

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

Ultra Light - .NET API リファレンス

目次

1	Ultra Light .NET API リファレンス	16
1.1	ULActiveSyncListener インタフェース.....	20
	ActiveSyncInvoked(bool) メソッド.....	20
1.2	ULBulkCopy クラス.....	23
	ULBulkCopy コンストラクタ.....	25
	Close() メソッド.....	29
	Dispose() メソッド.....	29
	WriteToServer メソッド.....	30
	BatchSize プロパティ.....	33
	BulkCopyTimeout プロパティ.....	34
	ColumnMappings プロパティ.....	34
	DestinationTableName プロパティ.....	35
	NotifyAfter プロパティ.....	36
	ULRowsCopied イベント.....	36
1.3	ULBulkCopyColumnMapping クラス.....	37
	ULBulkCopyColumnMapping コンストラクタ.....	39
	DestinationColumn プロパティ.....	43
	DestinationOrdinal プロパティ.....	44
	SourceColumn プロパティ.....	45
	SourceOrdinal プロパティ.....	45
1.4	ULBulkCopyColumnMappingCollection クラス.....	46
	Add メソッド.....	48
	Contains(ULBulkCopyColumnMapping) メソッド.....	54
	CopyTo(ULBulkCopyColumnMapping[], int) メソッド.....	54
	IndexOf(ULBulkCopyColumnMapping) メソッド.....	55
	Remove(ULBulkCopyColumnMapping) メソッド.....	56
	RemoveAt(int) メソッド.....	57
	this[int index] プロパティ.....	57
1.5	ULCommand クラス.....	58
	ULCommand コンストラクタ.....	63
	BeginExecuteNonQuery メソッド.....	67
	BeginExecuteReader メソッド.....	70
	Cancel() メソッド.....	75
	CreateDbParameter() メソッド.....	75

CreateParameter() メソッド	76
Dispose(bool) メソッド	76
EndExecuteNonQuery(IAsyncResult) メソッド	77
EndExecuteReader(IAsyncResult) メソッド	81
ExecuteDbDataReader(CommandBehavior) メソッド	84
ExecuteNonQuery() メソッド	84
ExecuteReader メソッド	85
ExecuteResultSet メソッド	89
ExecuteScalar() メソッド	92
ExecuteTable メソッド	94
Prepare() メソッド	97
CommandText プロパティ	98
CommandTimeout プロパティ	99
CommandType プロパティ	99
Connection プロパティ	100
DbConnection プロパティ	101
DbParameterCollection プロパティ	101
DbTransaction プロパティ	101
DesignTimeVisible プロパティ	102
IndexName プロパティ	102
Parameters プロパティ	103
Plan プロパティ	104
Transaction プロパティ	104
UpdatedRowSource プロパティ	105
1.6 ULCCommandBuilder クラス	105
ULCCommandBuilder コンストラクタ	108
ApplyParameterInfo(DbParameter, DataRow, StatementType, bool) メソッド	110
GetDeleteCommand メソッド	110
GetInsertCommand メソッド	113
GetParameterName メソッド	116
GetParameterPlaceholder(int) メソッド	117
GetUpdateCommand メソッド	117
InitializeCommand(DbCommand) メソッド	120
SetRowUpdatingHandler(DbDataAdapter) メソッド	120
DataAdapter プロパティ	121
1.7 ULConnection クラス	121
ULConnection コンストラクタ	128
BeginDbTransaction(IsolationLevel) メソッド	131
BeginSynchronize メソッド	131

BeginTransaction メソッド	134
CancelGetNotification(string) メソッド	137
CancelSynchronize(IAsyncResult) メソッド	138
ChangeDatabase(string) メソッド	139
ChangeEncryptionKey(string) メソッド	140
ChangePassword(string, string) メソッド	141
Close() メソッド	141
CountUploadRows(string, long) メソッド	142
CreateCommand() メソッド	143
CreateDbCommand() メソッド	144
CreateNotificationQueue(string, string) メソッド	144
DeclareEvent(string) メソッド	145
DestroyNotificationQueue(string) メソッド	146
Dispose(bool) メソッド	147
EndSynchronize(IAsyncResult) メソッド	147
ExecuteTable メソッド	148
GetLastDownloadTime(string) メソッド	154
GetNewUUID() メソッド	155
GetNotification(string, int) メソッド	156
GetNotificationParameter(string, string) メソッド	157
GetSchema メソッド	158
GrantConnectTo(string, string) メソッド	162
Open() メソッド	163
RegisterForEvent(string, string, string, bool) メソッド	164
ResetLastDownloadTime(string) メソッド	165
RevokeConnectFrom(string) メソッド	165
RollbackPartialDownload() メソッド	166
SendNotification(string, string, string) メソッド	167
SetSyncListener(ULSyncProgressListener) メソッド	168
StartSynchronizationDelete() メソッド	169
StopSynchronizationDelete() メソッド	170
Synchronize メソッド	171
TriggerEvent(string, string) メソッド	173
ValidateDatabase メソッド	174
ConnectionString プロパティ	177
ConnectionTimeout プロパティ	178
Database プロパティ	179
DatabaseID プロパティ	179
DataSource プロパティ	180

	GlobalAutoIncrementUsage プロパティ	180
	LastIdentity プロパティ	181
	Schema プロパティ	182
	ServerVersion プロパティ	182
	State プロパティ	183
	SyncParms プロパティ	183
	SyncResult プロパティ	184
	InfoMessage イベント	184
	StateChange イベント	186
1.8	ULConnectionParms クラス	187
	ULConnectionParms() コンストラクタ	190
	ToString() メソッド	190
	AdditionalParms プロパティ	191
	CacheSize プロパティ	192
	ConnectionString プロパティ	193
	DatabaseOnDesktop プロパティ	194
	DatabaseOnDevice プロパティ	194
	EncryptionKey プロパティ	195
	Password プロパティ	195
	UserID プロパティ	196
1.9	ULConnectionStringBuilder クラス	197
	ULConnectionStringBuilder コンストラクタ	201
	ContainsKey(string) メソッド	202
	EquivalentTo(DbConnectionStringBuilder) メソッド	203
	GetShortName(string) メソッド	203
	Remove(string) メソッド	204
	TryGetValue(string, out Object) メソッド	205
	CacheSize プロパティ	206
	ConnectionString プロパティ	206
	DatabaseKey プロパティ	207
	DatabaseName プロパティ	208
	DatabaseOnDesktop プロパティ	208
	DatabaseOnDevice プロパティ	209
	OrderedTableScans プロパティ	209
	Password プロパティ	210
	ReserveSize プロパティ	211
	StartLine プロパティ	212
	this[string keyword] プロパティ	212
	UserID プロパティ	213

1.10	ULCreateParms クラス	214
	ULCreateParms() コンストラクタ	218
	ToString() メソッド	218
	CaseSensitive プロパティ	219
	ChecksumLevel プロパティ	219
	DateFormat プロパティ	220
	DateOrder プロパティ	220
	FIPS プロパティ	221
	MaxHashSize プロパティ	221
	NearestCentury プロパティ	222
	Obfuscate プロパティ	222
	PageSize プロパティ	223
	Precision プロパティ	223
	Scale プロパティ	224
	TimeFormat プロパティ	224
	TimestampFormat プロパティ	225
	TimestampIncrement プロパティ	225
	UTF8Encoding プロパティ	226
1.11	ULCursorSchema クラス	226
	GetColumnCount() メソッド	229
	GetColumnID(string) メソッド	229
	GetColumnName(int) メソッド	230
	GetColumnPrecision(int) メソッド	231
	GetColumnScale(int) メソッド	232
	GetColumnSize(int) メソッド	233
	GetColumnSQLName(int) メソッド	234
	GetColumnULDbType(int) メソッド	235
	GetSchemaTable() メソッド	236
	VerifyOpen() メソッド	236
	ColumnCount プロパティ	237
	IsOpen プロパティ	237
	Name プロパティ	238
1.12	ULDataAdapter クラス	238
	ULDataAdapter コンストラクタ	241
	Dispose(bool) メソッド	245
	Fill メソッド	246
	FillSchema メソッド	247
	GetFillParameters() メソッド	249
	Update(DataRow[], DataTableMapping) メソッド	249

	DeleteCommand プロパティ	250
	InsertCommand プロパティ	251
	SelectCommand プロパティ	251
	TableMappings プロパティ	252
	UpdateCommand プロパティ	253
	RowUpdated イベント	254
	RowUpdating イベント	254
1.13	ULDatabaseManager クラス	255
	CreateDatabase(string, string) メソッド	257
	DropDatabase(string) メソッド	258
	SetActiveSyncListener(string, ULActiveSyncListener) メソッド	260
	SetServerSyncListener(string, string, ULServerSyncListener) メソッド	261
	SignalSyncIsComplete() メソッド	262
	ValidateDatabase(string, ULDBValid) メソッド	263
	RuntimeType プロパティ	264
1.14	ULDatabaseSchema クラス	265
	GetDatabaseProperty(string) メソッド	267
	GetPublicationName(int) メソッド	269
	GetTableName(int) メソッド	270
	SetDatabaseOption(string, string) メソッド	271
	IsCaseSensitive プロパティ	272
	IsOpen プロパティ	273
	PublicationCount プロパティ	274
	TableCount プロパティ	274
1.15	ULDataReader クラス	275
	Close() メソッド	282
	Dispose(bool) メソッド	282
	GetBoolean(int) メソッド	283
	GetByte(int) メソッド	284
	GetBytes メソッド	285
	GetChar(int) メソッド	288
	GetChars(int, long, char[], int, int) メソッド	289
	GetDataTypeName(int) メソッド	291
	GetDateTime(int) メソッド	292
	GetDbDataReader(int) メソッド	293
	GetDecimal(int) メソッド	293
	GetDouble(int) メソッド	294
	GetEnumerator() メソッド	295
	GetFieldType(int) メソッド	295

GetFloat(int) メソッド	296
GetGuid(int) メソッド	297
GetInt16(int) メソッド	298
GetInt32(int) メソッド	299
GetInt64(int) メソッド	300
GetName(int) メソッド	301
GetOrdinal(string) メソッド	302
GetRowCount(int) メソッド	303
GetSchemaTable() メソッド	304
GetString(int) メソッド	306
GetTimeSpan(int) メソッド	307
GetUInt16(int) メソッド	308
GetUInt32(int) メソッド	309
GetUInt64(int) メソッド	310
GetValue(int) メソッド	311
GetValues(object[]) メソッド	312
IsDBNull(int) メソッド	313
MoveAfterLast() メソッド	314
MoveBeforeFirst() メソッド	314
MoveFirst() メソッド	315
MoveLast() メソッド	315
MoveNext() メソッド	316
MovePrevious() メソッド	317
MoveRelative(int) メソッド	317
NextResult() メソッド	318
Read() メソッド	319
Validate メソッド	320
Depth プロパティ	321
FieldCount プロパティ	321
HasRows プロパティ	322
IsBOF プロパティ	322
IsClosed プロパティ	323
IsEOF プロパティ	323
RecordsAffected プロパティ	324
RowCount プロパティ	324
Schema プロパティ	325
this プロパティ	326
1.16 ULException クラス	328
NativeError プロパティ	329

	Source プロパティ	329
1.17	ULFactory クラス	330
	CreateCommand() メソッド	333
	CreateCommandBuilder() メソッド	333
	CreateConnection() メソッド	334
	CreateConnectionStringBuilder() メソッド	334
	CreateDataAdapter() メソッド	335
	CreateParameter() メソッド	336
	CanCreateDataSourceEnumerator プロパティ	336
1.18	ULFileTransfer クラス	337
	ULFileTransfer() コンストラクタ	340
	DownloadFile メソッド	341
	UploadFile メソッド	344
	AuthenticationParms プロパティ	347
	AuthStatus プロパティ	348
	AuthValue プロパティ	348
	FileAuthCode プロパティ	349
	FileName プロパティ	349
	LocalFileName プロパティ	350
	LocalPath プロパティ	351
	Password プロパティ	351
	RemoteKey プロパティ	352
	ResumePartialDownload プロパティ	353
	Stream プロパティ	353
	StreamErrorCode プロパティ	354
	StreamErrorSystem プロパティ	355
	StreamParms プロパティ	355
	TransferredFile プロパティ	356
	UserName プロパティ	357
	Version プロパティ	357
1.19	ULFileTransferProgressData クラス	358
	BytesReceived プロパティ	359
	FileSize プロパティ	360
	Flags プロパティ	360
	ResumedAtSize プロパティ	361
1.20	ULFileTransferProgressListener インタフェース	361
	FileTransferProgressed(ULFileTransferProgressData) メソッド	362
1.21	ULIndexSchema クラス	363
	Close() メソッド	365

	GetColumnName(short) メソッド	365
	IsColumnDescending(string) メソッド	366
	ColumnCount プロパティ	367
	IsForeignKey プロパティ	368
	IsForeignKeyCheckOnCommit プロパティ	368
	IsForeignKeyNullable プロパティ	369
	IsOpen プロパティ	370
	IsPrimaryKey プロパティ	370
	IsUniqueIndex プロパティ	371
	IsUniqueKey プロパティ	371
	Name プロパティ	372
	ReferencedIndexName プロパティ	372
	ReferencedTableName プロパティ	373
1.22	ULInfoMessageEventArgs クラス	373
	ToString() メソッド	374
	Message プロパティ	375
	NativeError プロパティ	376
	Source プロパティ	376
1.23	ULMetaDataCollectionNames クラス	377
	Columns プロパティ	379
	DataSourceInformation プロパティ	380
	DataTypes プロパティ	381
	ForeignKeys プロパティ	382
	IndexColumns プロパティ	383
	Indexes プロパティ	384
	MetaDataCollections プロパティ	385
	Publications プロパティ	386
	ReservedWords プロパティ	387
	Restrictions プロパティ	388
	Tables プロパティ	389
1.24	ULParameter クラス	389
	ULParameter コンストラクタ	393
	ResetDbType() メソッド	400
	ToString() メソッド	401
	DbType プロパティ	401
	Direction プロパティ	402
	IsNullable プロパティ	403
	Offset プロパティ	403
	ParameterName プロパティ	404

	Precision プロパティ	405
	Scale プロパティ	405
	Size プロパティ	406
	SourceColumn プロパティ	407
	SourceColumnNullMapping プロパティ	407
	SourceVersion プロパティ	408
	ULDbType プロパティ	408
	Value プロパティ	409
1.25	ULParameterCollection クラス	410
	Add メソッド	413
	AddRange メソッド	421
	Clear() メソッド	423
	Contains メソッド	423
	CopyTo(Array, int) メソッド	425
	GetEnumerator() メソッド	426
	GetParameter メソッド	427
	IndexOf メソッド	428
	Insert(int, object) メソッド	430
	Remove(object) メソッド	431
	RemoveAt メソッド	432
	SetParameter メソッド	434
	Count プロパティ	435
	IsFixedSize プロパティ	435
	IsReadOnly プロパティ	436
	IsSynchronized プロパティ	436
	SyncRoot プロパティ	437
	this プロパティ	437
1.26	ULResultSet クラス	439
	AppendBytes(int, byte[], int, int) メソッド	446
	AppendChars(int, char[], int, int) メソッド	447
	Delete() メソッド	449
	SetBoolean(int, bool) メソッド	449
	SetByte(int, byte) メソッド	450
	SetBytes(int, byte[]) メソッド	451
	SetDateTime(int, DateTime) メソッド	453
	SetDBNull(int) メソッド	454
	SetDecimal(int, decimal) メソッド	455
	SetDouble(int, double) メソッド	456
	SetFloat(int, float) メソッド	457

	SetGuid(int, Guid) メソッド	458
	SetInt16(int, short) メソッド	459
	SetInt32(int, int) メソッド	461
	SetInt64(int, long) メソッド	462
	SetString(int, string) メソッド	463
	SetTimeSpan(int, TimeSpan) メソッド	464
	SetToDefault(int) メソッド	465
	SetUInt16(int, ushort) メソッド	466
	SetUInt32(int, uint) メソッド	467
	SetUInt64(int, ulong) メソッド	468
	Update() メソッド	470
	UpdateBegin() メソッド	470
1.27	ULResultSetSchema クラス	471
	VerifyOpen() メソッド	473
	Name プロパティ	473
1.28	ULRowsCopiedEventArgs クラス	474
	ULRowsCopiedEventArgs(long) コンストラクタ	475
	Abort プロパティ	475
	RowsCopied プロパティ	476
1.29	ULRowUpdatedEventArgs クラス	476
	ULRowUpdatedEventArgs(DataRow, IDbCommand, StatementType, DataTableMapping) コンストラクタ	478
	Command プロパティ	478
	RecordsAffected プロパティ	479
1.30	ULRowUpdatingEventArgs クラス	479
	ULRowUpdatingEventArgs(DataRow, IDbCommand, StatementType, DataTableMapping) コンストラクタ	481
	Command プロパティ	481
1.31	ULServerSyncListener インタフェース	482
	ServerSyncInvoked(string) メソッド	483
1.32	ULSqlProgressData クラス (廃止予定)	484
	CurrentScript プロパティ	485
	ScriptCount プロパティ	486
	State プロパティ	486
1.33	ULSyncParms クラス	487
	CopyFrom(ULSyncParms) メソッド	490
	ToString() メソッド	490
	AdditionalParms プロパティ	491
	AuthenticationParms プロパティ	492

	DownloadOnly プロパティ	492
	KeepPartialDownload プロパティ	493
	NewPassword プロパティ	494
	Password プロパティ	495
	PingOnly プロパティ	496
	Publications プロパティ	497
	ResumePartialDownload プロパティ	497
	SendDownloadAck プロパティ	498
	Stream プロパティ	499
	StreamParms プロパティ	499
	UploadOnly プロパティ	500
	UserName プロパティ	501
	Version プロパティ	502
1.34	ULSyncProgressData クラス	502
	CurrentDownloadRowCount プロパティ	505
	Flags プロパティ	506
	IgnoredDeletes プロパティ	507
	IgnoredUpdates プロパティ	507
	IsFinalSyncProgress プロパティ	508
	ReceivedBytes プロパティ	508
	ReceivedDeletes プロパティ	509
	ReceivedInserts プロパティ	510
	ReceivedUpdates プロパティ	510
	SentBytes プロパティ	511
	SentDeletes プロパティ	512
	SentInserts プロパティ	512
	SentUpdates プロパティ	513
	State プロパティ	513
	SyncTableCount プロパティ	514
	SyncTableIndex プロパティ	515
	TableID プロパティ	515
	TableName プロパティ	516
	TotalDownloadRowCount プロパティ	516
	TruncateDeletes プロパティ	517
1.35	ULSyncProgressListener インタフェース	518
	SyncProgressed(ULSyncProgressData) メソッド	518
1.36	ULSyncResult クラス	519
	AuthStatus プロパティ	521
	AuthValue プロパティ	522

	IgnoredRows プロパティ	522
	PartialDownloadRetained プロパティ	523
	StreamErrorCode プロパティ	523
	StreamErrorParameters プロパティ	524
	StreamErrorSystem プロパティ	524
	Timestamp プロパティ	525
	UploadOK プロパティ	525
1.37	ULTable クラス	526
	DeleteAllRows() メソッド	532
	FindBegin() メソッド	533
	FindFirst メソッド	534
	FindLast メソッド	537
	FindNext メソッド	540
	FindPrevious メソッド	543
	Insert() メソッド	545
	InsertBegin() メソッド	546
	LookupBackward メソッド	547
	LookupBegin() メソッド	550
	LookupForward メソッド	551
	Truncate() メソッド	554
	Schema プロパティ	554
1.38	ULTableSchema クラス	555
	GetColumnDefaultValue(int) メソッド	559
	GetColumnPartitionSize(int) メソッド	560
	GetIndex(string) メソッド	561
	GetIndexName(int) メソッド	562
	GetOptimalIndex(int) メソッド	563
	GetPublicationPredicate(string) メソッド	564
	IsColumnAutoIncrement(int) メソッド	564
	IsColumnCurrentDate(int) メソッド	565
	IsColumnCurrentTime(int) メソッド	566
	IsColumnCurrentTimestamp(int) メソッド	567
	IsColumnCurrentUTCTimestamp(int) メソッド	568
	IsColumnGlobalAutoIncrement(int) メソッド	569
	IsColumnNewUUID(int) メソッド	570
	IsColumnNullable(int) メソッド	571
	IsInPublication(string) メソッド	572
	VerifyOpen() メソッド	572
	IndexCount プロパティ	573

	IsNeverSynchronized プロパティ	573
	Name プロパティ	574
	PrimaryKey プロパティ	574
	UploadUnchangedRows プロパティ	575
1.39	ULTransaction クラス	575
	Commit() メソッド	577
	Dispose(bool) メソッド	577
	Rollback() メソッド	578
	Connection プロパティ	578
	DbConnection プロパティ	579
	IsolationLevel プロパティ	579
1.40	ULInfoMessageEventHandler(object, ULInfoMessageEventArgs) デリゲート	580
1.41	ULRowsCopiedEventHandler(object, ULRowsCopiedEventArgs) デリゲート	581
1.42	ULRowUpdatedEventHandler(object, ULRowUpdatedEventArgs) デリゲート	581
1.43	ULRowUpdatingEventHandler(object, ULRowUpdatingEventArgs) デリゲート	582
1.44	ULSyncProgressedDlg(IAsyncResult, ULSyncProgressData) デリゲート	583
1.45	ULAuthStatusCode 列挙体	584
1.46	ULBulkCopyOptions 列挙体	585
1.47	ULDateOrder 列挙体	586
1.48	ULDbType 列挙体	586
1.49	ULDBValid 列挙体	590
1.50	ULRuntimeType 列挙体	590
1.51	ULSqlProgressState 列挙体	591
1.52	ULStreamType 列挙体	592
1.53	ULSyncProgressState 列挙体	593
2	このマニュアルの印刷、再生、および再配布	597

1 Ultra Light .NET API リファレンス

.NET Framework 2.0 および .NET Compact Framework 2.0 用 Ultra Light.NET データプロバイダの API を使用して、Ultra Light の .NET アプリケーションを作成することができます。

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

Ultra Light.NET の Ultra Light エンジンのランタイムを使用するには、RuntimeType プロパティを適切な値に設定してから、他の Ultra Light.NET API を使用します。

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

- CultureInfo.CurrentUICulture
- CultureInfo.CurrentCulture
- EN

Sap.Data.UltraLite ネームスペースでよく使用される高レベルクラスのいくつかについて次のリストにまとめて示します。

ULConnection

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

ULTable

各 ULTable オブジェクトを使用して、単一テーブル内のデータにアクセスできます。

ULCommand オブジェクト

各 ULCommand オブジェクトには、データベースに対して実行される SQL 文が格納されます。

ULDataReader オブジェクト

各 ULDataReader オブジェクトには、単一のクエリの結果セットが格納されます。

ULSyncParms

ULSyncParms オブジェクトを使用して、Ultra Light データベースを Mobile Link サーバと同期させます。

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

ネームスペース

```
Sap.Data.UltraLite
```


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:

[ULActiveSyncListener インタフェース \[20 ページ\]](#)

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

[ULBulkCopy クラス \[23 ページ\]](#)

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

[ULBulkCopyColumnMapping クラス \[37 ページ\]](#)

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

[ULBulkCopyColumnMappingCollection クラス \[46 ページ\]](#)

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

[ULCommand クラス \[58 ページ\]](#)

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

[ULCommandBuilder クラス \[105 ページ\]](#)

System.Data.DataSet の変更内容を、関連するデータベース内のデータに一致させる単一テーブルのコマンドを、自動的に生成します。

[ULConnection クラス \[121 ページ\]](#)

Ultra Light.NET データベースへの接続を表します。

[ULConnectionParms クラス \[187 ページ\]](#)

UL 拡張: Ultra Light データベースへの接続を開く接続文字列を作成します。

[ULConnectionStringBuilder クラス \[197 ページ\]](#)

Ultra Light データベースへの接続を開く接続文字列を作成します。

[ULCreateParms クラス \[214 ページ\]](#)

UL 拡張: Ultra Light データベースを作成するときの作成時オプションの文字列を作成します。

[ULCursorSchema クラス \[226 ページ\]](#)

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

[ULDataAdapter クラス \[238 ページ\]](#)

System.Data.DataSet に入力したりデータベースを更新したりするために使用する一連のコマンドとデータベース接続を表します。

[ULDatabaseManager クラス \[255 ページ\]](#)

UL 拡張: データベースの作成、削除、検証を行う静的メソッドを提供します。

[ULDatabaseSchema クラス \[265 ページ\]](#)

UL 拡張: Ultra Light.NET データベースのスキーマを表します。

[ULDataReader クラス \[275 ページ\]](#)

Ultra Light データベースの読み込み専用の双方向カーソルを表します。

[ULException クラス \[328 ページ\]](#)

Ultra Light.NET データベースによって返される SQL エラーを表します。

[ULFactory クラス \[330 ページ\]](#)

データソースクラスの Sap.Data.UltraLite プロバイダの実装のインスタンスを作成する、メソッドのセットを表します。

[ULFileTransfer クラス \[337 ページ\]](#)

UL 拡張: Mobile Link サーバを使用して、リモートデータベースからファイルを転送します。

[ULFileTransferProgressData クラス \[358 ページ\]](#)

UL 拡張: ファイル転送の進行状況のモニタリングデータを返します。

[ULFileTransferProgressListener インタフェース \[361 ページ\]](#)

UL 拡張: ファイル転送プログレスイベントを受信するリスナインタフェースです。

[ULIndexSchema クラス \[363 ページ\]](#)

UL 拡張: Ultra Light テーブルのインデックスのスキーマを表します。

[ULInfoMessageEventArgs クラス \[373 ページ\]](#)

ULConnection.InfoMessage イベントのデータを提供します。

[ULMetaDataCollectionNames クラス \[377 ページ\]](#)

メタデータコレクションを取得する ULConnection.GetSchema(String,String[]) メソッドで使用する定数のリストを提供します。

[ULParameter クラス \[389 ページ\]](#)

ULCommand オブジェクトのパラメータを表します。

[ULParameterCollection クラス \[410 ページ\]](#)

ULCommand オブジェクトのすべてのパラメータを示します。

[ULResultSet クラス \[439 ページ\]](#)

UL 拡張: Ultra Light データベースの編集可能な結果セットを表します。

[ULResultSetSchema クラス \[471 ページ\]](#)

UL 拡張: Ultra Light の結果セットのスキーマを表します。

[ULRowsCopiedEventArgs クラス \[474 ページ\]](#)

ULRowsCopiedEventHandler オブジェクトに渡される引数のセットを表します。

[ULRowUpdatedEventArgs クラス \[476 ページ\]](#)

ULDataAdapter.RowUpdated イベントのデータを提供します。

[ULRowUpdatingEventArgs クラス \[479 ページ\]](#)

ULDataAdapter.RowUpdating イベントのデータを提供します。

[ULServerSyncListener インタフェース \[482 ページ\]](#)

UL 拡張: サーバの同期メッセージを受信するリスナインタフェースです。

[ULSqlProgressData クラス \(廃止予定\) \[484 ページ\]](#)

UL 拡張: SQL パススルースクリプトの進行状況のモニタリングデータを返します。

[ULSyncParms クラス \[487 ページ\]](#)

UL 拡張: Ultra Light データベースの同期方法を定義する同期パラメータを表します。

[ULSyncProgressData クラス \[502 ページ\]](#)

UL 拡張: 同期の進行状況のモニタリングデータを返します。

[ULSyncProgressListener インタフェース \[518 ページ\]](#)

UL 拡張: 同期プログ्रेसイベントを受信するリスナーインタフェースです。

[ULSyncResult クラス \[519 ページ\]](#)

UL 拡張: 前回の同期のステータスを表します。

[ULTable クラス \[526 ページ\]](#)

UL 拡張: Ultra Light データベース内のテーブルを表します。

[ULTableSchema クラス \[555 ページ\]](#)

UL 拡張: Ultra Light のテーブルのスキーマを表します。

[ULTransaction クラス \[575 ページ\]](#)

SQL トランザクションを表します。

[ULInfoMessageEventHandler\(object, ULInfoMessageEventArgs\) デリゲート \[580 ページ\]](#)

ULConnection.InfoMessage イベントを処理するメソッドを示します。

[ULRowsCopiedEventHandler\(object, ULRowsCopiedEventArgs\) デリゲート \[581 ページ\]](#)

ULBulkCopy.ULRowsCopied イベントを処理するメソッドを示します。

[ULRowUpdatedEventHandler\(object, ULRowUpdatedEventArgs\) デリゲート \[581 ページ\]](#)

ULDataAdapter.RowUpdated イベントを処理するメソッドを示します。

[ULRowUpdatingEventHandler\(object, ULRowUpdatingEventArgs\) デリゲート \[582 ページ\]](#)

ULDataAdapter.RowUpdating イベントを処理するメソッドを示します。

[ULSyncProgressedDlg\(IAsyncResult, ULSyncProgressData\) デリゲート \[583 ページ\]](#)

同期進捗情報と同期中に起動されたメソッドを表します。

[ULAuthStatusCode 列挙体 \[584 ページ\]](#)

UL 拡張: Mobile Link ユーザ認証の実行中にレポートされる可能性のあるステータスコードを列挙します。

[ULBulkCopyOptions 列挙体 \[585 ページ\]](#)

ULBulkCopy クラスのインスタンスで使用する、1 つ以上のオプションを指定するビット単位フラグです。

[ULDateOrder 列挙体 \[586 ページ\]](#)

UL 拡張: データベースがサポートできる日付順を列挙します。

[ULDbType 列挙体 \[586 ページ\]](#)

Ultra Light.NET データベースのデータ型を列挙します。

[ULDBValid 列挙体 \[590 ページ\]](#)

Ultra Light.NET データベース検証メソッドを列挙します。

[ULRuntimeType 列挙体 \[590 ページ\]](#)

UL 拡張: Ultra Light.NET ランタイムのタイプを列挙します。

[ULSqlProgressState 列挙体 \[591 ページ\]](#)

UL 拡張: SQL パススルースクリプトの実行中に発生する可能性のあるすべてのステータスを列挙します。

[ULStreamType 列挙体 \[592 ページ\]](#)

UL 拡張: 同期に使用する Mobile Link 同期ストリームのタイプを列挙します。

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

UL 拡張: 同期中に発生する可能性のあるすべてのステータスを列挙します。

1.1 ULActiveSyncListener インタフェース

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

構文

Visual Basic

```
Public Interface ULActiveSyncListener
```

C#

```
public interface ULActiveSyncListener
```

メンバー

ULActiveSyncListener 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 void	ActiveSyncInvoked(bool) [20 ページ]	ActiveSync 用の Mobile Link プロバイダがアプリケーションを呼び出して同期を実行するときに起動されます。

このセクションの内容:

[ActiveSyncInvoked\(bool\) メソッド \[20 ページ\]](#)

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

1.1.1 ActiveSyncInvoked(bool) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

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

備考

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

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

例

次のコードフラグメントは、ActiveSync 要求の受信方法と UI スレッドでの同期の実行方法を示しています。

```
' Visual Basic
Imports Sap.Data.UltraLite
Public Class MainWindow
    Inherits System.Windows.Forms.Form
    Implements ULActiveSyncListener
    Private conn As ULConnection
    Public Sub New(ByVal args() As String)
        MyBase.New()
        ' This call is required by the Windows Form Designer.
        InitializeComponent()
        ' Add any initialization after the InitializeComponent call.
        ULConnection.DatabaseManager.SetActiveSyncListener( _
            "myCompany.myapp", Me _
        )
        ' Create Connection
        ...
    End Sub
    Protected Overrides Sub OnClosing( _
        ByVal e As System.ComponentModel.CancelEventArgs _
    )
        ULConnection.DatabaseManager.SetActiveSyncListener( _
            Nothing, Nothing _
        )
        MyBase.OnClosing(e)
    End Sub
    Public Sub ActiveSyncInvoked( _
        ByVal launchedByProvider As Boolean _
    ) Implements ULActiveSyncListener.ActiveSyncInvoked
        Me.Invoke(New EventHandler(AddressOf Me.ActiveSyncAction))
    End Sub
```

```

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

```

対応する C# 言語のコードを次に示します。

```

// C#
using Sap.Data.UltraLite;
public class Form1 : System.Windows.Forms.Form, ULActiveSyncListener
{
    private System.Windows.Forms.MainMenu mainMenu1;
    private ULConnection conn;
    public Form1()
    {
        //
        // Required for Windows Form Designer support.
        // InitializeComponent();
        //
        // TODO: Add any constructor code after the
        // InitializeComponent call.
        //
        ULDatabaseManager.SetActiveSyncListener(
            "myCompany.myapp", this
        );
        // Create connection
        ...
    }
    protected override void Dispose( bool disposing )
    {
        base.Dispose( disposing );
    }
    protected override void OnClosing(
        System.ComponentModel.CancelEventArgs e )
    {
        ULDatabaseManager.SetActiveSyncListener(null, null);
        base.OnClosing(e);
    }
    public void ActiveSyncInvoked(bool launchedByProvider)
    {
        this.Invoke( new EventHandler( ActiveSyncHandler ) );
    }
    internal void ActiveSyncHandler(object sender, EventArgs e)
    {
        conn.Synchronize();
        ULDatabaseManager.SignalSyncIsComplete();
    }
}

```

関連情報

[SignalSyncIsComplete\(\) メソッド \[262 ページ\]](#)

1.2 ULBulkCopy クラス

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

構文

Visual Basic

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

C#

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

メンバー

ULBulkCopy 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	ULBulkCopy [25 ページ]	ULBulkCopy オブジェクトを、指定された ULConnection オブジェクトで初期化します。

メソッド

変更子とタイプ	メソッド	説明
public void	Close() [29 ページ]	ULBulkCopy オブジェクトを閉じます。
public void	Dispose() [29 ページ]	ULBulkCopy オブジェクトを破棄します。
public void	WriteToServer [30 ページ]	指定された System.Data.DataRow オブジェクトの配列内のすべてのローを、ULBulkCopy オブジェクトの DestinationTableName フィールドで指定される送信先テーブルにコピーします。

プロパティ

変更子とタイプ	プロパティ	説明
public int	BatchSize [33 ページ]	各バッチの中のローの数を取得または設定します。
public int	BulkCopyTimeout [34 ページ]	タイムアウトするまでに完了するオペレーションの秒数を取得または設定します。
public ULBulkCopyColumnMappingCollection	ColumnMappings [34 ページ]	ULBulkCopyColumnMapping 項目のコレクションを返します。

変更子とタイプ	プロパティ	説明
public string	DestinationTableName [35 ページ]	サーバ上の送信先テーブルの名前を取得または設定します。
public int	NotifyAfter [36 ページ]	通知イベントが生成されるまでに処理されるローの数を指定します。

イベント

変更子とタイプ	イベント	説明
public ULRowsCopiedEventHandler	ULRowsCopied [36 ページ]	NotifyAfter プロパティで指定される数のローが処理されるたびに、このイベントが発生します。

備考

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

このセクションの内容:

[ULBulkCopy コンストラクタ \[25 ページ\]](#)

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

[Close\(\) メソッド \[29 ページ\]](#)

ULBulkCopy オブジェクトを閉じます。

[Dispose\(\) メソッド \[29 ページ\]](#)

ULBulkCopy オブジェクトを破棄します。

[WriteToServer メソッド \[30 ページ\]](#)

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

[BatchSize プロパティ \[33 ページ\]](#)

各バッチの中のローの数を取得または設定します。

[BulkCopyTimeout プロパティ \[34 ページ\]](#)

タイムアウトするまでに完了するオペレーションの秒数を取得または設定します。

[ColumnMappings プロパティ \[34 ページ\]](#)

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

[DestinationTableName プロパティ \[35 ページ\]](#)

サーバ上の送信先テーブルの名前を取得または設定します。

[NotifyAfter プロパティ \[36 ページ\]](#)

通知イベントが生成されるまでに処理されるローの数を指定します。

[ULRowsCopied イベント \[36 ページ\]](#)

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

1.2.1 ULBulkCopy コンストラクタ

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

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	ULBulkCopy(ULConnection) [25 ページ]	ULBulkCopy オブジェクトを、指定された ULConnection オブジェクトで初期化します。
public	ULBulkCopy(ULConnection, ULBulkCopyOptions, ULTransaction) [26 ページ]	ULBulkCopy オブジェクトを、指定された ULConnection オブジェクト、コピーオプション、ULTransaction オブジェクトで初期化します。
public	ULBulkCopy(string) [27 ページ]	ULBulkCopy オブジェクトを、指定された接続文字列で初期化します。
public	ULBulkCopy(string, ULBulkCopyOptions) [28 ページ]	ULBulkCopy オブジェクトを、指定された接続文字列とコピーオプションで初期化します。

このセクションの内容:

[ULBulkCopy\(ULConnection\) コンストラクタ \[25 ページ\]](#)

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

[ULBulkCopy\(ULConnection, ULBulkCopyOptions, ULTransaction\) コンストラクタ \[26 ページ\]](#)

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

[ULBulkCopy\(string\) コンストラクタ \[27 ページ\]](#)

ULBulkCopy オブジェクトを、指定された接続文字列で初期化します。

[ULBulkCopy\(string, ULBulkCopyOptions\) コンストラクタ \[28 ページ\]](#)

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

1.2.1.1 ULBulkCopy(ULConnection) コンストラクタ

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

構文

Visual Basic

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

C#

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

パラメータ

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

備考

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

関連情報

[ULConnection クラス \[121 ページ\]](#)

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

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

構文

Visual Basic

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

C#

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

パラメータ

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

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

externalTransaction バルクコピーが発生する、既存の ULTransaction オブジェクト。この値が NULL 参照 (Visual Basic の Nothing) でない場合は、バルクコピー操作が内部で実行されます。外部トランザクションと ULBulkCopyOptions.UseInternalTransaction オプションの両方を指定すると、エラーになります。

備考

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

関連情報

[ULConnection クラス \[121 ページ\]](#)

[ULTransaction クラス \[575 ページ\]](#)

1.2.1.3 ULBulkCopy(string) コンストラクタ

ULBulkCopy オブジェクトを、指定された接続文字列で初期化します。

構文

Visual Basic

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

C#

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

パラメータ

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

備考

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

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

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

関連情報

[ConnectionString プロパティ \[177 ページ\]](#)

[ULConnectionParms クラス \[187 ページ\]](#)

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

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

構文

Visual Basic

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

C#

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

パラメータ

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

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

備考

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

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

関連情報

[ConnectionString プロパティ \[177 ページ\]](#)

[ULBulkCopyOptions 列挙体 \[585 ページ\]](#)

1.2.2 Close() メソッド

ULBulkCopy オブジェクトを閉じます。

構文

Visual Basic

```
Public Sub Close ()
```

C#

```
public void Close ()
```

1.2.3 Dispose() メソッド

ULBulkCopy オブジェクトを破棄します。

構文

Visual Basic

```
Public Sub Dispose ()
```

C#

```
public void Dispose ()
```

1.2.4 WriteToServer メソッド

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

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public void	WriteToServer(DataRow[]) [31 ページ]	指定された System.Data.DataRow オブジェクトの配列内のすべてのローを、ULBulkCopy オブジェクトの DestinationTableName フィールドで指定される送信先テーブルにコピーします。
public void	WriteToServer(DataTable) [31 ページ]	指定された System.Data.DataTable のすべてのローを、ULBulkCopy.DestinationTableName プロパティで指定される送信先テーブルにコピーします。
public void	WriteToServer(DataTable, DataRowState) [32 ページ]	指定されたローの状態を持つ指定された System.Data.DataTable のすべてのローを、ULBulkCopy.DestinationTableName プロパティで指定される送信先テーブルにコピーします。
public void	WriteToServer(IDataReader) [33 ページ]	指定された System.Data.DataReader のすべてのローを、ULBulkCopy.DestinationTableName プロパティで指定される送信先テーブルにコピーします。

このセクションの内容:

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

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

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

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

[WriteToServer\(DataTable, DataRowState\) メソッド \[32 ページ\]](#)

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

[WriteToServer\(IDataReader\) メソッド \[33 ページ\]](#)

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

1.2.4.1 WriteToServer(DataRow[]) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

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

関連情報

[DestinationTableName プロパティ \[35 ページ\]](#)

1.2.4.2 WriteToServer(DataTable) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

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

関連情報

[DestinationTableName プロパティ \[35 ページ\]](#)

1.2.4.3 WriteToServer(DataTable, DataRowState) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

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

rowState System.Data.DataRowState 列挙体の値。ローステータスに一致するローのみ、送信先にコピーされます。

備考

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

関連情報

[DestinationTableName プロパティ \[35 ページ\]](#)

1.2.4.4 WriteToServer(IDataReader) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

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

関連情報

[DestinationTableName プロパティ \[35 ページ\]](#)

1.2.5 BatchSize プロパティ

各バッチの中のローの数を取得または設定します。

構文

Visual Basic

```
Public Property BatchSize As Integer
```

C#

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

備考

各バッチが終了すると、バッチ内のローがサーバに送信されます。

各バッチの中のロー数。デフォルトは 0 です。

0 に設定すると、すべてのローが 1 つのバッチで送信されます。

0 未満の値を設定すると、エラーになります。

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

1.2.6 BulkCopyTimeout プロパティ

タイムアウトするまでに完了するオペレーションの秒数を取得または設定します。

構文

Visual Basic

```
Public Property BulkCopyTimeout As Integer
```

C#

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

備考

デフォルト値は 30 秒です。

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

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

ゼロ未満の値を設定すると、エラーがスローされます。

1.2.7 ColumnMappings プロパティ

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

構文

Visual Basic

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

C#

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

備考

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

デフォルトでは、空のコレクションです。

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

WriteToServer メソッドの実行時に ColumnMappings オブジェクトが空の場合は、ソース内の先頭のカラムが送信先の先頭のカラムにマッピングされ、2 番目は 2 番目にマッピングされます。以降についても同様です。この処理は、カラムの型が変換可能な場合、ソースカラム以上の送信先カラムがある場合、余分な送信先カラムが NULL 入力可のカラムである場合に行われます。

関連情報

[ULBulkCopyColumnMapping クラス \[37 ページ\]](#)

1.2.8 DestinationTableName プロパティ

サーバ上の送信先テーブルの名前を取得または設定します。

構文

Visual Basic

```
Public Property DestinationTableName As String
```

C#

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

備考

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

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

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

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

1.2.9 NotifyAfter プロパティ

通知イベントが生成されるまでに処理されるローの数を指定します。

構文

Visual Basic

```
Public Property NotifyAfter As Integer
```

C#

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

備考

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

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

この値をゼロ未満の値に設定すると、エラーがスローされます。

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

関連情報

[BulkCopyTimeout プロパティ \[34 ページ\]](#)

1.2.10 ULRowsCopied イベント

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

構文

Visual Basic

```
Public Event ULRowsCopied As ULRowsCopiedEventHandler
```

C#

```
public ULRowsCopiedEventHandler ULRowsCopied;
```

備考

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

関連情報

[NotifyAfter プロパティ \[36 ページ\]](#)

1.3 ULBulkCopyColumnMapping クラス

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

構文

Visual Basic

```
Public NotInheritable Class ULBulkCopyColumnMapping
```

C#

```
public sealed class ULBulkCopyColumnMapping
```

メンバー

ULBulkCopyColumnMapping 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

変数とタイプ	コンストラクタ	説明
public	ULBulkCopyColumnMapping [39 ページ]	新しいカラムマッピングを作成します。

プロパティ

変数とタイプ	プロパティ	説明
public string	DestinationColumn [43 ページ]	マッピングされる送信先データベーステーブルのカラムの名前を指定します。
public int	DestinationOrdinal [44 ページ]	マッピングされる送信先データベーステーブルにおけるカラムの順序を指定します。

変更子とタイプ	プロパティ	説明
public string	SourceColumn [45 ページ]	データソースにマッピングされるカラムの名前を指定します。
public int	SourceOrdinal [45 ページ]	データソース内のソースカラムの序数位置を指定します。

備考

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

このセクションの内容:

[ULBulkCopyColumnMapping コンストラクタ \[39 ページ\]](#)

新しいカラムマッピングを作成します。

[DestinationColumn プロパティ \[43 ページ\]](#)

マッピングされる送信先データベーステーブルのカラムの名前を指定します。

[DestinationOrdinal プロパティ \[44 ページ\]](#)

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

[SourceColumn プロパティ \[45 ページ\]](#)

データソースにマッピングされるカラムの名前を指定します。

[SourceOrdinal プロパティ \[45 ページ\]](#)

データソース内のソースカラムの序数位置を指定します。

関連情報

[ULBulkCopy クラス \[23 ページ\]](#)

1.3.1 ULBulkCopyColumnMapping コンストラクタ

新しいカラムマッピングを作成します。

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	ULBulkCopyColumnMapping() [40 ページ]	新しいカラムマッピングを作成します。
public	ULBulkCopyColumnMapping(int, int) [40 ページ]	カラムの順序または名前を使用してソースカラムと送信先カラムを参照する、新しいカラムマッピングを作成します。
public	ULBulkCopyColumnMapping(int, string) [41 ページ]	ソースカラムを参照するカラムの序数と、送信先カラムを参照するカラム名を使用して、新しいカラムマッピングを作成します。
public	ULBulkCopyColumnMapping(string, int) [42 ページ]	ソースカラムを参照するカラム名と、送信先カラムを参照するカラムの順序を使用して、新しいカラムマッピングを作成します。
public	ULBulkCopyColumnMapping(string, string) [42 ページ]	カラムの名前を使用してソースカラムと送信先カラムを参照する、新しいカラムマッピングを作成します。

このセクションの内容:

[ULBulkCopyColumnMapping\(\) コンストラクタ \[40 ページ\]](#)

新しいカラムマッピングを作成します。

[ULBulkCopyColumnMapping\(int, int\) コンストラクタ \[40 ページ\]](#)

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

[ULBulkCopyColumnMapping\(int, string\) コンストラクタ \[41 ページ\]](#)

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

[ULBulkCopyColumnMapping\(string, int\) コンストラクタ \[42 ページ\]](#)

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

[ULBulkCopyColumnMapping\(string, string\) コンストラクタ \[42 ページ\]](#)

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

1.3.1.1 ULBulkCopyColumnMapping() コンストラクタ

新しいカラムマッピングを作成します。

構文

Visual Basic

```
Public Sub ULBulkCopyColumnMapping ()
```

C#

```
public ULBulkCopyColumnMapping ()
```

備考

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

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

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

構文

Visual Basic

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

C#

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

パラメータ

sourceColumnOrdinal データソース内のソースカラムの序数位置。データソースの最初のカラムの序数位置は 0 です。

destinationColumnOrdinal 送信先テーブル内の送信先カラムの序数位置。テーブルの最初のカラムの序数位置は 0 です。

備考

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

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

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

構文

Visual Basic

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

C#

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

パラメータ

sourceColumnOrdinal データソース内のソースカラムの序数位置。データソースの最初のカラムの序数位置は 0 です。

destinationColumn 送信先テーブル内の送信先カラムの名前。

備考

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

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

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

構文

Visual Basic

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

C#

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

パラメータ

sourceColumn データソース内のソースカラムの名前。

destinationColumnOrdinal 送信先テーブル内の送信先カラムの序数位置。テーブルの最初のカラムの序数位置は 0 です。

備考

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

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

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

構文

Visual Basic

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

C#

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

パラメータ

sourceColumn データソース内のソースカラムの名前。

destinationColumn 送信先テーブル内の送信先カラムの名前。

備考

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

1.3.2 DestinationColumn プロパティ

マッピングされる送信先データベーステーブルのカラムの名前を指定します。

構文

Visual Basic

```
Public Property DestinationColumn As String
```

C#

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

備考

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

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

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

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

関連情報

[DestinationOrdinal プロパティ \[44 ページ\]](#)

[DestinationOrdinal プロパティ \[44 ページ\]](#)

1.3.3 DestinationOrdinal プロパティ

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

構文

Visual Basic

```
Public Property DestinationOrdinal As Integer
```

C#

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

備考

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

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

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

関連情報

[DestinationColumn プロパティ \[43 ページ\]](#)

[DestinationColumn プロパティ \[43 ページ\]](#)

1.3.4 SourceColumn プロパティ

データソースにマッピングされるカラムの名前を指定します。

構文

Visual Basic

```
Public Property SourceColumn As String
```

C#

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

備考

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

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

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

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

関連情報

[SourceOrdinal プロパティ \[45 ページ\]](#)

[SourceOrdinal プロパティ \[45 ページ\]](#)

1.3.5 SourceOrdinal プロパティ

データソース内のソースカラムの序数位置を指定します。

構文

Visual Basic

```
Public Property SourceOrdinal As Integer
```

C#

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

備考

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

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

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

関連情報

[SourceColumn プロパティ \[45 ページ\]](#)

1.4 ULBulkCopyColumnMappingCollection クラス

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

構文

Visual Basic

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

C#

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

メンバー

ULBulkCopyColumnMappingCollection 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 ULBulkCopyColumnMapping	Add [48 ページ]	指定された ULBulkCopyColumnMapping オブジェクトをコレクションに追加します。
public bool	Contains(ULBulkCopyColumnMapping) [54 ページ]	コレクション内に指定された ULBulkCopyColumnMapping オブジェクトが存在するかどうかを返します。

変数とタイプ	メソッド	説明
public void	CopyTo(ULBulkCopyColumnMapping[], int) [54 ページ]	ULBulkCopyColumnMappingCollection オブジェクトの要素を、特定のインデックスを先頭に、ULBulkCopyColumnMapping オブジェクトの配列にコピーします。
public int	IndexOf(ULBulkCopyColumnMapping) [55 ページ]	コレクション内で、指定された ULBulkCopyColumnMapping オブジェクトのインデックスを返します。
public void	Remove(ULBulkCopyColumnMapping) [56 ページ]	指定された ULBulkCopyColumnMapping オブジェクトを ULBulkCopyColumnMappingCollection オブジェクトから削除します。
public new void	RemoveAt(int) [57 ページ]	指定されたインデックス位置のマッピングをコレクションから削除します。

プロパティ

変数とタイプ	プロパティ	説明
public ULBulkCopyColumnMapping	this[int index] [57 ページ]	指定されたインデックス位置の ULBulkCopyColumnMapping オブジェクトを取得します。

備考

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

このセクションの内容:

[Add メソッド \[48 ページ\]](#)

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

[Contains\(ULBulkCopyColumnMapping\) メソッド \[54 ページ\]](#)

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

[CopyTo\(ULBulkCopyColumnMapping\[\], int\) メソッド \[54 ページ\]](#)

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

[IndexOf\(ULBulkCopyColumnMapping\) メソッド \[55 ページ\]](#)

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

[Remove\(ULBulkCopyColumnMapping\) メソッド \[56 ページ\]](#)

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

[RemoveAt\(int\) メソッド \[57 ページ\]](#)

指定されたインデックス位置のマッピングをコレクションから削除します。

[this\[int index\] プロパティ \[57 ページ\]](#)

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

関連情報

[ULBulkCopyColumnMapping クラス \[37 ページ\]](#)

1.4.1 Add メソッド

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

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public ULBulkCopyColumnMapping	Add(ULBulkCopyColumnMapping) [49 ページ]	指定された ULBulkCopyColumnMapping オブジェクトをコレクションに追加します。
public ULBulkCopyColumnMapping	Add(int, int) [50 ページ]	ソースカラムと送信先カラムの両方を指定する順序を使用して、新しい ULBulkCopyColumnMapping オブジェクトを作成し、このマッピングをコレクションに追加します。
public ULBulkCopyColumnMapping	Add(int, string) [51 ページ]	カラムの順序を使用してソースカラムを参照し、カラム名を使用して送信先カラムを参照する、新しい ULBulkCopyColumnMapping オブジェクトを作成し、このマッピングをコレクションに追加します。
public ULBulkCopyColumnMapping	Add(string, int) [52 ページ]	カラム名を使用してソースカラムを参照し、カラムの順序を使用して送信先カラムを参照する、新しい ULBulkCopyColumnMapping オブジェクトを作成し、このマッピングをコレクションに追加します。
public ULBulkCopyColumnMapping	Add(string, string) [53 ページ]	ソースカラムと送信先カラムの両方を指定するカラム名を使用して、新しい ULBulkCopyColumnMapping オブジェクトを作成し、このマッピングをコレクションに追加します。

このセクションの内容:

[Add\(ULBulkCopyColumnMapping\) メソッド \[49 ページ\]](#)

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

[Add\(int, int\) メソッド \[50 ページ\]](#)

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

[Add\(int, string\) メソッド \[51 ページ\]](#)

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

[Add\(string, int\) メソッド \[52 ページ\]](#)

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

[Add\(string, string\) メソッド \[53 ページ\]](#)

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

1.4.1.1 Add(ULBulkCopyColumnMapping) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

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

備考

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

関連情報

[ULBulkCopyColumnMapping クラス \[37 ページ\]](#)

1.4.1.2 Add(int, int) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

sourceColumnOrdinal データソース内のソースカラムの序数位置。データソースの最初のカラムの序数位置は 0 です。

destinationColumnOrdinal 送信先テーブル内の送信先カラムの序数位置。テーブルの最初のカラムの序数位置は 0 です。

備考

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

関連情報

[ULBulkCopyColumnMapping クラス \[37 ページ\]](#)

1.4.1.3 Add(int, string) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

sourceColumnOrdinal データソース内のソースカラムの序数位置。データソースの最初のカラムの序数位置は 0 です。

destinationColumn 送信先テーブル内の送信先カラムの名前。

備考

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

関連情報

[ULBulkCopyColumnMapping クラス \[37 ページ\]](#)

1.4.1.4 Add(string, int) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

sourceColumn データソース内のソースカラムの名前。

destinationColumnOrdinal 送信先テーブル内の送信先カラムの序数位置。テーブルの最初のカラムの序数位置は 0 です。

備考

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

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

関連情報

[ULBulkCopyColumnMapping クラス \[37 ページ\]](#)

1.4.1.5 Add(string, string) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

sourceColumn データソース内のソースカラムの名前。

destinationColumn 送信先テーブル内の送信先カラムの名前。

備考

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

関連情報

[ULBulkCopyColumnMapping クラス \[37 ページ\]](#)

1.4.2 Contains(ULBulkCopyColumnMapping) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

value 有効な ULBulkCopyColumnMapping オブジェクト。

戻り値

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

関連情報

[ULBulkCopyColumnMapping クラス \[37 ページ\]](#)

1.4.3 CopyTo(ULBulkCopyColumnMapping[], int) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

array ULBulkCopyColumnMappingCollection オブジェクトの要素のコピー先である、1 次元の ULBulkCopyColumnMapping 配列。配列のインデックスは 0 から始まります。

index コピーが開始される、配列内の 0 から始まるインデックス。

関連情報

[ULBulkCopyColumnMapping クラス \[37 ページ\]](#)

1.4.4 IndexOf(ULBulkCopyColumnMapping) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

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

戻り値

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

関連情報

[ULBulkCopyColumnMapping クラス \[37 ページ\]](#)

1.4.5 Remove(ULBulkCopyColumnMapping) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

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

関連情報

[ULBulkCopyColumnMapping クラス \[37 ページ\]](#)

1.4.6 RemoveAt(int) メソッド

指定されたインデックス位置のマッピングをコレクションから削除します。

構文

Visual Basic

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

C#

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

パラメータ

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

1.4.7 this[int index] プロパティ

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

構文

Visual Basic

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

C#

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

備考

ULBulkCopyColumnMapping オブジェクトが返されます。

関連情報

[ULBulkCopyColumnMapping クラス \[37 ページ\]](#)

1.5 ULCommand クラス

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

構文

Visual Basic

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

C#

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

メンバー

ULCommand 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	ULCommand [63 ページ]	ULCommand オブジェクトを初期化します。

メソッド

変更子とタイプ	メソッド	説明
public IAsyncResult	BeginExecuteNonQuery [67 ページ]	この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始します。
public IAsyncResult	BeginExecuteReader [70 ページ]	この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始し、結果セットを取得します。
public override void	Cancel() [75 ページ]	このメソッドは Ultra Light.NET ではサポートされていません。
protected override DbParameter	CreateDbParameter() [75 ページ]	
public new ULParameter	CreateParameter() [76 ページ]	ULCommand オブジェクトにパラメータを指定するために ULParameter オブジェクトを提供します。
protected override void	Dispose(bool) [76 ページ]	ULCommand オブジェクトによって使用されるアンマネージリソースを解放するほか、必要に応じてマネージリソースを解放します。
public int	EndExecuteNonQuery(IAsyncResult) [77 ページ]	SQL 文の非同期実行を終了します。

変数とタイプ	メソッド	説明
public ULDataReader	EndExecuteReader(IAsyncResult) [81 ページ]	SQL 文の非同期実行を終了し、要求された ULDataReader を返します。
protected override DbDataReader	ExecuteDbDataReader(CommandBehavior) [84 ページ]	
public override unsafe int	ExecuteNonQuery() [84 ページ]	SQL INSERT 文、DELETE 文、UPDATE 文のように、結果セットを返さない文を実行します。
public new ULDataReader	ExecuteReader [85 ページ]	SQL SELECT 文を実行し、結果セットを返します。
public ULResultSet	ExecuteResultSet [89 ページ]	UL 拡張: SQL SELECT 文を実行し、ULResultSet オブジェクトとして結果セットを返します。
public override object	ExecuteScalar() [92 ページ]	SQL SELECT 文を実行し、単一の値を返します。
public ULTable	ExecuteTable [94 ページ]	UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。
public override unsafe void	Prepare() [97 ページ]	このコマンドの SQL 文を事前にコンパイルして格納します。

プロパティ

変数とタイプ	プロパティ	説明
public override string	CommandText [98 ページ]	ULCommand.CommandType プロパティが System.Data.CommandType.TableDirect である場合の、SQL 文のテキストまたはテーブルの名前を指定します。
public override int	CommandTimeout [99 ページ]	この機能は Ultra Light.NET ではサポートされていません。
public override CommandType	CommandType [99 ページ]	実行されるコマンドのタイプを指定します。
public new ULConnection	Connection [100 ページ]	ULCommand オブジェクトを実行する接続オブジェクトです。
protected override DbConnection	DbConnection [101 ページ]	
protected override DbParameterCollection	DbParameterCollection [101 ページ]	
protected override DbTransaction	DbTransaction [101 ページ]	
public override bool	DesignTimeVisible [102 ページ]	カスタマイズされた Windows Form Designer 制御で ULCommand オブジェクトを参照できるようにするかどうかを指定します。

変更子とタイプ	プロパティ	説明
public string	IndexName [102 ページ]	UL 拡張: ULCommand.CommandType プロパティが System.Data.CommandType.TableDirect である場合に、テーブルを開く (ソートする) インデックスの名前を指定します。
public new ULParameterCollection	Parameters [103 ページ]	現在の文のパラメータを指定します。
public unsafe string	Plan [104 ページ]	UL 拡張: Ultra Light.NET がクエリの実行に使用するアクセスプランを返します。
public new ULTransaction	Transaction [104 ページ]	ULCommand オブジェクトが実行される ULTransaction オブジェクトを指定します。
public override UpdateRowSource	UpdatedRowSource [105 ページ]	ULDataAdapterUpdate メソッドによって使用されるときにコマンドの結果が DataRow に適用される方法を指定します。

備考

このオブジェクトを使用すると、文またはクエリを効率的に何回も実行できます。

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

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

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

メソッド	説明
ULCommand.ExecuteNonQuery	SQL INSERT 文、DELETE 文、UPDATE 文のように、結果セットを返さない文を実行します。
ULCommand.ExecuteReader()	SQL SELECT 文を実行し、ULDataReader オブジェクトで結果セットを返します。読み込み専用の結果セットの作成には、このメソッドを使用します。
ULCommand.ExecuteResultSet()	UL 拡張: SQL SELECT 文を実行し、ULResultSet オブジェクトで結果セットを返します。可変の結果セットの作成には、このメソッドを使用します。
ULCommand.ExecuteScalar	SQL SELECT 文を実行し、単一の値を返します。
ULCommand.ExecuteTable()	UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。ULCommand.CommandText プロパティはテーブルの名前として解釈され、ULCommand.IndexName プロパティはテーブルのソート順の指定に使用できます。ULCommand.CommandType プロパティは System.Data.CommandType.TableDirect であることが必要です。

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

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

このセクションの内容:

[ULCommand コンストラクタ \[63 ページ\]](#)

ULCommand オブジェクトを初期化します。

[BeginExecuteNonQuery メソッド \[67 ページ\]](#)

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

[BeginExecuteReader メソッド \[70 ページ\]](#)

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

[Cancel\(\) メソッド \[75 ページ\]](#)

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

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

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

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

[Dispose\(bool\) メソッド \[76 ページ\]](#)

ULCommand オブジェクトによって使用されるアンマネージリソースを解放するほか、必要に応じてマネージリソースを解放します。

[EndExecuteNonQuery\(IAsyncResult\) メソッド \[77 ページ\]](#)

SQL 文の非同期実行を終了します。

[EndExecuteReader\(IAsyncResult\) メソッド \[81 ページ\]](#)

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

[ExecuteDbDataReader\(CommandBehavior\) メソッド \[84 ページ\]](#)

[ExecuteNonQuery\(\) メソッド \[84 ページ\]](#)

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

[ExecuteReader メソッド \[85 ページ\]](#)

SQL SELECT 文を実行し、結果セットを返します。

[ExecuteResultSet メソッド \[89 ページ\]](#)

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

[ExecuteScalar\(\) メソッド \[92 ページ\]](#)

SQL SELECT 文を実行し、単一の値を返します。

[ExecuteTable メソッド \[94 ページ\]](#)

UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。

[Prepare\(\) メソッド \[97 ページ\]](#)

このコマンドの SQL 文を事前にコンパイルして格納します。

[CommandText プロパティ \[98 ページ\]](#)

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

[CommandTimeout プロパティ \[99 ページ\]](#)

この機能は Ultra Light.NET ではサポートされていません。

[CommandType プロパティ \[99 ページ\]](#)

実行されるコマンドのタイプを指定します。

[Connection プロパティ \[100 ページ\]](#)

ULCommand オブジェクトを実行する接続オブジェクトです。

[DbConnection プロパティ \[101 ページ\]](#)

[DbParameterCollection プロパティ \[101 ページ\]](#)

[DbTransaction プロパティ \[101 ページ\]](#)

[DesignTimeVisible プロパティ \[102 ページ\]](#)

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

[IndexName プロパティ \[102 ページ\]](#)

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

[Parameters プロパティ \[103 ページ\]](#)

現在の文のパラメータを指定します。

[Plan プロパティ \[104 ページ\]](#)

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

[Transaction プロパティ \[104 ページ\]](#)

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

[UpdatedRowSource プロパティ \[105 ページ\]](#)

ULDataAdapter.Update メソッドによって使用されるときにコマンドの結果が DataRow に適用される方法を指定します。

関連情報

[CreateCommand\(\) メソッド \[143 ページ\]](#)

[Transaction プロパティ \[104 ページ\]](#)

[ExecuteNonQuery\(\) メソッド \[84 ページ\]](#)

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

[ExecuteResultSet\(\) メソッド \[89 ページ\]](#)

[ExecuteScalar\(\) メソッド \[92 ページ\]](#)

[ExecuteTable\(\) メソッド \[94 ページ\]](#)

[ULTable クラス \[526 ページ\]](#)

[CommandText プロパティ \[98 ページ\]](#)

[CommandType プロパティ \[99 ページ\]](#)

[IndexName プロパティ \[102 ページ\]](#)

[ULResultSet クラス \[439 ページ\]](#)

1.5.1 ULCommand コンストラクタ

ULCommand オブジェクトを初期化します。

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	ULCommand() [63 ページ]	ULCommand オブジェクトを初期化します。
public	ULCommand(string) [64 ページ]	ULCommand オブジェクトを、指定されたコマンドテキストで初期化します。
public	ULCommand(string, ULConnection) [65 ページ]	ULCommand オブジェクトを、指定されたコマンドテキストと接続で初期化します。
public	ULCommand(string, ULConnection, ULTransaction) [66 ページ]	ULCommand オブジェクトを、指定されたコマンドテキスト、接続、トランザクションで初期化します。

このセクションの内容:

[ULCommand\(\) コンストラクタ \[63 ページ\]](#)

ULCommand オブジェクトを初期化します。

[ULCommand\(string\) コンストラクタ \[64 ページ\]](#)

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

[ULCommand\(string, ULConnection\) コンストラクタ \[65 ページ\]](#)

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

[ULCommand\(string, ULConnection, ULTransaction\) コンストラクタ \[66 ページ\]](#)

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

1.5.1.1 ULCommand() コンストラクタ

ULCommand オブジェクトを初期化します。

構文

Visual Basic

```
Public Sub ULCommand ()
```

C#

```
public ULCommand ()
```

備考

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

関連情報

[CreateCommand\(\) メソッド \[143 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

[CommandText プロパティ \[98 ページ\]](#)

[Connection プロパティ \[100 ページ\]](#)

[Transaction プロパティ \[104 ページ\]](#)

1.5.1.2 ULCommand(string) コンストラクタ

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

構文

Visual Basic

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

C#

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

パラメータ

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

備考

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

関連情報

[CreateCommand\(\) メソッド \[143 ページ\]](#)

[ULCommand\(\) コンストラクタ \[63 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

[Connection プロパティ \[100 ページ\]](#)

[Transaction プロパティ \[104 ページ\]](#)

[CommandType プロパティ \[99 ページ\]](#)

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

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

構文

Visual Basic

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

C#

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

パラメータ

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

connection 現在の接続を表す ULConnection オブジェクト。

備考

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

関連情報

[ULConnection クラス \[121 ページ\]](#)

[CreateCommand\(\) メソッド \[143 ページ\]](#)

[ULCommand\(\) コンストラクタ \[63 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

[Transaction プロパティ \[104 ページ\]](#)

[CommandType プロパティ \[99 ページ\]](#)

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

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

構文

Visual Basic

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

C#

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

パラメータ

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

connection 現在の接続を表す ULConnection オブジェクト。

transaction ULCommand オブジェクトが実行される ULTransaction オブジェクト。

関連情報

[CreateCommand\(\) メソッド \[143 ページ\]](#)

[ULCommand\(\) コンストラクタ \[63 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

[CommandType プロパティ \[99 ページ\]](#)

[ULTransaction クラス \[575 ページ\]](#)

1.5.2 BeginExecuteNonQuery メソッド

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

オーバーロードリスト

変数とタイプ	オーバーロード名	説明
public IAsyncResult	BeginExecuteNonQuery() [68 ページ]	この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始します。
public IAsyncResult	BeginExecuteNonQuery(AsyncCallback, object) [68 ページ]	コールバックプロシージャと状態情報を指定して、この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始します。

このセクションの内容:

[BeginExecuteNonQuery\(\) メソッド \[68 ページ\]](#)

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

[BeginExecuteNonQuery\(AsyncCallback, object\) メソッド \[68 ページ\]](#)

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

1.5.2.1 BeginExecuteNonQuery() メソッド

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

構文

Visual Basic

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

C#

```
public IAsyncResult BeginExecuteNonQuery ()
```

戻り値

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

例外

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

関連情報

[EndExecuteNonQuery\(IAsyncResult\) メソッド \[77 ページ\]](#)

[EndExecuteNonQuery\(IAsyncResult\) メソッド \[77 ページ\]](#)

1.5.2.2 BeginExecuteNonQuery(AsyncCallback, object) メソッド

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

構文

Visual Basic

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

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

C#

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

パラメータ

callback コマンドの実行が終了すると起動される System.AsyncCallback デリゲート。コールバックが必要ないことを示すには、NULL (Microsoft Visual Basic の場合は Nothing) を渡します。

stateObject コールバックプロシージャに渡される、ユーザ定義のステータスオブジェクト。コールバックプロシージャからこのオブジェクトを取得するには、System.IAsyncResult.AsyncState プロパティを使用します。

戻り値

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

例外

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

関連情報

[EndExecuteNonQuery\(IAsyncResult\) メソッド \[77 ページ\]](#)

1.5.3 BeginExecuteReader メソッド

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

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public IAsyncResult	BeginExecuteReader() [71 ページ]	この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始し、結果セットを取得します。
public IAsyncResult	BeginExecuteReader(AsyncCallback, object) [71 ページ]	コールバックプロシージャと状態情報を指定して、この ULCommand オブジェクトで記述される SQL 文の非同期実行を開始し、結果セットを取得します。
public IAsyncResult	BeginExecuteReader(AsyncCallback, object, CommandBehavior) [73 ページ]	コールバックプロシージャと状態情報を指定して、CommandBehavior 値の 1 つを使用してこの ULCommand オブジェクトで記述される SQL 文の非同期実行を開始し、結果セットを取得します。
public IAsyncResult	BeginExecuteReader(CommandBehavior) [74 ページ]	CommandBehavior 値の 1 つを使用してこの ULCommand オブジェクトで記述される SQL 文の非同期実行を開始し、結果セットを取得します。

このセクションの内容:

[BeginExecuteReader\(\) メソッド \[71 ページ\]](#)

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

[BeginExecuteReader\(AsyncCallback, object\) メソッド \[71 ページ\]](#)

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

[BeginExecuteReader\(AsyncCallback, object, CommandBehavior\) メソッド \[73 ページ\]](#)

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

[BeginExecuteReader\(CommandBehavior\) メソッド \[74 ページ\]](#)

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

1.5.3.1 BeginExecuteReader() メソッド

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

構文

Visual Basic

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

C#

```
public IAsyncResult BeginExecuteReader ()
```

戻り値

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

例外

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

関連情報

[EndExecuteReader\(IAsyncResult\) メソッド \[81 ページ\]](#)

[ULDataReader クラス \[275 ページ\]](#)

1.5.3.2 BeginExecuteReader(AsyncCallback, object) メソッド

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

構文

Visual Basic

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

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

C#

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

パラメータ

callback コマンドの実行が終了すると起動される System.AsyncCallback デリゲート。コールバックが必要ないことを示すには、NULL (Microsoft Visual Basic の場合は Nothing) を渡します。

stateObject コールバックプロシージャに渡される、ユーザ定義のステータスオブジェクト。コールバックプロシージャからこのオブジェクトを取得するには、System.IAsyncResult.AsyncState プロパティを使用します。

戻り値

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

例外

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

関連情報

[ULDataReader クラス \[275 ページ\]](#)

[EndExecuteReader\(IAsyncResult\) メソッド \[81 ページ\]](#)

[EndExecuteReader\(IAsyncResult\) メソッド \[81 ページ\]](#)

1.5.3.3 BeginExecuteReader(AsyncCallback, object, CommandBehavior) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

callback コマンドの実行が終了すると起動される System.AsyncCallback デリゲート。コールバックが必要ないことを示すには、NULL (Microsoft Visual Basic の場合は Nothing) を渡します。

stateObject コールバックプロシージャに渡される、ユーザ定義のステータスオブジェクト。コールバックプロシージャからこのオブジェクトを取得するには、System.IAsyncResult.AsyncState プロパティを使用します。

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

戻り値

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

例外

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

関連情報

[EndExecuteReader\(IAsyncResult\) メソッド \[81 ページ\]](#)

[ULDataReader クラス \[275 ページ\]](#)

1.5.3.4 BeginExecuteReader(CommandBehavior) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

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

戻り値

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

例外

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

関連情報

[EndExecuteReader\(IAsyncResult\) メソッド \[81 ページ\]](#)

[ULDataReader クラス \[275 ページ\]](#)

1.5.4 Cancel() メソッド

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

構文

Visual Basic

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

C#

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

備考

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

1.5.5 CreateDbParameter() メソッド

構文

Visual Basic

```
Protected Overrides Function CreateDbParameter () As DbParameter
```

C#

```
protected override DbParameter CreateDbParameter ()
```

1.5.6 CreateParameter() メソッド

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

構文

Visual Basic

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

C#

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

戻り値

ULParameter オブジェクトとして返される新しいパラメータ。

備考

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

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

関連情報

[ULParameter クラス \[389 ページ\]](#)

1.5.7 Dispose(bool) メソッド

ULCommand オブジェクトによって使用されるアンマネージリソースを解放するほか、必要に応じてマネージリソースを解放します。

構文

Visual Basic

```
Protected Overrides Sub Dispose (ByVal disposing As Boolean)
```

C#

```
protected override void Dispose (bool disposing)
```

パラメータ

disposing true の場合は、マネージリソースとアンマネージリソースの両方を破棄します。false の場合は、アンマネージリソースのみを破棄します。

1.5.8 EndExecuteNonQuery(IAsyncResult) メソッド

SQL 文の非同期実行を終了します。

構文

Visual Basic

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

C#

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

パラメータ

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

戻り値

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

例外

ArgumentException asyncResult パラメータが null (Microsoft Visual Basic の Nothing) です。

InvalidOperationException 1 回のコマンド実行に対して EndExecuteNonQuery(IAsyncResult) メソッドが複数回呼び出されたか、このメソッドがその実行メソッドと一致しませんでした。

備考

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

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

実行の完了を待機するには、4 とおりの方法があります。

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

```
' Visual Basic
Dim cmd As ULCommand = new ULCommand( _
    "UPDATE Departments" _
    + " SET DepartmentName = 'Engineering'" _
    + " WHERE DepartmentID=100", _
    conn _
)
Dim res As IAsyncResult res = _
    cmd.BeginExecuteNonQuery()
' Perform other work.
' This blocks until the command completes.
Dim rowCount As Integer = _
    cmd.EndExecuteNonQuery( res )
```

対応する C# 言語のコードを次に示します。

```
// C#
ULCommand cmd = new ULCommand(
    "UPDATE Departments"
    + " SET DepartmentName = 'Engineering'"
    + " WHERE DepartmentID=100",
    conn
);
IAsyncResult res = cmd.BeginExecuteNonQuery();
// Perform other work.
// This blocks until the command completes.
int rowCount = cmd.EndExecuteNonQuery( res );
```

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

```
' Visual Basic
Dim cmd As ULCommand = new ULCommand( _
    "UPDATE Departments" _
    + " SET DepartmentName = 'Engineering'" _
    + " WHERE DepartmentID=100", _
    conn _
)
Dim res As IAsyncResult res = _
    cmd.BeginExecuteNonQuery()
While( !res.IsCompleted )
    ' Perform other work.
End While
' This blocks until the command completes.
Dim rowCount As Integer = _
```

```
cmd.EndExecuteNonQuery( res )
```

対応する C# 言語のコードを次に示します。

```
// C#
ULCommand cmd = new ULCommand(
    "UPDATE Departments"
    + " SET DepartmentName = 'Engineering'"
    + " WHERE DepartmentID=100",
    conn
);
IAsyncResult res = cmd.BeginExecuteNonQuery();
while( !res.IsCompleted ) {
    // Perform other work.
}
// This blocks until the command completes.
int rowCount = cmd.EndExecuteNonQuery( res );
```

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

```
' Visual Basic
Dim cmd As ULCommand = new ULCommand( _
    "UPDATE Departments" _
    + " SET DepartmentName = 'Engineering'" _
    + " WHERE DepartmentID=100", _
    conn _
)
Dim res As IAsyncResult res = _
    cmd.BeginExecuteNonQuery()
' Perform other work.
Dim wh As WaitHandle = res.AsyncWaitHandle
wh.WaitOne()
' This does not block because the command is finished.
Dim rowCount As Integer = _
    cmd.EndExecuteNonQuery( res )
```

対応する C# 言語のコードを次に示します。

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

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

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

```

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

```

対応する C# 言語のコードを次に示します。

```

// C#
private void callbackFunction( IAsyncResult ar )
{
    ULCommand cmd = (ULCommand) ar.AsyncState;
    // This won't block since the command has completed.
    int rowCount = cmd.EndExecuteNonQuery();
}
// Elsewhere in the code
private void DoStuff()
{
    ULCommand cmd = new ULCommand(
        "UPDATE Departments"
        + " SET DepartmentName = 'Engineering'"
        + " WHERE DepartmentID=100",
        conn
    );
    IAsyncResult res = cmd.BeginExecuteNonQuery(
        callbackFunction, cmd
    );
    // Perform other work. The callback function
    // is called when the command completes.
}

```

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

関連情報

[BeginExecuteNonQuery\(\) メソッド \[68 ページ\]](#)

1.5.9 EndExecuteReader(IAsyncResult) メソッド

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

構文

Visual Basic

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

C#

```
public UDataReader EndExecuteReader (IAsyncResult asyncResult)
```

パラメータ

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

戻り値

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

例外

ArgumentException asyncResult パラメータが null (Microsoft Visual Basic の Nothing) です。

InvalidOperationException 1 回のコマンド実行に対して EndExecuteReader メソッドが複数回呼び出されたか、このメソッドがその実行メソッドに一致しませんでした。

備考

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

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

実行の完了を待機するには、4 とおりの方法があります。

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

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

対応する C# 言語のコードを次に示します。

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

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

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

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

```
' Visual Basic
Dim cmd As ULCommand = new ULCommand( _
    "SELECT * FROM Departments", conn _
)
Dim res As IAsyncResult res = _
    cmd.BeginExecuteReader()
' Perform other work.
Dim wh As WaitHandle = res.AsyncWaitHandle
wh.WaitOne()
```

```
' This does not block because the command is finished.
Dim reader As ULDataReader =
    cmd.EndExecuteReader( res )
```

対応する C# 言語のコードを次に示します。

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

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

```
' Visual Basic
Private Sub callbackFunction(ByVal ar As IAsyncResult)
    Dim cmd As ULCommand =
        CType(ar.AsyncState, ULCommand)
    ' This won't block since the command has completed.
    Dim reader As ULDataReader = cmd.EndExecuteReader()
End Sub
' Elsewhere in the code
Private Sub DoStuff()
    Dim cmd As ULCommand = new ULCommand(
        "SELECT * FROM Departments", conn
    )
    Dim res As IAsyncResult =
        cmd.BeginExecuteReader(
            callbackFunction, cmd
        )
    ' Perform other work. The callback function
    ' is called when the command completes.
End Sub
```

