PUBLIC
SQL Anywhere - UltraLite
Document Version: 17.01.0 – 2021-10-15

UltraLite - UWP API Reference

THE BEST RUN SAP

Content

1 UltraLite for Universal Windows Platform (UWP) API Reference. 11

1.1 Connection Class. 13

CancelGetNotification(String^) Method. 17

ChangeEncryptionKey(String^) Method. 18

Checkpoint() Method. 18

CloseObject() Method. 19

Commit() Method. 19

CountUploadRows(String^, unsigned int) Method. 19

CreateNotificationQueue(String^) Method. 20

DeclareEvent(String^) Method. 21

DestroyNotificationQueue(String^) Method. 21

ExecuteStatement(String^) Method. 22

GetDatabaseProperty(String^) Method. 22

GetDatabasePropertyInt(String^) Method. 23

GetDatabaseSchema() Method. 24

GetLastDownloadTime(String^) Method. 24

GetLastError(ULSqlCode *, String^*) Method. 25

GetLastIdentity() Method. 25

GetNotification(String^, unsigned int) Method. 26

GetNotificationParameter(String^, String^) Method. 26

GetSyncResult() Method. 27

GlobalAutoIncrementUsage() Method. 27

GrantConnectTo(String^, String^) Method. 28

OpenTable(String^, String^) Method. 28

PrepareStatement(String^) Method. 29

RegisterForEvent(String^, String^, String^, bool) Method. 29

ResetLastDownloadTime(String^) Method. 30

RevokeConnectFrom(String^) Method. 31

Rollback() Method. 31

RollbackPartialDownload() Method. 31

SendNotification(String^, String^, String^) Method. 32

SetDatabaseOption(String^, String^) Method. 33

SetDatabaseOptionInt(String^, unsigned int) Method. 33

StartSynchronizationDelete() Method. 33

StopSynchronizationDelete() Method. 34

Synchronize(String^, SyncObserver^) Method. 34

SynchronizeAsync(String^) Method. . . . 34
TriggerEvent(String^, String^) Method. . . . 35
ValidateDatabase(uint16, String^, ValidateCallback^) Method. . . . 36
ValidateDatabaseAsync(uint16flags, String^) Method. . . . 36

1.2 Constants Class. . . . 37
BLOB_CONTINUE Property. . . . 38
SYNC_ALL Property. . . . 38
SYNC_ALL_PUBS Property. . . . 38

1.3 Cursor Interface. . . . 39
AfterLast() Method. . . . 44
AppendBytes Method. . . . 44
AppendChars Method. . . . 46
BeforeFirst() Method. . . . 47
CloseObject() Method. . . . 48
Delete() Method. . . . 48
DeleteNamed(String^) Method. . . . 48
First() Method. . . . 48
GetBinaryLength Method. . . . 49
GetBool Method. . . . 50
GetByte Method. . . . 52
GetBytes Method. . . . 53
GetChars Method. . . . 55
GetDateTime Method. . . . 57
GetDouble Method. . . . 59
GetFloat Method. . . . 60
GetGuid Method. . . . 62
GetInt16 Method. . . . 63
GetInt32 Method. . . . 65
GetInt64 Method. . . . 66
GetRowCount(unsigned int) Method. . . . 68
GetState() Method. . . . 68
GetString Method. . . . 69
GetStringLength Method. . . . 70
GetUInt16 Method. . . . 72
GetUInt32 Method. . . . 73
GetUInt64 Method. . . . 75
IsNull Method. . . . 76
Last() Method. . . . 78
Next() Method. . . . 78
Previous() Method. . . . 78
Relative(int) Method. . . . 79

SetBool Method. 79
SetByte Method. 80
SetBytes Method. 82
SetDateTime Method. 83
SetDefault Method. 84
SetDouble Method. 86
SetFloat Method. 87
SetGuid Method. 88
SetInt16 Method. 89
SetInt32 Method. 91
SetInt64 Method. 92
SetNull Method. 93
SetString Method. 94
SetUInt16 Method. 96
SetUInt32 Method. 97
SetUInt64 Method. 99
Update() Method. 100
UpdateBegin() Method. 100
1.4 CursorSchema Interface. 101
GetColumnCount() Method. 102
GetColumnID(String^) Method. 102
GetColumnName(uint16, uint16) Method. 103
GetColumnPrecision(uint16) Method. 104
GetColumnScale(uint16) Method. 104
GetColumnSize(uint16) Method. 105
GetColumnSQLType(uint16) Method. 105
GetColumnType(uint16) Method. 106
IsAliased(uint16) Method. 106
1.5 DatabaseManager Class. 107
CreateDatabase(String^, String^) Method. 108
CreateFileTransfer(String^, uint16, String^, String^) Method. 109
DropDatabase(String^) Method. 110
Fini() Method. 110
GetLastError(ULSqlCode *, String^*) Method. 111
Init() Method. 111
OpenConnection(String^) Method. 112
ValidateDatabase(String^, uint16, ValidateCallback^) Method. 112
ValidateDatabaseAsync(String^, uint16flags) Method. 113
1.6 DatabaseSchema Class. 114
CloseObject() Method. 115
GetPublicationCount() Method. 115

GetPublications() Method. 116
GetTableCount() Method. 116
GetTables() Method. 116
GetTableSchema(String^) Method. 117
1.7 FileTransfer Class. 117
DownloadFile(FileTransferObserver^) Method. 119
DownloadFileAsync() Method. 120
GetResult() Method. 120
UploadFile(FileTransferObserver^) Method. 120
UploadFileAsync() Method. 121
1.8 IndexSchema Class. 121
CloseObject() Method. 123
GetColumnCount() Method. 123
GetColumnName(uint16) Method. 123
GetIndexColumnID(String^) Method. 124
GetIndexFlags() Method. 124
GetName() Method. 125
GetReferencedIndexName() Method. 125
GetReferencedTableName() Method. 125
GetTableName() Method. 126
IsColumnDescending(uint16) Method. 126
1.9 PreparedStatement Class. 127
AppendParameterByteChunk(uint16, const Array< uint8 >^, int) Method. 130
AppendParameterStringChunk(uint16, String^) Method. 131
CloseObject() Method. 131
ExecuteQuery() Method. 131
ExecuteQueryAsync() Method. 132
ExecuteStatement() Method. 132
ExecuteStatementAsync() Method. 132
GetParameterCount() Method. 133
GetParameterID(String^) Method. 133
GetParameterType(uint16) Method. 134
GetPlan() Method. 134
GetResultSetSchema() Method. 135
GetRowsAffectedCount() Method. 135
HasResultSet() Method. 135
SetParameterBool(uint16, bool) Method. 136
SetParameterByte(uint16, uint8) Method. 136
SetParameterBytes(uint16, const Array< uint8 >^, int) Method. 137
SetParameterDateTime(uint16, DateTime) Method. 137
SetParameterDouble(uint16, double) Method. 137

SetParameterFloat(uint16, float) Method. 138
SetParameterGuid(uint16, Guid) Method. 138
SetParameterInt16(uint16, int16) Method. 139
SetParameterInt32(uint16, int) Method. 139
SetParameterInt64(uint16, int64) Method. 139
SetParameterNull(uint16) Method. 140
SetParameterString(uint16, String^) Method. 140
SetParameterUInt16(uint16, uint16) Method. 141
SetParameterUInt32(uint16, unsigned int) Method. 141
SetParameterUInt64(uint16, uint64) Method. 141
1.10 ResultSet Class. 142
AfterLast() Method. 147
AppendBytes Method. 147
AppendChars Method. 149
BeforeFirst() Method. 150
CloseObject() Method. 151
Delete() Method. 151
DeleteNamed(String^) Method. 151
First() Method. 151
GetBinaryLength Method. 152
GetBool Method. 153
GetByte Method. 155
GetBytes Method. 156
GetChars Method. 158
GetDateTime Method. 160
GetDouble Method. 162
GetFloat Method. 163
GetGuid Method. 165
GetInt16 Method. 166
GetInt32 Method. 168
GetInt64 Method. 169
GetResultSetSchema() Method. 171
GetRowCount(unsigned int) Method. 171
GetState() Method. 172
GetString Method. 172
GetStringLength Method. 174
GetUInt16 Method. 175
GetUInt32 Method. 177
GetUInt64 Method. 178
IsNull Method. 180
Last() Method. 181

Next() Method. 181
Previous() Method. 182
Relative(int) Method. 182
SetBool Method. 183
SetByte Method. 184
SetBytes Method. 185
SetDateTime Method. 187
SetDefault Method. 188
SetDouble Method. 189
SetFloat Method. 190
SetGuid Method. 192
SetInt16 Method. 193
SetInt32 Method. 194
SetInt64 Method. 195
SetNull Method. 196
SetString Method. 198
SetUInt16 Method. 199
SetUInt32 Method. 200
SetUInt64 Method. 202
Update() Method. 203
UpdateBegin() Method. 203
1.11 ResultSetSchema Class. 204
GetColumnCount() Method. 205
GetColumnID(String^) Method. 205
GetColumnName(uint16, uint16type) Method. 206
GetColumnPrecision(uint16) Method. 207
GetColumnScale(uint16) Method. 207
GetColumnSize(uint16) Method. 208
GetColumnSQLType(uint16) Method. 208
GetColumnType(uint16) Method. 209
IsAliased(uint16) Method. 209
1.12 Table Class. 210
AfterLast() Method. 216
AppendBytes Method. 217
AppendChars Method. 218
BeforeFirst() Method. 220
CloseObject() Method. 220
Delete() Method. 221
DeleteAllRows() Method. 221
DeleteNamed(String^) Method. 221
Find(uint16) Method. 222

FindBegin() Method. . . . 222
FindFirst(uint16) Method. . . . 223
FindLast(uint16) Method. . . . 223
FindNext(uint16) Method. . . . 224
FindPrevious(uint16) Method. . . . 225
First() Method. . . . 225
GetBinaryLength Method. . . . 225
GetBool Method. . . . 227
GetByte Method. . . . 229
GetBytes Method. . . . 230
GetChars Method. . . . 232
GetDateTime Method. . . . 234
GetDouble Method. . . . 236
GetFloat Method. . . . 237
GetGuid Method. . . . 239
GetInt16 Method. . . . 240
GetInt32 Method. . . . 242
GetInt64 Method. . . . 243
GetRowCount(unsigned int) Method. . . . 245
GetState() Method. . . . 245
GetString Method. . . . 246
GetStringLength Method. . . . 247
GetTableSchema() Method. . . . 249
GetUInt16 Method. . . . 249
GetUInt32 Method. . . . 251
GetUInt64 Method. . . . 252
Insert() Method. . . . 254
InsertBegin() Method. . . . 254
IsNull Method. . . . 254
Last() Method. . . . 256
Lookup(uint16) Method. . . . 256
LookupBackward(uint16) Method. . . . 256
LookupBegin() Method. . . . 257
LookupForward(uint16) Method. . . . 257
Next() Method. . . . 258
Previous() Method. . . . 258
Relative(int) Method. . . . 259
SetBool Method. . . . 259
SetByte Method. . . . 260
SetBytes Method. . . . 262
SetDateTime Method. . . . 263

SetDefault Method. 264
SetDouble Method. 266
SetFloat Method. 267
SetGuid Method. 268
SetInt16 Method. 269
SetInt32 Method. 270
SetInt64 Method. 272
SetNull Method. 273
SetString Method. 274
SetUInt16 Method. 275
SetUInt32 Method. 277
SetUInt64 Method. 278
TruncateTable() Method. 279
Update() Method. 280
UpdateBegin() Method. 280
1.13 TableSchema Class. 280
CloseObject() Method. 283
GetColumnCount() Method. 283
GetColumnDefault(uint16) Method. 284
GetColumnDefaultType(uint16) Method. 284
GetColumnID(String^) Method. 285
GetColumnName(uint16, uint16type) Method. 285
GetColumnPrecision(uint16) Method. 286
GetColumnScale(uint16) Method. 286
GetColumnSize(uint16) Method. 287
GetColumnSQLType(uint16) Method. 287
GetColumnType(uint16) Method. 288
GetGlobalAutoincPartitionSize(uint16) Method. 288
GetIndexCount() Method. 289
GetIndexes() Method. 289
GetIndexSchema(String^) Method. 290
GetName() Method. 290
GetOptimalIndex(uint16) Method. 290
GetPrimaryKey() Method. 291
GetPublicationPredicate(String^) Method. 291
GetTableSyncType() Method. 292
InPublication(String^) Method. 292
IsAliased(uint16) Method. 293
IsColumnInIndex(uint16, String^) Method. 293
IsColumnNullable(uint16) Method. 294
1.14 FileTransferObserver(FileTransferStatus^) Delegate. 294

1.15 SyncObserver(SyncStatus^) Delegate. 294
1.16 ValidateCallback(ValidateData^) Delegate. 295
1.17 AuthStatusCode Enumeration. 295
1.18 ColumnDefaultType Enumeration. 296
1.19 ColumnNameType Enumeration. 296
1.20 ColumnSQLType Enumeration. 298
1.21 ColumnStorageType Enumeration. 299
1.22 CursorState Enumeration. 300
1.23 ErrorCodes Enumeration. 301
1.24 IndexFlag Enumeration. 301
1.25 StreamType Enumeration. 302
1.26 SyncState Enumeration. 303
1.27 TableSyncType Enumeration. 304
1.28 ValidateFlags Enumeration. 305
1.29 ValidateStatusId Enumeration. 306
1.30 FileTransferResult Structure. 307
1.31 FileTransferStatus Structure. 308
1.32 SyncResult Structure. 309
1.33 SyncStatus Structure. 311
1.34 ValidateData Structure. 313

1 UltraLite for Universal Windows Platform (UWP) API Reference

UltraLite for Universal Windows Platform (UWP) has a variety of API objects that you use from C#, C++, or JavaScript.

The following list describes some of the commonly used API objects:

DatabaseManager

Provides methods for managing databases and connections.

Connection

Represents a connection to an UltraLite database. You can create one or more Connection objects.

PreparedStatement, ResultSet

Create dynamic SQL statements, make queries, execute INSERT, UPDATE, and DELETE statements, and attain programmatic control over database result sets.

Namespace

```
UltraLite
```

In this section:

Connection Class [page 13]
Represents a connection to an UltraLite database.

Constants Class [page 37]
Specifies constants that can be used for the UltraLite UWP APIs.

Cursor Interface [page 39]
Represents a cursor in an UltraLite database.

CursorSchema Interface [page 101]
Common interface for ResultSetSchema and TableSchema.

DatabaseManager Class [page 107]
Manages connections and databases.

DatabaseSchema Class [page 114]
Represents the schema of an UltraLite database. Use Connection.GetDatabaseSchema to get a DatabaseSchema object.

FileTransfer Class [page 117]
Transfers a file from a remote database via the MobiLink server.

IndexSchema Class [page 121]
Represents the schema of an UltraLite table index. Use TableSchem.GetIndexSchema(indexName) to get an IndexSchema object.

PreparedStatement Class [page 127]
Represents a prepared SQL statement. Connection.PrepareStatement returns a PreparedStatement.

ResultSet Class [page 142]
Represents a result set in an UltraLite database. ResultSet objects are used for updates and deletes and are returned by PreparedStatement.ExecuteQuery().

ResultSetSchema Class [page 204]
Represents the schema of an UltraLite result set. ResultSet.GetResultSetSchema or PreparedStatement.GetResultSetSchema return a ResultSetSchema.

Table Class [page 210]
Represents a table in an UltraLite database. Connection.OpenTable returns a Table.

TableSchema Class [page 280]
Represents the schema of an UltraLite table. DatabaseSchema.GetTables() or GetTableSchema() or Table.GetTableSchema() return a TableSchema.

FileTransferObserver(FileTransferStatus^) Delegate [page 294]
The definition for a delegate that is called at various stages of the file transfer.

SyncObserver(SyncStatus^) Delegate [page 294]
The definition for a delegate that is called at various stages of the synchronization.

ValidateCallback(ValidateData^) Delegate [page 295]
The definition for a delegate that is called for validation feedback.

AuthStatusCode Enumeration [page 295]
Specifies MobiLink authentication status codes. SyncResult and FileTransferResult have AuthStatus fields that reuturn an AuthStatusCode value.

ColumnDefaultType Enumeration [page 296]
Identifies a column default type.

ColumnNameType Enumeration [page 296]
Specifies values that control how a column name is retrieved when describing a result set.

ColumnSQLType Enumeration [page 298]
Specifies the SQL types for a column. CursorSchema.GetColumnSQLType() returns a ColumnSQLType.

ColumnStorageType Enumeration [page 299]
Specifies the host variable types for a column.

CursorState Enumeration [page 300]
Specifies a possible result set or cursor states. Cursor.GetState() returns a CursorState.

ErrorCodes Enumeration [page 301]
Specifies the Platform::COMException HResult code that indicates an UltraLite exception.

IndexFlag Enumeration [page 301]
Specifies the flags (bit fields) that identify properties of an index.

StreamType Enumeration [page 302]
Specifies the possible values for the FileTransfer.Stream property.

SyncState Enumeration [page 303]
Indicates the current stage of synchronization.

TableSyncType Enumeration [page 304]
Identifies a table synchronization type.

ValidateFlags Enumeration [page 305]
: Represents validation input flags. The ValidateDatabase methods on Connection and DatabaseManager use ValidateFlags as the flags parameter.

ValidateStatusId Enumeration [page 306]
: Specifies possible status IDs for validate progress callbacks.

FileTransferResult Structure [page 307]
: Stores the file transfer result so that appropriate action can be taken in the application.

FileTransferStatus Structure [page 308]
: Stores status/progress information while the file upload/download is in progress.

SyncResult Structure [page 309]
: Stores the synchronization result so that appropriate action can be taken in the application. Connection.GetSyncResult() returns a SyncResult.

SyncStatus Structure [page 311]
: Stores status/progress information while the synchronization is in progress.

ValidateData Structure [page 313]
: Stores validation status information while the validation is in progress.

1.1 Connection Class

Represents a connection to an UltraLite database.

Namespace

```
UltraLite
```

Syntax

```
public ref class  sealed
```

Members

All members of Connection, including inherited members.

Methods

Modifier and Type	Method	Description
public unsigned int	CancelGetNotification(String^) [page 17]	Cancels any pending get-notification calls on all queues matching the given name.
public void	ChangeEncryptionKey(String^) [page 18]	Changes the database encryption key for an UltraLite database.
public void	Checkpoint() [page 18]	Performs a checkpoint operation, flushing any pending committed transactions to the database.
public void	Close() [page 19]	Destroys this connection and any remaining associated objects.
public void	Commit() [page 19]	Commits the current transaction.
public unsigned int	CountUploadRows(String^, unsigned int) [page 19]	Counts the number of rows that need to be uploaded for synchronization.
public void	CreateNotificationQueue(String^) [page 20]	Creates an event notification queue for this connection.
public void	DeclareEvent(String^) [page 21]	Declares an event that can be registered for and triggered.
public void	DestroyNotificationQueue(String^) [page 21]	Destroys the given event notification queue.
public void	ExecuteStatement(String^) [page 22]	Executes a SQL statement string directly.
public String	GetDatabaseProperty(String^) [page 22]	Obtains the value of a database property.
public unsigned int	GetDatabasePropertyInt(String^) [page 23]	Obtains the integer value of a database property.
public DatabaseSchema	GetDatabaseSchema() [page 24]	Returns an object that is used to query the schema of the database.
public DateTime	GetLastDownloadTime(String^) [page 24]	Obtains the last time a specified publication was downloaded.
public void	GetLastError(ULSqlCode *, String^*) [page 25]	Returns the error information associated with the last call on this Connection.
public uint64	GetLastIdentity() [page 25]	Gets the @@identity value.
public String	GetNotification(String^, unsigned int) [page 26]	Reads an event notification.
public String	GetNotificationParameter(String^, String^) [page 26]	Gets a parameter for the event notification that was just read by the GetNotification method.
public SyncResult	GetSyncResult() [page 27]	Gets the result of the last synchronization.
public uint16	GlobalAutoIncrementUsage() [page 27]	Obtains the percent of the default values used in all the columns that have global autoincrement defaults.

