UploadData event.
public void HandleUpload(UploadData ut) {
    UploadedTableData uploaded_t1 = ut.GetUploadedTableByName("remoteOrders");
    // ...
}
```

1.19.2 GetUploadedTables() メソッド

現在のアップロードトランザクションのすべてのアップロードテーブルデータを含む配列を取得します。

構文

Visual Basic

```
Public Function GetUploadedTables () As UploadedTableData()
```

C#

```
public UploadedTableData[] GetUploadedTables ()
```

戻り値

アップロードテーブルデータの配列。配列内でのテーブルの順序は、クライアントのアップロード順と同じです。

備考

テーブルの配列内での順序は、Mobile Link による SQL のローハンドリングでの順序と同じで、参照整合性違反を防ぐ最適な順序になります。データソースがリレーショナルデータベースの場合は、このテーブル順序を使用してください。

例

handle_UploadData 同期イベントに対して HandleUpload というメソッドを使用すると仮定した場合、次の例は、GetUploadedTables メソッドを使用して、現在のアップロードトランザクションの UploadedTableData インスタンスを返します。

```
// The method used for the handle_UploadData event.
public void HandleUpload(UploadData ud) {
    UploadedTableData[] tables = ud.GetUploadedTables();
    //...
}
```

関連情報

[UploadedTableData インタフェース \[106 ページ\]](#)

1.20 UploadedTableData インタフェース

1つのアップロードテーブルの情報を同期用にカプセル化します。

構文

Visual Basic

```
Public Interface UploadedTableData
```

C#

```
public interface UploadedTableData
```

メンバー

UploadedTableData 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

変数とタイプ	メソッド	説明
public IDataReader	GetDeletes() [107 ページ]	このアップロードテーブルデータに対する削除を保持した DataReader を取得します。
public IDataReader	GetInserts() [109 ページ]	このアップロードテーブルデータに対する挿入を保持した DataReader を取得します。
public string	GetName() [110 ページ]	このインスタンスのテーブル名を取得します。
public DataTable	GetSchemaTable() [110 ページ]	このダウンロードテーブルのメタデータを記述する DataTable を取得します。
public UpdateDataReader	GetUpdates() [111 ページ]	このアップロードテーブルデータに対する更新を保持した DataReader を取得します。

備考

挿入、更新、削除の各操作には、標準の ADO.NET IDataReader を使用してアクセスできます。テーブルメタデータへのアクセスには GetSchemaTable 呼び出しまたは挿入および削除データリーダーを使用できます。削除データリーダーには、テーブルのプライマリキーカラムのみが含まれます。

このセクションの内容:

[GetDeletes\(\) メソッド \[107 ページ\]](#)

このアップロードテーブルデータに対する削除を保持した DataReader を取得します。

[GetInserts\(\) メソッド \[109 ページ\]](#)

このアップロードテーブルデータに対する挿入を保持した DataReader を取得します。

[GetName\(\) メソッド \[110 ページ\]](#)

このインスタンスのテーブル名を取得します。

[GetSchemaTable\(\) メソッド \[110 ページ\]](#)

このダウンロードテーブルのメタデータを記述する DataTable を取得します。

[GetUpdates\(\) メソッド \[111 ページ\]](#)

このアップロードテーブルデータに対する更新を保持した DataReader を取得します。

1.20.1 GetDeletes() メソッド

このアップロードテーブルデータに対する削除を保持した DataReader を取得します。

構文

Visual Basic

```
Public Function GetDeletes () As IDataReader
```

C#

```
public IDataReader GetDeletes ()
```

戻り値

削除するローのプライマリキーカラムを保持した DataReader

備考

個々の削除は、このインスタンステーブルのローを一意に表すプライマリキー値で表されます。

i 注記

カラムのインデックスと順序は、このテーブルのスキーマのプロパティ DataTable.PrimaryKey の配列と一致します。

例

リモートクライアントに sparse_pk というテーブルがあるものとします。次の例は GetDeletes メソッドを使用して、削除するローのデータリーダーを取得します。この場合、削除 DataReader には 2 つのプライマリキーカラムが含まれます。各プライマリキーカラムのインデックスに注目してください。