対応する C# 言語のコードを次に示します。

```
// C#
private void callbackFunction( IAsyncResult ar )
{
    ULCommand cmd = (ULCommand) ar.AsyncState;
    // This won't block since the command has completed.
    ULDataReader reader = cmd.EndExecuteReader();
}
// Elsewhere in the code.
private void DoStuff()
{
    ULCommand cmd = new ULCommand(
        "SELECT * FROM Departments", conn
    );
    IAsyncResult res = cmd.BeginExecuteReader(callbackFunction, cmd);
    // Perform other work. The callback function
    // is called when the command completes.
}
```

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

関連情報

[BeginExecuteReader\(\) メソッド \[71 ページ\]](#)

[ULDataReader クラス \[275 ページ\]](#)

1.5.10 ExecuteDbDataReader(CommandBehavior) メソッド

構文

Visual Basic

```
Protected Overrides Function ExecuteDbDataReader (ByVal cmdBehavior As CommandBehavior) As DbDataReader
```

C#

```
protected override DbDataReader ExecuteDbDataReader (CommandBehavior cmdBehavior)
```

1.5.11 ExecuteNonQuery() メソッド

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

構文

Visual Basic

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

C#

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

戻り値

影響を受けたローの数。

例外

ULException class SQL エラーが発生しました。

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

備考

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

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

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

関連情報

[CommandText プロパティ \[98 ページ\]](#)

[CommandType プロパティ \[99 ページ\]](#)

[Connection プロパティ \[100 ページ\]](#)

[Parameters プロパティ \[103 ページ\]](#)

[Transaction プロパティ \[104 ページ\]](#)

1.5.12 ExecuteReader メソッド

SQL SELECT 文を実行し、結果セットを返します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public new ULDataReader	ExecuteReader() [86 ページ]	SQL SELECT 文を実行し、結果セットを返します。
public new ULDataReader	ExecuteReader(CommandBehavior) [87 ページ]	コマンドの動作を指定して SQL SELECT 文を実行し、結果セットを返します。

このセクションの内容:

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

SQL SELECT 文を実行し、結果セットを返します。

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

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

1.5.12.1 ExecuteReader() メソッド

SQL SELECT 文を実行し、結果セットを返します。

構文

Visual Basic

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

C#

```
public new UDataReader ExecuteReader ()
```

戻り値

UDataReader オブジェクトとして返される結果セット。

例外

ULException class SQL エラーが発生しました。

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

備考

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

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

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

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

関連情報

- [CommandText プロパティ \[98 ページ\]](#)
- [CommandType プロパティ \[99 ページ\]](#)
- [Connection プロパティ \[100 ページ\]](#)
- [Parameters プロパティ \[103 ページ\]](#)
- [ExecuteResultSet\(\) メソッド \[89 ページ\]](#)
- [ExecuteTable\(\) メソッド \[94 ページ\]](#)
- [Transaction プロパティ \[104 ページ\]](#)
- [ULDataAdapter クラス \[238 ページ\]](#)
- [ULDataReader クラス \[275 ページ\]](#)
- [ULTable クラス \[526 ページ\]](#)

1.5.12.2 ExecuteReader(CommandBehavior) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

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

戻り値

ULDataReader オブジェクトとして返される結果セット。

例外

ULException class SQL エラーが発生しました。

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

備考

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

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

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

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

関連情報

[CommandText プロパティ \[98 ページ\]](#)

[Connection プロパティ \[100 ページ\]](#)

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

[ExecuteResultSet\(CommandBehavior\) メソッド \[91 ページ\]](#)

[ExecuteTable\(CommandBehavior\) メソッド \[95 ページ\]](#)

[Parameters プロパティ \[103 ページ\]](#)

[Transaction プロパティ \[104 ページ\]](#)

[ULDataAdapter クラス \[238 ページ\]](#)

[ULDataReader クラス \[275 ページ\]](#)

[ULTable クラス \[526 ページ\]](#)

[ULDataReader クラス \[275 ページ\]](#)

1.5.13 ExecuteResultSet メソッド

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

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 ULResultSet	ExecuteResultSet() [89 ページ]	UL 拡張: SQL SELECT 文を実行し、ULResultSet オブジェクトとして結果セットを返します。
public ULResultSet	ExecuteResultSet(CommandBehavior) [91 ページ]	UL 拡張: コマンドの動作を指定して SQL SELECT 文を実行し、ULResultSet オブジェクトとして結果セットを返します。

このセクションの内容:

[ExecuteResultSet\(\) メソッド \[89 ページ\]](#)

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

[ExecuteResultSet\(CommandBehavior\) メソッド \[91 ページ\]](#)

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

1.5.13.1 ExecuteResultSet() メソッド

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

構文

Visual Basic

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

C#

```
public ULResultSet ExecuteResultSet ()
```

戻り値

ULResultSet オブジェクトとして返される結果セット。

例外

ULException class SQL エラーが発生しました。

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

備考

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

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

このメソッドは、動的 SQL による位置付け更新と削除をサポートします。

例

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

関連情報

[ULCommand クラス \[58 ページ\]](#)

[CommandText プロパティ \[98 ページ\]](#)

[Parameters プロパティ \[103 ページ\]](#)

[CommandType プロパティ \[99 ページ\]](#)

[ULResultSet クラス \[439 ページ\]](#)

[ULTable クラス \[526 ページ\]](#)

1.5.13.2 ExecuteResultSet(CommandBehavior) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

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

戻り値

ULResultSet オブジェクトとして返される結果セット。

例外

ULException class SQL エラーが発生しました。

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

備考

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

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

このメソッドは、動的 SQL による位置付け更新と削除をサポートします。

関連情報

[ULResultSet クラス \[439 ページ\]](#)

[CommandText プロパティ \[98 ページ\]](#)

[CommandType プロパティ \[99 ページ\]](#)

[Connection プロパティ \[100 ページ\]](#)

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

[ExecuteTable\(CommandBehavior\) メソッド \[95 ページ\]](#)

[Parameters プロパティ \[103 ページ\]](#)

[Transaction プロパティ \[104 ページ\]](#)

[ULDataAdapter クラス \[238 ページ\]](#)

[ULTable クラス \[526 ページ\]](#)

1.5.14 ExecuteScalar() メソッド

SQL SELECT 文を実行し、単一の値を返します。

構文

Visual Basic

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

C#

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

戻り値

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

例外

ULException class SQL エラーが発生しました。

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

備考

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

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

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

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

関連情報

[CommandText プロパティ \[98 ページ\]](#)

[CommandType プロパティ \[99 ページ\]](#)

[Connection プロパティ \[100 ページ\]](#)

[IndexName プロパティ \[102 ページ\]](#)

[Parameters プロパティ \[103 ページ\]](#)

[Transaction プロパティ \[104 ページ\]](#)

1.5.15 ExecuteTable メソッド

UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。

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 ULTable	ExecuteTable() [94 ページ]	UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。
public ULTable	ExecuteTable(CommandBehavior) [95 ページ]	UL 拡張: コマンドの動作を指定してデータベーステーブルを取得し、直接操作します。

このセクションの内容:

[ExecuteTable\(\) メソッド \[94 ページ\]](#)

UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。

[ExecuteTable\(CommandBehavior\) メソッド \[95 ページ\]](#)

UL 拡張: コマンドの動作を指定してデータベーステーブルを取得し、直接操作します。

1.5.15.1 ExecuteTable() メソッド

UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。

構文

Visual Basic

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

C#

```
public ULTable ExecuteTable ()
```

戻り値

ULTable オブジェクトとしてのテーブル。

例外

ULException class SQL エラーが発生しました。

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

備考

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

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

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

関連情報

[CommandText プロパティ \[98 ページ\]](#)

[CommandType プロパティ \[99 ページ\]](#)

[Connection プロパティ \[100 ページ\]](#)

[IndexName プロパティ \[102 ページ\]](#)

[Transaction プロパティ \[104 ページ\]](#)

[ULTable クラス \[526 ページ\]](#)

1.5.15.2 ExecuteTable(CommandBehavior) メソッド

UL 拡張: コマンドの動作を指定してデータベーステーブルを取得し、直接操作します。

構文

Visual Basic

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

C#

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

パラメータ

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

戻り値

ULTable オブジェクトとしてのテーブル。

例外

ULException class SQL エラーが発生しました。

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

備考

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

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

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

関連情報

[CommandText プロパティ \[98 ページ\]](#)

[CommandType プロパティ \[99 ページ\]](#)

[Connection プロパティ \[100 ページ\]](#)

[ExecuteTable\(\) メソッド \[94 ページ\]](#)

[IndexName プロパティ \[102 ページ\]](#)

[Transaction プロパティ \[104 ページ\]](#)

1.5.16 Prepare() メソッド

このコマンドの SQL 文を事前にコンパイルして格納します。

構文

Visual Basic

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

C#

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

例外

ULException class SQL エラーが発生しました。

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

備考

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

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

関連情報

[Connection プロパティ \[100 ページ\]](#)

[Transaction プロパティ \[104 ページ\]](#)

[CommandText プロパティ \[98 ページ\]](#)

1.5.17 CommandText プロパティ

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

構文

Visual Basic

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

C#

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

備考

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

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

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

例

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

対応する C# 言語のコードを次に示します。

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

関連情報

[ExecuteNonQuery\(\) メソッド \[84 ページ\]](#)

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

[ExecuteResultSet\(\) メソッド \[89 ページ\]](#)

[ExecuteScalar\(\) メソッド \[92 ページ\]](#)

[ExecuteTable\(\) メソッド \[94 ページ\]](#)

[CommandType プロパティ \[99 ページ\]](#)

1.5.18 CommandTimeout プロパティ

この機能は Ultra Light.NET ではサポートされていません。

構文

Visual Basic

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

C#

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

備考

値は常に 0 です。

1.5.19 CommandType プロパティ

実行されるコマンドのタイプを指定します。

構文

Visual Basic

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

C#

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

備考

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

サポートされているコマンドタイプは、次のとおりです。

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

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

関連情報

[CommandText プロパティ \[98 ページ\]](#)

[IndexName プロパティ \[102 ページ\]](#)

[ExecuteTable\(\) メソッド \[94 ページ\]](#)

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

[CommandText プロパティ \[98 ページ\]](#)

[ExecuteNonQuery\(\) メソッド \[84 ページ\]](#)

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

[ExecuteScalar\(\) メソッド \[92 ページ\]](#)

1.5.20 Connection プロパティ

ULCommand オブジェクトを実行する接続オブジェクトです。

構文

Visual Basic

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

C#

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

備考

コマンドを実行する ULConnection オブジェクト。

ULCommand オブジェクトは、開かれた接続がないと、実行できません。

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

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

関連情報

[ULConnection クラス \[121 ページ\]](#)

1.5.21 DbConnection プロパティ

構文

Visual Basic

```
Protected Overrides Property DbConnection As DbConnection
```

C#

```
protected override DbConnection DbConnection {get;set;}
```

1.5.22 DbParameterCollection プロパティ

構文

Visual Basic

```
Protected ReadOnly Overrides Property DbParameterCollection As  
DbParameterCollection
```

C#

```
protected override DbParameterCollection DbParameterCollection {get;}
```

1.5.23 DbTransaction プロパティ

構文

Visual Basic

```
Protected Overrides Property DbTransaction As DbTransaction
```

C#

```
protected override DbTransaction DbTransaction {get;set;}
```

1.5.24 DesignTimeVisible プロパティ

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

構文

Visual Basic

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

C#

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

備考

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

1.5.25 IndexName プロパティ

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

構文

Visual Basic

```
Public Property IndexName As String
```

C#

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

備考

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

関連情報

[ExecuteTable\(\) メソッド \[94 ページ\]](#)

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

[CommandType プロパティ \[99 ページ\]](#)

1.5.26 Parameters プロパティ

現在の文のパラメータを指定します。

構文

Visual Basic

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

C#

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

備考

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

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

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

関連情報

[ULParameterCollection クラス \[410 ページ\]](#)

[ULParameter クラス \[389 ページ\]](#)

[CommandText プロパティ \[98 ページ\]](#)

1.5.27 Plan プロパティ

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

構文

Visual Basic

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

C#

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

備考

このプロパティは、主に開発中の使用を目的とします。

クエリ実行プランのテキストベースの記述が含まれる文字列。

1.5.28 Transaction プロパティ

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

構文

Visual Basic

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

C#

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

備考

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

トランザクションがコミットまたはロールバックされた後でコマンドを再使用する場合は、このプロパティをリセットする必要があります。

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

関連情報

[BeginTransaction\(\) メソッド \[135 ページ\]](#)

[ULTransaction クラス \[575 ページ\]](#)

[Connection プロパティ \[100 ページ\]](#)

1.5.29 UpdatedRowSource プロパティ

ULDataAdapterUpdate メソッドによって使用されるときにコマンドの結果が DataRow に適用される方法を指定します。

構文

Visual Basic

```
Public Overrides Property UpdatedRowSource As UpdateRowSource
```

C#

```
public override UpdateRowSource UpdatedRowSource {get;set;}
```

備考

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

1.6 ULCommandBuilder クラス

System.Data.DataSet の変更内容を、関連するデータベース内のデータに一致させる単一テーブルのコマンドを、自動的に生成します。

構文

Visual Basic

```
Public Class ULCommandBuilder Inherits System.Data.Common.DbCommandBuilder
```

C#

```
public class ULCommandBuilder : System.Data.Common.DbCommandBuilder
```

メンバー

ULCommandBuilder 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	ULCommandBuilder [108 ページ]	ULCommandBuilder オブジェクトを初期化します。

メソッド

変更子とタイプ	メソッド	説明
protected override void	ApplyParameterInfo(DbParameter, DataRow, StatementType, bool) [110 ページ]	
public new ULCommand	GetDeleteCommand [110 ページ]	データベースで削除処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。
public new ULCommand	GetInsertCommand [113 ページ]	データベースで挿入処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。
protected override string	GetParameterName [116 ページ]	個々のトピックを参照してください。
protected override string	GetParameterPlaceholder(int) [117 ページ]	
public new ULCommand	GetUpdateCommand [117 ページ]	データベースで更新処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。
protected override DbCommand	InitializeCommand(DbCommand) [120 ページ]	
protected override void	SetRowUpdatingHandler(DbDataAdapter) [120 ページ]	

プロパティ

変更子とタイプ	プロパティ	説明
public new ULDataAdapter	DataAdapter [121 ページ]	SQL 文が自動的に生成される ULDataAdapter オブジェクトを取得または設定します。

備考

ULDataAdapter オブジェクトは、System.Data.DataSet の変更内容を関連するデータソース内のデータに一致させるために必要な SQL 文を、自動的に生成しません。ただし、ULDataAdapter オブジェクトの SelectCommand プロパティを設定すると単一テーブルを更新する SQL 文を自動的に生成するような、ULCommandBuilder オブジェクトを作成することはできます。これにより、設定していない追加の SQL 文が ULCommandBuilder オブジェクトによって生成されます。

例

次の例では、ULCommand オブジェクトを ULDataAdapter オブジェクトと ULConnection オブジェクトとともに使用して、データソースからローを選択します。この例では、接続文字列、クエリ文字列 (SQL SELECT 文)、データベーステーブル名を表す文字列を受け取り、次に ULCommandBuilder オブジェクトを作成します。

```
' Visual Basic
Public Shared Function SelectULRows(ByVal connectionString As String, _
    ByVal queryString As String, ByVal tableName As String)
    Dim connection As ULConnection = New ULConnection(connectionString)
    Dim adapter As ULDataAdapter = New ULDataAdapter()
        adapter.SelectCommand = New ULCommand(queryString, connection)
        Dim builder As ULCommandBuilder = New ULCommandBuilder(adapter)
    connection.Open()
    Dim dataSet As DataSet = New DataSet()
    adapter.Fill(dataSet, tableName)
    'Insert code to modify data in DataSet.
    'Without the ULCommandBuilder this line would fail
    adapter.Update(dataSet, tableName)
    Return dataSet
End Function
```

対応する C# 言語のコードを次に示します。

```
// C#
public static DataSet SelectULRows(string connectionString,
    string queryString, string tableName)
{
    using (ULConnection connection = new ULConnection(connectionString))
    {
        ULDataAdapter adapter = new ULDataAdapter();
        adapter.SelectCommand = new ULCommand(queryString, connection);
        ULCommandBuilder builder = new ULCommandBuilder(adapter);
        connection.Open();
        DataSet dataSet = new DataSet();
        adapter.Fill(dataSet, tableName);
        // Insert code to modify data in DataSet.
        // Without the ULCommandBuilder this line would fail
        adapter.Update(dataSet, tableName);
        return dataSet;
    }
}
```

このセクションの内容:

[ULCommandBuilder コンストラクタ \[108 ページ\]](#)

ULCommandBuilder オブジェクトを初期化します。

[ApplyParameterInfo\(DbParameter, DataRow, StatementType, bool\) メソッド \[110 ページ\]](#)

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

データベースで削除処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

[GetInsertCommand メソッド \[113 ページ\]](#)

データベースで挿入処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

[GetParameterName メソッド \[116 ページ\]](#)

[GetParameterPlaceholder\(int\) メソッド \[117 ページ\]](#)

[GetUpdateCommand メソッド \[117 ページ\]](#)

データベースで更新処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

[InitializeCommand\(DbCommand\) メソッド \[120 ページ\]](#)

[SetRowUpdatingHandler\(DbDataAdapter\) メソッド \[120 ページ\]](#)

[DataAdapter プロパティ \[121 ページ\]](#)

SQL 文が自動的に生成される ULDataAdapter オブジェクトを取得または設定します。

関連情報

[ULCommand クラス \[58 ページ\]](#)

[ULConnection クラス \[121 ページ\]](#)

[ULDataAdapter クラス \[238 ページ\]](#)

1.6.1 ULCommandBuilder コンストラクタ

ULCommandBuilder オブジェクトを初期化します。

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	ULCommandBuilder() [109 ページ]	ULCommandBuilder オブジェクトを初期化します。
public	ULCommandBuilder(ULDataAdapter) [109 ページ]	ULCommandBuilder オブジェクトを、指定された ULDataAdapter オブジェクトで初期化します。

このセクションの内容:

[ULCommandBuilder\(\) コンストラクタ \[109 ページ\]](#)

ULCommandBuilder オブジェクトを初期化します。

[ULCommandBuilder\(ULDataAdapter\) コンストラクタ \[109 ページ\]](#)

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

1.6.1.1 ULCommandBuilder() コンストラクタ

ULCommandBuilder オブジェクトを初期化します。

構文

Visual Basic

```
Public Sub ULCommandBuilder ()
```

C#

```
public ULCommandBuilder ()
```

関連情報

[ULCommandBuilder\(ULDataAdapter\) コンストラクタ \[109 ページ\]](#)

1.6.1.2 ULCommandBuilder(ULDataAdapter) コンストラクタ

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

構文

Visual Basic

```
Public Sub ULCommandBuilder (ByVal adapter As ULDataAdapter)
```

C#

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

パラメータ

adapter ULDataAdapter オブジェクト。

関連情報

[ULDataAdapter クラス \[238 ページ\]](#)

1.6.2 ApplyParameterInfo(DbParameter, DataRow, StatementType, bool) メソッド

構文

Visual Basic

```
Protected Overrides Sub ApplyParameterInfo (  
    ByVal parameter As DbParameter,  
    ByVal row As DataRow,  
    ByVal statementType As StatementType,  
    ByVal whereClause As Boolean  
)
```

C#

```
protected override void ApplyParameterInfo (  
    DbParameter parameter,  
    DataRow row,  
    StatementType statementType,  
    bool whereClause  
)
```

1.6.3 GetDeleteCommand メソッド

データベースで削除処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public new ULCommand	GetDeleteCommand() [111 ページ]	データベースで削除処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。
public new ULCommand	GetDeleteCommand(bool) [112 ページ]	データベースで削除処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

このセクションの内容:

[GetDeleteCommand\(\) メソッド \[111 ページ\]](#)

データベースで削除処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

[GetDeleteCommand\(bool\) メソッド \[112 ページ\]](#)

データベースで削除処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

1.6.3.1 GetDeleteCommand() メソッド

データベースで削除処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

構文

Visual Basic

```
Public Shadows Function GetDeleteCommand () As ULCommand
```

C#

```
public new ULCommand GetDeleteCommand ()
```

戻り値

削除の実行に必要な、自動的に生成された ULCommand オブジェクト。

例外

InvalidOperationException DbCommandBuilder.DataAdapter プロパティが初期化されていません。
DataAdapter.SelectCommand プロパティが初期化されていません。DataAdapter.SelectCommand.Connection プロパティが初期化されていません。動的 SQL の生成は、複数のベーステーブルに対してはサポートされていません。動的 SQL の生成は、重複したカラムを含む SelectCommand 値に対してはサポートされていません。DeleteCommand プロパティに対する動的 SQL の生成は、キーカラム情報を返さない SelectCommand 値に対してはサポートされていません。

備考

SQL 文が最初に生成された後で、アプリケーションが ULDataAdapter.SelectCommand 値を変更する場合は、DbCommandBuilder.RefreshSchema メソッドを明示的に呼び出す必要があります。この処理を行わないと、GetDeleteCommand メソッドは、正しくない可能性がある古い文の情報を使用し続けます。SQL 文が最初に生成されるのは、アプリケーションが DbDataAdapter.Update(System.Data.DataSet) または GetDeleteCommand メソッドを呼び出したときです。

関連情報

[ULCommand クラス \[58 ページ\]](#)

[SelectCommand プロパティ \[251 ページ\]](#)

1.6.3.2 GetDeleteCommand(bool) メソッド

データベースで削除処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

構文

Visual Basic

```
Public Shadows Function GetDeleteCommand (ByVal useColumnsForParameterNames As Boolean) As ULCommand
```

C#

```
public new ULCommand GetDeleteCommand (bool useColumnsForParameterNames)
```

パラメータ

useColumnsForParameterNames true の場合、可能であれば、カラム名に一致するパラメータ名を生成します。false の場合、@p1、@p2などを生成します。

戻り値

削除の実行に必要な、自動的に生成された ULCommand オブジェクト。

例外

InvalidOperationException DbCommandBuilder.DataAdapter プロパティが初期化されていません。DataAdapter.SelectCommand プロパティが初期化されていません。DataAdapter.SelectCommand.Connection プロパティが初期化されていません。動的 SQL の生成は、複数のベーステーブルに対してはサポートされていません。動的 SQL の生成は、重複したカラムを含む SelectCommand 値に対してはサポートされていません。DeleteCommand プロパティに対する動的 SQL の生成は、キーカラム情報を返さない SelectCommand 値に対してはサポートされていません。

備考

SQL 文が最初に生成された後で、アプリケーションが ULDataAdapter.SelectCommand 値を変更する場合は、DbCommandBuilder.RefreshSchema メソッドを明示的に呼び出す必要があります。この処理を行わないと、GetDeleteCommand メソッドは、正しくない可能性がある古い文の情報を使用し続けます。SQL 文が最初に生成されるのは、アプリケーションが DbDataAdapter.Update(System.Data.DataSet) または GetDeleteCommand メソッドを呼び出したときです。

関連情報

[ULCommand クラス \[58 ページ\]](#)

[SelectCommand プロパティ \[251 ページ\]](#)

1.6.4 GetInsertCommand メソッド

データベースで挿入処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

オーバーロードリスト

変数とタイプ	オーバーロード名	説明
public new ULCommand	GetInsertCommand() [113 ページ]	データベースで挿入処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。
public new ULCommand	GetInsertCommand(bool) [114 ページ]	データベースで挿入処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

このセクションの内容:

[GetInsertCommand\(\) メソッド \[113 ページ\]](#)

データベースで挿入処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

[GetInsertCommand\(bool\) メソッド \[114 ページ\]](#)

データベースで挿入処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

1.6.4.1 GetInsertCommand() メソッド

データベースで挿入処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

構文

Visual Basic

```
Public Shadows Function GetInsertCommand () As ULCommand
```

C#

```
public new ULCommand GetInsertCommand ()
```

戻り値

挿入の実行に必要な、自動的に生成された ULCommand オブジェクト。

例外

InvalidOperationException DbCommandBuilder.DataAdapter プロパティが初期化されていません。
DataAdapter.SelectCommand プロパティが初期化されていません。DataAdapter.SelectCommand.Connection プロパティが初期化されていません。InsertCommand プロパティに対する動的 SQL の生成は、修正可能なカラムを返さない SelectCommand 値に対してはサポートされていません。動的 SQL の生成は、複数のベーステーブルに対してはサポートされていません。動的 SQL の生成は、重複したカラムを含む SelectCommand 値に対してはサポートされていません。

備考

SQL 文が最初に生成された後で、アプリケーションが ULDataAdapter.SelectCommand 値を変更する場合は、DbCommandBuilder.RefreshSchema メソッドを明示的に呼び出す必要があります。この処理を行わないと、GetInsertCommand メソッドは、正しくない可能性がある古い文の情報を使用し続けます。SQL 文が最初に生成されるのは、アプリケーションが DbDataAdapter.Update(System.Data.DataSet) メソッドまたは GetInsertCommand メソッドを呼び出したときです。

関連情報

[ULCommand クラス \[58 ページ\]](#)

[SelectCommand プロパティ \[251 ページ\]](#)

1.6.4.2 GetInsertCommand(bool) メソッド

データベースで挿入処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

構文

Visual Basic

```
Public Shadows Function GetInsertCommand (ByVal useColumnsForParameterNames As Boolean) As ULCommand
```

C#

```
public new ULCommand GetInsertCommand (bool useColumnsForParameterNames)
```

パラメータ

`useColumnsForParameterNames` true の場合、可能であれば、カラム名に一致するパラメータ名を生成します。false の場合、@p1、@p2などを生成します。

戻り値

挿入の実行に必要な、自動的に生成された `ULCommand` オブジェクト。

例外

`InvalidOperationException` `DbCommandBuilder.DataAdapter` プロパティが初期化されていません。
`DataAdapter.SelectCommand` プロパティが初期化されていません。`DataAdapter.SelectCommand.Connection` プロパティが初期化されていません。`InsertCommand` プロパティに対する動的 SQL の生成は、修正可能なカラムを返さない `SelectCommand` 値に対してはサポートされていません。動的 SQL の生成は、複数のペーステーブルに対してはサポートされていません。動的 SQL の生成は、重複したカラムを含む `SelectCommand` 値に対してはサポートされていません。

備考

SQL 文が最初に生成された後で、アプリケーションが `ULDataAdapter.SelectCommand` 値を変更する場合は、`DbCommandBuilder.RefreshSchema` メソッドを明示的に呼び出す必要があります。この処理を行わないと、`GetInsertCommand` メソッドは、正しくない可能性がある古い文の情報を使用し続けます。SQL 文が最初に生成されるのは、アプリケーションが `DbDataAdapter.Update(System.Data.DataSet)` メソッドまたは `GetInsertCommand` メソッドを呼び出したときです。

関連情報

[ULCommand クラス \[58 ページ\]](#)

[SelectCommand プロパティ \[251 ページ\]](#)

1.6.5 GetParameterName メソッド

オーバードリスト

変更子とタイプ	オーバーロード名	説明
protected override string	GetParameterName(int) [116 ページ]	
protected override string	GetParameterName(string) [116 ページ]	

このセクションの内容:

[GetParameterName\(int\) メソッド \[116 ページ\]](#)

[GetParameterName\(string\) メソッド \[116 ページ\]](#)

1.6.5.1 GetParameterName(int) メソッド

構文

Visual Basic

```
Protected Overrides Function GetParameterName (ByVal index As Integer) As String
```

C#

```
protected override string GetParameterName (int index)
```

1.6.5.2 GetParameterName(string) メソッド

構文

Visual Basic

```
Protected Overrides Function GetParameterName (ByVal parameterName As String) As String
```

C#

```
protected override string GetParameterName (string parameterName)
```

1.6.6 GetParameterPlaceholder(int) メソッド

構文

Visual Basic

```
Protected Overrides Function GetParameterPlaceholder (ByVal index As Integer)  
As String
```

C#

```
protected override string GetParameterPlaceholder (int index)
```

1.6.7 GetUpdateCommand メソッド

データベースで更新処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

オーバーロードリスト

変数とタイプ	オーバーロード名	説明
public new ULCommand	GetUpdateCommand() [118 ページ]	データベースで更新処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。
public new ULCommand	GetUpdateCommand(bool) [119 ページ]	データベースで更新処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

このセクションの内容:

[GetUpdateCommand\(\) メソッド \[118 ページ\]](#)

データベースで更新処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

[GetUpdateCommand\(bool\) メソッド \[119 ページ\]](#)

データベースで更新処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

1.6.7.1 GetUpdateCommand() メソッド

データベースで更新処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

構文

Visual Basic

```
Public Shadows Function GetUpdateCommand () As ULCommand
```

C#

```
public new ULCommand GetUpdateCommand ()
```

戻り値

更新の実行に必要な、自動的に生成された ULCommand オブジェクト。

例外

InvalidOperationException DbCommandBuilder.DataAdapter プロパティが初期化されていません。

DataAdapter.SelectCommand プロパティが初期化されていません。DataAdapter.SelectCommand.Connection プロパティが初期化されていません。UpdateCommand プロパティに対する動的 SQL の生成は、修正可能なカラムを返さない SelectCommand 値に対してはサポートされていません。動的 SQL の生成は、複数のベーステーブルに対してはサポートされていません。動的 SQL の生成は、重複したカラムを含む SelectCommand 値に対してはサポートされていません。UpdateCommand プロパティに対する動的 SQL の生成は、キーカラム情報を返さない SelectCommand 値に対してはサポートされていません。

備考

SQL 文が最初に生成された後で、アプリケーションが ULDataAdapter.SelectCommand 値を変更する場合は、DbCommandBuilder.RefreshSchema メソッドを明示的に呼び出す必要があります。この処理を行わないと、GetUpdateCommand メソッドは、正しくない可能性がある古い文の情報を使用し続けます。SQL 文が最初に生成されるのは、アプリケーションが DbDataAdapter.Update(System.Data.DataSet) または GetUpdateCommand メソッドを呼び出したときです。

関連情報

[ULCommand クラス \[58 ページ\]](#)

1.6.7.2 GetUpdateCommand(bool) メソッド

データベースで更新処理を実行するのに必要な、自動的に生成された ULCommand オブジェクトを取得します。

構文

Visual Basic

```
Public Shadows Function GetUpdateCommand (ByVal useColumnsForParameterNames As Boolean) As ULCommand
```

C#

```
public new ULCommand GetUpdateCommand (bool useColumnsForParameterNames)
```

パラメータ

useColumnsForParameterNames true の場合、可能であれば、カラム名に一致するパラメータ名を生成します。false の場合、@p1、@p2などを生成します。

戻り値

更新の実行に必要な、自動的に生成された ULCommand オブジェクト。

例外

InvalidOperationException DbCommandBuilder.DataAdapter プロパティが初期化されていません。DataAdapter.SelectCommand プロパティが初期化されていません。DataAdapter.SelectCommand.Connection プロパティが初期化されていません。UpdateCommand プロパティに対する動的 SQL の生成は、修正可能なカラムを返さない SelectCommand 値に対してはサポートされていません。動的 SQL の生成は、複数のベーステーブルに対してはサポートされていません。動的 SQL の生成は、重複したカラムを含む SelectCommand 値に対してはサポートされていません。UpdateCommand プロパティに対する動的 SQL の生成は、キーカラム情報を返さない SelectCommand 値に対してはサポートされていません。

備考

SQL 文が最初に生成された後で、アプリケーションが `ULDataAdapter.SelectCommand` を変更する場合、`DbCommandBuilder.RefreshSchema` を明示的に呼び出す必要があります。この処理を行わないと、`GetUpdateCommand` メソッドは、正しくない可能性がある古い文の情報を使用し続けます。SQL 文が最初に生成されるのは、アプリケーションが `DbDataAdapter.Update(System.Data.DataSet)` または `GetUpdateCommand` メソッドを呼び出したときです。

関連情報

[ULCommand クラス \[58 ページ\]](#)

[SelectCommand プロパティ \[251 ページ\]](#)

1.6.8 InitializeCommand(DbCommand) メソッド

構文

Visual Basic

```
Protected Overrides Function InitializeCommand (ByVal command As DbCommand) As DbCommand
```

C#

```
protected override DbCommand InitializeCommand (DbCommand command)
```

1.6.9 SetRowUpdatingHandler(DbDataAdapter) メソッド

構文

Visual Basic

```
Protected Overrides Sub SetRowUpdatingHandler (ByVal adapter As DbDataAdapter)
```

C#

```
protected override void SetRowUpdatingHandler (DbDataAdapter adapter)
```


1.6.10 DataAdapter プロパティ

SQL 文が自動的に生成される ULDataAdapter オブジェクトを取得または設定します。

構文

Visual Basic

```
Public Shadows Property DataAdapter As ULDataAdapter
```

C#

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

備考

ULDataAdapter オブジェクト。

関連情報

[ULDataAdapter クラス \[238 ページ\]](#)

1.7 ULConnection クラス

Ultra Light.NET データベースへの接続を表します。

構文

Visual Basic

```
Public NotInheritable Class ULConnection Inherits  
System.Data.Common.DbConnection
```

C#

```
public sealed class ULConnection : System.Data.Common.DbConnection
```

メンバー

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

変数

変更子とタイプ	変数	説明
public const long	INVALID_DATABASE_ID	UL 拡張: ULConnection.DatabaseID プロパティが設定されていないことを示すデータベース ID 定数。
public const String	SYNC_ALL_DB	パブリケーションリストが空の場合、データベース全体を表します。
public const String	SYNC_ALL_PUBS	パブリケーション名 "*" はすべてのパブリケーションを表します。

コンストラクタ

変更子とタイプ	コンストラクタ	説明
public	ULConnection [128 ページ]	ULConnection オブジェクトを初期化します。

メソッド

変更子とタイプ	メソッド	説明
protected override DbTransaction	BeginDbTransaction(IsolationLevel) [131 ページ]	
public IAsyncResult	BeginSynchronize [131 ページ]	UL 拡張: 現在の SyncParms オブジェクトを使用して同期を非同期に起動します。
public new ULTransaction	BeginTransaction [134 ページ]	トランザクションオブジェクトを返します。
public unsafe int	CancelGetNotification(string) [137 ページ]	UL 拡張: 指定された名前に一致する、すべてのキューに登録されている保留中の取得通知コールをキャンセルします。
public void	CancelSynchronize(IAsyncResult) [138 ページ]	UL 拡張: 実行中の同期を次回キャンセルさせます。
public override void	ChangeDatabase(string) [139 ページ]	開いている接続の現在のデータベースを変更します。
public void	ChangeEncryptionKey(string) [140 ページ]	UL 拡張: データベースの暗号化キーを、指定された新しいキーに変更します。
public static void	ChangePassword(string, string) [141 ページ]	接続文字列に示されるユーザのパスワードを、指定される新しいパスワードに変更します。
public override void	Close() [141 ページ]	データベース接続を閉じます。
public unsafe long	CountUploadRows(string, long) [142 ページ]	UL 拡張: 次回の同期でアップロードする必要のあるロー数を返します。
public new ULCommand	CreateCommand() [143 ページ]	この接続と現在のトランザクションに関連付けられる ULCommand オブジェクトを作成して初期化します。
protected override DbCommand	CreateDbCommand() [144 ページ]	

変数とタイプ	メソッド	説明
public void	CreateNotificationQueue(string, string) [144 ページ]	UL 拡張: イベントキューを作成します。
public void	DeclareEvent(string) [145 ページ]	UL 拡張: 指定したイベントを宣言します。
public void	DestroyNotificationQueue(string) [146 ページ]	UL 拡張: イベントキューを破棄します。
protected override unsafe void	Dispose(bool) [147 ページ]	ULCommand オブジェクトによって使用されるアンマネージリソースを解放するほか、必要に応じてマネージリソースを解放します。
public void	EndSynchronize(IAsyncResult) [147 ページ]	UL 拡張: 非同期で起動された同期が終了するまでブロックします。
public ULTable	ExecuteTable [148 ページ]	UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。
public unsafe DateTime	GetLastDownloadTime(string) [154 ページ]	UL 拡張: 指定されたパブリケーションの最後のダウンロードの時刻を返します。
public unsafe Guid	GetNewUUID() [155 ページ]	UL 拡張: 新しい UUID (System.Guid) を生成します。
public string	GetNotification(string, int) [156 ページ]	UL 拡張: 通知またはタイムアウトをブロックします。
public string	GetNotificationParameter(string, string) [157 ページ]	UL 拡張: GetNotification メソッドが読み取ったばかりのイベントのパラメータの値を取得します。
public override DataTable	GetSchema [158 ページ]	サポートされているスキーマコレクションのリストを返します。
public void	GrantConnectTo(string, string) [162 ページ]	UL 拡張: 指定されたパスワードを持つユーザ ID に、Ultra Light データベースへのアクセスを許可します。
public override void	Open() [163 ページ]	以前に指定された接続文字列を使用してデータベースへの接続を開きます。
public void	RegisterForEvent(string, string, string, bool) [164 ページ]	UL 拡張: オブジェクトからイベントを取得するためのキューを登録します。
public void	ResetLastDownloadTime(string) [165 ページ]	UL 拡張: 最後のダウンロードの時刻をリセットします。
public void	RevokeConnectFrom(string) [165 ページ]	UL 拡張: 指定されたユーザ ID から Ultra Light データベースへのアクセス権を取り消します。
public void	RollbackPartialDownload() [166 ページ]	UL 拡張: 部分的なダウンロードからの、データベースへの未処理の変更をロールバックします。
public unsafe int	SendNotification(string, string, string) [167 ページ]	UL 拡張: 一致するキューに通知を送信します。
public void	SetSyncListener(ULSyncProgressListener) [168 ページ]	同期メッセージの処理に使用するリスナオブジェクトを指定します。

変更子とタイプ	メソッド	説明
public void	StartSynchronizationDelete() [169 ページ]	UL 拡張: 同期用に、この接続によって行われる以後のすべての削除にマークを付けます。
public void	StopSynchronizationDelete() [170 ページ]	UL 拡張: 削除操作が同期されないようにします。
public void	Synchronize [171 ページ]	UL 拡張: 現在の ULConnection.SyncParms オブジェクトを使用してデータベースを同期します。
public unsafe int	TriggerEvent(string, string) [173 ページ]	UL 拡張: イベントをトリガします。
public void	ValidateDatabase [174 ページ]	UL 拡張: 現在のデータベースの検証を実行します。

プロパティ

変更子とタイプ	プロパティ	説明
public override string	ConnectionString [177 ページ]	Ultra Light.NET データベースへの接続を開くためのパラメータを指定します。
public override int	ConnectionTimeout [178 ページ]	この機能は Ultra Light.NET ではサポートされていません。
public override string	Database [179 ページ]	接続を開くデータベースの名前を返します。
public unsafe long	DatabaseID [179 ページ]	UL 拡張: グローバルオートインクリメントカラムに使用するデータベース ID の値を指定します。
public override string	DataSource [180 ページ]	この機能は Ultra Light.NET ではサポートされていません。
public unsafe short	GlobalAutoIncrementUsage [180 ページ]	UL 拡張: 利用可能なグローバルオートインクリメントの値の使用済み比率 (%) を返します。
public unsafe ulong	LastIdentity [181 ページ]	UL 拡張: 直前に使用した identity の値を返します。
public ULDatabaseSchema	Schema [182 ページ]	UL 拡張: この接続に関連付けられている現在のデータベースのスキーマへのアクセスに使用します。
public override string	ServerVersion [182 ページ]	この機能は Ultra Light.NET ではサポートされていません。
public override ConnectionState	State [183 ページ]	接続の現在のステータスを返します。
public ULSyncParms	SyncParms [183 ページ]	UL 拡張: この接続の同期設定を指定します。
public ULSyncResult	SyncResult [184 ページ]	UL 拡張: この接続の最後の同期結果を返します。

イベント

変数とタイプ	イベント	説明
public ULInfoMessageEventHandler	InfoMessage [184 ページ]	Ultra Light.NET が、この接続の警告または情報メッセージを送信するときに発生します。
public override StateChangeEventHandler	StateChange [186 ページ]	この接続のステータスが変更されると発生します。

備考

Ultra Light.NET の Ultra Light エンジンのランタイムを使用するには、ULDatabaseManager.RuntimeType プロパティを適切な値に設定してから、他の Ultra Light.NET API を使用します。

既存のデータベースへの接続は、ULConnection.Open メソッドを使用して開かれます。

接続は、他の操作の実行前に開きます。また、その接続での操作がすべて終了したら、接続を閉じてからアプリケーションを終了します。また、接続で開いた結果セットとテーブルをすべて閉じてから、その接続を閉じます。

データベースのスキーマには、開いている接続の ULConnection.Schema 値を使用してアクセスできます。

このセクションの内容:

[ULConnection コンストラクタ \[128 ページ\]](#)

ULConnection オブジェクトを初期化します。

[BeginDbTransaction\(IsolationLevel\) メソッド \[131 ページ\]](#)

[BeginSynchronize メソッド \[131 ページ\]](#)

UL 拡張: 現在の SyncParms オブジェクトを使用して同期を非同期に起動します。

[BeginTransaction メソッド \[134 ページ\]](#)

トランザクションオブジェクトを返します。

[CancelGetNotification\(string\) メソッド \[137 ページ\]](#)

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

[CancelSynchronize\(IAsyncResult\) メソッド \[138 ページ\]](#)

UL 拡張: 実行中の同期を次回キャンセルさせます。

[ChangeDatabase\(string\) メソッド \[139 ページ\]](#)

開いている接続の現在のデータベースを変更します。

[ChangeEncryptionKey\(string\) メソッド \[140 ページ\]](#)

UL 拡張: データベースの暗号化キーを、指定された新しいキーに変更します。

[ChangePassword\(string, string\) メソッド \[141 ページ\]](#)

接続文字列に示されるユーザのパスワードを、指定される新しいパスワードに変更します。

[Close\(\) メソッド \[141 ページ\]](#)

データベース接続を閉じます。

[CountUploadRows\(string, long\) メソッド \[142 ページ\]](#)

UL 拡張: 次の同期でアップロードする必要のあるロー数を返します。

[CreateCommand\(\) メソッド \[143 ページ\]](#)

この接続と現在のトランザクションに関連付けられる ULCommand オブジェクトを作成して初期化します。

[CreateDbCommand\(\) メソッド \[144 ページ\]](#)

[CreateNotificationQueue\(string, string\) メソッド \[144 ページ\]](#)

UL 拡張: イベントキューを作成します。

[DeclareEvent\(string\) メソッド \[145 ページ\]](#)

UL 拡張: 指定したイベントを宣言します。

[DestroyNotificationQueue\(string\) メソッド \[146 ページ\]](#)

UL 拡張: イベントキューを破棄します。

[Dispose\(bool\) メソッド \[147 ページ\]](#)

ULCommand オブジェクトによって使用されるアンマネージリソースを解放するほか、必要に応じてマネージリソースを解放します。

[EndSynchronize\(IAsyncResult\) メソッド \[147 ページ\]](#)

UL 拡張: 非同期で起動された同期が終了するまでブロックします。

[ExecuteTable メソッド \[148 ページ\]](#)

UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。

[GetLastDownloadTime\(string\) メソッド \[154 ページ\]](#)

UL 拡張: 指定されたパブリケーションの最後のダウンロードの時刻を返します。

[GetNewUUID\(\) メソッド \[155 ページ\]](#)

UL 拡張: 新しい UUID (System.Guid) を生成します。

[GetNotification\(string, int\) メソッド \[156 ページ\]](#)

UL 拡張: 通知またはタイムアウトをブロックします。

[GetNotificationParameter\(string, string\) メソッド \[157 ページ\]](#)

UL 拡張: GetNotification メソッドが読み取ったばかりのイベントのパラメータの値を取得します。

[GetSchema メソッド \[158 ページ\]](#)

サポートされているスキーマコレクションのリストを返します。

[GrantConnectTo\(string, string\) メソッド \[162 ページ\]](#)

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

[Open\(\) メソッド \[163 ページ\]](#)

以前に指定された接続文字列を使用してデータベースへの接続を開きます。

[RegisterForEvent\(string, string, string, bool\) メソッド \[164 ページ\]](#)

UL 拡張: オブジェクトからイベントを取得するためのキューを登録します。

[ResetLastDownloadTime\(string\) メソッド \[165 ページ\]](#)

UL 拡張: 最後のダウンロードの時刻をリセットします。

[RevokeConnectFrom\(string\) メソッド \[165 ページ\]](#)

UL 拡張: 指定されたユーザ ID から Ultra Light データベースへのアクセス権を取り消します。

[RollbackPartialDownload\(\) メソッド \[166 ページ\]](#)

UL 拡張: 部分的なダウンロードからの、データベースへの未処理の変更をロールバックします。

[SendNotification\(string, string, string\) メソッド \[167 ページ\]](#)

UL 拡張: 一致するキューに通知を送信します。

[SetSyncListener\(ULSyncProgressListener\) メソッド \[168 ページ\]](#)

同期メッセージの処理に使用するリスナオブジェクトを指定します。

[StartSynchronizationDelete\(\) メソッド \[169 ページ\]](#)

UL 拡張: 同期用に、この接続によって行われる以後のすべての削除にマークを付けます。

[StopSynchronizationDelete\(\) メソッド \[170 ページ\]](#)

UL 拡張: 削除操作が同期されないようにします。

[Synchronize メソッド \[171 ページ\]](#)

UL 拡張: 現在の ULConnection.SyncParms オブジェクトを使用してデータベースを同期します。

[TriggerEvent\(string, string\) メソッド \[173 ページ\]](#)

UL 拡張: イベントをトリガします。

[ValidateDatabase メソッド \[174 ページ\]](#)

UL 拡張: 現在のデータベースの検証を実行します。

[ConnectionString プロパティ \[177 ページ\]](#)

Ultra Light.NET データベースへの接続を開くためのパラメータを指定します。

[ConnectionTimeout プロパティ \[178 ページ\]](#)

この機能は Ultra Light.NET ではサポートされていません。

[Database プロパティ \[179 ページ\]](#)

接続を開くデータベースの名前を返します。

[DatabaseID プロパティ \[179 ページ\]](#)

UL 拡張: グローバルオートインクリメントカラムに使用するデータベース ID の値を指定します。

[DataSource プロパティ \[180 ページ\]](#)

この機能は Ultra Light.NET ではサポートされていません。

[GlobalAutoIncrementUsage プロパティ \[180 ページ\]](#)

UL 拡張: 利用可能なグローバルオートインクリメントの値の使用済み比率 (%) を返します。

[LastIdentity プロパティ \[181 ページ\]](#)

UL 拡張: 直前に使用した identity の値を返します。

[Schema プロパティ \[182 ページ\]](#)

UL 拡張: この接続に関連付けられている現在のデータベースのスキーマへのアクセスに使用します。

[ServerVersion プロパティ \[182 ページ\]](#)

この機能は Ultra Light.NET ではサポートされていません。

[State プロパティ \[183 ページ\]](#)

接続の現在のステータスを返します。

[SyncParms プロパティ \[183 ページ\]](#)

UL 拡張: この接続の同期設定を指定します。

[SyncResult プロパティ \[184 ページ\]](#)

UL 拡張: この接続の最後の同期結果を返します。

[InfoMessage イベント \[184 ページ\]](#)

Ultra Light.NET が、この接続の警告または情報メッセージを送信するときに発生します。

[StateChange イベント \[186 ページ\]](#)

この接続のステータスが変更されると発生します。

関連情報

[Open\(\) メソッド \[163 ページ\]](#)

[Schema プロパティ \[182 ページ\]](#)

1.7.1 ULConnection コンストラクタ

ULConnection オブジェクトを初期化します。

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	ULConnection() [128 ページ]	ULConnection オブジェクトを初期化します。
public	ULConnection(string) [129 ページ]	ULConnection オブジェクトを、指定された接続文字列で初期化します。

このセクションの内容:

[ULConnection\(\) コンストラクタ \[128 ページ\]](#)

ULConnection オブジェクトを初期化します。

[ULConnection\(string\) コンストラクタ \[129 ページ\]](#)

ULConnection オブジェクトを、指定された接続文字列で初期化します。

1.7.1.1 ULConnection() コンストラクタ

ULConnection オブジェクトを初期化します。

構文

Visual Basic

```
Public Sub ULConnection ()
```


C#

```
public ULConnection ()
```

備考

接続を開いてから、データベース操作を実行してください。

Ultra Light.NET の Ultra Light エンジンのランタイムを使用するには、ULDatabaseManager.RuntimeType プロパティを適切な値に設定してから、他の Ultra Light.NET API を使用します。

ULConnection オブジェクトは、ULConnection.ConnectionString プロパティが設定されていないと、開くことができません。

関連情報

[Open\(\) メソッド \[163 ページ\]](#)

[ConnectionString プロパティ \[177 ページ\]](#)

1.7.1.2 ULConnection(string) コンストラクタ

ULConnection オブジェクトを、指定された接続文字列で初期化します。

構文

Visual Basic

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

C#

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

パラメータ

connectionString Ultra Light.NET の接続文字列。接続文字列は keyword=value のペアがセミコロンで区切られたリストです。

例外

ArgumentException 指定された接続文字列が無効です。

備考

接続を開いてから、データベース操作を実行してください。

Ultra Light.NET の Ultra Light エンジンのランタイムを使用するには、ULDatabaseManager.RuntimeType プロパティを適切な値に設定してから、他の Ultra Light.NET API を使用します。

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

例

次のコードでは、Windows Mobile デバイス上のデータベース ¥UltraLite¥MyDatabase.udb への接続を作成して開きます。

```
' Visual Basic
Dim openParms As ULConnectionParms = New ULConnectionParms
openParms.DatabaseOnDevice = "¥UltraLite¥MyDatabase.udb"
Dim conn As ULConnection = _
    New ULConnection( openParms.ToString() )
conn.Open()
```

対応する C# 言語のコードを次に示します。

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

関連情報

[Open\(\) メソッド \[163 ページ\]](#)

[ULConnection\(\) コンストラクタ \[128 ページ\]](#)

[ULConnectionParms クラス \[187 ページ\]](#)

[ConnectionString プロパティ \[177 ページ\]](#)

1.7.2 BeginDbTransaction(IsolationLevel) メソッド

構文

Visual Basic

```
Protected Overrides Function BeginDbTransaction (ByVal isolationLevel As IsolationLevel) As DbTransaction
```

C#

```
protected override DbTransaction BeginDbTransaction (IsolationLevel isolationLevel)
```

1.7.3 BeginSynchronize メソッド

UL 拡張: 現在の SyncParms オブジェクトを使用して同期を非同期に起動します。

オーバードリスト

変更子とタイプ	オーバード名	説明
public IAsyncResult	BeginSynchronize() [132 ページ]	UL 拡張: 現在の SyncParms オブジェクトを使用して同期を非同期に起動します。
public IAsyncResult	BeginSynchronize(Control, ULSyncProgressedDlg, object) [133 ページ]	UL 拡張: 現在の SyncParms を使用して同期を非同期に起動します。

このセクションの内容:

[BeginSynchronize\(\) メソッド \[132 ページ\]](#)

UL 拡張: 現在の SyncParms オブジェクトを使用して同期を非同期に起動します。

[BeginSynchronize\(Control, ULSyncProgressedDlg, object\) メソッド \[133 ページ\]](#)

UL 拡張: 現在の SyncParms を使用して同期を非同期に起動します。

1.7.3.1 BeginSynchronize() メソッド

UL 拡張: 現在の SyncParms オブジェクトを使用して同期を非同期に起動します。

構文

Visual Basic

```
Public Function BeginSynchronize () As IAsyncResult
```

C#

```
public IAsyncResult BeginSynchronize ()
```

戻り値

同期が完了したか、同期が終了するまでブロックされるかを判断するために使用できる IAsyncResult オブジェクト。

例外

ULException class SQL エラーが発生しました。

備考

このメソッドは、同期化を行う新しいスレッドを作成して、すぐに返します。EndSynchronize メソッドを呼び出して、同期が完了するまでブロックしてください。

関連情報

[BeginSynchronize\(Control, ULSyncProgressedDlg, object\) メソッド \[133 ページ\]](#)

[EndSynchronize\(IAsyncResult\) メソッド \[147 ページ\]](#)

[CancelSynchronize\(IAsyncResult\) メソッド \[138 ページ\]](#)

[ULSyncProgressedDlg\(IAsyncResult, ULSyncProgressData\) デリゲート \[583 ページ\]](#)

1.7.3.2 BeginSynchronize(Control, ULSyncProgressedDlg, object) メソッド

UL 拡張: 現在の SyncParms を使用して同期を非同期に起動します。

構文

Visual Basic

```
Public Function BeginSynchronize (  
    ByVal control As Control,  
    ByVal dlg As ULSyncProgressedDlg,  
    ByVal state As Object  
) As IAsyncResult
```

C#

```
public IAsyncResult BeginSynchronize (  
    Control control,  
    ULSyncProgressedDlg dlg,  
    object state  
)
```

パラメータ

control 同期スレッドが ULSyncProgressedDlg 呼び出しを起動するために使用する System.Windows.Forms.Control オブジェクト。

dlg 定期的起動されて同期の進行状況を更新する ULSyncProgressedDlg メソッド。

state このユーザコンテキストは、IAsyncResult.AsyncState を使用する ULSyncProgressedDlg メソッドでアクセスできます。

戻り値

同期が完了したか、同期が終了するまでブロックされるかを判断するために使用できる IAsyncResult オブジェクト。

例外

ULException class SQL エラーが発生しました。

備考

このメソッドは、同期を行った後すぐに戻る新しいメソッドを作成し、同期の進行状況を更新するために提供された `ULSyncProgressedDlg` メソッドを定期的に起動します。同期が完了するまでブロックするように `EndSynchronize` を呼び出してください。

関連情報

[BeginSynchronize\(\) メソッド \[132 ページ\]](#)

[EndSynchronize\(IAsyncResult\) メソッド \[147 ページ\]](#)

[CancelSynchronize\(IAsyncResult\) メソッド \[138 ページ\]](#)

[ULSyncProgressedDlg\(IAsyncResult, ULSyncProgressData\) デリゲート \[583 ページ\]](#)

1.7.4 BeginTransaction メソッド

トランザクションオブジェクトを返します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public new ULTransaction	BeginTransaction() [135 ページ]	トランザクションオブジェクトを返します。
public new ULTransaction	BeginTransaction(IsolationLevel) [136 ページ]	指定された独立性レベルのトランザクションオブジェクトを返します。

このセクションの内容:

[BeginTransaction\(\) メソッド \[135 ページ\]](#)

トランザクションオブジェクトを返します。

[BeginTransaction\(IsolationLevel\) メソッド \[136 ページ\]](#)

指定された独立性レベルのトランザクションオブジェクトを返します。

1.7.4.1 BeginTransaction() メソッド

トランザクションオブジェクトを返します。

構文

Visual Basic

```
Public Shadows Function BeginTransaction () As ULTransaction
```

C#

```
public new ULTransaction BeginTransaction ()
```

戻り値

新しいトランザクションを表す ULTransaction オブジェクト。

例外

ULException class 接続が閉じています。

InvalidOperationException ULConnection クラスは並列トランザクションをサポートしていません。

備考

トランザクションオブジェクトに関連付けられているコマンドは、単一のトランザクションとして実行されます。トランザクションは、ULTransaction.Commit メソッドまたは ULTransaction.Rollback メソッドで終了します。

トランザクションは IsolationLevel.ReadCommitted 値で作成されます。

コマンドをトランザクションオブジェクトに関連付けるには、ULCommand.Transaction プロパティを使用します。現在のトランザクションは、ULConnection.CreateCommand メソッドによって作成されたコマンドに自動的に関連付けられます。

デフォルトでは、接続はトランザクションを使用しません。また、すべてのコマンドは、実行されるときに自動的にコミットされます。現在のトランザクションがコミットまたはロールバックされると、次に BeginTransaction メソッドを呼び出すまで、接続はオートコミットモードおよび以前の独立性レベルに戻ります。

Ultra Light での独立性レベルの定義は、ADO.NET のマニュアルの IsolationLevel の説明とは若干異なります。

これは、System.Data.IDbConnection.BeginTransaction メソッドと

System.Data.Common.DbConnection.BeginTransaction メソッドが厳密に型指定されたものです。

関連情報

[Commit\(\) メソッド \[577 ページ\]](#)

[Rollback\(\) メソッド \[578 ページ\]](#)

[Transaction プロパティ \[104 ページ\]](#)

[CreateCommand\(\) メソッド \[143 ページ\]](#)

1.7.4.2 BeginTransaction(IsolationLevel) メソッド

指定された独立性レベルのトランザクションオブジェクトを返します。

構文

Visual Basic

```
Public Shadows Function BeginTransaction (ByVal isolationLevel As IsolationLevel) As ULTransaction
```

C#

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

パラメータ

isolationLevel 要求されたトランザクションの独立性レベル。Ultra Light.NET では、System.Data.IsolationLevel.ReadUncommitted および ReadCommitted 値のみがサポートされています。

戻り値

新しいトランザクションを表す ULTransaction オブジェクト。

例外

ULException class 接続が閉じているか、サポートされていない独立性レベルが指定されました。
InvalidOperationException ULConnection クラスは並列トランザクションをサポートしていません。

備考

トランザクションオブジェクトに関連付けられているコマンドは、単一のトランザクションとして実行されます。トランザクションは、ULTransaction.Commit メソッドまたは ULTransaction.Rollback メソッドで終了します。

コマンドをトランザクションオブジェクトに関連付けるには、ULCommand.Transaction プロパティを使用します。現在のトランザクションは、ULConnection.CreateCommand メソッドによって作成されたコマンドに自動的に関連付けられます。

デフォルトでは、接続はトランザクションを使用しません。また、すべてのコマンドは、実行されるときに自動的にコミットされます。現在のトランザクションがコミットまたはロールバックされると、次に BeginTransaction メソッドを呼び出すまで、接続はオートコミットモードおよび以前の独立性レベルに戻ります。

Ultra Light での独立性レベルの定義は、ADO.NET のマニュアルの IsolationLevel の説明とは若干異なります。

これは、System.Data.IDbConnection.BeginTransaction(System.Data.IsolationLevel) メソッドと System.Data.Common.DbConnection.BeginTransaction(System.Data.IsolationLevel) メソッドが厳密に型指定されたものです。

関連情報

[ULTransaction クラス \[575 ページ\]](#)

[BeginTransaction\(\) メソッド \[135 ページ\]](#)

[Commit\(\) メソッド \[577 ページ\]](#)

[Rollback\(\) メソッド \[578 ページ\]](#)

[Transaction プロパティ \[104 ページ\]](#)

[CreateCommand\(\) メソッド \[143 ページ\]](#)

1.7.5 CancelGetNotification(string) メソッド

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

構文

Visual Basic

```
Public Function CancelGetNotification (ByVal queueName As String) As Integer
```

C#

```
public unsafe int CancelGetNotification (string queueName)
```

パラメータ

`queueName` キュー名と一致する式。

戻り値

影響を受けるキューの数 (ブロックされた読み込み数とは異なります)。

例外

`ULException` class SQL エラーが発生しました。

備考

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

関連情報

[GetNotification\(string, int\) メソッド \[156 ページ\]](#)

[ULException クラス \[328 ページ\]](#)

1.7.6 CancelSynchronize(IAsyncResult) メソッド

UL 拡張: 実行中の同期を次回キャンセルさせます。

構文

Visual Basic

```
Public Sub CancelSynchronize (ByVal asyncResult As IAsyncResult)
```

C#

```
public void CancelSynchronize (IAsyncResult asyncResult)
```

パラメータ

asyncResult BeginSynchronize メソッドから返された IAsyncResult。

備考

このメソッドは、同期化スレッドに終了を伝えて、すぐに戻ります。EndSynchronize メソッドを呼び出して、同期が正常に終了するまでブロックしてください。

関連情報

[BeginSynchronize\(\) メソッド \[132 ページ\]](#)

[BeginSynchronize\(Control, ULSyncProgressedDlg, object\) メソッド \[133 ページ\]](#)

[EndSynchronize\(IAsyncResult\) メソッド \[147 ページ\]](#)

1.7.7 ChangeDatabase(string) メソッド

開いている接続の現在のデータベースを変更します。

構文

Visual Basic

```
Public Overrides Sub ChangeDatabase (ByVal connectionString As String)
```

C#

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

パラメータ

connectionString 新しいデータベースへの接続を開く完全な接続文字列。

備考

パラメータのエラーがあった場合でも、現在のデータベースへの接続は閉じられません。

UL 拡張: `connectionString` は、dbn 値でも dbf 値でもなく、完全な接続文字列です。

関連情報

[ConnectionString プロパティ \[177 ページ\]](#)

1.7.8 ChangeEncryptionKey(string) メソッド

UL 拡張: データベースの暗号化キーを、指定された新しいキーに変更します。

構文

Visual Basic

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

C#

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

パラメータ

newKey データベースの新しい暗号化キー。

例外

ULException class SQL エラーが発生しました。

備考

暗号化キーが失われた場合は、データベースを開くことができません。

関連情報

[EncryptionKey プロパティ \[195 ページ\]](#)

1.7.9 ChangePassword(string, string) メソッド

接続文字列に示されるユーザのパスワードを、指定される新しいパスワードに変更します。

構文

Visual Basic

```
Public Shared Sub ChangePassword (  
    ByVal connectionString As String,  
    ByVal newPassword As String  
)
```

C#

```
public static void ChangePassword (  
    string connectionString,  
    string newPassword  
)
```

パラメータ

connectionString 目的のデータベースに接続できるだけの情報が含まれる接続文字列。接続文字列には、ユーザ ID と現在のパスワードが含まれる可能性があります。

newPassword 設定する新しいパスワード。

例外

ArgumentNullException connectionString または newPassword parameter のいずれかが null です。

ArgumentException 接続文字列に、統合化セキュリティを使用するオプションが含まれています。

ULException class データベースを開くときに SQL エラーが発生しました。

1.7.10 Close() メソッド

データベース接続を閉じます。

構文

Visual Basic

```
Public Overrides Sub Close ()
```

C#

```
public override void Close ()
```

例外

ULException class SQL エラーが発生しました。

備考

Close メソッドは、保留中のトランザクションをロールバックし、その接続を閉じます。アプリケーションは、このメソッドを複数回呼び出すことができます。

1.7.11 CountUploadRows(string, long) メソッド

UL 拡張: 次回の同期でアップロードする必要のあるロー数を返します。

構文

Visual Basic

```
Public Function CountUploadRows (  
    ByVal pubs As String,  
    ByVal threshold As Long  
) As Long
```

C#

```
public unsafe long CountUploadRows (  
    string pubs,  
    long threshold  
)
```

パラメータ

pubs ローをチェックするパブリケーションのカンマ区切りのリスト。

threshold カウントするローの最大数。CountUploadRows の所要時間を制限します。値 0 は制限が最大であることを示します。値 1 は、同期の必要なローがあるかどうかを判別する場合に使用します。

戻り値

指定されたパブリケーションからアップロードする必要のあるロー数。

例外

ULException class SQL エラーが発生しました。

1.7.12 CreateCommand() メソッド

この接続と現在のトランザクションに関連付けられる ULCommand オブジェクトを作成して初期化します。

構文

Visual Basic

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

C#

```
public new ULCommand CreateCommand ()
```

戻り値

新しい ULCommand オブジェクト。

備考

ULCommand オブジェクトのプロパティを使用して、その動作を制御できます。

ULCommand.CommandText プロパティを設定してから、コマンドを実行する必要があります。

これは、System.Data.IDbConnection.CreateCommand メソッドと

System.Data.Common.DbConnection.CreateCommand メソッドが厳密に型指定されたものです。

関連情報

[ULCommand クラス \[58 ページ\]](#)

[CommandText プロパティ \[98 ページ\]](#)

1.7.13 CreateDbCommand() メソッド

構文

Visual Basic

```
Protected Overrides Function CreateDbCommand () As DbCommand
```

C#

```
protected override DbCommand CreateDbCommand ()
```

1.7.14 CreateNotificationQueue(string, string) メソッド

UL 拡張: イベントキューを作成します。

構文

Visual Basic

```
Public Sub CreateNotificationQueue (  
    ByVal queueName As String,  
    ByVal parameters As String  
)
```

C#

```
public void CreateNotificationQueue (  
    string queueName,  
    string parameters  
)
```

パラメータ

queueName 新しいキューの名前。

parameters 作成パラメータ。現在は使われず、NULL に設定されています。

例外

ULException class SQL エラーが発生しました。

備考

このメソッドは、この接続のイベント通知キューを作成します。キュー名は、接続ごとにスコープされるため、別々の接続で同じ名前を持つキューを作成できます。イベント通知が送信されると、データベース内で一致する名前を持つすべてのキューが、個別のインスタンスの通知を受け取ります。名前では、大文字と小文字は区別されません。RegisterForEvent イベントを呼び出したときに、キューが指定されていない場合は、接続ごとにデフォルトのキューが作成されます。

関連情報

[DestroyNotificationQueue\(string\) メソッド \[146 ページ\]](#)

[ULException クラス \[328 ページ\]](#)

1.7.15 DeclareEvent(string) メソッド

UL 拡張: 指定したイベントを宣言します。

構文

Visual Basic

```
Public Sub DeclareEvent (ByVal eventName As String)
```

C#

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

パラメータ

eventName イベント名。

例外

ULException class SQL エラーが発生しました。

備考

登録およびトリガされるイベントを宣言します。Ultra Light では、データベースまたは環境での操作によってトリガされるシステムイベントの一部が事前に定義されています。イベント名は、ユニークにする必要があります。名前では、大文字と小文字が区別されません。名前がすでに使用されているか無効な場合は、エラーをスローします。

関連情報

[CreateNotificationQueue\(string, string\) メソッド \[144 ページ\]](#)

[ULException クラス \[328 ページ\]](#)

1.7.16 DestroyNotificationQueue(string) メソッド

UL 拡張: イベントキューを破棄します。

構文

Visual Basic

```
Public Sub DestroyNotificationQueue (ByVal queueName As String)
```

C#

```
public void DestroyNotificationQueue (string queueName)
```

パラメータ

queueName キューの名前。

例外

ULException class SQL エラーが発生しました。

備考

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

関連情報

[CreateNotificationQueue\(string, string\) メソッド \[144 ページ\]](#)

[ULException クラス \[328 ページ\]](#)

1.7.17 Dispose(bool) メソッド

ULCommand オブジェクトによって使用されるアンマネージリソースを解放するほか、必要に応じてマネージリソースを解放します。

構文

Visual Basic

```
Protected Overrides Sub Dispose (ByVal disposing As Boolean)
```

C#

```
protected override unsafe void Dispose (bool disposing)
```

パラメータ

disposing true の場合は、マネージリソースとアンマネージリソースの両方を破棄します。**false** の場合は、アンマネージリソースのみを破棄します。

1.7.18 EndSynchronize(IAsyncResult) メソッド

UL 拡張: 非同期で起動された同期が終了するまでブロックします。

構文

Visual Basic

```
Public Sub EndSynchronize (ByVal asyncResult As IAsyncResult)
```

C#

```
public void EndSynchronize (IAsyncResult asyncResult)
```

パラメータ

asyncResult BeginSynchronize メソッドから返された IAsyncResult。

例外

ULException class SQL エラーが発生しました。

備考

同期中にエラーが発生した場合、ULException 例外がスローされます。

関連情報

[BeginSynchronize\(\) メソッド \[132 ページ\]](#)

[BeginSynchronize\(Control, ULSyncProgressedDlg, object\) メソッド \[133 ページ\]](#)

[CancelSynchronize\(IAsyncResult\) メソッド \[138 ページ\]](#)

1.7.19 ExecuteTable メソッド

UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。

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 ULTable	ExecuteTable(string) [149 ページ]	UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。
public ULTable	ExecuteTable(string, string) [150 ページ]	UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。
public ULTable	ExecuteTable(string, string, CommandBehavior) [152 ページ]	UL 拡張: コマンドの動作を指定してデータベーステーブルを取得し、直接操作します。

このセクションの内容:

[ExecuteTable\(string\) メソッド \[149 ページ\]](#)

UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。

[ExecuteTable\(string, string\) メソッド \[150 ページ\]](#)

UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。

[ExecuteTable\(string, string, CommandBehavior\) メソッド \[152 ページ\]](#)

UL 拡張: コマンドの動作を指定してデータベーステーブルを取得し、直接操作します。

1.7.19.1 ExecuteTable(string) メソッド

UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。

構文

Visual Basic

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

C#

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

パラメータ

tableName 開くテーブルの名前。

戻り値

ULTable オブジェクトとしてのテーブル。

例外

ULException class SQL エラーが発生しました。

InvalidOperationException *tableName* は無効です。

備考

テーブルのプライマリキーを使用して、テーブルが開かれます (ソートされます)。

このメソッドは、ULCommand オブジェクトを必要としない ULCommand.ExecuteTable メソッドのショートカットです。また、以前のバージョンの Ultra Light.NET から移行できるようにするために用意されています (Sap.UltraLite.Connection.GetTable メソッドと Sap.UltraLite.Table.Open メソッドが置き換えられます)。

例

次のコードでは、テーブルのプライマリキーを使用して MyTable というテーブルが開かれます。また、conn という開いた ULConnection インスタンスが想定されています。

```
' Visual Basic
Dim t As ULTable = conn.ExecuteTable("MyTable")
' The line above is equivalent to
' Dim cmd As ULCommand = conn.CreateCommand()
' cmd.CommandText = "MyTable"
' cmd.CommandType = CommandType.TableDirect
' Dim t As ULTable = cmd.ExecuteTable()
' cmd.Dispose()
```

対応する C# 言語のコードを次に示します。

```
// C#
ULTable t = conn.ExecuteTable("MyTable");
// The line above is equivalent to
// ULTable t;
// using(ULCommand cmd = conn.CreateCommand())
// {
//     cmd.CommandText = "MyTable";
//     cmd.CommandType = CommandType.TableDirect;
//     t = cmd.ExecuteTable();
// }
```

関連情報

[ExecuteTable\(string, string\) メソッド \[150 ページ\]](#)

[ULTable クラス \[526 ページ\]](#)

[ExecuteTable\(\) メソッド \[94 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

1.7.19.2 ExecuteTable(string, string) メソッド

UL 拡張: ULTable オブジェクトのデータベーステーブルを取得して直接操作します。

構文

Visual Basic

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

C#

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

パラメータ

tableName 開くテーブルの名前。

indexName テーブルを開く(ソートする) インデックスの名前。

戻り値

ULTable オブジェクトとしてのテーブル。

例外

ULException class SQL エラーが発生しました。

InvalidOperationException *tableName* は無効です。

備考

テーブルは、指定されたインデックスを使用して開かれます(ソートされます)。

このメソッドは、ULCommand オブジェクトを必要としない ULCommand.ExecuteTable メソッドのショートカットです。また、以前のバージョンの Ultra Light.NET から移行できるようにするために用意されています (Sap.UltraLite.Connection.GetTable メソッドと Sap.UltraLite.Table.Open メソッドが置き換えられます)。

例

次のコードでは、MyIndex というインデックスを使用して MyTable というテーブルが開かれます。また、conn という開いた ULConnection オブジェクトが想定されています。

```
' Visual Basic  
Dim t As ULTable = conn.ExecuteTable("MyTable", "MyIndex")  
' The line above is equivalent to  
' Dim cmd As ULCommand = conn.CreateCommand()  
' cmd.CommandText = "MyTable"  
' cmd.IndexName = "MyIndex"  
' cmd.CommandType = CommandType.TableDirect  
' Dim t As ULTable = cmd.ExecuteTable()
```

```
cmd.Dispose()
```

対応する C# 言語のコードを次に示します。

```
// C#
ULTable t = conn.ExecuteTable("MyTable", "MyIndex");
// The line above is equivalent to
// ULTable t;
// using(ULCommand cmd = conn.CreateCommand())
// {
//     cmd.CommandText = "MyTable";
//     cmd.IndexName = "MyIndex";
//     cmd.CommandType = CommandType.TableDirect;
//     t = cmd.ExecuteTable();
// }
```

関連情報

[ExecuteTable\(string\) メソッド \[149 ページ\]](#)

[ULTable クラス \[526 ページ\]](#)

[ExecuteTable\(\) メソッド \[94 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

1.7.19.3 ExecuteTable(string, string, CommandBehavior) メソッド

UL 拡張: コマンドの動作を指定してデータベーステーブルを取得し、直接操作します。

構文

Visual Basic

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

C#

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


パラメータ

tableName 開くテーブルの名前。

indexName テーブルを開く (ソートする) インデックスの名前。

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

戻り値

ULTable オブジェクトとしてのテーブル。

例外

ULException class SQL エラーが発生しました。

InvalidOperationException *tableName* は無効です。

備考

テーブルは、指定されたインデックスを使用して開かれます (ソートされます)。

このメソッドは、ULCommand オブジェクトを必要としない

ULCommand.ExecuteTable(System.Data.CommandBehavior) メソッドのショートカットです。また、以前のバージョンの Ultra Light.NET から移行できるようにするために用意されています (Sap.UltraLite.Connection.GetTable メソッドと Sap.UltraLite.Table.Open メソッドが置き換えられます)。

例

次のコードでは、MyIndex というインデックスを使用して MyTable というテーブルが開かれます。また、conn という、開いた ULConnection オブジェクトが想定されています。

```
' Visual Basic
Dim t As ULTable = conn.ExecuteTable( _
    "MyTable", "MyIndex", CommandBehavior.Default _
)
' The line above is equivalent to the following code:
' Dim cmd As ULCommand = conn.CreateCommand()
' cmd.CommandText = "MyTable"
' cmd.IndexName = "MyIndex"
' cmd.CommandType = CommandType.TableDirect
' Dim t As ULTable = cmd.ExecuteTable(CommandBehavior.Default)
' cmd.Dispose()
```

対応する C# 言語のコードを次に示します。

```
// C#
ULTable t = conn.ExecuteTable(
    "MyTable", "MyIndex", CommandBehavior.Default
);
// The line above is equivalent to the following code:
// ULTable t;
// using(ULCommand cmd = conn.CreateCommand())
// {
//     cmd.CommandText = "MyTable";
//     cmd.IndexName = "MyIndex";
//     cmd.CommandType = CommandType.TableDirect;
//     t = cmd.ExecuteTable(CommandBehavior.Default);
// }
```

関連情報

[ExecuteTable\(string\) メソッド \[149 ページ\]](#)

[ExecuteTable\(string, string\) メソッド \[150 ページ\]](#)

[ULTable クラス \[526 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

1.7.20 GetLastDownloadTime(string) メソッド

UL 拡張: 指定されたパブリケーションの最後のダウンロードの時刻を返します。

構文

Visual Basic

```
Public Function GetLastDownloadTime (ByVal publication As String) As Date
```

C#

```
public unsafe DateTime GetLastDownloadTime (string publication)
```

パラメータ

publication チェックするパブリケーション。

戻り値

最後のダウンロードのタイムスタンプ。パブリケーションに SYNC_ALL_DB 定数を使用した場合は、データベース全体を最後にダウンロードした時刻を返します。

例外

ULException class SQL エラーが発生しました。

備考

publication パラメータは、チェックするパブリケーションの名前です。

関連情報

[ResetLastDownloadTime\(string\) メソッド \[165 ページ\]](#)

1.7.21 GetNewUUID() メソッド

UL 拡張: 新しい UUID (System.Guid) を生成します。

構文

Visual Basic

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

C#

```
public unsafe Guid GetNewUUID ()
```

戻り値

System.Guid として返される新しい UUID。

例外

ULException class SQL エラーが発生しました。

備考

このメソッドは、.NET Compact Framework に含まれていないため、ここで提供されています。

1.7.22 GetNotification(string, int) メソッド

UL 拡張: 通知またはタイムアウトをブロックします。

構文

Visual Basic

```
Public Function GetNotification (  
    ByVal queueName As String,  
    ByVal wait_ms As Integer  
) As String
```

C#

```
public string GetNotification (  
    string queueName,  
    int wait_ms  
)
```

パラメータ

queueName 待機するキューの名前。

wait_ms 待機時間 (ミリ秒)。待機時間を無限に設定するには、System.Threading.Timeout.Infinite (-1) を使用します。

戻り値

待機時間の期限が切れるか、待機がキャンセルされた場合は NULL。それ以外の場合はイベント名。

例外

`ULException` class SQL エラーが発生しました。

備考

このメソッドは、イベント通知を読み込みます。この呼び出しは、通知が受信されるまで、または指定された待機時間が経過するまでブロックします。無期限に待機する場合は、`wait_ms` パラメータに `System.Threading.Timeout.Infinite` を渡します。待機をキャンセルするには、指定したキューに別の通知を送信するか、`CancelGetNotification` メソッドを使用します。通知を読み込んだら、`ReadNotificationParameter` メソッドを使用して追加のパラメータを取得します。

関連情報

[SendNotification\(string, string, string\) メソッド \[167 ページ\]](#)

[GetNotificationParameter\(string, string\) メソッド \[157 ページ\]](#)

[CancelGetNotification\(string\) メソッド \[137 ページ\]](#)

[ULException クラス \[328 ページ\]](#)

1.7.23 GetNotificationParameter(string, string) メソッド

UL 拡張: `GetNotification` メソッドが読み取ったばかりのイベントのパラメータの値を取得します。

構文

Visual Basic

```
Public Function GetNotificationParameter (  
    ByVal queueName As String,  
    ByVal parameterName As String  
) As String
```