Modifier and Type	Method	Description
public void	GrantConnectTo(String^, String^) [page 28]	Grants access to an UltraLite database for a new or existing user ID with the given password.
public Table	OpenTable(String^, String^) [page 28]	Opens a table.
public PreparedStatement	PrepareStatement(String^) [page 29]	Prepares a SQL statement.
public void	RegisterForEvent(String^, String^, String^, bool) [page 29]	Registers or unregisters a queue to receive notifications of an event.
public void	ResetLastDownloadTime(String^) [page 30]	Resets the last download time of a publication so that the application resynchronizes previously downloaded data.
public void	RevokeConnectFrom(String^) [page 31]	Revokes access from an UltraLite database for a user ID.
public void	Rollback() [page 31]	Rolls back the current transaction.
public void	RollbackPartialDownload() [page 31]	Rolls back the changes from a failed synchronization.
public unsigned int	SendNotification(String^, String^, String^) [page 32]	Sends a notification to all queues matching the given name.
public void	SetDatabaseOption(String^, String^) [page 33]	Sets the specified database option.
public void	SetDatabaseOptionInt(String^, unsigned int) [page 33]	Sets a database option.
public void	StartSynchronizationDelete() [page 33]	Sets START SYNCHRONIZATION DELETE for this connection.
public void	StopSynchronizationDelete() [page 34]	Sets STOP SYNCHRONIZATION DELETE for this connection.
public void	Synchronize(String^, SyncObserver^) [page 34]	Initiates synchronization in an UltraLite application.
public IAsyncActionWithProgress< SyncStatus^>	SynchronizeAsync(String^) [page 34]	Initiates synchronization in an UltraLite application.
public unsigned int	TriggerEvent(String^, String^) [page 35]	Triggers a user-defined event and sends notifications to all registered queues.
public void	ValidateDatabase(uint16, String^, ValidateCallback^) [page 36]	Validates the database on this connection.
public IAsyncActionWithProgress< ValidateData^>	ValidateDatabaseAsync(uint16flags, String^) [page 36]	Performs a ValidateDatabase operation asynchronously.

In this section:

CancelGetNotification(String^) Method [page 17]
Cancels any pending get-notification calls on all queues matching the given name.

ChangeEncryptionKey(String^) Method [page 18]
Changes the database encryption key for an UltraLite database.

Checkpoint() Method [page 18]
Performs a checkpoint operation, flushing any pending committed transactions to the database.

CloseObject() Method [page 19]
Destroys this connection and any remaining associated objects.

Commit() Method [page 19]
Commits the current transaction.

CountUploadRows(String^, unsigned int) Method [page 19]
Counts the number of rows that need to be uploaded for synchronization.

CreateNotificationQueue(String^) Method [page 20]
Creates an event notification queue for this connection.

DeclareEvent(String^) Method [page 21]
Declares an event that can be registered for and triggered.

DestroyNotificationQueue(String^) Method [page 21]
Destroys the given event notification queue.

ExecuteStatement(String^) Method [page 22]
Executes a SQL statement string directly.

GetDatabaseProperty(String^) Method [page 22]
Obtains the value of a database property.

GetDatabasePropertyInt(String^) Method [page 23]
Obtains the integer value of a database property.

GetDatabaseSchema() Method [page 24]
Returns an object that is used to query the schema of the database.

GetLastDownloadTime(String^) Method [page 24]
Obtains the last time a specified publication was downloaded.

GetLastError(ULSqlCode *, String^*) Method [page 25]
Returns the error information associated with the last call on this Connection.

GetLastIdentity() Method [page 25]
Gets the @@identity value.

GetNotification(String^, unsigned int) Method [page 26]
Reads an event notification.

GetNotificationParameter(String^, String^) Method [page 26]
Gets a parameter for the event notification that was just read by the GetNotification method.

GetSyncResult() Method [page 27]
Gets the result of the last synchronization.

GlobalAutoIncrementUsage() Method [page 27]
Obtains the percent of the default values used in all the columns that have global autoincrement defaults.

GrantConnectTo(String^, String^) Method [page 28]
Grants access to an UltraLite database for a new or existing user ID with the given password.

OpenTable(String^, String^) Method [page 28]
Opens a table.

PrepareStatement(String^) Method [page 29]

Prepares a SQL statement.

RegisterForEvent(String^, String^, String^, bool) Method [page 29]
: Registers or unregisters a queue to receive notifications of an event.

ResetLastDownloadTime(String^) Method [page 30]
: Resets the last download time of a publication so that the application resynchronizes previously downloaded data.

RevokeConnectFrom(String^) Method [page 31]
: Revokes access from an UltraLite database for a user ID.

Rollback() Method [page 31]
: Rolls back the current transaction.

RollbackPartialDownload() Method [page 31]
: Rolls back the changes from a failed synchronization.

SendNotification(String^, String^, String^) Method [page 32]
: Sends a notification to all queues matching the given name.

SetDatabaseOption(String^, String^) Method [page 33]
: Sets the specified database option.

SetDatabaseOptionInt(String^, unsigned int) Method [page 33]
: Sets a database option.

StartSynchronizationDelete() Method [page 33]
: Sets START SYNCHRONIZATION DELETE for this connection.

StopSynchronizationDelete() Method [page 34]
: Sets STOP SYNCHRONIZATION DELETE for this connection.

Synchronize(String^, SyncObserver^) Method [page 34]
: Initiates synchronization in an UltraLite application.

SynchronizeAsync(String^) Method [page 34]
: Initiates synchronization in an UltraLite application.

TriggerEvent(String^, String^) Method [page 35]
: Triggers a user-defined event and sends notifications to all registered queues.

ValidateDatabase(uint16, String^, ValidateCallback^) Method [page 36]
: Validates the database on this connection.

ValidateDatabaseAsync(uint16flags, String^) Method [page 36]
: Performs a ValidateDatabase operation asynchronously.

1.1.1 CancelGetNotification(String^) Method

Cancels any pending get-notification calls on all queues matching the given name.

Syntax

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

Parameters

queueName The name of the queue.

Returns

The number of affected queues (not necessarily the number of blocked reads).

1.1.2 ChangeEncryptionKey(String^) Method

Changes the database encryption key for an UltraLite database.

Syntax

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

Parameters

newKey The new encryption key for the database.

Remarks

Applications that call this method must first ensure that the user has either synchronized the database or created a reliable backup copy of the database. You need a reliable backup of the database because the ChangeEncryptionKey method is an operation that must run to completion. When the database encryption key changes, every row in the database is first decrypted with the old key and then encrypted with the new key and rewritten. This operation is not recoverable. If the encryption change operation does not complete, then the database is left in an invalid state and you cannot access it again.

1.1.3 Checkpoint() Method

Performs a checkpoint operation, flushing any pending committed transactions to the database.

Syntax

```
public void Checkpoint ()
```

Remarks

Any current transaction is not committed by calling the Checkpoint method. This method is used in conjunction with deferring automatic transaction checkpoints (by using the commit_flush connection parameter) as a performance enhancement.

The Checkpoint method ensures that all pending committed transactions are written to the database.

1.1.4 CloseObject() Method

Destroys this connection and any remaining associated objects.

Syntax

```
public void CloseObject ()
```

1.1.5 Commit() Method

Commits the current transaction.

Syntax

```
public void Commit ()
```

1.1.6 CountUploadRows(String^, unsigned int) Method

Counts the number of rows that need to be uploaded for synchronization.

Syntax

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

Parameters

publList A string containing a comma-separated list of publications to check. An empty string (the SYNC_ALL constant) implies all tables except tables marked as no sync. A string containing just an asterisk (the SYNC_ALL_PUBS constant) implies all tables referred to in any publication. Some tables may not be part of any publication and are not included if this value is *.

threshold Determines the maximum number of rows to count, thereby limiting the amount of time taken by the method. A threshold of 0 corresponds to no limit (that is, count all rows that need to be synchronized) and a threshold of 1 can be used to quickly determine whether any rows need to be synchronized.

Returns

The number of rows that need to be synchronized, either in a specified set of publications or in the whole database.

Remarks

Use this method to prompt users to synchronize or to determine when automatic background synchronization should take place.

The following call checks the entire database for the total number of rows to be synchronized:

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

The following call checks publications PUB1 and PUB2 for a maximum of 1000 rows:

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

The following call checks to see if any rows need to be synchronized in publications PUB1 and PUB2:

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

1.1.7 CreateNotificationQueue(String^) Method

Creates an event notification queue for this connection.

Syntax

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

Parameters

name The name for the new queue.

Remarks

Queue names are scoped per connection, so different connections can create queues with the same name. When an event notification is sent, all queues in the database with a matching name receive a separate instance of the notification. Names are case insensitive. A default queue is created on demand for each connection when calling the RegisterForEvent method if no queue is specified. This call fails with an error if the name already exists or is not valid.

1.1.8 DeclareEvent(String^) Method

Declares an event that can be registered for and triggered.

Syntax

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

Parameters

eventName The name for the new user-defined event.

Remarks

UltraLite predefines system events that are triggered by operations on the database or the environment. This method declares user-defined events. User-defined events are triggered with the TriggerEvent method. The event name must be unique. Names are case insensitive.

1.1.9 DestroyNotificationQueue(String^) Method

Destroys the given event notification queue.

Syntax

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

Parameters

name The name of the queue to destroy.

Remarks

A warning is signaled if unread notifications remain in the queue. Unread notifications are discarded. A connection's default event queue, if created, is destroyed when the connection is closed.

1.1.10 ExecuteStatement(String^) Method

Executes a SQL statement string directly.

Syntax

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

Parameters

sql The SQL script to execute.

Remarks

Use this method to execute a SELECT statement directly and retrieve a single result.

Use the PrepareStatement method to execute a statement repeatedly with variable parameters, or to fetch multiple results.

1.1.11 GetDatabaseProperty(String^) Method

Obtains the value of a database property.

Syntax

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

Parameters

propName The name of the property being requested.

Returns

A string containing the database property value is returned when run successfully; otherwise, returns NULL.

Remarks

The returned value points to a static buffer whose contents may be changed by any subsequent UltraLite call, so you must make a copy of the value if you need to save it.

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

1.1.12 GetDatabasePropertyInt(String^) Method

Obtains the integer value of a database property.

Syntax

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

Parameters

propName The name of the property being requested.

Returns

If the call is successful, the integer value of the property; otherwise, returns 0.

Remarks

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

1.1.13 GetDatabaseSchema() Method

Returns an object that is used to query the schema of the database.

Syntax

```
public DatabaseSchema GetDatabaseSchema ()
```

Returns

A DatabaseSchema object used to query the schema of the database.

1.1.14 GetLastDownloadTime(String^) Method

Obtains the last time a specified publication was downloaded.

Syntax

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

Parameters

publication The publication name.

Returns

The last download time. The value January 1, 1900 indicates that the publication has yet to be synchronized or that the time was reset.

Remarks

The following call populates the dt structure with the date and time that the pub1 publication was downloaded:

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

1.1.15 GetLastError(ULSqlCode *, String^*) Method

Returns the error information associated with the last call on this Connection.

Syntax

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

Remarks

See DatabaseManager::GetLastError.

1.1.16 GetLastIdentity() Method

Gets the @@identity value.

Syntax

```
public uint64 GetLastIdentity ()
```

Returns

The last value inserted into an autoincrement or global autoincrement column.

Remarks

This value is the last value inserted into an autoincrement or global autoincrement column for the database. This value is not recorded when the database is shut down, so calling this method returns 0 before any autoincrement values have been inserted.

i Note

The last value inserted may have been on another connection.

1.1.17 GetNotification(String^, unsigned int) Method

Reads an event notification.

Syntax

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

Parameters

queueName The queue to read or NULL for the default connection queue.
waitms The time, in milliseconds, to wait (block) before returning.

Returns

The name of the event read or NULL on error.

Remarks

This call blocks the calling thread until a notification is received or until the given wait period expires.

To wait indefinitely, set the waitms parameter to UL_READ_WAIT_INFINITE.

To cancel a wait, send another notification to the given queue or use the CancelGetNotification method.

Use the GetNotificationParameter method after reading a notification to retrieve additional parameters by name.

1.1.18 GetNotificationParameter(String^, String^) Method

Gets a parameter for the event notification that was just read by the GetNotification method.

Syntax

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

Parameters

queueName The queue to read or NULL for default connection queue.
parameterName The name of the parameter to read (or *).

Returns

The parameter value or NULL on error.

Remarks

Only the parameters from the most recently read notification on the given queue are available. Parameters are retrieved by name. A parameter name of * retrieves the entire parameter string.

1.1.19 GetSyncResult() Method

Gets the result of the last synchronization.

Syntax

```
public SyncResult GetSyncResult ()
```

1.1.20 GlobalAutoIncrementUsage() Method

Obtains the percent of the default values used in all the columns that have global autoincrement defaults.

Syntax

```
public uint16 GlobalAutoIncrementUsage ()
```

Returns

The percent of the global autoincrement values used by the counter.

Remarks

If the database contains more than one column with this default, then this value is calculated for all columns and the maximum is returned. For example, a return value of 99 indicates that few default values remain for at least one of the columns.

1.1.21 GrantConnectTo(String^, String^) Method

Grants access to an UltraLite database for a new or existing user ID with the given password.

Syntax

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

Parameters

uid A String representing the user ID. The maximum length is 31 characters.
pwd A String representing the password for the user ID.

Remarks

This method updates the password for an existing user when you specify an existing user ID.

1.1.22 OpenTable(String^, String^) Method

Opens a table.

Syntax

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

Parameters

tableName The name of the table to open.
indexName The name of the index to open the table on. Pass null to open on the primary key and the empty string to open the table unordered.

Returns

The Table object when the call is successful; otherwise, returns NULL.

Remarks

The cursor position is set before the first row when the application first opens a table.

1.1.23 PrepareStatement(String^) Method

Prepares a SQL statement.

Syntax

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

Parameters

sql The SQL statement to prepare.

Returns

The PreparedStatement object on success; otherwise, returns NULL.

1.1.24 RegisterForEvent(String^, String^, String^, bool) Method

Registers or unregisters a queue to receive notifications of an event.

Syntax

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

Parameters

eventName The system- or user-defined event to register for.
objectName The object to which the event applies (for example, a table name).
queueName NULL means use the default connection queue.
register_not_unreg Set true to register or false to unregister.

Remarks

If no queue name is supplied, then the default connection queue is implied, and created if required. Certain system events allow you to specify an object name that the event applies to. For example, the TableModified event can specify the table name. Unlike the SendNotification method, only the specific queue registered receives notifications of the event. Other queues with the same name on different connections do not receive notifications unless they are also explicitly registered.

The predefined system events are:

- TableModified Triggered when rows in a table are inserted, updated, or deleted. One notification is sent per request, no matter how many rows were affected by the request. The object_name parameter specifies the table to monitor. A value of "*" means all tables in the database. This event has a parameter named table_name whose value is the name of the modified table.
- Commit Triggered after any commit completes. This event has no parameters.
- SyncComplete Triggered after synchronization completes. This event has no parameters.

1.1.25 ResetLastDownloadTime(String^) Method

Resets the last download time of a publication so that the application resynchronizes previously downloaded data.

Syntax

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

Parameters

pubList A string containing a comma-separated list of publications to reset. An empty string means that all tables are included except tables marked as no sync. A string containing just an asterisk (*) denotes all publications. Some tables may not be part of any publication and are not included if this value is *.

Remarks

The following method call resets the last download time for all tables:

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

1.1.26 RevokeConnectFrom(String^) Method

Revokes access from an UltraLite database for a user ID.

Syntax

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

Parameters

uid The user ID to be excluded from database access.

1.1.27 Rollback() Method

Rolls back the current transaction.

Syntax

```
public void Rollback ()
```

1.1.28 RollbackPartialDownload() Method

Rolls back the changes from a failed synchronization.

Syntax

```
public void RollbackPartialDownload ()
```

Remarks

When using resumable downloads (synchronizing with the keep-partial-download option turned on), and a communication error occurs during the download phase of synchronization, UltraLite retains the changes that were downloaded so that the synchronization can resume from the place where it was interrupted. Use this method to discard this partial download when you no longer need to attempt resuming.

This method has effect only when using resumable downloads.

1.1.29 SendNotification(String^, String^, String^) Method

Sends a notification to all queues matching the given name.

Syntax

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

Parameters

queueName The target queue name (or *).
eventName The identity for notification.
parameters The optional list of parameters.

Returns

The number of notifications sent (the number of matching queues).

Remarks

This includes any matching queue on the current connection. This call does not block. Use the special queue name "*" to send the notification to all queues. The given event name does not need to correspond to any system- or user- defined event; it is passed through to identify the notification when read and has meaning only to the sender and receiver.

The *parameters* value specifies a semicolon delimited list of name=value pairs. After the notification is read, the parameter values are read with the GetNotificationParameter method.

1.1.30 SetDatabaseOption(String^, String^) Method

Sets the specified database option.

Syntax

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

Parameters

optName The name of the option being set.
value The new value of the option.

1.1.31 SetDatabaseOptionInt(String^, unsigned int) Method

Sets a database option.

Syntax

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

Parameters

optName The name of the option being set.
value The new value of the option.

1.1.32 StartSynchronizationDelete() Method

Sets START SYNCHRONIZATION DELETE for this connection.

Syntax

```
public void StartSynchronizationDelete ()
```

1.1.33 StopSynchronizationDelete() Method

Sets STOP SYNCHRONIZATION DELETE for this connection.

Syntax

```
public void StopSynchronizationDelete ()
```

1.1.34 Synchronize(String^, SyncObserver^) Method

Initiates synchronization in an UltraLite application.

Syntax

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

Parameters

syncParms A semicolon delimited list of Synchronization Profile options.
observer The observer callback to send status updates to.

Remarks

This method initiates synchronization with a MobiLink server. This method does not return until synchronization is complete; however, additional threads on separate connections can access the database during synchronization.

Do not call this method from a UI thread.

1.1.35 SynchronizeAsync(String^) Method

Initiates synchronization in an UltraLite application.

Syntax

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

Parameters

syncParms A semicolon delimited list of Synchronization Profile options.

Remarks

This method initiates synchronization with a MobiLink server. This method is asynchronous. Additional threads on separate connections can access the database during synchronization.

1.1.36 TriggerEvent(String^, String^) Method

Triggers a user-defined event and sends notifications to all registered queues.

Syntax

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

Parameters

eventName The name of the system- or user-defined event to trigger.
parameters The optional list of parameters as semicolon delimited name=value pairs.

Returns

The number of event notifications sent.

Remarks

After the notification is read, the parameter values are read with GetNotificationParameter.

1.1.37 ValidateDatabase(uint16, String^, ValidateCallback^) Method

Validates the database on this connection.

Syntax

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

Parameters

flags A flag that controls the type of validation. See the example below.
tableName A specific table to validate or null.
cb A function to receive validation progress information.

Remarks

Tables, indexes, and database pages can be validated depending on the flags passed to this routine. To receive information during the validation, implement a callback function and pass the address to this routine. To limit the validation to a specific table, pass in the table name as the tableName parameter to this method.

The flags parameter is combination of the following values:

- ULVF_TABLE
- ULVF_INDEX
- ULVF_DATABASE
- ULVF_EXPRESS or ULVF_FULL_VALIDATE

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

1.1.38 ValidateDatabaseAsync(uint16flags, String^) Method

Performs a ValidateDatabase operation asynchronously.

Syntax

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

Returns

An awaitable asynchronous action with progress reporting.

1.2 Constants Class

Specifies constants that can be used for the UltraLite UWP APIs.

Namespace

```
UltraLite
```

Syntax

```
public ref class sealed
```

Members

All members of Constants, including inherited members.

Properties

Modifier and Type	Property	Description
public static property int64	BLOB_CONTINUE [page 38]	Used when reading data with the ResultSet.GetChars or ResultSet.GetBytes methods.
public static property String	SYNC_ALL [page 38]	Synchronizes all tables in the database that are not marked as no sync, including tables that are not in any publication.
public static property String	SYNC_ALL_PUBS [page 38]	Synchronizes all tables in a publication.

In this section:

BLOB_CONTINUE Property [page 38]
    Used when reading data with the ResultSet.GetChars or ResultSet.GetBytes methods.

SYNC_ALL Property [page 38]
    Synchronizes all tables in the database that are not marked as no sync, including tables that are not in any publication.

SYNC_ALL_PUBS Property [page 38]
Synchronizes all tables in a publication.

1.2.1 BLOB_CONTINUE Property

Used when reading data with the ResultSet.GetChars or ResultSet.GetBytes methods.

Syntax

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

Remarks

This value indicates that the chunk of data to be read should continue from where the last chunk was read.

See ResultSet::GetChars ResultSet::GetBytes.

1.2.2 SYNC_ALL Property

Synchronizes all tables in the database that are not marked as no sync, including tables that are not in any publication.

Syntax

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

1.2.3 SYNC_ALL_PUBS Property

Synchronizes all tables in a publication.

Syntax

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

1.3 Cursor Interface

Represents a cursor in an UltraLite database.

Namespace

```
UltraLite
```

Syntax

```
public interface class
```

Members

All members of Cursor, including inherited members.