```
CREATE TABLE sparse_pk (  
    pcol1 INT NOT NULL,  
    col2 VARCHAR(200),  
    pcol3 INT NOT NULL,  
    PRIMARY KEY (pcol1, pcol3)  
);  
using Sap.MobiLink.Script;  
using System;  
using System.IO;  
using System.Data;  
using System.Text;  
...  
// The method used for the handle_UploadData event.  
public void HandleUpload(UploadData ut) {  
    // Get an UploadedTableData for the sparse_pk table.  
    UploadedTableData sparse_pk_table = ut.GetUploadedTableByName("sparse_pk");  
    // Get deletes uploaded by the MobiLink client.  
    using( IDataReader data_reader = sparse_pk_table.GetDeletes() ) {  
        while (data_reader.Read()) {  
            StringBuilder row_str = new StringBuilder("( ");  
            row_str.Append(data_reader.GetString(0)); // pcol1  
            row_str.Append(", ");  
            row_str.Append(data_reader.GetString(1)); // pcol3  
            row_str.Append(" )");  
            writer.WriteLine(row_str);  
        }  
    }  
}
```

1.20.2 GetInserts() メソッド

このアップロードテーブルデータに対する挿入を保持した DataReader を取得します。

構文

Visual Basic

```
Public Function GetInserts () As IDataReader
```

C#

```
public IDataReader GetInserts ()
```

戻り値

このテーブルデータの挿入を保持した DataReader

備考

各挿入は結果セットの1つのローで表されています。

例

```
CREATE TABLE sparse_pk (
  pcol1 INT NOT NULL,
  col2 VARCHAR(200),
  pcol3 INT NOT NULL,
  PRIMARY KEY (pcol1, pcol3)
);
using Sap.MobiLink.Script;
using System;
using System.IO;
using System.Data;
using System.Text;
...
// The method used for the handle_UploadData event.
public void HandleUpload(UploadData ut) {
  // Get an UploadedTableData for the sparse_pk table.
  UploadedTableData sparse_pk_table = ut.GetUploadedTableByName("sparse_pk");
  // Get inserts uploaded by the MobiLink client.
  using( IDataReader data_reader = sparse_pk_table.GetInserts() ) {
    while (data_reader.Read()) {
      StringBuilder row_str = new StringBuilder(" ");
      row_str.Append(data_reader.GetString(0)); // pcol1
      row_str.Append(", ");
      if (data_reader.IsDBNull(1)) {
        row_str.Append("<NULL>");
      }
      else {
        row_str.Append(data_reader.GetString(1)); // col2
      }
    }
  }
}
```

```
        row_str.Append(", ");
        row_str.Append(data_reader.GetString(2)); // pcol3
        row_str.Append(")");
        writer.WriteLine(row_str);
    }
}
```

1.20.3 GetName() メソッド

このインスタンスのテーブル名を取得します。

構文

Visual Basic

```
Public Function GetName () As String
```

C#

```
public string GetName ()
```

戻り値

このインスタンスのテーブル名。

備考

これはユーティリティ関数です。テーブル名には、このインスタンスのスキーマを使用してもアクセスできます。

1.20.4 GetSchemaTable() メソッド

このダウンロードテーブルのメタデータを記述する DataTable を取得します。

構文

Visual Basic

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

C#

```
public DataTable GetSchemaTable ()
```

戻り値

カラムメタデータが記述された DataTable。

備考

DataTable にカラム名情報を含める場合は、カラム名を送信するためのクライアント拡張オプションを指定してください。これは、デフォルトの動作です。

1.20.5 GetUpdates() メソッド

このアップロードテーブルデータに対する更新を保持した DataReader を取得します。

構文

Visual Basic

```
Public Function GetUpdates () As UpdateDataReader
```

C#

```
public UpdateDataReader GetUpdates ()
```

戻り値

このテーブルデータに対する更新を保持した DataReader

備考

結果セット内の各ローは 1 つの更新を表します。結果セットのモードは、新しいカラム値と古いカラム値とに切り替えることができます。

例

次の例に、GetUpdates メソッドの使用方法を示します。