C#

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

パラメータ

queueName 待機するキューの名前。

`parameterName` 値を返すパラメータの名前。

戻り値

パラメータが見つかった場合はパラメータ値、それ以外の場合は NULL。

例外

ULException class SQL エラーが発生しました。

備考

このメソッドは、`ULGetNotification` メソッドによって読み込まれたイベント通知のパラメータ値を取得します。指定されたキューでの最後に読み込まれた通知のパラメータのみ取得できます。

関連情報

[GetNotification\(string, int\) メソッド \[156 ページ\]](#)

[ULException クラス \[328 ページ\]](#)

1.7.24 GetSchema メソッド

サポートされているスキーマコレクションのリストを返します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public override DataTable	GetSchema() [159 ページ]	サポートされているスキーマコレクションのリストを返します。
public override DataTable	GetSchema(string) [159 ページ]	この <code>ULConnection</code> オブジェクトについて指定されたメタデータコレクションに関する情報を返します。

変更子とタイプ	オーバーロード名	説明
public override DataTable	GetSchema(string, string[]) [160 ページ]	この ULConnection オブジェクトのデータソースのスキーマ情報を返します。このとき、文字列が指定されている場合はスキーマ名として使用し、文字列配列が指定されている場合は制限値として使用します。

このセクションの内容:

[GetSchema\(\) メソッド \[159 ページ\]](#)

サポートされているスキーマコレクションのリストを返します。

[GetSchema\(string\) メソッド \[159 ページ\]](#)

この ULConnection オブジェクトについて指定されたメタデータコレクションに関する情報を返します。

[GetSchema\(string, string\[\]\) メソッド \[160 ページ\]](#)

この ULConnection オブジェクトのデータソースのスキーマ情報を返します。このとき、文字列が指定されている場合はスキーマ名として使用し、文字列配列が指定されている場合は制限値として使用します。

1.7.24.1 GetSchema() メソッド

サポートされているスキーマコレクションのリストを返します。

構文

Visual Basic

```
Public Overrides Function GetSchema () As DataTable
```

C#

```
public override DataTable GetSchema ()
```

1.7.24.2 GetSchema(string) メソッド

この ULConnection オブジェクトについて指定されたメタデータコレクションに関する情報を返します。

構文

Visual Basic

```
Public Overrides Function GetSchema (ByVal collection As String) As DataTable
```

C#

```
public override DataTable GetSchema (string collection)
```

パラメータ

collection メタデータコレクションの名前。名前が提供されない場合は、MetaDataCollections 値が使用されます。

関連情報

[GetSchema\(string, string\[\]\) メソッド \[160 ページ\]](#)

[ULConnection クラス \[121 ページ\]](#)

1.7.24.3 GetSchema(string, string[]) メソッド

この ULConnection オブジェクトのデータソースのスキーマ情報を返します。このとき、文字列が指定されている場合はスキーマ名として使用し、文字列配列が指定されている場合は制限値として使用します。

構文

Visual Basic

```
Public Overrides Function GetSchema (  
    ByVal collection As String,  
    ByVal restrictions As String()  
) As DataTable
```

C#

```
public override DataTable GetSchema (  
    string collection,  
    string[] restrictions  
)
```

パラメータ

collection メタデータコレクションの名前。名前が提供されない場合は、MetaDataCollections が使用されます。

restrictions 要求されるスキーマの制限値のセット。

戻り値

スキーマ情報が格納されている DataTable オブジェクト。

備考

このメソッドを使用すると、データベースに各種のメタデータを問い合わせることができます。メタデータの各型にはコレクション名が指定されており、そのデータを受け取るにはコレクション名を渡す必要があります。デフォルトのコレクション名は `MetaDataCollections` です。

引数を指定せずに、または `MetaDataCollections` というスキーマコレクション名を指定して `GetSchema` メソッドを呼び出すことによって、.NET データプロバイダに問い合わせをして、サポートされているスキーマコレクションのリストを判断できます。これによって、サポートされているスキーマコレクション (`CollectionName`)、それぞれがサポートする制限の数 (`NumberOfRestrictions`)、使用する識別子部分の数のリストから成る `DataTable` が返されます。

コレクション	メタデータ
<code>Columns</code>	データベース内のすべてのカラムに関する情報を返します。
<code>DataSourceInformation</code>	データベースプロバイダに関する情報を返します。
<code>DataTypes</code>	サポートされているデータ型のリストを返します。
<code>ForeignKeys</code>	データベース内のすべての外部キーに関する情報を返します。
<code>IndexColumns</code>	データベース内のすべてのインデックスカラムに関する情報を返します。
<code>Indexes</code>	データベース内のすべてのインデックスに関する情報を返します。
<code>MetaDataCollections</code>	すべてのコレクション名のリストを返します。
<code>Publications</code>	データベース内のすべてのパブリケーションに関する情報を返します。
<code>ReservedWords</code>	Ultra Light が使用する予約語のリストを返します。
<code>Restrictions</code>	<code>GetSchema</code> で使用される制限に関する情報を返します。
<code>Tables</code>	データベース内のすべてのテーブルに関する情報を返します。

これらのコレクション名は、`ULMetaDataCollectionNames` クラスの読み込み専用プロパティとしても使用できます。

返される結果は、`GetSchema` メソッドへの呼び出しの中で制限の配列を指定することによってフィルタできます。

各コレクションで有効な制限は、次の文を呼び出すことで問い合わせることができます。

```
GetSchema( "Restrictions" )
```

コレクションが 4 つの制限を必要とする場合、制限パラメータは、NULL、または 4 つの値から成る文字列です。

特定の制限をフィルタするには、フィルタする文字列を配列内の所定の位置に指定し、使用しない位置には NULL を指定します。たとえば、`Tables` コレクションには 3 つの制限があります。Table、TableType、SyncType です。

Table コレクションをフィルタするには、次の手順に従います。

```
GetSchema("Tables", new string[] { "my_table", NULL, NULL })
```

my_table という名前のすべてのテーブルに関する情報を返します。

```
GetSchema("Tables", new string[] { NULL, "User", NULL })
```

すべてのユーザテーブルに関する情報を返します。

関連情報

[ULConnection クラス \[121 ページ\]](#)

[ULMetaDataCollectionNames クラス \[377 ページ\]](#)

1.7.25 GrantConnectTo(string, string) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

uid データベースへのアクセス権を受け取るユーザ ID。

pwd ユーザ ID に関連付けられるパスワード。

備考

既存のユーザ ID が指定されていれば、この関数を使用してそのユーザのパスワードを更新します。Ultra Light では、最大で 4 人のユーザがサポートされます。このメソッドが有効になるのは、接続が開かれたときにユーザ認証が有効になっていた場合だけです。

関連情報

[UserID プロパティ \[196 ページ\]](#)

[Password プロパティ \[195 ページ\]](#)

[ConnectionString プロパティ \[177 ページ\]](#)

1.7.26 Open() メソッド

以前に指定された接続文字列を使用してデータベースへの接続を開きます。

構文

Visual Basic

```
Public Overrides Sub Open ()
```

C#

```
public override void Open ()
```

例外

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

ULException class データベースを開くときに SQL エラーが発生しました。

備考

接続は、使用し終わったら明示的に閉じるか、破棄する必要があります。

関連情報

[ConnectionString プロパティ \[177 ページ\]](#)

[State プロパティ \[183 ページ\]](#)

1.7.27 RegisterForEvent(string, string, string, bool) メソッド

UL 拡張: オブジェクトからイベントを取得するためのキューを登録します。

構文

Visual Basic

```
Public Sub RegisterForEvent (  
    ByVal eventName As String,  
    ByVal objectName As String,  
    ByVal queueName As String,  
    ByVal registerNotUnReg As Boolean  
)
```

C#

```
public void RegisterForEvent (  
    string eventName,  
    string objectName,  
    string queueName,  
    bool registerNotUnReg  
)
```

パラメータ

eventName イベント名。

objectName イベントを適用するオブジェクトの名前。たとえば、テーブル名です。

queueName 使用するイベントキューの名前。

registerNotUnReg 登録する場合は true、登録解除する場合は false。

例外

ULException class SQL エラーが発生しました。

備考

このメソッドは、イベントの通知を受信するキューを登録します。キュー名が指定されていない場合は、デフォルトの接続キューが暗黙で指定され、作成されます。特定のシステムイベントでは、そのイベントが適用されるオブジェクト名を指定できます。たとえば、TableModified イベントではテーブル名を指定できます。SendNotification メソッドとは異なり、登録された特定のキューのみイベントの通知を受信します。別の接続における同じ名前前のキューでは、明示的に登録されていないかぎり、通知は受信されません。キューまたはイベントが存在しない場合は、エラーをスローします。

関連情報

[DeclareEvent\(string\) メソッド \[145 ページ\]](#)

[CreateNotificationQueue\(string, string\) メソッド \[144 ページ\]](#)

[ULException クラス \[328 ページ\]](#)

1.7.28 ResetLastDownloadTime(string) メソッド

UL 拡張: 最後のダウンロードの時刻をリセットします。

構文

Visual Basic

```
Public Sub ResetLastDownloadTime (ByVal pubs As String)
```

C#

```
public void ResetLastDownloadTime (string pubs)
```

例外

ULException class SQL エラーが発生しました。

関連情報

[GetLastDownloadTime\(string\) メソッド \[154 ページ\]](#)

1.7.29 RevokeConnectFrom(string) メソッド

UL 拡張: 指定されたユーザ ID から Ultra Light データベースへのアクセス権を取り消します。

構文

Visual Basic

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

C#

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

パラメータ

uid データベースへのアクセス権を取り消すユーザ ID。

例外

ULException class SQL エラーが発生しました。

関連情報

[GrantConnectTo\(string, string\) メソッド \[162 ページ\]](#)

1.7.30 RollbackPartialDownload() メソッド

UL 拡張: 部分的なダウンロードからの、データベースへの未処理の変更をロールバックします。

構文

Visual Basic

```
Public Sub RollbackPartialDownload ()
```

C#

```
public void RollbackPartialDownload ()
```

例外

ULException class SQL エラーが発生しました。

関連情報

[KeepPartialDownload プロパティ \[493 ページ\]](#)

[ResumePartialDownload プロパティ \[497 ページ\]](#)

1.7.31 SendNotification(string, string, string) メソッド

UL 拡張: 一致するキューに通知を送信します。

構文

Visual Basic

```
Public Function SendNotification (  
    ByVal queueName As String,  
    ByVal eventName As String,  
    ByVal parameters As String  
) As Integer
```

C#

```
public unsafe int SendNotification (  
    string queueName,  
    string eventName,  
    string parameters  
)
```

パラメータ

queueName 使用するイベントキューの名前。

eventName イベント名。

parameters 受け渡すパラメータ。

戻り値

送信済みの通知の数 (一致するキューの数)。

例外

ULException class SQL エラーが発生しました。

備考

一致するキューの数を返します。

このメソッドは、指定された名前に一致するすべてのキュー (現在の接続におけるキューを含む) に通知を送信します。この呼び出しはブロックしません。特別なキュー名の "*" を使用すると、すべてのキューに送信します。

関連情報

[DeclareEvent\(string\) メソッド \[145 ページ\]](#)

[RegisterForEvent\(string, string, string, bool\) メソッド \[164 ページ\]](#)

[ULException クラス \[328 ページ\]](#)

1.7.32 SetSyncListener(ULSyncProgressListener) メソッド

同期メッセージの処理に使用するリスナオブジェクトを指定します。

構文

Visual Basic

```
Public Sub SetSyncListener (ByVal listener As ULSyncProgressListener)
```

C#

```
public void SetSyncListener (ULSyncProgressListener listener)
```

パラメータ

listener SyncProgressed メソッドを実装する ULSyncProgressListener オブジェクト。この接続で、同期メッセージに対して呼び出されます。

例外

ULException class SQL エラーが発生しました。

備考

SQL 文 SYNCHRONIZE profileName が実行されると、その進行状況のメッセージが NULL (Microsoft Visual Basic の場合は Nothing) でない場合は syncListener オブジェクトにルーティングされます。

リスナを削除するには、SetSyncListener メソッドへの呼び出しで NULL 参照を渡します。

関連情報

[ULSyncProgressListener インタフェース \[518 ページ\]](#)

1.7.33 StartSynchronizationDelete() メソッド

UL 拡張: 同期用に、この接続によって行われる以後のすべての削除にマークを付けます。

構文

Visual Basic

```
Public Sub StartSynchronizationDelete ()
```

C#

```
public void StartSynchronizationDelete ()
```

例外

ULException class SQL エラーが発生しました。

備考

このメソッドが呼び出されると、すべての削除操作が再度同期され、Ultra Light データベースから削除されたローが統合データベースからも削除されます。

関連情報

[StopSynchronizationDelete\(\) メソッド \[170 ページ\]](#)

[Truncate\(\) メソッド \[554 ページ\]](#)

1.7.34 StopSynchronizationDelete() メソッド

UL 拡張: 削除操作が同期されないようにします。

構文

Visual Basic

```
Public Sub StopSynchronizationDelete ()
```

C#

```
public void StopSynchronizationDelete ()
```

例外

ULException class SQL エラーが発生しました。

備考

領域を節約するために Ultra Light データベースの古い情報を削除して、統合データベースにはそれを残しておく場合に、このメソッドを使用すると便利です。

関連情報

[StartSynchronizationDelete\(\) メソッド \[169 ページ\]](#)

1.7.35 Synchronize メソッド

UL 拡張: 現在の ULConnection.SyncParms オブジェクトを使用してデータベースを同期します。

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 void	Synchronize() [171 ページ]	UL 拡張: 現在の ULConnection.SyncParms オブジェクトを使用してデータベースを同期します。
public void	Synchronize(ULSyncProgressListener) [172 ページ]	UL 拡張: 指定されたリスナに送信されるプログレスイベントとともに、現在の ULConnection.SyncParms オブジェクトを使用してデータベースを同期します。

このセクションの内容:

[Synchronize\(\) メソッド \[171 ページ\]](#)

UL 拡張: 現在の ULConnection.SyncParms オブジェクトを使用してデータベースを同期します。

[Synchronize\(ULSyncProgressListener\) メソッド \[172 ページ\]](#)

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

1.7.35.1 Synchronize() メソッド

UL 拡張: 現在の ULConnection.SyncParms オブジェクトを使用してデータベースを同期します。

構文

Visual Basic

```
Public Sub Synchronize ()
```

C#

```
public void Synchronize ()
```

例外

ULException class SQL エラーが発生しました。

備考

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

関連情報

[Synchronize\(ULSyncProgressListener\) メソッド \[172 ページ\]](#)

[SyncParms プロパティ \[183 ページ\]](#)

[SyncResult プロパティ \[184 ページ\]](#)

1.7.35.2 Synchronize(ULSyncProgressListener) メソッド

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

構文

Visual Basic

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

C#

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

パラメータ

listener 同期プログレスイベントを受信するオブジェクト。

例外

ULException class SQL エラーが発生しました。

備考

同期中のエラーは、`ULSyncProgressState.STATE_ERROR` イベントとして送信され、`ULExceptions` としてスローされます。

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

関連情報

[ULSyncProgressListener インタフェース \[518 ページ\]](#)

[Synchronize\(\) メソッド \[171 ページ\]](#)

[SyncParms プロパティ \[183 ページ\]](#)

[ULException クラス \[328 ページ\]](#)

[SyncResult プロパティ \[184 ページ\]](#)

1.7.36 TriggerEvent(string, string) メソッド

UL 拡張: イベントをトリガします。

構文

Visual Basic

```
Public Function TriggerEvent (  
    ByVal eventName As String,  
    ByVal parameters As String  
) As Integer
```

C#

```
public unsafe int TriggerEvent (  
    string eventName,  
    string parameters  
)
```

パラメータ

eventName トリガされるイベントの名前。

parameters 受け渡すパラメータ。

戻り値

送信済みのイベント通知の数。

例外

ULException class SQL エラーが発生しました。

備考

送信済みの通知の数を返します。

このメソッドは、イベントをトリガして、登録されたすべてのキューに通知を送信します。

関連情報

[DeclareEvent\(string\) メソッド \[145 ページ\]](#)

[RegisterForEvent\(string, string, string, bool\) メソッド \[164 ページ\]](#)

[ULException クラス \[328 ページ\]](#)

1.7.37 ValidateDatabase メソッド

UL 拡張: 現在のデータベースの検証を実行します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public void	ValidateDatabase(ULDBValid) [175 ページ]	UL 拡張: 現在のデータベースの検証を実行します。
public void	ValidateDatabase(ULDBValid, string) [176 ページ]	UL 拡張: 現在のデータベースの検証を実行します。

このセクションの内容:

[ValidateDatabase\(ULDBValid\) メソッド \[175 ページ\]](#)

UL 拡張: 現在のデータベースの検証を実行します。

[ValidateDatabase\(ULDBValid, string\) メソッド \[176 ページ\]](#)

UL 拡張: 現在のデータベースの検証を実行します。

1.7.37.1 ValidateDatabase(ULDBValid) メソッド

UL 拡張: 現在のデータベースの検証を実行します。

構文

Visual Basic

```
Public Sub ValidateDatabase (ByVal how As ULDBValid)
```

C#

```
public void ValidateDatabase (ULDBValid how)
```

パラメータ

how データベースを検証する方法を記述します。

例外

ULException class データベースが破損している場合、SQL_CORRUPT_ULTRALITE_INDEX または SQL_CORRUPT_ULTRALITE_DATABASE エラーが発生することがあります。

例

次のコードでは、現在のデータベースを検証します。

```
' Visual Basic  
conn.ValidateDatabase( Sap.Data.UltraLite.ULVF_INDEX )
```

対応する C# 言語のコードを次に示します。

```
// C#  
conn.ValidateDatabase( Sap.Data.UltraLite.ULVF_INDEX )
```

関連情報

[ValidateDatabase\(string, ULDBValid\) メソッド \[263 ページ\]](#)

[ULDBValid 列挙体 \[590 ページ\]](#)

1.7.37.2 ValidateDatabase(ULDBValid, string) メソッド

UL 拡張: 現在のデータベースの検証を実行します。

構文

Visual Basic

```
Public Sub ValidateDatabase (  
    ByVal how As ULDBValid,  
    ByVal tableName As String  
)
```

C#

```
public void ValidateDatabase (  
    ULDBValid how,  
    string tableName  
)
```

パラメータ

how データベースを検証する方法を記述します。

tableName NULL (Visual Basic の場合は Nothing) の場合はデータベース全体を検証し、それ以外の場合は指定されたテーブルだけを検証します。

例外

ULException class データベースが破損している場合、SQLE_CORRUPT_ULTRALITE_INDEX または SQLE_CORRUPT_ULTRALITE_DATABASE エラーが発生することがあります。

例

次のコードでは、現在のデータベースを検証します。

```
' Visual Basic  
conn.ValidateDatabase( Sap.Data.UltraLite.ULVF_INDEX, Nothing )
```

対応する C# 言語のコードを次に示します。

```
// C#  
conn.ValidateDatabase( Sap.Data.UltraLite.ULVF_INDEX, null )
```


関連情報

[ValidateDatabase\(string, ULDBValid\) メソッド \[263 ページ\]](#)

[ULDBValid 列挙体 \[590 ページ\]](#)

1.7.38 ConnectionString プロパティ

Ultra Light.NET データベースへの接続を開くためのパラメータを指定します。

構文

Visual Basic

```
Public Overrides Property ConnectionString As String
```

C#

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

備考

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

この接続を開くためのパラメータは、キーワードと値の組み合わせがセミコロンで区切られたリストにする必要があります。デフォルトは、空の文字列 (無効な接続文字列) です。

UL 拡張: Ultra Light.NET によって使用されるパラメータは Ultra Light データベースに固有であるため、その接続文字列は SQL Anywhere の接続文字列との互換性がありません。

パラメータ値は、単一引用符または二重引用符で囲むことができますが、引用符で囲まれた部分に同じ種類の引用符が含まれないことが条件になります。値にセミコロンが含まれるか、値が引用符で始まるか、値の前後にスペースが必要な場合は、値を引用符で囲む必要があります。

パラメータ値を引用符で囲まない場合は、値にセミコロンが含まれていないこと、また値が単一引用符と二重引用符のいずれかで始まることを確認してください。値の前後のスペースは無視されます。

デフォルトでは、UID=DBA と PWD=sql で接続が確立されます。データベースをより強力で保護するには、ユーザ名 DBA のパスワードを変更するか、新しいユーザを作成して (GrantConnectTo メソッドを使用)、DBA ユーザを削除します (RevokeConnectFrom を使用)。

例

次のコードでは、Windows Mobile デバイス上のデータベース ¥UltraLite¥MyDatabase.udb への接続を作成して開きます。

```
' Visual Basic
Dim openParms As ULConnectionParms = New ULConnectionParms
openParms.DatabaseOnDevice = "¥UltraLite¥MyDatabase.udb"
```

```
Dim conn As ULConnection = New ULConnection
conn.ConnectionString = openParms.ToString()
conn.Open()
```

対応する C# 言語のコードを次に示します。

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

関連情報

[Open\(\) メソッド \[163 ページ\]](#)

[ULConnectionParms クラス \[187 ページ\]](#)

[GrantConnectTo\(string, string\) メソッド \[162 ページ\]](#)

1.7.39 ConnectionTimeout プロパティ

この機能は Ultra Light.NET ではサポートされていません。

構文

Visual Basic

```
Public ReadOnly Overrides Property ConnectionTimeout As Integer
```

C#

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

備考

値は常に 0 です。

1.7.40 Database プロパティ

接続を開くデータベースの名前を返します。

構文

Visual Basic

```
Public ReadOnly Overrides Property Database As String
```

C#

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

備考

データベースの名前が含まれる文字列。

Windows Mobile デバイスでは、ULConnection オブジェクトは dbn、ce_file の順に接続文字列を参照します。

デスクトップマシンでは、ULConnection オブジェクトは dbn、nt_file の順に接続文字列を参照します。

1.7.41 DatabaseID プロパティ

UL 拡張: グローバルオートインクリメントカラムに使用するデータベース ID の値を指定します。

構文

Visual Basic

```
Public Property DatabaseID As Long
```

C#

```
public unsafe long DatabaseID {get;set;}
```

備考

現在のデータベースのデータベース ID の値。

データベース ID の値は、[0, System.UInt32.MaxValue] の範囲内であることが必要です。

ULConnection.INVALID_DATABASE_ID の値は、現在のデータベースにデータベース ID が設定されていないことを示すのに使用されます。

関連情報

[GetDatabaseProperty\(string\) メソッド \[267 ページ\]](#)

[SetDatabaseOption\(string, string\) メソッド \[271 ページ\]](#)

1.7.42 DataSource プロパティ

この機能は Ultra Light.NET ではサポートされていません。

構文

Visual Basic

```
Public ReadOnly Overrides Property DataSource As String
```

C#

```
public override string DataSource {get;}
```

備考

値は常に空の文字列です。

1.7.43 GlobalAutoIncrementUsage プロパティ

UL 拡張: 利用可能なグローバルオートインクリメントの値の使用済み比率 (%) を返します。

構文

Visual Basic

```
Public ReadOnly Property GlobalAutoIncrementUsage As Short
```

C#

```
public unsafe short GlobalAutoIncrementUsage {get;}
```

備考

利用可能なグローバルオートインクリメントの値の使用済み比率 (%)。これは、0 ~ 100 の範囲内の整数です。

比率が 100% に近づいたら、アプリケーションが ULConnection.DatabaseID 値を使用して、新しいグローバルデータベース ID を設定します。

関連情報

[ULDatabaseManager クラス \[255 ページ\]](#)

[DatabaseID プロパティ \[179 ページ\]](#)

1.7.44 LastIdentity プロパティ

UL 拡張: 直前に使用した identity の値を返します。

構文

Visual Basic

```
Public ReadOnly Property LastIdentity As ULong
```

C#

```
public unsafe ulong LastIdentity {get;}
```

備考

直前に使用した identity の値 (符号なし long)。

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

```
SELECT @identity
```

LastIdentity プロパティは、グローバルオートインクリメントカラムで使うと特に便利です。

このプロパティでは最後に割り当てられたデフォルト値がわかるだけなので、間違った結果を取らないために INSERT 文を実行した直後にこの値を取り出してください。

ときには、1つの INSERT 文にグローバルオートインクリメント型のカラムが複数含まれていることがあります。この場合、LastIdentity プロパティは、生成されたデフォルト値の 1つですが、その値の生成元のカラムを判別する信頼できる方法はありません。このため、このような状況を回避するようなデータベースの設計と INSERT 文の記述を行ってください。

1.7.45 Schema プロパティ

UL 拡張: この接続に関連付けられている現在のデータベースのスキーマへのアクセスに使用します。

構文

Visual Basic

```
Public ReadOnly Property Schema As ULDatabaseSchema
```

C#

```
public ULDatabaseSchema Schema {get;}
```

備考

この接続が開かれているデータベースのスキーマを表す ULDatabaseSchema オブジェクトへの参照してください。

このプロパティが有効なのは、接続が開かれている間だけです。

関連情報

[ULDatabaseSchema クラス \[265 ページ\]](#)

1.7.46 ServerVersion プロパティ

この機能は Ultra Light.NET ではサポートされていません。

構文

Visual Basic

```
Public ReadOnly Overrides Property ServerVersion As String
```

C#

```
public override string ServerVersion {get;}
```

備考

値は常に空の文字列です。

1.7.47 State プロパティ

接続の現在のステータスを返します。

構文

Visual Basic

```
Public ReadOnly Overrides Property State As ConnectionState
```

C#

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

備考

接続が開いている場合は `System.Data.ConnectionState.Open`、閉じている場合は `System.Data.ConnectionState.Closed` を返します。

関連情報

[StateChange イベント \[186 ページ\]](#)

1.7.48 SyncParms プロパティ

UL 拡張: この接続の同期設定を指定します。

構文

Visual Basic

```
Public ReadOnly Property SyncParms As ULSyncParms
```

C#

```
public ULSyncParms SyncParms {get;}
```

備考

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

関連情報

[Synchronize\(\) メソッド \[171 ページ\]](#)

[SyncResult プロパティ \[184 ページ\]](#)

[ULSyncParms クラス \[487 ページ\]](#)

1.7.49 SyncResult プロパティ

UL 拡張: この接続の最後の同期結果を返します。

構文

Visual Basic

```
Public ReadOnly Property SyncResult As ULSyncResult
```

C#

```
public ULSyncResult SyncResult {get;}
```

備考

この接続の最後の同期結果を表す ULSyncResult オブジェクトへの参照してください。

関連情報

[Synchronize\(\) メソッド \[171 ページ\]](#)

[SyncParms プロパティ \[183 ページ\]](#)

1.7.50 InfoMessage イベント

Ultra Light.NET が、この接続の警告または情報メッセージを送信するときに発生します。

構文

Visual Basic

```
Public Event InfoMessage As ULInfoMessageEventHandler
```


C#

```
public ULInfoMessageEventHandler InfoMessage;
```

備考

Ultra Light.NET の警告または情報メッセージを処理するには、ULInfoMessageEventHandler デリゲートを作成し、このイベントにアタッチする必要があります。

例

次のコードでは、情報メッセージのイベントハンドラが定義されます。

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

対応する C# 言語のコードを次に示します。

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

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

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

対応する C# 言語のコードを次に示します。

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

関連情報

[ULInfoMessageEventHandler\(object, ULInfoMessageEventArgs\) デリゲート \[580 ページ\]](#)

1.7.51 StateChange イベント

この接続のステータスが変更されると発生します。

構文

Visual Basic

```
Public Event StateChange As StateChangeEventHandler
```

C#

```
public override StateChangeEventHandler StateChange;
```

備考

ステータス変更メッセージを処理するには、System.Data.StateChangeEventHandler デリゲートを作成し、このイベントにアタッチする必要があります。

例

次のコードでは、ステータス変更のイベントハンドラが定義されます。

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

対応する C# 言語のコードを次に示します。

```
// C#
private void MyStateHandler(
    object obj, StateChangeEventArgs args
)
{
    System.Console.WriteLine(
        "StateHandler: " + args.OriginalState + " to "
        + args.CurrentState
    );
}
```

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

```
' Visual Basic
AddHandler conn.StateChange, AddressOf MyStateHandler
```

対応する C# 言語のコードを次に示します。

```
// C#
```

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

1.8 ULConnectionParms クラス

UL 拡張: Ultra Light データベースへの接続を開く接続文字列を作成します。

構文

Visual Basic

```
Public Class ULConnectionParms Inherits System.ComponentModel.Component
```

C#

```
public class ULConnectionParms : System.ComponentModel.Component
```

メンバー

ULConnectionParms 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	ULConnectionParms() [190 ページ]	ULConnectionParms インスタンスを、そのデフォルト値で初期化します。

メソッド

変更子とタイプ	メソッド	説明
public override string	ToString() [190 ページ]	このインスタンスの文字列表現を返します。

プロパティ

変更子とタイプ	プロパティ	説明
public string	AdditionalParms [191 ページ]	"名前=値" のペアをセミコロンで区切ったリストで、追加のパラメータを指定します。
public string	CacheSize [192 ページ]	キャッシュのサイズを指定します。
public string	ConnectionName [193 ページ]	接続の名前を指定します。
public string	DatabaseOnDesktop [194 ページ]	Windows デスクトッププラットフォーム上の Ultra Light データベースのパスとファイル名を指定します。

変更子とタイプ	プロパティ	説明
public string	DatabaseOnDevice [194 ページ]	Windows Mobile 上の Ultra Light データベースのパスとファイル名を指定します。
public string	EncryptionKey [195 ページ]	データベースを暗号化するためのキーを指定します。
public string	パスワード [195 ページ]	認証済みユーザのパスワードを指定します。
public string	UserID [196 ページ]	データベースで認証されるユーザを指定します。

備考

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

ULConnectionParms オブジェクトは、接続を開く (ULConnection.Open メソッドで)、またはデータベースを削除する (ULDatabaseManager.DropDatabase メソッドで) パラメータを指定するために使用します。

前後のスペースは、すべての値で無視されます。値には、前後のスペースやセミコロンを含めることはできません。また、一重引用符または二重引用符で始めることはできません。

接続文字列を作成するときは、データベースを識別し、オプションの接続設定を指定する必要があります。ULConnectionParms オブジェクトで適切なプロパティを設定して、接続パラメータをすべて指定したら、ULConnectionParms.ToString メソッドを使用して接続文字列を作成します。作成した文字列は、ULConnection(String) コンストラクタで新しい ULConnection オブジェクトを作成したり、既存の ULConnection オブジェクトの ULConnection.ConnectionString プロパティを設定したりするのに使用されます。

データベースの識別

各インスタンスには、データベースへのプラットフォーム固有のパスがあります。実行されているプラットフォームに対応する値だけが使用されます。たとえば、次のコードでは、パス ¥UltraLite¥mydb1.udb は Windows Mobile で使用され、mydb2.udb はその他のプラットフォームで使用されます。

```
' Visual Basic
Dim dbName As ULConnectionParms = new ULConnectionParms
dbName.DatabaseOnDevice = "¥UltraLite¥mydb1.udb"
dbName.DatabaseOnDesktop = "somedir¥mydb2.udb"
```

対応する C# 言語のコードを次に示します。

```
// C#
ULConnectionParms dbName = new ULConnectionParms();
dbName.DatabaseOnDevice = "¥UltraLite¥mydb1.udb";
dbName.DatabaseOnDesktop = "somedir¥mydb2.udb";
```

Ultra Light データベースファイルに推奨されるファイル拡張子は .udb です。Windows Mobile デバイスでは、デフォルトのデータベースは ¥UltraLiteDB¥ulstore.udb です。その他の Windows プラットフォームでは、デフォルトのデータベースは ulstore.udb です。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。

複数のデータベースを使用している場合は、各データベースのデータベース名を指定する必要があります。

オプションの接続設定

アプリケーションでの必要性やデータベースの作成方法によっては、デフォルト以外の `ULConnectionParms.UserID` 値、デフォルト以外の `ULConnectionParms.Password` 値、データベースの `ULConnectionParms.EncryptionKey` 値、接続の `ULConnectionParms.CacheSize` 値を指定することが必要になる場合があります。複数の接続を使用するアプリケーションでは、各接続に対してユニークな `ULConnectionParms.ConnectionName` 値を指定しなければなりません。

データベースは、1人の認証済みユーザ DBA で作成されます。このユーザの最初のパスワードは `sql` です。デフォルトでは、ユーザ ID DBA とパスワード `sql` を使用して、接続が開かれます。デフォルトのユーザを無効にするには、`ULConnection.RevokeConnectFrom` メソッドを使用します。ユーザを追加したりユーザのパスワードを変更するには、`ULConnection.GrantConnectTo` メソッドを使用します。

データベースを作成したときに暗号化キーを指定した場合は、そのデータベースへの以後の接続では、すべて同じ暗号化キーを使用する必要があります。データベースの暗号化キーを変更するには、`ULConnection.ChangeEncryptionKey` メソッドを使用します。

このセクションの内容:

[ULConnectionParms\(\) コンストラクタ \[190 ページ\]](#)

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

[ToString\(\) メソッド \[190 ページ\]](#)

このインスタンスの文字列表現を返します。

[AdditionalParms プロパティ \[191 ページ\]](#)

"名前=値" のペアをセミコロンで区切ったリストで、追加のパラメータを指定します。

[CacheSize プロパティ \[192 ページ\]](#)

キャッシュのサイズを指定します。

[ConnectionName プロパティ \[193 ページ\]](#)

接続の名前を指定します。

[DatabaseOnDesktop プロパティ \[194 ページ\]](#)

Windows デスクトッププラットフォーム上の Ultra Light データベースのパスとファイル名を指定します。

[DatabaseOnDevice プロパティ \[194 ページ\]](#)

Windows Mobile 上の Ultra Light データベースのパスとファイル名を指定します。

[EncryptionKey プロパティ \[195 ページ\]](#)

データベースを暗号化するためのキーを指定します。

[Password プロパティ \[195 ページ\]](#)

認証済みユーザのパスワードを指定します。

[UserID プロパティ \[196 ページ\]](#)

データベースで認証されるユーザを指定します。

関連情報

[Open\(\) メソッド \[163 ページ\]](#)

[DropDatabase\(string\) メソッド \[258 ページ\]](#)

[ToString\(\) メソッド \[190 ページ\]](#)

[ULConnection クラス \[121 ページ\]](#)

[ULConnection クラス \[121 ページ\]](#)
[ConnectionString プロパティ \[177 ページ\]](#)
[UserID プロパティ \[196 ページ\]](#)
[Password プロパティ \[195 ページ\]](#)
[EncryptionKey プロパティ \[195 ページ\]](#)
[CacheSize プロパティ \[192 ページ\]](#)
[ConnectionName プロパティ \[193 ページ\]](#)
[AdditionalParms プロパティ \[191 ページ\]](#)
[RevokeConnectFrom\(string\) メソッド \[165 ページ\]](#)
[GrantConnectTo\(string, string\) メソッド \[162 ページ\]](#)
[ChangeEncryptionKey\(string\) メソッド \[140 ページ\]](#)

1.8.1 ULConnectionParms() コンストラクタ

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

構文

Visual Basic

```
Public Sub ULConnectionParms ()
```

C#

```
public ULConnectionParms ()
```

1.8.2 ToString() メソッド

このインスタンスの文字列表現を返します。

構文

Visual Basic

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

C#

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

戻り値

"キーワード=値" の組み合わせがセミコロンで区切られたリスト形式の、このインスタンスの文字列表現。

1.8.3 AdditionalParms プロパティ

"名前=値" のペアをセミコロンで区切ったリストで、追加のパラメータを指定します。

構文

Visual Basic

```
Public Property AdditionalParms As String
```

C#

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

備考

ここで指定されるのは、使用頻度の低いパラメータです。

"キーワード=値" をセミコロンで区切ったリストによる追加のパラメータ。keyword=value リストの値は、ULConnection.ConnectionString のルールに従う必要があります。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

ページサイズや予約サイズのパラメータの値は、バイト単位で指定します。キロバイトの単位を示すにはサフィックス k または K を使用し、メガバイトの単位を示すにはサフィックス m または M を使用します。

追加のパラメータを次に示します。

キーワード	説明
dbn	接続する必要がある、ロードしたデータベースを識別します。データベースが起動されると、データベース名が割り当てられます。データベース名は、dbn パラメータを使用して明示的に割り当てられるか、または拡張子とパスが削除されたベースファイル名を使用して Ultra Light によって割り当てられます。接続が開かれると、Ultra Light は dbn 値が一致する実行中のデータベースをまず検索します。一致するデータベースが見つからない場合、Ultra Light は適切なデータベースのファイル名のパラメータ (DatabaseOnDevice プロパティまたは DatabaseOnDesktop プロパティを持つ) を使用して、新しいデータベースを起動します。このパラメータが必要なのは、同じベースファイル名を持つ 2 つの異なるデータベースにアプリケーション (または Ultra Light エンジン) がアクセスする場合です。このパラメータが使用されるのは、ULConnection.Open メソッドを使用して接続を開いたときだけです。

キーワード	説明
reserve_size	<p>Ultra Light の永続的データの保管に使用する、ファイルシステム領域を予約します。reserve_size パラメータを使用すると、データを挿入することなく、Ultra Light データベースが必要とするファイルシステム領域を事前に割り付けることができます。ファイルシステムの領域を予約すると、パフォーマンスが多少向上し、メモリが不足するという障害を防ぐことができます。デフォルトでは、永続ストアファイルのサイズは、アプリケーションがデータベースを更新して、サイズを大きくする必要が生じた場合にだけ大きくなります。reserve_size パラメータで予約されるファイルシステム領域には、未加工データだけでなく、永続ストアファイルのメタデータも含まれます。データベースのデータ量から、必要なファイルシステム領域を計算する場合は、メタデータのオーバーヘッドとデータの圧縮を考慮してください。</p> <p>reserve_size パラメータは、起動時に永続ストアファイルを設定された予約サイズまで大きくすることにより、領域を予約します。これは、そのファイルが以前に存在していたかどうかに関係なく行われます。このファイルはトランケートされません。パラメータ文字列 createParms.AdditionalParms = "reserve_size=2m" で、起動時に継続保管ファイルが少なくとも 2 MB であることが保証されます。このパラメータは、ULConnection.Open メソッドで接続開くときにのみ使用されます。</p>
start	<p>ロケーションを指定して、Ultra Light エンジンを開始します。現在実行中でないエンジンに接続する場合は、StartLine (START) 接続パラメータだけを指定します。ロケーションの指定は、Ultra Light エンジンがシステムパスに登録されていない場合にのみ必要です。</p>

関連情報

[RuntimeType プロパティ \[264 ページ\]](#)

[ConnectionString プロパティ \[177 ページ\]](#)

[DatabaseOnDevice プロパティ \[194 ページ\]](#)

[DatabaseOnDesktop プロパティ \[194 ページ\]](#)

[Open\(\) メソッド \[163 ページ\]](#)

1.8.4 CacheSize プロパティ

キャッシュのサイズを指定します。

構文

Visual Basic

```
Public Property CacheSize As String
```

C#

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


備考

キャッシュサイズを指定する文字列。デフォルトは NULL 参照 (Visual Basic の Nothing) です。この場合、デフォルトの 16 ページが使用されます。

キャッシュサイズの値は、バイト単位で指定します。キロバイトの単位を示すにはサフィックス k または K を使用し、メガバイトの単位を示すにはサフィックス m または M を使用します。

たとえば、次の例ではキャッシュサイズが 128 KB に設定されます。

```
connParms.CacheSize = "128k"
```

デフォルトのキャッシュサイズは 16 ページです。デフォルトのページサイズは 4 KB なので、デフォルトのキャッシュサイズは 64 KB です。キャッシュの最小サイズは、プラットフォームによって異なります。

デフォルトのキャッシュサイズは小さめの値です。テスト結果により、パフォーマンスの向上が必要と判断した場合は、キャッシュサイズを大きくしてください。

キャッシュサイズをデータベース自体のサイズよりも大きくすると、パフォーマンスは向上しません。また、キャッシュサイズが大きいと、その他の使用可能なアプリケーションの数が少なくなることがあります。

キャッシュサイズが未指定であるか、正しく指定されていない場合は、デフォルトのサイズが使用されます。

1.8.5 ConnectionName プロパティ

接続の名前を指定します。

構文

Visual Basic

```
Public Property ConnectionName As String
```

C#

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

備考

接続名が必要となるのは、データベースとの接続を複数作成する場合だけです。

接続の名前を指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

1.8.6 DatabaseOnDesktop プロパティ

Windows デスクトッププラットフォーム上の Ultra Light データベースのパスとファイル名を指定します。

構文

Visual Basic

```
Public Property DatabaseOnDesktop As String
```

C#

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

備考

データベースへの絶対パスまたは相対パスを指定する文字列。値が NULL 参照 (Visual Basic の Nothing) の場合、データベース ulstore.udb が使用されます。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

1.8.7 DatabaseOnDevice プロパティ

Windows Mobile 上の Ultra Light データベースのパスとファイル名を指定します。

構文

Visual Basic

```
Public Property DatabaseOnDevice As String
```

C#

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

備考

データベースへのフルパスを指定する文字列。値が NULL 参照 (Visual Basic の Nothing) の場合、データベース ¥UltraLiteDB¥ulstore.udb が使用されます。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

1.8.8 EncryptionKey プロパティ

データベースを暗号化するためのキーを指定します。

構文

Visual Basic

```
Public Property EncryptionKey As String
```

C#

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

備考

暗号化キーを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) で、暗号化は行われません。

すべての接続で、データベース作成時に指定されたキーと同じキーを使用する必要があります。キーを忘れた場合はデータベースにまったくアクセスできなくなります。

すべてのパスワードに共通することですが、最善の方法は、簡単には推測できないキー値を選択することです。キーの長さは任意ですが、短いと推測されやすいため、一般的には長い方が適しています。数字、文字、特殊文字を組み合わせると、キーは推測されにくくなります。

関連情報

[ChangeEncryptionKey\(string\) メソッド \[140 ページ\]](#)

1.8.9 Password プロパティ

認証済みユーザのパスワードを指定します。

構文

Microsoft Visual Basic

```
Public Property Password As String
```

C#

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

備考

データベースのユーザ ID を指定する文字列。デフォルト値は NULL 参照 (Microsoft Visual Basic の Nothing) です。
パスワードは、大文字と小文字が区別されます。

関連情報

[UserID プロパティ \[196 ページ\]](#)

1.8.10 UserID プロパティ

データベースで認証されるユーザを指定します。

構文

Visual Basic

```
Public Property UserID As String
```

C#

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

備考

データベースのユーザ ID を指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

ユーザ ID では大文字と小文字が区別されません。

データベースは最初、DBA という名前の 1 人の認証済みユーザで作成されます。

ユーザ ID とパスワードを両方とも入力しないと、ユーザ名 DBA とパスワード SQL が使用されます。データベースをより強力に保護するには、ユーザ名 DBA のパスワードを変更するか、新しいユーザを作成して (ULConnection.GrantConnectTo メソッドで)、DBA ユーザを削除します (ULConnection.RevokeConnectFrom メソッドで)。

関連情報

[Password プロパティ \[195 ページ\]](#)

[GrantConnectTo\(string, string\) メソッド \[162 ページ\]](#)

[RevokeConnectFrom\(string\) メソッド \[165 ページ\]](#)

1.9 ULConnectionStringBuilder クラス

Ultra Light データベースへの接続を開く接続文字列を作成します。

構文

Visual Basic

```
Public NotInheritable Class ULConnectionStringBuilder Inherits  
System.Data.Common.DbConnectionStringBuilder
```

C#

```
public sealed class ULConnectionStringBuilder :  
System.Data.Common.DbConnectionStringBuilder
```

メンバー

ULConnectionStringBuilder 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	ULConnectionStringBuilder [201 ページ]	ULConnectionStringBuilder オブジェクトを、そのデフォルト値で初期化します。

メソッド

変更子とタイプ	メソッド	説明
public override bool	ContainsKey(string) [202 ページ]	ULConnectionStringBuilder オブジェクトに特定のキーワードが含まれているかどうかを判断します。
public override bool	EquivalentTo(DbConnectionStringBuilder) [203 ページ]	この ULConnectionStringBuilder オブジェクト内の接続情報と指定された DbConnectionStringBuilder オブジェクト内の接続情報を比較します。
public static string	GetShortName(string) [203 ページ]	指定されたキーワードの短縮バージョンを取得します。
public override bool	Remove(string) [204 ページ]	指定されたキーが設定されたエントリを ULConnectionStringBuilder オブジェクトから削除します。
public override bool	TryGetValue(string, out Object) [205 ページ]	入力されたキーに対応する値を、この ULConnectionStringBuilder オブジェクトから取り出します。

プロパティ

変更子とタイプ	プロパティ	説明
public string	CacheSize [206 ページ]	UL 拡張: キャッシュのサイズを指定します。
public string	ConnectionName [206 ページ]	接続の名前を指定します。
public string	DatabaseKey [207 ページ]	データベースを暗号化するためのキーを指定します。
public string	DatabaseName [208 ページ]	データベース名、または接続する必要があるロードしたデータベースの名前を指定します。
public string	DatabaseOnDesktop [208 ページ]	UL 拡張: Windows デスクトッププラットフォーム上の Ultra Light データベースのパスとファイル名を指定します。
public string	DatabaseOnDevice [209 ページ]	UL 拡張: Windows Mobile 上の Ultra Light データベースのパスとファイル名を指定します。
public string	OrderedTableScans [209 ページ]	ORDER BY 句を指定しない SQL クエリで、デフォルトで順序付けされたテーブルスキャンを実行するかどうかを指定します。
public string	パスワード [210 ページ]	認証済みユーザのパスワードを指定します。
public string	ReserveSize [211 ページ]	UL 拡張: Ultra Light の永続的データの保管に使用する予約ファイルシステム領域を指定します。
public string	StartLine [212 ページ]	ロケーションを指定して、Ultra Light エンジンを開始します。
public override object	this[string keyword] [212 ページ]	指定された接続キーワードの値を指定します。
public string	UserID [213 ページ]	データベースで認証されるユーザを指定します。

備考

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

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

ULConnectionStringBuilder オブジェクトは、接続を開く (ULConnection.Open メソッドで)、またはデータベースを削除する (ULDatabaseManager.DropDatabase メソッドで) パラメータを指定するために使用します。

前後のスペースは、すべての値で無視されます。値には、前後のスペースやセミコロンを含めることはできません。また、一重引用符または二重引用符で始めることはできません。

接続文字列を作成するときは、データベースを識別し、オプションの接続設定を指定する必要があります。

ULConnectionStringBuilder オブジェクトで適切なプロパティを設定して、接続パラメータをすべて指定したら、System.Data.Common.DbConnectionStringBuilder.ConnectionString を使用して接続文字列を作成します。作成した文字列は、ULConnection(String) コンストラクタで新しい ULConnection オブジェクトを作成したり、既存の ULConnection オブジェクトの ULConnection.ConnectionString プロパティを設定したりするのに使用されます。

データベースの識別

各インスタンスには、データベースへのプラットフォーム固有のパスがあります。実行されているプラットフォームに対応する値だけが使用されます。たとえば、次のコードでは、パス ¥UltraLite¥mydb1.udb は Windows Mobile で使用され、mydb2.udb はその他のプラットフォームで使用されます。

```
' Visual Basic
Dim dbName As ULConnectionStringBuilder = _
    new ULConnectionStringBuilder
dbName.DatabaseOnDevice = "¥UltraLite¥mydb1.udb"
dbName.DatabaseOnDesktop = "somedir¥mydb2.udb"
```

対応する C# 言語のコードを次に示します。

```
// C#
ULConnectionStringBuilder dbName = new ULConnectionStringBuilder();
dbName.DatabaseOnDevice = "¥UltraLite¥mydb1.udb";
dbName.DatabaseOnDesktop = @"somedir¥mydb2.udb";
```

Ultra Light データベースファイルに推奨されるファイル拡張子は .udb です。Windows Mobile デバイスでは、デフォルトのデータベースは ¥UltraLiteDB¥ulstore.udb です。その他の Windows プラットフォームでは、デフォルトのデータベースは ulstore.udb です。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。

複数のデータベースを使用している場合は、各データベースのデータベース名を指定する必要があります。

オプションの接続設定

アプリケーションでの必要性やデータベースの作成方法によっては、デフォルト以外の ULConnectionStringBuilder.UserID 値、デフォルト以外の ULConnectionStringBuilder.Password 値、データベースの ULConnectionStringBuilder.EncryptionKey 値、接続の ULConnectionStringBuilder.CacheSize 値を指定することが必要になる場合があります。複数の接続を使用するアプリケーションでは、各接続に対してユニークな ULConnectionStringBuilder.ConnectionName 値を指定しなければなりません。

データベースは、1 人の認証済みユーザ DBA で作成されます。このユーザの最初のパスワードは sql です。デフォルトでは、ユーザ ID DBA とパスワード sql を使用して、接続が開かれます。デフォルトのユーザを無効にするには、ULConnection.RevokeConnectFrom メソッドを呼び出します。ユーザを追加したりユーザのパスワードを変更するには、ULConnection.GrantConnectTo メソッドを呼び出します。

データベースを作成したときに暗号化キーを指定した場合は、そのデータベースへの以後の接続では、すべて同じ暗号化キーを使用する必要があります。データベースの暗号化キーを変更するには、ULConnection.ChangeEncryptionKey メソッドを使用します。

このセクションの内容:

[ULConnectionStringBuilder コンストラクタ \[201 ページ\]](#)

ULConnectionStringBuilder オブジェクトを、そのデフォルト値で初期化します。

[ContainsKey\(string\) メソッド \[202 ページ\]](#)

ULConnectionStringBuilder オブジェクトに特定のキーワードが含まれているかどうかを判断します。

[EquivalentTo\(DbConnectionStringBuilder\) メソッド \[203 ページ\]](#)

この ULConnectionStringBuilder オブジェクト内の接続情報と指定された DbConnectionStringBuilder オブジェクト内の接続情報を比較します。

[GetShortName\(string\) メソッド \[203 ページ\]](#)

指定されたキーワードの短縮バージョンを取得します。

[Remove\(string\) メソッド \[204 ページ\]](#)

指定されたキーが設定されたエントリを ULConnectionStringBuilder オブジェクトから削除します。

[TryGetValue\(string, out Object\) メソッド \[205 ページ\]](#)

入力されたキーに対応する値を、この ULConnectionStringBuilder オブジェクトから取り出します。

[CacheSize プロパティ \[206 ページ\]](#)

UL 拡張: キャッシュのサイズを指定します。

[ConnectionString プロパティ \[206 ページ\]](#)

接続の名前を指定します。

[DatabaseKey プロパティ \[207 ページ\]](#)

データベースを暗号化するためのキーを指定します。

[DatabaseName プロパティ \[208 ページ\]](#)

データベース名、または接続する必要のあるロードしたデータベースの名前を指定します。

[DatabaseOnDesktop プロパティ \[208 ページ\]](#)

UL 拡張: Windows デスクトッププラットフォーム上の Ultra Light データベースのパスとファイル名を指定します。

[DatabaseOnDevice プロパティ \[209 ページ\]](#)

UL 拡張: Windows Mobile 上の Ultra Light データベースのパスとファイル名を指定します。

[OrderedTableScans プロパティ \[209 ページ\]](#)

ORDER BY 句を指定しない SQL クエリで、デフォルトで順序付けされたテーブルスキャンを実行するかどうかを指定します。

[Password プロパティ \[210 ページ\]](#)

認証済みユーザのパスワードを指定します。

[ReserveSize プロパティ \[211 ページ\]](#)

UL 拡張: Ultra Light の永続的データの保管に使用する予約ファイルシステム領域を指定します。

[StartLine プロパティ \[212 ページ\]](#)

ロケーションを指定して、Ultra Light エンジンを開始します。

[this\[string keyword\] プロパティ \[212 ページ\]](#)

指定された接続キーワードの値を指定します。

[UserID プロパティ \[213 ページ\]](#)

データベースで認証されるユーザを指定します。

関連情報

[Open\(\) メソッド \[163 ページ\]](#)

[DropDatabase\(string\) メソッド \[258 ページ\]](#)

[ULConnection クラス \[121 ページ\]](#)

[ULConnection クラス \[121 ページ\]](#)

[ConnectionString プロパティ \[177 ページ\]](#)

[DatabaseName プロパティ \[208 ページ\]](#)

- [UserID プロパティ \[213 ページ\]](#)
- [Password プロパティ \[210 ページ\]](#)
- [DatabaseKey プロパティ \[207 ページ\]](#)
- [CacheSize プロパティ \[206 ページ\]](#)
- [ConnectionName プロパティ \[206 ページ\]](#)
- [RevokeConnectFrom\(string\) メソッド \[165 ページ\]](#)
- [GrantConnectTo\(string, string\) メソッド \[162 ページ\]](#)
- [ChangeEncryptionKey\(string\) メソッド \[140 ページ\]](#)

1.9.1 ULConnectionStringBuilder コンストラクタ

ULConnectionStringBuilder オブジェクトを、そのデフォルト値で初期化します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public	ULConnectionStringBuilder() [201 ページ]	ULConnectionStringBuilder オブジェクトを、そのデフォルト値で初期化します。
public	ULConnectionStringBuilder(string) [202 ページ]	ULConnectionStringBuilder オブジェクトを、指定された接続文字列で初期化します。

このセクションの内容:

[ULConnectionStringBuilder\(\) コンストラクタ \[201 ページ\]](#)

ULConnectionStringBuilder オブジェクトを、そのデフォルト値で初期化します。

[ULConnectionStringBuilder\(string\) コンストラクタ \[202 ページ\]](#)

ULConnectionStringBuilder オブジェクトを、指定された接続文字列で初期化します。

1.9.1.1 ULConnectionStringBuilder() コンストラクタ

ULConnectionStringBuilder オブジェクトを、そのデフォルト値で初期化します。

構文

Visual Basic

```
Public Sub ULConnectionStringBuilder ()
```

C#

```
public ULConnectionStringBuilder ()
```

1.9.1.2 ULConnectionStringBuilder(string) コンストラクタ

ULConnectionStringBuilder オブジェクトを、指定された接続文字列で初期化します。

構文

Visual Basic

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

C#

```
public ULConnectionStringBuilder (string connectionString)
```

パラメータ

connectionString Ultra Light.NET の接続文字列。接続文字列は keyword=value のペアがセミコロンで区切られたリストです。

1.9.2 ContainsKey(string) メソッド

ULConnectionStringBuilder オブジェクトに特定のキーワードが含まれているかどうかを判断します。

構文

Visual Basic

```
Public Overrides Function ContainsKey (ByVal keyword As String) As Boolean
```

C#

```
public override bool ContainsKey (string keyword)
```

パラメータ

keyword 接続キーワードの名前。

戻り値

指定されたキーワードの値がこの接続文字列ビルダに含まれる場合は True、それ以外の場合は false を返します。

1.9.3 EquivalentTo(DbConnectionStringBuilder) メソッド

この ULConnectionStringBuilder オブジェクト内の接続情報と指定された DbConnectionStringBuilder オブジェクト内の接続情報を比較します。

構文

Visual Basic

```
Public Overrides Function EquivalentTo (ByVal connectionStringBuilder As DbConnectionStringBuilder) As Boolean
```

C#

```
public override bool EquivalentTo (DbConnectionStringBuilder connectionStringBuilder)
```

パラメータ

connectionStringBuilder この ULConnectionStringBuilder オブジェクトと比較する別の DbConnectionStringBuilder オブジェクト。

戻り値

このオブジェクトが指定された DbConnectionStringBuilder オブジェクトと同じ場合は True。それ以外の場合は false。

1.9.4 GetShortName(string) メソッド

指定されたキーワードの短縮バージョンを取得します。

構文

Visual Basic

```
Public Shared Function GetShortName (ByVal keyword As String) As String
```

C#

```
public static string GetShortName (string keyword)
```

パラメータ

keyword 取り出す項目のキー。

戻り値

キーワードが認識された場合は、指定されたキーワードの短縮バージョン。その他の場合は NULL。

1.9.5 Remove(string) メソッド

指定されたキーが設定されたエントリを ULConnectionStringBuilder オブジェクトから削除します。

構文

Visual Basic

```
Public Overrides Function Remove (ByVal keyword As String) As Boolean
```

C#

```
public override bool Remove (string keyword)
```

パラメータ

keyword 接続キーワードの名前。

戻り値

接続文字列内のキーが削除された場合は true、キーが存在しなかった場合は false。

1.9.6 TryGetValue(string, out Object) メソッド

入力されたキーに対応する値を、この ULConnectionStringBuilder オブジェクトから取り出します。

構文

Visual Basic

```
Public Overrides Function TryGetValue (
    ByVal keyword As String,
    ByVal value As Object
) As Boolean
```

C#

```
public override bool TryGetValue (
    string keyword,
    out Object value
)
```

パラメータ

keyword 取り出す項目のキー。

value キーに対応する値。

戻り値

キーワードが接続文字列内にある場合は true、それ以外は false。

備考

TryGetValue メソッドを使用すると、開発者は、安全に ULConnectionStringBuilder から値を取得できます。最初に ContainsKey メソッドを呼び出す必要はありません。TryGetValue メソッドは、存在しないキーを指定してこのメソッドを呼び出しても例外を発行しないため、値を取得する前にキーを検索する必要はありません。存在しないキーを指定して TryGetValue を呼び出すと、値パラメータに NULL 値 (Visual Basic の Nothing) が設定されます。

1.9.7 CacheSize プロパティ

UL 拡張: キャッシュのサイズを指定します。

構文

Visual Basic

```
Public Property CacheSize As String
```

C#

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

備考

キャッシュサイズを指定する文字列。デフォルトは NULL 参照 (Visual Basic の Nothing) です。この場合、デフォルトの 16 ページが使用されます。

キャッシュサイズの値は、バイト単位で指定します。キロバイトの単位を示すにはサフィックス k または K を使用し、メガバイトの単位を示すにはサフィックス m または M を使用します。

たとえば、次の例ではキャッシュサイズが 128 KB に設定されます。

```
connParms.CacheSize = "128k"
```

デフォルトのキャッシュサイズは 16 ページです。デフォルトのページサイズは 4 KB なので、デフォルトのキャッシュサイズは 64 KB です。キャッシュの最小サイズは、プラットフォームによって異なります。

デフォルトのキャッシュサイズは小さめの値です。テスト結果により、パフォーマンスの向上が必要と判断した場合は、キャッシュサイズを大きくしてください。

キャッシュサイズをデータベース自体のサイズよりも大きくすると、パフォーマンスは向上しません。また、キャッシュサイズが大きいと、その他の使用可能なアプリケーションの数が少なくなることがあります。

キャッシュサイズが未指定であるか、正しく指定されていない場合は、デフォルトのサイズが使用されます。

1.9.8 ConnectionName プロパティ

接続の名前を指定します。

構文

Visual Basic

```
Public Property ConnectionName As String
```

C#

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

備考

接続名が必要となるのは、データベースとの接続を複数作成する場合だけです。

接続の名前を指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

1.9.9 DatabaseKey プロパティ

データベースを暗号化するためのキーを指定します。

構文

Visual Basic

```
Public Property DatabaseKey As String
```

C#

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

備考

暗号化キーを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) で、暗号化は行われません。

すべての接続で、データベース作成時に指定されたキーと同じキーを使用する必要があります。キーを忘れた場合はデータベースにまったくアクセスできなくなります。

すべてのパスワードに共通することですが、最善の方法は、簡単には推測できないキー値を選択することです。キーの長さは任意ですが、短いと推測されやすいため、一般的には長い方が適しています。数字、文字、特殊文字を組み合わせると、キーは推測されにくくなります。

関連情報

[ChangeEncryptionKey\(string\) メソッド \[140 ページ\]](#)

1.9.10 DatabaseName プロパティ

データベース名、または接続する必要のあるロードしたデータベースの名前を指定します。

構文

Visual Basic

```
Public Property DatabaseName As String
```

C#

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

備考

データベースの名前を指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

データベースが起動されると、データベース名が割り当てられます。データベース名は、dbn パラメータを使用して明示的に割り当てられるか、または拡張子とパスが削除されたベースファイル名を使用して Ultra Light によって割り当てられます。

接続が開かれると、Ultra Light は dbn パラメータが一致する実行中のデータベースをまず検索します。一致するデータベースが見つからない場合、Ultra Light は適切なデータベースのファイル名のパラメータ (DatabaseOnDevice プロパティまたは DatabaseOnDesktop プロパティ) を使用して、新しいデータベースを起動します。

このパラメータが必要なのは、同じベースファイル名を持つ 2 つの異なるデータベースにアプリケーション (または Ultra Light エンジン) がアクセスする場合です。

関連情報

[DatabaseOnDevice プロパティ \[209 ページ\]](#)

[DatabaseOnDesktop プロパティ \[208 ページ\]](#)

1.9.11 DatabaseOnDesktop プロパティ

UL 拡張: Windows デスクトッププラットフォーム上の Ultra Light データベースのパスとファイル名を指定します。

構文

Visual Basic

```
Public Property DatabaseOnDesktop As String
```


C#

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

備考

データベースへの絶対パスまたは相対パスを指定する文字列。値が NULL 参照 (Visual Basic の Nothing) の場合、データベース ulstore.udb が使用されます。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

1.9.12 DatabaseOnDevice プロパティ

UL 拡張: Windows Mobile 上の Ultra Light データベースのパスとファイル名を指定します。

構文

Visual Basic

```
Public Property DatabaseOnDevice As String
```

C#

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

備考

データベースへのフルパスを指定する文字列。値が NULL 参照 (Visual Basic の Nothing) の場合、データベース ¥UltraLiteDB¥ulstore.udb が使用されます。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

1.9.13 OrderedTableScans プロパティ

ORDER BY 句を指定しない SQL クエリで、デフォルトで順序付けされたテーブルスキャンを実行するかどうかを指定します。

構文

Visual Basic

```
Public Property OrderedTableScans As String
```

C#

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

備考

順序付けされたテーブルスキャンを実行するかどうかを指定する boolean 文字列。たとえば、true と false、yes と no、1 と 0 などです。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

Ultra Light で動的 SQL を使用するとき、クエリ実行の順序は重要ではありません。Ultra Light は、プライマリキーインデックスを使用するのではなく、データベースページからローに直接アクセスします。これにより、ローをフェッチするパフォーマンスが改善されました。この最適化を使用するには、クエリが読み込み専用であり、すべてのローをスキャンする必要があります。

ローが特定の順序に並べることを期待する場合は、SQL クエリに ORDER BY 文を指定します。ただし、一部のアプリケーションでは、プライマリキーの順にローを返すようなデフォルトの動作に依存する可能性があります。このような場合は、OrderedTableScans パラメータを 1 (true、yes、on) に設定し、テーブル上で反復するときに以前の動作に戻すようにすることが必要です。

OrderedTableScans 値を 1 (true、yes、on) に設定しても、ORDER BY 句を指定しなかったり、クエリがインデックスを利用できなかったりすると、Ultra Light はプライマリキーを使用してデフォルトの動作を実行します。

1.9.14 Password プロパティ

認証済みユーザのパスワードを指定します。

構文

Visual Basic

```
Public Property Password As String
```

C#

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

備考

データベースのユーザ ID を指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

パスワードは、大文字と小文字が区別されます。

データベースが作成されると、ユーザ ID DBA のパスワードは sql に設定されます。

関連情報

[UserID プロパティ \[213 ページ\]](#)

1.9.15 ReserveSize プロパティ

UL 拡張: Ultra Light の永続的データの保管に使用する予約ファイルシステム領域を指定します。

構文

Visual Basic

```
Public Property ReserveSize As String
```

C#

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

備考

予約サイズを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

予約サイズパラメータの値は、バイト単位で指定します。キロバイトの単位を示すにはサフィックス k または K を使用し、メガバイトの単位を示すにはサフィックス m または M を使用します。

reserve_size パラメータを使用すると、データを挿入することなく、Ultra Light データベースが必要とするファイルシステム領域を事前に割り付けることができます。ファイルシステムの領域を予約すると、パフォーマンスが多少向上し、メモリが不足するという障害を防ぐことができます。デフォルトでは、永続ストアファイルのサイズは、アプリケーションがデータベースを更新して、サイズを大きくする必要が生じた場合にだけ大きくなります。

reserve_size で予約されるファイルシステム領域には、未加工データだけでなく、永続ストアファイルのメタデータも含まれます。データベースのデータ量から、必要なファイルシステム領域を計算する場合は、メタデータのオーバーヘッドとデータの圧縮を考慮してください。

reserve_size パラメータは、起動時に永続ストアファイルを設定された予約サイズまで大きくすることにより、領域を予約します。これは、そのファイルが以前に存在していたかどうかに関係なく行われます。このファイルはトランケートされません。

次のパラメータ文字列では、起動時に永続ストアファイルのサイズが最低 2 MB 確保されます。

```
connParams.ReserveSize = "2m"
```

1.9.16 StartLine プロパティ

ロケーションを指定して、Ultra Light エンジンを開始します。

構文

Visual Basic

```
Public Property StartLine As String
```

C#

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

備考

Ultra Light エンジンの実行可能ファイルのロケーションを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

現在実行中でないエンジンに接続する場合は、StartLine (START) 接続パラメータだけを指定します。

関連情報

[RuntimeType プロパティ \[264 ページ\]](#)

1.9.17 this[string keyword] プロパティ

指定された接続キーワードの値を指定します。

構文

Visual Basic

```
Public Overrides Property Item (ByVal keyword As String) As Object
```

C#

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

備考

指定された接続キーワードの値を表すオブジェクト。

接続キーワードと、ULConnectionStringBuilder クラスの対応するプロパティを次の表に示します。

キーワード	対応するプロパティ
cache_size	ULConnectionStringBuilder.CacheSize
ce_file	ULConnectionStringBuilder.DatabaseOnDevice
con	ULConnectionStringBuilder.ConnectionName
dbkey	ULConnectionStringBuilder.DatabaseKey
dbn	ULConnectionStringBuilder.DatabaseName
nt_file	ULConnectionStringBuilder.DatabaseOnDesktop
pwd	ULConnectionStringBuilder.Password
reserve_size	ULConnectionStringBuilder.ReserveSize
start	ULConnectionStringBuilder.StartLine
uid	ULConnectionStringBuilder.UserID

関連情報

- [CacheSize プロパティ \[206 ページ\]](#)
- [DatabaseOnDevice プロパティ \[209 ページ\]](#)
- [ConnectionName プロパティ \[206 ページ\]](#)
- [DatabaseKey プロパティ \[207 ページ\]](#)
- [DatabaseName プロパティ \[208 ページ\]](#)
- [DatabaseOnDesktop プロパティ \[208 ページ\]](#)
- [Password プロパティ \[210 ページ\]](#)
- [ReserveSize プロパティ \[211 ページ\]](#)
- [StartLine プロパティ \[212 ページ\]](#)
- [UserID プロパティ \[213 ページ\]](#)

1.9.18 UserID プロパティ

データベースで認証されるユーザを指定します。

構文

Visual Basic

```
Public Property UserID As String
```

C#

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

備考

データベースのユーザ ID を指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

ユーザ ID では大文字と小文字が区別されません。

データベースは最初、DBA という名前の 1 人の認証済みユーザで作成されます。

ユーザ ID とパスワードを両方とも入力しないと、ユーザ名 DBA とパスワード SQL が使用されます。データベースをより強力に保護するには、ユーザ名 DBA のパスワードを変更するか、新しいユーザを作成して (ULConnection.GrantConnectTo メソッドで)、DBA ユーザを削除します (ULConnection.RevokeConnectFrom メソッドで)。

関連情報

[Password プロパティ \[210 ページ\]](#)

[GrantConnectTo\(string, string\) メソッド \[162 ページ\]](#)

[RevokeConnectFrom\(string\) メソッド \[165 ページ\]](#)

1.10 ULCreateParms クラス

UL 拡張: Ultra Light データベースを作成するときの作成時オプションの文字列を作成します。

構文

Visual Basic

```
Public Class ULCreateParms
```

C#

```
public class ULCreateParms
```

メンバー

ULCreateParms 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	ULCreateParms() [218 ページ]	ULCreateParms オブジェクトを、そのデフォルト値で初期化します。

メソッド

変更子とタイプ	メソッド	説明
public override string	ToString() [218 ページ]	このインスタンスの文字列表現を返します。

プロパティ

変更子とタイプ	プロパティ	説明
public bool	CaseSensitive [219 ページ]	文字列を比較するときに、新しいデータベースで大文字と小文字を区別するかどうかを指定します。
public int	ChecksumLevel [219 ページ]	新しいデータベースで有効にするデータベースページのチェックサムのレベルを指定します。
public string	DateFormat [220 ページ]	新しいデータベースで文字列変換に使用される日付形式を指定します。
public ULDateOrder	DateOrder [220 ページ]	新しいデータベースで文字列変換に使用される日付順を指定します。
public bool	FIPS [221 ページ]	新しいデータベースで使用する暗号化 (AES_FIPS または AES) を指定します。
public int	MaxHashSize [221 ページ]	新しいデータベースでインデックスのハッシュに使用するデフォルトの最大バイト数を指定します。
public int	NearestCentury [222 ページ]	新しいデータベースで文字列変換に使用される最も近い世紀を指定します。
public bool	Obfuscate [222 ページ]	新しいデータベースが難読化を使用してデータベースを暗号化するかどうかを指定します。
public int	PageSize [223 ページ]	新しいデータベースのページサイズ (バイトまたはキロバイト単位) を指定します。
public int	Precision [223 ページ]	データベースで文字列変換に使用される浮動小数点の精度を指定します。
public int	Scale [224 ページ]	新しいデータベースによる文字列変換中に、計算結果が最大の精度でトランケートされる時の小数点以下の最大桁数を指定します。
public string	TimeFormat [224 ページ]	データベースで文字列変換に使用される時間形式を指定します。
public string	TimestampFormat [225 ページ]	データベースで文字列変換に使用されるタイムスタンプ形式を指定します。

変更子とタイプ	プロパティ	説明
public int	TimestampIncrement [225 ページ]	2つのユニークなタイムスタンプ間のマイクロ秒 (1,000,000 分の 1 秒) 単位の最小差を指定します。
public bool	UTF8Encoding [226 ページ]	新しいデータベースで使用する文字セット (UTF8 文字セットまたは照合に関連付けられた文字セット) を指定します。

備考

ULCreateParms オブジェクトを使用して、ULDatabaseManager.CreateDatabase メソッドでデータベースを作成するときのパラメータを指定します。

前後のスペースは、すべての値で無視されます。値には、前後のスペースやセミコロンを含めることはできません。また、一重引用符または二重引用符で始めることはできません。

ULCreateParms オブジェクトで適切なプロパティを設定して、作成パラメータをすべて指定したら、ULCreateParms.ToString メソッドを使用して作成パラメータ文字列を作成します。結果として返される文字列は ULDatabaseManager.CreateDatabase メソッドの createParms パラメータとして使用できます。

例

次のコードでは、Windows Mobile デバイス上にデータベース ¥UltraLite¥MyDatabase.udb を作成します。このデータベースは、大文字と小文字を区別し、UTF8 文字セットを使用して作成されます。

```
' Visual Basic
Dim createParms As ULCreateParms = New ULCreateParms
createParms.CaseSensitive = True
createParms.UTF8Encoding = True
Dim openParms As ULConnectionParms = New ULConnectionParms
openParms.DatabaseOnDevice = "¥UltraLite¥MyDatabase.udb"
ULConnection.DatabaseManager.CreateDatabase( _
    openParms.ToString(), _
    createParms.ToString() _
)
Dim conn As ULConnection = _
    New ULConnection( openParms.ToString() )
conn.Open()
```

対応する C# 言語のコードを次に示します。

```
// C#
ULCreateParms createParms = new ULCreateParms();
createParms.CaseSensitive = true;
createParms.UTF8Encoding = true;
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnDevice = ".udb";
ULConnection.DatabaseManager.CreateDatabase(
    openParms.ToString(),
    createParms.ToString()
);
ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```


このセクションの内容:

[ULCreateParms\(\) コンストラクタ \[218 ページ\]](#)

ULCreateParms オブジェクトを、そのデフォルト値で初期化します。

[ToString\(\) メソッド \[218 ページ\]](#)

このインスタンスの文字列表現を返します。

[CaseSensitive プロパティ \[219 ページ\]](#)

文字列を比較するときに、新しいデータベースで大文字と小文字を区別するかどうかを指定します。

[ChecksumLevel プロパティ \[219 ページ\]](#)

新しいデータベースで有効にするデータベースページのチェックサムのレベルを指定します。

[DateFormat プロパティ \[220 ページ\]](#)

新しいデータベースで文字列変換に使用される日付形式を指定します。

[DateOrder プロパティ \[220 ページ\]](#)

新しいデータベースで文字列変換に使用される日付順を指定します。

[FIPS プロパティ \[221 ページ\]](#)

新しいデータベースで使用する暗号化 (AES_FIPS または AES) を指定します。

[MaxHashSize プロパティ \[221 ページ\]](#)

新しいデータベースでインデックスのハッシュに使用するデフォルトの最大バイト数を指定します。

[NearestCentury プロパティ \[222 ページ\]](#)

新しいデータベースで文字列変換に使用される最も近い世紀を指定します。

[Obfuscate プロパティ \[222 ページ\]](#)

新しいデータベースが難読化を使用してデータベースを暗号化するかどうかを指定します。

[PageSize プロパティ \[223 ページ\]](#)

新しいデータベースのページサイズ (バイトまたはキロバイト単位) を指定します。

[Precision プロパティ \[223 ページ\]](#)

データベースで文字列変換に使用される浮動小数点の精度を指定します。

[Scale プロパティ \[224 ページ\]](#)

新しいデータベースによる文字列変換中に、計算結果が最大の精度でトランケートされるとき的小数点以下の最大桁数を指定します。

[TimeFormat プロパティ \[224 ページ\]](#)

データベースで文字列変換に使用される時間形式を指定します。

[TimestampFormat プロパティ \[225 ページ\]](#)

データベースで文字列変換に使用されるタイムスタンプ形式を指定します。

[TimestampIncrement プロパティ \[225 ページ\]](#)

2つのユニークなタイムスタンプ間のマイクロ秒 (1,000,000 分の 1 秒) 単位の最小差を指定します。

[UTF8Encoding プロパティ \[226 ページ\]](#)

新しいデータベースで使用する文字セット (UTF8 文字セットまたは照合に関連付けられた文字セット) を指定します。

関連情報

[GetDatabaseProperty\(string\) メソッド \[267 ページ\]](#)

[CreateDatabase\(string, string\) メソッド \[257 ページ\]](#)

[ToString\(\) メソッド \[218 ページ\]](#)

1.10.1 ULCreateParms() コンストラクタ

ULCreateParms オブジェクトを、そのデフォルト値で初期化します。

構文

Visual Basic

```
Public Sub ULCreateParms ()
```

C#

```
public ULCreateParms ()
```

1.10.2 ToString() メソッド

このインスタンスの文字列表現を返します。

構文

Visual Basic

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

C#

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

戻り値

"キーワード=値" の組み合わせがセミコロンで区切られたリスト形式の、このインスタンスの文字列表現。

1.10.3 CaseSensitive プロパティ

文字列を比較するときに、新しいデータベースで大文字と小文字を区別するかどうかを指定します。

構文

Visual Basic

```
Public Property CaseSensitive As Boolean
```

C#

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

備考

データベースで大文字と小文字が区別される場合は true、区別されない場合は false。デフォルトは false です。

このメソッドは、文字列データの比較とソートの方法にのみ影響します。テーブル名、カラム名、インデックス名、接続ユーザ ID などのデータベース識別子については、どのような場合であっても、大文字と小文字は区別されません。接続用のパスワードとデータベース暗号化キーでは、常に、大文字と小文字が区別されます。

1.10.4 ChecksumLevel プロパティ

新しいデータベースで有効にするデータベースページのチェックサムのレベルを指定します。

構文

Visual Basic

```
Public Property ChecksumLevel As Integer
```

C#

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

備考

チェックサムのレベルを指定する整数。有効な値は 0、1、2 です。デフォルトは 0 です。

1.10.5 DateFormat プロパティ

新しいデータベースで文字列変換に使用される日付形式を指定します。

構文

Visual Basic

```
Public Property DateFormat As String
```

C#

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

備考

日付形式を指定する文字列。値が NULL 参照 (Visual Basic の Nothing) の場合、データベースでは "YYYY-MM-DD" が使用されます。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

1.10.6 DateOrder プロパティ

新しいデータベースで文字列変換に使用される日付順を指定します。

構文

Visual Basic

```
Public Property DateOrder As UDateOrder
```

C#

```
public UDateOrder DateOrder {get;set;}
```

備考

文字列変換に使用される日付順を指定する UDateOrder 値。デフォルト値は YMD です。

関連情報

[UDateOrder 列挙体 \[586 ページ\]](#)

1.10.7 FIPS プロパティ

新しいデータベースで使用する暗号化 (AES_FIPS または AES) を指定します。

構文

Visual Basic

```
Public Property FIPS As Boolean
```

C#

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

備考

データベースで AES_FIPS を使用して暗号化される場合は true、AES を使用して暗号化される場合は false。デフォルトは false です。

新しいデータベースが作成されたときは、EncryptionKey 接続パラメータの値を指定して暗号化を有効にする必要があります。FIPS が true で、暗号化キーを指定していない場合、ULDatabaseManager.CreateDatabase メソッドは、暗号化キーが見つからないために失敗します。

関連情報

[EncryptionKey プロパティ \[195 ページ\]](#)

[CreateDatabase\(string, string\) メソッド \[257 ページ\]](#)

1.10.8 MaxHashSize プロパティ

新しいデータベースでインデックスのハッシュに使用するデフォルトの最大バイト数を指定します。

構文

Visual Basic