Methods

Modifier and Type	Method	Description
public void	AfterLast() [page 44]	Moves the cursor after the last row.
public void	AppendBytes [page 44]	Appends bytes to a column.
public void	AppendChars [page 46]	Appends a string chunk to a column.
public void	BeforeFirst() [page 47]	Moves the cursor before the first row.
public void	Close() [page 48]	Destroys this object.
public void	Delete() [page 48]	Deletes the current row and moves the cursor to the next valid row.
public void	DeleteNamed(String^) [page 48]	Deletes the current row and moves the cursor to the next valid row.
public void	First() [page 48]	Moves the cursor to the first row.
public int64	GetBinaryLength [page 49]	Gets the binary length of the value of a column.
public bool	GetBool [page 50]	Fetches a value from a column as a boolean.
public uint8	GetByte [page 52]	Fetches a value from a column as a byte.
public int64	GetBytes [page 53]	Gets a binary chunk from the column.
public int64	GetChars [page 55]	Gets a wide string chunk from the column.

Modifier and Type	Method	Description
public DateTime	GetDateTime [page 57]	Fetches a value from a column as a DateTime.
public double	GetDouble [page 59]	Fetches a value from a column as a double.
public float	GetFloat [page 60]	Fetches a value from a column as a float.
public Guid	GetGuid [page 62]	Fetches a value from a column as a GUID.
public int16	GetInt16 [page 63]	Fetches a value from a column as a 16-bit integer.
public int	GetInt32 [page 65]	Fetches a value from a column as an integer.
public int64	GetInt64 [page 66]	Fetches a value from a column as a 64-bit integer.
public unsigned int	GetRowCount(unsigned int) [page 68]	Gets the number of rows in the table.
public uint8	GetState() [page 68]	Gets the internal state of the cursor.
public String	GetString [page 69]	Fetches a value from a column as a string.
public int64	GetStringLength [page 70]	Gets the string length of the value of a column.
public uint16	GetUInt16 [page 72]	Fetches a value from a column as a 16-bit unsigned integer.
public unsigned int	GetUInt32 [page 73]	Fetches a value from a column as a 32-bit unsigned integer.
public uint64	GetUInt64 [page 75]	Fetches a value from a column as a 64-bit unsigned integer.
public bool	IsNull [page 76]	Checks whether a column value is NULL.
public void	Last() [page 78]	Moves the cursor to the last row.
public bool	Next() [page 78]	Moves the cursor forward one row.
public bool	Previous() [page 78]	Moves the cursor back one row.
public void	Relative(int) [page 79]	Moves the cursor by offset rows from the current cursor position.
public void	SetBool [page 79]	Sets a column to a boolean value.
public void	SetByte [page 80]	Sets a column to a byte value.
public void	SetBytes [page 82]	Sets a column to a binary value.
public void	SetDateTime [page 83]	Sets a column to a DateTime value.
public void	SetDefault [page 84]	Sets a column to its default value.
public void	SetDouble [page 86]	Sets a column to a double value.
public void	SetFloat [page 87]	Sets a column to a float value.

Modifier and Type	Method	Description
public void	SetGuid [page 88]	Sets a column to a GUID value.
public void	SetInt16 [page 89]	Sets a column to an integer value.
public void	SetInt32 [page 91]	Sets a column to an integer value.
public void	SetInt64 [page 92]	Sets a column to an integer value.
public void	SetNull [page 93]	Sets a column to null.
public void	SetString [page 94]	Sets a column to a string value.
public void	SetUInt16 [page 96]	Sets a column to an unsigned 16-bit integer value.
public void	SetUInt32 [page 97]	Sets a column to an unsigned 32-bit integer value.
public void	SetUInt64 [page 99]	Sets a column to an unsigned 64-bit integer value.
public void	Update() [page 100]	Updates the current row.
public void	UpdateBegin() [page 100]	Selects the update mode for setting columns.

Remarks

This interface is implemented by the ResultSet and Table classes.

In this section:

AfterLast() Method [page 44]
  Moves the cursor after the last row.

AppendBytes Method [page 44]
  Appends bytes to a column.

AppendChars Method [page 46]
  Appends a string chunk to a column.

BeforeFirst() Method [page 47]
  Moves the cursor before the first row.

CloseObject() Method [page 48]
  Destroys this object.

Delete() Method [page 48]
  Deletes the current row and moves the cursor to the next valid row.

DeleteNamed(String^) Method [page 48]
  Deletes the current row and moves the cursor to the next valid row.

First() Method [page 48]
  Moves the cursor to the first row.

GetBinaryLength Method [page 49]

Gets the binary length of the value of a column.

GetBool Method [page 50]
: Fetches a value from a column as a boolean.

GetByte Method [page 52]
: Fetches a value from a column as a byte.

GetBytes Method [page 53]
: Gets a binary chunk from the column.

GetChars Method [page 55]
: Gets a wide string chunk from the column.

GetDateTime Method [page 57]
: Fetches a value from a column as a DateTime.

GetDouble Method [page 59]
: Fetches a value from a column as a double.

GetFloat Method [page 60]
: Fetches a value from a column as a float.

GetGuid Method [page 62]
: Fetches a value from a column as a GUID.

GetInt16 Method [page 63]
: Fetches a value from a column as a 16-bit integer.

GetInt32 Method [page 65]
: Fetches a value from a column as a 32-bit signed integer.

GetInt64 Method [page 66]
: Fetches a value from a column as a 64-bit signed integer.

GetRowCount(unsigned int) Method [page 68]
: Gets the number of rows in the table.

GetState() Method [page 68]
: Gets the internal state of the cursor.

GetString Method [page 69]
: Fetches a value from a column as a string.

GetStringLength Method [page 70]
: Gets the string length of the value of a column.

GetUInt16 Method [page 72]
: Fetches a value from a column as a 16-bit unsigned integer.

GetUInt32 Method [page 73]
: Fetches a value from a column as a 32-bit unsigned integer.

GetUInt64 Method [page 75]
: Fetches a value from a column as a 64-bit unsigned integer.

IsNull Method [page 76]
: Checks whether a column value is NULL.

Last() Method [page 78]
: Moves the cursor to the last row.

Next() Method [page 78]
  Moves the cursor forward one row.

Previous() Method [page 78]
  Moves the cursor back one row.

Relative(int) Method [page 79]
  Moves the cursor by offset rows from the current cursor position.

SetBool Method [page 79]
  Sets a column to a boolean value.

SetByte Method [page 80]
  Sets a column to a byte value.

SetBytes Method [page 82]
  Sets a column to a BINARY value.

SetDateTime Method [page 83]
  Sets a column to a DateTime value.

SetDefault Method [page 84]
  Sets a column to its default value.

SetDouble Method [page 86]
  Sets a column to a double value.

SetFloat Method [page 87]
  Sets a column to a float value.

SetGuid Method [page 88]
  Sets a column to a GUID value.

SetInt16 Method [page 89]
  Sets a column to a 16-bit signed integer value.

SetInt32 Method [page 91]
  Sets a column to a 32-bit signed integer value.

SetInt64 Method [page 92]
  Sets a column to an integer value.

SetNull Method [page 93]
  Sets a column to null.

SetString Method [page 94]
  Sets a column to a string value.

SetUInt16 Method [page 96]
  Sets a column to an unsigned 16-bit integer value.

SetUInt32 Method [page 97]
  Sets a column to an unsigned 32-bit integer value.

SetUInt64 Method [page 99]
  Sets a column to an unsigned 64-bit integer value.

Update() Method [page 100]
  Updates the current row.

UpdateBegin() Method [page 100]
  Start update mode for setting columns.

1.3.1 AfterLast() Method

Moves the cursor after the last row.

Syntax

```
public void AfterLast ()
```

1.3.2 AppendBytes Method

Appends bytes to a column.

Overload list

Modifier and Type	Overload name	Description
public void	AppendBytes(String^, const Array< uint8 >^, int, int) [page 44]	Appends bytes to a column.
public void	AppendBytes(uint16, const Array< uint8 >^, int, int) [page 45]	Appends bytes to a column.

In this section:

AppendBytes(String^, const Array< uint8 >^, int, int) Method [page 44]
    Appends bytes to a column.

AppendBytes(uint16, const Array< uint8 >^, int, int) Method [page 45]
    Appends bytes to a column.

1.3.2.1 AppendBytes(String^, const Array< uint8 >^, int, int) Method

Appends bytes to a column.

Syntax

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

Parameters

cname The name of the column.
value The byte chunk to append.
offset The offset into the byte chunk at which to start.
size The size of the byte chunk in bytes.

Remarks

The given bytes are appended to the end of the column written so far by AppendBytes method calls.

1.3.2.2 AppendBytes(uint16, const Array< uint8 >^, int, int) Method

Appends bytes to a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The byte chunk to append.
offset The offset into the byte chunk at which to start.
size The size of the byte chunk in bytes.

Remarks

The given bytes are appended to the end of the column written so far by AppendBytes method calls.

1.3.3 AppendChars Method

Appends a string chunk to a column.

Overload list

Modifier and Type	Overload name	Description
public void	AppendChars(String^, const Array< wchar_t >^, int, int) [page 46]	Appends a string chunk to a column.
public void	AppendChars(uint16, const Array< wchar_t >^, int, int) [page 47]	Appends a string chunk to a column.

In this section:

AppendChars(String^, const Array< wchar_t >^, int, int) Method [page 46]
  Appends a string chunk to a column.

AppendChars(uint16, const Array< wchar_t >^, int, int) Method [page 47]
  Appends a string chunk to a column.

1.3.3.1 AppendChars(String^, const Array< wchar_t >^, int, int) Method

Appends a string chunk to a column.

Syntax

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

Parameters

cname The name of the column.
value The string chunk to append.
offset The offset into the string chunk at which to start.
size The length of the string chunk in characters.

Remarks

This method appends the given string to the end of the string written so far by AppendChars method calls.

1.3.3.2 AppendChars(uint16, const Array< wchar_t >^, int, int) Method

Appends a string chunk to a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The string chunk to append.
offset The offset into the string chunk at which to start.
size The length of the string chunk in characters.

Remarks

This method appends the given string to the end of the string written so far by AppendChars method calls.

1.3.4 BeforeFirst() Method

Moves the cursor before the first row.

Syntax

```
public void BeforeFirst ()
```

1.3.5 CloseObject() Method

Destroys this object.

Syntax

```
public void CloseObject ()
```

1.3.6 Delete() Method

Deletes the current row and moves the cursor to the next valid row.

Syntax

```
public void Delete ()
```

1.3.7 DeleteNamed(String^) Method

Deletes the current row and moves the cursor to the next valid row.

Syntax

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

Parameters

tableName A table name or its correlation (required when the database has multiple columns that share the same table name).

1.3.8 First() Method

Moves the cursor to the first row.

Syntax

```
public void First ()
```

1.3.9 GetBinaryLength Method

Gets the binary length of the value of a column.

Overload list

Modifier and Type	Overload name	Description
public int64	GetBinaryLength(String^) [page 49]	Gets the binary length of the value of a column.
public int64	GetBinaryLength(uint16) [page 50]	Gets the binary length of the value of a column.

In this section:

GetBinaryLength(String^) Method [page 49]
Gets the binary length of the value of a column.

GetBinaryLength(uint16) Method [page 50]
Gets the binary length of the value of a column.

1.3.9.1 GetBinaryLength(String^) Method

Gets the binary length of the value of a column.

Syntax

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

Parameters

cname The name of the column.

Returns

The length of the column value as a binary.

1.3.9.2 GetBinaryLength(uint16) Method

Gets the binary length of the value of a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The length of the column value as a binary.

1.3.10 GetBool Method

Fetches a value from a column as a boolean.

Overload list

Modifier and Type	Overload name	Description
public bool	GetBool(String^) [page 51]	Fetches a value from a column as a boolean.
public bool	GetBool(uint16) [page 51]	Fetches a value from a column as a boolean.

In this section:

GetBool(String^) Method [page 51]
Fetches a value from a column as a boolean.

GetBool(uint16) Method [page 51]
Fetches a value from a column as a boolean.

1.3.10.1 GetBool(String^) Method

Fetches a value from a column as a boolean.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a boolean.

1.3.10.2 GetBool(uint16) Method

Fetches a value from a column as a boolean.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a boolean.

1.3.11 GetByte Method

Fetches a value from a column as a byte.

Overload list

Modifier and Type	Overload name	Description
public uint8	GetByte(String^) [page 52]	Fetches a value from a column as a byte.
public uint8	GetByte(uint16) [page 53]	Fetches a value from a column as a byte.

In this section:

GetByte(String^) Method [page 52]
    Fetches a value from a column as a byte.

GetByte(uint16) Method [page 53]
    Fetches a value from a column as a byte.

1.3.11.1 GetByte(String^) Method

Fetches a value from a column as a byte.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a byte.

1.3.11.2 GetByte(uint16) Method

Fetches a value from a column as a byte.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a byte.

1.3.12 GetBytes Method

Gets a binary chunk from the column.

Overload list

Modifier and Type	Overload name	Description
public int64	GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) [page 54]	Gets a binary chunk from the column.
public int64	GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) [page 54]	Gets a binary chunk from the column.

In this section:

GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) Method [page 54]
Gets a binary chunk from the column.

GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) Method [page 54]
Gets a binary chunk from the column.

1.3.12.1 GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) Method

Gets a binary chunk from the column.

Syntax

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

Parameters

cname The name of the column.
dst The buffer to hold the bytes.
count The size of the buffer in bytes.
srcOffset The offset into the value at which to start reading, or the BLOB_CONTINUE constant to continue from where the last read ended.
dstOffset The offset into the destination buffer at which to copy the bytes.

Returns

The number of bytes copied to the destination buffer. If the dst value is NULL, then the number of bytes left is returned. An empty string is returned in the dst parameter when the column is null.

Remarks

The end of the value has been reached if 0 is returned.

Use the IsNull method to differentiate between null and empty strings.

1.3.12.2 GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) Method

Gets a binary chunk from the column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

dst The buffer to hold the bytes.

count The size of the buffer in bytes.

srcOffset The offset into the value at which to start reading, or the BLOB_CONTINUE constant to continue from where the last read ended.

dstOffset The offset into the destination buffer at which to copy the bytes.

Returns

The number of bytes copied to the destination buffer. If the dst value is NULL, then this method returns the number of bytes left. An empty string is returned in the dst parameter when the column is null.

Remarks

The end of the value has been reached if 0 is returned.

Use the IsNull method to differentiate between null and empty strings.

1.3.13 GetChars Method

Gets a wide string chunk from the column.

Overload list

Modifier and Type	Overload name	Description
public int64	GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) [page 56]	Gets a wide string chunk from the column.
public int64	GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) [page 56]	Gets a wide string chunk from the column.

In this section:

GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) Method [page 56]
Gets a wide string chunk from the column.

GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) Method [page 56]
Gets a wide string chunk from the column.

1.3.13.1 GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) Method

Gets a wide string chunk from the column.

Syntax

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

Parameters

cname The name of the column.

dst The buffer to hold the string chunk. The string is null terminated even if it is truncated.

count The size, in characters, of the buffer.

srcOffset The offset into the value at which to start reading, or the BLOB_CONTINUE constant to continue from where the last read ended.

dstOffset The offset into the destination buffer at which to copy the characters.

Returns

The number of characters copied to the destination buffer excluding the null terminator. If the dst value is NULL, then this method returns the number of characters left in the string. An empty string is returned in the dst parameter when the column is null.

Remarks

The end of the value has been reached if 0 is returned.

Use the IsNull method to differentiate between null and empty strings.

1.3.13.2 GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) Method

Gets a wide string chunk from the column.

Syntax

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

Parameters

cid The zero-based ordinal column number.
dst The buffer to hold the string chunk. The string is null terminated even if it is truncated.
count The size, in characters, of the buffer.
srcOffset The offset into the value at which to start reading, or the BLOB_CONTINUE constant to continue from where the last read ended.
dstOffset The offset into the destination buffer at which to copy the characters.

Returns

The number of characters copied to the destination buffer excluding the null terminator. If the dst value is NULL, then this method returns the number of characters left in the string. An empty string is returned in the dst parameter when the column is null.

Remarks

The end of the value has been reached if 0 is returned.

Use the IsNull method to differentiate between null and empty strings.

1.3.14 GetDateTime Method

Fetches a value from a column as a DateTime.

Overload list

Modifier and Type	Overload name	Description
public DateTime	GetDateTime(String^) [page 58]	Fetches a value from a column as a DateTime.
public DateTime	GetDateTime(uint16) [page 58]	Fetches a value from a column as a DateTime.

In this section:

GetDateTime(String^) Method [page 58]
Fetches a value from a column as a DateTime.

GetDateTime(uint16) Method [page 58]

Fetches a value from a column as a DateTime.

1.3.14.1 GetDateTime(String^) Method

Fetches a value from a column as a DateTime.

Syntax

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

Parameters

cname The name of the column.

Returns

The DateTime value.

1.3.14.2 GetDateTime(uint16) Method

Fetches a value from a column as a DateTime.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The DateTime value.

1.3.15 GetDouble Method

Fetches a value from a column as a double.

Overload list

Modifier and Type	Overload name	Description
public double	GetDouble(String^) [page 59]	Fetches a value from a column as a double.
public double	GetDouble(uint16) [page 60]	Fetches a value from a column as a double.

In this section:

GetDouble(String^) Method [page 59]
Fetches a value from a column as a double.

GetDouble(uint16) Method [page 60]
Fetches a value from a column as a double.

1.3.15.1 GetDouble(String^) Method

Fetches a value from a column as a double.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a double.

1.3.15.2 GetDouble(uint16) Method

Fetches a value from a column as a double.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a double.

1.3.16 GetFloat Method

Fetches a value from a column as a float.

Overload list

Modifier and Type	Overload name	Description
public float	GetFloat(String^) [page 61]	Fetches a value from a column as a float.
public float	GetFloat(uint16) [page 61]	Fetches a value from a column as a float.

In this section:

GetFloat(String^) Method [page 61]
Fetches a value from a column as a float.

GetFloat(uint16) Method [page 61]
Fetches a value from a column as a float.

1.3.16.1 GetFloat(String^) Method

Fetches a value from a column as a float.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a float.

1.3.16.2 GetFloat(uint16) Method

Fetches a value from a column as a float.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a float.

1.3.17 GetGuid Method

Fetches a value from a column as a GUID.

Overload list

Modifier and Type	Overload name	Description
public Guid	GetGuid(String^) [page 62]	Fetches a value from a column as a GUID.
public Guid	GetGuid(uint16) [page 63]	Fetches a value from a column as a GUID.

In this section:

GetGuid(String^) Method [page 62]
Fetches a value from a column as a GUID.

GetGuid(uint16) Method [page 63]
Fetches a value from a column as a GUID.

1.3.17.1 GetGuid(String^) Method

Fetches a value from a column as a GUID.

Syntax

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

Parameters

cname The name of the column.

Returns

The GUID value.

1.3.17.2 GetGuid(uint16) Method

Fetches a value from a column as a GUID.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The GUID value.

1.3.18 GetInt16 Method

Fetches a value from a column as a 16-bit integer.

Overload list

Modifier and Type	Overload name	Description
public int16	GetInt16(String^) [page 64]	Fetches a value from a column as a 16-bit integer.
public int16	GetInt16(uint16) [page 64]	Fetches a value from a column as a 16-bit integer.

In this section:

GetInt16(String^) Method [page 64]
Fetches a value from a column as a 16-bit integer.

GetInt16(uint16) Method [page 64]
Fetches a value from a column as a 16-bit integer.

1.3.18.1 GetInt16(String^) Method

Fetches a value from a column as a 16-bit integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as an integer.

1.3.18.2 GetInt16(uint16) Method

Fetches a value from a column as a 16-bit integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as an integer.

1.3.19 GetInt32 Method

Fetches a value from a column as a 32-bit signed integer.

Overload list

Modifier and Type	Overload name	Description
public int	GetInt32(String^) [page 65]	Fetches a value from a column as a 32-bit signed integer.
public int	GetInt32(uint16) [page 66]	Fetches a value from a column as a 32-bit signed integer.

In this section:

GetInt32(String^) Method [page 65]
Fetches a value from a column as a 32-bit signed integer.

GetInt32(uint16) Method [page 66]
Fetches a value from a column as a 32-bit signed integer.

1.3.19.1 GetInt32(String^) Method

Fetches a value from a column as a 32-bit signed integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a 32-bit signed integer.

1.3.19.2 GetInt32(uint16) Method

Fetches a value from a column as a 32-bit signed integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a 32-bit signed integer.

1.3.20 GetInt64 Method

Fetches a value from a column as a 64-bit signed integer.

Overload list

Modifier and Type	Overload name	Description
public int64	GetInt64(String^) [page 67]	Fetches a value from a column as a 64-bit signed integer.
public int64	GetInt64(uint16) [page 67]	Fetches a value from a column as a 64-bit signed integer.

In this section:

GetInt64(String^) Method [page 67]
Fetches a value from a column as a 64-bit signed integer.

GetInt64(uint16) Method [page 67]
Fetches a value from a column as a 64-bit signed integer.