```
CREATE TABLE sparse_pk (  
  pcol1 INT NOT NULL,  
  col2 VARCHAR(200),  
  pcol3 INT NOT NULL,  
  PRIMARY KEY (pcol1, pcol3)
```

```

);
using Sap.MobiLink.Script;
using System;
using System.IO;
using System.Data;
using System.Text;
...
// The method used for the handle_UploadData event.
public void HandleUpload(UploadData ut) {
    // Get an UploadedTableData for the sparse_pk table.
    UploadedTableData sparse_pk_table = ut.GetUploadedTableByName("sparse_pk");
    // Get updates uploaded by the MobiLink client.
    using( UpdatedDataReader data_reader = sparse_pk_table.GetInserts() ) {
        while (data_reader.Read()) {
            data_reader.SetNewRowValues();
            StringBuilder row_str = new StringBuilder("New values ( ");
            row_str.Append(data_reader.GetString(0)); // pcol1
            row_str.Append(", ");
            if (data_reader.IsDBNull(1)) {
                row_str.Append("<NULL>");
            }
            else {
                row_str.Append(data_reader.GetString(1)); // col2
            }
            row_str.Append(", ");
            row_str.Append(data_reader.GetString(2)); // pcol3
            row_str.Append(" )");
            data_reader.SetOldRowValues();
            row_str.Append(" Old Values (");
            row_str.Append(data_reader.GetString(0)); // pcol1
            row_str.Append(", ");
            if (data_reader.IsDBNull(1)) {
                row_str.Append("<NULL>");
            }
            else {
                row_str.Append(data_reader.GetString(1)); // col2
            }
            row_str.Append(", ");
            row_str.Append(data_reader.GetString(2)); // pcol3
            row_str.Append(" )");
            writer.WriteLine(row_str);
        }
    }
}
}

```

1.21 LogCallback(ServerContext, LogMessage) デリゲート

Mobile Link サーバがメッセージを出力したときに呼び出されます。

構文

Visual Basic

```

Public Delegate Sub LogCallback (
    ByVal sc As ServerContext,
    ByVal message As LogMessage
) As delegate void

```

C#

```
public delegate void LogCallback (  
    ServerContext sc,  
    LogMessage message  
);
```

備考

Mobile Link サーバがメッセージを出力したときに呼び出される ServerContext イベントで、このデリゲートの実装を登録できます。

関連情報

[ErrorListener イベント \[89 ページ\]](#)

[InfoListener イベント \[89 ページ\]](#)

[WarningListener イベント \[90 ページ\]](#)

1.22 ShutdownCallback(ServerContext) デリゲート

Mobile Link サーバが停止するときに呼び出されます。

構文

Visual Basic

```
Public Delegate Sub ShutdownCallback (ByVal sc As ServerContext) As delegate  
void
```

C#

```
public delegate void ShutdownCallback (ServerContext sc);
```

備考

Mobile Link サーバの停止時に呼び出される ShutdownListener イベントで、このデリゲートの実装を登録できます。

関連情報

[ShutdownListener イベント \[89 ページ\]](#)

1.23 SQLType 列挙体

可能性のあるすべての ODBC データ型を列挙します。

構文

Visual Basic