```
Public Property MaxHashSize As Integer
```

C#

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

備考

最大ハッシュサイズを指定する整数。値は、[0,32] の範囲内である必要があります。デフォルトは 8 です。

1.10.9 NearestCentury プロパティ

新しいデータベースで文字列変換に使用される最も近い世紀を指定します。

構文

Visual Basic

```
Public Property NearestCentury As Integer
```

C#

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

備考

最も近い世紀を指定する整数。値は、[0,100] の範囲内である必要があります。デフォルトは 50 です。

1.10.10 Obfuscate プロパティ

新しいデータベースが難読化を使用してデータベースを暗号化するかどうかを指定します。

構文

Visual Basic

```
Public Property Obfuscate As Boolean
```

C#

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

備考

データベースで難読化を使用して暗号化される場合は true、難読化されない場合は false。デフォルトは false です。

FIPS 暗号化が `ULCreateParms.FIPS` プロパティで有効になっている場合は、このオプションは無視されます。難読化が有効になっている場合に、新しいデータベースの作成時に `EncryptionKey` 接続パラメータに値を指定すると、暗号化キーは無視されます。

関連情報

[FIPS プロパティ \[221 ページ\]](#)

1.10.11 PageSize プロパティ

新しいデータベースのページサイズ (バイトまたはキロバイト単位) を指定します。

構文

Visual Basic

```
Public Property PageSize As Integer
```

C#

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

備考

ページサイズ (バイト単位) を指定する整数。有効な値は、1024 (1K)、2048 (2K)、4096 (4K)、8192 (8K)、16384 (16K) です。デフォルトは 4096 です。

1.10.12 Precision プロパティ

データベースで文字列変換に使用される浮動小数点の精度を指定します。

構文

Visual Basic

```
Public Property Precision As Integer
```

C#

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

備考

精度を指定する整数。値は、[1,127] の範囲内である必要があります。デフォルトは 30 です。

関連情報

[Scale プロパティ \[224 ページ\]](#)

1.10.13 Scale プロパティ

新しいデータベースによる文字列変換中に、計算結果が最大の精度でトランケートされる時の小数点以下の最大桁数を指定します。

構文

Visual Basic

```
Public Property Scale As Integer
```

C#

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

備考

位取りを指定する整数。値は、[0,127] の範囲内である必要があります。デフォルトは 6 です。

Scale 値は Precision 値以下である必要があります。それ以外の場合は、データベースの作成時にエラーが発生します。

1.10.14 TimeFormat プロパティ

データベースで文字列変換に使用される時間形式を指定します。

構文

Visual Basic

```
Public Property TimeFormat As String
```

C#

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


備考

時間形式を指定する文字列。値が NULL 参照 (Visual Basic の Nothing) の場合、データベースでは "HH:NN:SS.SSS" が使用されます。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

1.10.15 TimestampFormat プロパティ

データベースで文字列変換に使用されるタイムスタンプ形式を指定します。

構文

Visual Basic

```
Public Property TimestampFormat As String
```

C#

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

備考

タイムスタンプ形式を指定する文字列。値が NULL 参照 (Visual Basic の Nothing) の場合、データベースでは "YYYY-MM-DD HH:NN:SS.SSS" が使用されます。C# では、パス内の円記号をエスケープするか、アットマーク (@) で囲まれた文字列リテラルを使用する必要があります。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

1.10.16 TimestampIncrement プロパティ

2 つのユニークなタイムスタンプ間のマイクロ秒 (1,000,000 分の 1 秒) 単位の最小差を指定します。

構文

Visual Basic

```
Public Property TimestampIncrement As Integer
```

C#

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

備考

タイムスタンプのインクリメントを指定する整数。値は、[1,60000000] の範囲内である必要があります。デフォルトは 1 です。

1.10.17 UTF8Encoding プロパティ

新しいデータベースで使用する文字セット (UTF8 文字セットまたは照合に関連付けられた文字セット) を指定します。

構文

Visual Basic

```
Public Property UTF8Encoding As Boolean
```

C#

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

備考

データベースで UTF8 文字セットが使用される場合は true、照合に関連付けられた文字セットが使用される場合は false。デフォルトは false です。

照合に関連付けられた文字セット以外の文字セットに文字を格納する場合は、UTF8 文字セットを選択します。たとえば、US ソートを使用したいけれども、現地の綴りで国際住所を格納したいために UTF8Encoding を true に設定している場合は、1252LATIN1 照合を使用してデータベースを作成します。

1.11 ULCursorSchema クラス

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

構文

Visual Basic

```
Public MustInherit Class ULCursorSchema
```

C#

```
public abstract class ULCursorSchema
```

メンバー

ULCursorSchema 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
protected unsafe void	GetColumnCount() [229 ページ]	
public unsafe short	GetColumnID(string) [229 ページ]	指定されたカラムのカラム ID を返します。
public string	GetColumnName(int) [230 ページ]	指定されたカラム ID で識別されたカラムの名前を返します。
public unsafe int	GetColumnPrecision(int) [231 ページ]	カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの精度を返します。
public unsafe int	GetColumnScale(int) [232 ページ]	カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの位取りを返します。
public unsafe int	GetColumnSize(int) [233 ページ]	カラムがサイズ指定されたカラム (BINARY SQL 型または CHAR SQL 型) の場合は、指定されたカラム ID によって識別されたカラムのサイズを返します。
public string	GetColumnSQLName(int) [234 ページ]	指定されたカラム ID で識別されたカラムの名前を返します。
public unsafe ULDbType	GetColumnULDbType(int) [235 ページ]	指定されたカラム ID によって識別されたカラムの Ultra Light.NET データ型を返します。
public unsafe DataTable	GetSchemaTable() [236 ページ]	ULDataReader オブジェクトのカラムのスキーマが記述された System.Data.DataTable を返します。
protected virtual void	VerifyOpen() [236 ページ]	

プロパティ

変更子とタイプ	プロパティ	説明
public short	ColumnCount [237 ページ]	このカーソル内のカラム数を返します。
public bool	IsOpen [237 ページ]	カーソルのスキーマが現在開いているかどうかを確認します。
public abstract string	Name [238 ページ]	カーソルの名前を返します。

備考

このクラスは、ULTableSchema クラスと ULResultSetSchema クラスの抽象基本クラスです。

i 注記

Sap.UltraLite ネームスペースから移行するユーザの場合、カラム ID は、Sap.UltraLite ネームスペース内にあるため、1 ではなく 0 から始まります。

このセクションの内容:

[GetColumnCount\(\) メソッド \[229 ページ\]](#)

[GetColumnID\(string\) メソッド \[229 ページ\]](#)

指定されたカラムのカラム ID を返します。

[GetColumnName\(int\) メソッド \[230 ページ\]](#)

指定されたカラム ID で識別されたカラムの名前を返します。

[GetColumnPrecision\(int\) メソッド \[231 ページ\]](#)

カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの精度を返します。

[GetColumnScale\(int\) メソッド \[232 ページ\]](#)

カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの位取りを返します。

[GetColumnSize\(int\) メソッド \[233 ページ\]](#)

カラムがサイズ指定されたカラム (BINARY SQL 型または CHAR SQL 型) の場合は、指定されたカラム ID によって識別されたカラムのサイズを返します。

[GetColumnSQLName\(int\) メソッド \[234 ページ\]](#)

指定されたカラム ID で識別されたカラムの名前を返します。

[GetColumnULDbType\(int\) メソッド \[235 ページ\]](#)

指定されたカラム ID によって識別されたカラムの Ultra Light.NET データ型を返します。

[GetSchemaTable\(\) メソッド \[236 ページ\]](#)

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

[VerifyOpen\(\) メソッド \[236 ページ\]](#)

[ColumnCount プロパティ \[237 ページ\]](#)

このカーソル内のカラム数を返します。

[IsOpen プロパティ \[237 ページ\]](#)

カーソルのスキーマが現在開いているかどうかを確認します。

[Name プロパティ \[238 ページ\]](#)

カーソルの名前を返します。

関連情報

[ULTableSchema クラス \[555 ページ\]](#)

[ULResultSetSchema クラス \[471 ページ\]](#)

1.11.1 GetColumnCount() メソッド

構文

Visual Basic

```
Protected Sub GetColumnCount ()
```

C#

```
protected unsafe void GetColumnCount ()
```

1.11.2 GetColumnID(string) メソッド

指定されたカラムのカラム ID を返します。

構文

Visual Basic

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

C#

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

パラメータ

name カラム名。

戻り値

指定されたカラムのカラム ID。

例外

ULException class SQL エラーが発生しました。

備考

カラム ID の範囲は、0 ~ ColumnCount-1 です。

結果セットでは、すべてのカラムに名前があるとは限らず、すべてのカラム名がユニークであるとも限りません。エイリアスを使用していない場合は、非計算カラムの名前には、そのカラムの元となるテーブルの名前がプレフィクスとして付けられます。たとえば、MyTable.ID は、クエリ "SELECT ID FROM MyTable" の結果セットに含まれる唯一のカラムの名前です。

カラムの ID とカウントは、スキーマのアップグレード中に変更されることがあります。カラムを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後に再表示します。

関連情報

[ColumnCount プロパティ \[237 ページ\]](#)

[ColumnCount プロパティ \[237 ページ\]](#)

1.11.3 GetColumnName(int) メソッド

指定されたカラム ID で識別されたカラムの名前を返します。

構文

Visual Basic

```
Public Function GetColumnName (ByVal columnID As Integer) As String
```

C#

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

パラメータ

columnID カラムの ID。値は、[0,ColumnCount-1] の範囲内である必要があります。

戻り値

カラムの名前。カラムに名前がない場合は NULL 参照 (Visual Basic の Nothing)。SQL クエリでカラムのエイリアスが使用されている場合は、そのエイリアスが返されます。

例外

`ULException class` SQL エラーが発生しました。

備考

結果セットでは、すべてのカラムに名前があるとは限らず、すべてのカラム名がユニークであるとも限りません。エイリアスを使用していない場合は、非計算カラムの名前には、そのカラムの元となるテーブルの名前がプレフィクスとして付けられます。たとえば、`MyTable.ID` は、クエリ "SELECT ID FROM MyTable" の結果セットに含まれる唯一のカラムの名前です。

カラムの ID とカウントは、スキーマのアップグレード中に変更されることがあります。カラムを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後に再表示します。

関連情報

[ColumnCount プロパティ \[237 ページ\]](#)

[ColumnCount プロパティ \[237 ページ\]](#)

1.11.4 GetColumnPrecision(int) メソッド

カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの精度を返します。

構文

Visual Basic

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

C#

```
public unsafe int GetColumnPrecision (int columnID)
```

パラメータ

columnID カラムの ID。値は、`[0,ColumnCount-1]` の範囲内である必要があります。

戻り値

指定された数値カラムの精度。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetColumnULDbType\(int\) メソッド \[235 ページ\]](#)

[ColumnCount プロパティ \[237 ページ\]](#)

1.11.5 GetColumnScale(int) メソッド

カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの位取りを返します。

構文

Visual Basic

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

C#

```
public unsafe int GetColumnScale (int columnID)
```

パラメータ

columnID カラムの ID。値は、[0,ColumnCount-1] の範囲内である必要があります。

戻り値

指定された数値カラムの位取り。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetColumnULDbType\(int\) メソッド \[235 ページ\]](#)

[ColumnCount プロパティ \[237 ページ\]](#)

1.11.6 GetColumnSize(int) メソッド

カラムがサイズ指定されたカラム (BINARY SQL 型または CHAR SQL 型) の場合は、指定されたカラム ID によって識別されたカラムのサイズを返します。

構文

Visual Basic

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

C#

```
public unsafe int GetColumnSize (int columnID)
```

パラメータ

columnID カラムの ID。値は、[0,ColumnCount-1] の範囲内である必要があります。

戻り値

指定された、サイズ指定されたカラムのサイズ。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetColumnULDbType\(int\) メソッド \[235 ページ\]](#)

[ColumnCount プロパティ \[237 ページ\]](#)

1.11.7 GetColumnSQLName(int) メソッド

指定されたカラム ID で識別されたカラムの名前を返します。

構文

Visual Basic

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

C#

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

パラメータ

columnID カラムの ID。値は、[0,ColumnCount-1] の範囲内である必要があります。

戻り値

カラムの名前。カラムに名前がない場合は NULL 参照 (Visual Basic の Nothing)。SQL クエリでカラムのエイリアスが使用されている場合は、そのエイリアスが返されます。

例外

ULException class SQL エラーが発生しました。

備考

結果セットでは、すべてのカラムに名前があるとは限らず、すべてのカラム名がユニークであるとも限りません。エイリアスを使用している場合は、カラムの名前がエイリアスになります。

GetColumnSQLName メソッドは、GetColumnName メソッドとは異なります。これは、GetColumnSQLName メソッドは常に、非エイリアスカラム、非計算カラムの名前 (プレフィクスとしてテーブル名は付きません) のみを返すためです。この動作は、他の ADO.NET プロバイダの動作によく似ていますが、ユニークでない名前を生成する可能性が高くなります。

カラムの ID とカウントは、スキーマのアップグレード中に変更されることがあります。カラムを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後に再表示します。

関連情報

[ColumnCount プロパティ \[237 ページ\]](#)

[GetColumnName\(int\) メソッド \[230 ページ\]](#)

[ColumnCount プロパティ \[237 ページ\]](#)

1.11.8 GetColumnULDbType(int) メソッド

指定されたカラム ID によって識別されたカラムの Ultra Light.NET データ型を返します。

構文

Visual Basic

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

C#

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

パラメータ

columnID カラムの ID。値は、[0,ColumnCount-1] の範囲内である必要があります。

戻り値

ULDbType 列挙整数。

例外

ULException class SQL エラーが発生しました。

関連情報

[ColumnCount プロパティ \[237 ページ\]](#)

[ColumnCount プロパティ \[237 ページ\]](#)

[ULDbType 列挙体 \[586 ページ\]](#)

1.11.9 GetSchemaTable() メソッド

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

構文

Visual Basic

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

C#

```
public unsafe DataTable GetSchemaTable ()
```

戻り値

カラムのスキーマが記述された System.Data.DataTable。

関連情報

[GetSchemaTable\(\) メソッド \[304 ページ\]](#)

[ULDataReader クラス \[275 ページ\]](#)

1.11.10 VerifyOpen() メソッド

構文

Visual Basic

```
Protected Overridable Sub VerifyOpen ()
```

C#

```
protected virtual void VerifyOpen ()
```

1.11.11 ColumnCount プロパティ

このカーソル内のカラム数を返します。

構文

Visual Basic

```
Public ReadOnly Property ColumnCount As Short
```

C#

```
public short ColumnCount {get;}
```

備考

カーソル内のカラム数。カーソルのスキーマが閉じている場合は 0。

カラム ID の範囲は、0 ~ ColumnCount-1 です。

カラムの ID とカウントは、スキーマのアップグレード中に変更されることがあります。カラムを正しく識別するには、名前アクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後に再表示します。

1.11.12 IsOpen プロパティ

カーソルのスキーマが現在開いているかどうかを確認します。

構文

Visual Basic

```
Public ReadOnly Property IsOpen As Boolean
```

C#

```
public bool IsOpen {get;}
```

備考

カーソルのスキーマが現在開いている場合は true、閉じている場合は false。

1.11.13 Name プロパティ

カーソルの名前を返します。

構文

Visual Basic

```
Public ReadOnly Property Name As String
```

C#

```
public abstract string Name {get;}
```

備考

文字列として返されるカーソル名。

1.12 ULDataAdapter クラス

System.Data.DataSet に入力したりデータベースを更新したりするために使用する一連のコマンドとデータベース接続を表します。

構文

Visual Basic

```
Public NotInheritable Class ULDataAdapter Inherits  
System.Data.Common.DbDataAdapter Implements System.ICloneable
```

C#

```
public sealed class ULDataAdapter : System.Data.Common.DbDataAdapter,  
System.ICloneable
```

メンバー

ULDataAdapter 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	ULDataAdapter [241 ページ]	ULDataAdapter オブジェクトを初期化します。

メソッド

変更子とタイプ	メソッド	説明
protected override void	Dispose(bool) [245 ページ]	ULDataAdapter オブジェクトによって使用されるアンマネージリソースを解放するほか、必要に応じてマネージリソースを解放します。
protected override int	Fill [246 ページ]	個々のトピックを参照してください。
protected override DataTable[]	FillSchema [247 ページ]	個々のトピックを参照してください。
public new ULParameter[]	GetFillParameters() [249 ページ]	SELECT 文の実行時にユーザによって設定されるパラメータを返します。
protected override int	Update(DataRow[], DataTableMapping) [249 ページ]	

プロパティ

変更子とタイプ	プロパティ	説明
public new ULCommand	DeleteCommand [250 ページ]	System.Data.DataSet で削除されたローに該当するデータベース内のローを削除するために、DbDataAdapter.Update メソッドが呼び出されたときにデータベースに対して実行される ULCommand オブジェクトを指定します。
public new ULCommand	InsertCommand [251 ページ]	System.Data.DataSet で挿入されたローに該当するローをデータベースに挿入するために、DbDataAdapter.Update メソッドが呼び出されたときにデータベースに対して実行される ULCommand オブジェクトを指定します。
public new ULCommand	SelectCommand [251 ページ]	System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet) メソッド呼び出しまたは System.Data.Common.DbDataAdapter.FillSchema(System.Data.DataSet, System.Data.SchemaType) メソッド呼び出しの実行時に、System.Data.DataSet にコピーするための結果セットをデータベースから取得するために使用される ULCommand を指定します。
public new DataTableMappingCollection	TableMappings [252 ページ]	ソーステーブルと System.Data.DataTable 間のマスタマッピングを提供するコレクションを返します。

変更子とタイプ	プロパティ	説明
public new ULCommand	UpdateCommand [253 ページ]	System.Data.DataSet で更新されたローに該当するデータベース内のローを更新するために、System.Data.Common.DbDataAdapter.Update メソッドが呼び出されたときにデータベースに対して実行される ULCommand オブジェクトを指定します。

イベント

変更子とタイプ	イベント	説明
public ULRowUpdatedEventHandler	RowUpdated [254 ページ]	データソースに対してコマンドが実行された後の更新時に発生します。
public ULRowUpdatingEventHandler	RowUpdating [254 ページ]	データソースに対してコマンドが実行される前の更新時に発生します。

備考

System.Data.DataSet は、Ultra Light データベースとは別に、データをオフラインで処理するための方法を提供します。ULDataAdapter クラスは、System.Data.DataSet を一連の SQL 文に関連付けるためのメソッドを提供します。

Ultra Light はローカルのデータベースであり、Mobile Link には競合解決があるため、ULDataAdapter の使用は制限されています。ほとんどの場合、ULDataReader クラスまたは ULTable クラスを使用すると、より効率的にデータにアクセスできます。

このセクションの内容:

[ULDataAdapter コンストラクタ \[241 ページ\]](#)

ULDataAdapter オブジェクトを初期化します。

[Dispose\(bool\) メソッド \[245 ページ\]](#)

ULDataAdapter オブジェクトによって使用されるアンマネージリソースを解放するほか、必要に応じてマネージリソースを解放します。

[Fill メソッド \[246 ページ\]](#)

[FillSchema メソッド \[247 ページ\]](#)

[GetFillParameters\(\) メソッド \[249 ページ\]](#)

SELECT 文の実行時にユーザによって設定されるパラメータを返します。

[Update\(DataRow\[\], DataTableMapping\) メソッド \[249 ページ\]](#)

[DeleteCommand プロパティ \[250 ページ\]](#)

System.Data.DataSet で削除されたローに該当するデータベース内のローを削除するために、DbDataAdapter.Update メソッドが呼び出されたときにデータベースに対して実行される ULCommand オブジェクトを指定します。

[InsertCommand プロパティ \[251 ページ\]](#)

System.Data.DataSet で挿入されたローに該当するローをデータベースに挿入するために、DbDataAdapter.Update メソッドが呼び出されたときにデータベースに対して実行される ULCommand オブジェクトを指定します。

SelectCommand プロパティ [251 ページ]

System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet) メソッド呼び出しまたは System.Data.Common.DbDataAdapter.FillSchema(System.Data.DataSet, System.Data.SchemaType) メソッド呼び出しの実行時に、System.Data.DataSet にコピーするための結果セットをデータベースから取得するために使用される ULCommand を指定します。

TableMappings プロパティ [252 ページ]

ソーステーブルと System.Data.DataTable 間のマスタマッピングを提供するコレクションを返します。

UpdateCommand プロパティ [253 ページ]

System.Data.DataSet で更新されたローに該当するデータベース内のローを更新するために、System.Data.Common.DbDataAdapter.Update メソッドが呼び出されたときにデータベースに対して実行される ULCommand オブジェクトを指定します。

RowUpdated イベント [254 ページ]

データソースに対してコマンドが実行された後の更新時に発生します。

RowUpdating イベント [254 ページ]

データソースに対してコマンドが実行される前の更新時に発生します。

関連情報

[ULDataReader クラス \[275 ページ\]](#)

[ULTable クラス \[526 ページ\]](#)

1.12.1 ULDataAdapter コンストラクタ

ULDataAdapter オブジェクトを初期化します。

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	ULDataAdapter() [242 ページ]	ULDataAdapter オブジェクトを初期化します。
public	ULDataAdapter(ULCommand) [243 ページ]	ULDataAdapter オブジェクトを、指定された SELECT 文で初期化します。
public	ULDataAdapter(string, ULConnection) [243 ページ]	ULDataAdapter オブジェクトを、指定された SELECT 文と接続で初期化します。

変更子とタイプ	オーバーロード名	説明
public	ULDataAdapter(string, string) [244 ページ]	ULDataAdapter オブジェクトを、指定された SELECT 文と接続文字列で初期化します。

このセクションの内容:

[ULDataAdapter\(\) コンストラクタ \[242 ページ\]](#)

ULDataAdapter オブジェクトを初期化します。

[ULDataAdapter\(ULCommand\) コンストラクタ \[243 ページ\]](#)

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

[ULDataAdapter\(string, ULConnection\) コンストラクタ \[243 ページ\]](#)

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

[ULDataAdapter\(string, string\) コンストラクタ \[244 ページ\]](#)

ULDataAdapter オブジェクトを、指定された SELECT 文と接続文字列で初期化します。

1.12.1.1 ULDataAdapter() コンストラクタ

ULDataAdapter オブジェクトを初期化します。

構文

Visual Basic

```
Public Sub ULDataAdapter ()
```

C#

```
public ULDataAdapter ()
```

関連情報

[ULDataAdapter\(ULCommand\) コンストラクタ \[243 ページ\]](#)

[ULDataAdapter\(string, ULConnection\) コンストラクタ \[243 ページ\]](#)

[ULDataAdapter\(string, string\) コンストラクタ \[244 ページ\]](#)

1.12.1.2 ULDataAdapter(ULCommand) コンストラクタ

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

構文

Visual Basic

```
Public Sub ULDataAdapter (ByVal selectCommand As ULCommand)
```

C#

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

パラメータ

selectCommand System.Data.DataSet に配置するためのレコードをデータソースから選択するために System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet) の実行時に使用される ULCommand オブジェクト。

関連情報

[ULDataAdapter\(\) コンストラクタ \[242 ページ\]](#)

[ULDataAdapter\(string, ULConnection\) コンストラクタ \[243 ページ\]](#)

[ULDataAdapter\(string, string\) コンストラクタ \[244 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

1.12.1.3 ULDataAdapter(string, ULConnection) コンストラクタ

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

構文

Visual Basic

```
Public Sub ULDataAdapter (  
    ByVal selectCommandText As String,  
    ByVal selectConnection As ULConnection  
)
```

C#

```
public ULDataAdapter (  
    string selectCommandText,
```

```
ULConnection selectConnection  
)
```

パラメータ

selectCommandText ULDataAdapter オブジェクトの ULDataAdapter.SelectCommand メソッドによって使用される SELECT 文。

selectConnection データベースへの接続を定義する ULConnection オブジェクト。

関連情報

[ULDataAdapter\(\) コンストラクタ \[242 ページ\]](#)

[ULDataAdapter\(ULCommand\) コンストラクタ \[243 ページ\]](#)

[ULDataAdapter\(string, string\) コンストラクタ \[244 ページ\]](#)

[SelectCommand プロパティ \[251 ページ\]](#)

[ULConnection クラス \[121 ページ\]](#)

1.12.1.4 ULDataAdapter(string, string) コンストラクタ

ULDataAdapter オブジェクトを、指定された SELECT 文と接続文字列で初期化します。

構文

Visual Basic

```
Public Sub ULDataAdapter (  
    ByVal selectCommandText As String,  
    ByVal selectConnectionString As String  
)
```

C#

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

パラメータ

selectCommandText ULDataAdapter.SelectCommand メソッドによって使用される SELECT 文。

`selectConnectionString` Ultra Light.NET データベースの接続文字列。

関連情報

[ULDataAdapter\(\) コンストラクタ \[242 ページ\]](#)

[ULDataAdapter\(ULCommand\) コンストラクタ \[243 ページ\]](#)

[ULDataAdapter\(string, ULConnection\) コンストラクタ \[243 ページ\]](#)

[SelectCommand プロパティ \[251 ページ\]](#)

1.12.2 Dispose(bool) メソッド

ULDataAdapter オブジェクトによって使用されるアンマネージリソースを解放するほか、必要に応じてマネージリソースを解放します。

構文

Visual Basic

```
Protected Overrides Sub Dispose (ByVal disposing As Boolean)
```

C#

```
protected override void Dispose (bool disposing)
```

パラメータ

`disposing` true の場合は、マネージリソースとアンマネージリソースの両方を破棄します。false の場合は、アンマネージリソースのみを破棄します。

1.12.3 Fill メソッド

オーバロードリスト

変更子とタイプ	オーバロード名	説明
protected override int	Fill(DataSet, int, int, string, IDbCommand, CommandBehavior) [246 ページ]	
protected override int	Fill(DataTable[], int, int, IDbCommand, CommandBehavior) [247 ページ]	

このセクションの内容:

[Fill\(DataSet, int, int, string, IDbCommand, CommandBehavior\) メソッド](#) [246 ページ]

[Fill\(DataTable\[\], int, int, IDbCommand, CommandBehavior\) メソッド](#) [247 ページ]

1.12.3.1 Fill(DataSet, int, int, string, IDbCommand, CommandBehavior) メソッド

構文

Visual Basic

```
Protected Overrides Function Fill (  
    ByVal dataSet As DataSet,  
    ByVal startRecord As Integer,  
    ByVal maxRecords As Integer,  
    ByVal srcTable As String,  
    ByVal cmd As IDbCommand,  
    ByVal behavior As CommandBehavior  
) As Integer
```

C#

```
protected override int Fill (  
    DataSet dataSet,  
    int startRecord,  
    int maxRecords,  
    string srcTable,  
    IDbCommand cmd,  
    CommandBehavior behavior  
)
```

1.12.3.2 Fill(DataTable[], int, int, IDbCommand, CommandBehavior) メソッド

構文

Visual Basic

```
Protected Overrides Function Fill (  
    ByVal dataTables As DataTable(),  
    ByVal startRecord As Integer,  
    ByVal maxRecords As Integer,  
    ByVal cmd As IDbCommand,  
    ByVal behavior As CommandBehavior  
    ) As Integer
```

C#

```
protected override int Fill (  
    DataTable[] dataTables,  
    int startRecord,  
    int maxRecords,  
    IDbCommand cmd,  
    CommandBehavior behavior  
    )
```

1.12.4 FillSchema メソッド

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
protected override DataTable[]	FillSchema(DataSet, SchemaType, IDbCommand, string, CommandBehavior) [248 ページ]	
protected override DataTable	FillSchema(DataTable, SchemaType, IDbCommand, CommandBehavior) [248 ページ]	

このセクションの内容:

[FillSchema\(DataSet, SchemaType, IDbCommand, string, CommandBehavior\) メソッド \[248 ページ\]](#)

[FillSchema\(DataTable, SchemaType, IDbCommand, CommandBehavior\) メソッド \[248 ページ\]](#)

1.12.4.1 FillSchema(DataSet, SchemaType, IDbCommand, string, CommandBehavior) メソッド

構文

Visual Basic

```
Protected Overrides Function FillSchema (  
    ByVal dataSet As DataSet,  
    ByVal schemaType As SchemaType,  
    ByVal cmd As IDbCommand,  
    ByVal srcTable As String,  
    ByVal behavior As CommandBehavior  
) As DataTable()
```

C#

```
protected override DataTable[] FillSchema (  
    DataSet dataSet,  
    SchemaType schemaType,  
    IDbCommand cmd,  
    string srcTable,  
    CommandBehavior behavior  
)
```

1.12.4.2 FillSchema(DataTable, SchemaType, IDbCommand, CommandBehavior) メソッド

構文

Visual Basic

```
Protected Overrides Function FillSchema (  
    ByVal dataTable As DataTable,  
    ByVal schemaType As SchemaType,  
    ByVal cmd As IDbCommand,  
    ByVal behavior As CommandBehavior  
) As DataTable
```

C#

```
protected override DataTable FillSchema (  
    DataTable dataTable,  
    SchemaType schemaType,  
    IDbCommand cmd,  
    CommandBehavior behavior  
)
```


1.12.5 GetFillParameters() メソッド

SELECT 文の実行時にユーザによって設定されるパラメータを返します。

構文

Visual Basic

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

C#

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

戻り値

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

備考

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

関連情報

[ULParameter クラス \[389 ページ\]](#)

1.12.6 Update(DataRow[], DataTableMapping) メソッド

構文

Visual Basic

```
Protected Overrides Function Update (  
    ByVal dataRows As DataRow(),  
    ByVal tm As DataTableMapping  
) As Integer
```

C#

```
protected override int Update (  
    DataRow[] dataRows,
```

```
DataTableMapping tm  
)
```

1.12.7 DeleteCommand プロパティ

System.Data.DataSet で削除されたローに該当するデータベース内のローを削除するために、DbDataAdapter.Update メソッドが呼び出されたときにデータベースに対して実行される ULCommand オブジェクトを指定します。

構文

Visual Basic

```
Public Shadows Property DeleteCommand As ULCommand
```

C#

```
public new ULCommand DeleteCommand {get;set;}
```

備考

System.Data.DataSet で削除されたローに該当するデータベース内のローを削除するために実行される ULCommand オブジェクト。

DeleteCommand プロパティが既存の ULCommand オブジェクトに割り当てられる場合、ULCommand オブジェクトのクローンは作成されません。DeleteCommand プロパティは、既存の ULCommand オブジェクトへの参照を保持します。

これは、System.Data.IDbDataAdapter.DeleteCommand プロパティと System.Data.Common.DbDataAdapter.DeleteCommand プロパティが厳密に型指定されたものです。

関連情報

[ULCommand クラス \[58 ページ\]](#)

1.12.8 InsertCommand プロパティ

System.Data.DataSet で挿入されたローに該当するローをデータベースに挿入するために、DbDataAdapter.Update メソッドが呼び出されたときにデータベースに対して実行される ULCommand オブジェクトを指定します。

構文

Visual Basic

```
Public Shadows Property InsertCommand As ULCommand
```

C#

```
public new ULCommand InsertCommand {get;set;}
```

備考

System.Data.DataSet で挿入されたローに該当するデータベース内のローを挿入するために実行される ULCommand オブジェクト。

InsertCommand プロパティが既存の ULCommand オブジェクトに割り当てられる場合、ULCommand オブジェクトのクローンは作成されません。InsertCommand プロパティは、既存の ULCommand オブジェクトへの参照を保持します。

これは、System.Data.IDbDataAdapter.InsertCommand プロパティと System.Data.Common.DbDataAdapter.InsertCommand プロパティが厳密に型指定されたものです。

関連情報

[ULCommand クラス \[58 ページ\]](#)

1.12.9 SelectCommand プロパティ

System.Data.Common.DbDataAdapter.Fill(System.Data.DataSet) メソッド呼び出しまたは System.Data.Common.DbDataAdapter.FillSchema(System.Data.DataSet, System.Data.SchemaType) メソッド呼び出しの実行時に、System.Data.DataSet にコピーするための結果セットをデータベースから取得するために使用される ULCommand を指定します。

構文

Visual Basic

```
Public Shadows Property SelectCommand As ULCommand
```

C#

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

備考

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

SelectCommand プロパティが既存の ULCommand オブジェクトに割り当てられる場合、ULCommand オブジェクトのクローンは作成されません。SelectCommand プロパティは、既存の ULCommand オブジェクトへの参照を保持します。

SelectCommand プロパティがローを返さない場合は、System.Data.DataSet にテーブルが追加されず、例外も発生しません。SELECT 文は、ULDataAdapter(ULCommand)、ULDataAdapter(String,ULConnection)、ULDataAdapter(String,String) のコンストラクタにも指定できます。

これは、System.Data.IDbDataAdapter.SelectCommand プロパティと System.Data.Common.DbDataAdapter.SelectCommand プロパティが厳密に型指定されたものです。

関連情報

[ULCommand クラス \[58 ページ\]](#)

[ULDataAdapter\(ULCommand\) コンストラクタ \[243 ページ\]](#)

[ULDataAdapter クラス \[238 ページ\]](#)

[ULDataAdapter クラス \[238 ページ\]](#)

1.12.10 TableMappings プロパティ

ソーステーブルと System.Data.DataTable 間のマスタマッピングを提供するコレクションを返します。

構文

Visual Basic

```
Public ReadOnly Shadows Property TableMappings As DataTableMappingCollection
```

C#

```
public new DataTableMappingCollection TableMappings {get;}
```

備考

ソーステーブルと System.Data.DataTables とのマスタマッピングを提供する

System.Data.Common.DataTableMapping オブジェクトのコレクション。デフォルト値は空のコレクションです。

変更を調整する場合、ULDataAdapter オブジェクトは System.Data.Common.DataTableMappingCollection コレクションを使用して、データソースによって使用されるカラム名を、System.Data.DataSet によって使用されるカラム名に関連付けます。

これは、System.Data.IDbDataAdapter.TableMappings プロパティが厳密に型指定されたものです。

1.12.11 UpdateCommand プロパティ

System.Data.DataSet で更新されたローに該当するデータベース内のローを更新するために、

System.Data.Common.DbDataAdapter.Update メソッドが呼び出されたときにデータベースに対して実行される ULCommand オブジェクトを指定します。

構文

Visual Basic

```
Public Shadows Property UpdateCommand As ULCommand
```

C#

```
public new ULCommand UpdateCommand {get;set;}
```

備考

System.Data.DataSet で更新されたローに該当するデータベース内のローを更新するために実行される ULCommand オブジェクト。

UpdateCommand が既存の ULCommand オブジェクトに割り当てられる場合、ULCommand オブジェクトのクローンは作成されません。UpdateCommand プロパティは、既存の ULCommand オブジェクトへの参照を保持します。

このコマンドを実行するとローが返される場合、ULCommand オブジェクトの ULCommand.UpdatedRowSource プロパティの設定方法によっては、これらのローが System.Data.DataSet にマージされることがあります。

これは、System.Data.IDbDataAdapter.UpdateCommand プロパティと

System.Data.Common.DbDataAdapter.DeleteCommand プロパティが厳密に型指定されたものです。

関連情報

[ULCommand クラス \[58 ページ\]](#)

1.12.12 RowUpdated イベント

データソースに対してコマンドが実行された後の更新時に発生します。

構文

Visual Basic

```
Public Event RowUpdated As ULRowUpdatedEventHandler
```

C#

```
public ULRowUpdatedEventHandler RowUpdated;
```

備考

更新が試みられると、イベントが発生します。

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

関連情報

[ULRowUpdatedEventHandler\(object, ULRowUpdatedEventArgs\) デリゲート \[581 ページ\]](#)

1.12.13 RowUpdating イベント

データソースに対してコマンドが実行される前の更新時に発生します。

構文

Visual Basic

```
Public Event RowUpdating As ULRowUpdatingEventHandler
```

C#

```
public ULRowUpdatingEventHandler RowUpdating;
```

備考

更新が試みられると、イベントが発生します。

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

関連情報

[ULRowUpdatedEventHandler\(object, ULRowUpdatedEventArgs\) デリゲート \[581 ページ\]](#)

1.13 ULDatabaseManager クラス

UL 拡張: データベースの作成、削除、検証を行う静的メソッドを提供します。

構文

Visual Basic

```
Public NotInheritable Class ULDatabaseManager
```

C#

```
public sealed class ULDatabaseManager
```

メンバー

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

メソッド

変更子とタイプ	メソッド	説明
public static void	CreateDatabase(string, string) [257 ページ]	新しい Ultra Light データベースを作成します。
public static void	DropDatabase(string) [258 ページ]	指定されたデータベースを削除します。
public static void	SetActiveSyncListener(string, ULActiveSyncListener) [260 ページ]	ActiveSync 用 Mobile Link プロバイダからの ActiveSync の呼び出しを処理するリスナオブジェクトを指定します。
public static void	SetServerSyncListener(string, string, ULServerSyncListener) [261 ページ]	指定されたサーバ同期メッセージの処理に使用するリスナオブジェクトを指定します。

変更子とタイプ	メソッド	説明
public static void	SignalSyncsComplete() [262 ページ]	ActiveSync 用の Mobile Link プロバイダに対して、アプリケーションが同期を完了したことを通知します。
public static void	ValidateDatabase(string, ULDBValid) [263 ページ]	データベースで低レベルのインデックス検証を実行します。

プロパティ

変更子とタイプ	プロパティ	説明
public ULRuntimeType	RuntimeType [264 ページ]	Ultra Light.NET ランタイムタイプを指定します。

備考

Ultra Light.NET の Ultra Light エンジンのランタイムを使用するには、ULDatabaseManager.RuntimeType プロパティを適切な値に設定してから、他の Ultra Light.NET API を使用します。

例

次の例では、Ultra Light エンジンのランタイムが選択され、接続が作成されます。

```
' Visual Basic
ULDatabaseManager.RuntimeType = ULRuntimeType.UL_ENGINE_CLIENT
Dim conn As ULConnection = new ULConnection
' The RuntimeType is now locked
```

対応する C# 言語のコードを次に示します。

```
// C#
ULDatabaseManager.RuntimeType = ULRuntimeType.UL_ENGINE_CLIENT;
ULConnection conn = new ULConnection();
// The RuntimeType is now locked
```

このセクションの内容:

[CreateDatabase\(string, string\) メソッド \[257 ページ\]](#)

新しい Ultra Light データベースを作成します。

[DropDatabase\(string\) メソッド \[258 ページ\]](#)

指定されたデータベースを削除します。

[SetActiveSyncListener\(string, ULActiveSyncListener\) メソッド \[260 ページ\]](#)

Microsoft ActiveSync 用 Mobile Link プロバイダからの Microsoft ActiveSync の呼び出しを処理するリスナオブジェクトを指定します。

[SetServerSyncListener\(string, string, ULServerSyncListener\) メソッド \[261 ページ\]](#)

指定されたサーバ同期メッセージの処理に使用するリスナオブジェクトを指定します。

[SignalSyncsComplete\(\) メソッド \[262 ページ\]](#)

ActiveSync 用の Mobile Link プロバイダに対して、アプリケーションが同期を完了したことを通知します。

[ValidateDatabase\(string, ULDBValid\) メソッド \[263 ページ\]](#)

データベースで低レベルのインデックス検証を実行します。

[RuntimeType プロパティ \[264 ページ\]](#)

Ultra Light.NET ランタイムタイプを指定します。

1.13.1 CreateDatabase(string, string) メソッド

新しい Ultra Light データベースを作成します。

構文

Visual Basic

```
Public Shared Sub CreateDatabase (  
    ByVal connString As String,  
    ByVal createParms As String  
)
```

C#

```
public static void CreateDatabase (  
    string connString,  
    string createParms  
)
```

パラメータ

connString キーワードと値の組み合わせがセミコロンで区切られたリスト形式の、データベースを指定するためのパラメータ。

createParms キーワードと値の組み合わせがセミコロンで区切られたリスト形式の、新しいデータベースを設定するためのパラメータ。

例外

ULException class SQL エラーが発生しました。

例

次のコードでは、Windows Mobile デバイス上にデータベース ¥UltraLite¥MyDatabase.udb を作成し、接続を開きます。

```
' Visual Basic  
Dim openParms As ULConnectionParms = New ULConnectionParms
```

```
openParms.DatabaseOnDevice = ".udb"
ULDatabaseManager.CreateDatabase( _
    openParms.ToString(), _
    "" _
)
Dim conn As ULConnection = _
    New ULConnection( openParms.ToString() )
conn.Open()
```

対応する C# 言語のコードを次に示します。

```
// C#
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnDevice = ".udb";
ULDatabaseManager.CreateDatabase(
    openParms.ToString(),
    ""
);
ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```

関連情報

[Open\(\) メソッド \[163 ページ\]](#)

[ULConnectionParms クラス \[187 ページ\]](#)

[ULCreateParms クラス \[214 ページ\]](#)

1.13.2 DropDatabase(string) メソッド

指定されたデータベースを削除します。

構文

Visual Basic

```
Public Shared Sub DropDatabase (ByVal connString As String)
```

C#

```
public static void DropDatabase (string connString)
```

パラメータ

connString キーワードと値の組み合わせがセミコロンで区切られたリスト形式の、データベースを指定するためのパラメータ。

例外

ULException class SQL エラーが発生しました。

備考

接続が開かれているデータベースを削除することはできません。

例

次のコードは、Windows Mobile デバイス上にデータベース ¥UltraLite¥MyDatabase.udb を作成し、このデータベースへの接続を開きます。

```
' Visual Basic
Dim connParms As ULConnectionParms = New ULConnectionParms
connParms.DatabaseOnDevice = ".udb"
ULConnection.DatabaseManager.DropDatabase( _
    connParms.ToString() _
)
```

対応する C# 言語のコードを次に示します。

```
// C#
ULConnectionParms connParms = new ULConnectionParms();
connParms.DatabaseOnDevice = ".udb";
ULConnection.DatabaseManager.DropDatabase(
    connParms.ToString()
);
ULConnection conn = new ULConnection( openParms.ToString() );
conn.Open();
```

関連情報

[Open\(\) メソッド \[163 ページ\]](#)

[ULConnectionParms クラス \[187 ページ\]](#)

1.13.3 SetActiveSyncListener(string, UActiveSyncListener) メソッド

Microsoft ActiveSync 用 Mobile Link プロバイダからの Microsoft ActiveSync の呼び出しを処理するリスナオブジェクトを指定します。

構文

Microsoft Visual Basic

```
Public Shared Sub SetActiveSyncListener (  
    ByVal appClassName As String,  
    ByVal listener As UActiveSyncListener  
)
```

C#

```
public static void SetActiveSyncListener (  
    string appClassName,  
    UActiveSyncListener listener  
)
```

パラメータ

appClassName アプリケーションのユニークなクラス名。これは、アプリケーションが Microsoft ActiveSync で使用されるように登録されているときに使用されるクラス名です。

listener UActiveSyncListener オブジェクト。前のリスナを削除するには、NULL (Microsoft Visual Basic の Nothing) を使用します。

例外

ULException class SQL エラーが発生しました。

備考

appClassName パラメータは、アプリケーションの識別に使用されるユニークな識別子です。アプリケーションは、一度に1つの **appClassName** 値のみを使用できます。リスナが、特定の **appClassName** 値に登録されているときは、別の **appClassName** 値で SetServerSyncListener または SetActiveSyncListener メソッドを呼び出すと失敗します。

Microsoft ActiveSync リスナを削除するには、**listener** パラメータとして null 参照 (Microsoft Visual Basic の Nothing) を使用して SetActiveSyncListener メソッドを呼び出します。

すべてのリスナを削除するには、すべてのパラメータに NULL 参照 (Microsoft Visual Basic の Nothing) を使用して SetServerSyncListener メソッドを呼び出します。

アプリケーションは、すべてのリスナを削除してから終了する必要があります。

関連情報

[SetServerSyncListener\(string, string, ULServerSyncListener\) メソッド \[261 ページ\]](#)

[ULActiveSyncListener インタフェース \[20 ページ\]](#)

[ActiveSyncInvoked\(bool\) メソッド \[20 ページ\]](#)

1.13.4 SetServerSyncListener(string, string, ULServerSyncListener) メソッド

指定されたサーバ同期メッセージの処理に使用するリスナオブジェクトを指定します。

構文

Visual Basic

```
Public Shared Sub SetServerSyncListener (  
    ByVal messageName As String,  
    ByVal appClassName As String,  
    ByVal listener As ULServerSyncListener  
)
```

C#

```
public static void SetServerSyncListener (  
    string messageName,  
    string appClassName,  
    ULServerSyncListener listener  
)
```

パラメータ

messageName メッセージの名前。

appClassName アプリケーションのユニークなクラス名。これは、アプリケーションの識別に使用されるユニークな識別子です。

listener ULServerSyncListener オブジェクト。前のリスナを削除するには、NULL (Visual Basic の Nothing) を使用します。

例外

ULException class SQL エラーが発生しました。

備考

`appClassName` パラメータは、アプリケーションの識別に使用されるユニークな識別子です。アプリケーションは、一度に1つの `appClassName` 値のみを使用できます。リスナが、特定の `appClassName` 値に登録されているときは、別の `appClassName` 値で `SetServerSyncListener` または `SetActiveSyncListener` メソッドを呼び出すと失敗します。

特定のメッセージのリスナを削除するには、`listener` パラメータとして NULL 参照 (Visual Basic の Nothing) を使用して `SetServerSyncListener` メソッドを呼び出します。

すべてのリスナを削除するには、すべてのパラメータに NULL 参照 (Visual Basic の Nothing) を使用して `SetServerSyncListener` メソッドを呼び出します。

アプリケーションは、すべてのリスナを削除してから終了する必要があります。

関連情報

[SetActiveSyncListener\(string, ULActiveSyncListener\) メソッド \[260 ページ\]](#)

[ServerSyncInvoked\(string\) メソッド \[483 ページ\]](#)

1.13.5 SignalSyncIsComplete() メソッド

ActiveSync 用の Mobile Link プロバイダに対して、アプリケーションが同期を完了したことを通知します。

構文

Visual Basic

```
Public Shared Sub SignalSyncIsComplete ()
```

C#

```
public static void SignalSyncIsComplete ()
```

関連情報

[ActiveSyncInvoked\(bool\) メソッド \[20 ページ\]](#)

1.13.6 ValidateDatabase(string, ULDBValid) メソッド

データベースで低レベルのインデックス検証を実行します。

構文

Visual Basic

```
Public Shared Sub ValidateDatabase (
    ByVal start_parms As String,
    ByVal how As ULDBValid
)
```

C#

```
public static void ValidateDatabase (
    string start_parms,
    ULDBValid how
)
```

パラメータ

start_parms キーワードと値の組み合わせがセミコロンで区切られたリスト形式の、データベースを指定するためのパラメータ。

how データベースを検証する方法を記述します。

例外

ULException class SQL エラーが発生しました。

例

次のコードでは、Windows Mobile のデータベース ¥UltraLite¥MyDatabase.udb のインデックスを検証します。

```
' Visual Basic
Dim openParms As ULConnectionParms = New ULConnectionParms
openParms.DatabaseOnDevice = "¥UltraLite¥MyDatabase.udb"
ULConnection.DatabaseManager.ValidateDatabase(
    openParms.ToString(), Sap.Data.UltraLite.ULVF_INDEX )
```

対応する C# 言語のコードを次に示します。

```
// C#
ULConnectionParms openParms = new ULConnectionParms();
openParms.DatabaseOnDevice = ".udb";
ULConnection.DatabaseManager.ValidateDatabase(
    openParms.ToString(), Sap.Data.UltraLite.ULVF_INDEX );
```

関連情報

[ValidateDatabase\(ULDBValid\) メソッド \[175 ページ\]](#)

[ValidateDatabase\(ULDBValid, string\) メソッド \[176 ページ\]](#)

[ULDBValid 列挙体 \[590 ページ\]](#)

[ULConnectionParms クラス \[187 ページ\]](#)

1.13.7 RuntimeType プロパティ

Ultra Light.NET ランタイムタイプを指定します。

構文

Visual Basic

```
Public Shared Property RuntimeType As ULRuntimeType
```

C#

```
public ULRuntimeType RuntimeType {get;set;}
```

備考

ランタイムタイプを選択してから、その他の Ultra Light.NET API を使用してください。

アンマネージ Ultra Light.NET ランタイムのタイプを示す ULRuntimeType 値。

例

次の例では、Ultra Light エンジンのランタイムが選択され、接続が作成されます。

```
' Visual Basic
ULDatabaseManager.RuntimeType = ULRuntimeType.UL_ENGINE_CLIENT
Dim conn As ULConnection = new ULConnection
' The RuntimeType is now locked
```

対応する C# 言語のコードを次に示します。

```
// C#
ULDatabaseManager.RuntimeType = ULRuntimeType.UL_ENGINE_CLIENT;
ULConnection conn = new ULConnection();
// The RuntimeType is now locked
```


関連情報

[ULRuntimeType 列挙体 \[590 ページ\]](#)

1.14 ULDatabaseSchema クラス

UL 拡張: Ultra Light.NET データベースのスキーマを表します。

構文

Visual Basic

```
Public NotInheritable Class ULDatabaseSchema
```

C#

```
public sealed class ULDatabaseSchema
```

メンバー

ULDatabaseSchema 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 string	GetDatabaseProperty(string) [267 ページ]	指定されたデータベースのプロパティの値を返します。
public string	GetPublicationName(int) [269 ページ]	指定されたパブリケーション ID で識別されたパブリケーションの名前を返します。
public string	GetTableName(int) [270 ページ]	指定されたテーブル ID で識別されたテーブルの名前を返します。
public void	SetDatabaseOption(string, string) [271 ページ]	指定されたデータベースオプションの値を設定します。

プロパティ

変更子とタイプ	プロパティ	説明
public unsafe bool	IsCaseSensitive [272 ページ]	データベースで大文字と小文字が区別されるかどうかをチェックします。
public bool	IsOpen [273 ページ]	データベーススキーマが開いているかどうかを判断します。

変更子とタイプ	プロパティ	説明
public unsafe int	PublicationCount [274 ページ]	データベース内のパブリケーションの数を数えます。
public unsafe int	TableCount [274 ページ]	データベース内のテーブルの数を数えます。

備考

このクラスにはコンストラクタがありません。ULDatabaseSchema オブジェクトは、ULConnection.Schema オブジェクトとして接続にアタッチされ、接続が開いている間のみ有効です。

このセクションの内容:

[GetDatabaseProperty\(string\) メソッド \[267 ページ\]](#)

指定されたデータベースのプロパティの値を返します。

[GetPublicationName\(int\) メソッド \[269 ページ\]](#)

指定されたパブリケーション ID で識別されたパブリケーションの名前を返します。

[GetTableName\(int\) メソッド \[270 ページ\]](#)

指定されたテーブル ID で識別されたテーブルの名前を返します。

[SetDatabaseOption\(string, string\) メソッド \[271 ページ\]](#)

指定されたデータベースオプションの値を設定します。

[IsCaseSensitive プロパティ \[272 ページ\]](#)

データベースで大文字と小文字が区別されるかどうかをチェックします。

[IsOpen プロパティ \[273 ページ\]](#)

データベーススキーマが開いているかどうかを判断します。

[PublicationCount プロパティ \[274 ページ\]](#)

データベース内のパブリケーションの数を数えます。

[TableCount プロパティ \[274 ページ\]](#)

データベース内のテーブルの数を数えます。

関連情報

[Schema プロパティ \[182 ページ\]](#)

1.14.1 GetDatabaseProperty(string) メソッド

指定されたデータベースのプロパティの値を返します。

構文

Visual Basic

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

C#

```
public string GetDatabaseProperty (string name)
```

パラメータ

name 値を取得するデータベースプロパティの名前。プロパティ名では大文字と小文字が区別されません。

戻り値

文字列として返されるプロパティの値。

例外

ULException class SQL エラーが発生しました。

備考

識別されるプロパティは次のとおりです。

プロパティ	説明
CaseSensitive	大文字と小文字の区別のステータス。データベースで大文字と小文字が区別される場合は、ON を返します。それ以外の場合は、OFF を返します。データベースで大文字と小文字が区別されるかどうかは、テーブルのインデックスと結果セットのソート方法に影響します。大文字と小文字の区別は、接続の ULConnectionParms.UserID と ULConnectionParms.Password の値の検証方法には影響しません。ユーザ ID は常に大文字と小文字が区別されません。パスワードは常に大文字と小文字が区別されます。

プロパティ	説明
CharSet	データベースの文字セット。
ChecksumLevel	データベースで有効にするデータベースページのチェックサムのレベル。
Collation	データベースの照合順の名前。
ConnCount	データベースとの接続の数。
date_format	データベースによる文字列変換に使用される日付形式。この形式は、System.DateTime によって使用される形式と同じであるとは限りません。
date_order	データベースによる文字列変換に使用される日付順。
Encryption	データベースに適用されている暗号化のタイプ。None、Simple、AES、または AES_FIPS を返します。
File	データベースのファイル名。
global_database_id	グローバルオートインクリメントカラムに使用される global_database_id オプションの値。
isolation_level	あるトランザクションの操作が同時に実行される他のトランザクションの操作でも認識される程度を制御するのに使用される isolation_level オプションの値。この値は、接続単位で設定します。
MaxHashSize	インデックスのハッシュに使用する、デフォルトの最大バイト数。このプロパティは、インデックス単位で指定できます。
ml_remote_id	同期中にデータベースを識別するために使用する、ml_remote_id オプションの値。
Name	データベースの名前 (DBN)。
nearest_century	データベースによる文字列変換に使用される最も近い世紀。
PageSize	データベースのページサイズ (バイト)。
precision	データベースによる文字列変換に使用される浮動小数点の精度。
scale	データベースによる文字列変換中に、計算結果が最大 PRECISION 値にトランケートされる場合の小数点以下の最大桁数。
time_format	データベースによる文字列変換に使用される時間形式。この形式は、System.TimeSpan によって使用される形式と同じであるとは限りません。
timestamp_format	データベースによる文字列変換に使用されるタイムスタンプ形式。この形式は、System.DateTime によって使用される形式と同じであるとは限りません。
timestamp_increment	2つのユニークなタイムスタンプ間のマイクロ秒 (1,000,000 分の 1 秒) 単位の最小差。

関連情報

[SetDatabaseOption\(string, string\) メソッド \[271 ページ\]](#)

[UserID プロパティ \[196 ページ\]](#)

1.14.2 GetPublicationName(int) メソッド

指定されたパブリケーション ID で識別されたパブリケーションの名前を返します。

構文

Visual Basic

```
Public Function GetPublicationName (ByVal pubID As Integer) As String
```

C#

```
public string GetPublicationName (int pubID)
```

パラメータ

pubID パブリケーションの ID。値は、[1,PublicationCount] の範囲内である必要があります。

戻り値

文字列として返されるパブリケーション名。

例外

ULException class SQL エラーが発生しました。

備考

i 注記

パブリケーション ID とカウントは、スキーマのアップグレード中に変更されることがあります。パブリケーションを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後に再表示します。

関連情報

[PublicationCount プロパティ \[274 ページ\]](#)

[PublicationCount プロパティ \[274 ページ\]](#)

1.14.3 GetTableName(int) メソッド

指定されたテーブル ID で識別されたテーブルの名前を返します。

構文

Visual Basic

```
Public Function GetTableName (ByVal tableID As Integer) As String
```

C#

```
public string GetTableName (int tableID)
```

パラメータ

tableID テーブルの ID。値は、[1,TableCount] の範囲内である必要があります。

戻り値

文字列としてのテーブル名。

例外

ULException class SQL エラーが発生しました。

備考

テーブル ID は、スキーマのアップグレード中に変更されることがあります。テーブルを正しく識別するには、名前でアクセスするか、キャッシュされている ID をスキーマのアップグレード後に再表示します。

関連情報

[TableCount プロパティ \[274 ページ\]](#)

1.14.4 SetDatabaseOption(string, string) メソッド

指定されたデータベースオプションの値を設定します。

構文

Visual Basic

```
Public Sub SetDatabaseOption (  
    ByVal name As String,  
    ByVal value As String  
)
```

C#

```
public void SetDatabaseOption (  
    string name,  
    string value  
)
```

パラメータ

name データベースオプションの名前。オプション名では大文字と小文字が区別されません。

value オプションの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

データベースオプションを指定すると、コミットが実行されます。

トランザクションがアクティブなときにこのメソッドを使用することは、予期しない結果を引き起こす場合があるため、推奨できません。呼び出すと、接続の独立性レベルは変更されますが、`ULTransaction.IsolationLevel` 値は更新されません。

識別されるオプションは次のとおりです。

オプション	説明
global_database_id	グローバルオートインクリメントカラムに使用する値。値は、[0, System.UInt32.MaxValue] の範囲内であることが必要です。デフォルトは ULConnection.INVALID_DATABASE_ID 値 (現在のデータベースにデータベース ID が設定されていないことを示す) です。
isolation_level	あるトランザクションの操作が、同時に実行される他のトランザクションの操作で認識される程度を制御するのに使用される値。値は "read_uncommitted" または "read_committed" のいずれかです。デフォルトは "read_committed" です。接続の isolation_level を "read_uncommitted" に設定することは、BeginTransaction(System.Data.IsolationLevel.ReadUncommitted) および Commit() 呼び出してその接続におけるすべての操作をラップすることと同じです。同様に、"read_committed" は System.Data.IsolationLevel.ReadCommitted と同じです。現在のトランザクションの独立性レベルを設定する場合は、SetDatabaseOption() ではなく、BeginTransaction(IsolationLevel) を使用してください。Ultra Light での独立性レベルの定義は、ADO.NET のマニュアルの IsolationLevel の説明とは若干異なります。この値は、接続単位で設定します。
ml_remote_id	同期中にデータベースを識別するために使用する値。値として NULL 参照 (Visual Basic の Nothing) を使用して、データベースから ml_remote_id オプションを削除します。

関連情報

[GetDatabaseProperty\(string\) メソッド \[267 ページ\]](#)

1.14.5 IsCaseSensitive プロパティ

データベースで大文字と小文字が区別されるかどうかをチェックします。

構文

Visual Basic

```
Public ReadOnly Property IsCaseSensitive As Boolean
```

C#

```
public unsafe bool IsCaseSensitive {get;}
```


備考

データベースで大文字と小文字が区別される場合は true、区別されない場合は false。

データベースで大文字と小文字が区別されるかどうかは、テーブルのインデックスと結果セットのソート方法に影響します。また、大文字と小文字の区別は、ULConnectionParms.UserID と ULConnectionParms.Password の値の検証方法にも影響します。

関連情報

[GetDatabaseProperty\(string\) メソッド \[267 ページ\]](#)

[UserID プロパティ \[196 ページ\]](#)

[Password プロパティ \[195 ページ\]](#)

1.14.6 IsOpen プロパティ

データベーススキーマが開いているかどうかを判断します。

構文

Visual Basic

```
Public ReadOnly Property IsOpen As Boolean
```

C#

```
public bool IsOpen {get;}
```

備考

データベースのスキーマが現在開いている場合は true、現在閉じている場合は false。

ULDatabaseSchema オブジェクトが開いているのは、それがアタッチされている接続が開いている場合だけです。

1.14.7 PublicationCount プロパティ

データベース内のパブリケーションの数を数えます。

構文

Visual Basic

```
Public ReadOnly Property PublicationCount As Integer
```

C#

```
public unsafe int PublicationCount {get;}
```

備考

データベース内のパブリケーションの数です。

パブリケーション ID の範囲は、1 ~ PublicationCount 値 (両端の値を含む) です。

i 注記

パブリケーション ID とカウントは、スキーマのアップグレード中に変更されることがあります。パブリケーションを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後に再表示します。

関連情報

[GetPublicationName\(int\) メソッド \[269 ページ\]](#)

1.14.8 TableCount プロパティ

データベース内のテーブルの数を数えます。

構文

Visual Basic

```
Public ReadOnly Property TableCount As Integer
```

C#

```
public unsafe int TableCount {get;}
```

備考

データベース内のテーブルの数です。

テーブル ID の範囲は、1 ~ TableCount 値 (両端の値を含む) です。

i 注記

テーブルの ID とカウントは、スキーマのアップグレード中に変更されることがあります。テーブルを正しく識別するには、名前前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後に再表示します。

1.15 ULDataReader クラス

Ultra Light データベースの読み込み専用の双方向カーソルを表します。

構文

Visual Basic

```
Public Class ULDataReader Inherits System.Data.Common.DbDataReader Implements System.ComponentModel.IListSource
```

C#

```
public class ULDataReader : System.Data.Common.DbDataReader, System.ComponentModel.IListSource
```

メンバー

ULDataReader 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 override void	Close() [282 ページ]	カーソルを閉じます。
protected override void	Dispose(bool) [282 ページ]	
public override unsafe bool	GetBoolean(int) [283 ページ]	指定されたカラムの値を System.Boolean として返します。
public override unsafe byte	GetByte(int) [284 ページ]	指定されたカラムの値を符号なし 8 ビット値 (System.Byte) として返します。
public unsafe byte[]	GetBytes [285 ページ]	UL 拡張: 指定されたカラムの値を System.Bytes の値の配列として返します。

変更子とタイプ	メソッド	説明
public override char	GetChar(int) [288 ページ]	このメソッドは Ultra Light.NET ではサポートされていません。
public override unsafe long	GetChars(int, long, char[], int, int) [289 ページ]	指定されたオフセットで始まる、指定された ULDbType.LongVarchar カラムの値のサブセットを、コピー先の System.Char 配列の指定されたオフセットにコピーします。
public override string	GetDataTypeName(int) [291 ページ]	指定されたカラムのプロバイダのデータ型の名前を返します。
public override unsafe DateTime	GetDateTime(int) [292 ページ]	指定されたカラムの値を、ミリ秒の精度の System.DateTime 型として返します。
protected override DbDataReader	GetDbDataReader(int) [293 ページ]	
public override decimal	GetDecimal(int) [293 ページ]	指定されたカラムの値を System.Decimal 型として返します。
public override unsafe double	GetDouble(int) [294 ページ]	指定されたカラムの値を System.Double 型として返します。
public override IEnumerable	GetEnumerator() [295 ページ]	ULDataReader オブジェクトの反復処理を実行する System.Collections.IEnumerator 値を返します。
public override Type	GetFieldType(int) [295 ページ]	指定されたカラムに最適な System.Type 値を返します。
public override unsafe float	GetFloat(int) [296 ページ]	指定されたカラムの値を System.Single 型として返します。
public override unsafe Guid	GetGuid(int) [297 ページ]	指定されたカラムの値を UUID (System.Guid) 型として返します。
public override unsafe short	GetInt16(int) [298 ページ]	指定されたカラムの値を System.Int16 型として返します。
public override unsafe int	GetInt32(int) [299 ページ]	指定されたカラムの値を System.Int32 型として返します。
public override unsafe long	GetInt64(int) [300 ページ]	指定されたカラムの値を System.Int64 型として返します。
public override string	GetName(int) [301 ページ]	指定されたカラムの名前を返します。
public override unsafe int	GetOrdinal(string) [302 ページ]	指定されたカラムのカラム ID を返します。
public unsafe int	GetRowCount(int) [303 ページ]	UL 拡張: カーソル内のローの数を、スレッシュホールド以内で返します。
public override DataTable	GetSchemaTable() [304 ページ]	ULDataReader オブジェクトのカラムのメタデータが記述された System.Data.DataTable 値を返します。
public override unsafe String	GetString(int) [306 ページ]	指定されたカラムの値を System.String 型として返します。
public unsafe TimeSpan	GetTimeSpan(int) [307 ページ]	指定されたカラムの値を、ミリ秒の精度の System.TimeSpan 型として返します。

変数とタイプ	メソッド	説明
public unsafe ushort	GetUInt16(int) [308 ページ]	指定されたカラムの値を System.UInt16 型として返します。
public unsafe uint	GetUInt32(int) [309 ページ]	指定されたカラムの値を System.UInt32 型として返します。
public unsafe ulong	GetUInt64(int) [310 ページ]	指定されたカラムの値を System.UInt64 型として返します。
public override object	GetValue(int) [311 ページ]	指定されたカラムの値をネイティブフォーマットで返します。
public override int	GetValues(object[]) [312 ページ]	現在のローのすべてのカラム値を返します。
public override unsafe bool	IsDBNull(int) [313 ページ]	指定されたカラムの値が NULL かどうかをチェックします。
public void	MoveAfterLast() [314 ページ]	UL 拡張: カーソルの最後のローの後に、カーソルを配置します。
public void	MoveBeforeFirst() [314 ページ]	UL 拡張: カーソルの最初のローの前に、カーソルを配置します。
public unsafe bool	MoveFirst() [315 ページ]	UL 拡張: カーソルの最初のローに、カーソルを配置します。
public unsafe bool	MoveLast() [315 ページ]	UL 拡張: カーソルの最後のローに、カーソルを配置します。
public unsafe bool	MoveNext() [316 ページ]	UL 拡張: カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。
public unsafe bool	MovePrevious() [317 ページ]	UL 拡張: カーソルを前のローに配置するか、最初のローの前に配置します。
public unsafe bool	MoveRelative(int) [317 ページ]	UL 拡張: 現在のローを基準としてカーソルを配置します。
public override bool	NextResult() [318 ページ]	バッチ SQL 文の結果を読み込むときに ULDataReader オブジェクトを次の結果に進めます。
public override bool	Read() [319 ページ]	カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。
protected void	Validate [320 ページ]	個々のトピックを参照してください。

プロパティ

変数とタイプ	プロパティ	説明
public override int	Depth [321 ページ]	現在のローのネストの深さを返します。
public override int	FieldCount [321 ページ]	このカーソル内のカラム数を返します。
public override unsafe bool	HasRows [322 ページ]	ULDataReader オブジェクトに 1 つまたは複数のローがあるかどうかをチェックします。

変更子とタイプ	プロパティ	説明
public unsafe bool	IsBOF [322 ページ]	UL 拡張: 現在のローの位置が最初のローの前かどうかをチェックします。
public override bool	IsClosed [323 ページ]	カーソルが現在開いているかどうかを確認します。
public unsafe bool	IsEOF [323 ページ]	UL 拡張: 現在のローの位置が最後のローの後かどうかをチェックします。
public override int	RecordsAffected [324 ページ]	SQL 文の実行によって変更、挿入、または削除されたローの数を返します。
public int	RowCount [324 ページ]	UL 拡張: カーソル内のローの数を返します。
public ULCursorSchema	Schema [325 ページ]	UL 拡張: このカーソルのスキーマを保持します。
public override object	this [326 ページ]	指定されたカラムの値をネイティブフォーマットで返します。

備考

カーソルとは、テーブルまたはクエリからの結果セットの一連のローです。

ULDataReader クラスにはコンストラクタがありません。ULDataReader オブジェクトを取得するには、次のように ULCommand オブジェクトを実行します。

```
' Visual Basic
Dim cmd As ULCommand = new ULCommand(
    "SELECT emp_id FROM employee", conn_
)
Dim reader As ULDataReader = cmd.ExecuteReader()
```

対応する C# 言語のコードを次に示します。

```
// C#
ULCommand cmd = new ULCommand(
    "SELECT emp_id FROM employee", conn
);
ULDataReader reader = cmd.ExecuteReader();
```

UL 拡張: ADO.NET 標準では、結果セットで必要なのは前方への移動だけですが、ULDataReader オブジェクトは双方向です。ULDataReader の Move メソッドによって、結果セットを移動するときに最大限の柔軟性が得られます。

ULDataReader オブジェクトは、読み込み専用の結果セットです。より柔軟なオブジェクトで結果を操作する必要がある場合は、ULCommand.ExecuteResultSet メソッド、ULCommand.ExecuteTable メソッド、または ULDataAdapter クラスを使用します。ULDataReader クラスは必要に応じてローを取得しますが、ULDataAdapter クラスの場合、結果セットのすべてのローを取得しないと、オブジェクトに対してアクションを実行できません。結果セットのサイズが大きい場合、この違いのために ULDataReader クラスの方が応答時間が速くなります。

UL 拡張: ULDataReader オブジェクトのカラムはすべて、GetString メソッドを使用して取得できます。

このセクションの内容:

[Close\(\) メソッド \[282 ページ\]](#)

カーソルを閉じます。

[Dispose\(bool\) メソッド \[282 ページ\]](#)

[GetBoolean\(int\) メソッド \[283 ページ\]](#)

指定されたカラムの値を System.Boolean として返します。

[GetByte\(int\) メソッド \[284 ページ\]](#)

指定されたカラムの値を符号なし 8 ビット値 (System.Byte) として返します。

[GetBytes メソッド \[285 ページ\]](#)

UL 拡張: 指定されたカラムの値を System.Bytes の値の配列として返します。

[GetChar\(int\) メソッド \[288 ページ\]](#)

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

[GetChars\(int, long, char\[\], int, int\) メソッド \[289 ページ\]](#)

指定されたオフセットで始まる、指定された ULDbType.LongVarchar カラムの値のサブセットを、コピー先の System.Char 配列の指定されたオフセットにコピーします。

[GetDataTypeName\(int\) メソッド \[291 ページ\]](#)

指定されたカラムのプロバイダのデータ型の名前を返します。

[GetDateTime\(int\) メソッド \[292 ページ\]](#)

指定されたカラムの値を、ミリ秒の精度の System.DateTime 型として返します。

[GetDbDataReader\(int\) メソッド \[293 ページ\]](#)

[GetDecimal\(int\) メソッド \[293 ページ\]](#)

指定されたカラムの値を System.Decimal 型として返します。

[GetDouble\(int\) メソッド \[294 ページ\]](#)

指定されたカラムの値を System.Double 型として返します。

[GetEnumerator\(\) メソッド \[295 ページ\]](#)

ULDataReader オブジェクトの反復処理を実行する System.Collections.IEnumerator 値を返します。

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

指定されたカラムに最適な System.Type 値を返します。

[GetFloat\(int\) メソッド \[296 ページ\]](#)

指定されたカラムの値を System.Single 型として返します。

[GetGuid\(int\) メソッド \[297 ページ\]](#)

指定されたカラムの値を UUID (System.Guid) 型として返します。

[GetInt16\(int\) メソッド \[298 ページ\]](#)

指定されたカラムの値を System.Int16 型として返します。

[GetInt32\(int\) メソッド \[299 ページ\]](#)

指定されたカラムの値を System.Int32 型として返します。

[GetInt64\(int\) メソッド \[300 ページ\]](#)

指定されたカラムの値を System.Int64 型として返します。

[GetName\(int\) メソッド \[301 ページ\]](#)

指定されたカラムの名前を返します。

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

指定されたカラムのカラム ID を返します。

[GetRowCount\(int\) メソッド \[303 ページ\]](#)

UL 拡張: カーソル内のローの数を、スレッショルド以内で返します。

[GetSchemaTable\(\) メソッド \[304 ページ\]](#)

ULDataReader オブジェクトのカラムのメタデータが記述された System.Data.DataTable 値を返します。

[GetString\(int\) メソッド \[306 ページ\]](#)

指定されたカラムの値を System.String 型として返します。

[GetTimeSpan\(int\) メソッド \[307 ページ\]](#)

指定されたカラムの値を、ミリ秒の精度の System.TimeSpan 型として返します。

[GetUInt16\(int\) メソッド \[308 ページ\]](#)

指定されたカラムの値を System.UInt16 型として返します。

[GetUInt32\(int\) メソッド \[309 ページ\]](#)

指定されたカラムの値を System.UInt32 型として返します。

[GetUInt64\(int\) メソッド \[310 ページ\]](#)

指定されたカラムの値を System.UInt64 型として返します。

[GetValue\(int\) メソッド \[311 ページ\]](#)

指定されたカラムの値をネイティブフォーマットで返します。

[GetValues\(object\[\]\) メソッド \[312 ページ\]](#)

現在のローのすべてのカラム値を返します。

[IsDBNull\(int\) メソッド \[313 ページ\]](#)

指定されたカラムの値が NULL かどうかをチェックします。

[MoveAfterLast\(\) メソッド \[314 ページ\]](#)

UL 拡張: カーソルの最後のローの後に、カーソルを配置します。

[MoveBeforeFirst\(\) メソッド \[314 ページ\]](#)

UL 拡張: カーソルの最初のローの前に、カーソルを配置します。

[MoveFirst\(\) メソッド \[315 ページ\]](#)

UL 拡張: カーソルの最初のローに、カーソルを配置します。

[MoveLast\(\) メソッド \[315 ページ\]](#)

UL 拡張: カーソルの最後のローに、カーソルを配置します。

[MoveNext\(\) メソッド \[316 ページ\]](#)

UL 拡張: カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。

[MovePrevious\(\) メソッド \[317 ページ\]](#)

UL 拡張: カーソルを前のローに配置するか、最初のローの前に配置します。

[MoveRelative\(int\) メソッド \[317 ページ\]](#)

UL 拡張: 現在のローを基準としてカーソルを配置します。

[NextResult\(\) メソッド \[318 ページ\]](#)

バッチ SQL 文の結果を読み込むときに ULDataReader オブジェクトを次の結果に進めます。

[Read\(\) メソッド \[319 ページ\]](#)

カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置しません。

[Validate メソッド \[320 ページ\]](#)

[Depth プロパティ \[321 ページ\]](#)

現在のローのネストの深さを返します。

[FieldCount プロパティ \[321 ページ\]](#)

このカーソル内のカラム数を返します。

[HasRows プロパティ \[322 ページ\]](#)

ULDataReader オブジェクトに 1 つまたは複数のローがあるかどうかをチェックします。

[IsBOF プロパティ \[322 ページ\]](#)

UL 拡張: 現在のローの位置が最初のローの前かどうかをチェックします。

[IsClosed プロパティ \[323 ページ\]](#)

カーソルが現在開いているかどうかを確認します。

[IsEOF プロパティ \[323 ページ\]](#)

UL 拡張: 現在のローの位置が最後のローの後かどうかをチェックします。

[RecordsAffected プロパティ \[324 ページ\]](#)

SQL 文の実行によって変更、挿入、または削除されたローの数を返します。

[RowCount プロパティ \[324 ページ\]](#)

UL 拡張: カーソル内のローの数を返します。

[Schema プロパティ \[325 ページ\]](#)

UL 拡張: このカーソルのスキーマを保持します。

[this プロパティ \[326 ページ\]](#)

指定されたカラムの値をネイティブフォーマットで返します。

関連情報

[ULCommand クラス \[58 ページ\]](#)

[ExecuteResultSet\(\) メソッド \[89 ページ\]](#)

[ExecuteTable\(\) メソッド \[94 ページ\]](#)

[ULDataAdapter クラス \[238 ページ\]](#)

[GetString\(int\) メソッド \[306 ページ\]](#)

1.15.1 Close() メソッド

カーソルを閉じます。

構文

Visual Basic

```
Public Overrides Sub Close ()
```

C#

```
public override void Close ()
```

例外

ULException class SQL エラーが発生しました。

備考

すでに閉じられているカーソルを閉じるのはエラーではありません。

1.15.2 Dispose(bool) メソッド

構文

Visual Basic

```
Protected Overrides Sub Dispose (ByVal disposing As Boolean)
```

C#

```
protected override void Dispose (bool disposing)
```

1.15.3 GetBoolean(int) メソッド

指定されたカラムの値を System.Boolean として返します。

構文

Visual Basic

```
Public Overrides Function GetBoolean (ByVal colID As Integer) As Boolean
```

C#

```
public override unsafe bool GetBoolean (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Boolean として返されるカラム値。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.4 GetByte(int) メソッド

指定されたカラムの値を符号なし 8 ビット値 (System.Byte) として返します。

構文

Visual Basic

```
Public Overrides Function GetByte (ByVal colID As Integer) As Byte
```

C#

```
public override unsafe byte GetByte (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Byte として返されるカラム値。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.5 GetBytes メソッド

UL 拡張: 指定されたカラムの値を System.Bytes の値の配列として返します。

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 unsafe byte[]	GetBytes(int) [285 ページ]	UL 拡張: 指定されたカラムの値を System.Bytes の値の配列として返します。
public override unsafe long	GetBytes(int, long, byte[], int, int) [286 ページ]	指定されたオフセットで始まる、指定された ULDbType.LongBinary カラムの値のサブセットを、コピー先の System.Byte 配列の指定されたオフセットにコピーします。

このセクションの内容:

[GetBytes\(int\) メソッド \[285 ページ\]](#)

UL 拡張: 指定されたカラムの値を System.Bytes の値の配列として返します。

[GetBytes\(int, long, byte\[\], int, int\) メソッド \[286 ページ\]](#)

指定されたオフセットで始まる、指定された ULDbType.LongBinary カラムの値のサブセットを、コピー先の System.Byte 配列の指定されたオフセットにコピーします。

1.15.5.1 GetBytes(int) メソッド

UL 拡張: 指定されたカラムの値を System.Bytes の値の配列として返します。

構文

Visual Basic

```
Public Function GetBytes (ByVal colID As Integer) As Byte()
```

C#

```
public unsafe byte[] GetBytes (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0, ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Bytes 型の配列としてのカラム値。

例外

ULException class SQL エラーが発生しました。

備考

ULDbType.Binary 型、ULDbType.LongBinary 型、ULDbType.Uniqueidentifier 型のカラムの場合にのみ有効です。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[GetBytes\(int, long, byte\[\], int, int\) メソッド \[286 ページ\]](#)

1.15.5.2 GetBytes(int, long, byte[], int, int) メソッド

指定されたオフセットで始まる、指定された ULDbType.LongBinary カラムの値のサブセットを、コピー先の System.Byte 配列の指定されたオフセットにコピーします。

構文

Visual Basic