1.3.20.1 GetInt64(String^) Method

Fetches a value from a column as a 64-bit signed integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a 64-bit signed integer.

1.3.20.2 GetInt64(uint16) Method

Fetches a value from a column as a 64-bit signed integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a 64-bit signed integer.

1.3.21 GetRowCount(unsigned int) Method

Gets the number of rows in the table.

Syntax

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

Parameters

threshold The limit on the number of rows to count. Set to 0 to indicate no limit.

Returns

The number of rows in the table.

Remarks

This method is equivalent to executing the following statement:

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

1.3.22 GetState() Method

Gets the internal state of the cursor.

Syntax

```
public uint8 GetState ()
```

Returns

The state of the cursor.

1.3.23 GetString Method

Fetches a value from a column as a string.

Overload list

Modifier and Type	Overload name	Description
public String	GetString(String^) [page 69]	Fetches a value from a column as a string.
public String	GetString(uint16) [page 70]	Fetches a value from a column as a string.

In this section:

GetString(String^) Method [page 69]
Fetches a value from a column as a string.

GetString(uint16) Method [page 70]
Fetches a value from a column as a string.

1.3.23.1 GetString(String^) Method

Fetches a value from a column as a string.

Syntax

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

Parameters

cname The name of the column.

Returns

The string value.

1.3.23.2 GetString(uint16) Method

Fetches a value from a column as a string.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The string value.

1.3.24 GetStringLength Method

Gets the string length of the value of a column.

Overload list

Modifier and Type	Overload name	Description
public int64	GetStringLength(String^) [page 71]	Gets the string length of the value of a column.
public int64	GetStringLength(uint16) [page 71]	Gets the string length of the value of a column.

In this section:

GetStringLength(String^) Method [page 71]
  Gets the string length of the value of a column.

GetStringLength(uint16) Method [page 71]
  Gets the string length of the value of a column.

1.3.24.1 GetStringLength(String^) Method

Gets the string length of the value of a column.

Syntax

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

Parameters

cname The name of the column.

Returns

The number of characters of a string type column value. Returns 0 if the value is null or an error occurs.

1.3.24.2 GetStringLength(uint16) Method

Gets the string length of the value of a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The number of characters of a string type column value.

1.3.25 GetUInt16 Method

Fetches a value from a column as a 16-bit unsigned integer.

Overload list

Modifier and Type	Overload name	Description
public uint16	GetUInt16(String^) [page 72]	Fetches a value from a column as a 16-bit unsigned integer.
public uint16	GetUInt16(uint16) [page 73]	Fetches a value from a column as a 16-bit unsigned integer.

In this section:

GetUInt16(String^) Method [page 72]
Fetches a value from a column as a 16-bit unsigned integer.

GetUInt16(uint16) Method [page 73]
Fetches a value from a column as a 16-bit unsigned integer.

1.3.25.1 GetUInt16(String^) Method

Fetches a value from a column as a 16-bit unsigned integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a 16-bit unsigned integer.

1.3.25.2 GetUInt16(uint16) Method

Fetches a value from a column as a 16-bit unsigned integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a 16-bit unsigned integer.

1.3.26 GetUInt32 Method

Fetches a value from a column as a 32-bit unsigned integer.

Overload list

Modifier and Type	Overload name	Description
public unsigned int	GetUInt32(String^) [page 74]	Fetches a value from a column as a 32-bit unsigned integer.
public unsigned int	GetUInt32(uint16) [page 74]	Fetches a value from a column as a 32-bit unsigned integer.

In this section:

GetUInt32(String^) Method [page 74]
Fetches a value from a column as a 32-bit unsigned integer.

GetUInt32(uint16) Method [page 74]
Fetches a value from a column as a 32-bit unsigned integer.

1.3.26.1 GetUInt32(String^) Method

Fetches a value from a column as a 32-bit unsigned integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a 32-bit unsigned integer.

1.3.26.2 GetUInt32(uint16) Method

Fetches a value from a column as a 32-bit unsigned integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a 32-bit unsigned integer.

1.3.27 GetUInt64 Method

Fetches a value from a column as a 64-bit unsigned integer.

Overload list

Modifier and Type	Overload name	Description
public uint64	GetUInt64(String^) [page 75]	Fetches a value from a column as a 64-bit unsigned integer.
public uint64	GetUInt64(uint16) [page 76]	Fetches a value from a column as a 64-bit unsigned integer.

In this section:

GetUInt64(String^) Method [page 75]
Fetches a value from a column as a 64-bit unsigned integer.

GetUInt64(uint16) Method [page 76]
Fetches a value from a column as a 64-bit unsigned integer.

1.3.27.1 GetUInt64(String^) Method

Fetches a value from a column as a 64-bit unsigned integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a 64-bit unsigned integer.

1.3.27.2 GetUInt64(uint16) Method

Fetches a value from a column as a 64-bit unsigned integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a 64-bit unsigned integer.

1.3.28 IsNull Method

Checks whether a column value is NULL.

Overload list

Modifier and Type	Overload name	Description
public bool	IsNull(String^) [page 77]	Checks whether a column value is NULL.
public bool	IsNull(uint16) [page 77]	Checks whether a column value is NULL.

In this section:

IsNull(String^) Method [page 77]
Checks whether a column value is NULL.

IsNull(uint16) Method [page 77]
Checks whether a column value is NULL.

1.3.28.1 IsNull(String^) Method

Checks whether a column value is NULL.

Syntax

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

Parameters

cname The name of the column.

Returns

True if the column value is NULL.

1.3.28.2 IsNull(uint16) Method

Checks whether a column value is NULL.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

True if the column value is NULL.

1.3.29 Last() Method

Moves the cursor to the last row.

Syntax

```
public void Last ()
```

1.3.30 Next() Method

Moves the cursor forward one row.

Syntax

```
public bool Next ()
```

Returns

True if the cursor successfully moves forward. An error can still be signaled when the cursor moves successfully to the next row. For example, there could be conversion errors while evaluating the SELECT expressions. In this case, errors are also returned when retrieving the column values. False is returned if the cursor fails to move forward. For example, there is not a next row. In this case, the resulting cursor position is set after the last row.

1.3.31 Previous() Method

Moves the cursor back one row.

Syntax

```
public bool Previous ()
```

Returns

True if the cursor successfully moves back one row. False if it fails to move backward. The resulting cursor position is set before the first row.

1.3.32 Relative(int) Method

Moves the cursor by offset rows from the current cursor position.

Syntax

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

Parameters

offset The number of rows to move.

1.3.33 SetBool Method

Sets a column to a boolean value.

Overload list

Modifier and Type	Overload name	Description
public void	SetBool(String^, bool) [page 79]	Sets a column to a boolean value.
public void	SetBool(uint16, bool) [page 80]	Sets a column to a boolean value.

In this section:

SetBool(String^, bool) Method [page 79]
Sets a column to a boolean value.

SetBool(uint16, bool) Method [page 80]
Sets a column to a boolean value.

1.3.33.1 SetBool(String^, bool) Method

Sets a column to a boolean value.

Syntax

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

Parameters

cname The name of the column.
value The boolean value.

1.3.33.2 SetBool(uint16, bool) Method

Sets a column to a boolean value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The boolean value.

1.3.34 SetByte Method

Sets a column to a byte value.

Overload list

Modifier and Type	Overload name	Description
public void	SetByte(String^, uint8) [page 81]	Sets a column to a byte value.
public void	SetByte(uint16, uint8) [page 81]	Sets a column to a byte value.

In this section:

SetByte(String^, uint8) Method [page 81]
Sets a column to a byte value.

SetByte(uint16, uint8) Method [page 81]
Sets a column to a byte value.

1.3.34.1 SetByte(String^, uint8) Method

Sets a column to a byte value.

Syntax

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

Parameters

cname The name of the column.
value The byte value.

1.3.34.2 SetByte(uint16, uint8) Method

Sets a column to a byte value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The byte value.

1.3.35 SetBytes Method

Sets a column to a BINARY value.

Overload list

Modifier and Type	Overload name	Description
public void	SetBytes(String^, const Array< uint8 >^, int) [page 82]	Sets a column to a BINARY value.
public void	SetBytes(uint16, const Array< uint8 >^, int) [page 83]	Sets a column to a BINARY value.

In this section:

SetBytes(String^, const Array< uint8 >^, int) Method [page 82]
Sets a column to a BINARY value.

SetBytes(uint16, const Array< uint8 >^, int) Method [page 83]
Sets a column to a BINARY value.

1.3.35.1 SetBytes(String^, const Array< uint8 >^, int) Method

Sets a column to a BINARY value.

Syntax

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

Parameters

cname The name of the column.
value The binary value. Passing NULL is equivalent to calling the SetNull method.
length The length in bytes to get from the byte array value.

1.3.35.2 SetBytes(uint16, const Array< uint8 >^, int) Method

Sets a column to a BINARY value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The binary value. Passing NULL is equivalent to calling the SetNull method.
length The length in bytes to get from the byte array value.

1.3.36 SetDateTime Method

Sets a column to a DateTime value.

Overload list

Modifier and Type	Overload name	Description
public void	SetDateTime(String^, DateTime) [page 84]	Sets a column to a DateTime value.
public void	SetDateTime(uint16, DateTime) [page 84]	Sets a column to a DateTime value.

In this section:

SetDateTime(String^, DateTime) Method [page 84]
Sets a column to a DateTime value.

SetDateTime(uint16, DateTime) Method [page 84]
Sets a column to a DateTime value.

1.3.36.1 SetDateTime(String^, DateTime) Method

Sets a column to a DateTime value.

Syntax

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

Parameters

cname The name of the column.
value The DateTime value.

1.3.36.2 SetDateTime(uint16, DateTime) Method

Sets a column to a DateTime value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The DateTime value.

1.3.37 SetDefault Method

Sets a column to its default value.

Overload list

Modifier and Type	Overload name	Description
public void	SetDefault(String^) [page 85]	Sets a column to its default value.

Modifier and Type	Overload name	Description
public void	SetDefault(uint16) [page 85]	Sets a column to its default value.

In this section:

SetDefault(String^) Method [page 85]
Sets a column to its default value.

SetDefault(uint16) Method [page 85]
Sets a column to its default value.

1.3.37.1 SetDefault(String^) Method

Sets a column to its default value.

Syntax

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

Parameters

cname The name of the column.

1.3.37.2 SetDefault(uint16) Method

Sets a column to its default value.

Syntax

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

Parameters

cid The zero-based ordinal column number.

1.3.38 SetDouble Method

Sets a column to a double value.

Overload list

Modifier and Type	Overload name	Description
public void	SetDouble(String^, double) [page 86]	Sets a column to a double value.
public void	SetDouble(uint16, double) [page 86]	Sets a column to a double value.

In this section:

SetDouble(String^, double) Method [page 86]
Sets a column to a double value.

SetDouble(uint16, double) Method [page 86]
Sets a column to a double value.

1.3.38.1 SetDouble(String^, double) Method

Sets a column to a double value.

Syntax

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

Parameters

cname The name of the column.
value The double value.

1.3.38.2 SetDouble(uint16, double) Method

Sets a column to a double value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The double value.

1.3.39 SetFloat Method

Sets a column to a float value.

Overload list

Modifier and Type	Overload name	Description
public void	SetFloat(String^, float) [page 87]	Sets a column to a float value.
public void	SetFloat(uint16, float) [page 88]	Sets a column to a float value.

In this section:

SetFloat(String^, float) Method [page 87]
Sets a column to a float value.

SetFloat(uint16, float) Method [page 88]
Sets a column to a float value.

1.3.39.1 SetFloat(String^, float) Method

Sets a column to a float value.

Syntax

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

Parameters

cname The name of the column.
value The float value.

1.3.39.2 SetFloat(uint16, float) Method

Sets a column to a float value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The float value.

1.3.40 SetGuid Method

Sets a column to a GUID value.

Overload list

Modifier and Type	Overload name	Description
public void	SetGuid(String^, Guid) [page 88]	Sets a column to a GUID value.
public void	SetGuid(uint16, Guid) [page 89]	Sets a column to a GUID value.

In this section:

SetGuid(String^, Guid) Method [page 88]
    Sets a column to a GUID value.

SetGuid(uint16, Guid) Method [page 89]
    Sets a column to a GUID value.

1.3.40.1 SetGuid(String^, Guid) Method

Sets a column to a GUID value.

Syntax

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

Parameters

cname The name of the column.
value The GUID value.

1.3.40.2 SetGuid(uint16, Guid) Method

Sets a column to a GUID value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The GUID value.

1.3.41 SetInt16 Method

Sets a column to a 16-bit signed integer value.

Overload list

Modifier and Type	Overload name	Description
public void	SetInt16(String^, int16) [page 90]	Sets a column to a 16-bit signed integer value.
public void	SetInt16(uint16, int16) [page 90]	Sets a column to a 16-bit signed integer value.

In this section:

SetInt16(String^, int16) Method [page 90]
    Sets a column to a 16-bit signed integer value.

SetInt16(uint16, int16) Method [page 90]
    Sets a column to a 16-bit signed integer value.

1.3.41.1 SetInt16(String^, int16) Method

Sets a column to a 16-bit signed integer value.

Syntax

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

Parameters

cname The name of the column.
value The 16-bit signed integer value.

1.3.41.2 SetInt16(uint16, int16) Method

Sets a column to a 16-bit signed integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The 16-bit signed integer value.

1.3.42 SetInt32 Method

Sets a column to a 32-bit signed integer value.

Overload list

Modifier and Type	Overload name	Description
public void	SetInt32(String^, int) [page 91]	Sets a column to a 32-bit signed integer value.
public void	SetInt32(uint16, int) [page 92]	Sets a column to a 32-bit signed integer value.

In this section:

SetInt32(String^, int) Method [page 91]
    Sets a column to a 32-bit signed integer value.

SetInt32(uint16, int) Method [page 92]
    Sets a column to an integer value.

1.3.42.1 SetInt32(String^, int) Method

Sets a column to a 32-bit signed integer value.

Syntax

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

Parameters

cname The name of the column.
value The 32-bit signed integer value.

1.3.42.2 SetInt32(uint16, int) Method

Sets a column to an integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The signed integer value.

1.3.43 SetInt64 Method

Sets a column to an integer value.

Overload list

Modifier and Type	Overload name	Description
public void	SetInt64(String^, int64) [page 92]	Sets a column to an integer value.
public void	SetInt64(uint16, int64) [page 93]	Sets a column to an integer value.

In this section:

SetInt64(String^, int64) Method [page 92]
Sets a column to an integer value.

SetInt64(uint16, int64) Method [page 93]
Sets a column to an integer value.

1.3.43.1 SetInt64(String^, int64) Method

Sets a column to an integer value.

Syntax

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

Parameters

cname The name of the column.
value The signed integer value.

1.3.43.2 SetInt64(uint16, int64) Method

Sets a column to an integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The signed integer value.

1.3.44 SetNull Method

Sets a column to null.

Overload list

Modifier and Type	Overload name	Description
public void	SetNull(String^) [page 94]	Sets a column to null.
public void	SetNull(uint16) [page 94]	Sets a column to null.

In this section:

SetNull(String^) Method [page 94]
Sets a column to null.

SetNull(uint16) Method [page 94]
Sets a column to null.

1.3.44.1 SetNull(String^) Method

Sets a column to null.

Syntax

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

Parameters

cname The name of the column.

1.3.44.2 SetNull(uint16) Method

Sets a column to null.

Syntax

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

Parameters

cid The zero-based ordinal column number.

1.3.45 SetString Method

Sets a column to a string value.

Overload list

Modifier and Type	Overload name	Description
public void	SetString(String^, String^) [page 95]	Sets a column to a string value.
public void	SetString(uint16, String^) [page 95]	Sets a column to a string value.

In this section:

SetString(String^, String^) Method [page 95]
Sets a column to a string value.

SetString(uint16, String^) Method [page 95]
Sets a column to a string value.

1.3.45.1 SetString(String^, String^) Method

Sets a column to a string value.

Syntax

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

Parameters

cname The name of the column.

value The string value.

1.3.45.2 SetString(uint16, String^) Method

Sets a column to a string value.

Syntax

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

Parameters

cid The zero-based ordinal column number.

value The string value.

1.3.46 SetUInt16 Method

Sets a column to an unsigned 16-bit integer value.

Overload list

Modifier and Type	Overload name	Description
public void	SetUInt16(String^, uint16) [page 96]	Sets a column to an unsigned 16-bit integer value.
public void	SetUInt16(uint16, uint16) [page 97]	Sets a column to an unsigned 16-bit integer value.

In this section:

SetUInt16(String^, uint16) Method [page 96]
Sets a column to an unsigned 16-bit integer value.

SetUInt16(uint16, uint16) Method [page 97]
Sets a column to an unsigned 16-bit integer value.

1.3.46.1 SetUInt16(String^, uint16) Method

Sets a column to an unsigned 16-bit integer value.

Syntax

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

Parameters

cname The name of the column.
value The unsigned integer value.

1.3.46.2 SetUInt16(uint16, uint16) Method

Sets a column to an unsigned 16-bit integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The unsigned integer value.

1.3.47 SetUInt32 Method

Sets a column to an unsigned 32-bit integer value.

Overload list

Modifier and Type	Overload name	Description
public void	SetUInt32(String^, unsigned int) [page 98]	Sets a column to an unsigned 32-bit integer value.
public void	SetUInt32(uint16, unsigned int) [page 98]	Sets a column to an unsigned 32-bit integer value.

In this section:

SetUInt32(String^, unsigned int) Method [page 98]
Sets a column to an unsigned 32-bit integer value.

SetUInt32(uint16, unsigned int) Method [page 98]
Sets a column to an unsigned 32-bit integer value.

1.3.47.1 SetUInt32(String^, unsigned int) Method

Sets a column to an unsigned 32-bit integer value.

Syntax

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

Parameters

cname The name of the column.
value The unsigned integer value.

1.3.47.2 SetUInt32(uint16, unsigned int) Method

Sets a column to an unsigned 32-bit integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The unsigned integer value.

1.3.48 SetUInt64 Method

Sets a column to an unsigned 64-bit integer value.

Overload list

Modifier and Type	Overload name	Description
public void	SetUInt64(String^, uint64) [page 99]	Sets a column to an unsigned 64-bit integer value.
public void	SetUInt64(uint16, uint64) [page 100]	Sets a column to an unsigned 64-bit integer value.

In this section:

SetUInt64(String^, uint64) Method [page 99]
    Sets a column to an unsigned 64-bit integer value.

SetUInt64(uint16, uint64) Method [page 100]
    Sets a column to an unsigned 64-bit integer value.

1.3.48.1 SetUInt64(String^, uint64) Method

Sets a column to an unsigned 64-bit integer value.

Syntax

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

Parameters

cname The name of the column.

value The unsigned integer value.

1.3.48.2 SetUInt64(uint16, uint64) Method

Sets a column to an unsigned 64-bit integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The unsigned integer value.

1.3.49 Update() Method

Updates the current row.

Syntax

```
public void Update ()
```

1.3.50 UpdateBegin() Method

Start update mode for setting columns.

Syntax

```
public void UpdateBegin ()
```

Remarks

Columns in the primary key cannot be modified when UltraLite is in update mode. ULResultSet.Update() updates the row after UpdateBegin() is called. Connection.Commit() or Rollback() commits or rolls back the transaction.

1.4 CursorSchema Interface

Common interface for ResultSetSchema and TableSchema.

Namespace

```
UltraLite
```

Syntax

```
public interface class
```

Members

All members of CursorSchema, including inherited members.

Methods

Modifier and Type	Method	Description
public uint16	GetColumnCount() [page 102]	Gets the number of columns in the result set or table.
public uint16	GetColumnID(String^) [page 102]	Gets the zero-based column ID from its name.
public String	GetColumnName(uint16, uint16type) [page 103]	Gets the name of a column given its zero-based ID.
public uint16	GetColumnPrecision(uint16) [page 104]	Gets the precision of a numeric column.
public uint16	GetColumnScale(uint16) [page 104]	Gets the scale of a numeric column.
public int	GetColumnSize(uint16) [page 105]	Gets the size of the column.
public uint16	GetColumnSQLType(uint16) [page 105]	Gets the SQL type of a column.
public uint16	GetColumnType(uint16) [page 106]	Gets the storage/host variable type of a column.
public bool	IsAliased(uint16) [page 106]	Indicates whether the column in a result set was given an alias.

In this section:

GetColumnCount() Method [page 102]
  Gets the number of columns in the result set or table.

GetColumnID(String^) Method [page 102]

Gets the zero-based column ID from its name.

GetColumnName(uint16, uint16) Method [page 103]
: Gets the name of a column given its zero-based ID.

GetColumnPrecision(uint16) Method [page 104]
: Gets the precision of a numeric column.

GetColumnScale(uint16) Method [page 104]
: Gets the scale of a numeric column.