```
Public Enum SQLType
```

C#

```
enum SQLType
```

メンバー

メンバー名	説明
SQL_TYPE_NULL	NULL データ型。
SQL_UNKNOWN_TYPE	不定のデータ型。
SQL_CHAR	設定されたサイズの UTF-8 文字配列。 .NET 型 String を持ちます。
SQL_NUMERIC	設定されたサイズと精度の数値。 .NET 型 decimal を持ちます。
SQL_DECIMAL	設定されたサイズと精度の 10 進数字。 .NET 型 decimal を持ちます。
SQL_INTEGER	32 ビット整数値。 .NET 型 Int32 を持ちます。
SQL_SMALLINT	16 ビット整数値。 .NET 型 Int16 を持ちます。
SQL_FLOAT	ODBC ドライバで精度が定義された浮動小数点数。 .NET 型 Double を持ちます。

メンバー名	説明
SQL_REAL	単精度の浮動小数点数。 .NET 型 Single を持ちます。
SQL_DOUBLE	倍精度浮動小数点数。 .NET 型 Double を持ちます。
SQL_DATE	日付。 .NET 型 DateTime を持ちます。
SQL_DATETIME	日付と時刻。 .NET 型 DateTime を持ちます。
SQL_TIME	時刻。 .NET 型 DateTime を持ちます。
SQL_INTERVAL	時間の間隔。 .NET 型 TimeSpan を持ちます。
SQL_TIMESTAMP	タイムスタンプ。 .NET 型 DateTime を持ちます。
SQL_VARCHAR	ユーザ設定の最大長を持ち、NULL で終了する UTF-8 文字列。 .NET 型 String を持ちます。
SQL_TYPE_DATE	日付。 .NET 型 DateTime を持ちます。
SQL_TYPE_TIME	時刻。 .NET 型 DateTime を持ちます。
SQL_TYPE_TIMESTAMP	タイムスタンプ。 .NET 型 DateTime を持ちます。
SQL_DEFAULT	デフォルト型。 型を持ちません。
SQL_ARD_TYPE	ARD オブジェクト。 型を持ちません。
SQL_BIT	シングルビット。 .NET 型 Boolean を持ちます。
SQL_TINYINT	8ビット整数値。 .NET SByte 型を持ちます。
SQL_BIGINT	64ビット整数値。 .NET 型 Int64 を持ちます。

メンバー名	説明
SQL_LONGVARBINARY	ドライバ依存の最大長を持つ可変長バイナリデータ。 .NET 型 byte[] を持ちます。
SQL_VARBINARY	ユーザ指定した最大長を持つ可変長バイナリデータ。 .NET 型 byte[] を持ちます。
SQL_BINARY	固定長のバイナリデータ。 .NET 型 byte[] を持ちます。
SQL_LONGVARCHAR	ドライバ依存の最大長を持ち、NULL で終了する UTF-8 文字列。 .NET 型 String を持ちます。
SQL_GUID	GUID。 .NET 型 Guid を持ちます。
SQL_WCHAR	固定サイズの Unicode 文字配列。 .NET 型 String を持ちます。
SQL_WVARCHAR	ユーザ定義の最大長を持ち、NULL で終了するユニコード文字列。 .NET 型 String を持ちます。
SQL_WLONGVARCHAR	ドライバ依存の最大長を持ち、NULL で終了するユニコード文字列。 .NET 型 String を持ちます。
SQL_SS_TIMESTAMPOFFSET	タイムゾーンオフセットを持つタイムスタンプ。 .NET 型 ap.MobiLink.Script.DateTimeWithTimeZone を持ちます。 Microsoft SQL Server と Oracle データベースに対してのみ使用できます。

備考

各 SQLType は、.NET 型に対応しています。

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著作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、本稼働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](http://go.sap.com/registration/contact.html)

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

本書のいかなる部分も、SAP SE 又は SAP の関連会社の明示的な許可なくして、いかなる形式でも、いかなる目的にも複製又は伝送することはできません。本書に記載された情報は、予告なしに変更されることがあります。SAP SE 及びその頒布業者によって販売される一部のソフトウェア製品には、他のソフトウェアベンダーの専有ソフトウェアコンポーネントが含まれています。製品仕様は、国ごとに変わる場合があります。

これらの文書は、いかなる種類の表明又は保証もなしで、情報提供のみを目的として、SAP SE 又はその関連会社によって提供され、SAP 又はその関連会社は、これら文書に関する誤記脱落等の過失に対する責任を負うものではありません。SAP 又はその関連会社の製品及びサービスに対する唯一の保証は、当該製品及びサービスに伴う明示的な保証がある場合に、これに規定されたものに限られます。本書のいかなる記述も、追加の保証となるものではありません。

本書に記載される SAP 及びその他の SAP の製品やサービス、並びにそれらの個々のロゴは、ドイツ及びその他の国における SAP SE (又は SAP の関連会社) の商標若しくは登録商標です。本書に記載されたその他のすべての製品およびサービス名は、それぞれの企業の商標です。

商標に関する詳細の情報や通知については、<http://www.sap.com/corporate-en/legal/copyright/index.epx> をご覧ください。