```
Public Overrides Function GetBytes (  
    ByVal colID As Integer,  
    ByVal srcOffset As Long,  
    ByVal dst As Byte(),  
    ByVal dstOffset As Integer,  
    ByVal count As Integer  
    ) As Long
```

C#

```
public override unsafe long GetBytes (  
    int colID,  
    long srcOffset,  
    byte[] dst,  
    int dstOffset,  
    int count
```

)

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

srcOffset カラム値の開始位置。最初の値は 0 です。

dst コピー先の配列。

dstOffset コピー先の配列の開始位置。

count コピーされるバイト数。

戻り値

実際にコピーされたバイト数。

例外

ULException class SQL エラーが発生しました。

備考

null 参照 (Visual Basic の Nothing) である *dst* バッファを渡すと、GetBytes メソッドはフィールドの長さをバイト数で返します。

値の *srcOffset* から *srcOffset+count-1* までの位置のバイトが、コピー先の配列の *dstOffset* から *dstOffset+count-1* までの位置に、それぞれコピーされます。*count* のバイト数がコピーされる前に、値の末尾が検出された場合は、コピー先の配列の残りは変更されないままになります。

次のいずれかに該当する場合、コード ULSQLCode.SQLE_INVALID_PARAMETER とともに ULException オブジェクトがスローされ、追加先は修正されません。

- *srcOffset* が負です。
- *dstOffset* が負です。
- *count* が負です。
- *dstOffset+count* は *dst* の長さより長い。

その他のエラーの場合は、それに応じたエラーコードとともに ULException オブジェクトがスローされます。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[GetBytes\(int\) メソッド \[285 ページ\]](#)

[ULException クラス \[328 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.6 GetChar(int) メソッド

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

構文

Visual Basic

```
Public Overrides Function GetChar (ByVal colID As Integer) As Char
```

C#

```
public override char GetChar (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

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

例外

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

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[GetString\(int\) メソッド \[306 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.7 GetChars(int, long, char[], int, int) メソッド

指定されたオフセットで始まる、指定された `ULDbType.LongVarchar` カラムの値のサブセットを、コピー先の `System.Char` 配列の指定されたオフセットにコピーします。

構文

Visual Basic

```
Public Overrides Function GetChars (  
    ByVal colID As Integer,  
    ByVal srcOffset As Long,  
    ByVal dst As Char(),  
    ByVal dstOffset As Integer,  
    ByVal count As Integer  
) As Long
```

C#

```
public override unsafe long GetChars (  
    int colID,  
    long srcOffset,  
    char[] dst,  
    int dstOffset,  
    int count  
)
```

パラメータ

colID カラムの ID 番号。値は、`[0, ULDataReader.FieldCount-1]` の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

srcOffset カラム値の開始位置。最初の値は 0 です。

dst コピー先の配列。

dstOffset コピー先の配列の開始位置。

count コピーされる文字数。

戻り値

実際にコピーされた文字数。

例外

ULException class SQL エラーが発生しました。

備考

null 参照 (Visual Basic の Nothing) である *dst* バッファを渡すと、GetChars メソッドはフィールドの長さを文字で返します。

値の *srcOffset* から *srcOffset+count-1* までの位置の文字が、コピー先の配列の *dstOffset* から *dstOffset+count-1* までの位置に、それぞれコピーされます。*count* の文字数がコピーされる前に、値の末尾が検出された場合は、コピー先の配列の残りは変更されないままになります。

次のいずれかに該当する場合、コード ULSQLCode.SQLE_INVALID_PARAMETER 定数とともに ULException オブジェクトがスローされ、追加先は修正されません。

- *srcOffset* が負の値である
- *dstOffset* が負の値である
- *count* が負の値である
- *dstOffset + count* の値は *dst* の長さより長い。

その他のエラーの場合は、それに応じたエラーコードとともに ULException オブジェクトがスローされます。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[ULException クラス \[328 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.8 GetDataTypeName(int) メソッド

指定されたカラムのプロバイダのデータ型の名前を返します。

構文

Visual Basic

```
Public Overrides Function GetDataTypeName (ByVal colID As Integer) As String
```

C#

```
public override string GetDataTypeName (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

カラムの ULDbType 型に対応する文字列。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetColumnULDbType\(int\) メソッド \[235 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

[ULDbType 列挙体 \[586 ページ\]](#)

1.15.9 GetDateTime(int) メソッド

指定されたカラムの値を、ミリ秒の精度の System.DateTime 型として返します。

構文

Visual Basic

```
Public Overrides Function GetDateTime (ByVal colID As Integer) As Date
```

C#

```
public override unsafe DateTime GetDateTime (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.DateTime 型として返されるカラム値。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.10 GetDbDataReader(int) メソッド

構文

Visual Basic

```
Protected Overrides Function GetDbDataReader (ByVal i As Integer) As DbDataReader
```

C#

```
protected override DbDataReader GetDbDataReader (int i)
```

1.15.11 GetDecimal(int) メソッド

指定されたカラムの値を System.Decimal 型として返します。

構文

Visual Basic

```
Public Overrides Function GetDecimal (ByVal colID As Integer) As Decimal
```

C#

```
public override decimal GetDecimal (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Decimal 型として返されるカラム値。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.12 GetDouble(int) メソッド

指定されたカラムの値を System.Double 型として返します。

構文

Visual Basic

```
Public Overrides Function GetDouble (ByVal colID As Integer) As Double
```

C#

```
public override unsafe double GetDouble (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Double 型として返されるカラム値。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.13 GetEnumerator() メソッド

ULDataReader オブジェクトの反復処理を実行する System.Collections.IEnumerator 値を返します。

構文

Visual Basic

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

C#

```
public override IEnumerator GetEnumerator ()
```

戻り値

ULDataReader オブジェクトの System.Collections.IEnumerator。

1.15.14 GetFieldType(int) メソッド

指定されたカラムに最適な System.Type 値を返します。

構文

Visual Basic

```
Public Overrides Function GetFieldType (ByVal colID As Integer) As Type
```

C#

```
public override Type GetFieldType (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

カラムの System.Type 値。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetDataTypeName\(int\) メソッド \[291 ページ\]](#)

[GetColumnULDbType\(int\) メソッド \[235 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.15 GetFloat(int) メソッド

指定されたカラムの値を System.Single 型として返します。

構文

Visual Basic

```
Public Overrides Function GetFloat (ByVal colID As Integer) As Single
```

C#

```
public override unsafe float GetFloat (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Single 型として返されるカラム値。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.16 GetGuid(int) メソッド

指定されたカラムの値を UUID (System.Guid) 型として返します。

構文

Visual Basic

```
Public Overrides Function GetGuid (ByVal colID As Integer) As Guid
```

C#

```
public override unsafe Guid GetGuid (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

GUID 型として返されるカラム値。

例外

ULException class SQL エラーが発生しました。

備考

このメソッドが有効なのは、ULDbType.UniqueIdentifier 型のカラム、または長さが 16 の ULDbType.Binary 型のカラムの場合だけです。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[GetColumnULDbType\(int\) メソッド \[235 ページ\]](#)

[GetColumnSize\(int\) メソッド \[233 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.17 GetInt16(int) メソッド

指定されたカラムの値を System.Int16 型として返します。

構文

Visual Basic

```
Public Overrides Function GetInt16 (ByVal colID As Integer) As Short
```

C#

```
public override unsafe short GetInt16 (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Int16 型として返されるカラム値。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.18 GetInt32(int) メソッド

指定されたカラムの値を System.Int32 型として返します。

構文

Visual Basic

```
Public Overrides Function GetInt32 (ByVal colID As Integer) As Integer
```

C#

```
public override unsafe int GetInt32 (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Int32 型として返されるカラム値。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.19 GetInt64(int) メソッド

指定されたカラムの値を System.Int64 型として返します。

構文

Visual Basic

```
Public Overrides Function GetInt64 (ByVal colID As Integer) As Long
```

C#

```
public override unsafe long GetInt64 (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.Int64 型として返されるカラム値。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.20 GetName(int) メソッド

指定されたカラムの名前を返します。

構文

Visual Basic

```
Public Overrides Function GetName (ByVal colID As Integer) As String
```

C#

```
public override string GetName (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

カラムの名前。カラムに名前がない場合は NULL 参照 (Visual Basic の Nothing)。SQL クエリでカラムのエイリアスが使用されている場合は、そのエイリアスが返されます。

例外

ULException class SQL エラーが発生しました。

備考

結果セットでは、すべてのカラムに名前があるとは限らず、すべてのカラム名がユニークであるとも限りません。エイリアスを使用していない場合は、非計算カラムの名前には、そのカラムの元となるテーブルの名前がプレフィクスとして付けられます。たとえば、MyTable.ID の値は、クエリ "SELECT ID FROM MyTable" の結果セットに含まれる唯一のカラムの名前です。

このメソッドは、ULCursorSchema.GetColumnName メソッドと同じです。

関連情報

[FieldCount プロパティ \[321 ページ\]](#)

[GetSchemaTable\(\) メソッド \[304 ページ\]](#)

[GetColumnName\(int\) メソッド \[230 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.21 GetOrdinal(string) メソッド

指定されたカラムのカラム ID を返します。

構文

Visual Basic

```
Public Overrides Function GetOrdinal (ByVal columnName As String) As Integer
```

C#

```
public override unsafe int GetOrdinal (string columnName)
```

パラメータ

columnName カラム名。

戻り値

指定されたカラムのカラム ID。

例外

ULException class SQL エラーが発生しました。

備考

カラム ID の範囲は、0 ~ ULDataReader.FieldCount-1 です。

結果セットでは、すべてのカラムに名前があるとは限らず、すべてのカラム名がユニークであるとも限りません。エイリアスを使用していない場合は、非計算カラムの名前には、そのカラムの元となるテーブルの名前がプレフィクスとして付けられます。たとえば、MyTable.ID の値は、クエリ "SELECT ID FROM MyTable" の結果セットに含まれる唯一のカラムの名前です。

カラムの ID とカウントは、スキーマのアップグレード中に変更されることがあります。カラムを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後に再表示します。

このメソッドは、ULCursorSchema.GetColumnID メソッドと同じです。

関連情報

[GetSchemaTable\(\) メソッド \[304 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

[GetColumnID\(string\) メソッド \[229 ページ\]](#)

1.15.22 GetRowCount(int) メソッド

UL 拡張: カーソル内のローの数を、スレッシュヨルド以内で返します。

構文

Visual Basic

```
Public Function GetRowCount (ByVal threshold As Integer) As Integer
```

C#

```
public unsafe int GetRowCount (int threshold)
```

パラメータ

threshold ローカウントのスレッシュヨルド制限。

例外

ULException class SQL エラーが発生しました。

備考

カーソル内のロー数。

RowCount プロパティではカーソルのローを通過する必要があるため、クエリが複雑で大きな負荷がかかります。GetRowCount(threshold) メソッドを使用することにより、少なくともスレッシュホールドのローが存在するかどうかを呼び出し元で判断できます。ロー数がスレッシュホールドより低い場合はロー数が返され、そうでない場合はスレッシュホールドが返されます。高いスレッシュホールドを使用して、このメソッドを再度呼び出すことができます。

スレッシュホールドが 0 の場合は、RowCount プロパティが返されます。

関連情報

[RowCount プロパティ \[324 ページ\]](#)

1.15.23 GetSchemaTable() メソッド

ULDataReader オブジェクトの各カラムのメタデータが記述された System.Data.DataTable 値を返します。

構文

Visual Basic

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

C#

```
public override DataTable GetSchemaTable ()
```

戻り値

ULDataReader の各カラムのスキーマが記述された System.Data.DataTable。

備考

GetSchemaTable メソッドは、各カラムに関するメタデータを次の順で返します。

DataTable カラム	説明
ColumnName	カラムの名前。カラムに名前がない場合は NULL 参照 (Visual Basic の Nothing)。SQL クエリでカラムのエイリアスが使用されている場合は、そのエイリアスが返されます。結果セットでは、すべてのカラムに名前があるとは限らず、すべてのカラム名がユニークであるとも限りません。
ColumnOrdinal	カラムの ID。値の範囲は、[0, FieldCount -1] です。
ColumnSize	サイズ指定されたカラムの場合、カラムの値の最大長。その他のカラムの場合、これは、そのデータ型のバイト単位のサイズです。
NumericPrecision	数値カラム (ProviderType ULDbType.Decimal または ULDbType.Numeric) の精度。カラムが数値型ではない場合は DBNull。
NumericScale	数値カラム (ProviderType ULDbType.Decimal または ULDbType.Numeric) の位取り。カラムが数値型ではない場合は DBNull。
IsUnique	カラムが取得元のテーブル (BaseTableName) でユニークな非計算カラムである場合は true。
IsKey	カラムが、結果セットのユニークキーからともに取得された結果セットの一連のカラムのいずれかである場合は true。IsKey の値が true に設定されたカラムのセットは、結果セット内のローをユニークに識別する最小限のセットである必要はありません。
BaseCatalogName	カラムが含まれているデータベース内のカタログの名前。Ultra Light.NET の場合、この値は常に DBNull です。
BaseColumnName	データベースの BaseTableName テーブルにあるカラムの元の名前。カラムが計算される場合やこの情報を特定できない場合は DBNull になります。
BaseSchemaName	カラムが含まれているデータベース内のスキーマの名前。Ultra Light.NET の場合、この値は常に DBNull です。
BaseTableName	カラムが含まれているデータベースのテーブルの名前。カラムが計算される場合と、この情報を特定できない場合は、DBNull です。
DataType	この型のカラムに最適な .NET データ型。
AllowDBNull	カラムが NULL 入力可である場合は true、NULL 入力不可である場合またはこの情報を特定できない場合は false。
ProviderType	カラムの ULDbType 値。
IsIdentity	カラムが identity カラムである場合は true、identity カラムでない場合は false。Ultra Light.NET の場合、この値は常に false です。
IsAutoIncrement	カラムがオートインクリメントカラムまたはグローバルオートインクリメントカラムである場合は true、それ以外のカラムである場合 (または、この情報を特定できない場合) は false。
IsRowVersion	書き込みできない永続的なロー識別子がカラムに含まれており、ローを識別する以外は意味を持たない値がカラムにある場合は true。Ultra Light.NET の場合、この値は常に false です。

DataTable カラム	説明
IsLong	カラムが ULDbType.LongVarchar カラムまたは ULDbType.LongBinary カラムである場合は true、それ以外のカラムの場合は false。
IsReadOnly	カラムが読み込み専用である場合は true、カラムが修正可能であるか、カラムのアクセス権を特定できない場合は false。
IsAliased	カラム名がエイリアスの場合は true、エイリアスでない場合は false。
IsExpression	カラムが式の場合は true、カラム値の場合は false。

関連情報

[Schema プロパティ \[325 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

[ULDbType 列挙体 \[586 ページ\]](#)

1.15.24 GetString(int) メソッド

指定されたカラムの値を System.String 型として返します。

構文

Visual Basic

```
Public Overrides Function GetString (ByVal colID As Integer) As String
```

C#

```
public override unsafe String GetString (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.String 型として返されるカラム値。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.25 GetTimeSpan(int) メソッド

指定されたカラムの値を、ミリ秒の精度の System.TimeSpan 型として返します。

構文

Visual Basic

```
Public Function GetTimeSpan (ByVal colID As Integer) As TimeSpan
```

C#

```
public unsafe TimeSpan GetTimeSpan (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.TimeSpan 型として返されるカラム値。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.26 GetUInt16(int) メソッド

指定されたカラムの値を System.UInt16 型として返します。

構文

Visual Basic

```
Public Function GetUInt16 (ByVal colID As Integer) As UShort
```

C#

```
public unsafe ushort GetUInt16 (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.UInt16 型として返されるカラム値。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.27 GetUInt32(int) メソッド

指定されたカラムの値を System.UInt32 型として返します。

構文

Visual Basic

```
Public Function GetUInt32 (ByVal colID As Integer) As UInteger
```

C#

```
public unsafe uint GetUInt32 (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.UInt32 型として返されるカラム値。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.28 GetUInt64(int) メソッド

指定されたカラムの値を System.UInt64 型として返します。

構文

Visual Basic

```
Public Function GetUInt64 (ByVal colID As Integer) As ULong
```

C#

```
public unsafe ulong GetUInt64 (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

System.UInt64 型として返されるカラム値。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.29 GetValue(int) メソッド

指定されたカラムの値をネイティブフォーマットで返します。

構文

Visual Basic

```
Public Overrides Function GetValue (ByVal colID As Integer) As Object
```

C#

```
public override object GetValue (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

そのカラムに最適な .NET 型としてのカラム値。カラムが NULL の場合は DBNull 型になります。

例外

ULException class SQL エラーが発生しました。

備考

このメソッドは、ULDataReader.this[int] メソッドと同じです。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.30 GetValues(object[]) メソッド

現在のローのすべてのカラム値を返します。

構文

Visual Basic

```
Public Overrides Function GetValues (ByVal values As Object()) As Integer
```

C#

```
public override int GetValues (object[] values)
```

パラメータ

values ロー全体を保持する System.Objects 配列。

戻り値

取り出されるカラム値の数。配列の長さがカラムの数 (ULDataReader.FieldCount) よりも大きい場合は、FieldCount の項目のみが取り出され、配列の残りは変更されないままになります。

例外

ArgumentNullException *values* 配列が NULL であるか、長さがゼロです。

ULException class SQL エラーが発生しました。

備考

ほとんどのアプリケーションについて、GetValues メソッドは、各カラムを個々に取り出すのではなく、すべてのカラムを取り出す効率的な方法を提供します。

結果のローに含まれるカラムの数より少ないカラムが含まれる System.Object 配列を渡すことができます。System.Object 配列が保持するデータ量のみが配列にコピーされます。また、結果のローに含まれるカラムの数より長い System.Object 配列を渡すこともできます。

このメソッドは、NULL データベースカラムに対して DBNull 型を返します。その他のカラムの場合は、カラムの値をネイティブフォーマットで返します。

関連情報

[FieldCount プロパティ \[321 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[GetValue\(int\) メソッド \[311 ページ\]](#)

1.15.31 IsDBNull(int) メソッド

指定されたカラムの値が NULL かどうかをチェックします。

構文

Visual Basic

```
Public Overrides Function IsDBNull (ByVal colID As Integer) As Boolean
```

C#

```
public override unsafe bool IsDBNull (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

戻り値

値が NULL の場合は true、NULL でない場合は false。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.32 MoveAfterLast() メソッド

UL 拡張: カーソルの最後のローの後に、カーソルを配置します。

構文

Visual Basic

```
Public Sub MoveAfterLast ()
```

C#

```
public void MoveAfterLast ()
```

例外

ULException class SQL エラーが発生しました。

1.15.33 MoveBeforeFirst() メソッド

UL 拡張: カーソルの最初のローの前に、カーソルを配置します。

構文

Visual Basic

```
Public Sub MoveBeforeFirst ()
```

C#

```
public void MoveBeforeFirst ()
```

例外

ULException class SQL エラーが発生しました。

1.15.34 MoveFirst() メソッド

UL 拡張: カーソルの最初のローに、カーソルを配置します。

構文

Visual Basic

```
Public Function MoveFirst () As Boolean
```

C#

```
public unsafe bool MoveFirst ()
```

戻り値

成功した場合は true、失敗した場合は false。たとえば、ローがない場合、メソッドは失敗します。

例外

ULException class SQL エラーが発生しました。

1.15.35 MoveLast() メソッド

UL 拡張: カーソルの最後のローに、カーソルを配置します。

構文

Visual Basic

```
Public Function MoveLast () As Boolean
```

C#

```
public unsafe bool MoveLast ()
```

戻り値

成功した場合は true、失敗した場合は false。たとえば、ローがない場合、メソッドは失敗します。

例外

ULException class SQL エラーが発生しました。

1.15.36 MoveNext() メソッド

UL 拡張: カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。

構文

Visual Basic

```
Public Function MoveNext () As Boolean
```

C#

```
public unsafe bool MoveNext ()
```

戻り値

成功した場合は true、失敗した場合は false。たとえば、それ以上ローがない場合、メソッドは失敗します。

例外

ULException class SQL エラーが発生しました。

備考

このメソッドは、ULDataReader.Read メソッドと同じです。

関連情報

[Read\(\) メソッド \[319 ページ\]](#)

1.15.37 MovePrevious() メソッド

UL 拡張: カーソルを前のローに配置するか、最初のローの前に配置します。

構文

Visual Basic

```
Public Function MovePrevious () As Boolean
```

C#

```
public unsafe bool MovePrevious ()
```

戻り値

成功した場合は true、失敗した場合は false。たとえば、それ以上ローがない場合、メソッドは失敗します。

例外

ULException class SQL エラーが発生しました。

1.15.38 MoveRelative(int) メソッド

UL 拡張: 現在のローを基準としてカーソルを配置します。

構文

Visual Basic

```
Public Function MoveRelative (ByVal offset As Integer) As Boolean
```

C#

```
public unsafe bool MoveRelative (int offset)
```

パラメータ

offset 移動するローの数。負の値を指定すると、後方に移動します。

戻り値

成功した場合は true、失敗した場合は false。たとえば、最初または最後のローを超えて移動する場合、メソッドは失敗します。

例外

ULException class SQL エラーが発生しました。

備考

指定された移動先にローがない場合には、false が返されます。その場合のカーソル位置は、*offset* が正であるときには最後のローの後 (ULDataReader.IsEOF メソッド) になり、*offset* 値が負であるときには最初のローの前 (ULDataReader.IsBOF メソッド) になります。

関連情報

[IsEOF プロパティ \[323 ページ\]](#)

[IsBOF プロパティ \[322 ページ\]](#)

1.15.39 NextResult() メソッド

バッチ SQL 文の結果を読み込むときに ULDataReader オブジェクトを次の結果に進めます。

構文

Visual Basic

```
Public Overrides Function NextResult () As Boolean
```

C#

```
public override bool NextResult ()
```

戻り値

さらに結果セットがある場合は true、そうでない場合は false。Ultra Light.NET では、常に false が返されます。

例外

ULException class ULDataReader オブジェクトが開かれていません。

備考

UL 拡張: Ultra Light.NET ではバッチ SQL 文はサポートされていないため、ULDataReader オブジェクトは常に、最初の (唯一の) 結果セットに配置されます。NextResult メソッドの呼び出しは効果がありません。

1.15.40 Read() メソッド

カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。

構文

Visual Basic

```
Public Overrides Function Read () As Boolean
```

C#

```
public override bool Read ()
```

戻り値

成功した場合は true、失敗した場合は false。たとえば、それ以上ローがない場合、メソッドは失敗します。

例外

ULException class SQL エラーが発生しました。

備考

このメソッドは、ULDataReader.MoveNext メソッドと同じです。

関連情報

[MoveNext\(\) メソッド \[316 ページ\]](#)

1.15.41 Validate メソッド

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
protected void	Validate() [320 ページ]	
protected void	Validate(int) [320 ページ]	

このセクションの内容:

[Validate\(\) メソッド \[320 ページ\]](#)

[Validate\(int\) メソッド \[320 ページ\]](#)

1.15.41.1 Validate() メソッド

構文

Visual Basic

```
Protected Sub Validate ()
```

C#

```
protected void Validate ()
```

1.15.41.2 Validate(int) メソッド

構文

Visual Basic

```
Protected Sub Validate (ByVal colID As Integer)
```


C#

```
protected void Validate (int colID)
```

1.15.42 Depth プロパティ

現在のローのネストの深さを返します。

 構文

Visual Basic

```
Public ReadOnly Overrides Property Depth As Integer
```

C#

```
public override int Depth {get;}
```

備考

最も外側のテーブルの深さは 0 です。

Ultra Light.NET のすべての結果セットの深さは 0 です。

1.15.43 FieldCount プロパティ

このカーソル内のカラム数を返します。

 構文

Visual Basic

```
Public ReadOnly Overrides Property FieldCount As Integer
```

C#

```
public override int FieldCount {get;}
```

戻り値

整数としてのカーソル内のカラム数。カーソルが閉じている場合は 0 を返します。

備考

このメソッドは、ULCursorSchema.ColumnCount メソッドと同じです。

関連情報

[ColumnCount プロパティ \[237 ページ\]](#)

1.15.44 HasRows プロパティ

ULDataReader オブジェクトに 1 つまたは複数のローがあるかどうかをチェックします。

構文

Visual Basic

```
Public ReadOnly Overrides Property HasRows As Boolean
```

C#

```
public override unsafe bool HasRows {get;}
```

備考

結果セットに少なくとも 1 つのローがある場合は True、ローがない場合は false。

1.15.45 IsBOF プロパティ

UL 拡張: 現在のローの位置が最初のローの前かどうかをチェックします。

構文

Visual Basic

```
Public ReadOnly Property IsBOF As Boolean
```

C#

```
public unsafe bool IsBOF {get;}
```

備考

現在のローの位置が最初のローの前になる場合は true、それ以外の場合は false。

1.15.46 IsClosed プロパティ

カーソルが現在開いているかどうかを確認します。

構文

Visual Basic

```
Public ReadOnly Overrides Property IsClosed As Boolean
```

C#

```
public override bool IsClosed {get;}
```

備考

カーソルが現在開いている場合は true、閉じている場合は false。

1.15.47 IsEOF プロパティ

UL 拡張: 現在のローの位置が最後のローの後かどうかをチェックします。

構文

Visual Basic

```
Public ReadOnly Property IsEOF As Boolean
```

C#

```
public unsafe bool IsEOF {get;}
```

備考

現在のローの位置が最後のローの後になる場合は true、それ以外の場合は false。

1.15.48 RecordsAffected プロパティ

SQL 文の実行によって変更、挿入、または削除されたローの数を返します。

構文

Visual Basic

```
Public ReadOnly Overrides Property RecordsAffected As Integer
```

C#

```
public override int RecordsAffected {get;}
```

備考

SELECT 文または CommandType.TableDirect テーブルの場合、この値は -1 です。

SQL 文の実行によって変更、挿入、または削除されたローの数。

1.15.49 RowCount プロパティ

UL 拡張: カーソル内のローの数を返します。

構文

Visual Basic

```
Public ReadOnly Property RowCount As Integer
```

C#

```
public int RowCount {get;}
```

備考

カーソル内のロー数。

RowCount メソッドを使用すると、古いローを削除して領域を節約するタイミングを判断できます。

ULConnection.StopSynchronizationDelete メソッドを使用すると、統合データベースからは削除しないで Ultra Light データベースから古いローを削除できます。

関連情報

[StartSynchronizationDelete\(\) メソッド \[169 ページ\]](#)

[StopSynchronizationDelete\(\) メソッド \[170 ページ\]](#)

1.15.50 Schema プロパティ

UL 拡張: このカーソルのスキーマを保持します。

構文

Visual Basic

```
Public ReadOnly Property Schema As ULCursorSchema
```

C#

```
public ULCursorSchema Schema {get;}
```

備考

結果セットの場合、結果セットのスキーマを表す `ULResultSetSchema` オブジェクト。テーブルの場合、テーブルのスキーマを表す `ULTableSchema` オブジェクト。

このプロパティは、`ULDataReader.GetSchemaTable` メソッドからの結果に示されない Ultra Light.NET の詳細情報を含め、カーソルの完全なスキーマを表します。

関連情報

[ULTableSchema クラス \[555 ページ\]](#)

[GetSchemaTable\(\) メソッド \[304 ページ\]](#)

[ULResultSetSchema クラス \[471 ページ\]](#)

1.15.51 this プロパティ

指定されたカラムの値をネイティブフォーマットで返します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public override object	this[int colID] [326 ページ]	指定されたカラムの値をネイティブフォーマットで返します。
public override object	this[string name] [327 ページ]	指定された名前のカラムの値をネイティブフォーマットで返します。

このセクションの内容:

[this\[int colID\] プロパティ \[326 ページ\]](#)

指定されたカラムの値をネイティブフォーマットで返します。

[this\[string name\] プロパティ \[327 ページ\]](#)

指定された名前のカラムの値をネイティブフォーマットで返します。

1.15.51.1 this[int colID] プロパティ

指定されたカラムの値をネイティブフォーマットで返します。

構文

Visual Basic

```
Public ReadOnly Overrides Property Item (ByVal colID As Integer) As Object
```

C#

```
public override object this[int colID] {get;}
```

戻り値

そのカラムに最適な .NET 型としてのカラム値。カラムが NULL の場合は DBNull。

備考

C# では、このプロパティは `ULDataReader` クラスのインデクサです。

このメソッドは、機能的には `ULDataReader.GetValue(int)` メソッドと同じです。

関連情報

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[GetValue\(int\) メソッド \[311 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.15.51.2 this[string name] プロパティ

指定された名前のカラムの値をネイティブフォーマットで返します。

構文

Visual Basic

```
Public ReadOnly Overrides Property Item (ByVal nameAs String) As Object
```

C#

```
public override object this[string name] {get;}
```

戻り値

そのカラムに最適な .NET 型としてのカラム値。カラムが NULL の場合は `DBNull`。

備考

C# では、このプロパティは `ULDataReader` オブジェクトのインデクサです。

結果セットでは、すべてのカラムに名前があるとは限らず、すべてのカラム名がユニークであるとも限りません。エイリアスを使用していない場合は、非計算カラムの名前には、そのカラムの元となるテーブルの名前がプレフィクスとして付けられます。たとえば、`MyTable.ID` の値は、"SELECT ID FROM MyTable" クエリの結果セットに含まれる唯一のカラムの名前です。

カラムに何回もアクセスするときは、名前ではなく、カラム ID でアクセスすると効率が良くなります。

このメソッドは次と同じです。

```
dataReader.GetValue( dataReader.GetOrdinal( name ) )
```

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetValue\(int\) メソッド \[311 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

1.16 ULErrorException クラス

Ultra Light.NET データベースによって返される SQL エラーを表します。

構文

Visual Basic

```
Public NotInheritable Class ULErrorException Inherits System.ApplicationException
```

C#

```
public sealed class ULErrorException : System.ApplicationException
```

メンバー

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

プロパティ

変更子とタイプ	プロパティ	説明
public ULSQLCode	NativeError [329 ページ]	データベースによって返される SQLCODE 値を返します。
public override string	Source [329 ページ]	エラーを生成したプロバイダの名前を返します。

備考

NativeError プロパティにエラーが返されたことを示す SQLCODE。

.NET Compact Framework では、このクラスはシリアル化可能ではありません。

このセクションの内容:

[NativeError プロパティ \[329 ページ\]](#)

データベースによって返される SQLCODE 値を返します。

[Source プロパティ \[329 ページ\]](#)

エラーを生成したプロバイダの名前を返します。

関連情報

[NativeError プロパティ \[329 ページ\]](#)

1.16.1 NativeError プロパティ

データベースによって返される SQLCODE 値を返します。

構文

Visual Basic

```
Public ReadOnly Property NativeError As ULSQLCode
```

C#

```
public ULSQLCode NativeError {get;}
```

備考

データベースによって返される ULSQLCode 値。

1.16.2 Source プロパティ

エラーを生成したプロバイダの名前を返します。

構文

Visual Basic

```
Public ReadOnly Overrides Property Source As String
```

C#

```
public override string Source {get;}
```

備考

プロバイダが Ultra Light.NET であることを示す文字列値。

1.17 ULFactory クラス

データソースクラスの Sap.Data.UltraLite プロバイダの実装のインスタンスを作成する、メソッドのセットを表します。

構文

Visual Basic

```
Public Class ULFactory Inherits System.Data.Common.DbProviderFactory
```

C#

```
public class ULFactory : System.Data.Common.DbProviderFactory
```

メンバー

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

変数

変更子とタイプ	変数	説明
public static readonly ULFactory	Instance	<p>ULFactory クラスのシングルトンインスタンスを表します。</p> <p>ULFactory クラスは、.NET Compact Framework 2.0 では使用できません。</p> <p>ULFactory はシングルトンクラスであり、このクラスにはこのインスタンスだけが存在できることを意味します。</p> <p>通常、このフィールドを直接使用することはありません。代わりに、ULFactory クラスのこのインスタンスへの参照を取得するには、System.Data.Common.DbProviderFactories.GetFactory(String) メソッドを使用します。</p>

メソッド

変更子とタイプ	メソッド	説明
public override DbCommand	CreateCommand() [333 ページ]	厳密に型指定された System.Data.Common.DbCommand インスタンスを返します。
public override DbCommandBuilder	CreateCommandBuilder() [333 ページ]	厳密に型指定された System.Data.Common.DbCommandBuilder インスタンスを返します。
public override DbConnection	CreateConnection() [334 ページ]	厳密に型指定された System.Data.Common.DbConnection インスタンスを返します。
public override DbConnectionStringBuilder	CreateConnectionStringBuilder() [334 ページ]	厳密に型指定された System.Data.Common.DbConnectionStringBuilder インスタンスを返します。
public override DbDataAdapter	CreateDataAdapter() [335 ページ]	厳密に型指定された System.Data.Common.DbDataAdapter インスタンスを返します。
public override DbParameter	CreateParameter() [336 ページ]	厳密に型指定された System.Data.Common.DbParameter インスタンスを返します。

プロパティ

変更子とタイプ	プロパティ	説明
public override bool	CanCreateDataSourceEnumerator [336 ページ]	Ultra Light.NET で DbDataSourceEnumerator クラスがサポートされないことを示す false を返します。

備考

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

ADO.NET 2.0 には System.Data.Common.DbProviderFactories と System.Data.Common.DbProviderFactory という 2 つクラスが新しく追加され、プロバイダに依存しないコードを簡単に作成できるようになりました。これらを Ultra Light.NET で使用するには、GetFactory メソッドに渡されるプロバイダの不変名として Sap.Data.UltraLite を指定します。次に例を示します。

```
' Visual Basic
Dim factory As DbProviderFactory =
    DbProviderFactories.GetFactory( "Sap.Data.UltraLite" )
Dim conn As DbConnection =
    factory.CreateConnection()
```

対応する C# 言語のコードを次に示します。

```
// C#
DbProviderFactory factory =
    DbProviderFactories.GetFactory( "Sap.Data.UltraLite" );
DbConnection conn = factory.CreateConnection();
```

この例の中の conn は、ULConnection オブジェクトとして作成されます。

ADO.NET 2.0 におけるプロバイダファクトリと汎用プログラミングについては、<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnvs05/html/vsgenerics.asp> を参照してください。Ultra Light.NET では、CreateCommandBuilder、CreateDataSourceEnumerator、CreatePermission の各メソッドをサポートしていません。独自の属性: sealed

このセクションの内容:

[CreateCommand\(\) メソッド \[333 ページ\]](#)

厳密に型指定された System.Data.Common.DbCommand インスタンスを返します。

[CreateCommandBuilder\(\) メソッド \[333 ページ\]](#)

厳密に型指定された System.Data.Common.DbCommandBuilder インスタンスを返します。

[CreateConnection\(\) メソッド \[334 ページ\]](#)

厳密に型指定された System.Data.Common.DbConnection インスタンスを返します。

[CreateConnectionStringBuilder\(\) メソッド \[334 ページ\]](#)

厳密に型指定された System.Data.Common.DbConnectionStringBuilder インスタンスを返します。

[CreateDataAdapter\(\) メソッド \[335 ページ\]](#)

厳密に型指定された System.Data.Common.DbDataAdapter インスタンスを返します。

[CreateParameter\(\) メソッド \[336 ページ\]](#)

厳密に型指定された System.Data.Common.DbParameter インスタンスを返します。

[CanCreateDataSourceEnumerator プロパティ \[336 ページ\]](#)

Ultra Light.NET で DbDataSourceEnumerator クラスがサポートされないことを示す false を返します。

1.17.1 CreateCommand() メソッド

厳密に型指定された System.Data.Common.DbCommand インスタンスを返します。

構文

Visual Basic

```
Public Overrides Function CreateCommand () As DbCommand
```

C#

```
public override DbCommand CreateCommand ()
```

戻り値

DbCommand として型指定された新しい ULCommand インスタンス。

関連情報

[ULCommand クラス \[58 ページ\]](#)

1.17.2 CreateCommandBuilder() メソッド

厳密に型指定された System.Data.Common.DbCommandBuilder インスタンスを返します。

構文

Visual Basic

```
Public Overrides Function CreateCommandBuilder () As DbCommandBuilder
```

C#

```
public override DbCommandBuilder CreateCommandBuilder ()
```

戻り値

DbCommandBuilder として型指定された新しい ULCommandBuilder インスタンス。

関連情報

[ULCommandBuilder クラス \[105 ページ\]](#)

1.17.3 CreateConnection() メソッド

厳密に型指定された System.Data.Common.DbConnection インスタンスを返します。

構文

Visual Basic

```
Public Overrides Function CreateConnection () As DbConnection
```

C#

```
public override DbConnection CreateConnection ()
```

戻り値

DbConnection として型指定された新しい ULConnection インスタンス。

関連情報

[ULConnection クラス \[121 ページ\]](#)

1.17.4 CreateConnectionStringBuilder() メソッド

厳密に型指定された System.Data.Common.DbConnectionStringBuilder インスタンスを返します。

構文

Visual Basic

```
Public Overrides Function CreateConnectionStringBuilder () As  
DbConnectionStringBuilder
```

C#

```
public override DbConnectionStringBuilder CreateConnectionStringBuilder ()
```

戻り値

DbConnectionStringBuilder として型指定された新しい ULConnectionStringBuilder インスタンス。

関連情報

[ULConnectionStringBuilder クラス \[197 ページ\]](#)

1.17.5 CreateDataAdapter() メソッド

厳密に型指定された System.Data.Common.DbDataAdapter インスタンスを返します。

構文

Visual Basic

```
Public Overrides Function CreateDataAdapter () As DbDataAdapter
```

C#

```
public override DbDataAdapter CreateDataAdapter ()
```

戻り値

DbDataAdapter として型指定された新しい ULDataAdapter インスタンス。

関連情報

[ULDataAdapter クラス \[238 ページ\]](#)

1.17.6 CreateParameter() メソッド

厳密に型指定された System.Data.Common.DbParameter インスタンスを返します。

構文

Visual Basic

```
Public Overrides Function CreateParameter () As DbParameter
```

C#

```
public override DbParameter CreateParameter ()
```

戻り値

DbParameter として型指定された新しい ULParameter インスタンス。

関連情報

[ULParameter クラス \[389 ページ\]](#)

1.17.7 CanCreateDataSourceEnumerator プロパティ

Ultra Light.NET で DbDataSourceEnumerator クラスがサポートされないことを示す false を返します。

構文

Visual Basic

```
Public ReadOnly Overrides Property CanCreateDataSourceEnumerator As Boolean
```

C#

```
public override bool CanCreateDataSourceEnumerator {get;}
```

備考

false は ULFactory クラスに CreateDataSourceEnumerator メソッドが実装されていないことを示します。

1.18 ULFileTransfer クラス

UL 拡張: Mobile Link サーバを使用して、リモートデータベースからファイルを転送します。

構文

Visual Basic

```
Public NotInheritable Class ULFileTransfer
```

C#

```
public sealed class ULFileTransfer
```

メンバー

ULFileTransfer 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

変数とタイプ	コンストラクタ	説明
public	ULFileTransfer() [340 ページ]	ULFileTransfer オブジェクトを初期化します。

メソッド

変数とタイプ	メソッド	説明
public bool	DownloadFile [341 ページ]	このオブジェクトのプロパティで指定されたファイルをダウンロードします。
public bool	UploadFile [344 ページ]	このオブジェクトのプロパティで指定されたファイルをアップロードします。

プロパティ

変数とタイプ	プロパティ	説明
public unsafe String[]	AuthenticationParms [347 ページ]	カスタムユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。
public ULAuthStatusCode	AuthStatus [348 ページ]	前回行われたファイル転送の認証ステータスコードを返します。
public long	AuthValue [348 ページ]	カスタムユーザ認証同期スクリプトからの戻り値を返します。
public ushort	FileAuthCode [349 ページ]	前回行われたファイル転送の authenticate_file_transfer スクリプトからの戻り値を返します。

変更子とタイプ	プロパティ	説明
public string	FileName [349 ページ]	ダウンロードするファイルの名前を指定します。
public string	LocalFileName [350 ページ]	ダウンロードファイルのローカル名を指定します。
public string	LocalPath [351 ページ]	ファイルをダウンロードする場所を指定します。
public string	パスワード [351 ページ]	UserName 値で指定されたユーザの Mobile Link パスワードです。
public string	RemoteKey [352 ページ]	Mobile Link サーバが Mobile Link クライアントをユニークに識別するキーです。
public bool	ResumePartialDownload [353 ページ]	前の部分的なダウンロードを再開するか、破棄するかを指定します。
public UStreamType	Stream [353 ページ]	ファイル転送に使用する Mobile Link 同期ストリームを指定します。
public UStreamErrorCode	StreamErrorCode [354 ページ]	前回行われたファイル転送のストリーム自体によってレポートされるエラーを返します。
public int	StreamErrorSystem [355 ページ]	ストリームエラーシステム固有のコードを返します。
public string	StreamParms [355 ページ]	同期ストリームの設定パラメータを指定します。
public bool	TransferredFile [356 ページ]	前回行われたファイル転送時にファイルが実際にダウンロードされたかどうかを確認します。
public string	UserName [357 ページ]	Mobile Link サーバが Mobile Link クライアントを識別するユーザ名です。
public string	Version [357 ページ]	使用する同期スクリプトを指定します。

備考

ファイルを転送するためにデータベースに接続する必要はありません。ただし、Ultra Light ランタイムを使用した Ultra Light データベースをアプリケーションで使用する場合は、この API や Ultra Light.NET API を使用する前に、ULDatabaseManager.RuntimeType に適切な値を設定する必要があります。

ファイルを転送するには、ULFileTransfer.FileName、ULFileTransfer.Stream、ULFileTransfer.UserName、ULFileTransfer.Version の値を設定する必要があります。

このセクションの内容:

[ULFileTransfer\(\) コンストラクタ \[340 ページ\]](#)

ULFileTransfer オブジェクトを初期化します。

[DownloadFile メソッド \[341 ページ\]](#)

このオブジェクトのプロパティで指定されたファイルをダウンロードします。

[UploadFile メソッド \[344 ページ\]](#)

このオブジェクトのプロパティで指定されたファイルをアップロードします。

[AuthenticationParms プロパティ \[347 ページ\]](#)

カスタムユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。

[AuthStatus プロパティ \[348 ページ\]](#)

前回行われたファイル転送の認証ステータスコードを返します。

[AuthValue プロパティ \[348 ページ\]](#)

カスタムユーザ認証同期スクリプトからの戻り値を返します。

[FileAuthCode プロパティ \[349 ページ\]](#)

前回行われたファイル転送の authenticate_file_transfer スクリプトからの戻り値を返します。

[FileName プロパティ \[349 ページ\]](#)

ダウンロードするファイルの名前を指定します。

[LocalFileName プロパティ \[350 ページ\]](#)

ダウンロードファイルのローカル名を指定します。

[LocalPath プロパティ \[351 ページ\]](#)

ファイルをダウンロードする場所を指定します。

[Password プロパティ \[351 ページ\]](#)

UserName 値で指定されたユーザの Mobile Link パスワードです。

[RemoteKey プロパティ \[352 ページ\]](#)

Mobile Link サーバが Mobile Link クライアントをユニークに識別するキーです。

[ResumePartialDownload プロパティ \[353 ページ\]](#)

前の部分的なダウンロードを再開するか、破棄するかを指定します。

[Stream プロパティ \[353 ページ\]](#)

ファイル転送に使用する Mobile Link 同期ストリームを指定します。

[StreamErrorCode プロパティ \[354 ページ\]](#)

前回行われたファイル転送のストリーム自体によってレポートされるエラーを返します。

[StreamErrorSystem プロパティ \[355 ページ\]](#)

ストリームエラーシステム固有のコードを返します。

[StreamParms プロパティ \[355 ページ\]](#)

同期ストリームの設定パラメータを指定します。

[TransferredFile プロパティ \[356 ページ\]](#)

前回行われたファイル転送時にファイルが実際にダウンロードされたかどうかを確認します。

[UserName プロパティ \[357 ページ\]](#)

Mobile Link サーバが Mobile Link クライアントを識別するユーザ名です。

[Version プロパティ \[357 ページ\]](#)

使用する同期スクリプトを指定します。

関連情報

[FileName プロパティ \[349 ページ\]](#)
[Stream プロパティ \[353 ページ\]](#)
[UserName プロパティ \[357 ページ\]](#)
[Version プロパティ \[357 ページ\]](#)

1.18.1 ULFileTransfer() コンストラクタ

ULFileTransfer オブジェクトを初期化します。

構文

Visual Basic

```
Public Sub ULFileTransfer ()
```

C#

```
public ULFileTransfer ()
```

備考

接続を開いてから、データベース操作を実行してください。

ファイルを転送するためにデータベースに接続する必要はありません。ただし、Ultra Light ランタイムを使用した Ultra Light データベースをアプリケーションで使用する場合は、この API や Ultra Light.NET API を使用する前に、ULDatabaseManager.RuntimeType に適切な値を設定する必要があります。

ULFileTransfer オブジェクトに ULFileTransfer.FileName、ULFileTransfer.Stream、ULFileTransfer.UserName、ULFileTransfer.Version の値が設定されていないと、ファイルを転送できません。

関連情報

[DownloadFile\(\) メソッド \[341 ページ\]](#)
[UploadFile\(\) メソッド \[345 ページ\]](#)
[FileName プロパティ \[349 ページ\]](#)
[Stream プロパティ \[353 ページ\]](#)
[UserName プロパティ \[357 ページ\]](#)
[Version プロパティ \[357 ページ\]](#)

1.18.2 DownloadFile メソッド

このオブジェクトのプロパティで指定されたファイルをダウンロードします。

オーバーロードリスト

変数とタイプ	オーバーロード名	説明
public bool	DownloadFile() [341 ページ]	このオブジェクトのプロパティで指定されたファイルをダウンロードします。
public bool	DownloadFile(ULFileTransferProgressListener) [343 ページ]	指定されたリスナに送信されるプログレスイベントとともに、このオブジェクトのプロパティで指定されたファイルをダウンロードします。

このセクションの内容:

[DownloadFile\(\) メソッド \[341 ページ\]](#)

このオブジェクトのプロパティで指定されたファイルをダウンロードします。

[DownloadFile\(ULFileTransferProgressListener\) メソッド \[343 ページ\]](#)

指定されたリスナに送信されるプログレスイベントとともに、このオブジェクトのプロパティで指定されたファイルをダウンロードします。

1.18.2.1 DownloadFile() メソッド

このオブジェクトのプロパティで指定されたファイルをダウンロードします。

構文

Visual Basic

```
Public Function DownloadFile () As Boolean
```

C#

```
public bool DownloadFile ()
```

戻り値

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

備考

ULFileTransfer.FileName 値で指定されたファイルが、Mobile Link サーバによって ULFileTransfer.LocalPath 値にダウンロードされます。このとき、ULFileTransfer.Stream、ULFileTransfer.UserName、ULFileTransfer.Password、および ULFileTransfer.Version の各値が使用されます。

ファイルが破損することを防ぐため、Ultra Light.NET はテンポラリファイルにダウンロードし、ダウンロードが完了したときのみローカルファイルを置換します。ダウンロードに影響を及ぼすその他のプロパティは、ULFileTransfer.LocalFileName、ULFileTransfer.AuthenticationParms、ULFileTransfer.ResumePartialDownload です。

結果の詳細なステータスは、このオブジェクトの ULFileTransfer.AuthStatus、ULFileTransfer.AuthStatus、ULFileTransfer.FileAuthCode、ULFileTransfer.TransferredFile、ULFileTransfer.StreamErrorCode、ULFileTransfer.StreamErrorSystem の各値でレポートされます。

関連情報

[DownloadFile\(ULFileTransferProgressListener\) メソッド \[343 ページ\]](#)

[FileName プロパティ \[349 ページ\]](#)

[LocalPath プロパティ \[351 ページ\]](#)

[Stream プロパティ \[353 ページ\]](#)

[UserName プロパティ \[357 ページ\]](#)

[Password プロパティ \[351 ページ\]](#)

[Version プロパティ \[357 ページ\]](#)

[LocalFileName プロパティ \[350 ページ\]](#)

[AuthenticationParms プロパティ \[347 ページ\]](#)

[ResumePartialDownload プロパティ \[353 ページ\]](#)

[AuthStatus プロパティ \[348 ページ\]](#)

[AuthValue プロパティ \[348 ページ\]](#)

[FileAuthCode プロパティ \[349 ページ\]](#)

[TransferredFile プロパティ \[356 ページ\]](#)

[StreamErrorCode プロパティ \[354 ページ\]](#)

[StreamErrorSystem プロパティ \[355 ページ\]](#)

[StreamErrorCode プロパティ \[354 ページ\]](#)

[UploadFile\(\) メソッド \[345 ページ\]](#)

1.18.2.2 DownloadFile(ULFileTransferProgressListener) メソッド

指定されたリスナに送信されるプログレスイベントとともに、このオブジェクトのプロパティで指定されたファイルをダウンロードします。

構文

Visual Basic

```
Public Function DownloadFile (ByVal listener As  
ULFileTransferProgressListener) As Boolean
```

C#

```
public bool DownloadFile (ULFileTransferProgressListener listener)
```

パラメータ

listener ファイル転送プログレスイベントを受信するオブジェクト。

戻り値

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

備考

ULFileTransfer.FileName 値で指定されたファイルが、Mobile Link サーバによって ULFileTransfer.LocalPath 値にダウンロードされます。このとき、ULFileTransfer.Stream、ULFileTransfer.UserName、ULFileTransfer.Password、および ULFileTransfer.Version の各値が使用されます。

ファイルが破損することを防ぐため、Ultra Light.NET はテンポラリファイルにダウンロードし、ダウンロードが完了したときのみローカルファイルを置換します。ダウンロードに影響を及ぼすその他のプロパティは、ULFileTransfer.LocalFileName、ULFileTransfer.AuthenticationParms、ULFileTransfer.ResumePartialDownload です。

結果の詳細なステータスは、このオブジェクトの ULFileTransfer.AuthStatus、ULFileTransfer.AuthStatus、ULFileTransfer.FileAuthCode、ULFileTransfer.TransferredFile、ULFileTransfer.StreamErrorCode、ULFileTransfer.StreamErrorSystem の各値でレポートされます。

エラーが発生した場合、リスナにデータが送信されないことがあります。

関連情報

[FileName プロパティ \[349 ページ\]](#)
[LocalPath プロパティ \[351 ページ\]](#)
[Stream プロパティ \[353 ページ\]](#)
[UserName プロパティ \[357 ページ\]](#)
[Password プロパティ \[351 ページ\]](#)
[Version プロパティ \[357 ページ\]](#)
[LocalFileName プロパティ \[350 ページ\]](#)
[AuthenticationParms プロパティ \[347 ページ\]](#)
[ResumePartialDownload プロパティ \[353 ページ\]](#)
[AuthStatus プロパティ \[348 ページ\]](#)
[AuthValue プロパティ \[348 ページ\]](#)
[FileAuthCode プロパティ \[349 ページ\]](#)
[TransferredFile プロパティ \[356 ページ\]](#)
[StreamErrorCode プロパティ \[354 ページ\]](#)
[StreamErrorSystem プロパティ \[355 ページ\]](#)
[StreamErrorCode プロパティ \[354 ページ\]](#)
[UploadFile\(\) メソッド \[345 ページ\]](#)

1.18.3 UploadFile メソッド

このオブジェクトのプロパティで指定されたファイルをアップロードします。

オーバーロードリスト

変数とタイプ	オーバーロード名	説明
public bool	UploadFile() [345 ページ]	このオブジェクトのプロパティで指定されたファイルをアップロードします。
public bool	UploadFile(ULFileTransferProgressListener) [346 ページ]	指定されたリスナに送信されるプログ्रेसイベントとともに、このオブジェクトのプロパティで指定されたファイルをアップロードします。

このセクションの内容:

[UploadFile\(\) メソッド \[345 ページ\]](#)

このオブジェクトのプロパティで指定されたファイルをアップロードします。

[UploadFile\(ULFileTransferProgressListener\) メソッド \[346 ページ\]](#)

指定されたリスナに送信されるプログ्रेसイベントとともに、このオブジェクトのプロパティで指定されたファイルをアップロードします。

1.18.3.1 UploadFile() メソッド

このオブジェクトのプロパティで指定されたファイルをアップロードします。

構文

Visual Basic

```
Public Function UploadFile () As Boolean
```

C#

```
public bool UploadFile ()
```

戻り値

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

備考

ULFileTransfer.FileName 値で指定されたファイルが、ULFileTransfer.LocalPath 値から Mobile Link サーバにアップロードされます。このとき、ULFileTransfer.Stream、ULFileTransfer.UserName、ULFileTransfer.Password、および ULFileTransfer.Version の各値が使用されます。

結果の詳細なステータスは、このオブジェクトの ULFileTransfer.AuthStatus、ULFileTransfer.AuthStatus、ULFileTransfer.FileAuthCode、ULFileTransfer.TransferredFile、ULFileTransfer.StreamErrorCode、ULFileTransfer.StreamErrorSystem の各値でレポートされます。

関連情報

[UploadFile\(ULFileTransferProgressListener\) メソッド \[346 ページ\]](#)

[FileName プロパティ \[349 ページ\]](#)

[LocalPath プロパティ \[351 ページ\]](#)

[Stream プロパティ \[353 ページ\]](#)

[UserName プロパティ \[357 ページ\]](#)

[Password プロパティ \[351 ページ\]](#)

[Version プロパティ \[357 ページ\]](#)

[LocalFileName プロパティ \[350 ページ\]](#)

[AuthenticationParms プロパティ \[347 ページ\]](#)

[ResumePartialDownload プロパティ \[353 ページ\]](#)

[AuthStatus プロパティ \[348 ページ\]](#)

[AuthValue プロパティ \[348 ページ\]](#)

[FileAuthCode プロパティ \[349 ページ\]](#)

[TransferredFile プロパティ \[356 ページ\]](#)

[StreamErrorCode プロパティ \[354 ページ\]](#)

[StreamErrorSystem プロパティ \[355 ページ\]](#)

[StreamErrorCode プロパティ \[354 ページ\]](#)

1.18.3.2 UploadFile(ULFileTransferProgressListener) メソッド

指定されたリスナに送信されるプログレスイベントとともに、このオブジェクトのプロパティで指定されたファイルをアップロードします。

構文

Visual Basic

```
Public Function UploadFile (ByVal listener As ULFileTransferProgressListener)  
As Boolean
```

C#

```
public bool UploadFile (ULFileTransferProgressListener listener)
```

パラメータ

listener ファイル転送プログレスイベントを受信するオブジェクト。

戻り値

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

備考

ULFileTransfer.FileName 値で指定されたファイルが、ULFileTransfer.LocalPath 値から Mobile Link サーバにアップロードされます。このとき、ULFileTransfer.Stream、ULFileTransfer.UserName、ULFileTransfer.Password、および ULFileTransfer.Version の各値が使用されます。

結果の詳細なステータスは、このオブジェクトの `ULFileTransfer.AuthStatus`、`ULFileTransfer.AuthStatus`、`ULFileTransfer.FileAuthCode`、`ULFileTransfer.TransferredFile`、`ULFileTransfer.StreamErrorCode`、`ULFileTransfer.StreamErrorSystem` の各値でレポートされます。

エラーが発生した場合、リスナにデータが送信されないことがあります。

関連情報

[UploadFile\(\) メソッド \[345 ページ\]](#)
[FileName プロパティ \[349 ページ\]](#)
[LocalPath プロパティ \[351 ページ\]](#)
[Stream プロパティ \[353 ページ\]](#)
[UserName プロパティ \[357 ページ\]](#)
[Password プロパティ \[351 ページ\]](#)
[Version プロパティ \[357 ページ\]](#)
[LocalFileName プロパティ \[350 ページ\]](#)
[AuthenticationParms プロパティ \[347 ページ\]](#)
[ResumePartialDownload プロパティ \[353 ページ\]](#)
[AuthStatus プロパティ \[348 ページ\]](#)
[AuthValue プロパティ \[348 ページ\]](#)
[FileAuthCode プロパティ \[349 ページ\]](#)
[TransferredFile プロパティ \[356 ページ\]](#)
[StreamErrorCode プロパティ \[354 ページ\]](#)
[StreamErrorSystem プロパティ \[355 ページ\]](#)
[StreamErrorCode プロパティ \[354 ページ\]](#)

1.18.4 AuthenticationParms プロパティ

カスタムユーザ認証スクリプト (Mobile Link `authenticate_parameters` 接続イベント) のパラメータを指定します。

構文

Visual Basic

```
Public Property AuthenticationParms As String()
```

C#

```
public unsafe String[] AuthenticationParms {get;set;}
```

備考

それぞれに認証パラメータが格納された、文字列の配列 (配列のエントリが NULL であると、同期エラーになります)。デフォルト値は NULL 参照 (Visual Basic の Nothing) で、認証パラメータは指定されません。

最初の 255 文字列のみが使用されます。それぞれの文字列は認証パラメータの Mobile Link サーバの制限 (現在は 4000 UTF8 バイト) よりも長くはなりません。

1.18.5 AuthStatus プロパティ

前回行われたファイル転送の認証ステータスコードを返します。

構文

Visual Basic

```
Public ReadOnly Property AuthStatus As ULAuthStatusCode
```

C#

```
public ULAuthStatusCode AuthStatus {get;}
```

備考

前回行われたファイル転送の認証ステータスを示す ULAuthStatusCode 値の 1 つ。

関連情報

[ULAuthStatusCode 列挙体 \[584 ページ\]](#)

1.18.6 AuthValue プロパティ

カスタムユーザ認証同期スクリプトからの戻り値を返します。

構文

Visual Basic

```
Public ReadOnly Property AuthValue As Long
```

C#

```
public long AuthValue {get;}
```

備考

カスタムユーザ認証同期スクリプトから返された long 型整数。

1.18.7 FileAuthCode プロパティ

前回行われたファイル転送の authenticate_file_transfer スクリプトからの戻り値を返します。

構文

Visual Basic

```
Public ReadOnly Property FileAuthCode As UShort
```

C#

```
public ushort FileAuthCode {get;}
```

備考

前回行われたファイル転送の authenticate_file_transfer スクリプトから戻された unsigned short の整数。

1.18.8 FileName プロパティ

ダウンロードするファイルの名前を指定します。

構文

Visual Basic

```
Public Property FileName As String
```

C#

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

備考

Mobile Link サーバによって認識されたファイルの名前を指定する文字列。このプロパティにはデフォルト値がないので、明示的に設定してください。

FileName の値は、Mobile Link を実行するサーバ上にあるファイルの名前です。指定されたファイルは、まず UserName サブフォルダで、次にルートディレクトリで検索されます (ルートディレクトリは、Mobile Link サーバの -ftr オプションで指定)。Mobile Link サーバでファイルを検索できるようにするため、FileName の値にはドライブまたはパスの情報を含まないでください。たとえば、"myfile.txt" は有効ですが、"somedir¥myfile.txt"、"..¥myfile.txt"、"c:¥myfile.txt" は無効です。

関連情報

[LocalFileName プロパティ \[350 ページ\]](#)

[DownloadFile\(\) メソッド \[341 ページ\]](#)

[UploadFile\(\) メソッド \[345 ページ\]](#)

1.18.9 LocalFileName プロパティ

ダウンロードファイルのローカル名を指定します。

構文

Visual Basic

```
Public Property LocalFileName As String
```

C#

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

備考

ダウンロードファイルのローカル名を指定する文字列。値が null 参照 (Visual Basic の Nothing) の場合、FileName の値が使用されます。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

関連情報

[FileName プロパティ \[349 ページ\]](#)

1.18.10 LocalPath プロパティ

ファイルをダウンロードする場所を指定します。

構文

Visual Basic

```
Public Property LocalPath As String
```

C#

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

備考

ファイルのローカルディレクトリを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

デフォルトのローカルディレクトリは、デバイスのオペレーティングシステムによって異なります。

- Windows モバイルデバイスの場合、LocalPath の値は NULL 参照 (Visual Basic の Nothing) になり、ファイルはルート (¥) ディレクトリに格納されます。
- デスクトップアプリケーションの場合、LocalPath の値は NULL 参照 (Visual Basic の Nothing) になり、ファイルは現在のディレクトリに格納されます。

関連情報

[DownloadFile\(\) メソッド \[341 ページ\]](#)

[UploadFile\(\) メソッド \[345 ページ\]](#)

1.18.11 Password プロパティ

UserName 値で指定されたユーザの Mobile Link パスワードです。

構文

Visual Basic

```
Public Property Password As String
```

C#

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

備考

Mobile Link パスワードを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) で、パスワードは指定されません。

Mobile Link ユーザ名とパスワードは他のデータベースユーザ ID やパスワードとは別のもので、アプリケーションを Mobile Link サーバに対して識別し、認証するために使用されます。

関連情報

[UserName プロパティ \[357 ページ\]](#)

[DownloadFile\(\) メソッド \[341 ページ\]](#)

[UploadFile\(\) メソッド \[345 ページ\]](#)

1.18.12 RemoteKey プロパティ

Mobile Link サーバが Mobile Link クライアントをユニークに識別するキーです。

構文

Visual Basic

```
Public Property RemoteKey As String
```

C#

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

備考

リモートキーを指定する文字列。このプロパティにはデフォルト値がないので、明示的に設定してください。

Mobile Link サーバはこの値をさまざまなスクリプトに渡して、このクライアントをユニークに識別します。

関連情報

[DownloadFile\(\) メソッド \[341 ページ\]](#)

[UploadFile\(\) メソッド \[345 ページ\]](#)

1.18.13 ResumePartialDownload プロパティ

前の部分的なダウンロードを再開するか、破棄するかを指定します。

構文

Visual Basic

```
Public Property ResumePartialDownload As Boolean
```

C#

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

備考

前の部分的なダウンロードを再開する場合は true、破棄する場合は false。デフォルトは false です。

Ultra Light.NET では、ULFileTransferListener オブジェクトを使用して、通信エラーやユーザによるアボートが原因で失敗したダウンロードを再起動できます。Ultra Light.NET は、ダウンロードを受信しながら処理します。ダウンロードが中断した場合は、部分的なダウンロードファイルが保持されるため、次の転送中に再開できます。

ファイルがサーバ上で更新されている場合は、部分的なダウンロードは破棄され、新しくダウンロードが開始されます。

関連情報

[DownloadFile\(\) メソッド \[341 ページ\]](#)

[UploadFile\(\) メソッド \[345 ページ\]](#)

1.18.14 Stream プロパティ

ファイル転送に使用する Mobile Link 同期ストリームを指定します。

構文

Visual Basic

```
Public Property Stream As ULStreamType
```

C#

```
public ULStreamType Stream {get;set;}
```

備考

使用する同期ストリームのタイプを指定する `ULStreamType` 値の 1 つ。デフォルトは `ULStreamType.TCPIP` の値です。

ほとんどの同期ストリームでは、Mobile Link サーバのアドレスを識別したり、その他の動作を制御したりするパラメータが必要です。これらのパラメータは、`ULFileTransfer.StreamParms` の値によって指定されます。

ストリームタイプがそのプラットフォームに無効な値に設定されている場合、ストリームタイプは `ULStreamType.TCPIP` の値に設定されます。

関連情報

[ULStreamType 列挙体 \[592 ページ\]](#)

[StreamParms プロパティ \[355 ページ\]](#)

[DownloadFile\(\) メソッド \[341 ページ\]](#)

[UploadFile\(\) メソッド \[345 ページ\]](#)

1.18.15 StreamErrorCode プロパティ

前回行われたファイル転送のストリーム自体によってレポートされるエラーを返します。

構文

Visual Basic

```
Public ReadOnly Property StreamErrorCode As ULStreamErrorCode
```

C#

```
public ULStreamErrorCode StreamErrorCode {get;}
```

備考

ストリーム自体がレポートしたエラーを示す `ULStreamErrorCode` 値の 1 つです。エラーが発生しなければ、`ULStreamErrorCode.NONE` 値になります。

1.18.16 StreamErrorSystem プロパティ

ストリームエラーシステム固有のコードを返します。

構文

Visual Basic

```
Public ReadOnly Property StreamErrorSystem As Integer
```

C#

```
public int StreamErrorSystem {get;}
```

備考

ストリームエラーシステム固有のコードを示す整数。

1.18.17 StreamParms プロパティ

同期ストリームの設定パラメータを指定します。

構文

Visual Basic

```
Public Property StreamParms As String
```

C#

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

備考

キーワードと値の組み合わせがセミコロンで区切られたリスト形式の、ストリームのパラメータを指定する文字列。デフォルトは NULL 参照です。(Visual Basic の Nothing)

StreamParms 値は、同期ストリームに使用されるすべてのパラメータが含まれている文字列です。パラメータは、"名前=値" のペアをセミコロンで区切ったリスト ("param1=value1;param2=value2") で指定します。

関連情報

[Stream プロパティ \[353 ページ\]](#)

[DownloadFile\(\) メソッド \[341 ページ\]](#)

[UploadFile\(\) メソッド \[345 ページ\]](#)

[ULStreamType 列挙体 \[592 ページ\]](#)

1.18.18 TransferredFile プロパティ

前回行われたファイル転送時にファイルが実際にダウンロードされたかどうかを確認します。

構文

Visual Basic

```
Public ReadOnly Property TransferredFile As Boolean
```

C#

```
public bool TransferredFile {get;}
```

備考

ファイルがダウンロードされている場合は true、それ以外の場合は false。

DownloadFile または UploadFile のメソッドが呼び出された時点でファイルがすでに最新だった場合、結果は true になりますが、TransferredFile は false に設定されます。エラーが発生して DownloadFile または UploadFile のメソッドが false を返した場合、TransferredFile は false に設定されます。

関連情報

[DownloadFile\(\) メソッド \[341 ページ\]](#)

[UploadFile\(\) メソッド \[345 ページ\]](#)

1.18.19 UserName プロパティ

Mobile Link サーバが Mobile Link クライアントを識別するユーザ名です。

構文

Visual Basic

```
Public Property UserName As String
```

C#

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

備考

ユーザ名を指定する文字列。このプロパティにはデフォルト値がないので、明示的に設定してください。

Mobile Link サーバは、この値を使用して、ダウンロードするファイルを特定します。Mobile Link ユーザ名とパスワードは他のデータベースユーザ ID やパスワードとは別のもので、アプリケーションを Mobile Link サーバに対して識別し、認証するために使用されます。

関連情報

[Password プロパティ \[351 ページ\]](#)

[DownloadFile\(\) メソッド \[341 ページ\]](#)

[UploadFile\(\) メソッド \[345 ページ\]](#)

1.18.20 Version プロパティ

使用する同期スクリプトを指定します。

構文

Visual Basic

```
Public Property Version As String
```

C#

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

備考

使用する同期スクリプトのバージョンを指定する文字列。このプロパティにはデフォルト値がないので、明示的に設定してください。

統合データベースの同期スクリプトは、それぞれバージョン文字列でマーク付けされます。Ultra Light アプリケーションは、バージョン文字列により、同期スクリプトのセットから選択できます。

関連情報

[DownloadFile\(\) メソッド \[341 ページ\]](#)

[UploadFile\(\) メソッド \[345 ページ\]](#)

1.19 ULFileTransferProgressData クラス

UL 拡張: ファイル転送の進行状況のモニタリングデータを返します。

構文

Visual Basic

```
Public Class ULFileTransferProgressData
```

C#

```
public class ULFileTransferProgressData
```

メンバー

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

変数

変更子とタイプ	変数	説明
public const int	FLAG_IS_BLOCKING	Mobile Link サーバからの応答の待機中、ファイル転送はブロックされていることを示すフラグです。

プロパティ

変更子とタイプ	プロパティ	説明
public ulong	BytesReceived [359 ページ]	現在までに受信したバイト数を返します。
public ulong	FileSize [360 ページ]	転送されるファイルのサイズを返します。
public int	Flags [360 ページ]	現在の状態に関連する追加情報を示す、現在のファイル転送フラグを返します。
public ulong	ResumedAtSize [361 ページ]	転送が再開されるファイル内のポイントを返します。

このセクションの内容:

[BytesReceived プロパティ \[359 ページ\]](#)

現在までに受信したバイト数を返します。

[FileSize プロパティ \[360 ページ\]](#)

転送されるファイルのサイズを返します。

[Flags プロパティ \[360 ページ\]](#)

現在の状態に関連する追加情報を示す、現在のファイル転送フラグを返します。

[ResumedAtSize プロパティ \[361 ページ\]](#)

転送が再開されるファイル内のポイントを返します。

関連情報

[ULFileTransferProgressListener インタフェース \[361 ページ\]](#)

1.19.1 BytesReceived プロパティ

現在までに受信したバイト数を返します。

構文

Visual Basic

```
Public ReadOnly Property BytesReceived As ULong
```

C#

```
public ulong BytesReceived {get;}
```

備考

現在までに受信したバイト数。

1.19.2 FileSize プロパティ

転送されるファイルのサイズを返します。

構文

Visual Basic

```
Public ReadOnly Property FileSize As ULong
```

C#

```
public ulong FileSize {get;}
```

備考

ファイルのサイズ (バイト単位)。

1.19.3 Flags プロパティ

現在の状態に関連する追加情報を示す、現在のファイル転送フラグを返します。

構文

Visual Basic

```
Public ReadOnly Property Flags As Integer
```

C#

```
public int Flags {get;}
```

備考

フラグの組み合わせを保持する整数。

1.19.4 ResumedAtSize プロパティ

転送が再開されるファイル内のポイントを返します。

構文

Visual Basic

```
Public ReadOnly Property ResumedAtSize As ULong
```

C#

```
public ulong ResumedAtSize {get;}
```

備考

以前に転送されたバイト数。

1.20 ULFileTransferProgressListener インタフェース

UL 拡張: ファイル転送プログレスイベントを受信するリスナーインタフェースです。

構文

Visual Basic

```
Public Interface ULFileTransferProgressListener
```

C#

```
public interface ULFileTransferProgressListener
```

メンバー

ULFileTransferProgressListener 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 bool	FileTransferProgressed(ULFileTransferProgressData) [362 ページ]	ユーザに進行状況を通知するために、ファイル転送中に起動されます。

このセクションの内容:

[FileTransferProgressed\(ULFileTransferProgressData\) メソッド \[362 ページ\]](#)
ユーザに進行状況を通知するために、ファイル転送中に起動されます。

関連情報

[DownloadFile\(ULFileTransferProgressListener\) メソッド \[343 ページ\]](#)

1.20.1 FileTransferProgressed(ULFileTransferProgressData) メソッド

ユーザに進行状況を通知するために、ファイル転送中に起動されます。

構文

Visual Basic

```
Public Function FileTransferProgressed (ByVal data As  
ULFileTransferProgressData) As Boolean
```

C#

```
public bool FileTransferProgressed (ULFileTransferProgressData data)
```

パラメータ

data 最新の同期のプログレスデータを保持している ULFileTransferProgressData オブジェクト。

戻り値

このメソッドは、転送をキャンセルする場合は true を、続行する場合は false を返します。

備考

このメソッドは、転送をキャンセルする場合は true を、続行する場合は false を返します。

FileTransferProgressed の呼び出し中に、Ultra Light.NET API のメソッドを呼び出さないでください。

関連情報

[ULFileTransferProgressData クラス \[358 ページ\]](#)

1.21 ULIndexSchema クラス

UL 拡張: Ultra Light テーブルのインデックスのスキーマを表します。

構文

Visual Basic

```
Public NotInheritable Class ULIndexSchema
```

C#

```
public sealed class ULIndexSchema
```

メンバー

ULIndexSchema 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	Close() [365 ページ]	ULIndexSchema オブジェクトを閉じます。
public unsafe string	GetColumnName(short) [365 ページ]	このインデックス内の colOrdinalInIndex 番目のカラム名を返します。
public unsafe bool	IsColumnDescending(string) [366 ページ]	指定されたカラムが、インデックスによって降順で使用されるかどうかをチェックします。

プロパティ

変数とタイプ	プロパティ	説明
public unsafe short	ColumnCount [367 ページ]	インデックス内のカラム数を返します。
public unsafe bool	IsForeignKey [368 ページ]	インデックスが外部キーであるかどうかをチェックします。
public unsafe bool	IsForeignKeyCheckOnCommit [368 ページ]	外部キーの参照整合性のチェックが、コミット時に行われるか、挿入時と更新時に行われるかを確認します。
public unsafe bool	IsForeignKeyNullable [369 ページ]	外部キーが NULL 入力可であるかどうかをチェックします。

変更子とタイプ	プロパティ	説明
public bool	IsOpen [370 ページ]	インデックススキーマが開いているか、閉じているかを調べます。
public unsafe bool	IsPrimaryKey [370 ページ]	インデックスがプライマリキーであるかどうかをチェックします。
public unsafe bool	IsUniqueIndex [371 ページ]	インデックスがユニークであるかどうかをチェックします。
public unsafe bool	IsUniqueKey [371 ページ]	インデックスがユニークキーであるかどうかをチェックします。
public unsafe string	Name [372 ページ]	インデックスの名前を返します。
public unsafe string	ReferencedIndexName [372 ページ]	インデックスが外部キーである場合、参照されるプライマリインデックスの名前です。
public unsafe string	ReferencedTableName [373 ページ]	インデックスが外部キーである場合、参照されるプライマリテーブルの名前です。

備考

このクラスにはコンストラクタがありません。インデックスのスキーマは、`ULTableSchema.PrimaryKey` メソッド、`ULTableSchema.GetIndex(string)` メソッド、`ULTableSchema.GetOptimalIndex(int)` メソッドを使用して作成されます。

このセクションの内容:

[Close\(\) メソッド \[365 ページ\]](#)

ULIndexSchema オブジェクトを閉じます。

[GetColumnName\(short\) メソッド \[365 ページ\]](#)

このインデックス内の `colOrdinalInIndex` 番目のカラム名を返します。

[IsColumnDescending\(string\) メソッド \[366 ページ\]](#)

指定されたカラムが、インデックスによって降順で使用されるかどうかをチェックします。

[ColumnCount プロパティ \[367 ページ\]](#)

インデックス内のカラム数を返します。

[IsForeignKey プロパティ \[368 ページ\]](#)

インデックスが外部キーであるかどうかをチェックします。

[IsForeignKeyCheckOnCommit プロパティ \[368 ページ\]](#)

外部キーの参照整合性のチェックが、コミット時に行われるか、挿入時と更新時に行われるかを確認します。

[IsForeignKeyNullable プロパティ \[369 ページ\]](#)

外部キーが NULL 入力可であるかどうかをチェックします。

[IsOpen プロパティ \[370 ページ\]](#)

インデックススキーマが開いているか、閉じているかを調べます。

[IsPrimaryKey プロパティ \[370 ページ\]](#)

インデックスがプライマリキーであるかどうかをチェックします。

[IsUniqueIndex プロパティ \[371 ページ\]](#)

インデックスがユニークであるかどうかをチェックします。

[IsUniqueKey プロパティ \[371 ページ\]](#)

インデックスがユニークキーであるかどうかをチェックします。

[Name プロパティ \[372 ページ\]](#)

インデックスの名前を返します。

[ReferencedIndexName プロパティ \[372 ページ\]](#)

インデックスが外部キーである場合、参照されるプライマリインデックスの名前です。

[ReferencedTableName プロパティ \[373 ページ\]](#)

インデックスが外部キーである場合、参照されるプライマリテーブルの名前です。

関連情報

[PrimaryKey プロパティ \[574 ページ\]](#)

[GetIndex\(string\) メソッド \[561 ページ\]](#)

[GetOptimalIndex\(int\) メソッド \[563 ページ\]](#)

[ULTableSchema クラス \[555 ページ\]](#)

1.21.1 Close() メソッド

ULIndexSchema オブジェクトを閉じます。

構文

Visual Basic

```
Public Sub Close ()
```

C#

```
public void Close ()
```

1.21.2 GetColumnName(short) メソッド

このインデックス内の colOrdinalInIndex 番目のカラム名を返します。

構文

Visual Basic

```
Public Function GetColumnName (ByVal colOrdinalInIndex As Short) As String
```

C#

```
public unsafe string GetColumnName (short colOrdinalInIndex)
```

パラメータ

colOrdinalInIndex インデックス内の対象のカラムの順序。値は、[1,ColumnCount] の範囲内である必要があります。

戻り値

カラム名。

例外

ULException class SQL エラーが発生しました。

備考

カラムの順序とカウントは、スキーマのアップグレード中に変更されることがあります。インデックスからのカラムの順序は、特定のテーブルの同じ物理的カラムを参照する場合であっても、テーブルまたは別のインデックスのカラム ID とは異なります。

関連情報

[ColumnCount プロパティ \[367 ページ\]](#)

[ColumnCount プロパティ \[367 ページ\]](#)

1.21.3 IsColumnDescending(string) メソッド

指定されたカラムが、インデックスによって降順で使用されるかどうかをチェックします。

構文

Visual Basic

```
Public Function IsColumnDescending (ByVal name As String) As Boolean
```

C#

```
public unsafe bool IsColumnDescending (string name)
```

パラメータ

name カラム名。

戻り値

カラムが降順で使用される場合は true、昇順で使用される場合は false。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetColumnName\(short\) メソッド \[365 ページ\]](#)

[ColumnCount プロパティ \[367 ページ\]](#)