GetColumnSize(uint16) Method [page 105]
: Gets the size of the column.

GetColumnSQLType(uint16) Method [page 105]
: Gets the SQL type of a column.

GetColumnType(uint16) Method [page 106]
: Gets the storage/host variable type of a column.

IsAliased(uint16) Method [page 106]
: Indicates whether the column in a result set was given an alias.

1.4.1 GetColumnCount() Method

Gets the number of columns in the result set or table.

Syntax

```
public uint16 GetColumnCount ()
```

Returns

The number of columns in the result set or table.

1.4.2 GetColumnID(String^) Method

Gets the zero-based column ID from its name.

Syntax

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

Parameters

columnName The column name.

Returns

The column ID.

1.4.3 GetColumnName(uint16, uint16) Method

Gets the name of a column given its zero-based ID.

Syntax

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

Parameters

cid The zero-based ordinal column number.
type The desired column name type.

Returns

A string containing the column name, if found; otherwise, an error is thrown.

Remarks

Depending on the type selected and how the column was declared in the SELECT statement, the column name may be returned in the form [table-name].[column-name].

The type parameter specifies what type of column name to return.

1.4.4 GetColumnPrecision(uint16) Method

Gets the precision of a numeric column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The precision of the numeric column.

1.4.5 GetColumnScale(uint16) Method

Gets the scale of a numeric column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The scale of the numeric column.

1.4.6 GetColumnSize(uint16) Method

Gets the size of the column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The length in bytes of a fixed CHAR or BINARY column, or a GEOMETRY type column..

1.4.7 GetColumnSQLType(uint16) Method

Gets the SQL type of a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

SQLTYPE_BAD_INDEX if the column does not exist.

1.4.8 GetColumnType(uint16) Method

Gets the storage/host variable type of a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

TYPE_BAD_INDEX if the column does not exist.

1.4.9 IsAliased(uint16) Method

Indicates whether the column in a result set was given an alias.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

True if the column is aliased; otherwise, returns false.

1.5 DatabaseManager Class

Manages connections and databases.

Namespace

```
UltraLite
```

Syntax

```
public ref class sealed
```

Members

All members of DatabaseManager, including inherited members.

Methods

Modifier and Type	Method	Description
public static Connection	CreateDatabase(String^, String^) [page 108]	Creates a new database.
public static FileTransfer	CreateFileTransfer(String^, uint16, String^, String^) [page 109]	Creates a FileTransfer object.
public static void	DropDatabase(String^) [page 110]	Erases an existing database that is not currently running.
public static void	Fini() [page 110]	Finalizes the UltraLite runtime.
public static void	GetLastError(ULSqlCode *, String^*) [page 111]	Returns the error information associated with the last call to this DatabaseManager.
public static bool	Init() [page 111]	Initializes the UltraLite runtime.
public static Connection	OpenConnection(String^) [page 112]	Opens a new connection to an existing database.
public static void	ValidateDatabase(String^, uint16, ValidateCallback^) [page 112]	Performs low level and index validation on a database.
public static IAsyncActionWithProgress< ValidateData^>	ValidateDatabaseAsync(String^, uint16flags) [page 113]	Performs a ValidateDatabase operation asynchronously.

Remarks

The Init method must be called in a thread-safe environment before any other calls can be made. The Fini method must be called in a similarly thread-safe environment when finished.

> **Note**
>
> This class is static. Do not create an instance of it.

In this section:

CreateDatabase(String^, String^) Method [page 108]
: Creates a new database.

CreateFileTransfer(String^, uint16, String^, String^) Method [page 109]
: Creates a FileTransfer object.

DropDatabase(String^) Method [page 110]
: Erases an existing database that is not currently running.

Fini() Method [page 110]
: Finalizes the UltraLite runtime.

GetLastError(ULSqlCode *, String^*) Method [page 111]
: Returns the error information associated with the last call to this DatabaseManager.

Init() Method [page 111]
: Initializes the UltraLite runtime.

OpenConnection(String^) Method [page 112]
: Opens a new connection to an existing database.

ValidateDatabase(String^, uint16, ValidateCallback^) Method [page 112]
: Performs low level and index validation on a database.

ValidateDatabaseAsync(String^, uint16flags) Method [page 113]
: Performs a ValidateDatabase operation asynchronously.

1.5.1 CreateDatabase(String^, String^) Method

Creates a new database.

Syntax

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

Parameters

connParms A semicolon-separated string of connection parameters, which are set as keyword=value pairs. The connection string must include the name of the database. These parameters are the same set of parameters that can be specified when you connect to a database.

createParms A semicolon-separated string of database creation parameters, which are set as keyword=value pairs. For example: page_size=2048;obfuscate=yes.

Remarks

The database is created with information provided in two sets of parameters.

The connParms parameter is a set of standard connection parameters that are applicable whenever the database is accessed, such as the file name or the encryption key.

The createParms parameter is a set of parameters that are only relevant when creating a database, such as checksum-level, page size, collation, and time and date format.

Example

The following code illustrates how to use the CreateDatabase method to create an UltraLite database as the file `mydb.udb`:

```
String mypath = ApplicationData.Current.LocalFolder.Path;
            String connParms = "dbf=" + mypath + "\\mydb.udb";
            Connection conn = null;
            try {
            conn = DatabaseManager.CreateDatabase(connParms,
"kdf_iterations=100");
            } catch( Exception ex ) {
            // report error
            }
```

1.5.2 CreateFileTransfer(String^, uint16, String^, String^) Method

Creates a FileTransfer object.

Syntax

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

Parameters

fileName The name of the server file to transfer. This parameter must not contain any path information.
stream One of the constants defined in the StreamType enumeration. It is used to identify the type of communication stream.
userName The MobiLink user name.
version The MobiLink script version.

1.5.3 DropDatabase(String^) Method

Erases an existing database that is not currently running.

Syntax

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

Parameters

parms The database identification parameters (a connection string).

1.5.4 Fini() Method

Finalizes the UltraLite runtime.

Syntax

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

Remarks

This method must be called only once by a single thread when the application has finished. This method is not thread safe.

1.5.5 GetLastError(ULSqlCode *, String^*) Method

Returns the error information associated with the last call to this DatabaseManager.

Syntax

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

Parameters

errorCode Out parameter holding the returned UltraLite SQL error code.

errorParms Out parameter holding the returned String of error parameters, or an empty string if there are no error parameters. The error parameters are a comma-separated list of values for %n parameters in an error message.

Remarks

See Connection::GetLastError.

1.5.6 Init() Method

Initializes the UltraLite runtime.

Syntax

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

Returns

True on success; otherwise, returns false. This method returns False if it is called more than once.

Remarks

This method must be called only once by a single thread before any other calls can be made. This method is not thread safe.

This method does not usually fail unless memory is unavailable.

1.5.7 OpenConnection(String^) Method

Opens a new connection to an existing database.

Syntax

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

Parameters

connParms The connection string.

Remarks

The connection string is a set of semicolon delimited option=value connection parameters that indicates which database to connect to and what options to use for the connection. For example, after securely obtaining your encryption key, the resulting connection string might be: "DBF=mydb.udb;DBKEY=iyntTZld9OEa#&G".

The following is a list of possible errors (you can obtain them from GetLastError):

- SQLE_INVALID_PARSE_PARAMETER - connParms was not formatted properly.
- SQLE_UNRECOGNIZED_OPTION - A connection option name was likely misspelled.
- SQLE_INVALID_OPTION_VALUE - A connection option value was not specified properly.
- SQLE_ULTRALITE_DATABASE_NOT_FOUND - The specified database could not be found.
- SQLE_INVALID_LOGON - You supplied an invalid user ID or an incorrect password.
- SQLE_TOO_MANY_CONNECTIONS - You exceeded the maximum number of concurrent database connections.

1.5.8 ValidateDatabase(String^, uint16, ValidateCallback^) Method

Performs low level and index validation on a database.

Syntax

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

Parameters

connParms The parameters used to connect to the database.
flags A flag that controls the type of validation; see the example below.
cb A function to receive validation progress information.

Remarks

The flags parameter is combination of the following values:

- ULVF_TABLE
- ULVF_INDEX
- ULVF_DATABASE
- ULVF_EXPRESS or ULVF_FULL_VALIDATE

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

1.5.9 ValidateDatabaseAsync(String^, uint16flags) Method

Performs a ValidateDatabase operation asynchronously.

Syntax

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

Parameters

connParms The parameters used to connect to the database.
flags Controls the type of validation.

Returns

An awaitable asynchronous action with progress reporting.

1.6 DatabaseSchema Class

Represents the schema of an UltraLite database. Use Connection.GetDatabaseSchema to get a DatabaseSchema object.

Namespace

```
UltraLite
```

Syntax

```
public ref class  sealed
```

Members

All members of DatabaseSchema, including inherited members.

Methods

Modifier and Type	Method	Description
public void	Close() [page 115]	Destroys this object.
public uint8	GetPublicationCount() [page 115]	Gets the number of publications in the database.
public IIterator< String^>	GetPublications() [page 116]	Gets an iterator over a collection of all publications (names) in the database.
public uint16	GetTableCount() [page 116]	Returns the number of tables in the da-tabase.
public IIterator< TableSchema^>	GetTables() [page 116]	Gets an iterator over a collection of all tables (schema) in the database.
public TableSchema	GetTableSchema(String^) [page 117]	Returns the schema of the named ta-ble.

In this section:

CloseObject() Method [page 115]
    Destroys this object.

GetPublicationCount() Method [page 115]
    Gets the number of publications in the database.

GetPublications() Method [page 116]
    Gets an iterator over a collection of all publications (names) in the database.

GetTableCount() Method [page 116]
Returns the number of tables in the database.

GetTables() Method [page 116]
Gets an iterator over a collection of all tables (schema) in the database.

GetTableSchema(String^) Method [page 117]
Returns the schema of the named table.

1.6.1 CloseObject() Method

Destroys this object.

Syntax

```
public void CloseObject ()
```

1.6.2 GetPublicationCount() Method

Gets the number of publications in the database.

Syntax

```
public uint8 GetPublicationCount ()
```

Returns

The number of publications in the database.

Remarks

Publication IDs range from 1 to the number returned by this method.

1.6.3 GetPublications() Method

Gets an iterator over a collection of all publications (names) in the database.

Syntax

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

Returns

An iterator over a collection of String objects.

1.6.4 GetTableCount() Method

Returns the number of tables in the database.

Syntax

```
public uint16 GetTableCount ()
```

Returns

An integer that represents the number of tables.

1.6.5 GetTables() Method

Gets an iterator over a collection of all tables (schema) in the database.

Syntax

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

Returns

An iterator over a collection of TableSchema objects.

1.6.6 GetTableSchema(String^) Method

Returns the schema of the named table.

Syntax

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

Parameters

tableName The name of the table.

Returns

A TableSchema object for the given table; otherwise, returns UL_NULL if the table does not exist.

1.7 FileTransfer Class

Transfers a file from a remote database via the MobiLink server.

Namespace

```
UltraLite
```

Syntax

```
public ref class sealed
```

Members

All members of FileTransfer, including inherited members.

Variables

Modifier and Type	Variable	Description
public property String	FileName	Specifies the name of the file to download.
public property String	LocalPath	Specifies where to download the file.
public property String	LocalFileName	Specifies the local file name for the downloaded file.
public property String	EncryptionKey	The encryption key used to encrypt a compatible file. This property is reserved for future use. If this parameter is non-NULL, a download file is expected to be a file that is compatible with encryption, and is encrypted when stored in the file system on the device. If this parameter is NULL (the default), a download file is treated normally.
public property uint16	Stream	Specifies the MobiLink synchronization stream to use for the file transfer.
public property String	StreamParms	Specifies the parameters to configure the synchronization stream.
public property String	UserName	Specifies the user name that identifies the MobiLink client to the MobiLink server.
public property String	Password	Specifies the MobiLink password for the user specified by the UserName value. If the password is left unset, it defautls to "".
public property String	RemoteKey	Specifies the key that uniquely identifies the MobiLink client to the MobiLink server.
public property String	Version	Specifies which synchronization script to use.
public property bool	ResumePartialDownload	Specifies whether to resume or discard a previous partial download.
public property Array< String^>	AuthenticationParms	Specifies parameters for a custom user authentication script (MobiLink authenticate_parameters connection event).

Methods

Modifier and Type	Method	Description
public bool	DownloadFile(FileTransferObserver^) [page 119]	Downloads the file specified by the properties of this object.
public IAsyncOperationWithProgress< bool, FileTransferStatus^>	DownloadFileAsync() [page 120]	Performs a DownloadFile operation asynchronously.

Modifier and Type	Method	Description
public FileTransferResult	GetResult() [page 120]	Returns the result of the file transfer.
public bool	UploadFile(FileTransferObserver^) [page 120]	Uploads the file specified by the properties of this object.
public IAsyncOperationWithProgress< bool, FileTransferStatus^>	UploadFileAsync() [page 121]	Performs an UploadFile operation asynchronously.

Remarks

You do not need a database connection to perform a file transfer. To transfer a file, set the FileTransfer.FileName, FileTransfer.Stream, FileTransfer.UserName and FileTransfer.Version values. Use DatabaseManager.CreateFileTransfer() to create a FileTransfer object.

In this section:

DownloadFile(FileTransferObserver^) Method [page 119]
Downloads the file specified by the properties of FileTransfer.GetResult.

DownloadFileAsync() Method [page 120]
Performs a DownloadFile operation asynchronously.

GetResult() Method [page 120]
Returns the result of the file transfer.

UploadFile(FileTransferObserver^) Method [page 120]
Uploads the file specified by the properties of this object.

UploadFileAsync() Method [page 121]
Performs an UploadFile operation asynchronously.

1.7.1 DownloadFile(FileTransferObserver^) Method

Downloads the file specified by the properties of FileTransfer.GetResult.

Syntax

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

Parameters

observer The observer callback to send status updates to.

Returns

True if successful, and false otherwise (check the FileTransferResult.streamErrorCode value and other status properties for the reason).

Remarks

The file specified by the FileTransfer.FileName value is downloaded by the MobiLink server to the FileTransfer.LocalPath value by using the FileTransfer.Stream, FileTransfer.UserName, FileTransfer.Password, and FileTransfer.Version values. To avoid file corruption, UltraLite downloads to a temporary file and only replaces the local file once the download completes. Other properties that affect the download are: FileTransfer.LocalFileName, FileTransfer.AuthenticationParms, and FileTransfer.ResumePartialDownload. A detailed result status is reported by the GetResult method of this object.

1.7.2 DownloadFileAsync() Method

Performs a DownloadFile operation asynchronously.

Syntax

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

1.7.3 GetResult() Method

Returns the result of the file transfer.

Syntax

```
public FileTransferResult GetResult ()
```

1.7.4 UploadFile(FileTransferObserver^) Method

Uploads the file specified by the properties of this object.

Syntax

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

Parameters

observer The observer callback to send status updates to.

Returns

True if successful, and false otherwise (check the FileTransfer.streamErrorCode value and other status properties for the reason).

Remarks

The file specified by the FileTransfer.FileName value is uploaded to the MobiLink server from the FileTransfer.LocalPath value using the FileTransfer.Stream, FileTransfer.UserName, FileTransfer.Password, and FileTransfer.Version values. A detailed result status is reported by the GetResult method of this object.

1.7.5 UploadFileAsync() Method

Performs an UploadFile operation asynchronously.

Syntax

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

1.8 IndexSchema Class

Represents the schema of an UltraLite table index. Use TableSchem.GetIndexSchema(indexName) to get an IndexSchema object.

Namespace

```
UltraLite
```

Syntax

```
public ref class  sealed
```

Members

All members of IndexSchema, including inherited members.

Methods

Modifier and Type	Method	Description
public void	Close() [page 123]	Destroys this object.
public uint16	GetColumnCount() [page 123]	Gets the number of columns in the index.
public String	GetColumnName(uint16) [page 123]	Gets the name of the column given the position of the column in the index.
public uint16	GetIndexColumnID(String^) [page 124]	Gets the zero-based index column ID from its name.
public uint16	GetIndexFlags() [page 124]	Gets the index property flags bit field.
public String	GetName() [page 125]	Gets the name of the index.
public String	GetReferencedIndexName() [page 125]	Gets the associated primary index name.
public String	GetReferencedTableName() [page 125]	Gets the associated primary table name.
public String	GetTableName() [page 126]	Gets the name of the table containing this index.
public bool	IsColumnDescending(uint16) [page 126]	Determines whether the column is in descending order.

In this section:

CloseObject() Method [page 123]
    Destroys this object.

GetColumnCount() Method [page 123]
    Gets the number of columns in the index.

GetColumnName(uint16) Method [page 123]
    Gets the name of the column given the position of the column in the index.

GetIndexColumnID(String^) Method [page 124]
    Gets the zero-based index column ID from its name.

GetIndexFlags() Method [page 124]
    Gets the index property flags bit field.

GetName() Method [page 125]
    Gets the name of the index.

GetReferencedIndexName() Method [page 125]
    Gets the associated primary index name.

GetReferencedTableName() Method [page 125]
    Gets the associated primary table name.

GetTableName() Method [page 126]

Gets the name of the table containing this index.

IsColumnDescending(uint16) Method [page 126]
Determines whether the column is in descending order.

1.8.1 CloseObject() Method

Destroys this object.

Syntax

```
public void CloseObject ()
```

1.8.2 GetColumnCount() Method

Gets the number of columns in the index.

Syntax

```
public uint16 GetColumnCount ()
```

Returns

The number of columns in the index.

1.8.3 GetColumnName(uint16) Method

Gets the name of the column given the position of the column in the index.

Syntax

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

Parameters

col_id_in_index The zero-based ordinal number indicating the position of the column in the index.

Returns

The name of the column.

1.8.4 GetIndexColumnID(String^) Method

Gets the zero-based index column ID from its name.

Syntax

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

Parameters

columnName The column name.

Returns

The column ID.

1.8.5 GetIndexFlags() Method

Gets the index property flags bit field.

Syntax

```
public uint16 GetIndexFlags ()
```

Returns

IndexFlag

1.8.6 GetName() Method

Gets the name of the index.

Syntax

```
public String GetName ()
```

Returns

The name of the index.

1.8.7 GetReferencedIndexName() Method

Gets the associated primary index name.

Syntax

```
public String GetReferencedIndexName ()
```

Returns

The name of the referenced index.

Remarks

This method applies to foreign keys only.

1.8.8 GetReferencedTableName() Method

Gets the associated primary table name.

Syntax

```
public String GetReferencedTableName ()
```

Returns

The name of the referenced table.

Remarks

This method applies to foreign keys only.

1.8.9 GetTableName() Method

Gets the name of the table containing this index.

Syntax

```
public String GetTableName ()
```

Returns

The name of the table containing this index.

1.8.10 IsColumnDescending(uint16) Method

Determines whether the column is in descending order.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

True if the column is in descending order; otherwise, returns false.

1.9 PreparedStatement Class

Represents a prepared SQL statement. Connection.PrepareStatement returns a PreparedStatement.

Namespace

```
UltraLite
```

Syntax

```
public ref class  sealed
```

Members

All members of PreparedStatement, including inherited members.

Methods

Modifier and Type	Method	Description
public void	AppendParameterByteChunk(uint16, const Array< uint8 >^, int) [page 130]	Sets a large binary parameter that is broken down into several chunks.
public void	AppendParameterStringChunk(uint16, String^) [page 131]	Sets a large string parameter that is broken down into several chunks.
public void	Close() [page 131]	Destroys this object.
public ResultSet	ExecuteQuery() [page 131]	Executes a SQL SELECT statement as a query.
public IAsyncOperation< ResultSet^>	ExecuteQueryAsync() [page 132]	Performs an ExecuteQuery operation asynchronously.
public void	ExecuteStatement() [page 132]	Executes a statement that does not return a result set, such as a SQL INSERT, DELETE, or UPDATE statement.
public IAsyncAction	ExecuteStatementAsync() [page 132]	Performs an ExecuteStatement operation asynchronously.

Modifier and Type	Method	Description
public uint16	GetParameterCount() [page 133]	Gets the number of input parameters for this statement.
public uint16	GetParameterID(String^) [page 133]	Gets the zero-based ordinal for a parameter name.
public uint16	GetParameterType(uint16) [page 134]	Gets the storage/host variable type of a parameter.
public String	GetPlan() [page 134]	Gets a text-based description of the query execution plan.
public ResultSetSchema	GetResultSetSchema() [page 135]	Gets the schema for the result set.
public int	GetRowsAffectedCount() [page 135]	Gets the number of rows affected by the last statement.
public bool	HasResultSet() [page 135]	Determines whether the SQL statement has a result set.
public void	SetParameterBool(uint16, bool) [page 136]	Sets a parameter to a boolean value.
public void	SetParameterByte(uint16, uint8) [page 136]	Sets a parameter to an 8-bit integer value.
public void	SetParameterBytes(uint16, const Array< uint8 >^, int) [page 137]	Sets a parameter to a byte array value.
public void	SetParameterDateTime(uint16, DateTime) [page 137]	Sets a parameter to a DateTime value.
public void	SetParameterDouble(uint16, double) [page 137]	Sets a parameter to a double value.
public void	SetParameterFloat(uint16, float) [page 138]	Sets a parameter to a float value.
public void	SetParameterGuid(uint16, Guid) [page 138]	Sets a parameter to a GUID value.
public void	SetParameterInt16(uint16, int16) [page 139]	Sets a parameter to a 16-bit integer value.
public void	SetParameterInt32(uint16, int) [page 139]	Sets a parameter to a 32-bit integer value.
public void	SetParameterInt64(uint16, int64) [page 139]	Sets a parameter to a 64-bit integer value.
public void	SetParameterNull(uint16) [page 140]	Sets a parameter to null.
public void	SetParameterString(uint16, String^) [page 140]	Sets a parameter to a string value.
public void	SetParameterUInt16(uint16, uint16) [page 141]	Sets a parameter to an unsigned 16-bit integer value.
public void	SetParameterUInt32(uint16, unsigned int) [page 141]	Sets a parameter to an unsigned 32-bit integer value.
public void	SetParameterUInt64(uint16, uint64) [page 141]	Sets a parameter to an unsigned 64-bit integer value.

In this section:

AppendParameterByteChunk(uint16, const Array< uint8 >^, int) Method [page 130]
    Sets a large binary parameter that is broken down into several chunks.

AppendParameterStringChunk(uint16, String^) Method [page 131]
    Sets a large string parameter that is broken down into several chunks.

CloseObject() Method [page 131]
    Destroys this object.

ExecuteQuery() Method [page 131]
    Executes a prepared statement that returns a result set.

ExecuteQueryAsync() Method [page 132]
    Performs an ExecuteQuery operation asynchronously.

ExecuteStatement() Method [page 132]
    Executes a prepared statement that does not return a result set, such as a SQL INSERT, DELETE, or UPDATE statement.

ExecuteStatementAsync() Method [page 132]
    Performs an ExecuteStatement operation asynchronously.

GetParameterCount() Method [page 133]
    Gets the number of input parameters for the prepared statement.

GetParameterID(String^) Method [page 133]
    Gets the zero-based ordinal for a parameter name.

GetParameterType(uint16) Method [page 134]
    Gets the storage/host variable type of a parameter.

GetPlan() Method [page 134]
    Gets a text-based description of the query execution plan.

GetResultSetSchema() Method [page 135]
    Gets the schema for the result set.

GetRowsAffectedCount() Method [page 135]
    Gets the number of rows affected by the last statement.

HasResultSet() Method [page 135]
    Determines whether the prepared SQL statement has a result set.

SetParameterBool(uint16, bool) Method [page 136]
    Sets a parameter to a boolean value.

SetParameterByte(uint16, uint8) Method [page 136]
    Sets a parameter to an 8-bit integer value.

SetParameterBytes(uint16, const Array< uint8 >^, int) Method [page 137]
    Sets a parameter to a byte array value.

SetParameterDateTime(uint16, DateTime) Method [page 137]
    Sets a parameter to a DateTime value.

SetParameterDouble(uint16, double) Method [page 137]
    Sets a parameter to a double value.

SetParameterFloat(uint16, float) Method [page 138]

Sets a parameter to a float value.

SetParameterGuid(uint16, Guid) Method [page 138]
Sets a parameter to a GUID value.

SetParameterInt16(uint16, int16) Method [page 139]
Sets a parameter to a 16-bit integer value.

SetParameterInt32(uint16, int) Method [page 139]
Sets a parameter to a 32-bit integer value.

SetParameterInt64(uint16, int64) Method [page 139]
Sets a parameter to a 64-bit integer value.

SetParameterNull(uint16) Method [page 140]
Sets a parameter to null.

SetParameterString(uint16, String^) Method [page 140]
Sets a parameter to a string value.

SetParameterUInt16(uint16, uint16) Method [page 141]
Sets a parameter to an unsigned 16-bit integer value.

SetParameterUInt32(uint16, unsigned int) Method [page 141]
Sets a parameter to an unsigned 32-bit integer value.

SetParameterUInt64(uint16, uint64) Method [page 141]
Sets a parameter to an unsigned 64-bit integer value.

1.9.1 AppendParameterByteChunk(uint16, const Array< uint8 >^, int) Method

Sets a large binary parameter that is broken down into several chunks.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.
value The byte chunk to append.
valueSize The size of the buffer.

1.9.2 AppendParameterStringChunk(uint16, String^) Method

Sets a large string parameter that is broken down into several chunks.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.
value The string chunk to append.

1.9.3 CloseObject() Method

Destroys this object.

Syntax

```
public void CloseObject ()
```

1.9.4 ExecuteQuery() Method

Executes a prepared statement that returns a result set.

Syntax

```
public ResultSet ExecuteQuery ()
```

Returns

The ResultSet object that contains the results of the query as a set of rows.

1.9.5 ExecuteQueryAsync() Method

Performs an ExecuteQuery operation asynchronously.

Syntax

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

Returns

An awaitable asynchronous operation for the ResultSet.

Remarks

Executes a prepared statement that returns a result set.

1.9.6 ExecuteStatement() Method

Executes a prepared statement that does not return a result set, such as a SQL INSERT, DELETE, or UPDATE statement.

Syntax

```
public void ExecuteStatement ()
```

1.9.7 ExecuteStatementAsync() Method

Performs an ExecuteStatement operation asynchronously.

Syntax

```
public IAsyncAction ExecuteStatementAsync ()
```

Returns

An awaitable asynchronous action.

Remarks

Executes a prepared statement that does not return a result set, such as a SQL INSERT, DELETE, or UPDATE statement.

1.9.8 GetParameterCount() Method

Gets the number of input parameters for the prepared statement.

Syntax

```
public uint16 GetParameterCount ()
```

Returns

The number of input parameters for the prepared statement.

1.9.9 GetParameterID(String^) Method

Gets the zero-based ordinal for a parameter name.

Syntax

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

Parameters

name The name of the host variable.

Returns

The zero-based ordinal for a parameter name.

1.9.10 GetParameterType(uint16) Method

Gets the storage/host variable type of a parameter.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.

Returns

The type of the specified parameter.

1.9.11 GetPlan() Method

Gets a text-based description of the query execution plan.

Syntax

```
public String GetPlan ()
```

Returns

The plan text.

Remarks

This method is intended primarily for use during development.

An empty string is returned if there is no plan. Plans exist when the prepared statement is a SQL query.

When the plan is obtained before the associated query has been executed, the plan shows the operations used to execute the query. The plan additionally shows the number of rows each operation produced when the plan is obtained after the query has been executed. This plan can be used to gain insight about the execution of the query.

1.9.12 GetResultSetSchema() Method

Gets the schema for the result set.

Syntax

```
public ResultSetSchema GetResultSetSchema ()
```

Returns

A ResultSetSchema object that can be used to get information about the schema of the result set.

1.9.13 GetRowsAffectedCount() Method

Gets the number of rows affected by the last statement.

Syntax

```
public int GetRowsAffectedCount ()
```

Returns

The number of rows affected by the last statement. If this information is not available (for instance, the statement alters the schema rather than data), then the return value is -1.

1.9.14 HasResultSet() Method

Determines whether the prepared SQL statement has a result set.

Syntax

```
public bool HasResultSet ()
```

Returns

True if a result set is generated when this statement is executed; otherwise, returns false if no result set is generated.

1.9.15 SetParameterBool(uint16, bool) Method

Sets a parameter to a boolean value.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.
value The boolean value.

1.9.16 SetParameterByte(uint16, uint8) Method

Sets a parameter to an 8-bit integer value.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.
value The integer value.

1.9.17 SetParameterBytes(uint16, const Array< uint8 >^, int) Method

Sets a parameter to a byte array value.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.
value The byte array value.
length The length in bytes to get from the byte array value.

1.9.18 SetParameterDateTime(uint16, DateTime) Method

Sets a parameter to a DateTime value.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.
value The DateTime value.

1.9.19 SetParameterDouble(uint16, double) Method

Sets a parameter to a double value.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.
value The double value.

1.9.20 SetParameterFloat(uint16, float) Method

Sets a parameter to a float value.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.
value The float value.

1.9.21 SetParameterGuid(uint16, Guid) Method

Sets a parameter to a GUID value.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.
value The GUID value.

1.9.22 SetParameterInt16(uint16, int16) Method

Sets a parameter to a 16-bit integer value.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.
value The integer value.

1.9.23 SetParameterInt32(uint16, int) Method

Sets a parameter to a 32-bit integer value.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.
value The integer value.

1.9.24 SetParameterInt64(uint16, int64) Method

Sets a parameter to a 64-bit integer value.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.
value The integer value.

1.9.25 SetParameterNull(uint16) Method

Sets a parameter to null.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.

1.9.26 SetParameterString(uint16, String^) Method

Sets a parameter to a string value.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.
value The string value. SQLE_INVALID_PARAMETER is set if this parameter is greater than 32 KB in length. For large strings, call the AppendParameterStringChunk method instead.

1.9.27 SetParameterUInt16(uint16, uint16) Method

Sets a parameter to an unsigned 16-bit integer value.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.
value The integer value.

1.9.28 SetParameterUInt32(uint16, unsigned int) Method

Sets a parameter to an unsigned 32-bit integer value.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.
value The integer value.

1.9.29 SetParameterUInt64(uint16, uint64) Method

Sets a parameter to an unsigned 64-bit integer value.

Syntax

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

Parameters

pid The zero-based ordinal of the parameter.

value The integer value.

1.10 ResultSet Class

Represents a result set in an UltraLite database. ResultSet objects are used for updates and deletes and are returned by PreparedStatement.ExecuteQuery().

Namespace

```
UltraLite
```

Syntax

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

Members

All members of ResultSet, including inherited members.

Methods

Modifier and Type	Method	Description
public virtual void	AfterLast() [page 147]	Moves the cursor after the last row.
public virtual void	AppendBytes [page 147]	Appends bytes to a column.
public virtual void	AppendChars [page 149]	Appends a string chunk to a column.
public virtual void	BeforeFirst() [page 150]	Moves the cursor before the first row.
public virtual void	Close() [page 151]	Destroys this object.
public virtual void	Delete() [page 151]	Deletes the current row and moves the cursor to the next valid row.
public virtual void	DeleteNamed(String^) [page 151]	Deletes the current row and moves the cursor to the next valid row.
public virtual void	First() [page 151]	Moves the cursor to the first row.
public virtual int64	GetBinaryLength [page 152]	Gets the binary length of the value of a column.

Modifier and Type	Method	Description
public virtual bool	GetBool [page 153]	Fetches a value from a column as a boolean.
public virtual uint8	GetByte [page 155]	Fetches a value from a column as a byte.
public virtual int64	GetBytes [page 156]	Gets a binary chunk from the column.
public virtual int64	GetChars [page 158]	Gets a wide string chunk from the col-umn.
public virtual DateTime	GetDateTime [page 160]	Fetches a value from a column as a Da-teTime.
public virtual double	GetDouble [page 162]	Fetches a value from a column as a double.
public virtual float	GetFloat [page 163]	Fetches a value from a column as a float.
public virtual Guid	GetGuid [page 165]	Fetches a value from a column as a GUID.
public virtual int16	GetInt16 [page 166]	Fetches a value from a column as a 16-bit integer.
public virtual int	GetInt32 [page 168]	Fetches a value from a column as an in-teger.
public virtual int64	GetInt64 [page 169]	Fetches a value from a column as a 64-bit integer.
public ResultSetSchema	GetResultSetSchema() [page 171]	Returns an object that can be used to get information about the result set.
public virtual unsigned int	GetRowCount(unsigned int) [page 171]	Gets the number of rows in the table.
public virtual uint8	GetState() [page 172]	Gets the internal state of the cursor.
public virtual String	GetString [page 172]	Fetches a value from a column as a string.
public virtual int64	GetStringLength [page 174]	Gets the string length of the value of a column.
public virtual uint16	GetUInt16 [page 175]	Fetches a value from a column as a 16-bit unsigned integer.
public virtual unsigned int	GetUInt32 [page 177]	Fetches a value from a column as a 32-bit unsigned integer.
public virtual uint64	GetUInt64 [page 178]	Fetches a value from a column as a 64-bit unsigned integer.
public virtual bool	IsNull [page 180]	Checks whether a column value is NULL.
public virtual void	Last() [page 181]	Moves the cursor to the last row.
public virtual bool	Next() [page 181]	Moves the cursor forward one row.
public virtual bool	Previous() [page 182]	Moves the cursor back one row.

Modifier and Type	Method	Description
public virtual void	Relative(int) [page 182]	Moves the cursor by offset rows from the current cursor position.
public virtual void	SetBool [page 183]	Sets a column to a boolean value.
public virtual void	SetByte [page 184]	Sets a column to a byte value.
public virtual void	SetBytes [page 185]	Sets a column to a binary value.
public virtual void	SetDateTime [page 187]	Sets a column to a DateTime value.
public virtual void	SetDefault [page 188]	Sets a column to its default value.
public virtual void	SetDouble [page 189]	Sets a column to a double value.
public virtual void	SetFloat [page 190]	Sets a column to a float value.
public virtual void	SetGuid [page 192]	Sets a column to a GUID value.
public virtual void	SetInt16 [page 193]	Sets a column to an integer value.
public virtual void	SetInt32 [page 194]	Sets a column to an integer value.
public virtual void	SetInt64 [page 195]	Sets a column to an integer value.
public virtual void	SetNull [page 196]	Sets a column to null.
public virtual void	SetString [page 198]	Sets a column to a string value.
public virtual void	SetUInt16 [page 199]	Sets a column to an unsigned 16-bit integer value.
public virtual void	SetUInt32 [page 200]	Sets a column to an unsigned 32-bit integer value.
public virtual void	SetUInt64 [page 202]	Sets a column to an unsigned 64-bit integer value.
public virtual void	Update() [page 203]	Updates the current row.
public virtual void	UpdateBegin() [page 203]	Selects the update mode for setting columns.

In this section:

AfterLast() Method [page 147]
Moves the cursor after the last row.

AppendBytes Method [page 147]
Appends bytes to a column.

AppendChars Method [page 149]
Appends a string chunk to a column.

BeforeFirst() Method [page 150]
Moves the cursor before the first row.

CloseObject() Method [page 151]
Destroys this object.

Delete() Method [page 151]
Deletes the current row and moves the cursor to the next valid row.

DeleteNamed(String^) Method [page 151]
    Deletes the current row and moves the cursor to the next valid row.

First() Method [page 151]
    Moves the cursor to the first row.

GetBinaryLength Method [page 152]
    Gets the binary length of the value of a column.

GetBool Method [page 153]
    Fetches a value from a column as a boolean.

GetByte Method [page 155]
    Fetches a value from a column as a byte.

GetBytes Method [page 156]
    Gets a binary chunk from the column.

GetChars Method [page 158]
    Gets a wide string chunk from the column.

GetDateTime Method [page 160]
    Fetches a value from a column as a DateTime.

GetDouble Method [page 162]
    Fetches a value from a column as a double.

GetFloat Method [page 163]
    Fetches a value from a column as a float.

GetGuid Method [page 165]
    Fetches a value from a column as a GUID.

GetInt16 Method [page 166]
    Fetches a value from a column as a 16-bit integer.

GetInt32 Method [page 168]
    Fetches a value from a column as an integer.

GetInt64 Method [page 169]
    Fetches a value from a column as a 64-bit integer.

GetResultSetSchema() Method [page 171]
    Returns an object that can be used to get information about the result set.

GetRowCount(unsigned int) Method [page 171]
    Gets the number of rows in the table.

GetState() Method [page 172]
    Gets the internal state of the cursor.

GetString Method [page 172]
    Fetches a value from a column as a string.

GetStringLength Method [page 174]
    Gets the string length of the value of a column.

GetUInt16 Method [page 175]
    Fetches a value from a column as a 16-bit unsigned integer.

GetUInt32 Method [page 177]
    Fetches a value from a column as a 32-bit unsigned integer.

GetUInt64 Method [page 178]
: Fetches a value from a column as a 64-bit unsigned integer.

IsNull Method [page 180]
: Checks whether a column value is NULL.

Last() Method [page 181]
: Moves the cursor to the last row.

Next() Method [page 181]
: Moves the cursor forward one row.

Previous() Method [page 182]
: Moves the cursor back one row.

Relative(int) Method [page 182]
: Moves the cursor by offset rows from the current cursor position.

SetBool Method [page 183]
: Sets a column to a boolean value.

SetByte Method [page 184]
: Sets a column to a byte value.

SetBytes Method [page 185]
: Sets a column to a binary value.

SetDateTime Method [page 187]
: Sets a column to a DateTime value.

SetDefault Method [page 188]
: Sets a column to its default value.

SetDouble Method [page 189]
: Sets a column to a double value.

SetFloat Method [page 190]
: Sets a column to a float value.

SetGuid Method [page 192]
: Sets a column to a GUID value.

SetInt16 Method [page 193]
: Sets a column to an integer value.

SetInt32 Method [page 194]
: Sets a column to an integer value.

SetInt64 Method [page 195]
: Sets a column to an integer value.

SetNull Method [page 196]
: Sets a column to null.

SetString Method [page 198]
: Sets a column to a string value.

SetUInt16 Method [page 199]
: Sets a column to an unsigned 16-bit integer value.

SetUInt32 Method [page 200]
: Sets a column to an unsigned 32-bit integer value.

SetUInt64 Method [page 202]
    Sets a column to an unsigned 64-bit integer value.

Update() Method [page 203]
    Updates the current row.

UpdateBegin() Method [page 203]
    Selects the update mode for setting columns.

1.10.1 AfterLast() Method

Moves the cursor after the last row.

Syntax

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

1.10.2 AppendBytes Method

Appends bytes to a column.

Overload list

Modifier and Type	Overload name	Description
public virtual void	AppendBytes(String^, const Array< uint8 >^, int, int) [page 148]	Appends bytes to a column.
public virtual void	AppendBytes(uint16, const Array< uint8 >^, int, int) [page 148]	Appends bytes to a column.

In this section:

AppendBytes(String^, const Array< uint8 >^, int, int) Method [page 148]
    Appends bytes to a column.

AppendBytes(uint16, const Array< uint8 >^, int, int) Method [page 148]
    Appends bytes to a column.

1.10.2.1 AppendBytes(String^, const Array< uint8 >^, int, int) Method

Appends bytes to a column.

Syntax

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

Parameters

cname The name of the column.
value The byte chunk to append.
offset The offset into the byte chunk at which to start.
size The size of the byte chunk in bytes.

Remarks

The given bytes are appended to the end of the column written so far by AppendBytes method calls.

1.10.2.2 AppendBytes(uint16, const Array< uint8 >^, int, int) Method

Appends bytes to a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The byte chunk to append.
offset The offset into the byte chunk at which to start.
size The size of the byte chunk in bytes.

Remarks

The given bytes are appended to the end of the column written so far by AppendBytes method calls.

1.10.3 AppendChars Method

Appends a string chunk to a column.

Overload list

Modifier and Type	Overload name	Description
public virtual void	AppendChars(String^, const Array< wchar_t >^, int, int) [page 149]	Appends a string chunk to a column.
public virtual void	AppendChars(uint16, const Array< wchar_t >^, int, int) [page 150]	Appends a string chunk to a column.

In this section:

AppendChars(String^, const Array< wchar_t >^, int, int) Method [page 149]
    Appends a string chunk to a column.

AppendChars(uint16, const Array< wchar_t >^, int, int) Method [page 150]
    Appends a string chunk to a column.

1.10.3.1 AppendChars(String^, const Array< wchar_t >^, int, int) Method

Appends a string chunk to a column.

Syntax

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

Parameters

cname The name of the column.
value The string chunk to append.
offset The offset into the string chunk at which to start.

size The length of the string chunk in characters.

Remarks

This method appends the given string to the end of the string written so far by AppendChars method calls.

1.10.3.2 AppendChars(uint16, const Array< wchar_t >^, int, int) Method

Appends a string chunk to a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The string chunk to append.
offset The offset into the string chunk at which to start.
size The length of the string chunk in characters.

Remarks

This method appends the given string to the end of the string written so far by AppendChars method calls.

1.10.4 BeforeFirst() Method

Moves the cursor before the first row.

Syntax

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

1.10.5 CloseObject() Method

Destroys this object.

Syntax

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

1.10.6 Delete() Method

Deletes the current row and moves the cursor to the next valid row.

Syntax

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

1.10.7 DeleteNamed(String^) Method

Deletes the current row and moves the cursor to the next valid row.