1.21.4 ColumnCount プロパティ

インデックス内のカラム数を返します。

構文

Visual Basic

```
Public ReadOnly Property ColumnCount As Short
```

C#

```
public unsafe short ColumnCount {get;}
```

備考

インデックス内のカラム数。

インデックス内のカラム序数の範囲は、1 ~ ColumnCount 値 (両端を含む) です。

カラムの順序とカウントは、スキーマのアップグレード中に変更されることがあります。インデックスからのカラムの順序は、特定のテーブルの同じ物理的カラムを参照する場合であっても、テーブルまたは別のインデックスのカラム ID とは異なります。

1.21.5 IsForeignKey プロパティ

インデックスが外部キーであるかどうかをチェックします。

構文

Visual Basic

```
Public ReadOnly Property IsForeignKey As Boolean
```

C#

```
public unsafe bool IsForeignKey {get;}
```

備考

インデックスが外部キーである場合は true、外部キーでない場合は false。

外部キー内のカラムは、別のテーブルの NULL 以外のユニークインデックスを参照することができます。

1.21.6 IsForeignKeyCheckOnCommit プロパティ

外部キーの参照整合性のチェックが、コミット時に行われるか、挿入時と更新時に行われるかを確認します。

構文

Visual Basic

```
Public ReadOnly Property IsForeignKeyCheckOnCommit As Boolean
```

C#

```
public unsafe bool IsForeignKeyCheckOnCommit {get;}
```


備考

参照整合性がコミット時にチェックされる場合は true、挿入時と更新時にチェックされる場合は false。

関連情報

[IsForeignKey プロパティ \[368 ページ\]](#)

1.21.7 IsForeignKeyNullable プロパティ

外部キーが NULL 入力可であるかどうかをチェックします。

構文

Visual Basic

```
Public ReadOnly Property IsForeignKeyNullable As Boolean
```

C#

```
public unsafe bool IsForeignKeyNullable {get;}
```

備考

外部キーが NULL 入力可の場合は true、NULL 入力不可の場合は false。

関連情報

[IsForeignKey プロパティ \[368 ページ\]](#)

1.21.8 IsOpen プロパティ

インデックススキーマが開いているか、閉じているかを調べます。

構文

Visual Basic

```
Public ReadOnly Property IsOpen As Boolean
```

C#

```
public bool IsOpen {get;}
```

備考

インデックススキーマが開いている場合は true、閉じている場合は false。

1.21.9 IsPrimaryKey プロパティ

インデックスがプライマリキーであるかどうかをチェックします。

構文

Visual Basic

```
Public ReadOnly Property IsPrimaryKey As Boolean
```

C#

```
public unsafe bool IsPrimaryKey {get;}
```

備考

インデックスがプライマリキーである場合は true、プライマリキーでない場合は false。

プライマリキー内のカラムでは NULL は許可されません。

1.21.10 IsUniqueIndex プロパティ

インデックスがユニークであるかどうかをチェックします。

構文

Visual Basic

```
Public ReadOnly Property IsUniqueIndex As Boolean
```

C#

```
public unsafe bool IsUniqueIndex {get;}
```

備考

インデックスがユニークである場合は true、ユニークでない場合は false。

ユニークなインデックス内のカラムは NULL であることがあります。

1.21.11 IsUniqueKey プロパティ

インデックスがユニークキーであるかどうかをチェックします。

構文

Visual Basic

```
Public ReadOnly Property IsUniqueKey As Boolean
```

C#

```
public unsafe bool IsUniqueKey {get;}
```

備考

インデックスがユニークキーである場合は true、ユニークキーでない場合は false。

ユニークキー内のカラムでは NULL は許可されません。

1.21.12 Name プロパティ

インデックスの名前を返します。

構文

Visual Basic

```
Public ReadOnly Property Name As String
```

C#

```
public unsafe string Name {get;}
```

備考

インデックスの名前を指定する文字列。

1.21.13 ReferencedIndexName プロパティ

インデックスが外部キーである場合、参照されるプライマリインデックスの名前です。

構文

Visual Basic

```
Public ReadOnly Property ReferencedIndexName As String
```

C#

```
public unsafe string ReferencedIndexName {get;}
```

備考

参照されるプライマリインデックスの名前を指定する文字列。

関連情報

[IsForeignKey プロパティ \[368 ページ\]](#)

1.21.14 ReferencedTableName プロパティ

インデックスが外部キーである場合、参照されるプライマリテーブルの名前です。

構文

Visual Basic

```
Public ReadOnly Property ReferencedTableName As String
```

C#

```
public unsafe string ReferencedTableName {get;}
```

備考

参照されるプライマリテーブルの名前を指定する文字列。

関連情報

[IsForeignKey プロパティ \[368 ページ\]](#)

1.22 ULInfoMessageEventArgs クラス

ULConnection.InfoMessage イベントのデータを提供します。

構文

Visual Basic

```
Public NotInheritable Class ULInfoMessageEventArgs Inherits System.EventArgs
```

C#

```
public sealed class ULInfoMessageEventArgs : System.EventArgs
```

メンバー

ULInfoMessageEventArgs 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 override string	ToString() [374 ページ]	ULConnection.InfoMessage イベントの文字列表現。

プロパティ

変更子とタイプ	プロパティ	説明
public string	Message [375 ページ]	データベースによって返される情報メッセージまたは警告メッセージの文字列です。
public ULSQLCode	NativeError [376 ページ]	データベースによって返される情報メッセージまたは警告に対応する SQLCODE です。
public string	Source [376 ページ]	メッセージを返す ADO.NET データプロバイダの名前です。

このセクションの内容:

[ToString\(\) メソッド \[374 ページ\]](#)

ULConnection.InfoMessage イベントの文字列表現。

[Message プロパティ \[375 ページ\]](#)

データベースによって返される情報メッセージまたは警告メッセージの文字列です。

[NativeError プロパティ \[376 ページ\]](#)

データベースによって返される情報メッセージまたは警告に対応する SQLCODE です。

[Source プロパティ \[376 ページ\]](#)

メッセージを返す ADO.NET データプロバイダの名前です。

関連情報

[InfoMessage イベント \[184 ページ\]](#)

1.22.1 ToString() メソッド

ULConnection.InfoMessage イベントの文字列表現。

構文

Visual Basic

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

C#

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

戻り値

情報メッセージまたは警告メッセージの文字列。

備考

ULConnection.InfoMessage イベントの文字列表現。

関連情報

[InfoMessage イベント \[184 ページ\]](#)

1.22.2 Message プロパティ

データベースによって返される情報メッセージまたは警告メッセージの文字列です。

構文

Visual Basic

```
Public ReadOnly Property Message As String
```

C#

```
public string Message {get;}
```

備考

情報メッセージまたは警告メッセージが含まれる文字列。

1.22.3 NativeError プロパティ

データベースによって返される情報メッセージまたは警告に対応する SQLCODE です。

構文

Visual Basic

```
Public ReadOnly Property NativeError As ULSQLCode
```

C#

```
public ULSQLCode NativeError {get;}
```

備考

情報メッセージまたは警告の ULSQLCode 値。

1.22.4 Source プロパティ

メッセージを返す ADO.NET データプロバイダの名前です。

構文

Visual Basic

```
Public ReadOnly Property Source As String
```

C#

```
public string Source {get;}
```

備考

文字列 "Ultra Light.NET Data Provider"。

1.23 ULMetaDataCollectionNames クラス

メタデータコレクションを取得する `ULConnection.GetSchema(String, String[])` メソッドで使用する定数のリストを提供します。

構文

Visual Basic

```
Public NotInheritable Class ULMetaDataCollectionNames
```

C#

```
public sealed class ULMetaDataCollectionNames
```

メンバー

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

プロパティ

変数とタイプ	プロパティ	説明
public string	Columns [379 ページ]	Columns コレクションを示す <code>ULConnection.GetSchema(String)</code> メソッドで使用する定数を提供します。
public string	DataSourceInformation [380 ページ]	DataSourceInformation コレクションを示す <code>ULConnection.GetSchema(String)</code> メソッドで使用する定数を提供します。
public string	DataTypes [381 ページ]	DataTypes コレクションを示す <code>ULConnection.GetSchema(String)</code> メソッドで使用する定数を提供します。
public string	ForeignKeys [382 ページ]	ForeignKeys コレクションを示す <code>ULConnection.GetSchema(String)</code> メソッドで使用する定数を提供します。
public string	IndexColumns [383 ページ]	IndexColumns コレクションを示す <code>ULConnection.GetSchema(String)</code> メソッドで使用する定数を提供します。
public string	Indexes [384 ページ]	Indexes コレクションを示す <code>ULConnection.GetSchema(String)</code> メソッドで使用する定数を提供します。
public string	MetaDataCollections [385 ページ]	MetaDataCollections コレクションを示す <code>ULConnection.GetSchema(String)</code> メソッドで使用する定数を提供します。

変更子とタイプ	プロパティ	説明
public string	Publications [386 ページ]	Publications コレクションを示す ULConnection.GetSchema(String) メソッドで使用する定数を提供します。
public string	ReservedWords [387 ページ]	ReservedWords コレクションを示す ULConnection.GetSchema(String) メソッドで使用する定数を提供します。
public string	Restrictions [388 ページ]	Restrictions コレクションを示す ULConnection.GetSchema(String) メソッドで使用する定数を提供します。
public string	Tables [389 ページ]	Tables コレクションを示す ULConnection.GetSchema(String) メソッドで使用する定数を提供します。

このセクションの内容:

[Columns プロパティ \[379 ページ\]](#)

Columns コレクションを示す ULConnection.GetSchema(String) メソッドで使用する定数を提供します。

[DataSourceInformation プロパティ \[380 ページ\]](#)

DataSourceInformation コレクションを示す ULConnection.GetSchema(String) メソッドで使用する定数を提供します。

[DataTypes プロパティ \[381 ページ\]](#)

DataTypes コレクションを示す ULConnection.GetSchema(String) メソッドで使用する定数を提供します。

[ForeignKeys プロパティ \[382 ページ\]](#)

ForeignKeys コレクションを示す ULConnection.GetSchema(String) メソッドで使用する定数を提供します。

[IndexColumns プロパティ \[383 ページ\]](#)

IndexColumns コレクションを示す ULConnection.GetSchema(String) メソッドで使用する定数を提供します。

[Indexes プロパティ \[384 ページ\]](#)

Indexes コレクションを示す ULConnection.GetSchema(String) メソッドで使用する定数を提供します。

[MetaDataCollections プロパティ \[385 ページ\]](#)

MetaDataCollections コレクションを示す ULConnection.GetSchema(String) メソッドで使用する定数を提供します。

[Publications プロパティ \[386 ページ\]](#)

Publications コレクションを示す ULConnection.GetSchema(String) メソッドで使用する定数を提供します。

[ReservedWords プロパティ \[387 ページ\]](#)

ReservedWords コレクションを示す ULConnection.GetSchema(String) メソッドで使用する定数を提供します。

[Restrictions プロパティ \[388 ページ\]](#)

Restrictions コレクションを示す ULConnection.GetSchema(String) メソッドで使用する定数を提供します。

[Tables プロパティ \[389 ページ\]](#)

Tables コレクションを示す ULConnection.GetSchema(String) メソッドで使用する定数を提供します。

1.23.1 Columns プロパティ

Columns コレクションを示す `ULConnection.GetSchema(String)` メソッドで使用する定数を提供します。

構文

Visual Basic

```
Public Shared ReadOnly Property Columns As String
```

C#

```
public string Columns {get;}
```

備考

Columns コレクションの名前を表す文字列。

例

次のコードは、Columns コレクションを使用して DataTable オブジェクトを設定します。

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.Columns )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.Columns );
```

関連情報

[GetSchema\(string\) メソッド \[159 ページ\]](#)

1.23.2 DataSourceInformation プロパティ

DataSourceInformation コレクションを示す `ULConnection.GetSchema(String)` メソッドで使用する定数を提供します。

構文

Visual Basic

```
Public Shared ReadOnly Property DataSourceInformation As String
```

C#

```
public string DataSourceInformation {get;}
```

備考

DataSourceInformation コレクションの名前を表す文字列。

例

次のコードは、DataSourceInformation コレクションを使用して DataTable オブジェクトを設定します。

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.DataSourceInformation )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.DataSourceInformation );
```

関連情報

[GetSchema\(string\) メソッド \[159 ページ\]](#)

1.23.3 DataTypes プロパティ

DataTypes コレクションを示す `ULConnection.GetSchema(String)` メソッドで使用する定数を提供します。

構文

Visual Basic

```
Public Shared ReadOnly Property DataTypes As String
```

C#

```
public string DataTypes {get;}
```

備考

DataTypes コレクションの名前を表す文字列。

例

次のコードは、DataTypes コレクションを使用して DataTable オブジェクトを設定します。

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.DataTypes )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.DataTypes );
```

関連情報

[GetSchema\(string\) メソッド \[159 ページ\]](#)

1.23.4 ForeignKeys プロパティ

ForeignKeys コレクションを示す `ULConnection.GetSchema(String)` メソッドで使用する定数を提供します。

構文

Visual Basic

```
Public Shared ReadOnly Property ForeignKeys As String
```

C#

```
public string ForeignKeys {get;}
```

備考

ForeignKeys コレクションの名前を表す文字列。

例

次のコードは、ForeignKeys コレクションを使用して DataTable オブジェクトを設定します。

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.ForeignKeys )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.ForeignKeys );
```

関連情報

[GetSchema\(string\) メソッド \[159 ページ\]](#)

1.23.5 IndexColumns プロパティ

IndexColumns コレクションを示す `ULConnection.GetSchema(String)` メソッドで使用する定数を提供します。

構文

Visual Basic

```
Public Shared ReadOnly Property IndexColumns As String
```

C#

```
public string IndexColumns {get;}
```

備考

IndexColumns コレクションの名前を表す文字列。

例

次のコードは、IndexColumns コレクションを使用して DataTable オブジェクトを設定します。

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.IndexColumns )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.IndexColumns );
```

関連情報

[GetSchema\(string\) メソッド \[159 ページ\]](#)

1.23.6 Indexes プロパティ

Indexes コレクションを示す `ULConnection.GetSchema(String)` メソッドで使用する定数を提供します。

構文

Visual Basic

```
Public Shared ReadOnly Property Indexes As String
```

C#

```
public string Indexes {get;}
```

備考

Indexes コレクションの名前を表す文字列。

例

次のコードは、Indexes コレクションを使用して DataTable オブジェクトを設定します。

```
' Visual Basic  
Dim schema As DataTable = _  
    conn.GetSchema( ULMetaDataCollectionNames.Indexes )
```

対応する C# 言語のコードを次に示します。

```
// C#  
DataTable schema =  
    conn.GetSchema( ULMetaDataCollectionNames.Indexes );
```

関連情報

[GetSchema\(string\) メソッド \[159 ページ\]](#)

1.23.7 MetaDataCollections プロパティ

MetaDataCollections コレクションを示す `ULConnection.GetSchema(String)` メソッドで使用する定数を提供します。

構文

Visual Basic

```
Public Shared ReadOnly Property MetaDataCollections As String
```

C#

```
public string MetaDataCollections {get;}
```

備考

MetaDataCollections コレクションの名前を表す文字列。

例

次のコードは、MetaDataCollections コレクションを使用して DataTable オブジェクトを設定します。

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.MetaDataCollections )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.MetaDataCollections );
```

関連情報

[GetSchema\(string\) メソッド \[159 ページ\]](#)

1.23.8 Publications プロパティ

Publications コレクションを示す `ULConnection.GetSchema(String)` メソッドで使用する定数を提供します。

構文

Visual Basic

```
Public Shared ReadOnly Property Publications As String
```

C#

```
public string Publications {get;}
```

備考

Publications コレクションの名前を示す文字列。

例

次のコードは、Publications コレクションを使用して DataTable オブジェクトを設定します。

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.Publications )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.Publications );
```

関連情報

[GetSchema\(string\) メソッド \[159 ページ\]](#)

1.23.9 ReservedWords プロパティ

ReservedWords コレクションを示す `ULConnection.GetSchema(String)` メソッドで使用する定数を提供します。

構文

Visual Basic

```
Public Shared ReadOnly Property ReservedWords As String
```

C#

```
public string ReservedWords {get;}
```

備考

ReservedWords コレクションの名前を表す文字列。

例

次のコードは、ReservedWords コレクションを使用して DataTable オブジェクトを設定します。

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.ReservedWords )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.ReservedWords );
```

関連情報

[GetSchema\(string\) メソッド \[159 ページ\]](#)

1.23.10 Restrictions プロパティ

Restrictions コレクションを示す `ULConnection.GetSchema(String)` メソッドで使用する定数を提供します。

構文

Visual Basic

```
Public Shared ReadOnly Property Restrictions As String
```

C#

```
public string Restrictions {get;}
```

備考

Restrictions コレクションの名前を表す文字列。

例

次のコードは、Restrictions コレクションを使用して DataTable オブジェクトを設定します。

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.Restrictions )
```

対応する C# 言語のコードを次に示します。

```
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.Restrictions );
```

関連情報

[GetSchema\(string\) メソッド \[159 ページ\]](#)

1.23.11 Tables プロパティ

Tables コレクションを示す `ULConnection.GetSchema(String)` メソッドで使用する定数を提供します。

構文

Visual Basic

```
Public Shared ReadOnly Property Tables As String
```

C#

```
public string Tables {get;}
```

備考

Tables コレクションの名前を表す文字列。

例

次のコードは、Tables コレクションを使用して DataTable オブジェクトを設定します。

```
' Visual Basic
Dim schema As DataTable = _
    conn.GetSchema( ULMetaDataCollectionNames.Tables )
// C#
DataTable schema =
    conn.GetSchema( ULMetaDataCollectionNames.Tables );
```

関連情報

[GetSchema\(string\) メソッド \[159 ページ\]](#)

1.24 ULParameter クラス

ULCommand オブジェクトのパラメータを表します。

構文

Visual Basic

```
Public NotInheritable Class ULParameter Inherits
System.Data.Common.DbParameter Implements System.ICloneable
```

C#

```
public sealed class ULParameter : System.Data.Common.DbParameter,  
System.ICloneable
```

メンバー

ULParameter 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	ULParameter [393 ページ]	値として NULL (Visual Basic の Nothing) を使用して、ULParameter オブジェクトを初期化します。

メソッド

変更子とタイプ	メソッド	説明
public override void	ResetDbType() [400 ページ]	このメソッドは Ultra Light.NET ではサポートされていません。
public override string	ToString() [401 ページ]	このインスタンスの文字列表現を返します。

プロパティ

変更子とタイプ	プロパティ	説明
public override DbType	DbType [401 ページ]	パラメータの System.Data.DbType を指定します。
public override ParameterDirection	Direction [402 ページ]	パラメータが入力専用、出力専用、双方向性、またはストアドプロシージャ戻り値パラメータのいずれであるかを示す値です。
public override bool	IsNullable [403 ページ]	パラメータが NULL 値を受け入れるかどうかを指定します。
public int	Offset [403 ページ]	ULParameter.Value のオフセットを指定します。
public override string	ParameterName [404 ページ]	パラメータの名前を指定します。
public byte	Precision [405 ページ]	ULParameter.Value プロパティを表すために使用される最大桁数を指定します。
public byte	Scale [405 ページ]	ULParameter.Value プロパティが解決される小数点の桁の数を指定します。
public override int	Size [406 ページ]	カラム内のデータの最大サイズを指定します。
public override string	SourceColumn [407 ページ]	DataSet オブジェクトにマッピングされ、値をロードしたり返したりするときに使用するソースカラムの名前を指定します。

変数とタイプ	プロパティ	説明
public override bool	SourceColumnNullMapping [407 ページ]	ソースカラムが NULL 入力可であるかどうかを指定します。
public override DataRowVersion	SourceVersion [408 ページ]	ULParameter.Value プロパティをロードするときに使用する System.Data.DataRowVersion 値。
public ULDbType	ULDbType [408 ページ]	パラメータの Sap.Data.UltraLite.ULDbType を指定します。
public override object	Value [409 ページ]	パラメータの値を指定します。

備考

ULParameter オブジェクトは、その多数のコンストラクタのいずれかを使用したり、ULCommand.CreateParameter メソッドを使用したりして直接作成できます。定数 0 および 0.0 の特別な処理と、オーバーロードされたメソッドの解決方法のため、ULParameter(string,object) コンストラクタを使用するときは、定数値をオブジェクトタイプに明示的にキャストすることをお奨めします。次に例を示します。

```
' Visual Basic
Dim p As ULParameter = New ULParameter( "", CType( 0, Object ) )
```

対応する C# 言語のコードを次に示します。

```
// C#
ULParameter p = new ULParameter( "", (object)0 );
```

パラメータ (ULCommand.CreateParameter メソッドによって作成されたものを含む) は、使用される ULCommand.Parameters コレクションに追加する必要があります。すべてのパラメータは、位置パラメータとして扱われ、コマンドによって追加された順序で使用されます。

Ultra Light.NET では、パラメータは IN パラメータとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value プロパティだけです。

このセクションの内容:

[ULParameter コンストラクタ \[393 ページ\]](#)

値として NULL (Visual Basic の Nothing) を使用して、ULParameter オブジェクトを初期化します。

[ResetDbType\(\) メソッド \[400 ページ\]](#)

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

[ToString\(\) メソッド \[401 ページ\]](#)

このインスタンスの文字列表現を返します。

[DbType プロパティ \[401 ページ\]](#)

パラメータの System.Data.DbType を指定します。

[Direction プロパティ \[402 ページ\]](#)

パラメータが入力専用、出力専用、双方向、またはストアードプロシージャの戻り値パラメータのいずれであるかを示す値を表します。

[IsNullable プロパティ \[403 ページ\]](#)

パラメータが NULL 値を受け入れるかどうかを指定します。

[Offset プロパティ \[403 ページ\]](#)

ULParameter.Value のオフセットを指定します。

[ParameterName プロパティ \[404 ページ\]](#)

パラメータの名前を指定します。

[Precision プロパティ \[405 ページ\]](#)

ULParameter.Value プロパティを表すために使用される最大桁数を指定します。

[Scale プロパティ \[405 ページ\]](#)

ULParameter.Value プロパティが解決される小数点の桁の数を指定します。

[Size プロパティ \[406 ページ\]](#)

カラム内のデータの最大サイズを指定します。

[SourceColumn プロパティ \[407 ページ\]](#)

DataSet オブジェクトにマッピングされ、値をロードしたり返したりするときに使用するソースカラムの名前を指定します。

[SourceColumnNullMapping プロパティ \[407 ページ\]](#)

ソースカラムが NULL 入力可であるかどうかを指定します。

[SourceVersion プロパティ \[408 ページ\]](#)

ULParameter.Value プロパティをロードするときに使用する System.Data.DataRowVersion 値。

[ULDbType プロパティ \[408 ページ\]](#)

パラメータの Sap.Data.UltraLite.ULDbType を指定します。

[Value プロパティ \[409 ページ\]](#)

パラメータの値を指定します。

関連情報

[ULCommand クラス \[58 ページ\]](#)

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

[ULParameter\(string, object\) コンストラクタ \[399 ページ\]](#)

[Parameters プロパティ \[103 ページ\]](#)

[Value プロパティ \[409 ページ\]](#)

1.24.1 ULParameter コンストラクタ

値として NULL (Visual Basic の Nothing) を使用して、ULParameter オブジェクトを初期化します。

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	ULParameter() [394 ページ]	値として NULL (Visual Basic の Nothing) を使用して、ULParameter オブジェクトを初期化します。
public	ULParameter(string, ULDbType) [394 ページ]	ULParameter オブジェクトを、指定されたパラメータ名とデータ型で初期化します。
public	ULParameter(string, ULDbType, int) [395 ページ]	ULParameter オブジェクトを、指定されたパラメータ名とデータ型で初期化します。
public	ULParameter(string, ULDbType, int, ParameterDirection, bool, byte, byte, string, DataRowVersion, object) [397 ページ]	指定されたパラメータ名、データ型、長さ、方向、NULL 入力属性、数値精度、数値の位取り、ソースカラム、ソースのバージョン、値で、ULParameter オブジェクトを初期化します。
public	ULParameter(string, ULDbType, int, string) [398 ページ]	ULParameter オブジェクトを、指定されたパラメータ名、データ型、長さで初期化します。
public	ULParameter(string, object) [399 ページ]	ULParameter オブジェクトを、指定されたパラメータ名と値で初期化します。

このセクションの内容:

[ULParameter\(\) コンストラクタ \[394 ページ\]](#)

値として NULL (Visual Basic の Nothing) を使用して、ULParameter オブジェクトを初期化します。

[ULParameter\(string, ULDbType\) コンストラクタ \[394 ページ\]](#)

ULParameter オブジェクトを、指定されたパラメータ名とデータ型で初期化します。

[ULParameter\(string, ULDbType, int\) コンストラクタ \[395 ページ\]](#)

ULParameter オブジェクトを、指定されたパラメータ名とデータ型で初期化します。

[ULParameter\(string, ULDbType, int, ParameterDirection, bool, byte, byte, string, DataRowVersion, object\) コンストラクタ \[397 ページ\]](#)

指定されたパラメータ名、データ型、長さ、方向、NULL 入力属性、数値精度、数値の位取り、ソースカラム、ソースのバージョン、値で、ULParameter オブジェクトを初期化します。

[ULParameter\(string, ULDbType, int, string\) コンストラクタ \[398 ページ\]](#)

ULParameter オブジェクトを、指定されたパラメータ名、データ型、長さで初期化します。

[ULParameter\(string, object\) コンストラクタ \[399 ページ\]](#)

ULParameter オブジェクトを、指定されたパラメータ名と値で初期化します。

1.24.1.1 ULParameter() コンストラクタ

値として NULL (Visual Basic の Nothing) を使用して、ULParameter オブジェクトを初期化します。

構文

Visual Basic

```
Public Sub ULParameter ()
```

C#

```
public ULParameter ()
```

例

次のコードでは、値が 3 である ULParameter パラメータが作成され、cmd という ULCommand オブジェクトに追加されます。

```
' Visual Basic
Dim p As ULParameter = New ULParameter
p.Value = 3
cmd.Parameters.Add( p )
```

対応する C# 言語のコードを次に示します。

```
// C#
ULParameter p = new ULParameter();
p.Value = 3;
cmd.Parameters.Add( p );
```

関連情報

[Value プロパティ \[409 ページ\]](#)

[ULParameter\(string, object\) コンストラクタ \[399 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

1.24.1.2 ULParameter(string, ULDbType) コンストラクタ

ULParameter オブジェクトを、指定されたパラメータ名とデータ型で初期化します。

構文

Visual Basic

```
Public Sub ULParameter (
    ByVal parameterName As String,
```

```
ByVal dbType As ULDbType  
)
```

C#

```
public ULParameter (  
    string parameterName,  
    ULDbType dbType  
)
```

パラメータ

parameterName パラメータの名前。名前のないパラメータの場合は、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand オブジェクトでパラメータの名前が使用されません。

dbType Sap.Data.UltraLite.ULDbType 値の 1 つ。

備考

このコンストラクタの使用はお奨めしません。これは、他のデータプロバイダとの互換性のために用意されています。

Ultra Light.NET では、パラメータは IN パラメータとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なものは、ULParameter.Value プロパティだけです。

関連情報

[ULParameter\(\) コンストラクタ \[394 ページ\]](#)

[ULParameter\(string, object\) コンストラクタ \[399 ページ\]](#)

[Value プロパティ \[409 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

1.24.1.3 ULParameter(string, ULDbType, int) コンストラクタ

ULParameter オブジェクトを、指定されたパラメータ名とデータ型で初期化します。

構文

Visual Basic

```
Public Sub ULParameter (  
    ByVal parameterName As String,  
    ByVal dbType As ULDbType,
```

```
ByVal size As Integer  
)
```

C#

```
public ULParameter (  
    string parameterName,  
    ULDbType dbType,  
    int size  
)
```

パラメータ

parameterName パラメータの名前。名前のないパラメータの場合は、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand オブジェクトでパラメータの名前が使用されません。

dbType Sap.Data.UltraLite.ULDbType 値の 1 つ。

size パラメータの長さ。

備考

このコンストラクタの使用はお奨めしません。これは、他のデータプロバイダとの互換性のために用意されています。

Ultra Light.NET では、パラメータは IN パラメータとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なものは、ULParameter.Value プロパティだけです。

関連情報

[ULParameter\(\) コンストラクタ \[394 ページ\]](#)

[ULParameter\(string, object\) コンストラクタ \[399 ページ\]](#)

[Value プロパティ \[409 ページ\]](#)

1.24.1.4 ULParameter(string, ULDbType, int, ParameterDirection, bool, byte, byte, string, DataRowVersion, object) コンストラクタ

指定されたパラメータ名、データ型、長さ、方向、NULL 入力属性、数値精度、数値の位取り、ソースカラム、ソースのバージョン、値で、ULParameter オブジェクトを初期化します。

構文

Visual Basic

```
Public Sub ULParameter (  
    ByVal parameterName As String,  
    ByVal dbType As ULDbType,  
    ByVal size As Integer,  
    ByVal direction As ParameterDirection,  
    ByVal isNullable As Boolean,  
    ByVal precision As Byte,  
    ByVal scale As Byte,  
    ByVal sourceColumn As String,  
    ByVal sourceVersion As DataRowVersion,  
    ByVal value As Object  
)
```

C#

```
public ULParameter (  
    string parameterName,  
    ULDbType dbType,  
    int size,  
    ParameterDirection direction,  
    bool isNullable,  
    byte precision,  
    byte scale,  
    string sourceColumn,  
    DataRowVersion sourceVersion,  
    object value  
)
```

パラメータ

parameterName パラメータの名前。名前のないパラメータの場合は、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand オブジェクトでパラメータの名前が使用されません。

dbType Sap.Data.UltraLite.ULDbType 値の 1 つ。

size パラメータの長さ。

direction System.Data.ParameterDirection 値の 1 つ。

isNullable フィールドの値を NULL にできる場合は true、できない場合は false。

precision Value プロパティが解決される小数点の左右の桁の合計数。

scale Value プロパティが解決される小数点までの桁の合計数。

sourceColumn マッピング対象のソースカラムの名前。

sourceVersion System.Data.DataRowVersion 値の 1 つ。
value パラメータ値を生成する System.Object を渡します。

例外

ULException class Ultra Light.NET では、System.Data.ParameterDirection.Input 方向しかサポートされていません。

備考

このコンストラクタの使用はお奨めしません。これは、他のデータプロバイダとの互換性のために用意されています。

関連情報

[ULParameter\(\) コンストラクタ \[394 ページ\]](#)

[ULParameter\(string, object\) コンストラクタ \[399 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

1.24.1.5 ULParameter(string, ULDbType, int, string) コンストラクタ

ULParameter オブジェクトを、指定されたパラメータ名、データ型、長さで初期化します。

構文

Visual Basic

```
Public Sub ULParameter (  
    ByVal parameterName As String,  
    ByVal dbType As ULDbType,  
    ByVal size As Integer,  
    ByVal sourceColumn As String  
)
```

C#

```
public ULParameter (  
    string parameterName,  
    ULDbType dbType,  
    int size,  
    string sourceColumn  
)
```

パラメータ

parameterName パラメータの名前。名前のないパラメータの場合は、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand オブジェクトでパラメータの名前が使用されません。

dbType Sap.Data.UltraLite.ULDbType 値の 1 つ。

size パラメータの長さ。

sourceColumn マッピング対象のソースカラムの名前。

備考

このコンストラクタの使用はお奨めしません。これは、他のデータプロバイダとの互換性のために用意されています。

Ultra Light.NET では、パラメータは IN パラメータとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なものは、ULParameter.Value プロパティだけです。

関連情報

[ULParameter\(\) コンストラクタ \[394 ページ\]](#)

[ULParameter\(string, object\) コンストラクタ \[399 ページ\]](#)

[Value プロパティ \[409 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

1.24.1.6 ULParameter(string, object) コンストラクタ

ULParameter オブジェクトを、指定されたパラメータ名と値で初期化します。

構文

Visual Basic

```
Public Sub ULParameter (  
    ByVal parameterName As String,  
    ByVal value As Object  
)
```

C#

```
public ULParameter (  
    string parameterName,  
    object value  
)
```

パラメータ

parameterName パラメータの名前。名前のないパラメータの場合は、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand オブジェクトでパラメータの名前が使用されません。

value パラメータ値を生成する System.Object クラスを渡します。

備考

定数 0 と 0.0 の特別な処理と、オーバーロードされたメソッドの解決方法のため、このコンストラクタを使用するときは、定数値をオブジェクトタイプに明示的にキャストすることをお奨めします。

例

次のコードでは、値が 0 である ULParameter パラメータが作成され、cmd という ULCommand オブジェクトに追加されます。

```
' Visual Basic
cmd.Parameters.Add( New ULParameter( "", CType( 0, Object ) ) )
```

対応する C# 言語のコードを次に示します。

```
// C#
cmd.Parameters.Add( new ULParameter( "", (object)0 ) );
```

関連情報

[ULParameter\(\) コンストラクタ \[394 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

1.24.2 ResetDbType() メソッド

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

構文

Visual Basic

```
Public Overrides Sub ResetDbType ()
```

C#

```
public override void ResetDbType ()
```


備考

Ultra Light.NET では、パラメータは IN パラメータとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value プロパティだけです。

関連情報

[Value プロパティ \[409 ページ\]](#)

1.24.3 ToString() メソッド

このインスタンスの文字列表現を返します。

構文

Visual Basic

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

C#

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

戻り値

パラメータの名前。

1.24.4 DbType プロパティ

パラメータの System.Data.DbType を指定します。

構文

Visual Basic

```
Public Overrides Property DbType As DbType
```

C#

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

備考

System.Data.DbType 値の 1 つ。

ULParameter.ULDbType と DbType プロパティはリンクされます。このため DbType プロパティを設定すると、ULParameter.ULDbType プロパティは、サポートされている Sap.Data.UltraLite.ULDbType 値に変更されます。

関連情報

[ULDbType プロパティ \[408 ページ\]](#)

1.24.5 Direction プロパティ

パラメータが入力専用、出力専用、双方向、またはストアードプロシージャの戻り値パラメータのいずれであるかを示す値を表します。

構文

Visual Basic

```
Public Overrides Property Direction As ParameterDirection
```

C#

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

備考

System.Data.ParameterDirection 値の 1 つ。

Ultra Light.NET では、パラメータは IN パラメータとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なものは、ULParameter.Value プロパティだけです。

関連情報

[Value プロパティ \[409 ページ\]](#)

1.24.6 IsNullable プロパティ

パラメータが NULL 値を受け入れるかどうかを指定します。

構文

Visual Basic

```
Public Overrides Property IsNullable As Boolean
```

C#

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

備考

NULL 値を受け入れる場合は true、受け入れない場合は false。デフォルトは false です。NULL 値は DBNull クラスを使用して処理されます。

Ultra Light.NET では、パラメータは IN パラメータとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value プロパティだけです。

関連情報

[Value プロパティ \[409 ページ\]](#)

1.24.7 Offset プロパティ

ULParameter.Value のオフセットを指定します。

構文

Visual Basic

```
Public Property Offset As Integer
```

C#

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

備考

値のオフセット。デフォルトは 0 です。

Ultra Light.NET では、パラメータは IN パラメータとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なものは、ULParameter.Value プロパティだけです。

関連情報

[Value プロパティ \[409 ページ\]](#)

1.24.8 ParameterName プロパティ

パラメータの名前を指定します。

構文

Visual Basic

```
Public Overrides Property ParameterName As String
```

C#

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

備考

パラメータの名前を表す文字列。名前のないパラメータの場合は空の文字列 ("")。NULL 参照 (Visual Basic の Nothing) を指定すると、空の文字列が使用されます。

Ultra Light.NET では、ULCommand オブジェクトはパラメータの名前を使用しません。すべてのパラメータは、位置パラメータとして扱われ、コマンドによって追加された順序で使用されます。

関連情報

[ULCommand クラス \[58 ページ\]](#)

1.24.9 Precision プロパティ

ULParameter.Value プロパティを表すために使用される最大桁数を指定します。

構文

Visual Basic

```
Public Property Precision As Byte
```

C#

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

備考

ULParameter.Value プロパティを示すために使用される最大桁数。デフォルト値は 0 です。これは、データプロバイダが ULParameter.Value プロパティの精度を設定することを示します。

Ultra Light.NET では、パラメータは IN パラメータとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value プロパティだけです。

関連情報

[Value プロパティ \[409 ページ\]](#)

1.24.10 Scale プロパティ

ULParameter.Value プロパティが解決される小数点の桁の数を指定します。

構文

Visual Basic

```
Public Property Scale As Byte
```

C#

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

備考

ULParameter.Value プロパティが解決される小数点の桁の数。デフォルトは 0 です。

Ultra Light.NET では、パラメータは IN パラメータとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なものは、ULParameter.Value プロパティだけです。

関連情報

[Value プロパティ \[409 ページ\]](#)

1.24.11 Size プロパティ

カラム内のデータの最大サイズを指定します。

構文

Visual Basic

```
Public Overrides Property Size As Integer
```

C#

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

備考

カラム内のデータの最大サイズ。デフォルト値はパラメータ値から推測されます。Size プロパティは、バイナリと文字列型に対して使用されます。

Ultra Light.NET では、パラメータは IN パラメータとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なものは、ULParameter.Value プロパティだけです。

関連情報

[Value プロパティ \[409 ページ\]](#)

1.24.12 SourceColumn プロパティ

DataSet オブジェクトにマッピングされ、値をロードしたり返したりするときに使用するソースカラムの名前を指定します。

構文

Visual Basic

```
Public Overrides Property SourceColumn As String
```

C#

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

備考

DataSet オブジェクトにマッピングされ、値をロードしたり返したりするときに使用するソースカラムの名前を指定する文字列。

Ultra Light.NET では、パラメータは IN パラメータとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なものは、ULParameter.Value プロパティだけです。

関連情報

[Value プロパティ \[409 ページ\]](#)

1.24.13 SourceColumnNullMapping プロパティ

ソースカラムが NULL 入力可であるかどうかを指定します。

構文

Visual Basic

```
Public Overrides Property SourceColumnNullMapping As Boolean
```

C#

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

備考

ソースカラムが NULL 入力可の場合は true、そうでない場合は false。

Ultra Light.NET では、パラメータは IN パラメータとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value プロパティだけです。

関連情報

[Value プロパティ \[409 ページ\]](#)

1.24.14 SourceVersion プロパティ

ULParameter.Value プロパティをロードするときに使用する System.Data.DataRowVersion 値。

構文

Visual Basic

```
Public Overrides Property SourceVersion As DataRowVersion
```

C#

```
public override DataRowVersion SourceVersion {get;set;}
```

関連情報

[Value プロパティ \[409 ページ\]](#)

1.24.15 ULDbType プロパティ

パラメータの Sap.Data.UltraLite.ULDbType を指定します。

構文

Visual Basic

```
Public Property ULDbType As ULDbType
```

C#

```
public ULDbType ULDbType {get;set;}
```


備考

Sap.Data.UltraLite.ULDbType 値の 1 つ。

ULDbType プロパティと ULParameter.DbType プロパティはリンクされます。このため ULDbType プロパティを設定すると、ULParameter.DbType プロパティは、サポートされている System.Data.DbType 値に変更されます。

Ultra Light.NET では、パラメータは IN パラメータとしてのみ使用できるほか、マッピング情報はすべて無視されます。重要なのは、ULParameter.Value プロパティだけです。

関連情報

[Value プロパティ \[409 ページ\]](#)

[DbType プロパティ \[401 ページ\]](#)

1.24.16 Value プロパティ

パラメータの値を指定します。

構文

Visual Basic

```
Public Overrides Property Value As Object
```

C#

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

備考

パラメータ値を指定する System.Object クラスを渡します。

値は、型変換やマッピングは行われずに、データプロバイダにそのまま送信されます。コマンドが実行されると、コマンドは必要な型に値を変換しますが、値を変換できない場合は ULSQLCode.SQLE_CONVERSION_ERROR とともに ULException オブジェクトを通知します。

関連情報

[ULException クラス \[328 ページ\]](#)

1.25 ULParameterCollection クラス

ULCommand オブジェクトのすべてのパラメータを示します。

構文

Visual Basic

```
Public NotInheritable Class ULParameterCollection Inherits  
System.Data.Common.DbParameterCollection
```

C#

```
public sealed class ULParameterCollection :  
System.Data.Common.DbParameterCollection
```

メンバー

ULParameterCollection 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 ULParameter	Add [413 ページ]	ULParameter オブジェクトをコレクションに追加します。
public override void	AddRange [421 ページ]	ULParameterCollection の末尾に値の配列を追加します。
public override void	Clear() [423 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 override bool	Contains [423 ページ]	ULParameter オブジェクトがコレクション内に存在するかどうかをチェックします。
public override void	CopyTo(Array, int) [425 ページ]	ULParameter オブジェクトを ULParameterCollection から指定された配列にコピーします。
public override IEnumerator	GetEnumerator() [426 ページ]	コレクションの列挙子を返します。
protected override DbParameter	GetParameter [427 ページ]	個々のトピックを参照してください。
public override int	IndexOf [428 ページ]	コレクション内の ULParameter オブジェクトのロケーションを返します。
public override void	Insert(int, object) [430 ページ]	コレクション内の指定されたインデックス位置に ULParameter オブジェクトを挿入します。
public override void	Remove(object) [431 ページ]	ULParameter オブジェクトをコレクションから削除します。

変更子とタイプ	メソッド	説明
public override void	RemoveAt [432 ページ]	コレクションの指定されたインデックス位置のパラメータを削除します。
protected override void	SetParameter [434 ページ]	個々のトピックを参照してください。

プロパティ

変更子とタイプ	プロパティ	説明
public override int	Count [435 ページ]	コレクション内の ULParameter オブジェクトの数を返します。
public override bool	IsFixedSize [435 ページ]	ULParameterCollection オブジェクトのサイズが固定かどうかを示します。
public override bool	IsReadOnly [436 ページ]	ULParameterCollection オブジェクトが読み込み専用かどうかを示します。
public override bool	IsSynchronized [436 ページ]	ULParameterCollection オブジェクトが同期されるかどうかを示します。
public override object	SyncRoot [437 ページ]	SAPParameterCollection オブジェクトへのアクセスを同期するために使用できるオブジェクトを返します。
public new ULParameter	this [437 ページ]	指定されたインデックス位置にある ULParameter オブジェクトを返します。

備考

コレクション内のすべてのパラメータは、位置パラメータとして扱われ、ULCommand.CommandText 値の疑問符のプレースホルダと同じ順序で指定されます。たとえば、コレクション内の最初のパラメータは SQL 文の最初の疑問符に対応し、コレクション内の 2 番目のパラメータは SQL 文の 2 番目の疑問符に対応します。ULCommand.CommandText 値内の疑問符の数は、少なくともコレクション内のパラメータの数と同じでなければなりません。パラメータが不足している場合は、NULL が代用されます。

ULParameterCollection クラスにはコンストラクタがありません。ULParameterCollection オブジェクトは、ULCommand.Parameters プロパティから取得します。

このセクションの内容:

[Add メソッド \[413 ページ\]](#)

ULParameter オブジェクトをコレクションに追加します。

[AddRange メソッド \[421 ページ\]](#)

ULParameterCollection の末尾に値の配列を追加します。

[Clear\(\) メソッド \[423 ページ\]](#)

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 \[423 ページ\]](#)

ULParameter オブジェクトがコレクション内に存在するかどうかをチェックします。

[CopyTo\(Array, int\) メソッド \[425 ページ\]](#)

ULParameter オブジェクトを ULParameterCollection から指定された配列にコピーします。

[GetEnumerator\(\) メソッド \[426 ページ\]](#)

コレクションの列挙子を返します。

[GetParameter メソッド \[427 ページ\]](#)

[IndexOf メソッド \[428 ページ\]](#)

コレクション内の ULParameter オブジェクトのロケーションを返します。

[Insert\(int, object\) メソッド \[430 ページ\]](#)

コレクション内の指定されたインデックス位置に ULParameter オブジェクトを挿入します。

[Remove\(object\) メソッド \[431 ページ\]](#)

ULParameter オブジェクトをコレクションから削除します。

[RemoveAt メソッド \[432 ページ\]](#)

コレクションの指定されたインデックス位置のパラメータを削除します。

[SetParameter メソッド \[434 ページ\]](#)

[Count プロパティ \[435 ページ\]](#)

コレクション内の ULParameter オブジェクトの数を返します。

[IsFixedSize プロパティ \[435 ページ\]](#)

ULParameterCollection オブジェクトのサイズが固定かどうかを示します。

[IsReadOnly プロパティ \[436 ページ\]](#)

ULParameterCollection オブジェクトが読み込み専用かどうかを示します。

[IsSynchronized プロパティ \[436 ページ\]](#)

ULParameterCollection オブジェクトが同期されるかどうかを示します。

[SyncRoot プロパティ \[437 ページ\]](#)

SAPParameterCollection オブジェクトへのアクセスを同期するために使用できるオブジェクトを返します。

[this プロパティ \[437 ページ\]](#)

指定されたインデックス位置にある ULParameter オブジェクトを返します。

関連情報

[ULCommand クラス \[58 ページ\]](#)

[CommandText プロパティ \[98 ページ\]](#)

[Parameters プロパティ \[103 ページ\]](#)

1.25.1 Add メソッド

ULParameter オブジェクトをコレクションに追加します。

オーバーロードリスト

変数とタイプ	オーバーロード名	説明
public ULParameter	Add(ULParameter) [414 ページ]	ULParameter オブジェクトをコレクションに追加します。
public override int	Add(object) [415 ページ]	ULParameter オブジェクトをコレクションに追加します。
public ULParameter	Add(string, ULDbType) [416 ページ]	指定されたパラメータ名とデータ型を使用して作成された新しい ULParameter オブジェクトをコレクションに追加します。
public ULParameter	Add(string, ULDbType, int) [417 ページ]	指定されたパラメータ名、データ型、長さを使用して作成された新しい ULParameter オブジェクトをコレクションに追加します。
public ULParameter	Add(string, ULDbType, int, string) [418 ページ]	指定されたパラメータ名、データ型、長さ、ソースカラム名を使用して作成された新しい ULParameter オブジェクトをコレクションに追加します。
public ULParameter	Add(string, object) [420 ページ]	指定されたパラメータ名と値を使用して作成された新しい ULParameter オブジェクトをコレクションに追加します。

このセクションの内容:

[Add\(ULParameter\) メソッド \[414 ページ\]](#)

ULParameter オブジェクトをコレクションに追加します。

[Add\(object\) メソッド \[415 ページ\]](#)

ULParameter オブジェクトをコレクションに追加します。

[Add\(string, ULDbType\) メソッド \[416 ページ\]](#)

指定されたパラメータ名とデータ型を使用して作成された新しい ULParameter オブジェクトをコレクションに追加します。

[Add\(string, ULDbType, int\) メソッド \[417 ページ\]](#)

指定されたパラメータ名、データ型、長さを使用して作成された新しい ULParameter オブジェクトをコレクションに追加します。

[Add\(string, ULDbType, int, string\) メソッド \[418 ページ\]](#)

指定されたパラメータ名、データ型、長さ、ソースカラム名を使用して作成された新しい ULParameter オブジェクトをコレクションに追加します。

[Add\(string, object\) メソッド \[420 ページ\]](#)

指定されたパラメータ名と値を使用して作成された新しい ULParameter オブジェクトをコレクションに追加します。

1.25.1.1 Add(ULParameter) メソッド

ULParameter オブジェクトをコレクションに追加します。

構文

Visual Basic

```
Public Function Add (ByVal value As ULParameter) As ULParameter
```

C#

```
public ULParameter Add (ULParameter value)
```

パラメータ

value コレクションに追加される ULParameter オブジェクト。

戻り値

新しい ULParameter オブジェクト。

例外

ArgumentNullException 値を null (Visual Basic の Nothing) にすることはできません。

ArgumentException ULParameter オブジェクトは、一度しかコレクションに追加できません。

備考

コレクション内のすべてのパラメータは、位置パラメータとして扱われ、ULCommand.CommandText 値内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。たとえば、コレクション内の最初のパラメータは SQL 文の最初の疑問符に対応し、コレクション内の 2 番目のパラメータは SQL 文の 2 番目の疑問符に対応します。ULCommand.CommandText 値内の疑問符の数は、少なくともコレクション内のパラメータの数と同じでなければなりません。パラメータが不足している場合は、NULL が代用されます。

関連情報

[Add\(string, object\) メソッド \[420 ページ\]](#)

[ULParameter クラス \[389 ページ\]](#)

[CommandText プロパティ \[98 ページ\]](#)

1.25.1.2 Add(object) メソッド

ULParameter オブジェクトをコレクションに追加します。

構文

Visual Basic

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

C#

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

パラメータ

value コレクションに追加される ULParameter オブジェクト。

戻り値

新しい ULParameter オブジェクトのインデックス。

例外

ArgumentNullException 値を null (Visual Basic の Nothing) にすることはできません。

InvalidCastException 指定される値は ULParameter オブジェクトである必要があります。

ArgumentException ULParameter オブジェクトは、一度しかコレクションに追加できません。

備考

コレクション内のすべてのパラメータは、位置パラメータとして扱われ、ULCommand.CommandText 値内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。たとえば、コレクション内の最初のパラメータは

SQL 文の最初の疑問符に対応し、コレクション内の 2 番目のパラメータは SQL 文の 2 番目の疑問符に対応します。ULCommand.CommandText 値内の疑問符の数は、少なくともコレクション内のパラメータの数と同じでなければなりません。パラメータが不足している場合は、NULL が代用されます。

関連情報

[Add\(ULParameter\) メソッド \[414 ページ\]](#)

[Add\(string, object\) メソッド \[420 ページ\]](#)

[CommandText プロパティ \[98 ページ\]](#)

[ULParameter クラス \[389 ページ\]](#)

1.25.1.3 Add(string, ULDbType) メソッド

指定されたパラメータ名とデータ型を使用して作成された新しい ULParameter オブジェクトをコレクションに追加します。

構文

Visual Basic

```
Public Function Add (  
    ByVal parameterName As String,  
    ByVal ulDbType As ULDbType  
) As ULParameter
```

C#

```
public ULParameter Add (  
    string parameterName,  
    ULDbType ulDbType  
)
```

パラメータ

parameterName パラメータの名前。名前のないパラメータの場合は、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand オブジェクトでパラメータの名前が使用されません。

ulDbType Sap.Data.UltraLite.ULDbType 値の 1 つ。

戻り値

新しい ULParameter オブジェクト。

備考

コレクション内のすべてのパラメータは、位置パラメータとして扱われ、ULCommand.CommandText 値内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。たとえば、コレクション内の最初のパラメータは SQL 文の最初の疑問符に対応し、コレクション内の 2 番目のパラメータは SQL 文の 2 番目の疑問符に対応します。ULCommand.CommandText 値内の疑問符の数は、少なくともコレクション内のパラメータの数と同じでなければなりません。パラメータが不足している場合は、NULL が代用されます。

関連情報

[Add\(ULParameter\) メソッド \[414 ページ\]](#)

[Add\(string, object\) メソッド \[420 ページ\]](#)

[ULParameter クラス \[389 ページ\]](#)

[CommandText プロパティ \[98 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

1.25.1.4 Add(string, ULDbType, int) メソッド

指定されたパラメータ名、データ型、長さを使用して作成された新しい ULParameter オブジェクトをコレクションに追加します。

構文

Visual Basic

```
Public Function Add (  
    ByVal parameterName As String,  
    ByVal ulDbType As ULDbType,  
    ByVal size As Integer  
) As ULParameter
```

C#

```
public ULParameter Add (  
    string parameterName,  
    ULDbType ulDbType,  
    int size  
)
```

パラメータ

parameterName パラメータの名前。名前のないパラメータの場合は、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand オブジェクトでパラメータの名前が使用されません。

ulDbType Sap.Data.UltraLite.ULDbType 値の 1 つ。
size パラメータの長さ。

戻り値

新しい ULParameter オブジェクト。

備考

コレクション内のすべてのパラメータは、位置パラメータとして扱われ、ULCommand.CommandText 値内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。たとえば、コレクション内の最初のパラメータは SQL 文の最初の疑問符に対応し、コレクション内の 2 番目のパラメータは SQL 文の 2 番目の疑問符に対応します。ULCommand.CommandText 値内の疑問符の数は、少なくともコレクション内のパラメータの数と同じでなければなりません。パラメータが不足している場合は、NULL が代用されます。

関連情報

[Add\(ULParameter\) メソッド \[414 ページ\]](#)

[Add\(string, object\) メソッド \[420 ページ\]](#)

[ULParameter クラス \[389 ページ\]](#)

[CommandText プロパティ \[98 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

1.25.1.5 Add(string, ULDbType, int, string) メソッド

指定されたパラメータ名、データ型、長さ、ソースカラム名を使用して作成された新しい ULParameter オブジェクトをコレクションに追加します。

構文

Visual Basic

```
Public Function Add (  
    ByVal parameterName As String,  
    ByVal ulDbType As ULDbType,  
    ByVal size As Integer,  
    ByVal sourceColumn As String  
) As ULParameter
```

C#

```
public ULParameter Add (
```

```
string parameterName,  
ULDbType ulDbType,  
int size,  
string sourceColumn  
)
```

パラメータ

parameterName パラメータの名前。名前のないパラメータの場合は、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand オブジェクトでパラメータの名前が使用されません。

ulDbType Sap.Data.UltraLite.ULDbType 値の 1 つ。

size パラメータの長さ。

sourceColumn マッピング対象のソースカラムの名前。

戻り値

新しい ULParameter オブジェクト。

備考

コレクション内のすべてのパラメータは、位置パラメータとして扱われ、ULCommand.CommandText 値内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。たとえば、コレクション内の最初のパラメータは SQL 文の最初の疑問符に対応し、コレクション内の 2 番目のパラメータは SQL 文の 2 番目の疑問符に対応します。

ULCommand.CommandText 値内の疑問符の数は、少なくともコレクション内のパラメータの数と同じでなければなりません。パラメータが不足している場合は、NULL が代用されます。

関連情報

[Add\(ULParameter\) メソッド \[414 ページ\]](#)

[Add\(string, object\) メソッド \[420 ページ\]](#)

[ULParameter クラス \[389 ページ\]](#)

[CommandText プロパティ \[98 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

1.25.1.6 Add(string, object) メソッド

指定されたパラメータ名と値を使用して作成された新しい ULParameter オブジェクトをコレクションに追加します。

構文

Visual Basic

```
Public Function Add (  
    ByVal parameterName As String,  
    ByVal value As Object  
) As ULParameter
```

C#

```
public ULParameter Add (  
    string parameterName,  
    object value  
)
```

パラメータ

parameterName パラメータの名前。名前のないパラメータの場合は、この値に空の文字列 ("") または NULL 参照 (Visual Basic の Nothing) を使用します。Ultra Light.NET では、ULCommand オブジェクトでパラメータの名前が使用されません。

value パラメータの値である System.Object。

戻り値

新しい ULParameter オブジェクト。

備考

コレクション内のすべてのパラメータは、位置パラメータとして扱われ、ULCommand.CommandText 値内の対応する疑問符のプレースホルダと同じ順序でコレクションに追加される必要があります。たとえば、コレクション内の最初のパラメータは SQL 文の最初の疑問符に対応し、コレクション内の 2 番目のパラメータは SQL 文の 2 番目の疑問符に対応します。ULCommand.CommandText 値内の疑問符の数は、少なくともコレクション内のパラメータの数と同じでなければなりません。パラメータが不足している場合は、NULL が代用されます。

定数 0 と 0.0 の特別な処理と、オーバーロードされたメソッドの解決方法のため、このメソッドを使用するときは、定数値をオブジェクトタイプに明示的にキャストすることをお奨めします。

例

次のコードでは、0 の ULParameter 値が、cmd という ULCommand オブジェクトに追加されます。

```
' Visual Basic
cmd.Parameters.Add( "", CType( 0, Object ) )
```

対応する C# 言語のコードを次に示します。

```
// C#
cmd.Parameters.Add( "", (object)0 );
```

関連情報

[Add\(ULParameter\) メソッド \[414 ページ\]](#)

[ULParameter クラス \[389 ページ\]](#)

[CommandText プロパティ \[98 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

1.25.2 AddRange メソッド

ULParameterCollection の末尾に値の配列を追加します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public override void	AddRange(Array) [422 ページ]	ULParameterCollection の末尾に値の配列を追加します。
public void	AddRange(ULParameter[]) [422 ページ]	ULParameterCollection の末尾に値の配列を追加します。

このセクションの内容:

[AddRange\(Array\) メソッド \[422 ページ\]](#)

ULParameterCollection の末尾に値の配列を追加します。

[AddRange\(ULParameter\[\]\) メソッド \[422 ページ\]](#)

ULParameterCollection の末尾に値の配列を追加します。

1.25.2.1 AddRange(Array) メソッド

ULParameterCollection の末尾に値の配列を追加します。

構文

Visual Basic

```
Public Overrides Sub AddRange (ByVal values As Array)
```

C#

```
public override void AddRange (Array values)
```

パラメータ

values このコレクションの末尾に追加する ULParameter オブジェクトの配列。

関連情報

[ULParameter クラス \[389 ページ\]](#)

1.25.2.2 AddRange(ULParameter[]) メソッド

ULParameterCollection の末尾に値の配列を追加します。

構文

Visual Basic

```
Public Sub AddRange (ByVal values As ULParameter())
```

C#

```
public void AddRange (ULParameter[] values)
```

パラメータ

values このコレクションの末尾に追加する ULParameter オブジェクトの配列。

備考

これは、DbParameterCollection.AddRange(Array) メソッドが厳密に型指定されたものです。

関連情報

[ULParameter クラス \[389 ページ\]](#)

[ULParameterCollection クラス \[410 ページ\]](#)

1.25.3 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 Overrides Sub Clear ()
```

C#

```
public override void Clear ()
```

1.25.4 Contains メソッド

ULParameter オブジェクトがコレクション内に存在するかどうかをチェックします。

オーバーロードリスト

変数とタイプ	オーバーロード名	説明
public override bool	Contains(object) [424 ページ]	ULParameter オブジェクトがコレクション内に存在するかどうかをチェックします。
public override bool	Contains(string) [425 ページ]	指定された名前の ULParameter オブジェクトがコレクション内に存在するかどうかをチェックします。

このセクションの内容:

[Contains\(object\) メソッド \[424 ページ\]](#)

ULParameter オブジェクトがコレクション内に存在するかどうかをチェックします。

[Contains\(string\) メソッド \[425 ページ\]](#)

指定された名前の ULParameter オブジェクトがコレクション内に存在するかどうかをチェックします。

1.25.4.1 Contains(object) メソッド

ULParameter オブジェクトがコレクション内に存在するかどうかをチェックします。

構文

Visual Basic

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

C#

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

パラメータ

value チェック対象の ULParameter オブジェクト。

戻り値

コレクションに、その ULParameter オブジェクトがある場合は true、ない場合は false。

関連情報

[Contains\(string\) メソッド \[425 ページ\]](#)

[ULParameter クラス \[389 ページ\]](#)

1.25.4.2 Contains(string) メソッド

指定された名前の ULParameter オブジェクトがコレクション内に存在するかどうかをチェックします。

構文

Visual Basic

```
Public Overrides Function Contains (ByVal value As String) As Boolean
```

C#

```
public override bool Contains (string value)
```

パラメータ

value 検索対象のパラメータの名前。

戻り値

コレクションに、その ULParameter オブジェクトがある場合は true、ない場合は false。

関連情報

[Contains\(object\) メソッド \[424 ページ\]](#)

[ULParameter クラス \[389 ページ\]](#)

1.25.5 CopyTo(Array, int) メソッド

ULParameter オブジェクトを ULParameterCollection から指定された配列にコピーします。

構文

Visual Basic

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

C#

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

パラメータ

array ULParameter オブジェクトのコピー先の配列。

index 配列の開始インデックス。

関連情報

[ULParameter クラス \[389 ページ\]](#)

1.25.6 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 Overrides Function GetEnumerator () As System.Collections.IEnumerator
```

C#

```
public override IEnumerator GetEnumerator ()
```

戻り値

コレクション内のパラメータを列挙する ArrayList 列挙子。

1.25.7 GetParameter メソッド

オーバードリスト

変更子とタイプ	オーバーロード名	説明
protected override DbParameter	GetParameter(int) [427 ページ]	
protected override DbParameter	GetParameter(string) [427 ページ]	

このセクションの内容:

[GetParameter\(int\) メソッド \[427 ページ\]](#)

[GetParameter\(string\) メソッド \[427 ページ\]](#)

1.25.7.1 GetParameter(int) メソッド

構文

Visual Basic

```
Protected Overrides Function GetParameter (ByVal index As Integer) As DbParameter
```

C#

```
protected override DbParameter GetParameter (int index)
```

1.25.7.2 GetParameter(string) メソッド

構文

Visual Basic

```
Protected Overrides Function GetParameter (ByVal parameterName As String) As DbParameter
```

C#

```
protected override DbParameter GetParameter (string parameterName)
```

1.25.8 IndexOf メソッド

コレクション内の ULParameter オブジェクトのロケーションを返します。

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 override int	IndexOf(object) [428 ページ]	コレクション内の ULParameter オブジェクトのロケーションを返します。
public override int	IndexOf(string) [429 ページ]	指定された名前を持つ ULParameter オブジェクトのコレクション内のロケーションを返します。

このセクションの内容:

[IndexOf\(object\) メソッド \[428 ページ\]](#)

コレクション内の ULParameter オブジェクトのロケーションを返します。

[IndexOf\(string\) メソッド \[429 ページ\]](#)

指定された名前を持つ ULParameter オブジェクトのコレクション内のロケーションを返します。

1.25.8.1 IndexOf(object) メソッド

コレクション内の ULParameter オブジェクトのロケーションを返します。

構文

Visual Basic

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

C#

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

パラメータ

value 検索する ULParameter オブジェクト。

戻り値

コレクション内の ULParameter オブジェクトの 0 から始まるインデックス。パラメータが見つからない場合は -1。

例外

InvalidCastException 指定される値は ULParameter オブジェクトである必要があります。

関連情報

[IndexOf\(string\) メソッド \[429 ページ\]](#)

[ULParameter クラス \[389 ページ\]](#)

1.25.8.2 IndexOf(string) メソッド

指定された名前を持つ ULParameter オブジェクトのコレクション内のロケーションを返します。

構文

Visual Basic

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

C#

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

パラメータ

parameterName 検索するパラメータの名前。

戻り値

コレクション内の ULParameter オブジェクトの 0 から始まるインデックス。パラメータが見つからない場合は -1。

関連情報

[IndexOf\(object\) メソッド \[428 ページ\]](#)

[ULParameter クラス \[389 ページ\]](#)

1.25.9 Insert(int, object) メソッド

コレクション内の指定されたインデックス位置に ULParameter オブジェクトを挿入します。

構文

Visual Basic

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

C#

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

パラメータ

index パラメータの挿入先となる、コレクション内の 0 から始まるインデックス。

value 挿入する ULParameter オブジェクト。

例外

IndexOutOfRangeException インデックスは無効です。

ArgumentNullException null 参照 (Visual Basic の Nothing) を使用してパラメータを設定することはできません。

InvalidCastException 指定される値は ULParameter オブジェクトである必要があります。

関連情報

[ULParameter クラス \[389 ページ\]](#)

1.25.10 Remove(object) メソッド

ULParameter オブジェクトをコレクションから削除します。

構文

Visual Basic

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

C#

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

パラメータ

value 削除する ULParameter オブジェクト。

例外

ArgumentNullException null 参照 (Visual Basic の Nothing) を使用してパラメータを設定することはできません。

InvalidCastException 指定される値は ULParameter オブジェクトである必要があります。

ArgumentException コレクション内に、指定されたパラメータがありません。

関連情報

[ULParameter クラス \[389 ページ\]](#)

1.25.11 RemoveAt メソッド

コレクションの指定されたインデックス位置のパラメータを削除します。

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 override void	RemoveAt(int) [432 ページ]	コレクションの指定されたインデックス位置のパラメータを削除します。
public override void	RemoveAt(string) [433 ページ]	指定された名前のパラメータをコレクションから削除します。

このセクションの内容:

[RemoveAt\(int\) メソッド \[432 ページ\]](#)

コレクションの指定されたインデックス位置のパラメータを削除します。

[RemoveAt\(string\) メソッド \[433 ページ\]](#)

指定された名前のパラメータをコレクションから削除します。

1.25.11.1 RemoveAt(int) メソッド

コレクションの指定されたインデックス位置のパラメータを削除します。

構文

Visual Basic

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

C#

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

パラメータ

index 削除するパラメータの 0 から始まるインデックス。値は、`[0,ULParameterCollection.Count-1]` の範囲内である必要があります。コレクションの最初のパラメータのインデックス値は 0 です。

例外

IndexOutOfRangeException インデックスは無効です。

関連情報

[RemoveAt\(string\) メソッド \[433 ページ\]](#)

[Count プロパティ \[435 ページ\]](#)

1.25.11.2 RemoveAt(string) メソッド

指定された名前のパラメータをコレクションから削除します。

構文

Visual Basic

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

C#

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

パラメータ

parameterName 取得するパラメータの名前。

例外

IndexOutOfRangeException 指定した名前のパラメータがありません。

関連情報

[RemoveAt\(int\) メソッド \[432 ページ\]](#)

1.25.12 SetProperty メソッド

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
protected override void	SetProperty(int, DbParameter) [434 ページ]	
protected override void	SetProperty(string, DbParameter) [434 ページ]	

このセクションの内容:

[SetProperty\(int, DbParameter\) メソッド \[434 ページ\]](#)

[SetProperty\(string, DbParameter\) メソッド \[434 ページ\]](#)

1.25.12.1 SetProperty(int, DbParameter) メソッド

構文

Visual Basic

```
Protected Overrides Sub SetProperty (  
    ByVal index As Integer,  
    ByVal parm As DbParameter  
)
```

C#

```
protected override void SetProperty (  
    int index,  
    DbParameter parm  
)
```

1.25.12.2 SetProperty(string, DbParameter) メソッド

構文

Visual Basic

```
Protected Overrides Sub SetProperty (  
    ByVal index As Integer,  
    ByVal parm As DbParameter  
)
```

```
ByVal parameterName As String,  
ByVal parm As DbParameter  
)
```

C#

```
protected override void SetParameter (  
    string parameterName,  
    DbParameter parm  
)
```

1.25.13 Count プロパティ

コレクション内の ULParameter オブジェクトの数を返します。

構文

Visual Basic

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

C#

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

備考

コレクション内の ULParameter オブジェクトの数。

1.25.14 IsFixedSize プロパティ

ULParameterCollection オブジェクトのサイズが固定かどうかを示します。

構文

Visual Basic

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

C#

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

備考

コレクションのサイズが固定の場合は true、そうでない場合は false。

1.25.15 IsReadOnly プロパティ

ULParameterCollection オブジェクトが読み込み専用かどうかを示します。

構文

Visual Basic

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

C#

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

備考

コレクションが読み込み専用の場合は true、そうでない場合は false。

1.25.16 IsSynchronized プロパティ

ULParameterCollection オブジェクトが同期されるかどうかを示します。

構文

Visual Basic

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

C#

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

備考

コレクションが同期している場合は true、そうでない場合は false。

1.25.17 SyncRoot プロパティ

SAPParameterCollection オブジェクトへのアクセスを同期するために使用できるオブジェクトを返します。

構文

Visual Basic

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

C#

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

備考

このコレクションへのアクセスの同期に使用されるオブジェクト。

1.25.18 this プロパティ

指定されたインデックス位置にある ULParameter オブジェクトを返します。

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 new ULParameter	this[int index] [438 ページ]	指定されたインデックス位置にある ULParameter オブジェクトを返します。
public new ULParameter	this[string parameterName] [438 ページ]	指定された名前の ULParameter オブジェクトを返します。

このセクションの内容:

[this\[int index\] プロパティ \[438 ページ\]](#)

指定されたインデックス位置にある ULParameter オブジェクトを返します。

[this\[string parameterName\] プロパティ \[438 ページ\]](#)

指定された名前の ULParameter オブジェクトを返します。

1.25.18.1 this[int index] プロパティ

指定されたインデックス位置にある ULParameter オブジェクトを返します。

構文

Visual Basic

```
Public Shadows Property Item (ByVal indexAs Integer) As ULParameter
```

C#

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

戻り値

指定されたインデックス位置の ULParameter オブジェクト。

備考

C# では、このプロパティは ULParameterCollection クラスのインデクサです。

これは、DbParameterCollection.this[int] プロパティが厳密に型指定されたものです。

関連情報

[ULParameter クラス \[389 ページ\]](#)

1.25.18.2 this[string parameterName] プロパティ

指定された名前の ULParameter オブジェクトを返します。

構文

Visual Basic

```
Public Shadows Property Item (ByVal parameterNameAs String) As ULParameter
```

C#

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

戻り値

指定された名前の ULParameter オブジェクト。

備考

C# では、このプロパティは ULParameterCollection クラスのインデクサです。

これは、DbParameterCollection.this[string] プロパティが厳密に型指定されたものです。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetValue\(int\) メソッド \[311 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[ULParameter クラス \[389 ページ\]](#)

1.26 ULResultSet クラス

UL 拡張: Ultra Light データベースの編集可能な結果セットを表します。

構文

Visual Basic

```
Public Class ULResultSet Inherits ULDataReader
```

C#

```
public class ULResultSet : ULDataReader
```

メンバー

ULResultSet 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 unsafe void	AppendBytes(int, byte[], int, int) [446 ページ]	指定された System.Bytes 配列の指定されたサブセットを、指定された ULDbType.LongBinary カラムの新しい値に追加します。
public unsafe void	AppendChars(int, char[], int, int) [447 ページ]	指定された System.Chars 配列の指定されたサブセットを、指定された ULDbType.LongVarchar カラムの新しい値に追加します。
public void	Delete() [449 ページ]	現在の行を削除します。
public void	SetBoolean(int, bool) [449 ページ]	指定されたカラムの値を、System.Boolean を使用して設定します。
public void	SetByte(int, byte) [450 ページ]	指定されたカラムの値を、System.Byte (符号なし 8 ビット整数) を使用して設定します。
public unsafe void	SetBytes(int, byte[]) [451 ページ]	指定されたカラムの値を、System.Bytes を使用して設定します。
public unsafe void	SetDateTime(int, DateTime) [453 ページ]	指定されたカラムの値を、System.DateTime を使用して設定します。
public void	SetDBNull(int) [454 ページ]	カラムを NULL に設定します。
public void	SetDecimal(int, decimal) [455 ページ]	指定されたカラムの値を、System.Decimal を使用して設定します。
public void	SetDouble(int, double) [456 ページ]	指定されたカラムの値を、System.Double を使用して設定します。
public void	SetFloat(int, float) [457 ページ]	指定されたカラムの値を、System.Single を使用して設定します。
public unsafe void	SetGuid(int, Guid) [458 ページ]	指定されたカラムの値を、System.Guid を使用して設定します。
public void	SetInt16(int, short) [459 ページ]	指定されたカラムの値を、System.Int16 を使用して設定します。
public void	SetInt32(int, int) [461 ページ]	指定されたカラムの値を、System.Int32 を使用して設定します。
public void	SetInt64(int, long) [462 ページ]	指定されたカラムの値を、Int64 を使用して設定します。
public unsafe void	SetString(int, string) [463 ページ]	指定されたカラムの値を、System.String を使用して設定します。
public unsafe void	SetTimeSpan(int, TimeSpan) [464 ページ]	指定されたカラムの値を、System.TimeSpan を使用して設定します。
public void	SetToDefault(int) [465 ページ]	指定されたカラムの値を、そのデフォルト値に設定します。
public void	SetUInt16(int, ushort) [466 ページ]	指定されたカラムの値を、System.UInt16 を使用して設定します。

変数とタイプ	メソッド	説明
public void	SetUInt32(int, uint) [467 ページ]	指定されたカラムの値を、System.UInt32 を使用して設定します。
public void	SetUInt64(int, ulong) [468 ページ]	指定されたカラムの値を、System.UInt64 を使用して設定します。
public void	Update() [470 ページ]	現在のカラム値 (set メソッドを使用して指定されます) で新しいローを更新します。
public void	UpdateBegin() [470 ページ]	現在のローを更新する準備を行います。

ULDataReader から継承されたメンバー

変数とタイプ	メンバー	説明
public override void	Close() [282 ページ]	カーソルを閉じます。
public override int	Depth [321 ページ]	現在のローのネストの深さを返します。
protected override void	Dispose(bool) [282 ページ]	
public override int	FieldCount [321 ページ]	このカーソル内のカラム数を返します。
public override unsafe bool	GetBoolean(int) [283 ページ]	指定されたカラムの値を System.Boolean として返します。
public override unsafe byte	GetByte(int) [284 ページ]	指定されたカラムの値を符号なし 8 ビット値 (System.Byte) として返します。
public unsafe byte[]	GetBytes(int) [285 ページ]	UL 拡張: 指定されたカラムの値を System.Bytes の値の配列として返します。
public override unsafe long	GetBytes(int, long, byte[], int, int) [286 ページ]	指定されたオフセットで始まる、指定された ULDbType.LongBinary カラムの値のサブセットを、コピー先の System.Byte 配列の指定されたオフセットにコピーします。
public override char	GetChar(int) [288 ページ]	このメソッドは Ultra Light.NET ではサポートされていません。
public override unsafe long	GetChars(int, long, char[], int, int) [289 ページ]	指定されたオフセットで始まる、指定された ULDbType.LongVarchar カラムの値のサブセットを、コピー先の System.Char 配列の指定されたオフセットにコピーします。
public override string	GetDataTypeName(int) [291 ページ]	指定されたカラムのプロバイダのデータ型の名前を返します。
public override unsafe DateTime	GetDateTime(int) [292 ページ]	指定されたカラムの値を、ミリ秒の精度の System.DateTime 型として返します。
protected override DbDataReader	GetDbDataReader(int) [293 ページ]	
public override decimal	GetDecimal(int) [293 ページ]	指定されたカラムの値を System.Decimal 型として返します。
public override unsafe double	GetDouble(int) [294 ページ]	指定されたカラムの値を System.Double 型として返します。

変更子とタイプ	メンバー	説明
public override IEnumerator	GetEnumerator() [295 ページ]	ULDataReader オブジェクトの反復処理を実行する System.Collections.IEnumerator 値を返します。
public override Type	GetFieldType(int) [295 ページ]	指定されたカラムに最適な System.Type 値を返します。
public override unsafe float	GetFloat(int) [296 ページ]	指定されたカラムの値を System.Single 型として返します。
public override unsafe Guid	GetGuid(int) [297 ページ]	指定されたカラムの値を UUID (System.Guid) 型として返します。
public override unsafe short	GetInt16(int) [298 ページ]	指定されたカラムの値を System.Int16 型として返します。
public override unsafe int	GetInt32(int) [299 ページ]	指定されたカラムの値を System.Int32 型として返します。
public override unsafe long	GetInt64(int) [300 ページ]	指定されたカラムの値を System.Int64 型として返します。
public override string	GetName(int) [301 ページ]	指定されたカラムの名前を返します。
public override unsafe int	GetOrdinal(string) [302 ページ]	指定されたカラムのカラム ID を返します。
public unsafe int	GetRowCount(int) [303 ページ]	UL 拡張: カーソル内のローの数を、スレッシュホールド以内で返します。
public override DataTable	GetSchemaTable() [304 ページ]	ULDataReader オブジェクトのカラムのメタデータが記述された System.Data.DataTable 値を返します。
public override unsafe String	GetString(int) [306 ページ]	指定されたカラムの値を System.String 型として返します。
public unsafe TimeSpan	GetTimeSpan(int) [307 ページ]	指定されたカラムの値を、ミリ秒の精度の System.TimeSpan 型として返します。
public unsafe ushort	GetUInt16(int) [308 ページ]	指定されたカラムの値を System.UInt16 型として返します。
public unsafe uint	GetUInt32(int) [309 ページ]	指定されたカラムの値を System.UInt32 型として返します。
public unsafe ulong	GetUInt64(int) [310 ページ]	指定されたカラムの値を System.UInt64 型として返します。
public override object	GetValue(int) [311 ページ]	指定されたカラムの値をネイティブフォーマットで返します。
public override int	GetValues(object[]) [312 ページ]	現在のローのすべてのカラム値を返します。
public override unsafe bool	HasRows [322 ページ]	ULDataReader オブジェクトに 1 つまたは複数のローがあるかどうかをチェックします。
public unsafe bool	IsBOF [322 ページ]	UL 拡張: 現在のローの位置が最初のローの前かどうかをチェックします。

変更子とタイプ	メンバー	説明
public override bool	IsClosed [323 ページ]	カーソルが現在開いているかどうかを確認します。
public override unsafe bool	IsDBNull(int) [313 ページ]	指定されたカラムの値が NULL かどうかをチェックします。
public unsafe bool	IsEOF [323 ページ]	UL 拡張: 現在のローの位置が最後のローの後かどうかをチェックします。
public void	MoveAfterLast() [314 ページ]	UL 拡張: カーソルの最後のローの後に、カーソルを配置します。
public void	MoveBeforeFirst() [314 ページ]	UL 拡張: カーソルの最初のローの前に、カーソルを配置します。
public unsafe bool	MoveFirst() [315 ページ]	UL 拡張: カーソルの最初のローに、カーソルを配置します。
public unsafe bool	MoveLast() [315 ページ]	UL 拡張: カーソルの最後のローに、カーソルを配置します。
public unsafe bool	MoveNext() [316 ページ]	UL 拡張: カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。
public unsafe bool	MovePrevious() [317 ページ]	UL 拡張: カーソルを前のローに配置するか、最初のローの前に配置します。
public unsafe bool	MoveRelative(int) [317 ページ]	UL 拡張: 現在のローを基準としてカーソルを配置します。
public override bool	NextResult() [318 ページ]	バッチ SQL 文の結果を読み込むときに ULDataReader オブジェクトを次の結果に進めます。
public override bool	Read() [319 ページ]	カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。
public override int	RecordsAffected [324 ページ]	SQL 文の実行によって変更、挿入、または削除されたローの数を返します。
public int	RowCount [324 ページ]	UL 拡張: カーソル内のローの数を返します。
public ULCursorSchema	Schema [325 ページ]	UL 拡張: このカーソルのスキーマを保持します。
public override object	this[int colID] [326 ページ]	指定されたカラムの値をネイティブフォーマットで返します。
public override object	this[string name] [327 ページ]	指定された名前のカラムの値をネイティブフォーマットで返します。
protected void	Validate() [320 ページ]	
protected void	Validate(int) [320 ページ]	

備考

このクラスにはコンストラクタがありません。結果セットは、ULCommand.ExecuteResultSet メソッドを使用して作成されます。

```
' Visual Basic
Dim cmd As ULCommand = new ULCommand( _
    "SELECT emp_id FROM employee", conn _
)
Dim resultSet As ULResultSet = cmd.ExecuteResultSet()
```

対応する C# 言語のコードを次に示します。

```
// C#
ULCommand cmd = new ULCommand(
    "SELECT emp_id FROM employee", conn
);
ULResultSet resultSet = cmd.ExecuteResultSet();
```

ULResultSet オブジェクトは、位置付け更新や削除の実行対象となる編集可能な結果を表します。編集可能な結果セットには、ULCommand.ExecuteTable メソッドまたは ULDataAdapter クラスを使用します。

このセクションの内容:

[AppendBytes\(int, byte\[\], int, int\) メソッド \[446 ページ\]](#)

指定された System.Bytes 配列の指定されたサブセットを、指定された ULDbType.LongBinary カラムの新しい値に追加します。

[AppendChars\(int, char\[\], int, int\) メソッド \[447 ページ\]](#)

指定された System.Chars 配列の指定されたサブセットを、指定された ULDbType.LongVarchar カラムの新しい値に追加します。

[Delete\(\) メソッド \[449 ページ\]](#)

現在の行を削除します。

[SetBoolean\(int, bool\) メソッド \[449 ページ\]](#)

指定されたカラムの値を、System.Boolean を使用して設定します。

[SetByte\(int, byte\) メソッド \[450 ページ\]](#)

指定されたカラムの値を、System.Byte (符号なし 8 ビット整数) を使用して設定します。

[SetBytes\(int, byte\[\]\) メソッド \[451 ページ\]](#)

指定されたカラムの値を、System.Bytes を使用して設定します。

[SetDateTime\(int, DateTime\) メソッド \[453 ページ\]](#)

指定されたカラムの値を、System.DateTime を使用して設定します。

[SetDBNull\(int\) メソッド \[454 ページ\]](#)

カラムを NULL に設定します。

[SetDecimal\(int, decimal\) メソッド \[455 ページ\]](#)

指定されたカラムの値を、System.Decimal を使用して設定します。

[SetDouble\(int, double\) メソッド \[456 ページ\]](#)

指定されたカラムの値を、System.Double を使用して設定します。

[SetFloat\(int, float\) メソッド \[457 ページ\]](#)

指定されたカラムの値を、System.Single を使用して設定します。

[SetGuid\(int, Guid\) メソッド \[458 ページ\]](#)

指定されたカラムの値を、System.Guid を使用して設定します。

[SetInt16\(int, short\) メソッド \[459 ページ\]](#)

指定されたカラムの値を、System.Int16 を使用して設定します。

[SetInt32\(int, int\) メソッド \[461 ページ\]](#)

指定されたカラムの値を、System.Int32 を使用して設定します。

[SetInt64\(int, long\) メソッド \[462 ページ\]](#)

指定されたカラムの値を、Int64 を使用して設定します。

[SetString\(int, string\) メソッド \[463 ページ\]](#)

指定されたカラムの値を、System.String を使用して設定します。

[SetTimeSpan\(int, TimeSpan\) メソッド \[464 ページ\]](#)

指定されたカラムの値を、System.TimeSpan を使用して設定します。

[SetToDefault\(int\) メソッド \[465 ページ\]](#)

指定されたカラムの値を、そのデフォルト値に設定します。

[SetUInt16\(int, ushort\) メソッド \[466 ページ\]](#)

指定されたカラムの値を、System.UInt16 を使用して設定します。

[SetUInt32\(int, uint\) メソッド \[467 ページ\]](#)

指定されたカラムの値を、System.UInt32 を使用して設定します。

[SetUInt64\(int, ulong\) メソッド \[468 ページ\]](#)

指定されたカラムの値を、System.UInt64 を使用して設定します。

[Update\(\) メソッド \[470 ページ\]](#)

現在のカラム値 (set メソッドを使用して指定されます) で新しいローを更新します。

[UpdateBegin\(\) メソッド \[470 ページ\]](#)

現在のローを更新する準備を行います。

関連情報

[ExecuteResultSet\(\) メソッド \[89 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

[ExecuteTable\(\) メソッド \[94 ページ\]](#)

[ULDataAdapter クラス \[238 ページ\]](#)

[ULDataReader クラス \[275 ページ\]](#)

1.26.1 AppendBytes(int, byte[], int, int) メソッド

指定された System.Bytes 配列の指定されたサブセットを、指定された ULDbType.LongBinary カラムの新しい値に追加します。

構文

Visual Basic

```
Public Sub AppendBytes (  
    ByVal colID As Integer,  
    ByVal val As Byte(),  
    ByVal srcOffset As Integer,  
    ByVal count As Integer  
)
```

C#

```
public unsafe void AppendBytes (  
    int colID,  
    byte[] val,  
    int srcOffset,  
    int count  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

val カラムの現在の新しい値に追加する値。

srcOffset ソース配列の開始位置。

count コピーされるバイト数。

例外

ULException class SQL エラーが発生しました。

備考

srcOffset (0 から始まります) から、配列 **val** の **srcOffset+count-1** までの位置のバイトが、指定されたカラムの値に追加されます。

挿入時には、ULTable.InsertBegin は新しい値をカラムのデフォルト値に初期化します。ローのデータは、ULTable.Insert が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

更新時の、カラムに対する最初の追加では、現在のカラム値がクリアされてから新しい値が追加されます。

次のいずれかに該当する場合、コード `ULSQLCode.SQLE_INVALID_PARAMETER` とともに `ULException` がスローされ、追加先は修正されません。

- `val` が NULL です。
- `srcOffset` が負です。
- `count` が負です。
- `srcOffset+count` が `val` の長さより長い。

その他のエラーの場合は、それに応じたエラーコードとともに `ULException` がスローされます。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[InsertBegin\(\) メソッド \[546 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[ULException クラス \[328 ページ\]](#)

[ULException クラス \[328 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.2 AppendChars(int, char[], int, int) メソッド

指定された `System.Chars` 配列の指定されたサブセットを、指定された `ULDbType.LongVarchar` カラムの新しい値に追加します。

構文

Visual Basic

```
Public Sub AppendChars (  
    ByVal colID As Integer,  
    ByVal val As Char(),  
    ByVal srcOffset As Integer,  
    ByVal count As Integer  
)
```

C#

```
public unsafe void AppendChars (  
    int colID,  
    char[] val,  
    int srcOffset,  
    int count  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

val カラムの現在の新しい値に追加する値。

srcOffset ソース配列の開始位置。

count コピーされるバイト数。

例外

ULException class SQL エラーが発生しました。

備考

srcOffset (0 から始まります) から、配列 *val* の *srcOffset+count-1* までの位置の文字が、指定されたカラムの値に追加されます。挿入時には、`ULTable.InsertBegin` は新しい値をカラムのデフォルト値に初期化します。ローのデータは、`ULTable.Insert` が実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

更新時の、カラムに対する最初の追加では、現在のカラム値がクリアされてから新しい値が追加されます。

次のいずれかに該当する場合、コード `ULSQLCode.SQLE_INVALID_PARAMETER` とともに `ULException` がスローされ、追加先は修正されません。

- *val* が NULL です。
- *srcOffset* が負です。
- *count* が負です。
- *srcOffset+count* が *val* の長さより長い。

その他のエラーの場合は、それに応じたエラーコードとともに `ULException` がスローされます。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[InsertBegin\(\) メソッド \[546 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[ULException クラス \[328 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.3 Delete() メソッド

現在の行を削除します。

構文

Visual Basic

```
Public Sub Delete ()
```

C#

```
public void Delete ()
```

例外

ULException class SQL エラーが発生しました。

関連情報

[StartSynchronizationDelete\(\) メソッド \[169 ページ\]](#)

[StopSynchronizationDelete\(\) メソッド \[170 ページ\]](#)

1.26.4 SetBoolean(int, bool) メソッド

指定されたカラムの値を、System.Boolean を使用して設定します。

構文

Visual Basic

```
Public Sub SetBoolean (  
    ByVal colID As Integer,  
    ByVal val As Boolean  
)
```

C#

```
public void SetBoolean (  
    int colID,  
    bool val  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.5 SetByte(int, byte) メソッド

指定されたカラムの値を、System.Byte (符号なし 8 ビット整数) を使用して設定します。

構文

Visual Basic

```
Public Sub SetByte (  
    ByVal colID As Integer,  
    ByVal val As Byte  
)
```

C#

```
public void SetByte (
```

```
int colID,  
byte val  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

ローのデータは、**ULTable.Insert** または **Update** メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.6 SetBytes(int, byte[]) メソッド

指定されたカラムの値を、**System.Bytes** を使用して設定します。

構文

Visual Basic

```
Public Sub SetBytes (
```

```
ByVal colID As Integer,  
ByVal val As Byte()  
)
```

C#

```
public unsafe void SetBytes (  
    int colID,  
    byte[] val  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

ULDbType.Binary 型または ULDbType.LongBinary 型のカラム、あるいは値の長さが 16 の場合の ULDbType.Uniqueidentifier 型のカラムにのみ適しています。ローのデータは、ULTable.Insert または Update メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.7 SetDateTime(int, DateTime) メソッド

指定されたカラムの値を、System.DateTime を使用して設定します。

構文

Visual Basic

```
Public Sub SetDateTime (  
    ByVal colID As Integer,  
    ByVal val As Date  
)
```

C#

```
public unsafe void SetDateTime (  
    int colID,  
    DateTime val  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

設定された値の精度はミリ秒です。ローのデータは、ULTable.Insert または Update メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.8 SetDBNull(int) メソッド

カラムを NULL に設定します。

構文

Visual Basic

```
Public Sub SetDBNull (ByVal colID As Integer)
```

C#

```
public void SetDBNull (int colID)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

例外

ULException class SQL エラーが発生しました。

備考

データは、ULTable.Insert または Update を実行するまでは、実際には変更されません。また、変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[IsColumnNullable\(int\) メソッド \[571 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.9 SetDecimal(int, decimal) メソッド

指定されたカラムの値を、System.Decimal を使用して設定します。

構文

Visual Basic

```
Public Sub SetDecimal (  
    ByVal colID As Integer,  
    ByVal val As Decimal  
)
```

C#

```
public void SetDecimal (  
    int colID,  
    decimal val  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.10 SetDouble(int, double) メソッド

指定されたカラムの値を、System.Double を使用して設定します。

構文

Visual Basic

```
Public Sub SetDouble (  
    ByVal colID As Integer,  
    ByVal val As Double  
)
```

C#

```
public void SetDouble (  
    int colID,  
    double val  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.11 SetFloat(int, float) メソッド

指定されたカラムの値を、System.Single を使用して設定します。

構文

Visual Basic

```
Public Sub SetFloat (  
    ByVal colID As Integer,  
    ByVal val As Single  
)
```

C#

```
public void SetFloat (  
    int colID,  
    float val  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.12 SetGuid(int, Guid) メソッド

指定されたカラムの値を、System.Guid を使用して設定します。

構文

Visual Basic

```
Public Sub SetGuid (  
    ByVal colID As Integer,  
    ByVal val As Guid  
)
```

C#

```
public unsafe void SetGuid (  
    int colID,  
    Guid val  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。ULDbType.UniqueIdentifier 型のカラムか、または長さが 16 の ULDbType.Binary 型のカラムの場合にのみ有効です。

関連情報

[GetNewUUID\(\) メソッド \[155 ページ\]](#)

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[GetColumnSize\(int\) メソッド \[233 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.13 SetInt16(int, short) メソッド

指定されたカラムの値を、System.Int16 を使用して設定します。

構文

Visual Basic

```
Public Sub SetInt16 (  
    ByVal colID As Integer,  
    ByVal val As Short
```

```
)
```

C#

```
public void SetInt16 (  
    int colID,  
    short val  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.14 SetInt32(int, int) メソッド

指定されたカラムの値を、System.Int32 を使用して設定します。

構文

Visual Basic

```
Public Sub SetInt32 (  
    ByVal colID As Integer,  
    ByVal val As Integer  
)
```

C#

```
public void SetInt32 (  
    int colID,  
    int val  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.15 SetInt64(int, long) メソッド

指定されたカラムの値を、Int64 を使用して設定します。

構文

Visual Basic

```
Public Sub SetInt64 (  
    ByVal colID As Integer,  
    ByVal val As Long  
)
```

C#

```
public void SetInt64 (  
    int colID,  
    long val  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.16 SetString(int, string) メソッド

指定されたカラムの値を、System.String を使用して設定します。

構文

Visual Basic

```
Public Sub SetString (  
    ByVal colID As Integer,  
    ByVal val As String  
)
```

C#

```
public unsafe void SetString (  
    int colID,  
    string val  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.17 SetTimeSpan(int, TimeSpan) メソッド

指定されたカラムの値を、System.TimeSpan を使用して設定します。

構文

Visual Basic

```
Public Sub SetTimeSpan (  
    ByVal colID As Integer,  
    ByVal val As TimeSpan  
)
```

C#

```
public unsafe void SetTimeSpan (  
    int colID,  
    TimeSpan val  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

`ULException class` SQL エラーが発生しました。

備考

設定された値は、ミリ秒の精度であり、0 ~ 24 時までの間の負でない値に正規化されます。ローのデータは、`ULTable.Insert` または `Update` メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.18 SetToDefault(int) メソッド

指定されたカラムの値を、そのデフォルト値に設定します。

構文

Visual Basic

```
Public Sub SetToDefault (ByVal colID As Integer)
```

C#

```
public void SetToDefault (int colID)
```

パラメータ

`colID` カラムの ID 番号。値は、`[0,ULDataReader.FieldCount-1]` の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

例外

ULException class SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[GetColumnDefaultValue\(int\) メソッド \[559 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.19 SetUInt16(int, ushort) メソッド

指定されたカラムの値を、System.UInt16 を使用して設定します。

構文

Visual Basic

```
Public Sub SetUInt16 (  
    ByVal colID As Integer,  
    ByVal val As UShort  
)
```

C#

```
public void SetUInt16 (  
    int colID,  
    ushort val  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.20 SetUInt32(int, uint) メソッド

指定されたカラムの値を、System.UInt32 を使用して設定します。

構文

Visual Basic

```
Public Sub SetUInt32 (  
    ByVal colID As Integer,  
    ByVal val As UInteger  
)
```

C#

```
public void SetUInt32 (  
    int colID,  
    uint val
```

)

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内であることが必要です。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.21 SetUInt64(int, ulong) メソッド

指定されたカラムの値を、System.UInt64 を使用して設定します。

構文

Visual Basic

```
Public Sub SetUInt64 (  
    ByVal colID As Integer,
```

```
        ByVal val As ULong  
    )
```

C#

```
public void SetUInt64 (  
    int colID,  
    ulong val  
)
```

パラメータ

colID カラムの ID 番号。値は、[0,ULDataReader.FieldCount-1] の範囲内である必要があります。カーソルの先頭カラムの ID 値は 0 です。

val カラムの新しい値。

例外

ULException class SQL エラーが発生しました。

備考

ローのデータは、ULTable.Insert または Update メソッドが実行されるまで実際には変更されません。また、その変更は、コミットされないかぎり、永続化されません。

関連情報

[GetOrdinal\(string\) メソッド \[302 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

[Update\(\) メソッド \[470 ページ\]](#)

[FieldCount プロパティ \[321 ページ\]](#)

1.26.22 Update() メソッド

現在のカラム値 (set メソッドを使用して指定されます) で新しいローを更新します。

構文

Visual Basic

```
Public Sub Update ()
```

C#

```
public void Update ()
```

例外

ULException class SQL エラーが発生しました。

関連情報

[UpdateBegin\(\) メソッド \[470 ページ\]](#)

1.26.23 UpdateBegin() メソッド

現在のローを更新する準備を行います。

構文

Visual Basic

```
Public Sub UpdateBegin ()
```

C#

```
public void UpdateBegin ()
```

例外

ULException class SQL エラーが発生しました。

備考

カラム値は、適切な setType メソッドまたは AppendType メソッドを呼び出すことによって修正します。カラムに対する最初の追加では、現在のカラム値がクリアされてから新しい値が追加されます。

ローのデータは、Update メソッドが実行されるまで実際には変更されません。また、変更は、コミットされないかぎり、永続化されません。

テーブルを開くのに使用されるインデックス内のカラムを修正すると、アクティブな検索処理に予期しない影響を及ぼします。テーブルのプライマリキー内のカラムは更新できません。

関連情報

[Update\(\) メソッド \[470 ページ\]](#)

1.27 ULResultSetSchema クラス

UL 拡張: Ultra Light の結果セットのスキーマを表します。

構文

Visual Basic

```
Public NotInheritable Class ULResultSetSchema Inherits ULCursorSchema
```

C#

```
public sealed class ULResultSetSchema : ULCursorSchema
```

メンバー

ULResultSetSchema 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
protected virtual override void	VerifyOpen() [473 ページ]	

プロパティ

変数とタイプ	プロパティ	説明
public override string	Name [473 ページ]	カーソルの名前を返します。

ULCursorSchema 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 short	ColumnCount [237 ページ]	このカーソル内のカラム数を返します。
protected unsafe void	GetColumnCount() [229 ページ]	
public unsafe short	GetColumnID(string) [229 ページ]	指定されたカラムのカラム ID を返します。
public string	GetColumnName(int) [230 ページ]	指定されたカラム ID で識別されたカラムの名前を返します。
public unsafe int	GetColumnPrecision(int) [231 ページ]	カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの精度を返します。
public unsafe int	GetColumnScale(int) [232 ページ]	カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの位取りを返します。
public unsafe int	GetColumnSize(int) [233 ページ]	カラムがサイズ指定されたカラム (BINARY SQL 型または CHAR SQL 型) の場合は、指定されたカラム ID によって識別されたカラムのサイズを返します。
public string	GetColumnSQLName(int) [234 ページ]	指定されたカラム ID で識別されたカラムの名前を返します。
public unsafe ULDbType	GetColumnULDbType(int) [235 ページ]	指定されたカラム ID によって識別されたカラムの Ultra Light.NET データ型を返します。
public unsafe DataTable	GetSchemaTable() [236 ページ]	ULDataReader オブジェクトのカラムのスキーマが記述された System.Data.DataTable を返します。
public bool	IsOpen [237 ページ]	カーソルのスキーマが現在開いているかどうかを確認します。

備考

このクラスにはコンストラクタがありません。ULResultSetSchema オブジェクトは、その ULDataReader.Schema プロパティとして結果セットにアタッチされます。

結果セットのスキーマが有効なのは、データリーダーが開かれている間だけです。

このセクションの内容:

[VerifyOpen\(\) メソッド](#) [473 ページ]

[Name](#) プロパティ [473 ページ]

カーソルの名前を返します。

関連情報

[ULCommand クラス \[58 ページ\]](#)

[ULDataReader クラス \[275 ページ\]](#)

[Schema プロパティ \[325 ページ\]](#)

[ULCursorSchema クラス \[226 ページ\]](#)

1.27.1 VerifyOpen() メソッド

構文

Visual Basic

```
Protected Overridable Overrides Sub VerifyOpen ()
```

C#

```
protected virtual override void VerifyOpen ()
```

1.27.2 Name プロパティ

カーソルの名前を返します。

構文

Visual Basic

```
Public ReadOnly Overrides Property Name As String
```

C#

```
public override string Name {get;}
```

備考

ULResultSetSchema を生成した SQL 文。

1.28 ULRowsCopiedEventArgs クラス

ULRowsCopiedEventHandler オブジェクトに渡される引数のセットを表します。

構文

Visual Basic

```
Public NotInheritable Class ULRowsCopiedEventArgs
```

C#

```
public sealed class ULRowsCopiedEventArgs
```

メンバー

ULRowsCopiedEventArgs 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	ULRowsCopiedEventArgs(long) [475 ページ]	ULRowsCopiedEventArgs オブジェクトの新しいインスタンスを作成します。

プロパティ

変更子とタイプ	プロパティ	説明
public bool	Abort [475 ページ]	バルクコピーオペレーションをアボートするかどうかを示す値を取得または設定します。
public long	RowsCopied [476 ページ]	現在のバルクコピーオペレーションでコピーされるローの数を返します。

備考

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

このセクションの内容:

[ULRowsCopiedEventArgs\(long\) コンストラクタ \[475 ページ\]](#)

ULRowsCopiedEventArgs オブジェクトの新しいインスタンスを作成します。

[Abort プロパティ \[475 ページ\]](#)

バルクコピーオペレーションをアボートするかどうかを示す値を取得または設定します。

[RowsCopied プロパティ \[476 ページ\]](#)

現在のバルクコピーオペレーションでコピーされるローの数を返します。

関連情報

[ULRowsCopiedEventHandler\(object, ULRowsCopiedEventArgs\) デリゲート \[581 ページ\]](#)

1.28.1 ULRowsCopiedEventArgs(long) コンストラクタ

ULRowsCopiedEventArgs オブジェクトの新しいインスタンスを作成します。

構文

Visual Basic

```
Public Sub ULRowsCopiedEventArgs (ByVal rowsCopied As Long)
```

C#

```
public ULRowsCopiedEventArgs (long rowsCopied)
```

パラメータ

rowsCopied 現在のバルクコピーオペレーションでコピーされるローの数を示す 64 ビット整数値。

備考

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

1.28.2 Abort プロパティ

バルクコピーオペレーションをアボートするかどうかを示す値を取得または設定します。

構文

Visual Basic

```
Public Property Abort As Boolean
```

C#

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

備考

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

1.28.3 RowsCopied プロパティ

現在のバルクコピーオペレーションでコピーされるローの数を返します。

≡ 構文

Visual Basic

```
Public ReadOnly Property RowsCopied As Long
```

C#

```
public long RowsCopied {get;}
```

備考

コピーされたロー数を表す long 型整数。

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

1.29 ULRowUpdatedEventArgs クラス

ULDataAdapter.RowUpdated イベントのデータを提供します。

≡ 構文

Visual Basic

```
Public NotInheritable Class ULRowUpdatedEventArgs Inherits  
System.Data.Common.RowUpdatedEventArgs
```

C#

```
public sealed class ULRowUpdatedEventArgs :  
System.Data.Common.RowUpdatedEventArgs
```

メンバー

ULRowUpdatedEventArgs 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	ULRowUpdatedEventArgs(DataRow, IDbCommand, StatementType, DataTableMapping) [478 ページ]	ULRowUpdatedEventArgs クラスの新しいインスタンスを初期化します。

プロパティ

変更子とタイプ	プロパティ	説明
public new ULCommand	Command [478 ページ]	DbDataAdapter.Update メソッドが呼び出されると実行される ULCommand オブジェクトを返します。
public new int	RecordsAffected [479 ページ]	SQL 文の実行によって変更、挿入、または削除されたローの数を返します。

このセクションの内容:

[ULRowUpdatedEventArgs\(DataRow, IDbCommand, StatementType, DataTableMapping\) コンストラクタ \[478 ページ\]](#)

ULRowUpdatedEventArgs クラスの新しいインスタンスを初期化します。

[Command プロパティ \[478 ページ\]](#)

DbDataAdapter.Update メソッドが呼び出されると実行される ULCommand オブジェクトを返します。

[RecordsAffected プロパティ \[479 ページ\]](#)

SQL 文の実行によって変更、挿入、または削除されたローの数を返します。

関連情報

[RowUpdated イベント \[254 ページ\]](#)

1.29.1 ULRowUpdatedEventArgs(DataRow, IDbCommand, StatementType, DataTableMapping) コンストラクタ

ULRowUpdatedEventArgs クラスの新しいインスタンスを初期化します。

構文

Visual Basic

```
Public Sub ULRowUpdatedEventArgs (  
    ByVal row As DataRow,  
    ByVal command As IDbCommand,  
    ByVal statementType As StatementType,  
    ByVal tableMapping As DataTableMapping  
)
```

C#

```
public ULRowUpdatedEventArgs (  
    DataRow row,  
    IDbCommand command,  
    StatementType statementType,  
    DataTableMapping tableMapping  
)
```

パラメータ

row DbDataAdapter.Update 呼び出しにより送信される System.Data.DataRow。

command DbDataAdapter.Update メソッドが呼び出されると実行される System.Data.IDbCommand。

statementType 実行されたクエリのタイプを指定する System.Data.StatementType 値の 1 つ。

tableMapping DbDataAdapter.Update 呼び出しにより送信される System.Data.Common.DataTableMapping。

1.29.2 Command プロパティ

DbDataAdapter.Update メソッドが呼び出されると実行される ULCommand オブジェクトを返します。

構文

Visual Basic

```
Public ReadOnly Shadows Property Command As ULCommand
```

C#

```
public new ULCommand Command {get;}
```

備考

更新で実行された ULCommand オブジェクト。

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

関連情報

[ULCommand クラス \[58 ページ\]](#)

1.29.3 RecordsAffected プロパティ

SQL 文の実行によって変更、挿入、または削除されたローの数を返します。

構文

Visual Basic

```
Public ReadOnly Shadows Property RecordsAffected As Integer
```

C#

```
public new int RecordsAffected {get;}
```

備考

SELECT 文の場合、この値は -1 です。

変更、挿入、または削除されたローの数。文が失敗したときにローが影響されなかった場合は 0、SELECT 文の場合は -1。

1.30 ULRowUpdatingEventArgs クラス

ULDataAdapter.RowUpdating イベントのデータを提供します。

構文

Visual Basic

```
Public NotInheritable Class ULRowUpdatingEventArgs Inherits  
System.Data.Common.RowUpdatingEventArgs
```

C#

```
public sealed class ULRowUpdatingEventArgs :  
System.Data.Common.RowUpdatingEventArgs
```

メンバー

ULRowUpdatingEventArgs 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	ULRowUpdatingEventArgs(DataRow, IDbCommand, StatementType, DataTableMapping) [481 ページ]	ULRowUpdatingEventArgs クラスの新しいインスタンスを初期化します。

プロパティ

変更子とタイプ	プロパティ	説明
public new ULCommand	Command [481 ページ]	DbDataAdapter.Update メソッドの実行時に実行する ULCommand オブジェクトを指定します。

このセクションの内容:

[ULRowUpdatingEventArgs\(DataRow, IDbCommand, StatementType, DataTableMapping\) コンストラクタ](#) [481 ページ]

ULRowUpdatingEventArgs クラスの新しいインスタンスを初期化します。

[Command](#) プロパティ [481 ページ]

DbDataAdapter.Update メソッドの実行時に実行する ULCommand オブジェクトを指定します。

関連情報

[RowUpdating イベント](#) [254 ページ]

1.30.1 ULRowUpdatingEventArgs(DataRow, IDbCommand, StatementType, DataTableMapping) コンストラクタ

ULRowUpdatingEventArgs クラスの新しいインスタンスを初期化します。

構文

Visual Basic

```
Public Sub ULRowUpdatingEventArgs (  
    ByVal row As DataRow,  
    ByVal command As IDbCommand,  
    ByVal statementType As StatementType,  
    ByVal tableMapping As DataTableMapping  
)
```

C#

```
public ULRowUpdatingEventArgs (  
    DataRow row,  
    IDbCommand command,  
    StatementType statementType,  
    DataTableMapping tableMapping  
)
```

パラメータ

row 更新する System.Data.DataRow。

command 更新時に実行される System.Data.IDbCommand。

statementType 実行されたクエリのタイプを指定する System.Data.StatementType 値の 1 つ。

tableMapping DbDataAdapter.Update 呼び出しにより送信される System.Data.Common.DataTableMapping 値。

1.30.2 Command プロパティ

DbDataAdapter.Update メソッドの実行時に実行する ULCommand オブジェクトを指定します。

構文

Visual Basic

```
Public Shadows Property Command As ULCommand
```

C#

```
public new ULCommand Command {get;set;}
```

備考

更新時に実行される ULCommand オブジェクト。

これは、System.Data.Common.RowUpdatingEventArgs.Command 値が厳密に型指定されたものです。

関連情報

[ULCommand クラス \[58 ページ\]](#)

1.31 ULServerSyncListener インタフェース

UL 拡張: サーバの同期メッセージを受信するリスナインタフェースです。

構文

Visual Basic

```
Public Interface ULServerSyncListener
```

C#

```
public interface ULServerSyncListener
```

メンバー

ULServerSyncListener 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	ServerSyncInvoked(string) [483 ページ]	サーバ起動同期用の Mobile Link Listener がアプリケーションを呼び出して同期を実行するときに起動されます。

このセクションの内容:

[ServerSyncInvoked\(string\) メソッド \[483 ページ\]](#)

サーバ起動同期用の Mobile Link Listener がアプリケーションを呼び出して同期を実行するときに起動されます。

1.31.1 ServerSyncInvoked(string) メソッド

サーバ起動同期用の Mobile Link Listener がアプリケーションを呼び出して同期を実行するときに起動されます。

構文

Visual Basic

```
Public Sub ServerSyncInvoked (ByVal messageName As String)
```

C#

```
public void ServerSyncInvoked (string messageName)
```

パラメータ

messageName アプリケーションに送信されるメッセージの名前。

備考

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

例

```
Imports Sap.Data.UltraLite
```

```
Public Class MainWindow Inherits System.Windows.Forms.Form Implements ULServerSyncListener
```

```
Private conn As ULConnection
```

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

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

```
'Add any initialization after the InitializeComponent() call
```

```
ULConnection.DatabaseManager.SetServerSyncListener( _ "myCompany.mymsg", "myCompany.myapp", Me  
_ ) 'Create Connection ...End Sub
```

```
Protected Overrides Sub OnClosing( _ ByVal e As System.ComponentModel.CancelEventArgs _ )
```

```
ULConnection.DatabaseManager.SetServerSyncListener( _ Nothing, Nothing, Nothing _ )
```

```
MyBase.OnClosing(e) End Sub
```

```
Public Sub ServerSyncInvoked(ByVal messageName As String) _ Implements
```

```
ULServerSyncListener.ServerSyncInvoked
```

```
Me.Invoke(New EventHandler(AddressOf Me.ServerSyncAction)) End Sub
```

```
Public Sub ServerSyncAction( _ ByVal sender As Object, ByVal e As EventArgs _ ) ' Do Server sync
conn.Synchronize() End Sub End Class
```

次の C# コードフラグメントは、サーバ同期要求の受信方法と UI スレッドでの同期の実行方法を示しています。

```
using Sap.Data.UltraLite;
public class Form1 : System.Windows.Forms.Form, ULServerSyncListener
{
    private System.Windows.Forms.MainMenu mainMenu1;
    private ULConnection conn;
    public Form1()
    {
        //
        // Required for Windows Form Designer support
        //
        InitializeComponent();
        //
        // TODO: Add any constructor code after
        // InitializeComponent call
        //
        ULConnection.DatabaseManager.SetServerSyncListener(
            "myCompany.mymsg", "myCompany.myapp", this
        );
        // Create connection
        ...
    }
    protected override void Dispose( bool disposing )
    {
        base.Dispose( disposing );
    }
    protected override void OnClosing(
        System.ComponentModel.CancelEventArgs e)
    {
        ULConnection.DatabaseManager.SetServerSyncListener(
            null, null, null
        );
        base.OnClosing(e);
    }
    public void ServerSyncInvoked( string messageName )
    {
        this.Invoke( new EventHandler( ServerSyncHandler ) );
    }
    internal void ServerSyncHandler(object sender, EventArgs e)
    {
        conn.Synchronize();
    }
}
```

1.32 ULSqlProgressData クラス (廃止予定)

UL 拡張: SQL パススルースクリプトの進行状況のモニタリングデータを返します。

構文

Microsoft Visual Basic

```
Public Class ULSqlProgressData
```

C#

```
public class ULSqlProgressData
```

メンバー

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

プロパティ

変更子とタイプ	プロパティ	説明
public long	CurrentScript [485 ページ]	これまでに実行されたスクリプトのインデックス。
public long	ScriptCount [486 ページ]	実行中のスクリプトの数を返します。
public ULSqlProgressState	State [486 ページ]	現在の進行状況のステータスを返します。

このセクションの内容:

[CurrentScript プロパティ \[485 ページ\]](#)

これまでに実行されたスクリプトのインデックス。

[ScriptCount プロパティ \[486 ページ\]](#)

実行中のスクリプトの数を返します。

[State プロパティ \[486 ページ\]](#)

現在の進行状況のステータスを返します。

1.32.1 CurrentScript プロパティ

これまでに実行されたスクリプトのインデックス。

構文

Visual Basic

```
Public ReadOnly Property CurrentScript As Long
```

C#

```
public long CurrentScript {get;}
```

備考

実行中のスクリプトの現在のインデックス。

1.32.2 ScriptCount プロパティ

実行中のスクリプトの数を返します。

構文

Visual Basic

```
Public ReadOnly Property ScriptCount As Long
```

C#

```
public long ScriptCount {get;}
```

備考

実行中のスクリプトの数。

1.32.3 State プロパティ

現在の進行状況のステータスを返します。

構文

Visual Basic

```
Public ReadOnly Property State As ULSqlProgressState
```

C#

```
public ULSqlProgressState State {get;}
```

備考

現在の SQL パススルーのコールバックステータスを指定する ULSqlProgressState 値の 1 つ。

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.33 ULSyncParms クラス

UL 拡張: Ultra Light データベースの同期方法を定義する同期パラメータを表します。

構文

Visual Basic

```
Public NotInheritable Class ULSyncParms
```

C#

```
public sealed class ULSyncParms
```

メンバー

ULSyncParms 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	CopyFrom(ULSyncParms) [490 ページ]	指定された ULSyncParms オブジェクトのプロパティを、この ULSyncParms オブジェクトにコピーします。
public override string	ToString() [490 ページ]	このインスタンスの文字列表現を返します。

プロパティ

変更子とタイプ	プロパティ	説明
public string	AdditionalParms [491 ページ]	"名前=値" のペアをセミコロンで区切ったリストで、追加の同期パラメータを指定します。
public string[]	AuthenticationParms [492 ページ]	カスタムユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。
public bool	DownloadOnly [492 ページ]	同期時のアップロードを無効にするか、有効にするかを指定します。
public bool	KeepPartialDownload [493 ページ]	同期時の部分的なダウンロードを無効にするか、有効にするかを指定します。

変更子とタイプ	プロパティ	説明
public string	NewPassword [494 ページ]	UserName で指定されたユーザの新しい Mobile Link パスワードを指定します。
public string	パスワード [495 ページ]	UserName で指定されたユーザの Mobile Link パスワードです。
public bool	PingOnly [496 ページ]	実際に同期を行う代わりに、クライアントが Mobile Link サーバに ping のみを行うかどうかを指定します。
public string	Publications [497 ページ]	同期させるパブリケーションを指定します。
public bool	ResumePartialDownload [497 ページ]	前の部分的なダウンロードを再開するか、破棄するかを指定します。
public bool	SendDownloadAck [498 ページ]	同期中に、クライアントが Mobile Link サーバにダウンロード確認を送信するかどうかを指定します。
public UStreamType	Stream [499 ページ]	同期に使用する Mobile Link 同期ストリームを指定します。
public string	StreamParms [499 ページ]	同期ストリームの設定パラメータを指定します。
public bool	UploadOnly [500 ページ]	同期時のダウンロードを無効にするか、有効にするかを指定します。
public string	UserName [501 ページ]	Mobile Link サーバが Mobile Link クライアントをユニークに識別するユーザ名です。
public string	Version [502 ページ]	使用する同期スクリプトを指定します。

備考

このクラスにはコンストラクタがありません。各接続には、ULConnection.SyncParms プロパティとしてアタッチされた、固有の ULSyncParms インスタンスがあります。

同期コマンド (ULSyncParms.DownloadOnly プロパティ、ULSyncParms.PingOnly プロパティ、ULSyncParms.ResumePartialDownload プロパティ、または ULSyncParms.UploadOnly プロパティ) は、一度に 1 つしか指定できません。これらの複数のパラメータが true に設定されていると、ULSQLCode.SQLE_SYNC_INFO_INVALID SQLException が ULConnection.Synchronize メソッドによってスローされます。

その他の ULSQLCode.SQLE_SYNC_INFO_INVALID エラーの原因には、ULSyncParms.Stream 値または ULSyncParms.Version 値を指定していないことが含まれます。

このセクションの内容:

[CopyFrom\(ULSyncParms\) メソッド \[490 ページ\]](#)

指定された ULSyncParms オブジェクトのプロパティを、この ULSyncParms オブジェクトにコピーします。

[ToString\(\) メソッド \[490 ページ\]](#)

このインスタンスの文字列表現を返します。

[AdditionalParms プロパティ \[491 ページ\]](#)

"名前=値" のペアをセミコロンで区切ったリストで、追加の同期パラメータを指定します。

[AuthenticationParms プロパティ \[492 ページ\]](#)

カスタムユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。

[DownloadOnly プロパティ \[492 ページ\]](#)

同期時のアップロードを無効にするか、有効にするかを指定します。

[KeepPartialDownload プロパティ \[493 ページ\]](#)

同期時の部分的なダウンロードを無効にするか、有効にするかを指定します。

[NewPassword プロパティ \[494 ページ\]](#)

UserName で指定されたユーザの新しい Mobile Link パスワードを指定します。

[Password プロパティ \[495 ページ\]](#)

UserName で指定されたユーザの Mobile Link パスワードです。

[PingOnly プロパティ \[496 ページ\]](#)

実際に同期を行う代わりに、クライアントが Mobile Link サーバに ping のみを行うかどうかを指定します。

[Publications プロパティ \[497 ページ\]](#)

同期させるパブリケーションを指定します。

[ResumePartialDownload プロパティ \[497 ページ\]](#)

前の部分的なダウンロードを再開するか、破棄するかを指定します。

[SendDownloadAck プロパティ \[498 ページ\]](#)

同期中に、クライアントが Mobile Link サーバにダウンロード確認を送信するかどうかを指定します。

[Stream プロパティ \[499 ページ\]](#)

同期に使用する Mobile Link 同期ストリームを指定します。

[StreamParms プロパティ \[499 ページ\]](#)

同期ストリームの設定パラメータを指定します。

[UploadOnly プロパティ \[500 ページ\]](#)

同期時のダウンロードを無効にするか、有効にするかを指定します。

[UserName プロパティ \[501 ページ\]](#)

Mobile Link サーバが Mobile Link クライアントをユニークに識別するユーザ名です。

[Version プロパティ \[502 ページ\]](#)

使用する同期スクリプトを指定します。

関連情報

[ULConnection クラス \[121 ページ\]](#)

[SyncParms プロパティ \[183 ページ\]](#)

[Synchronize\(\) メソッド \[171 ページ\]](#)

[DownloadOnly プロパティ \[492 ページ\]](#)

[PingOnly プロパティ \[496 ページ\]](#)

[ResumePartialDownload プロパティ \[497 ページ\]](#)

[UploadOnly プロパティ \[500 ページ\]](#)

[Synchronize\(\) メソッド \[171 ページ\]](#)

[Stream プロパティ \[499 ページ\]](#)

[Version プロパティ \[502 ページ\]](#)

1.33.1 CopyFrom(ULSyncParms) メソッド

指定された ULSyncParms オブジェクトのプロパティを、この ULSyncParms オブジェクトにコピーします。

構文

Visual Basic

```
Public Sub CopyFrom (ByVal src As ULSyncParms)
```

C#

```
public void CopyFrom (ULSyncParms src)
```

パラメータ

src コピー元のオブジェクト。

関連情報

[ULSyncParms クラス \[487 ページ\]](#)

1.33.2 ToString() メソッド

このインスタンスの文字列表現を返します。

構文

Visual Basic

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

C#

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

戻り値

"キーワード=値" の組み合わせがセミコロンで区切られたリスト形式の、このインスタンスの文字列表現。

1.33.3 AdditionalParms プロパティ

"名前=値" のペアをセミコロンで区切ったリストで、追加の同期パラメータを指定します。

構文

Visual Basic

```
Public Property AdditionalParms As String
```

C#

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

戻り値

"名前=値" のペアをセミコロンで区切ったリスト形式の文字列。

備考

このプロパティを使用すると、他の定義済みパラメータでは簡単に指定できない追加の同期パラメータをいくつか指定できます。

例

```
private ULSyncParms info;  
// ...  
info.AdditionalParms =  
    "AllowDownloadDupRows=1;  
    CheckpointStore=1;  
    DisableConcurrency=1;  
    TableOrder=Customer,Sales"
```

1.33.4 AuthenticationParms プロパティ

カスタムユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。

構文

Visual Basic

```
Public Property AuthenticationParms As String()
```

C#

```
public string[] AuthenticationParms {get;set;}
```

戻り値

それぞれに認証パラメータが格納された、文字列の配列 (配列のエントリが NULL であると、同期エラーになります)。デフォルト値は NULL 参照 (Visual Basic の Nothing) で、認証パラメータは指定されません。

備考

最初の 255 文字列のみが使用されます。それぞれの文字列は認証パラメータの Mobile Link サーバの制限 (現在は 4000 UTF8 バイト) よりも長くはなりません。

1.33.5 DownloadOnly プロパティ

同期時のアップロードを無効にするか、有効にするかを指定します。

構文

Visual Basic

```
Public Property DownloadOnly As Boolean
```

C#

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

戻り値

同期時のアップロードを無効にする場合は true、有効にする場合は false。デフォルトは false です。

備考

同期コマンド (ULSyncParms.DownloadOnly プロパティ、ULSyncParms.PingOnly プロパティ、ULSyncParms.ResumePartialDownload プロパティ、または ULSyncParms.UploadOnly プロパティ) は、一度に 1 つしか指定できません。これらの複数のパラメータが true に設定されていると、ULSQLCode.SQLE_SYNC_INFO_INVALID SQLException が ULConnection.Synchronize メソッドによってスローされます。

関連情報

[UploadOnly プロパティ \[500 ページ\]](#)

[PingOnly プロパティ \[496 ページ\]](#)

[ResumePartialDownload プロパティ \[497 ページ\]](#)

[UploadOnly プロパティ \[500 ページ\]](#)

[Synchronize\(\) メソッド \[171 ページ\]](#)

1.33.6 KeepPartialDownload プロパティ

同期時の部分的なダウンロードを無効にするか、有効にするかを指定します。

構文

Visual Basic

```
Public Property KeepPartialDownload As Boolean
```

C#

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

戻り値

同期中に部分ダウンロードを有効にして保存する場合は true に設定します。そうではなく、部分ダウンロードを無効にしてエラーが発生したときにダウンロードをロールバックする場合は false に設定します。デフォルトは false です。

備考

Ultra Light.NET では、ULSyncProgressListener オブジェクトを使用して、通信エラーやユーザによるアボートが原因で失敗したダウンロードを再開できます。Ultra Light.NET は、ダウンロードを受信しながら処理します。ダウンロードが中断した場合は、部分的なダウンロードトランザクションがデータベース内に残るため、次の同期中に再開できます。

部分的なダウンロードが保持された場合、ULConnection.Synchronize メソッドの終了時に、ULConnection.ULSyncResult.PartialDownloadRetained プロパティが true に設定されます。

PartialDownloadRetained プロパティが設定されている場合は、ダウンロードを再開できます。このため、ULConnection.ULSyncParms.ResumePartialDownload property を true に設定して ULConnection.Synchronize メソッドを呼び出します。別の通信エラーの発生に備えて、KeepPartialDownload プロパティも true に設定しておきます。ダウンロードが省略された場合は、アップロードは行われません。

再開したダウンロードで受信するダウンロードは、最初にダウンロードを開始したときと同じものです。最新のデータが必要な場合は、再開されたダウンロードの完了直後に、もう一度ダウンロードできます。

ダウンロードを再開する場合、ULSyncParms プロパティの多くは関係ありません。たとえば、Publications プロパティは使用されません。受信するパブリケーションは、最初のダウンロード時に要求したものです。設定する必要があるプロパティは、ResumePartialDownload と UserName だけです。KeepPartialDownload プロパティは、必要に応じて設定可能であり、正常に機能します。

部分的なダウンロードがあり、このダウンロードが必要なくなった場合は、ULConnection.RollbackPartialDownload メソッドを呼び出して、失敗したダウンロードトランザクションをロールバックできます。また、同期を再試行するときに ResumePartialDownload プロパティを指定しなかった場合は、次の同期が開始される前に、部分的なダウンロードがロールバックされます。

関連情報

[PartialDownloadRetained プロパティ \[523 ページ\]](#)

[ResumePartialDownload プロパティ \[497 ページ\]](#)

[RollbackPartialDownload\(\) メソッド \[166 ページ\]](#)

[ResumePartialDownload プロパティ \[497 ページ\]](#)

[UserName プロパティ \[501 ページ\]](#)

[RollbackPartialDownload\(\) メソッド \[166 ページ\]](#)

1.33.7 NewPassword プロパティ

UserName で指定されたユーザの新しい Mobile Link パスワードを指定します。

構文

Visual Basic

```
Public Property NewPassword As String
```

C#

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

戻り値

新しい Mobile Link パスワードを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) で、パスワードは変更されません。

備考

新しいパスワードが有効になるのは、次の同期の後です。

関連情報

[UserName プロパティ \[501 ページ\]](#)

1.33.8 Password プロパティ

UserName で指定されたユーザの Mobile Link パスワードです。

構文

Visual Basic

```
Public Property Password As String
```

C#

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

戻り値

Mobile Link パスワードを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) で、パスワードは指定されません。

備考

Mobile Link ユーザ名とパスワードは他のデータベースユーザ ID やパスワードとは別のもので、アプリケーションを Mobile Link サーバに対して識別し、認証するために使用されます。

関連情報

[NewPassword プロパティ \[494 ページ\]](#)

[UserName プロパティ \[501 ページ\]](#)

1.33.9 PingOnly プロパティ

実際に同期を行う代わりに、クライアントが Mobile Link サーバに ping のみを行うかどうかを指定します。

構文

Visual Basic

```
Public Property PingOnly As Boolean
```

C#

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

戻り値

クライアントが Mobile Link サーバに ping のみを行うように指定する場合は true、クライアントが実際に同期を行うように指定する場合は false。デフォルトは false です。

備考

同期コマンド (ULSyncParms.DownloadOnly プロパティ、ULSyncParms.PingOnly プロパティ、ULSyncParms.ResumePartialDownload プロパティ、または ULSyncParms.UploadOnly プロパティ) は、一度に1つしか指定できません。これらの複数のパラメータが true に設定されていると、ULSQLCode.SQLE_SYNC_INFO_INVALID SQLException が ULConnection.Synchronize メソッドによってスローされます。

関連情報

[DownloadOnly プロパティ \[492 ページ\]](#)

[ResumePartialDownload プロパティ \[497 ページ\]](#)

[UploadOnly プロパティ \[500 ページ\]](#)

[Synchronize\(\) メソッド \[171 ページ\]](#)

1.33.10 Publications プロパティ

同期させるパブリケーションを指定します。

構文

Visual Basic

```
Public Property Publications As String
```

C#

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

戻り値

カンマ (,) 区切りパブリケーション名、特別値 `ULConnection.SYNC_ALL_PUBS`、または特別値 `ULConnection.SYNC_ALL_DB` が含まれる文字列。デフォルトは `ULConnection.SYNC_ALL_DB` です。

1.33.11 ResumePartialDownload プロパティ

前の部分的なダウンロードを再開するか、破棄するかを指定します。

構文

Visual Basic

```
Public Property ResumePartialDownload As Boolean
```

C#

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

戻り値

前の部分的なダウンロードを再開する場合は `true`、破棄する場合は `false`。デフォルトは `false` です。

備考

同期コマンド (`ULSyncParms.DownloadOnly` プロパティ、`ULSyncParms.PingOnly` プロパティ、`ULSyncParms.ResumePartialDownload` プロパティ、または `ULSyncParms.UploadOnly` プロパティ) は、一度に 1 つし

か指定できません。これらの複数のパラメータが true に設定されていると、ULSQLCode.SQLE_SYNC_INFO_INVALID SQLException が ULConnection.Synchronize メソッドによってスローされます。

関連情報

[KeepPartialDownload プロパティ \[493 ページ\]](#)

[DownloadOnly プロパティ \[492 ページ\]](#)

[PingOnly プロパティ \[496 ページ\]](#)

[UploadOnly プロパティ \[500 ページ\]](#)

[Synchronize\(\) メソッド \[171 ページ\]](#)

[PartialDownloadRetained プロパティ \[523 ページ\]](#)

1.33.12 SendDownloadAck プロパティ

同期中に、クライアントが Mobile Link サーバにダウンロード確認を送信するかどうかを指定します。

構文

Visual Basic

```
Public Property SendDownloadAck As Boolean
```

C#

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

戻り値

True に設定すると、クライアントが Mobile Link サーバにダウンロード確認を送信するように指定されます。False に設定すると、ダウンロード確認は送信されません。デフォルトは false です。

備考

ダウンロード確認は、ダウンロードがリモートで完全に適用されてコミットされた後 (正の確認)、またはダウンロードに失敗した後 (負の確認) に送信されます。

クライアントがダウンロード確認を送信する場合、Mobile Link サーバのデータベースワークスレッドは、クライアントがダウンロードを適用してコミットするまで待機します。クライアントがダウンロード確認を送信しない場合、Mobile Link サーバは、次の同期のため、より早く解放されます。

1.33.13 Stream プロパティ

同期に使用する Mobile Link 同期ストリームを指定します。

構文

Visual Basic

```
Public Property Stream As ULStreamType
```

C#

```
public ULStreamType Stream {get;set;}
```

戻り値

使用する同期ストリームのタイプを指定する ULStreamType 値の 1 つ。デフォルト値は ULStreamType.TCPIP です。

備考

ほとんどの同期ストリームでは、Mobile Link サーバのアドレスを識別したり、その他の動作を制御したりするパラメータが必要です。これらのパラメータは、ULSyncParms.StreamParms プロパティで指定します。

ストリームタイプが、プラットフォームに適さない無効な値に設定されていると、ストリームタイプは ULStreamType.TCPIP に設定されます。

関連情報

[ULStreamType 列挙体 \[592 ページ\]](#)

[StreamParms プロパティ \[499 ページ\]](#)

1.33.14 StreamParms プロパティ

同期ストリームの設定パラメータを指定します。

構文

Visual Basic

```
Public Property StreamParms As String
```

C#

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

戻り値

キーワードと値の組み合わせがセミicolonで区切られたリスト形式の、ストリームのパラメータを指定する文字列。デフォルト値は NULL 参照 (Visual Basic の Nothing) です。

備考

StreamParms は、同期ストリームに使用されるすべてのパラメータが含まれている文字列です。パラメータは、"名前=値" のペアをセミicolonで区切ったリスト ("param1=value1;param2=value2") で指定します。

関連情報

[Stream プロパティ \[499 ページ\]](#)

[ULStreamType 列挙体 \[592 ページ\]](#)

1.33.15 UploadOnly プロパティ

同期時のダウンロードを無効にするか、有効にするかを指定します。

構文

Visual Basic

```
Public Property UploadOnly As Boolean
```

C#

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

戻り値

ダウンロードを無効にする場合は true、有効にする場合は false。デフォルトは false です。

備考

同期コマンド (ULSyncParms.DownloadOnly プロパティ、ULSyncParms.PingOnly プロパティ、ULSyncParms.ResumePartialDownload プロパティ、または ULSyncParms.UploadOnly プロパティ) は、一度に 1 つしか指定できません。これらの複数のパラメータが true に設定されていると、ULSQLCode.SQLE_SYNC_INFO_INVALID SQLException が ULConnection.Synchronize メソッドによってスローされます。

関連情報

[DownloadOnly プロパティ \[492 ページ\]](#)

[PingOnly プロパティ \[496 ページ\]](#)

[ResumePartialDownload プロパティ \[497 ページ\]](#)

[Synchronize\(\) メソッド \[171 ページ\]](#)

1.33.16 UserName プロパティ

Mobile Link サーバが Mobile Link クライアントをユニークに識別するユーザ名です。

構文

Visual Basic

```
Public Property UserName As String
```

C#

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

戻り値

ユーザ名を指定する文字列。このパラメータにはデフォルト値がないので、明示的に設定してください。

備考

Mobile Link サーバでは、この値を使用して、ダウンロードする内容の決定、同期ステータスの記録、同期中の割り込みからの復帰を行います。このユーザ名とパスワードは他のデータベースユーザ ID やパスワードとは別のもので、アプリケーションを Mobile Link サーバに対して識別し、認証するために使用されます。

関連情報

[Password プロパティ \[495 ページ\]](#)

1.33.17 Version プロパティ

使用する同期スクリプトを指定します。

構文

Visual Basic

```
Public Property Version As String
```

C#

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

戻り値

使用する同期スクリプトのバージョンを指定する文字列。このパラメータにはデフォルト値がないので、明示的に設定してください。

備考

統合データベースの同期スクリプトは、それぞれバージョン文字列でマーク付けされます。たとえば、異なる文字列バージョンによって特定される 2 種類の `download_cursor` スクリプトがあります。Ultra Light アプリケーションは、バージョン文字列により、同期スクリプトのセットから選択できます。

1.34 ULSyncProgressData クラス

UL 拡張: 同期の進行状況のモニタリングデータを返します。

構文

Visual Basic

```
Public Class ULSyncProgressData
```

C#

```
public class ULSyncProgressData
```

メンバー

ULSyncProgressData 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 const int	FLAG_IS_BLOCKING	Mobile Link サーバからの応答の待機中、同期はブロックされていることを示すフラグです。
public const int	FLAG_LAST_UPLOAD_RECEIVED	サーバが最後のアップロードを受信したかどうかを示すフラグ。 受信しなかった場合は、直前の同期からのパブリケーションが次回もう一度同期されるときにアップロードが再送信されます。これは現在の同期中である場合があります。

プロパティ

変数とタイプ	プロパティ	説明
public int	CurrentDownloadRowCount [505 ページ]	これまでにダウンロードされたローの数を返します。
public int	Flags [506 ページ]	現在の状態に関連する追加情報を示す、現在の同期フラグを返します。
public int	IgnoredDeletes [507 ページ]	これまでに受信したローのうち削除されたものの数を返します。
public int	IgnoredUpdates [507 ページ]	これまでに受信したローのうち更新されたものの数を返します。
public bool	IsFinalSyncProgress [508 ページ]	これが最終同期進捗メッセージの場合は、true を返します。
public long	ReceivedBytes [508 ページ]	現在までに受信したバイト数を返します。
public int	ReceivedDeletes [509 ページ]	現在までに受信した削除済みのローの数を返します。
public int	ReceivedInserts [510 ページ]	現在までに受信した挿入済みのローの数を返します。
public int	ReceivedUpdates [510 ページ]	現在までに受信した更新済みのローの数を返します。
public long	SentBytes [511 ページ]	現在までに送信されたバイト数を返します。

変更子とタイプ	プロパティ	説明
public int	SentDeletes [512 ページ]	現在までに送信された削除済みのローの数を返します。
public int	SentInserts [512 ページ]	現在までに送信された挿入済みのローの数を返します。
public int	SentUpdates [513 ページ]	現在までに送信された更新済みのローの数を返します。
public ULSyncProgressState	State [513 ページ]	現在の同期のステータスを返します。
public int	SyncTableCount [514 ページ]	同期中のテーブルの数を返します。
public int	SyncTableIndex [515 ページ]	現在同期中のテーブルのインデックスを、1 から、同期に関与しているテーブルの合計数までの範囲で返します。
public int	TableID [515 ページ]	現在同期中のテーブルのデータベースインデックスを返します。
public string	TableName [516 ページ]	アップロード中またはダウンロード中の現在のテーブルの名前を返します。
public int	TotalDownloadRowCount [516 ページ]	ダウンロードで受信するローの合計数を返します。
public int	TruncateDeletes [517 ページ]	受信したローのうちtruncate操作によって削除されたものの数を返します。

このセクションの内容:

[CurrentDownloadRowCount プロパティ \[505 ページ\]](#)

これまでにダウンロードされたローの数を返します。

[Flags プロパティ \[506 ページ\]](#)

現在の状態に関連する追加情報を示す、現在の同期フラグを返します。

[IgnoredDeletes プロパティ \[507 ページ\]](#)

これまでに受信したローのうち削除されたものの数を返します。

[IgnoredUpdates プロパティ \[507 ページ\]](#)

これまでに受信したローのうち更新されたものの数を返します。

[IsFinalSyncProgress プロパティ \[508 ページ\]](#)

これが最終同期進捗メッセージの場合は、true を返します。

[ReceivedBytes プロパティ \[508 ページ\]](#)

現在までに受信したバイト数を返します。

[ReceivedDeletes プロパティ \[509 ページ\]](#)

現在までに受信した削除済みのローの数を返します。

[ReceivedInserts プロパティ \[510 ページ\]](#)

現在までに受信した挿入済みのローの数を返します。

[ReceivedUpdates プロパティ \[510 ページ\]](#)

現在までに受信した更新済みのローの数を返します。

[SentBytes プロパティ \[511 ページ\]](#)

現在までに送信されたバイト数を返します。

[SentDeletes プロパティ \[512 ページ\]](#)

現在までに送信された削除済みのローの数を返します。

[SentInserts プロパティ \[512 ページ\]](#)

現在までに送信された挿入済みのローの数を返します。

[SentUpdates プロパティ \[513 ページ\]](#)

現在までに送信された更新済みのローの数を返します。

[State プロパティ \[513 ページ\]](#)

現在の同期のステータスを返します。

[SyncTableCount プロパティ \[514 ページ\]](#)

同期中のテーブルの数を返します。

[SyncTableIndex プロパティ \[515 ページ\]](#)

現在同期中のテーブルのインデックスを、1 から、同期に関与しているテーブルの合計数までの範囲で返します。

[TableID プロパティ \[515 ページ\]](#)

現在同期中のテーブルのデータベースインデックスを返します。

[TableName プロパティ \[516 ページ\]](#)

アップロード中またはダウンロード中の現在のテーブルの名前を返します。

[TotalDownloadRowCount プロパティ \[516 ページ\]](#)

ダウンロードで受信するローの合計数を返します。

[TruncateDeletes プロパティ \[517 ページ\]](#)

受信したローのうちトランケート操作によって削除されたものの数を返します。

関連情報

[ULSyncProgressListener インタフェース \[518 ページ\]](#)

1.34.1 CurrentDownloadRowCount プロパティ

これまでにダウンロードされたローの数を返します。

構文

Visual Basic

```
Public ReadOnly Property CurrentDownloadRowCount As Integer
```

C#

```
public int CurrentDownloadRowCount {get;}
```

戻り値

これまでにダウンロードされたローの数。

備考

この数には、ReceivedInserts、ReceivedUpdates、または ReceivedDeletes に含まれていない、重複するローが含まれます。

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.34.2 Flags プロパティ

現在の状態に関連する追加情報を示す、現在の同期フラグを返します。

構文

Visual Basic

```
Public ReadOnly Property Flags As Integer
```

C#

```
public int Flags {get;}
```

戻り値

フラグの組み合わせを保持する整数。

1.34.3 IgnoredDeletes プロパティ

これまでに受信したローのうち削除されたものの数を返します。

構文

Visual Basic

```
Public ReadOnly Property IgnoredDeletes As Integer
```

C#

```
public int IgnoredDeletes {get;}
```

戻り値

これまでに受信したローのうち、すでに削除されているローの数。

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.34.4 IgnoredUpdates プロパティ

これまでに受信したローのうち更新されたものの数を返します。

構文

Visual Basic

```
Public ReadOnly Property IgnoredUpdates As Integer
```

C#

```
public int IgnoredUpdates {get;}
```

戻り値

これまでに受信したローのうち、すでに更新されているローの数。

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.34.5 IsFinalSyncProgress プロパティ

これが最終同期進捗メッセージの場合は、true を返します。

構文

Visual Basic

```
Public ReadOnly Property IsFinalSyncProgress As Boolean
```

C#

```
public bool IsFinalSyncProgress {get;}
```

戻り値

これが最終同期進捗メッセージの場合は、true になります。

1.34.6 ReceivedBytes プロパティ

現在までに受信したバイト数を返します。

構文

Visual Basic

```
Public ReadOnly Property ReceivedBytes As Long
```

C#

```
public long ReceivedBytes {get;}
```

戻り値

現在までに受信したバイト数。

備考

この情報は、すべてのステータスで更新されます。

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.34.7 ReceivedDeletes プロパティ

現在までに受信した削除済みのローの数を返します。

構文

Visual Basic

```
Public ReadOnly Property ReceivedDeletes As Integer
```

C#

```
public int ReceivedDeletes {get;}
```

戻り値

現在までに受信した削除済みのローの数。

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.34.8 ReceivedInserts プロパティ

現在までに受信した挿入済みのローの数を返します。

構文

Visual Basic

```
Public ReadOnly Property ReceivedInserts As Integer
```

C#

```
public int ReceivedInserts {get;}
```

戻り値

現在までに受信した挿入済みのローの数。

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.34.9 ReceivedUpdates プロパティ

現在までに受信した更新済みのローの数を返します。

構文

Visual Basic

```
Public ReadOnly Property ReceivedUpdates As Integer
```

C#

```
public int ReceivedUpdates {get;}
```

戻り値

現在までに受信した更新済みのローの数。

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.34.10 SentBytes プロパティ

現在までに送信されたバイト数を返します。

構文

Visual Basic

```
Public ReadOnly Property SentBytes As Long
```

C#

```
public long SentBytes {get;}
```

戻り値

現在までに送信されたバイト数。

備考

この情報は、すべてのステータスで更新されます。

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.34.11 SentDeletes プロパティ

現在までに送信された削除済みのローの数を返します。

構文

Visual Basic

```
Public ReadOnly Property SentDeletes As Integer
```

C#

```
public int SentDeletes {get;}
```

戻り値

現在までに送信された削除済みのローの数。

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.34.12 SentInserts プロパティ

現在までに送信された挿入済みのローの数を返します。

構文

Visual Basic

```
Public ReadOnly Property SentInserts As Integer
```

C#

```
public int SentInserts {get;}
```

戻り値

現在までに送信された挿入済みのローの数。

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.34.13 SentUpdates プロパティ

現在までに送信された更新済みのローの数を返します。

構文

Visual Basic

```
Public ReadOnly Property SentUpdates As Integer
```

C#

```
public int SentUpdates {get;}
```

戻り値

現在までに送信された更新済みのローの数。

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.34.14 State プロパティ

現在の同期のステータスを返します。

構文

Visual Basic

```
Public ReadOnly Property State As ULSyncProgressState
```

C#

```
public ULSyncProgressState State {get;}
```

戻り値

現在の同期のステータスを指定する `ULSyncProgressState` 値の 1 つ。

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.34.15 SyncTableCount プロパティ

同期中のテーブルの数を返します。

構文

Visual Basic

```
Public ReadOnly Property SyncTableCount As Integer
```

C#

```
public int SyncTableCount {get;}
```

戻り値

同期中のテーブルの数。テーブルごとに送信と受信のフェーズがあります。したがって、この数は同期されるテーブルの数より多い場合があります。

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.34.16 SyncTableIndex プロパティ

現在同期中のテーブルのインデックスを、1 から、同期に関与しているテーブルの合計数までの範囲で返します。

構文

Visual Basic

```
Public ReadOnly Property SyncTableIndex As Integer
```

C#

```
public int SyncTableIndex {get;}
```

戻り値

現在同期中のテーブルのインデックス (1 ~ SyncTableCount プロパティの値)

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.34.17 TableID プロパティ

現在同期中のテーブルのデータベースインデックスを返します。

構文

Visual Basic

```
Public ReadOnly Property TableID As Integer
```

C#

```
public int TableID {get;}
```

戻り値

データベースインデックス (1 ~ ULDatabaseSchema.TableCount プロパティの値)

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

[TableCount プロパティ \[274 ページ\]](#)

1.34.18 TableName プロパティ

アップロード中またはダウンロード中の現在のテーブルの名前を返します。

構文

Visual Basic

```
Public ReadOnly Property TableName As String
```

C#

```
public string TableName {get;}
```

戻り値

同期される現在のテーブルの名前、該当しない場合は NULL。

1.34.19 TotalDownloadRowCount プロパティ

ダウンロードで受信するローの合計数を返します。

構文

Visual Basic

```
Public ReadOnly Property TotalDownloadRowCount As Integer
```

C#

```
public int TotalDownloadRowCount {get;}
```

戻り値

ダウンロードで受信するローの数。

備考

この数には、ReceivedInserts、ReceivedUpdates、または ReceivedDeletes に含まれていない、重複するローが含まれます。この値は、同期によって最初のテーブルの STATE_RECEIVING_TABLE ステータスが入力されるまで、設定されません。

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.34.20 TruncateDeletes プロパティ

受信したローのうちtruncate操作によって削除されたものの数を返します。

構文

Visual Basic

```
Public ReadOnly Property TruncateDeletes As Integer
```

C#

```
public int TruncateDeletes {get;}
```

戻り値

truncate操作によって削除されたローの数

関連情報

[ULSyncProgressState 列挙体 \[593 ページ\]](#)

1.35 ULSyncProgressListener インタフェース

UL 拡張: 同期プログ्रेसイベントを受信するリスナインタフェースです。

構文

Visual Basic

```
Public Interface ULSyncProgressListener
```

C#

```
public interface ULSyncProgressListener
```

メンバー

ULSyncProgressListener 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 bool	SyncProgressed(ULSyncProgressData) [518 ページ]	ユーザに進行状況を通知するために、同期処理中に呼び出されます。

このセクションの内容:

[SyncProgressed\(ULSyncProgressData\) メソッド \[518 ページ\]](#)

ユーザに進行状況を通知するために、同期処理中に呼び出されます。

関連情報

[Synchronize\(ULSyncProgressListener\) メソッド \[172 ページ\]](#)

1.35.1 SyncProgressed(ULSyncProgressData) メソッド

ユーザに進行状況を通知するために、同期処理中に呼び出されます。

構文

Visual Basic

```
Public Function SyncProgressed (ByVal data As ULSyncProgressData) As Boolean
```

C#

```
public bool SyncProgressed (ULSyncProgressData data)
```

パラメータ

data 最新の同期のプログレスデータを保持している ULSyncProgressData オブジェクト。

戻り値

このメソッドは、同期をキャンセルする場合は true を、続行する場合は false を返す必要があります。

備考

このメソッドは、同期をキャンセルする場合は true を、続行する場合は false を返す必要があります。

SyncProgressed の呼び出し中に、Ultra Light.NET API のメソッドを呼び出さないでください。

関連情報

[ULSyncProgressData クラス \[502 ページ\]](#)

1.36 ULSyncResult クラス

UL 拡張: 前回の同期のステータスを表します。

構文

Visual Basic

```
Public Class ULSyncResult
```

C#

```
public class ULSyncResult
```

メンバー

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

プロパティ

変更子とタイプ	プロパティ	説明
public ULAuthStatusCode	AuthStatus [521 ページ]	前回試行された同期の認証ステータスコードを返します。
public long	AuthValue [522 ページ]	カスタムユーザ認証同期スクリプトからの戻り値を返します。
public bool	IgnoredRows [522 ページ]	前回行われた同期で、アップロードされたローが無視されたかどうかを確認します。
public bool	PartialDownloadRetained [523 ページ]	前回行われた同期で、部分的なダウンロードが保持されたかどうかを確認します。
public ULStreamErrorCode	StreamErrorCode [523 ページ]	ストリーム自体によってレポートされるエラーを返します。
public string	StreamErrorParameters [524 ページ]	ストリームエラーパラメータをカンマで区切ったリストを返します。
public int	StreamErrorSystem [524 ページ]	ストリームエラーシステム固有のコードを返します。
public DateTime	Timestamp [525 ページ]	前回の同期のタイムスタンプを返します。
public bool	UploadOK [525 ページ]	前回のアップロード同期が成功したかどうかをチェックします。

備考

このクラスにはコンストラクタがありません。各接続には、ULConnection.SyncResult プロパティとしてアタッチされた、固有の ULSyncResult インスタンスがあります。ULSyncResult インスタンスが有効なのは、接続が開かれている間だけです。

このセクションの内容:

[AuthStatus プロパティ \[521 ページ\]](#)

前回試行された同期の認証ステータスコードを返します。

[AuthValue プロパティ \[522 ページ\]](#)

カスタムユーザ認証同期スクリプトからの戻り値を返します。

[IgnoredRows プロパティ \[522 ページ\]](#)

前回行われた同期で、アップロードされたローが無視されたかどうかを確認します。

[PartialDownloadRetained プロパティ \[523 ページ\]](#)

前回行われた同期で、部分的なダウンロードが保持されたかどうかを確認します。

[StreamErrorCode プロパティ \[523 ページ\]](#)

ストリーム自体によってレポートされるエラーを返します。

[StreamErrorParameters プロパティ \[524 ページ\]](#)

ストリームエラーパラメータをカンマで区切ったリストを返します。

[StreamErrorSystem プロパティ \[524 ページ\]](#)

ストリームエラーシステム固有のコードを返します。

[Timestamp プロパティ \[525 ページ\]](#)

前回の同期のタイムスタンプを返します。

[UploadOK プロパティ \[525 ページ\]](#)

前回のアップロード同期が成功したかどうかをチェックします。

関連情報

[SyncResult プロパティ \[184 ページ\]](#)

[Synchronize\(\) メソッド \[171 ページ\]](#)

1.36.1 AuthStatus プロパティ

前回試行された同期の認証ステータスコードを返します。

構文

Visual Basic

```
Public ReadOnly Property AuthStatus As ULAuthStatusCode
```

C#

```
public ULAuthStatusCode AuthStatus {get;}
```

備考

前回行われた同期の認証ステータスを示す ULAuthStatusCode 値の 1 つ。

関連情報

[ULAuthStatusCode 列挙体 \[584 ページ\]](#)

1.36.2 AuthValue プロパティ

カスタムユーザ認証同期スクリプトからの戻り値を返します。

構文

Visual Basic

```
Public ReadOnly Property AuthValue As Long
```

C#

```
public long AuthValue {get;}
```

備考

カスタムユーザ認証同期スクリプトから返された long 型整数。

1.36.3 IgnoredRows プロパティ

前回行われた同期で、アップロードされたローが無視されたかどうかを確認します。

構文

Visual Basic

```
Public ReadOnly Property IgnoredRows As Boolean
```

C#

```
public bool IgnoredRows {get;}
```

備考

前回の同期中にアップロードされたローが無視された場合は true、ローが無視されなかった場合は false。

関連情報

[DownloadOnly プロパティ \[492 ページ\]](#)

1.36.4 PartialDownloadRetained プロパティ

前回行われた同期で、部分的なダウンロードが保持されたかどうかを確認します。

構文

Visual Basic

```
Public ReadOnly Property PartialDownloadRetained As Boolean
```

C#

```
public bool PartialDownloadRetained {get;}
```

備考

ダウンロードが中断され、部分的なダウンロードが保持された場合は true、ダウンロードが中断されなかった場合または部分的なダウンロードがロールバックされた場合は false。

関連情報

[KeepPartialDownload プロパティ \[493 ページ\]](#)

1.36.5 StreamErrorCode プロパティ

ストリーム自体によってレポートされるエラーを返します。

構文

Visual Basic

```
Public ReadOnly Property StreamErrorCode As UStreamErrorCode
```

C#

```
public UStreamErrorCode StreamErrorCode {get;}
```

備考

ストリーム自体によってレポートされるエラーを示す UStreamErrorCode 値の 1 つ。ただし、エラーが発生しなかった場合は UStreamErrorCode.NONE。

1.36.6 StreamErrorParameters プロパティ

ストリームエラーパラメータをカンマで区切ったリストを返します。

構文

Visual Basic

```
Public ReadOnly Property StreamErrorParameters As String
```

C#

```
public string StreamErrorParameters {get;}
```

備考

StreamErrorCode プロパティでレポートされるストリームエラーコードのエラーパラメータをカンマで区切ったリストが含まれます。エラーにパラメータがない場合、またエラーが設定されていない場合は、空の文字列になります。

関連情報

[StreamErrorCode プロパティ \[354 ページ\]](#)

1.36.7 StreamErrorSystem プロパティ

ストリームエラーシステム固有のコードを返します。

構文

Visual Basic

```
Public ReadOnly Property StreamErrorSystem As Integer
```

C#

```
public int StreamErrorSystem {get;}
```

備考

ストリームエラーシステム固有のコードを示す整数。

1.36.8 Timestamp プロパティ

前回の同期のタイムスタンプを返します。

構文

Visual Basic

```
Public ReadOnly Property Timestamp As Date
```

C#

```
public DateTime Timestamp {get;}
```

備考

前回の同期のタイムスタンプを指定する System.DateTime 構造体。

1.36.9 UploadOK プロパティ

前回のアップロード同期が成功したかどうかをチェックします。

構文

Visual Basic

```
Public ReadOnly Property UploadOK As Boolean
```

C#

```
public bool UploadOK {get;}
```

備考

前回のアップロード同期が成功であった場合は true、不成功であった場合は false。

1.37 UTable クラス

UL 拡張: Ultra Light データベース内のテーブルを表します。

構文

Visual Basic

```
Public Class UTable Inherits ULResultSet
```

C#

```
public class UTable : ULResultSet
```

メンバー

UTable 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 void	DeleteAllRows() [532 ページ]	テーブルのすべてのローを削除します。
public void	FindBegin() [533 ページ]	テーブルで新規に検索を実行する準備を行います。
public bool	FindFirst [534 ページ]	テーブルを先頭から順方向に移動しながら、現在のインデックスの値かそのセット全体に完全に一致するローを検索します。
public bool	FindLast [537 ページ]	テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に完全に一致するローを検索します。
public bool	FindNext [540 ページ]	現在の位置からテーブルを順方向に移動しながら、次のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、UTable.FindFirst 検索を続行します。
public bool	FindPrevious [543 ページ]	現在の位置からテーブルを逆方向に移動しながら、前のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、UTable.FindLast 検索を続行します。
public void	Insert() [545 ページ]	現在のカラム値 (set メソッドを使用して指定されます) で新しいローを挿入します。
public void	InsertBegin() [546 ページ]	現在のすべてのカラムをデフォルト値に設定して、テーブルに新しいローを挿入する準備を行います。

変数とタイプ	メソッド	説明
public bool	LookupBackward [547 ページ]	テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より小さい値を持つローを検索します。
public void	LookupBegin() [550 ページ]	テーブルで新規に検索を実行する準備を行います。
public bool	LookupForward [551 ページ]	テーブルを最初から順方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より大きい値を持つローを検索します。
public void	Truncate() [554 ページ]	テーブル内のすべてのローを削除し、STOP SYNCHRONIZATION DELETE を一時的にアクティブにします。

プロパティ

変数とタイプ	プロパティ	説明
public new ULTableSchema	Schema [554 ページ]	テーブルスキーマを保持します。

ULResultSet から継承されたメンバー

変数とタイプ	メンバー	説明
public unsafe void	AppendBytes(int, byte[], int, int) [446 ページ]	指定された System.Bytes 配列の指定されたサブセットを、指定された ULDbType.LongBinary カラムの新しい値に追加します。
public unsafe void	AppendChars(int, char[], int, int) [447 ページ]	指定された System.Chars 配列の指定されたサブセットを、指定された ULDbType.LongVarchar カラムの新しい値に追加します。
public void	Delete() [449 ページ]	現在の行を削除します。
public void	SetBoolean(int, bool) [449 ページ]	指定されたカラムの値を、System.Boolean を使用して設定します。
public void	SetByte(int, byte) [450 ページ]	指定されたカラムの値を、System.Byte (符号なし 8 ビット整数) を使用して設定します。
public unsafe void	SetBytes(int, byte[]) [451 ページ]	指定されたカラムの値を、System.Bytes を使用して設定します。
public unsafe void	SetDateTime(int, DateTime) [453 ページ]	指定されたカラムの値を、System.DateTime を使用して設定します。
public void	SetDBNull(int) [454 ページ]	カラムを NULL に設定します。
public void	SetDecimal(int, decimal) [455 ページ]	指定されたカラムの値を、System.Decimal を使用して設定します。
public void	SetDouble(int, double) [456 ページ]	指定されたカラムの値を、System.Double を使用して設定します。