Syntax

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

Parameters

tableName A table name or its correlation (required when the database has multiple columns that share the same table name).

1.10.8 First() Method

Moves the cursor to the first row.

Syntax

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

1.10.9 GetBinaryLength Method

Gets the binary length of the value of a column.

Overload list

Modifier and Type	Overload name	Description
public virtual int64	GetBinaryLength(String^) [page 152]	Gets the binary length of the value of a column.
public virtual int64	GetBinaryLength(uint16) [page 153]	Gets the binary length of the value of a column.

In this section:

GetBinaryLength(String^) Method [page 152]
Gets the binary length of the value of a column.

GetBinaryLength(uint16) Method [page 153]
Gets the binary length of the value of a column.

1.10.9.1 GetBinaryLength(String^) Method

Gets the binary length of the value of a column.

Syntax

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

Parameters

cname The name of the column.

Returns

The length of the column value as a binary.

1.10.9.2 GetBinaryLength(uint16) Method

Gets the binary length of the value of a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The length of the column value as a binary.

1.10.10 GetBool Method

Fetches a value from a column as a boolean.

Overload list

Modifier and Type	Overload name	Description
public virtual bool	GetBool(String^) [page 154]	Fetches a value from a column as a boolean.
public virtual bool	GetBool(uint16) [page 154]	Fetches a value from a column as a boolean.

In this section:

GetBool(String^) Method [page 154]
Fetches a value from a column as a boolean.

GetBool(uint16) Method [page 154]
Fetches a value from a column as a boolean.

1.10.10.1 GetBool(String^) Method

Fetches a value from a column as a boolean.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a boolean.

1.10.10.2 GetBool(uint16) Method

Fetches a value from a column as a boolean.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a boolean.

1.10.11 GetByte Method

Fetches a value from a column as a byte.

Overload list

Modifier and Type	Overload name	Description
public virtual uint8	GetByte(String^) [page 155]	Fetches a value from a column as a byte.
public virtual uint8	GetByte(uint16) [page 156]	Fetches a value from a column as a byte.

In this section:

GetByte(String^) Method [page 155]
Fetches a value from a column as a byte.

GetByte(uint16) Method [page 156]
Fetches a value from a column as a byte.

1.10.11.1 GetByte(String^) Method

Fetches a value from a column as a byte.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a byte.

1.10.11.2 GetByte(uint16) Method

Fetches a value from a column as a byte.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a byte.

1.10.12 GetBytes Method

Gets a binary chunk from the column.

Overload list

Modifier and Type	Overload name	Description
public virtual int64	GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) [page 157]	Gets a binary chunk from the column.
public virtual int64	GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) [page 157]	Gets a binary chunk from the column.

In this section:

GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) Method [page 157]
Gets a binary chunk from the column.

GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) Method [page 157]
Gets a binary chunk from the column.

1.10.12.1 GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) Method

Gets a binary chunk from the column.

Syntax

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

Parameters

cname The name of the column.
dst The buffer to hold the bytes.
count The size of the buffer in bytes.
srcOffset The offset into the value at which to start reading, or the BLOB_CONTINUE constant to continue from where the last read ended.
dstOffset The offset into the destination buffer at which to copy the bytes.

Returns

The number of bytes copied to the destination buffer. If the dst value is NULL, then the number of bytes left is returned. An empty string is returned in the dst parameter when the column is null.

Remarks

The end of the value has been reached if 0 is returned.

Use the IsNull method to differentiate between null and empty strings.

1.10.12.2 GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) Method

Gets a binary chunk from the column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

dst The buffer to hold the bytes.

count The size of the buffer in bytes.

srcOffset The offset into the value at which to start reading, or the BLOB_CONTINUE constant to continue from where the last read ended.

dstOffset The offset into the destination buffer at which to copy the bytes.

Returns

The number of bytes copied to the destination buffer. If the dst value is NULL, then this method returns the number of bytes left. An empty string is returned in the dst parameter when the column is null.

Remarks

The end of the value has been reached if 0 is returned.

Use the IsNull method to differentiate between null and empty strings.

1.10.13 GetChars Method

Gets a wide string chunk from the column.

Overload list

Modifier and Type	Overload name	Description
public virtual int64	GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) [page 159]	Gets a wide string chunk from the column.
public virtual int64	GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) [page 159]	Gets a wide string chunk from the column.

In this section:

GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) Method [page 159]
Gets a wide string chunk from the column.

GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) Method [page 159]
Gets a wide string chunk from the column.

1.10.13.1 GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) Method

Gets a wide string chunk from the column.

Syntax

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

Parameters

cname The name of the column.
dst The buffer to hold the string chunk. The string is null terminated even if it is truncated.
count The size, in characters, of the buffer.
srcOffset The offset into the value at which to start reading, or the BLOB_CONTINUE constant to continue from where the last read ended.
dstOffset The offset into the destination buffer at which to copy the characters.

Returns

The number of characters copied to the destination buffer excluding the null terminator. If the dst value is NULL, then this method returns the number of characters left in the string. An empty string is returned in the dst parameter when the column is null.

Remarks

The end of the value has been reached if 0 is returned.

Use the IsNull method to differentiate between null and empty strings.

1.10.13.2 GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) Method

Gets a wide string chunk from the column.

Syntax

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

Parameters

cid The zero-based ordinal column number.
dst The buffer to hold the string chunk. The string is null terminated even if it is truncated.
count The size, in characters, of the buffer.
srcOffset The offset into the value at which to start reading, or the BLOB_CONTINUE constant to continue from where the last read ended.
dstOffset The offset into the destination buffer at which to copy the characters.

Returns

The number of characters copied to the destination buffer excluding the null terminator. If the dst value is NULL, then this method returns the number of characters left in the string. An empty string is returned in the dst parameter when the column is null.

Remarks

The end of the value has been reached if 0 is returned.

Use the IsNull method to differentiate between null and empty strings.

1.10.14 GetDateTime Method

Fetches a value from a column as a DateTime.

Overload list

Modifier and Type	Overload name	Description
public virtual DateTime	GetDateTime(String^) [page 161]	Fetches a value from a column as a DateTime.
public virtual DateTime	GetDateTime(uint16) [page 161]	Fetches a value from a column as a DateTime.

In this section:

GetDateTime(String^) Method [page 161]
Fetches a value from a column as a DateTime.

GetDateTime(uint16) Method [page 161]

Fetches a value from a column as a DateTime.

1.10.14.1 GetDateTime(String^) Method

Fetches a value from a column as a DateTime.

Syntax

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

Parameters

cname The name of the column.

Returns

The DateTime value.

1.10.14.2 GetDateTime(uint16) Method

Fetches a value from a column as a DateTime.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The DateTime value.

1.10.15 GetDouble Method

Fetches a value from a column as a double.

Overload list

Modifier and Type	Overload name	Description
public virtual double	GetDouble(String^) [page 162]	Fetches a value from a column as a double.
public virtual double	GetDouble(uint16) [page 163]	Fetches a value from a column as a double.

In this section:

GetDouble(String^) Method [page 162]
Fetches a value from a column as a double.

GetDouble(uint16) Method [page 163]
Fetches a value from a column as a double.

1.10.15.1 GetDouble(String^) Method

Fetches a value from a column as a double.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a double.

1.10.15.2 GetDouble(uint16) Method

Fetches a value from a column as a double.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a double.

1.10.16 GetFloat Method

Fetches a value from a column as a float.

Overload list

Modifier and Type	Overload name	Description
public virtual float	GetFloat(String^) [page 164]	Fetches a value from a column as a float.
public virtual float	GetFloat(uint16) [page 164]	Fetches a value from a column as a float.

In this section:

GetFloat(String^) Method [page 164]
Fetches a value from a column as a float.

GetFloat(uint16) Method [page 164]
Fetches a value from a column as a float.

1.10.16.1 GetFloat(String^) Method

Fetches a value from a column as a float.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a float.

1.10.16.2 GetFloat(uint16) Method

Fetches a value from a column as a float.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a float.

1.10.17 GetGuid Method

Fetches a value from a column as a GUID.

Overload list

Modifier and Type	Overload name	Description
public virtual Guid	GetGuid(String^) [page 165]	Fetches a value from a column as a GUID.
public virtual Guid	GetGuid(uint16) [page 166]	Fetches a value from a column as a GUID.

In this section:

GetGuid(String^) Method [page 165]
Fetches a value from a column as a GUID.

GetGuid(uint16) Method [page 166]
Fetches a value from a column as a GUID.

1.10.17.1 GetGuid(String^) Method

Fetches a value from a column as a GUID.

Syntax

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

Parameters

cname The name of the column.

Returns

The GUID value.

1.10.17.2 GetGuid(uint16) Method

Fetches a value from a column as a GUID.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The GUID value.

1.10.18 GetInt16 Method

Fetches a value from a column as a 16-bit integer.

Overload list

Modifier and Type	Overload name	Description
public virtual int16	GetInt16(String^) [page 167]	Fetches a value from a column as a 16-bit integer.
public virtual int16	GetInt16(uint16) [page 167]	Fetches a value from a column as a 16-bit integer.

In this section:

GetInt16(String^) Method [page 167]
Fetches a value from a column as a 16-bit integer.

GetInt16(uint16) Method [page 167]
Fetches a value from a column as a 16-bit integer.

1.10.18.1 GetInt16(String^) Method

Fetches a value from a column as a 16-bit integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as an integer.

1.10.18.2 GetInt16(uint16) Method

Fetches a value from a column as a 16-bit integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as an integer.

1.10.19 GetInt32 Method

Fetches a value from a column as an integer.

Overload list

Modifier and Type	Overload name	Description
public virtual int	GetInt32(String^) [page 168]	Fetches a value from a column as an integer.
public virtual int	GetInt32(uint16) [page 169]	Fetches a value from a column as an integer.

In this section:

GetInt32(String^) Method [page 168]
Fetches a value from a column as an integer.

GetInt32(uint16) Method [page 169]
Fetches a value from a column as an integer.

1.10.19.1 GetInt32(String^) Method

Fetches a value from a column as an integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as an integer.

1.10.19.2 GetInt32(uint16) Method

Fetches a value from a column as an integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as an integer.

1.10.20 GetInt64 Method

Fetches a value from a column as a 64-bit integer.

Overload list

Modifier and Type	Overload name	Description
public virtual int64	GetInt64(String^) [page 170]	Fetches a value from a column as a 64-bit integer.
public virtual int64	GetInt64(uint16) [page 170]	Fetches a value from a column as a 64-bit integer.

In this section:

GetInt64(String^) Method [page 170]
  Fetches a value from a column as a 64-bit integer.

GetInt64(uint16) Method [page 170]
  Fetches a value from a column as a 64-bit integer.

1.10.20.1 GetInt64(String^) Method

Fetches a value from a column as a 64-bit integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as an integer.

1.10.20.2 GetInt64(uint16) Method

Fetches a value from a column as a 64-bit integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as an integer.

1.10.21 GetResultSetSchema() Method

Returns an object that can be used to get information about the result set.

Syntax

```
public ResultSetSchema GetResultSetSchema ()
```

Returns

A ResultSetSchema object that can be used to get information about the result set.

1.10.22 GetRowCount(unsigned int) Method

Gets the number of rows in the table.

Syntax

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

Parameters

threshold The limit on the number of rows to count. Set to 0 to indicate no limit.

Returns

The number of rows in the table.

Remarks

This method is equivalent to executing the following statement:

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

1.10.23 GetState() Method

Gets the internal state of the cursor.

Syntax

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

Returns

The state of the cursor.

1.10.24 GetString Method

Fetches a value from a column as a string.

Overload list

Modifier and Type	Overload name	Description
public virtual String	GetString(String^) [page 172]	Fetches a value from a column as a string.
public virtual String	GetString(uint16) [page 173]	Fetches a value from a column as a string.

In this section:

GetString(String^) Method [page 172]
Fetches a value from a column as a string.

GetString(uint16) Method [page 173]
Fetches a value from a column as a string.

1.10.24.1 GetString(String^) Method

Fetches a value from a column as a string.

Syntax

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

Parameters

cname The name of the column.

Returns

The string value.

1.10.24.2 GetString(uint16) Method

Fetches a value from a column as a string.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The string value.

1.10.25 GetStringLength Method

Gets the string length of the value of a column.

Overload list

Modifier and Type	Overload name	Description
public virtual int64	GetStringLength(String^) [page 174]	Gets the string length of the value of a column.
public virtual int64	GetStringLength(uint16) [page 175]	Gets the string length of the value of a column.

In this section:

GetStringLength(String^) Method [page 174]
Gets the string length of the value of a column.

GetStringLength(uint16) Method [page 175]
Gets the string length of the value of a column.

1.10.25.1 GetStringLength(String^) Method

Gets the string length of the value of a column.

Syntax

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

Parameters

cname The name of the column.

Returns

The number of characters of a string type column value.

1.10.25.2 GetStringLength(uint16) Method

Gets the string length of the value of a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The number of characters of a string type column value.

1.10.26 GetUInt16 Method

Fetches a value from a column as a 16-bit unsigned integer.

Overload list

Modifier and Type	Overload name	Description
public virtual uint16	GetUInt16(String^) [page 176]	Fetches a value from a column as a 16-bit unsigned integer.
public virtual uint16	GetUInt16(uint16) [page 176]	Fetches a value from a column as a 16-bit unsigned integer.

In this section:

GetUInt16(String^) Method [page 176]
Fetches a value from a column as a 16-bit unsigned integer.

GetUInt16(uint16) Method [page 176]
Fetches a value from a column as a 16-bit unsigned integer.

1.10.26.1 GetUInt16(String^) Method

Fetches a value from a column as a 16-bit unsigned integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as an unsigned integer.

1.10.26.2 GetUInt16(uint16) Method

Fetches a value from a column as a 16-bit unsigned integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as an unsigned integer.

1.10.27 GetUInt32 Method

Fetches a value from a column as a 32-bit unsigned integer.

Overload list

Modifier and Type	Overload name	Description
public virtual unsigned int	GetUInt32(String^) [page 177]	Fetches a value from a column as a 32-bit unsigned integer.
public virtual unsigned int	GetUInt32(uint16) [page 178]	Fetches a value from a column as a 32-bit unsigned integer.

In this section:

GetUInt32(String^) Method [page 177]
Fetches a value from a column as a 32-bit unsigned integer.

GetUInt32(uint16) Method [page 178]
Fetches a value from a column as a 32-bit unsigned integer.

1.10.27.1 GetUInt32(String^) Method

Fetches a value from a column as a 32-bit unsigned integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as an unsigned integer.

1.10.27.2 GetUInt32(uint16) Method

Fetches a value from a column as a 32-bit unsigned integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as an unsigned integer.

1.10.28 GetUInt64 Method

Fetches a value from a column as a 64-bit unsigned integer.

Overload list

Modifier and Type	Overload name	Description
public virtual uint64	GetUInt64(String^) [page 179]	Fetches a value from a column as a 64-bit unsigned integer.
public virtual uint64	GetUInt64(uint16) [page 179]	Fetches a value from a column as a 64-bit unsigned integer.

In this section:

GetUInt64(String^) Method [page 179]
Fetches a value from a column as a 64-bit unsigned integer.

GetUInt64(uint16) Method [page 179]
Fetches a value from a column as a 64-bit unsigned integer.

1.10.28.1 GetUInt64(String^) Method

Fetches a value from a column as a 64-bit unsigned integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as an unsigned integer.

1.10.28.2 GetUInt64(uint16) Method

Fetches a value from a column as a 64-bit unsigned integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as an unsigned integer.

1.10.29 IsNull Method

Checks whether a column value is NULL.

Overload list

Modifier and Type	Overload name	Description
public virtual bool	IsNull(String^) [page 180]	Checks whether a column value is NULL.
public virtual bool	IsNull(uint16) [page 181]	Checks whether a column value is NULL.

In this section:

IsNull(String^) Method [page 180]
Checks whether a column value is NULL.

IsNull(uint16) Method [page 181]
Checks whether a column value is NULL.

1.10.29.1 IsNull(String^) Method

Checks whether a column value is NULL.

Syntax

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

Parameters

cname The name of the column.

Returns

True if the column value is NULL.

1.10.29.2 IsNull(uint16) Method

Checks whether a column value is NULL.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

True if the column value is NULL.

1.10.30 Last() Method

Moves the cursor to the last row.

Syntax

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

1.10.31 Next() Method

Moves the cursor forward one row.

Syntax

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

Returns

True if the cursor successfully moves forward. An error can still be signaled when the cursor moves successfully to the next row. For example, there could be conversion errors while evaluating the SELECT expressions. In this case, errors are also returned when retrieving the column values. False is returned if the cursor fails to move forward. For example, there is not a next row. In this case, the resulting cursor position is set after the last row.

1.10.32 Previous() Method

Moves the cursor back one row.

Syntax

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

Returns

True if the cursor successfully moves back one row. False if it fails to move backward. The resulting cursor position is set before the first row.

1.10.33 Relative(int) Method

Moves the cursor by offset rows from the current cursor position.

Syntax

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

Parameters

offset The number of rows to move.

1.10.34 SetBool Method

Sets a column to a boolean value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetBool(String^, bool) [page 183]	Sets a column to a boolean value.
public virtual void	SetBool(uint16, bool) [page 183]	Sets a column to a boolean value.

In this section:

SetBool(String^, bool) Method [page 183]
Sets a column to a boolean value.

SetBool(uint16, bool) Method [page 183]
Sets a column to a boolean value.

1.10.34.1 SetBool(String^, bool) Method

Sets a column to a boolean value.

Syntax

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

Parameters

cname The name of the column.
value The boolean value.

1.10.34.2 SetBool(uint16, bool) Method

Sets a column to a boolean value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The boolean value.

1.10.35 SetByte Method

Sets a column to a byte value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetByte(String^, uint8) [page 184]	Sets a column to a byte value.
public virtual void	SetByte(uint16, uint8) [page 185]	Sets a column to a byte value.

In this section:

SetByte(String^, uint8) Method [page 184]
    Sets a column to a byte value.

SetByte(uint16, uint8) Method [page 185]
    Sets a column to a byte value.

1.10.35.1 SetByte(String^, uint8) Method

Sets a column to a byte value.

Syntax

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

Parameters

cname The name of the column.
value The byte value.

1.10.35.2 SetByte(uint16, uint8) Method

Sets a column to a byte value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The byte value.

1.10.36 SetBytes Method

Sets a column to a binary value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetBytes(String^, const Array< uint8 >^, int) [page 186]	Sets a column to a binary value.
public virtual void	SetBytes(uint16, const Array< uint8 >^, int) [page 186]	Sets a column to a binary value.

In this section:

SetBytes(String^, const Array< uint8 >^, int) Method [page 186]
Sets a column to a binary value.

SetBytes(uint16, const Array< uint8 >^, int) Method [page 186]
Sets a column to a binary value.

1.10.36.1 SetBytes(String^, const Array< uint8 >^, int) Method

Sets a column to a binary value.

Syntax

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

Parameters

cname The name of the column.
value The binary value. Passing NULL is equivalent to calling the SetNull method.
length The length in bytes to get from the byte array value.

1.10.36.2 SetBytes(uint16, const Array< uint8 >^, int) Method

Sets a column to a binary value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The binary value. Passing NULL is equivalent to calling the SetNull method.
length The length in bytes to get from the byte array value.

1.10.37 SetDateTime Method

Sets a column to a DateTime value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetDateTime(String^, DateTime) [page 187]	Sets a column to a DateTime value.
public virtual void	SetDateTime(uint16, DateTime) [page 188]	Sets a column to a DateTime value.

In this section:

SetDateTime(String^, DateTime) Method [page 187]
Sets a column to a DateTime value.

SetDateTime(uint16, DateTime) Method [page 188]
Sets a column to a DateTime value.

1.10.37.1 SetDateTime(String^, DateTime) Method

Sets a column to a DateTime value.

Syntax

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

Parameters

cname The name of the column.
value The DateTime value.

1.10.37.2 SetDateTime(uint16, DateTime) Method

Sets a column to a DateTime value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The DateTime value.

1.10.38 SetDefault Method

Sets a column to its default value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetDefault(String^) [page 188]	Sets a column to its default value.
public virtual void	SetDefault(uint16) [page 189]	Sets a column to its default value.

In this section:

SetDefault(String^) Method [page 188]
Sets a column to its default value.

SetDefault(uint16) Method [page 189]
Sets a column to its default value.

1.10.38.1 SetDefault(String^) Method

Sets a column to its default value.

Syntax

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

Parameters

cname The name of the column.

1.10.38.2 SetDefault(uint16) Method

Sets a column to its default value.

Syntax

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

Parameters

cid The zero-based ordinal column number.

1.10.39 SetDouble Method

Sets a column to a double value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetDouble(String^, double) [page 190]	Sets a column to a double value.
public virtual void	SetDouble(uint16, double) [page 190]	Sets a column to a double value.