変更子とタイプ	メンバー	説明
public void	SetFloat(int, float) [457 ページ]	指定されたカラムの値を、System.Single を使用して設定します。
public unsafe void	SetGuid(int, Guid) [458 ページ]	指定されたカラムの値を、System.Guid を使用して設定します。
public void	SetInt16(int, short) [459 ページ]	指定されたカラムの値を、System.Int16 を使用して設定します。
public void	SetInt32(int, int) [461 ページ]	指定されたカラムの値を、System.Int32 を使用して設定します。
public void	SetInt64(int, long) [462 ページ]	指定されたカラムの値を、Int64 を使用して設定します。
public unsafe void	SetString(int, string) [463 ページ]	指定されたカラムの値を、System.String を使用して設定します。
public unsafe void	SetTimeSpan(int, TimeSpan) [464 ページ]	指定されたカラムの値を、System.TimeSpan を使用して設定します。
public void	SetToDefault(int) [465 ページ]	指定されたカラムの値を、そのデフォルト値に設定します。
public void	SetUInt16(int, ushort) [466 ページ]	指定されたカラムの値を、System.UInt16 を使用して設定します。
public void	SetUInt32(int, uint) [467 ページ]	指定されたカラムの値を、System.UInt32 を使用して設定します。
public void	SetUInt64(int, ulong) [468 ページ]	指定されたカラムの値を、System.UInt64 を使用して設定します。
public void	Update() [470 ページ]	現在のカラム値 (set メソッドを使用して指定されます) で新しいローを更新します。
public void	UpdateBegin() [470 ページ]	現在のローを更新する準備を行います。

ULDataReader 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 override void	Close() [282 ページ]	カーソルを閉じます。
public override int	Depth [321 ページ]	現在のローのネストの深さを返します。
protected override void	Dispose(bool) [282 ページ]	
public override int	FieldCount [321 ページ]	このカーソル内のカラム数を返します。
public override unsafe bool	GetBoolean(int) [283 ページ]	指定されたカラムの値を System.Boolean として返します。
public override unsafe byte	GetByte(int) [284 ページ]	指定されたカラムの値を符号なし 8 ビット値 (System.Byte) として返します。
public unsafe byte[]	GetBytes(int) [285 ページ]	UL 拡張: 指定されたカラムの値を System.Bytes の値の配列として返します。

変数とタイプ	メンバー	説明
public override unsafe long	GetBytes(int, long, byte[], int, int) [286 ページ]	指定されたオフセットで始まる、指定された ULDbType.LongBinary カラムの値のサブセットを、コピー先の System.Byte 配列の指定されたオフセットにコピーします。
public override char	GetChar(int) [288 ページ]	このメソッドは Ultra Light.NET ではサポートされていません。
public override unsafe long	GetChars(int, long, char[], int, int) [289 ページ]	指定されたオフセットで始まる、指定された ULDbType.LongVarchar カラムの値のサブセットを、コピー先の System.Char 配列の指定されたオフセットにコピーします。
public override string	GetDataTypeName(int) [291 ページ]	指定されたカラムのプロバイダのデータ型の名前を返します。
public override unsafe DateTime	GetDateTime(int) [292 ページ]	指定されたカラムの値を、ミリ秒の精度の System.DateTime 型として返します。
protected override DbDataReader	GetDbDataReader(int) [293 ページ]	
public override decimal	GetDecimal(int) [293 ページ]	指定されたカラムの値を System.Decimal 型として返します。
public override unsafe double	GetDouble(int) [294 ページ]	指定されたカラムの値を System.Double 型として返します。
public override IEnumerable	GetEnumerator() [295 ページ]	ULDataReader オブジェクトの反復処理を実行する System.Collections.IEnumerable 値を返します。
public override Type	GetFieldType(int) [295 ページ]	指定されたカラムに最適な System.Type 値を返します。
public override unsafe float	GetFloat(int) [296 ページ]	指定されたカラムの値を System.Single 型として返します。
public override unsafe Guid	GetGuid(int) [297 ページ]	指定されたカラムの値を UUID (System.Guid) 型として返します。
public override unsafe short	GetInt16(int) [298 ページ]	指定されたカラムの値を System.Int16 型として返します。
public override unsafe int	GetInt32(int) [299 ページ]	指定されたカラムの値を System.Int32 型として返します。
public override unsafe long	GetInt64(int) [300 ページ]	指定されたカラムの値を System.Int64 型として返します。
public override string	GetName(int) [301 ページ]	指定されたカラムの名前を返します。
public override unsafe int	GetOrdinal(string) [302 ページ]	指定されたカラムのカラム ID を返します。
public unsafe int	GetRowCount(int) [303 ページ]	UL 拡張: カーソル内のローの数を、スレッシュホルド以内で返します。
public override DataTable	GetSchemaTable() [304 ページ]	ULDataReader オブジェクトのカラムのメタデータが記述された System.Data.DataTable 値を返します。

変更子とタイプ	メンバー	説明
public override unsafe String	GetString(int) [306 ページ]	指定されたカラムの値を System.String 型として返します。
public unsafe TimeSpan	GetTimeSpan(int) [307 ページ]	指定されたカラムの値を、ミリ秒の精度の System.TimeSpan 型として返します。
public unsafe ushort	GetUInt16(int) [308 ページ]	指定されたカラムの値を System.UInt16 型として返します。
public unsafe uint	GetUInt32(int) [309 ページ]	指定されたカラムの値を System.UInt32 型として返します。
public unsafe ulong	GetUInt64(int) [310 ページ]	指定されたカラムの値を System.UInt64 型として返します。
public override object	GetValue(int) [311 ページ]	指定されたカラムの値をネイティブフォーマットで返します。
public override int	GetValues(object[]) [312 ページ]	現在のローのすべてのカラム値を返します。
public override unsafe bool	HasRows [322 ページ]	ULDataReader オブジェクトに 1 つまたは複数のローがあるかどうかをチェックします。
public unsafe bool	IsBOF [322 ページ]	UL 拡張: 現在のローの位置が最初のローの前かどうかをチェックします。
public override bool	IsClosed [323 ページ]	カーソルが現在開いているかどうかを確認します。
public override unsafe bool	IsDBNull(int) [313 ページ]	指定されたカラムの値が NULL かどうかをチェックします。
public unsafe bool	IsEOF [323 ページ]	UL 拡張: 現在のローの位置が最後のローの後かどうかをチェックします。
public void	MoveAfterLast() [314 ページ]	UL 拡張: カーソルの最後のローの後に、カーソルを配置します。
public void	MoveBeforeFirst() [314 ページ]	UL 拡張: カーソルの最初のローの前に、カーソルを配置します。
public unsafe bool	MoveFirst() [315 ページ]	UL 拡張: カーソルの最初のローに、カーソルを配置します。
public unsafe bool	MoveLast() [315 ページ]	UL 拡張: カーソルの最後のローに、カーソルを配置します。
public unsafe bool	MoveNext() [316 ページ]	UL 拡張: カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。
public unsafe bool	MovePrevious() [317 ページ]	UL 拡張: カーソルを前のローに配置するか、最初のローの前に配置します。
public unsafe bool	MoveRelative(int) [317 ページ]	UL 拡張: 現在のローを基準としてカーソルを配置します。
public override bool	NextResult() [318 ページ]	バッチ SQL 文の結果を読み込むときに ULDataReader オブジェクトを次の結果に進めます。

変数とタイプ	メンバー	説明
public override bool	Read() [319 ページ]	カーソルを次のローに配置します。ただし、カーソルがすでに最後のローにある場合は最後のローの後に配置します。
public override int	RecordsAffected [324 ページ]	SQL 文の実行によって変更、挿入、または削除されたローの数を返します。
public int	RowCount [324 ページ]	UL 拡張: カーソル内のローの数を返します。
public ULCursorSchema	Schema [325 ページ]	UL 拡張: このカーソルのスキーマを保持します。
public override object	this[int colID] [326 ページ]	指定されたカラムの値をネイティブフォーマットで返します。
public override object	this[string name] [327 ページ]	指定された名前のカラムの値をネイティブフォーマットで返します。
protected void	Validate() [320 ページ]	
protected void	Validate(int) [320 ページ]	

備考

このクラスにはコンストラクタがありません。テーブルは、ULCommand.ExecuteTable メソッドを使用して作成されます。

このセクションの内容:

[DeleteAllRows\(\) メソッド](#) [532 ページ]

テーブルのすべてのローを削除します。

[FindBegin\(\) メソッド](#) [533 ページ]

テーブルで新規に検索を実行する準備を行います。

[FindFirst](#) メソッド [534 ページ]

テーブルを先頭から順方向に移動しながら、現在のインデックスの値かそのセット全体に完全に一致するローを検索します。

[FindLast](#) メソッド [537 ページ]

テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に完全に一致するローを検索します。

[FindNext](#) メソッド [540 ページ]

現在の位置からテーブルを順方向に移動しながら、次のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindFirst 検索を続行します。

[FindPrevious](#) メソッド [543 ページ]

現在の位置からテーブルを逆方向に移動しながら、前のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindLast 検索を続行します。

[Insert\(\)](#) メソッド [545 ページ]

現在のカラム値 (set メソッドを使用して指定されます) で新しいローを挿入します。

[InsertBegin\(\)](#) メソッド [546 ページ]

現在のすべてのカラムをデフォルト値に設定して、テーブルに新しいローを挿入する準備を行います。

[LookupBackward メソッド \[547 ページ\]](#)

テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より小さい値を持つローを検索します。

[LookupBegin\(\) メソッド \[550 ページ\]](#)

テーブルで新規に検索を実行する準備を行います。

[LookupForward メソッド \[551 ページ\]](#)

テーブルを最初から順方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より大きい値を持つローを検索します。

[Truncate\(\) メソッド \[554 ページ\]](#)

テーブル内のすべてのローを削除し、STOP SYNCHRONIZATION DELETE を一時的にアクティブにします。

[Schema プロパティ \[554 ページ\]](#)

テーブルスキーマを保持します。

関連情報

[ExecuteTable\(\) メソッド \[94 ページ\]](#)

[ULCommand クラス \[58 ページ\]](#)

[ULResultSet クラス \[439 ページ\]](#)

1.37.1 DeleteAllRows() メソッド

テーブルのすべてのローを削除します。

構文

Visual Basic

```
Public Sub DeleteAllRows ()
```

C#

```
public void DeleteAllRows ()
```

例外

ULException class SQL エラーが発生しました。

備考

アプリケーションによっては、テーブル内のローをすべて削除してから、新しいデータセットをテーブルにダウンロードする方が便利なことがあります。ULConnection.StopSynchronizationDelete メソッドを使用すると、統合データベースからは削除しないで Ultra Light データベースからローを削除できます。

関連情報

[Truncate\(\) メソッド \[554 ページ\]](#)

[StopSynchronizationDelete\(\) メソッド \[170 ページ\]](#)

1.37.2 FindBegin() メソッド

テーブルで新規に検索を実行する準備を行います。

構文

Visual Basic

```
Public Sub FindBegin ()
```

C#

```
public void FindBegin ()
```

例外

ULException class SQL エラーが発生しました。

備考

検索する値は、テーブルを開いたインデックス内のカラムで適切な setType メソッドを呼び出して指定します。

関連情報

[FindFirst\(\) メソッド \[534 ページ\]](#)

[FindFirst\(short\) メソッド \[535 ページ\]](#)

[FindLast\(\) メソッド \[537 ページ\]](#)

[FindLast\(short\) メソッド \[538 ページ\]](#)

1.37.3 FindFirst メソッド

テーブルを先頭から順方向に移動しながら、現在のインデックスの値かそのセット全体に完全に一致するローを検索します。

オーバードリスト

変数とタイプ	オーバーロード名	説明
public bool	FindFirst() [534 ページ]	テーブルを先頭から順方向に移動しながら、現在のインデックスの値かそのセット全体に完全に一致するローを検索します。
public unsafe bool	FindFirst(short) [535 ページ]	テーブルを先頭から順方向に移動しながら、現在のインデックスの値かそのセットの一部に完全に一致するローを検索します。

このセクションの内容:

[FindFirst\(\) メソッド \[534 ページ\]](#)

テーブルを先頭から順方向に移動しながら、現在のインデックスの値かそのセット全体に完全に一致するローを検索します。

[FindFirst\(short\) メソッド \[535 ページ\]](#)

テーブルを先頭から順方向に移動しながら、現在のインデックスの値かそのセットの一部に完全に一致するローを検索します。

1.37.3.1 FindFirst() メソッド

テーブルを先頭から順方向に移動しながら、現在のインデックスの値かそのセット全体に完全に一致するローを検索します。

構文

Visual Basic

```
Public Function FindFirst () As Boolean
```

C#

```
public bool FindFirst ()
```

戻り値

成功した場合は true、失敗した場合は false。

例外

ULException class SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値と完全に一致した最初のローで停止します。失敗した場合、カーソル位置は最後のローの後ろ (ULDataReader.IsEOF プロパティで示される) になります。

検索を行う前にそのつど FindBegin メソッドを呼び出す必要があります。

関連情報

[FindBegin\(\) メソッド \[533 ページ\]](#)

[FindNext\(\) メソッド \[540 ページ\]](#)

[FindPrevious\(\) メソッド \[543 ページ\]](#)

[FindFirst\(short\) メソッド \[535 ページ\]](#)

[IsEOF プロパティ \[323 ページ\]](#)

[FindBegin\(\) メソッド \[533 ページ\]](#)

1.37.3.2 FindFirst(short) メソッド

テーブルを先頭から順方向に移動しながら、現在のインデックスの値がそのセットの一部に完全に一致するローを検索します。

構文

Visual Basic

```
Public Function FindFirst (ByVal numColumns As Short) As Boolean
```

C#

```
public unsafe bool FindFirst (short numColumns)
```

パラメータ

numColumns 複合インデックスのための、検索で使用するカラムの数。たとえば、3つのカラムのインデックスがあり、最初のカラムにのみ基づいて一致する値を検索する場合は、最初のカラムに値を設定してから、値を1に指定します。

戻り値

成功した場合は true、失敗した場合は false。

例外

ULException class SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値と完全に一致した最初のローで停止します。失敗した場合、カーソル位置は最後のローの後ろ (ULDataReader.IsEOF プロパティで示される) になります。

検索を行う前にそのつど FindBegin メソッドを呼び出す必要があります。

関連情報

[FindBegin\(\) メソッド \[533 ページ\]](#)

[FindNext\(short\) メソッド \[541 ページ\]](#)

[FindPrevious\(short\) メソッド \[544 ページ\]](#)

[FindFirst\(\) メソッド \[534 ページ\]](#)

[IsEOF プロパティ \[323 ページ\]](#)

[FindBegin\(\) メソッド \[533 ページ\]](#)

1.37.4 FindLast メソッド

テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に完全に一致するローを検索します。

オーバーロードリスト

変数とタイプ	オーバーロード名	説明
public bool	FindLast() [537 ページ]	テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に完全に一致するローを検索します。
public unsafe bool	FindLast(short) [538 ページ]	テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセットの一部に完全に一致するローを検索します。

このセクションの内容:

[FindLast\(\) メソッド \[537 ページ\]](#)

テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に完全に一致するローを検索します。

[FindLast\(short\) メソッド \[538 ページ\]](#)

テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセットの一部に完全に一致するローを検索します。

1.37.4.1 FindLast() メソッド

テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に完全に一致するローを検索します。

構文

Visual Basic

```
Public Function FindLast () As Boolean
```

C#

```
public bool FindLast ()
```

戻り値

成功した場合は true、失敗した場合は false。

例外

ULException class SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値と完全に一致した最初のローで停止します。失敗した場合、カーソル位置は最初のローの前 (ULDataReader.IsEOF プロパティで示される) になります。

検索を行う前にそのつど FindBegin メソッドを呼び出す必要があります。

関連情報

[FindBegin\(\) メソッド \[533 ページ\]](#)

[FindNext\(\) メソッド \[540 ページ\]](#)

[FindPrevious\(\) メソッド \[543 ページ\]](#)

[FindLast\(short\) メソッド \[538 ページ\]](#)

[IsBOF プロパティ \[322 ページ\]](#)

[FindBegin\(\) メソッド \[533 ページ\]](#)

1.37.4.2 FindLast(short) メソッド

テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセットの一部に完全に一致するローを検索します。

構文

Visual Basic

```
Public Function FindLast (ByVal numColumns As Short) As Boolean
```

C#

```
public unsafe bool FindLast (short numColumns)
```

パラメータ

numColumns 複合インデックスのための、検索で使用するカラムの数。たとえば、3つのカラムのインデックスがあり、最初のカラムにのみ基づいて一致する値を検索する場合は、最初のカラムに値を設定してから、値を1に指定します。

戻り値

成功した場合は true、失敗した場合は false。

例外

ULException class SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値と完全に一致した最初のローで停止します。失敗した場合、カーソル位置は最初のローの前 (ULDataReader.IsEOF プロパティで示される) になります。

検索を行う前にそのつど FindBegin メソッドを呼び出す必要があります。

関連情報

[FindBegin\(\) メソッド \[533 ページ\]](#)

[FindNext\(short\) メソッド \[541 ページ\]](#)

[FindPrevious\(short\) メソッド \[544 ページ\]](#)

[FindLast\(\) メソッド \[537 ページ\]](#)

[IsBOF プロパティ \[322 ページ\]](#)

[FindBegin\(\) メソッド \[533 ページ\]](#)

1.37.5 FindNext メソッド

現在の位置からテーブルを順方向に移動しながら、次のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindFirst 検索を続行します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public bool	FindNext() [540 ページ]	現在の位置からテーブルを順方向に移動しながら、次のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindFirst 検索を続行します。
public unsafe bool	FindNext(short) [541 ページ]	現在の位置からテーブルを順方向に移動しながら、次のローが現在のインデックスの値かその部分的なセットに完全に一致するかどうかを調べて、ULTable.FindFirst 検索を続行します。

このセクションの内容:

[FindNext\(\) メソッド](#) [540 ページ]

現在の位置からテーブルを順方向に移動しながら、次のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindFirst 検索を続行します。

[FindNext\(short\) メソッド](#) [541 ページ]

現在の位置からテーブルを順方向に移動しながら、次のローが現在のインデックスの値かその部分的なセットに完全に一致するかどうかを調べて、ULTable.FindFirst 検索を続行します。

1.37.5.1 FindNext() メソッド

現在の位置からテーブルを順方向に移動しながら、次のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindFirst 検索を続行します。

構文

Visual Basic

```
Public Function FindNext () As Boolean
```

C#

```
public bool FindNext ()
```

戻り値

成功した場合は true、失敗した場合は false。

例外

ULException class SQL エラーが発生しました。

備考

インデックスの値と完全に一致すると、カーソルは次のローで停止します。失敗した場合、カーソル位置は最後のローの後ろ (ULDataReader.IsEOF プロパティで示される) になります。

検索されるカラムの値がローの更新において修正された場合の FindNext メソッドの動作は不確定です。

関連情報

[FindFirst\(\) メソッド \[534 ページ\]](#)

[FindNext\(short\) メソッド \[541 ページ\]](#)

[IsEOF プロパティ \[323 ページ\]](#)

1.37.5.2 FindNext(short) メソッド

現在の位置からテーブルを順方向に移動しながら、次のローが現在のインデックスの値かその部分的なセットに完全に一致するかどうかを調べて、ULTable.FindFirst 検索を続行します。

構文

Visual Basic

```
Public Function FindNext (ByVal numColumns As Short) As Boolean
```

C#

```
public unsafe bool FindNext (short numColumns)
```

パラメータ

numColumns 複合インデックスのための、検索で使用するカラムの数。たとえば、3つのカラムのインデックスがあり、最初のカラムにのみ基づいて一致する値を検索する場合は、最初のカラムに値を設定してから、値を1に指定します。

戻り値

成功した場合は true、失敗した場合は false。

例外

ULException class SQL エラーが発生しました。

備考

インデックスの値と完全に一致すると、カーソルは次のローで停止します。失敗した場合、カーソル位置は最後のローの後ろ (ULDataReader.IsEOF プロパティで示される) になります。

検索されるカラムの値がローの更新において修正された場合の FindNext メソッドの動作は不確定です。

関連情報

[FindFirst\(short\) メソッド \[535 ページ\]](#)

[FindNext\(\) メソッド \[540 ページ\]](#)

[FindFirst\(\) メソッド \[534 ページ\]](#)

[IsEOF プロパティ \[323 ページ\]](#)

1.37.6 FindPrevious メソッド

現在の位置からテーブルを逆方向に移動しながら、前のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindLast 検索を続行します。

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	FindPrevious() [543 ページ]	現在の位置からテーブルを逆方向に移動しながら、前のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindLast 検索を続行します。
public unsafe bool	FindPrevious(short) [544 ページ]	現在の位置からテーブルを逆方向に移動しながら、前のローが現在のインデックスの値かその部分的なセットに完全に一致するかどうかを調べて、ULTable.FindLast 検索を続行します。

このセクションの内容:

[FindPrevious\(\) メソッド \[543 ページ\]](#)

現在の位置からテーブルを逆方向に移動しながら、前のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindLast 検索を続行します。

[FindPrevious\(short\) メソッド \[544 ページ\]](#)

現在の位置からテーブルを逆方向に移動しながら、前のローが現在のインデックスの値かその部分的なセットに完全に一致するかどうかを調べて、ULTable.FindLast 検索を続行します。

1.37.6.1 FindPrevious() メソッド

現在の位置からテーブルを逆方向に移動しながら、前のローが現在のインデックスの値かそのセット全体に完全に一致するかどうかを調べて、ULTable.FindLast 検索を続行します。

構文

Visual Basic

```
Public Function FindPrevious () As Boolean
```

C#

```
public bool FindPrevious ()
```

戻り値

成功した場合は true、失敗した場合は false。

例外

ULException class SQL エラーが発生しました。

備考

インデックスの値と完全に一致すると、カーソルは前のローで停止します。失敗した場合、カーソル位置は最初のローの前 (ULDataReader.IsEOF プロパティで示される) になります。

検索されるカラムの値がローの更新において修正された場合の FindPrevious メソッドの動作は不確定です。

関連情報

[FindLast\(\) メソッド \[537 ページ\]](#)

[FindPrevious\(short\) メソッド \[544 ページ\]](#)

[IsBOF プロパティ \[322 ページ\]](#)

1.37.6.2 FindPrevious(short) メソッド

現在の位置からテーブルを逆方向に移動しながら、前のローが現在のインデックスの値かその部分的なセットに完全に一致するかどうかを調べて、ULTable.FindLast 検索を続行します。

構文

Visual Basic

```
Public Function FindPrevious (ByVal numColumns As Short) As Boolean
```

C#

```
public unsafe bool FindPrevious (short numColumns)
```


パラメータ

numColumns 複合インデックスのための、検索で使用するカラムの数。たとえば、3つのカラムのインデックスがあり、最初のカラムにのみ基づいて一致する値を検索する場合は、最初のカラムに値を設定してから、値を1に指定します。

戻り値

成功した場合は true、失敗した場合は false。

例外

ULException class SQL エラーが発生しました。

備考

インデックスの値と完全に一致すると、カーソルは前のローで停止します。失敗した場合、カーソル位置は最初のローの前 (ULDataReader.IsEOF プロパティで示される) になります。

検索されるカラムの値がローの更新において修正された場合の FindPrevious メソッドの動作は不確定です。

関連情報

[FindLast\(\) メソッド \[537 ページ\]](#)

[FindLast\(short\) メソッド \[538 ページ\]](#)

[FindPrevious\(\) メソッド \[543 ページ\]](#)

[IsBOF プロパティ \[322 ページ\]](#)

1.37.7 Insert() メソッド

現在のカラム値 (set メソッドを使用して指定されます) で新しいローを挿入します。

構文

Visual Basic

```
Public Sub Insert ()
```

C#

```
public void Insert ()
```

例外

ULException class SQL エラーが発生しました。

備考

挿入を行う前に `ULTable.InsertBegin` メソッドを呼び出してください。

関連情報

[InsertBegin\(\) メソッド \[546 ページ\]](#)

1.37.8 InsertBegin() メソッド

現在のすべてのカラムをデフォルト値に設定して、テーブルに新しいローを挿入する準備を行います。

構文

Visual Basic

```
Public Sub InsertBegin ()
```

C#

```
public void InsertBegin ()
```

例外

ULException class SQL エラーが発生しました。

備考

適切な SetType メソッドまたは AppendType メソッドを呼び出して、挿入するデフォルト以外の値を指定します。

insert メソッドが実行されないと、ローが実際に挿入されることも、ロー内のデータが実際に変更されることもありません。また、その変更がコミットされないかぎり、永続化されません。

関連情報

[Insert\(\) メソッド \[545 ページ\]](#)

[Insert\(\) メソッド \[545 ページ\]](#)

1.37.9 LookupBackward メソッド

テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より小さい値を持つローを検索します。

オーバーロードリスト

変更子とタイプ	オーバーロード名	説明
public bool	LookupBackward() [548 ページ]	テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より小さい値を持つローを検索します。
public unsafe bool	LookupBackward(short) [549 ページ]	テーブルを最後から逆方向に移動しながら、現在のインデックス内の特定の値または値の部分的セットに一致するか、それより小さい値を持つローを検索します。

このセクションの内容:

[LookupBackward\(\) メソッド \[548 ページ\]](#)

テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より小さい値を持つローを検索します。

[LookupBackward\(short\) メソッド \[549 ページ\]](#)

テーブルを最後から逆方向に移動しながら、現在のインデックス内の特定の値または値の部分的セットに一致するか、それより小さい値を持つローを検索します。

1.37.9.1 LookupBackward() メソッド

テーブルを最後から逆方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より小さい値を持つローを検索します。

構文

Visual Basic

```
Public Function LookupBackward () As Boolean
```

C#

```
public bool LookupBackward ()
```

戻り値

成功した場合は true、失敗した場合は false。

例外

ULException class SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値に一致する最初のローか、それより少ない値の最初のローで停止します。失敗した場合（検索する値より小さい値のローがない場合）、カーソル位置は最初のローの前 (ULDataReader.IsBOF プロパティで示される) になります。

検索を行う前にそのつど LookupBegin メソッドを呼び出す必要があります。

関連情報

[LookupBegin\(\) メソッド \[550 ページ\]](#)

[LookupBackward\(short\) メソッド \[549 ページ\]](#)

[IsBOF プロパティ \[322 ページ\]](#)

1.37.9.2 LookupBackward(short) メソッド

テーブルを最後から逆方向に移動しながら、現在のインデックス内の特定の値または値の部分的セットに一致するか、それより小さい値を持つローを検索します。

構文

Visual Basic

```
Public Function LookupBackward (ByVal numColumns As Short) As Boolean
```

C#

```
public unsafe bool LookupBackward (short numColumns)
```

パラメータ

numColumns 複合インデックスのための、ルックアップで使用するカラムの数。たとえば、3つのカラムのインデックスがあり、最初のカラムにのみ基づいて一致する値を検索する場合は、最初のカラムに値を設定してから、値を1に指定します。

戻り値

成功した場合は true、失敗した場合は false。

例外

ULException class SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値に一致する最初のローか、それより少ない値の最初のローで停止します。失敗した場合（検索する値より小さい値のローがない場合）、カーソル位置は最初のローの前 (ULDataReader.IsBOF プロパティで示される) になります。

検索を行う前にそのつど LookupBegin メソッドを呼び出す必要があります。

関連情報

[LookupBegin\(\) メソッド \[550 ページ\]](#)

[IsBOF プロパティ \[322 ページ\]](#)

1.37.10 LookupBegin() メソッド

テーブルで新規に検索を実行する準備を行います。

構文

Visual Basic

```
Public Sub LookupBegin ()
```

C#

```
public void LookupBegin ()
```

例外

ULException class SQL エラーが発生しました。

備考

検索する値は、テーブルを開いたインデックス内のコラムで適切な setType メソッドを呼び出して指定します。

関連情報

[LookupForward\(\) メソッド \[551 ページ\]](#)

[LookupForward\(short\) メソッド \[552 ページ\]](#)

[LookupBackward\(\) メソッド \[548 ページ\]](#)

[LookupBackward\(short\) メソッド \[549 ページ\]](#)

1.37.11 LookupForward メソッド

テーブルを最初から順方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より大きい値を持つローを検索します。

オーバーロードリスト

変数とタイプ	オーバーロード名	説明
public bool	LookupForward() [551 ページ]	テーブルを最初から順方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より大きい値を持つローを検索します。
public unsafe bool	LookupForward(short) [552 ページ]	テーブルを最初から順方向に移動しながら、現在のインデックスの値またはそのセットの一部に一致するか、その値より大きい値を持つローを検索します。

このセクションの内容:

[LookupForward\(\) メソッド \[551 ページ\]](#)

テーブルを最初から順方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より大きい値を持つローを検索します。

[LookupForward\(short\) メソッド \[552 ページ\]](#)

テーブルを最初から順方向に移動しながら、現在のインデックスの値またはそのセットの一部に一致するか、その値より大きい値を持つローを検索します。

1.37.11.1 LookupForward() メソッド

テーブルを最初から順方向に移動しながら、現在のインデックスの値またはそのセット全体に一致するか、その値より大きい値を持つローを検索します。

構文

Visual Basic

```
Public Function LookupForward () As Boolean
```

C#

```
public bool LookupForward ()
```

戻り値

成功した場合は true、失敗した場合は false。

例外

InvalidOperationException class SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値に一致するか、それより大きい値の最初のローで停止します。失敗した場合（検索する値より大きい値のローがない場合）、カーソル位置は最後のローの後ろ (ULDataReader.IsEOF プロパティで示される) になります。

検索を行う前にそのつど LookupBegin メソッドを呼び出す必要があります。

関連情報

[LookupBegin\(\) メソッド \[550 ページ\]](#)

[LookupForward\(short\) メソッド \[552 ページ\]](#)

[IsEOF プロパティ \[323 ページ\]](#)

1.37.11.2 LookupForward(short) メソッド

テーブルを最初から順方向に移動しながら、現在のインデックスの値またはそのセットの一部に一致するか、その値より大きい値を持つローを検索します。

構文

Visual Basic

```
Public Function LookupForward (ByVal numColumns As Short) As Boolean
```

C#

```
public unsafe bool LookupForward (short numColumns)
```


パラメータ

numColumns 複合インデックスのための、ルックアップで使用するカラムの数。たとえば、3つのカラムのインデックスがあり、最初のカラムにのみ基づいて一致する値を検索する場合は、最初のカラムに値を設定してから、値を1に指定します。

戻り値

成功した場合は true、失敗した場合は false。

例外

ULException class SQL エラーが発生しました。

備考

検索する値を指定するには、インデックスの各カラムに値を設定します。カーソルは、インデックスの値に一致するか、それより大きい値の最初のローで停止します。失敗した場合（検索する値より大きい値のローがない場合）、カーソル位置は最後のローの後ろ (ULDataReader.IsEOF プロパティで示される) になります。

検索を行う前にそのつど LookupBegin メソッドを呼び出す必要があります。

関連情報

[LookupBegin\(\) メソッド \[550 ページ\]](#)

[LookupForward\(\) メソッド \[551 ページ\]](#)

[IsEOF プロパティ \[323 ページ\]](#)

[LookupBegin\(\) メソッド \[550 ページ\]](#)

1.37.12 Truncate() メソッド

テーブル内のすべてのローを削除し、STOP SYNCHRONIZATION DELETE を一時的にアクティブにします。

構文

Visual Basic

```
Public Sub Truncate ()
```

C#

```
public void Truncate ()
```

例外

ULException class SQL エラーが発生しました。

関連情報

[DeleteAllRows\(\) メソッド \[532 ページ\]](#)

1.37.13 Schema プロパティ

テーブルスキーマを保持します。

構文

Visual Basic

```
Public ReadOnly Shadows Property Schema As ULTableSchema
```

C#

```
public new ULTableSchema Schema {get;}
```

備考

このプロパティが有効なのは、接続が開かれている間だけです。

テーブルスキーマを表す ULTableSchema オブジェクト。

このプロパティは、ULDataReader.GetSchemaTable メソッドからの結果に示されない Ultra Light.NET の詳細情報を含め、テーブルの完全なスキーマを表します。

関連情報

[ULTableSchema クラス \[555 ページ\]](#)

1.38 ULTableSchema クラス

UL 拡張: Ultra Light のテーブルのスキーマを表します。

構文

Visual Basic

```
Public NotInheritable Class ULTableSchema Inherits ULCursorSchema
```

C#

```
public sealed class ULTableSchema : ULCursorSchema
```

メンバー

ULTableSchema 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 unsafe string	GetColumnDefaultValue(int) [559 ページ]	指定されたカラムのデフォルト値を返します。
public unsafe ulong	GetColumnPartitionSize(int) [560 ページ]	指定されたカラムに割り当てられているグローバルオートインクリメントの分割サイズを返します。
public unsafe ULIndexSchema	GetIndex(string) [561 ページ]	指定されたインデックスのインデックススキーマを返します。
public unsafe string	GetIndexName(int) [562 ページ]	指定されたインデックス ID で識別されたインデックスの名前を返します。
public unsafe string	GetOptimalIndex(int) [563 ページ]	指定されたカラムを使用してテーブルを検索するために最適なインデックスです。

変更子とタイプ	メソッド	説明
public unsafe string	GetPublicationPredicate(string) [564 ページ]	指定されたパブリケーションに含まれているテーブルのパブリケーション述部を返します。
public unsafe bool	IsColumnAutoIncrement(int) [564 ページ]	指定されたカラムのデフォルトがオートインクリメントに設定されているかどうかをチェックします。
public unsafe bool	IsColumnCurrentDate(int) [565 ページ]	指定されたカラムのデフォルトが、現在の日付 (ULDbType.Date 値) に設定されているかどうかをチェックします。
public unsafe bool	IsColumnCurrentTime(int) [566 ページ]	指定されたカラムのデフォルトが、現在の時刻 (ULDbType.Time 値) に設定されているかどうかをチェックします。
public unsafe bool	IsColumnCurrentTimestamp(int) [567 ページ]	指定されたカラムのデフォルトが、現在のタイムスタンプ (ULDbType.TimeStamp 値) に設定されているかどうかをチェックします。
public unsafe bool	IsColumnCurrentUTCTimestamp(int) [568 ページ]	指定されたカラムのデフォルトが、現在の UTC タイムスタンプ (ULDbType.TimeStamp 値) に設定されているかどうかをチェックします。
public unsafe bool	IsColumnGlobalAutoIncrement(int) [569 ページ]	指定されたカラムのデフォルトがグローバルオートインクリメントに設定されているかどうかをチェックします。
public unsafe bool	IsColumnNewUUID(int) [570 ページ]	指定されたカラムのデフォルトが新しい UUID (System.Guid 値) に設定されているかどうかをチェックします。
public unsafe bool	IsColumnNullable(int) [571 ページ]	指定されたカラムが NULL 入力可であるかどうかをチェックします。
public unsafe bool	IsInPublication(string) [572 ページ]	テーブルが、指定されたパブリケーションに含まれているかどうかをチェックします。
protected virtual override void	VerifyOpen() [572 ページ]	

プロパティ

変更子とタイプ	プロパティ	説明
public unsafe int	IndexCount [573 ページ]	テーブルのインデックス数を返します。
public unsafe bool	IsNeverSynchronized [573 ページ]	テーブルが、まったく同期されないようにマーク付けされているかどうかをチェックします。
public override string	Name [574 ページ]	テーブルの名前を返します。
public unsafe ULIndexSchema	PrimaryKey [574 ページ]	テーブルのプライマリキーのインデックススキーマを返します。
public unsafe bool	UploadUnchangedRows [575 ページ]	データベースが、変更されていないローをアップロードするかどうかをチェックします。

ULCursorSchema から継承されたメンバー

変数とタイプ	メンバー	説明
public short	ColumnCount [237 ページ]	このカーソル内のカラム数を返します。
protected unsafe void	GetColumnCount() [229 ページ]	
public unsafe short	GetColumnID(string) [229 ページ]	指定されたカラムのカラム ID を返します。
public string	GetColumnName(int) [230 ページ]	指定されたカラム ID で識別されたカラムの名前を返します。
public unsafe int	GetColumnPrecision(int) [231 ページ]	カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの精度を返します。
public unsafe int	GetColumnScale(int) [232 ページ]	カラムが数値カラム (NUMERIC SQL 型) の場合は、指定されたカラム ID によって識別されたカラムの位取りを返します。
public unsafe int	GetColumnSize(int) [233 ページ]	カラムがサイズ指定されたカラム (BINARY SQL 型または CHAR SQL 型) の場合は、指定されたカラム ID によって識別されたカラムのサイズを返します。
public string	GetColumnSQLName(int) [234 ページ]	指定されたカラム ID で識別されたカラムの名前を返します。
public unsafe ULDbType	GetColumnULDbType(int) [235 ページ]	指定されたカラム ID によって識別されたカラムの Ultra Light.NET データ型を返します。
public unsafe DataTable	GetSchemaTable() [236 ページ]	ULDataReader オブジェクトのカラムのスキーマが記述された System.Data.DataTable を返します。
public bool	IsOpen [237 ページ]	カーソルのスキーマが現在開いているかどうかを確認します。

備考

このクラスにはコンストラクタがありません。ULTableSchema オブジェクトは、その ULTable.Schema プロパティとしてテーブルにアタッチされます。

このセクションの内容:

[GetColumnDefaultValue\(int\) メソッド \[559 ページ\]](#)

指定されたカラムのデフォルト値を返します。

[GetColumnPartitionSize\(int\) メソッド \[560 ページ\]](#)

指定されたカラムに割り当てられているグローバルオートインクリメントの分割サイズを返します。

[GetIndex\(string\) メソッド \[561 ページ\]](#)

指定されたインデックスのインデックススキーマを返します。

[GetIndexName\(int\) メソッド \[562 ページ\]](#)

指定されたインデックス ID で識別されたインデックスの名前を返します。

[GetOptimallIndex\(int\) メソッド \[563 ページ\]](#)

指定されたカラムを使用してテーブルを検索するために最適なインデックスです。

[GetPublicationPredicate\(string\) メソッド \[564 ページ\]](#)

指定されたパブリケーションに含まれているテーブルのパブリケーション述部を返します。

[IsColumnAutoIncrement\(int\) メソッド \[564 ページ\]](#)

指定されたカラムのデフォルトがオートインクリメントに設定されているかどうかをチェックします。

[IsColumnCurrentDate\(int\) メソッド \[565 ページ\]](#)

指定されたカラムのデフォルトが、現在の日付 (ULDbType.Date 値) に設定されているかどうかをチェックします。

[IsColumnCurrentTime\(int\) メソッド \[566 ページ\]](#)

指定されたカラムのデフォルトが、現在の時刻 (ULDbType.Time 値) に設定されているかどうかをチェックします。

[IsColumnCurrentTimestamp\(int\) メソッド \[567 ページ\]](#)

指定されたカラムのデフォルトが、現在のタイムスタンプ (ULDbType.TimeStamp 値) に設定されているかどうかをチェックします。

[IsColumnCurrentUTCTimestamp\(int\) メソッド \[568 ページ\]](#)

指定されたカラムのデフォルトが、現在の UTC タイムスタンプ (ULDbType.TimeStamp 値) に設定されているかどうかをチェックします。

[IsColumnGlobalAutoIncrement\(int\) メソッド \[569 ページ\]](#)

指定されたカラムのデフォルトがグローバルオートインクリメントに設定されているかどうかをチェックします。

[IsColumnNewUUID\(int\) メソッド \[570 ページ\]](#)

指定されたカラムのデフォルトが新しい UUID (System.Guid 値) に設定されているかどうかをチェックします。

[IsColumnNullable\(int\) メソッド \[571 ページ\]](#)

指定されたカラムが NULL 入力可であるかどうかをチェックします。

[IsInPublication\(string\) メソッド \[572 ページ\]](#)

テーブルが、指定されたパブリケーションに含まれているかどうかをチェックします。

[VerifyOpen\(\) メソッド \[572 ページ\]](#)

[IndexCount プロパティ \[573 ページ\]](#)

テーブルのインデックス数を返します。

[IsNeverSynchronized プロパティ \[573 ページ\]](#)

テーブルが、まったく同期されないようにマーク付けされているかどうかをチェックします。

[Name プロパティ \[574 ページ\]](#)

テーブルの名前を返します。

[PrimaryKey プロパティ \[574 ページ\]](#)

テーブルのプライマリキーのインデックススキーマを返します。

[UploadUnchangedRows プロパティ \[575 ページ\]](#)

データベースが、変更されていないローをアップロードするかどうかをチェックします。

関連情報

[Schema プロパティ \[554 ページ\]](#)

[ULCursorSchema クラス \[226 ページ\]](#)

1.38.1 GetColumnDefaultValue(int) メソッド

指定されたカラムのデフォルト値を返します。

構文

Visual Basic

```
Public Function GetColumnDefaultValue (ByVal columnID As Integer) As String
```

C#

```
public unsafe string GetColumnDefaultValue (int columnID)
```

パラメータ

columnID カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内である必要があります。テーブルの先頭カラムの ID 値は 0 です。

戻り値

指定されたカラムの文字列としてのデフォルト値。デフォルト値が NULL の場合は NULL 参照 (Visual Basic の Nothing)。

例外

ULException class SQL エラーが発生しました。

関連情報

[ColumnCount プロパティ \[237 ページ\]](#)

1.38.2 GetColumnPartitionSize(int) メソッド

指定されたカラムに割り当てられているグローバルオートインクリメントの分割サイズを返します。

構文

Visual Basic

```
Public Function GetColumnPartitionSize (ByVal columnID As Integer) As ULong
```

C#

```
public unsafe ulong GetColumnPartitionSize (int columnID)
```

パラメータ

columnID カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内である必要があります。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムのグローバルオートインクリメントの分割サイズ (System.UInt64 構造体)。

例外

ULException class SQL エラーが発生しました。

備考

テーブルのすべてのグローバルオートインクリメントカラムは、同じグローバルオートインクリメントの分割サイズを共有します。

関連情報

[IsColumnGlobalAutoIncrement\(int\) メソッド \[569 ページ\]](#)

[ColumnCount プロパティ \[237 ページ\]](#)

1.38.3 GetIndex(string) メソッド

指定されたインデックスのインデックススキーマを返します。

構文

Visual Basic

```
Public Function GetIndex (ByVal name As String) As ULIndexSchema
```

C#

```
public unsafe ULIndexSchema GetIndex (string name)
```

パラメータ

name インデックスの名前。

戻り値

指定されたインデックスを表す ULIndexSchema オブジェクト。

例外

ULException class SQL エラーが発生しました。

関連情報

[ULIndexSchema クラス \[363 ページ\]](#)

1.38.4 GetIndexName(int) メソッド

指定されたインデックス ID で識別されたインデックスの名前を返します。

構文

Visual Basic

```
Public Function GetIndexName (ByVal indexID As Integer) As String
```

C#

```
public unsafe string GetIndexName (int indexID)
```

パラメータ

indexID インデックスの ID。値は、[1,IndexCount] の範囲内である必要があります。

戻り値

文字列として返されるインデックス名。

例外

ULException class SQL エラーが発生しました。

備考

インデックスの ID とカウントは、スキーマのアップグレード中に変更されることがあります。インデックスを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後に再表示します。

関連情報

[IndexCount プロパティ \[573 ページ\]](#)

1.38.5 GetOptimalIndex(int) メソッド

指定されたカラムを使用してテーブルを検索するために最適なインデックスです。

構文

Visual Basic

```
Public Function GetOptimalIndex (ByVal columnID As Integer) As String
```

C#

```
public unsafe string GetOptimalIndex (int columnID)
```

パラメータ

columnID カラムの ID 番号。テーブルの先頭カラムの ID 値は 0 です。

戻り値

指定されたカラムの最適なインデックスを表す `ULIndexSchema` オブジェクト。

例外

ULException class SQL エラーが発生しました。

備考

指定されたカラムは、インデックス内の最初のカラムですが、インデックスには複数のカラムがある場合があります。

関連情報

[ColumnCount プロパティ \[237 ページ\]](#)

[ULIndexSchema クラス \[363 ページ\]](#)

1.38.6 GetPublicationPredicate(string) メソッド

指定されたパブリケーションに含まれているテーブルのパブリケーション述部を返します。

構文

Visual Basic

```
Public Function GetPublicationPredicate (ByVal pubName As String) As String
```

C#

```
public unsafe string GetPublicationPredicate (string pubName)
```

パラメータ

pubName パブリケーションの名前。

戻り値

文字列として返されるパブリケーション述部。

例外

ULException class SQL エラーが発生しました。

1.38.7 IsColumnAutoIncrement(int) メソッド

指定されたカラムのデフォルトがオートインクリメントに設定されているかどうかをチェックします。

構文

Visual Basic

```
Public Function IsColumnAutoIncrement (ByVal columnID As Integer) As Boolean
```

C#

```
public unsafe bool IsColumnAutoIncrement (int columnID)
```

パラメータ

columnID カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内であることが必要です。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムがオートインクリメントされる場合は true、オートインクリメントされない場合は false。

例外

ULException class SQL エラーが発生しました。

関連情報

[ColumnCount プロパティ \[237 ページ\]](#)

1.38.8 IsColumnCurrentDate(int) メソッド

指定されたカラムのデフォルトが、現在の日付 (ULDbType.Date 値) に設定されているかどうかをチェックします。

構文

Visual Basic

```
Public Function IsColumnCurrentDate (ByVal columnID As Integer) As Boolean
```

C#

```
public unsafe bool IsColumnCurrentDate (int columnID)
```

パラメータ

columnID カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内であることが必要です。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムのデフォルトが現在の日付に設定されている場合は true、現在の日付に設定されていない場合は false。

例外

ULException class SQL エラーが発生しました。

関連情報

[ColumnCount プロパティ \[237 ページ\]](#)

1.38.9 IsColumnCurrentTime(int) メソッド

指定されたカラムのデフォルトが、現在の時刻 (ULDbType.Time 値) に設定されているかどうかをチェックします。

構文

Visual Basic

```
Public Function IsColumnCurrentTime (ByVal columnID As Integer) As Boolean
```

C#

```
public unsafe bool IsColumnCurrentTime (int columnID)
```

パラメータ

columnID カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内である必要があります。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムのデフォルトが現在の時刻に設定されている場合は true、現在の時刻に設定されていない場合は false。

例外

ULException class SQL エラーが発生しました。

関連情報

[ColumnCount プロパティ \[237 ページ\]](#)

1.38.10 IsColumnCurrentTimestamp(int) メソッド

指定されたカラムのデフォルトが、現在のタイムスタンプ (ULDbType.TimeStamp 値) に設定されているかどうかをチェックします。

構文

Visual Basic

```
Public Function IsColumnCurrentTimestamp (ByVal columnID As Integer) As Boolean
```

C#

```
public unsafe bool IsColumnCurrentTimestamp (int columnID)
```

パラメータ

columnID カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内である必要があります。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムのデフォルトが現在のタイムスタンプに設定されている場合は true、現在のタイムスタンプに設定されていない場合は false。

例外

ULException class SQL エラーが発生しました。

関連情報

[ColumnCount プロパティ \[237 ページ\]](#)

1.38.11 IsColumnCurrentUTCTimestamp(int) メソッド

指定されたカラムのデフォルトが、現在の UTC タイムスタンプ (ULDbType.TimeStamp 値) に設定されているかどうかをチェックします。

構文

Visual Basic

```
Public Function IsColumnCurrentUTCTimestamp (ByVal columnID As Integer) As Boolean
```

C#

```
public unsafe bool IsColumnCurrentUTCTimestamp (int columnID)
```

パラメータ

columnID カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内である必要があります。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムのデフォルトが現在の UTC タイムスタンプに設定されている場合は true、現在のタイムスタンプに設定されていない場合は false。

例外

ULException class SQL エラーが発生しました。

関連情報

[ColumnCount プロパティ \[237 ページ\]](#)

1.38.12 IsColumnGlobalAutoIncrement(int) メソッド

指定されたカラムのデフォルトがグローバルオートインクリメントに設定されているかどうかをチェックします。

構文

Visual Basic

```
Public Function IsColumnGlobalAutoIncrement (ByVal columnID As Integer) As Boolean
```

C#

```
public unsafe bool IsColumnGlobalAutoIncrement (int columnID)
```

パラメータ

columnID カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内である必要があります。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムがグローバルオートインクリメントされる場合は true、グローバルオートインクリメントされない場合は false。

例外

ULException class SQL エラーが発生しました。

関連情報

[GetColumnPartitionSize\(int\) メソッド \[560 ページ\]](#)

[DatabaseID プロパティ \[179 ページ\]](#)

[ColumnCount プロパティ \[237 ページ\]](#)

1.38.13 IsColumnNewUUID(int) メソッド

指定されたカラムのデフォルトが新しい UUID (System.Guid 値) に設定されているかどうかをチェックします。

構文

Visual Basic

```
Public Function IsColumnNewUUID (ByVal columnID As Integer) As Boolean
```

C#

```
public unsafe bool IsColumnNewUUID (int columnID)
```

パラメータ

columnID カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内である必要があります。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムのデフォルトが新しい UUID に設定されている場合は true、新しい UUID に設定されていない場合は false。

例外

ULException class SQL エラーが発生しました。

関連情報

[ColumnCount プロパティ \[237 ページ\]](#)

1.38.14 IsColumnNullable(int) メソッド

指定されたカラムが NULL 入力可であるかどうかをチェックします。

構文

Visual Basic

```
Public Function IsColumnNullable (ByVal columnID As Integer) As Boolean
```

C#

```
public unsafe bool IsColumnNullable (int columnID)
```

パラメータ

columnID カラムの ID 番号。値は、[0,ULCursorSchema.ColumnCount-1] の範囲内である必要があります。テーブルの先頭カラムの ID 値は 0 です。

戻り値

カラムが NULL 入力可の場合は true、NULL 入力不可の場合は false。

例外

ULException class SQL エラーが発生しました。

関連情報

[ColumnCount プロパティ \[237 ページ\]](#)

1.38.15 IsInPublication(string) メソッド

テーブルが、指定されたパブリケーションに含まれているかどうかをチェックします。

構文

Visual Basic

```
Public Function IsInPublication (ByVal pubName As String) As Boolean
```

C#

```
public unsafe bool IsInPublication (string pubName)
```

パラメータ

pubName パブリケーションの名前。

戻り値

テーブルがパブリケーション内にある場合は true、パブリケーション内にない場合は false。

例外

ULException class SQL エラーが発生しました。

1.38.16 VerifyOpen() メソッド

構文

Visual Basic

```
Protected Overridable Overrides Sub VerifyOpen ()
```

C#

```
protected virtual override void VerifyOpen ()
```

1.38.17 IndexCount プロパティ

テーブルのインデックス数を返します。

構文

Visual Basic

```
Public ReadOnly Property IndexCount As Integer
```

C#

```
public unsafe int IndexCount {get;}
```

備考

テーブルのインデックスの数。テーブルスキーマが閉じている場合は 0。

インデックス ID の範囲は、1 ~ IndexCount 値 (両端の値を含む) です。

i 注記

インデックスの ID とカウントは、スキーマのアップグレード中に変更されることがあります。インデックスを正しく識別するには、名前でアクセスするか、キャッシュされている ID とカウントをスキーマのアップグレード後に再表示します。

1.38.18 IsNeverSynchronized プロパティ

テーブルが、まったく同期されないようにマーク付けされているかどうかをチェックします。

構文

Visual Basic

```
Public ReadOnly Property IsNeverSynchronized As Boolean
```

C#

```
public unsafe bool IsNeverSynchronized {get;}
```

備考

テーブルが、まったく同期されないようにマーク付けされている場合は true、そうでない場合は false。

まったく同期されないようにマーク付けされているテーブルは、パブリケーションに含まれているものであっても、まったく同期されていません。このようなテーブルは、「非同期」テーブルと呼ばれることもあります。

1.38.19 Name プロパティ

テーブルの名前を返します。

構文

Visual Basic

```
Public ReadOnly Overrides Property Name As String
```

C#

```
public override string Name {get;}
```

備考

文字列としてのテーブル名。

1.38.20 PrimaryKey プロパティ

テーブルのプライマリキーのインデックススキーマを返します。

構文

Visual Basic

```
Public ReadOnly Property PrimaryKey As ULIndexSchema
```

C#

```
public unsafe ULIndexSchema PrimaryKey {get;}
```

備考

テーブルのプライマリキーを表す ULIndexSchema オブジェクト。

関連情報

[ULIndexSchema クラス \[363 ページ\]](#)

1.38.21 UploadUnchangedRows プロパティ

データベースが、変更されていないローをアップロードするかどうかをチェックします。

構文

Visual Basic

```
Public ReadOnly Property UploadUnchangedRows As Boolean
```

C#

```
public unsafe bool UploadUnchangedRows {get;}
```

備考

テーブルが、同期時に常にすべてのローをアップロードするようにマーク付けされている場合は true、変更されたローのみをアップロードするようにマーク付けされている場合は false。

すべてのローをアップロードするようにマーク付けされているテーブルでは、その同期時に、変更されたローと未変更のローがアップロードされます。このようなテーブルは、"完全同期" テーブルと呼ばれることもあります。

1.39 ULTransaction クラス

SQL トランザクションを表します。

構文

Visual Basic

```
Public NotInheritable Class ULTransaction Inherits  
System.Data.Common.DbTransaction
```

C#

```
public sealed class ULTransaction : System.Data.Common.DbTransaction
```

メンバー

ULTransaction 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 override void	Commit() [577 ページ]	データベーストランザクションをコミットします。
protected override void	Dispose(bool) [577 ページ]	
public override void	Rollback() [578 ページ]	データベースへのトランザクションの未処理の変更をロールバックします。

プロパティ

変更子とタイプ	プロパティ	説明
public new ULConnection	Connection [578 ページ]	トランザクションに対応する接続を返します。
protected override DbConnection	DbConnection [579 ページ]	
public override IsolationLevel	IsolationLevel [579 ページ]	トランザクションの独立性レベルを返します。

備考

ULTransaction クラスにはコンストラクタがありません。ULTransaction オブジェクトを取得するには、ULConnection.BeginTransaction メソッドを使用します。コマンドをトランザクションに関連付けるには、ULCommand.Transaction プロパティを使用します。

トランザクションがコミットまたはロールバックされると、接続はすべての操作の実行時にそれらを自動的にコミットする状態に戻ります。さらに操作をグループ化するには、新しいトランザクションを作成する必要があります。

このセクションの内容:

[Commit\(\) メソッド](#) [577 ページ]

データベーストランザクションをコミットします。

[Dispose\(bool\) メソッド](#) [577 ページ]

[Rollback\(\) メソッド](#) [578 ページ]

データベースへのトランザクションの未処理の変更をロールバックします。

[Connection プロパティ](#) [578 ページ]

トランザクションに対応する接続を返します。

[DbConnection プロパティ](#) [579 ページ]

[IsolationLevel プロパティ](#) [579 ページ]

トランザクションの独立性レベルを返します。

関連情報

[BeginTransaction\(\) メソッド \[135 ページ\]](#)

[Transaction プロパティ \[104 ページ\]](#)

1.39.1 Commit() メソッド

データベーストランザクションをコミットします。

構文

Visual Basic

```
Public Overrides Sub Commit ()
```

C#

```
public override void Commit ()
```

備考

トランザクションがコミットまたはロールバックされると、接続はすべての操作の実行時にそれらを自動的にコミットする状態に戻ります。さらに操作をグループ化するには、新しいトランザクションを作成する必要があります。

データベースエラーが原因で Commit メソッドの呼び出しが失敗した場合 (たとえば、参照整合性エラーなど)、トランザクションはアクティブなまま残ります。問題を訂正し、もう一度 Commit メソッドを呼び出します。または、ULTransaction.Rollback メソッドを呼び出してトランザクションを完了させます。

関連情報

[Rollback\(\) メソッド \[578 ページ\]](#)

1.39.2 Dispose(bool) メソッド

構文

Visual Basic

```
Protected Overrides Sub Dispose (ByVal disposing As Boolean)
```

C#

```
protected override void Dispose (bool disposing)
```

1.39.3 Rollback() メソッド

データベースへのトランザクションの未処理の変更をロールバックします。

≡ 構文

Visual Basic

```
Public Overrides Sub Rollback ()
```

C#

```
public override void Rollback ()
```

備考

トランザクションがコミットまたはロールバックされると、接続はすべての操作の実行時にそれらを自動的にコミットする状態に戻ります。さらに操作をグループ化するには、新しいトランザクションを作成する必要があります。

関連情報

[Commit\(\) メソッド \[577 ページ\]](#)

1.39.4 Connection プロパティ

トランザクションに対応する接続を返します。

≡ 構文

Visual Basic

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

C#

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

備考

トランザクションに関連付けられている ULConnection オブジェクト。トランザクションが無効な場合は NULL 参照 (Visual Basic の Nothing)。

これは、System.Data.IDbTransaction.Connection プロパティと System.Data.Common.DbCommand.Connection プロパティが厳密に型指定されたものです。

関連情報

[BeginTransaction\(\) メソッド \[135 ページ\]](#)

[ULConnection クラス \[121 ページ\]](#)

1.39.5 DbConnection プロパティ

構文

Visual Basic

```
Protected ReadOnly Overrides Property DbConnection As DbConnection
```

C#

```
protected override DbConnection DbConnection {get;}
```

1.39.6 IsolationLevel プロパティ

トランザクションの独立性レベルを返します。

構文

Visual Basic

```
Public ReadOnly Overrides Property IsolationLevel As IsolationLevel
```

C#

```
public override IsolationLevel IsolationLevel {get;}
```

備考

System.Data.IsolationLevel 値の 1 つ。Ultra Light.NET では、System.Data.IsolationLevel.ReadUncommitted 値のみがサポートされています。

関連情報

[BeginTransaction\(\) メソッド \[135 ページ\]](#)

1.40 ULInfoMessageEventHandler(object, ULInfoMessageEventArgs) デリゲート

ULConnection.InfoMessage イベントを処理するメソッドを示します。

構文

Visual Basic

```
Public Delegate Sub ULInfoMessageEventHandler (
    ByVal obj As Object,
    ByVal args As ULInfoMessageEventArgs
) As delegate void
```

C#

```
public delegate void ULInfoMessageEventHandler (
    object obj,
    ULInfoMessageEventArgs args
);
```

パラメータ

obj イベントを送信する接続。

args イベントデータが含まれる ULInfoMessageEventArgs オブジェクト。

関連情報

[InfoMessage イベント \[184 ページ\]](#)

[ULInfoMessageEventArgs クラス \[373 ページ\]](#)

1.41 ULRowsCopiedEventHandler(object, ULRowsCopiedEventArgs) デリゲート

ULBulkCopy.ULRowsCopied イベントを処理するメソッドを示します。

構文

Visual Basic

```
Public Delegate Sub ULRowsCopiedEventHandler (  
    ByVal sender As Object,  
    ByVal rowsCopiedEventArgs As ULRowsCopiedEventArgs  
) As delegate void
```

C#

```
public delegate void ULRowsCopiedEventHandler (  
    object sender,  
    ULRowsCopiedEventArgs rowsCopiedEventArgs  
) ;
```

備考

ULRowsCopiedEventHandler デリゲートは、.NET Compact Framework 2.0 では使用できません。

関連情報

[ULRowsCopied イベント \[36 ページ\]](#)

1.42 ULRowUpdatedEventHandler(object, ULRowUpdatedEventArgs) デリゲート

ULDataAdapter.RowUpdated イベントを処理するメソッドを示します。

構文

Visual Basic

```
Public Delegate Sub ULRowUpdatedEventHandler (  
    ByVal sender As Object,  
    ByVal e As ULRowUpdatedEventArgs  
) As delegate void
```

C#

```
public delegate void ULRowUpdatedEventHandler (
    object sender,
    ULRowUpdatedEventArgs e
);
```

パラメータ

sender イベントを送信する接続。

e イベントデータが含まれる ULRowUpdatedEventArgs オブジェクト。

関連情報

[RowUpdated イベント \[254 ページ\]](#)

[ULRowUpdatedEventArgs クラス \[476 ページ\]](#)

1.43 ULRowUpdatingEventHandler(object, ULRowUpdatingEventArgs) デリゲート

ULDataAdapter.RowUpdating イベントを処理するメソッドを示します。

構文

Visual Basic

```
Public Delegate Sub ULRowUpdatingEventHandler (
    ByVal sender As Object,
    ByVal e As ULRowUpdatingEventArgs
) As delegate void
```

C#

```
public delegate void ULRowUpdatingEventHandler (
    object sender,
    ULRowUpdatingEventArgs e
);
```

パラメータ

sender イベントを送信する接続。

e イベントデータが含まれる `ULRowUpdatingEventArgs` オブジェクト。

関連情報

[RowUpdating イベント \[254 ページ\]](#)

[ULRowUpdatingEventArgs クラス \[479 ページ\]](#)

1.44 ULSyncProgressedDlg(IAsyncResult, ULSyncProgressData) デリゲート

同期進捗情報と同期中に起動されたメソッドを表します。

構文

Visual Basic

```
Public Delegate Sub ULSyncProgressedDlg (  
    ByVal result As IAsyncResult,  
    ByVal data As ULSyncProgressData  
) As delegate void
```

C#

```
public delegate void ULSyncProgressedDlg (  
    IAsyncResult result,  
    ULSyncProgressData data  
);
```

パラメータ

result `BeginSynchronize` メソッドから返された `IAsyncResult` オブジェクト。`result.AsyncState` を使用して `BeginSynchronize` メソッドに提供されるオブジェクトにアクセスします。

data 最新の同期のプログレスデータを保持している `ULSyncProgressData` オブジェクト。

備考

GUI 作業を行うか、この方法の中で `Ultra Light.NET API` 呼び出しを作成することが安全です。同期は、このメソッドに対する呼び出し中には保持されません。

関連情報

[BeginSynchronize\(Control, ULSyncProgressedDlg, object\) メソッド \[133 ページ\]](#)

[ULSyncProgressData クラス \[502 ページ\]](#)

1.45 ULAuthStatusCode 列挙体

UL 拡張: Mobile Link ユーザ認証の実行中にレポートされる可能性のあるステータスコードを列挙します。

構文

Visual Basic

```
Public Enum ULAuthStatusCode
```

C#

```
enum ULAuthStatusCode
```

メンバー

メンバー名	説明	Value
UNKNOWN	認証ステータスが不明です。接続がまだ同期を実行していない可能性があります (UNKNOWN = 0)。	0
VALID	ユーザ ID とパスワードは、同期時には有効でした (VALID = 1)。	1
VALID_BUT_EXPIRES_SOON	ユーザ ID とパスワードは、同期時には有効でしたが、まもなく有効期限が切れます (VALID_BUT_EXPIRES_SOON = 2)。	2
EXPIRED	ユーザ ID またはパスワードの有効期限が切れているため、認証に失敗しました (EXPIRED = 3)。	3
INVALID	ユーザ ID またはパスワードが正しくないため、認証に失敗しました (INVALID = 4)。	4
IN_USE	ユーザ ID がすでに使用されているため、認証に失敗しました (IN_USE = 5)。	5

関連情報

[AuthStatus プロパティ \[521 ページ\]](#)

1.46 ULBulkCopyOptions 列挙体

ULBulkCopy クラスのインスタンスで使用する、1 つ以上のオプションを指定するビット単位フラグです。

構文

Visual Basic

```
Public Enum ULBulkCopyOptions
```

C#

```
enum ULBulkCopyOptions
```

メンバー

メンバー名	説明	Value
デフォルト	これだけを指定すると、デフォルトの動作が使用されます。	0x0
KeepIdentity	これを指定すると、IDENTITY カラムにコピーされる元の値が保持されます。 デフォルトでは、送信先テーブルでは新しい ID 番号が生成されます。	0x1
UseInternalTransaction	これを指定すると、バルクコピーオペレーションの各バッチがトランザクションの中で実行されます。 これが指定されない場合、トランザクションは使用されません。このオプションを指定し、コンストラクタに ULTransaction オブジェクトも指定すると、System.ArgumentException が発生します。	0x2

備考

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

ULBulkCopyOptions 列挙体は、ULBulkCopy インスタンスの構築時に使用され、WriteToServer メソッドの動作方法を指定します。

関連情報

[ULBulkCopy クラス \[23 ページ\]](#)

1.47 ULDateOrder 列挙体

UL 拡張: データベースがサポートできる日付順を列挙します。

構文

Visual Basic

```
Public Enum ULDateOrder
```

C#

```
enum ULDateOrder
```

メンバー

メンバー名	説明
YMD	年、月、日の順に示された日付。
MDY	月、日、年の順に示された日付。
DMY	日、月、年の順に示された日付。

1.48 ULDbType 列挙体

Ultra Light.NET データベースのデータ型を列挙します。

構文

Visual Basic

```
Public Enum ULDbType
```

C#

```
enum ULDbType
```

メンバー

メンバー名	説明	値
BigInt	符号付き 64 ビット整数値	ULNET_TYPE_S_BIG
Binary	指定された最大長のバイナリデータ。 列挙値 <i>Binary</i> と <i>VarBinary</i> は同等のエイリアスです。	ULNET_TYPE_BINARY
Bit	1 ビットフラグ	ULNET_TYPE_BIT
Char	指定された長さの文字データ。 Ultra Light.NET では、この型は常に Unicode 文字をサポートします。 <i>Char</i> 型と <i>VarChar</i> 型は、完全に互換性があります。	ULNET_TYPE_FIXCHAR
日付	日付情報	ULNET_TYPE_DATE
DateTime	タイムスタンプ情報 (日付、時刻)。 列挙値 <i>DateTime</i> と <i>TimeStamp</i> は同等のエイリアスです。	ULNET_TYPE_TIMESTAMP
Decimal	精度と桁数が指定された正確な数値データ。 列挙値 <i>Decimal</i> と <i>Numeric</i> は同等のエイリアスです。	ULNET_TYPE_NUMERIC
Double	倍精度浮動小数点数 (8 バイト)	ULNET_TYPE_DOUBLE
Float	単精度浮動小数点数 (4 バイト)。 列挙値 <i>Float</i> と <i>Real</i> は同等のエイリアスです。	ULNET_TYPE_REAL
整数	符号なし 32 ビット整数値	ULNET_TYPE_S_LONG
LongBinary	可変長のバイナリデータ	ULNET_TYPE_LONGBINARY
LongVarchar	可変長の文字データ。 Ultra Light.NET では、この型は常に Unicode 文字をサポートします。	ULNET_TYPE_LONGVARCHAR

メンバー名	説明	値
Numeric	精度と桁数が指定された正確な数値データ。 列挙値 <i>Decimal</i> と <i>Numeric</i> は同等のエイリアスです。	ULNET_TYPE_NUMERIC
Real	単精度浮動小数点数 (4 バイト)。 列挙値 <i>Float</i> と <i>Real</i> は同等のエイリアスです。	ULNET_TYPE_REAL
SmallInt	符号付き 16 ビット整数値	ULNET_TYPE_S_SHORT
STGeometry	ST_Geometry 情報	ULNET_TYPE_ST_GEOMETRY
Time	時刻情報	ULNET_TYPE_TIME
TimeStamp	タイムスタンプ情報 (日付、時刻)。 列挙値 <i>DateTime</i> と <i>TimeStamp</i> は同等のエイリアスです。	ULNET_TYPE_TIMESTAMP
TimeStampWithTimeZone	タイムゾーンオフセット付きのタイムスタンプ情報 (日付、時刻)	ULNET_TYPE_TIMESTAMP_WITH_TIME_ZONE
TinyInt	符号なし 8 ビット整数値	ULNET_TYPE_TINY
UniquelIdentifier	ユニバーサルユニーク識別子 (UUID/GUID)	ULNET_TYPE_UUID
UnsignedBigInt	符号なし 64 ビット整数値	ULNET_TYPE_U_BIG
UnsignedInt	符号なし 32 ビット整数値	ULNET_TYPE_U_LONG
UnsignedSmallInt	符号なし 16 ビット整数値	ULNET_TYPE_U_SHORT
VarBinary	指定された最大長のバイナリデータ。 列挙値 <i>Binary</i> と <i>VarBinary</i> は同等のエイリアスです。	ULNET_TYPE_BINARY
VarChar	指定された最大長の文字データ。 Ultra Light.NET では、この型は常に Unicode 文字をサポートします。 <i>Char</i> 型と <i>VarChar</i> 型は、完全に互換性があります。	ULNET_TYPE_CHAR

備考

下の表には、各 ULDbType との互換性がある .NET 型がリストされています。整数型の場合、テーブルのカラムは、常により小さい整数型を使用して設定できるほか、実際の値がその型の範囲内にあるかぎり、より大きい型を使用して設定することも可能です。

ULDbType	互換性のある .NET 型	C# 組み込みタイプ	Visual Basic 組み込みタイプ
<i>Binary</i> 、 <i>VarBinary</i>	System.Byte[], または System.Guid (サイズが 16 の場合)	byte[]	Byte()
<i>Bit</i>	System.Boolean	bool	Boolean
<i>Char</i> 、 <i>VarChar</i>	System.String	String	String
<i>Date</i>	System.DateTime	DateTime (組み込みタイプなし)	Date
<i>Double</i>	System.Double	double	Double
<i>LongBinary</i>	System.Byte[]	byte[]	Byte()
<i>LongVarchar</i>	System.String	String	String
<i>Decimal</i> 、 <i>Numeric</i>	System.Decimal	decimal	Decimal
<i>Float</i> 、 <i>Real</i>	System.Single	float	Single
<i>BigInt</i>	System.Int64	long	Long
<i>Integer</i>	System.Int32	int	Integer
<i>SmallInt</i>	System.Int16	short	Short
<i>ST_Geometry</i>	System.String	String	String
<i>Time</i>	System.TimeSpan	TimeSpan (組み込みタイプなし)	TimeSpan (組み込みタイプなし)
<i>DateTime</i> 、 <i>TimeStamp</i>	System.DateTime	DateTime (組み込みタイプなし)	Date
<i>TimeStampWithTimeZone</i>	System.String	String	String
<i>TinyInt</i>	System.Byte	byte	Byte
<i>UnsignedBigInt</i>	System.UInt64	ulong	UInt64 (組み込みタイプなし)
<i>UnsignedInt</i>	System.UInt32	uint	UInt32 (組み込みタイプなし)
<i>UnsignedSmallInt</i>	System.UInt16	ushort	UInt16 (組み込みタイプなし)
<i>UniqueIdentifier</i>	System.Guid	Guid (組み込みタイプなし)	Guid (組み込みタイプなし)

長さが 16 のバイナリカラムには、UniqueIdentifier 型との完全な互換性があります。

関連情報

[GetFieldType\(int\) メソッド \[295 ページ\]](#)

[GetDataTypeName\(int\) メソッド \[291 ページ\]](#)

[GetColumnULDbType\(int\) メソッド \[235 ページ\]](#)

1.49 ULDBValid 列挙体

Ultra Light.NET データベース検証メソッドを列挙します。

構文

Visual Basic

```
Public Enum ULDBValid
```

C#

```
enum ULDBValid
```

メンバー

メンバー名	説明	値
EXPRESS_VALIDATE	完全度は低いが、より高速な検証を実行します。	0
FULL_VALIDATE	テーブル、インデックス、すべてのデータベースページを検証します。	1

関連情報

[ValidateDatabase\(string, ULDBValid\) メソッド \[263 ページ\]](#)

[ValidateDatabase\(ULDBValid\) メソッド \[175 ページ\]](#)

[ValidateDatabase\(ULDBValid, string\) メソッド \[176 ページ\]](#)

1.50 ULRuntimeType 列挙体

UL 拡張: Ultra Light.NET ランタイムのタイプを列挙します。

構文

Visual Basic

```
Public Enum ULRuntimeType
```

C#

```
enum ULRuntimeType
```

メンバー

メンバー名	説明	値
STANDALONE_UL	スタンドアロンの Ultra Light.NET ランタイムを選択します。 スタンドアロンのランタイムは、データベースに直接アクセスします。データベースへのアクセスは高速になりますが、データベースを共有することはできません。	0
UL_ENGINE_CLIENT	Ultra Light エンジンのランタイムを選択します。 Ultra Light.NET エンジンクライアントは、Ultra Light エンジンと通信してデータベースにアクセスします。異なるアプリケーションでデータベースを共有できます。	1

関連情報

[RuntimeType プロパティ \[264 ページ\]](#)

1.51 ULSqlProgressState 列挙体

UL 拡張: SQL パススルースクリプトの実行中に発生する可能性のあるすべてのステータスを列挙します。

構文

Visual Basic

```
Public Enum ULSqlProgressState
```

C#

```
enum ULSqlProgressState
```

メンバー

メンバー名	説明	値
STATE_STARTING	実行されたスクリプトはまだありません。	0
STATE_RUNNING_SCRIPT	SQL パススルースクリプトを実行中です。	1
STATE_DONE	スクリプトは正常に完了しました。	2
STATE_ERROR	スクリプトは完了しましたが、エラーが発生しました。	3

関連情報

[ULSqlProgressData クラス \(廃止予定\) \[484 ページ\]](#)

1.52 ULStreamType 列挙体

UL 拡張: 同期に使用する Mobile Link 同期ストリームのタイプを列挙します。

構文

Visual Basic

```
Public Enum ULStreamType
```

C#

```
enum ULStreamType
```

メンバー

メンバー名	説明
TCPIP	TCP/IP を介して同期します。


メンバー名	説明
HTTP	HTTP を介して同期します。 HTTP ストリームは基本となるトランスポートとして TCP/IP を使用します。Ultra Light アプリケーションは Web ブラウザとして機能し、Mobile Link サーバは Web サーバとして機能します。Ultra Light アプリケーションは、サーバへのデータ送信のために POST 要求を送り、サーバからのデータの読み込みのために GET 要求を送ります。
HTTPS	HTTPS を介して同期します (トランスポートレイヤセキュリティを適用した HTTP)。
TLS	TLS (TCP/IP を介して同期)

関連情報

[Stream プロパティ \[499 ページ\]](#)

1.53 ULSyncProgressState 列挙体

UL 拡張: 同期中に発生する可能性のあるすべてのステータスを列挙します。

 構文

Visual Basic

```
Public Enum ULSyncProgressState
```

C#

```
enum ULSyncProgressState
```

メンバー

メンバー名	説明	Value
STATE_STARTING	同期処理はまだ行われていません。	0
STATE_CONNECTING	同期ストリームは構築されていますが、まだ開かれていません。	1

メンバー名	説明	Value
STATE_RESUMING_DOWNLOAD	部分ダウンロードを再開しようとしたときに入るオプションの状態。 成功すると、同期が STATE_RECEIVING_TABLE 状態に進みます。再開できない場合は、STATE_ERROR になります。	2
STATE_SENDING_HEADER	同期ストリームが開かれ、ヘッダが送信されようとしています。	3
STATE_SENDING_CHECK_SYNC_REQUEST	前回のアップロードの状態が不明なため、ステータスをチェックする要求が送信されます。	4
STATE_WAITING_FOR_CHECK_SYNC_RESPONSE	サーバが同期チェック要求に応答するのを待機しています。	5
STATE_PROCESSING_CHECK_SYNC_RESPONSE	同期チェック要求への応答が受け取られ、処理されています。	6
STATE_SENDING_TABLE	テーブルが送信されています。 進行状況をモニタするには、 ULSyncProgressData.SyncTableIndex プロパティと ULSyncProgressData.SyncTableCount プロパティを使用します。	7
STATE_SENDING_DATA	現在のテーブルのデータが送信されています。 ULSyncProgressData.SentBytes、 ULSyncProgressData.SentInserts、 ULSyncProgressData.SentUpdates、および ULSyncProgressData.SentDeletes の プロパティは更新されました。	8
STATE_FINISHING_UPLOAD	アップロードの完了処理中です。 送信された最終的なロー数が、このイベントに含まれます。	9
STATE_WAITING_FOR_UPLOAD_ACK	サーバがアップロードの受信を確認するのを待機しています。	10
STATE_PROCESSING_UPLOAD_ACK	サーバがアップロードの受信を確認しました。	11
STATE_WAITING_FOR_DOWNLOAD	サーバがダウンロードの送信を開始するのを待機しています。	12

メンバー名	説明	Value
STATE_RECEIVING_TABLE	テーブルを受信しています。 進行状況をモニタするには、 ULSyncProgressData.SyncTableIndex プロパティと ULSyncProgressData.SyncTableCount プロパティを使用します。	13
STATE_RECEIVING_DATA	現在のテーブルのデータを受信しています。 ULSyncProgressData.ReceivedBytes、 ULSyncProgressData.ReceivedInserts、 ULSyncProgressData.ReceivedUpdates 、および ULSyncProgressData.ReceivedDeletes のプロパティは更新されました。	14
STATE_COMMITTING_DOWNLOAD	ダウンロードをコミットしています。 受信された最終的なロー数が、このイベント に含まれます。	15
STATE_ROLLING_BACK_DOWNLOAD	ダウンロード中にエラーが発生したため、同 期によってダウンロードがロールバックされ ています。 後続の STATE_ERROR 進行状況レポート にエラーがレポートされます。	16
STATE_SENDING_DOWNLOAD_ACK	ダウンロード完了の確認が送信されていま す。	17
STATE_DISCONNECTING	同期ストリームが閉じられようとしています。	18
STATE_DONE	同期は正常に完了しました。	19
STATE_ERROR	同期は完了しましたが、エラーが発生しまし た。	20
STATE_CANCELLED	同期がキャンセルされました。	21

関連情報

[ULSyncProgressData クラス \[502 ページ\]](#)

[ReceivedBytes プロパティ \[508 ページ\]](#)

[ReceivedInserts プロパティ \[510 ページ\]](#)

[ReceivedUpdates プロパティ \[510 ページ\]](#)

[ReceivedDeletes プロパティ \[509 ページ\]](#)

[SentBytes プロパティ \[511 ページ\]](#)

[SentInserts プロパティ \[512 ページ\]](#)

[SentUpdates プロパティ \[513 ページ\]](#)

[SentDeletes プロパティ \[512 ページ\]](#)

[SyncTableIndex プロパティ \[515 ページ\]](#)

[SyncTableCount プロパティ \[514 ページ\]](#)

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を行いません。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。