In this section:

SetDouble(String^, double) Method [page 190]
Sets a column to a double value.

SetDouble(uint16, double) Method [page 190]
Sets a column to a double value.

1.10.39.1 SetDouble(String^, double) Method

Sets a column to a double value.

Syntax

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

Parameters

cname The name of the column.
value The double value.

1.10.39.2 SetDouble(uint16, double) Method

Sets a column to a double value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The double value.

1.10.40 SetFloat Method

Sets a column to a float value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetFloat(String^, float) [page 191]	Sets a column to a float value.

Modifier and Type	Overload name	Description
public virtual void	SetFloat(uint16, float) [page 191]	Sets a column to a float value.

In this section:

SetFloat(String^, float) Method [page 191]
Sets a column to a float value.

SetFloat(uint16, float) Method [page 191]
Sets a column to a float value.

1.10.40.1 SetFloat(String^, float) Method

Sets a column to a float value.

Syntax

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

Parameters

cname The name of the column.
value The float value.

1.10.40.2 SetFloat(uint16, float) Method

Sets a column to a float value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The float value.

1.10.41 SetGuid Method

Sets a column to a GUID value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetGuid(String^, Guid) [page 192]	Sets a column to a GUID value.
public virtual void	SetGuid(uint16, Guid) [page 192]	Sets a column to a GUID value.

In this section:

SetGuid(String^, Guid) Method [page 192]
Sets a column to a GUID value.

SetGuid(uint16, Guid) Method [page 192]
Sets a column to a GUID value.

1.10.41.1 SetGuid(String^, Guid) Method

Sets a column to a GUID value.

Syntax

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

Parameters

cname The name of the column.
value The GUID value.

1.10.41.2 SetGuid(uint16, Guid) Method

Sets a column to a GUID value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The GUID value.

1.10.42 SetInt16 Method

Sets a column to an integer value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetInt16(String^, int16) [page 193]	Sets a column to an integer value.
public virtual void	SetInt16(uint16, int16) [page 194]	Sets a column to an integer value.

In this section:

SetInt16(String^, int16) Method [page 193]
    Sets a column to an integer value.

SetInt16(uint16, int16) Method [page 194]
    Sets a column to an integer value.

1.10.42.1 SetInt16(String^, int16) Method

Sets a column to an integer value.

Syntax

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

Parameters

cname The name of the column.
value The signed integer value.

1.10.42.2 SetInt16(uint16, int16) Method

Sets a column to an integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The signed integer value.

1.10.43 SetInt32 Method

Sets a column to an integer value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetInt32(String^, int) [page 194]	Sets a column to an integer value.
public virtual void	SetInt32(uint16, int) [page 195]	Sets a column to an integer value.

In this section:

SetInt32(String^, int) Method [page 194]
Sets a column to an integer value.

SetInt32(uint16, int) Method [page 195]
Sets a column to an integer value.

1.10.43.1 SetInt32(String^, int) Method

Sets a column to an integer value.

Syntax

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

Parameters

cname The name of the column.
value The signed integer value.

1.10.43.2 SetInt32(uint16, int) Method

Sets a column to an integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The signed integer value.

1.10.44 SetInt64 Method

Sets a column to an integer value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetInt64(String^, int64) [page 196]	Sets a column to an integer value.
public virtual void	SetInt64(uint16, int64) [page 196]	Sets a column to an integer value.

In this section:

SetInt64(String^, int64) Method [page 196]
Sets a column to an integer value.

SetInt64(uint16, int64) Method [page 196]
Sets a column to an integer value.

1.10.44.1 SetInt64(String^, int64) Method

Sets a column to an integer value.

Syntax

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

Parameters

cname The name of the column.
value The signed integer value.

1.10.44.2 SetInt64(uint16, int64) Method

Sets a column to an integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The signed integer value.

1.10.45 SetNull Method

Sets a column to null.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetNull(String^) [page 197]	Sets a column to null.

Modifier and Type	Overload name	Description
public virtual void	SetNull(uint16) [page 197]	Sets a column to null.

In this section:

SetNull(String^) Method [page 197]
Sets a column to null.

SetNull(uint16) Method [page 197]
Sets a column to null.

1.10.45.1 SetNull(String^) Method

Sets a column to null.

Syntax

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

Parameters

cname The name of the column.

1.10.45.2 SetNull(uint16) Method

Sets a column to null.

Syntax

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

Parameters

cid The zero-based ordinal column number.

1.10.46 SetString Method

Sets a column to a string value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetString(String^, String^) [page 198]	Sets a column to a string value.
public virtual void	SetString(uint16, String^) [page 198]	Sets a column to a string value.

In this section:

SetString(String^, String^) Method [page 198]
Sets a column to a string value.

SetString(uint16, String^) Method [page 198]
Sets a column to a string value.

1.10.46.1 SetString(String^, String^) Method

Sets a column to a string value.

Syntax

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

Parameters

cname The name of the column.
value The string value.

1.10.46.2 SetString(uint16, String^) Method

Sets a column to a string value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The string value.

1.10.47 SetUInt16 Method

Sets a column to an unsigned 16-bit integer value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetUInt16(String^, uint16) [page 199]	Sets a column to an unsigned 16-bit integer value.
public virtual void	SetUInt16(uint16, uint16) [page 200]	Sets a column to an unsigned 16-bit integer value.

In this section:

SetUInt16(String^, uint16) Method [page 199]
Sets a column to an unsigned 16-bit integer value.

SetUInt16(uint16, uint16) Method [page 200]
Sets a column to an unsigned 16-bit integer value.

1.10.47.1 SetUInt16(String^, uint16) Method

Sets a column to an unsigned 16-bit integer value.

Syntax

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

Parameters

cname The name of the column.
value The unsigned integer value.

1.10.47.2 SetUInt16(uint16, uint16) Method

Sets a column to an unsigned 16-bit integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.

value The unsigned integer value.

1.10.48 SetUInt32 Method

Sets a column to an unsigned 32-bit integer value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetUInt32(String^, unsigned int) [page 201]	Sets a column to an unsigned 32-bit in-teger value.
public virtual void	SetUInt32(uint16, unsigned int) [page 201]	Sets a column to an unsigned 32-bit in-teger value.

In this section:

SetUInt32(String^, unsigned int) Method [page 201]
Sets a column to an unsigned 32-bit integer value.

SetUInt32(uint16, unsigned int) Method [page 201]
Sets a column to an unsigned 32-bit integer value.

1.10.48.1 SetUInt32(String^, unsigned int) Method

Sets a column to an unsigned 32-bit integer value.

Syntax

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

Parameters

cname The name of the column.
value The unsigned integer value.

1.10.48.2 SetUInt32(uint16, unsigned int) Method

Sets a column to an unsigned 32-bit integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The unsigned integer value.

1.10.49 SetUInt64 Method

Sets a column to an unsigned 64-bit integer value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetUInt64(String^, uint64) [page 202]	Sets a column to an unsigned 64-bit integer value.
public virtual void	SetUInt64(uint16, uint64) [page 203]	Sets a column to an unsigned 64-bit integer value.

In this section:

SetUInt64(String^, uint64) Method [page 202]
    Sets a column to an unsigned 64-bit integer value.

SetUInt64(uint16, uint64) Method [page 203]
    Sets a column to an unsigned 64-bit integer value.

1.10.49.1 SetUInt64(String^, uint64) Method

Sets a column to an unsigned 64-bit integer value.

Syntax

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

Parameters

cname The name of the column.
value The unsigned integer value.

1.10.49.2 SetUInt64(uint16, uint64) Method

Sets a column to an unsigned 64-bit integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The unsigned integer value.

1.10.50 Update() Method

Updates the current row.

Syntax

```
public virtual void Update ()
```

1.10.51 UpdateBegin() Method

Selects the update mode for setting columns.

Syntax

```
public virtual void UpdateBegin ()
```

Remarks

Columns in the primary key cannot be modified when UltraLite is in update mode.

1.11 ResultSetSchema Class

Represents the schema of an UltraLite result set. ResultSet.GetResultSetSchema or PreparedStatement.GetResultSetSchema return a ResultSetSchema.

Namespace

```
UltraLite
```

Syntax

```
public ref class  sealed : CursorSchema
```

Members

All members of ResultSetSchema, including inherited members.

Methods

Modifier and Type	Method	Description
public virtual uint16	GetColumnCount() [page 205]	Gets the number of columns in the result set or table.
public virtual uint16	GetColumnID(String^) [page 205]	Gets the zero-based column ID from its name.
public virtual String	GetColumnName(uint16, uint16type) [page 206]	Gets the name of a column given its zero-based ID.
public virtual uint16	GetColumnPrecision(uint16) [page 207]	Gets the precision of a numeric column.
public virtual uint16	GetColumnScale(uint16) [page 207]	Gets the scale of a numeric column.
public virtual int	GetColumnSize(uint16) [page 208]	Gets the size of the column.
public virtual uint16	GetColumnSQLType(uint16) [page 208]	Gets the SQL type of a column.
public virtual uint16	GetColumnType(uint16) [page 209]	Gets the storage/host variable type of a column.
public virtual bool	IsAliased(uint16) [page 209]	Indicates whether the column in a result set was given an alias.

In this section:

GetColumnCount() Method [page 205]
Gets the number of columns in the result set or table.

GetColumnID(String^) Method [page 205]
    Gets the zero-based column ID from its name.

GetColumnName(uint16, uint16type) Method [page 206]
    Gets the name of a column given its zero-based ID.

GetColumnPrecision(uint16) Method [page 207]
    Gets the precision of a numeric column.

GetColumnScale(uint16) Method [page 207]
    Gets the scale of a numeric column.

GetColumnSize(uint16) Method [page 208]
    Gets the size of the column.

GetColumnSQLType(uint16) Method [page 208]
    Gets the SQL type of a column.

GetColumnType(uint16) Method [page 209]
    Gets the storage/host variable type of a column.

IsAliased(uint16) Method [page 209]
    Indicates whether the column in a result set was given an alias.

1.11.1 GetColumnCount() Method

Gets the number of columns in the result set or table.

Syntax

```
public virtual uint16 GetColumnCount ()
```

Returns

The number of columns in the result set or table.

1.11.2 GetColumnID(String^) Method

Gets the zero-based column ID from its name.

Syntax

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

Parameters

columnName The column name.

Returns

The column ID.

1.11.3 GetColumnName(uint16, uint16type) Method

Gets the name of a column given its zero-based ID.

Syntax

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

Parameters

cid The zero-based ordinal column number.
type The desired column name type.

Returns

A string containing the column name, if found.

Remarks

Depending on the type selected and how the column was declared in the SELECT statement, the column name may be returned in the form [table-name].[column-name].

The type parameter is used to specify what type of column name to return.

1.11.4 GetColumnPrecision(uint16) Method

Gets the precision of a numeric column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The precision

1.11.5 GetColumnScale(uint16) Method

Gets the scale of a numeric column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The scale.

1.11.6 GetColumnSize(uint16) Method

Gets the size of the column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column size.

1.11.7 GetColumnSQLType(uint16) Method

Gets the SQL type of a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

SQLTYPE_BAD_INDEX if the column does not exist.

1.11.8 GetColumnType(uint16) Method

Gets the storage/host variable type of a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

TYPE_BAD_INDEX if the column does not exist.

1.11.9 IsAliased(uint16) Method

Indicates whether the column in a result set was given an alias.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

True if the column is aliased; otherwise, returns false.

1.12 Table Class

Represents a table in an UltraLite database. Connection.OpenTable returns a Table.

Namespace

```
UltraLite
```

Syntax

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

Members

All members of Table, including inherited members.

Methods

Modifier and Type	Method	Description
public virtual void	AfterLast() [page 216]	Moves the cursor after the last row.
public virtual void	AppendBytes [page 217]	Appends bytes to a column.
public virtual void	AppendChars [page 218]	Appends a string chunk to a column.
public virtual void	BeforeFirst() [page 220]	Moves the cursor before the first row.
public virtual void	Close() [page 220]	Destroys this object.
public virtual void	Delete() [page 221]	Deletes the current row and moves the cursor to the next valid row.
public void	DeleteAllRows() [page 221]	Deletes all rows from a table.
public virtual void	DeleteNamed(String^) [page 221]	Deletes the current row and moves the cursor to the next valid row.
public bool	Find(uint16) [page 222]	Performs an exact match lookup based on the current index by scanning for-ward through the table.
public void	FindBegin() [page 222]	Prepares to perform a new Find call on a table by entering find mode.
public bool	FindFirst(uint16) [page 223]	Performs an exact match lookup based on the current index by scanning for-ward through the table.

Modifier and Type	Method	Description
public bool	FindLast(uint16) [page 223]	Performs an exact match lookup based on the current index by scanning backward through the table.
public bool	FindNext(uint16) [page 224]	Gets the next row that exactly matches the index.
public bool	FindPrevious(uint16) [page 225]	Gets the previous row that exactly matches the index.
public virtual void	First() [page 225]	Moves the cursor to the first row.
public virtual int64	GetBinaryLength [page 225]	Gets the binary length of the value of a column.
public virtual bool	GetBool [page 227]	Fetches a value from a column as a boolean.
public virtual uint8	GetByte [page 229]	Fetches a value from a column as a byte.
public virtual int64	GetBytes [page 230]	Gets a binary chunk from the column.
public virtual int64	GetChars [page 232]	Gets a wide string chunk from the column.
public virtual DateTime	GetDateTime [page 234]	Fetches a value from a column as a DateTime.
public virtual double	GetDouble [page 236]	Fetches a value from a column as a double.
public virtual float	GetFloat [page 237]	Fetches a value from a column as a float.
public virtual Guid	GetGuid [page 239]	Fetches a value from a column as a GUID.
public virtual int16	GetInt16 [page 240]	Fetches a value from a column as a 16-bit integer.
public virtual int	GetInt32 [page 242]	Fetches a value from a column as an integer.
public virtual int64	GetInt64 [page 243]	Fetches a value from a column as a 64-bit integer.
public virtual unsigned int	GetRowCount(unsigned int) [page 245]	Gets the number of rows in the table.
public virtual uint8	GetState() [page 245]	Gets the internal state of the cursor.
public virtual String	GetString [page 246]	Fetches a value from a column as a string.
public virtual int64	GetStringLength [page 247]	Gets the string length of the value of a column.
public TableSchema	GetTableSchema() [page 249]	Returns a TableSchema object that can be used to get schema information about the table.
public virtual uint16	GetUInt16 [page 249]	Fetches a value from a column as a 16-bit unsigned integer.

Modifier and Type	Method	Description
public virtual unsigned int	GetUInt32 [page 251]	Fetches a value from a column as a 32-bit unsigned integer.
public virtual uint64	GetUInt64 [page 252]	Fetches a value from a column as a 64-bit unsigned integer.
public void	Insert() [page 254]	Inserts a new row into the table.
public void	InsertBegin() [page 254]	Selects the insert mode for setting columns.
public virtual bool	IsNull [page 254]	Checks whether a column value is NULL.
public virtual void	Last() [page 256]	Moves the cursor to the last row.
public bool	Lookup(uint16) [page 256]	Performs a lookup based on the current index by scanning forward through the table.
public bool	LookupBackward(uint16) [page 256]	Performs a lookup based on the current index by scanning backward through the table.
public void	LookupBegin() [page 257]	Prepares to perform a new lookup on a table.
public bool	LookupForward(uint16) [page 257]	Performs a lookup based on the current index by scanning forward through the table.
public virtual bool	Next() [page 258]	Moves the cursor forward one row.
public virtual bool	Previous() [page 258]	Moves the cursor back one row.
public virtual void	Relative(int) [page 259]	Moves the cursor by offset rows from the current cursor position.
public virtual void	SetBool [page 259]	Sets a column to a boolean value.
public virtual void	SetByte [page 260]	Sets a column to a byte value.
public virtual void	SetBytes [page 262]	Sets a column to a binary value.
public virtual void	SetDateTime [page 263]	Sets a column to a DateTime value.
public virtual void	SetDefault [page 264]	Sets a column to its default value.
public virtual void	SetDouble [page 266]	Sets a column to a double value.
public virtual void	SetFloat [page 267]	Sets a column to a float value.
public virtual void	SetGuid [page 268]	Sets a column to a GUID value.
public virtual void	SetInt16 [page 269]	Sets a column to an integer value.
public virtual void	SetInt32 [page 270]	Sets a column to an integer value.
public virtual void	SetInt64 [page 272]	Sets a column to an integer value.
public virtual void	SetNull [page 273]	Sets a column to null.
public virtual void	SetString [page 274]	Sets a column to a string value.

Modifier and Type	Method	Description
public virtual void	SetUInt16 [page 275]	Sets a column to an unsigned 16-bit integer value.
public virtual void	SetUInt32 [page 277]	Sets a column to an unsigned 32-bit integer value.
public virtual void	SetUInt64 [page 278]	Sets a column to an unsigned 64-bit integer value.
public void	TruncateTable() [page 279]	Truncates the table and temporarily activates STOP SYNCHRONIZATION DELETE.
public virtual void	Update() [page 280]	Updates the current row.
public virtual void	UpdateBegin() [page 280]	Selects the update mode for setting columns.

In this section:

AfterLast() Method [page 216]
Moves the cursor after the last row.

AppendBytes Method [page 217]
Appends bytes to a column.

AppendChars Method [page 218]
Appends a string chunk to a column.

BeforeFirst() Method [page 220]
Moves the cursor before the first row.

CloseObject() Method [page 220]
Destroys this object.

Delete() Method [page 221]
Deletes the current row and moves the cursor to the next valid row.

DeleteAllRows() Method [page 221]
Deletes all rows from a table.

DeleteNamed(String^) Method [page 221]
Deletes the current row and moves the cursor to the next valid row.

Find(uint16) Method [page 222]
Performs an exact match lookup based on the current index by scanning forward through the table.

FindBegin() Method [page 222]
Prepares to perform a new Find call on a table by entering find mode.

FindFirst(uint16) Method [page 223]
Performs an exact match lookup based on the current index by scanning forward through the table.

FindLast(uint16) Method [page 223]
Performs an exact match lookup based on the current index by scanning backward through the table.

FindNext(uint16) Method [page 224]
Gets the next row that exactly matches the index.

FindPrevious(uint16) Method [page 225]
Gets the previous row that exactly matches the index.

First() Method [page 225]
Moves the cursor to the first row.

GetBinaryLength Method [page 225]
Gets the binary length of the value of a column.

GetBool Method [page 227]
Fetches a value from a column as a boolean.

GetByte Method [page 229]
Fetches a value from a column as a byte.

GetBytes Method [page 230]
Gets a binary chunk from the column.

GetChars Method [page 232]
Gets a wide string chunk from the column.

GetDateTime Method [page 234]
Fetches a value from a column as a DateTime.

GetDouble Method [page 236]
Fetches a value from a column as a double.

GetFloat Method [page 237]
Fetches a value from a column as a float.

GetGuid Method [page 239]
Fetches a value from a column as a GUID.

GetInt16 Method [page 240]
Fetches a value from a column as a 16-bit integer.

GetInt32 Method [page 242]
Fetches a value from a column as an integer.

GetInt64 Method [page 243]
Fetches a value from a column as a 64-bit integer.

GetRowCount(unsigned int) Method [page 245]
Gets the number of rows in the table.

GetState() Method [page 245]
Gets the internal state of the cursor.

GetString Method [page 246]
Fetches a value from a column as a string.

GetStringLength Method [page 247]
Gets the string length of the value of a column.

GetTableSchema() Method [page 249]
Returns a TableSchema object that can be used to get schema information about the table.

GetUInt16 Method [page 249]
Fetches a value from a column as a 16-bit unsigned integer.

GetUInt32 Method [page 251]
Fetches a value from a column as a 32-bit unsigned integer.

GetUInt64 Method [page 252]
: Fetches a value from a column as a 64-bit unsigned integer.

Insert() Method [page 254]
: Inserts a new row into the table.

InsertBegin() Method [page 254]
: Selects the insert mode for setting columns.

IsNull Method [page 254]
: Checks whether a column value is NULL.

Last() Method [page 256]
: Moves the cursor to the last row.

Lookup(uint16) Method [page 256]
: Performs a lookup based on the current index by scanning forward through the table.

LookupBackward(uint16) Method [page 256]
: Performs a lookup based on the current index by scanning backward through the table.

LookupBegin() Method [page 257]
: Prepares to perform a new lookup on a table.

LookupForward(uint16) Method [page 257]
: Performs a lookup based on the current index by scanning forward through the table.

Next() Method [page 258]
: Moves the cursor forward one row.

Previous() Method [page 258]
: Moves the cursor back one row.

Relative(int) Method [page 259]
: Moves the cursor by offset rows from the current cursor position.

SetBool Method [page 259]
: Sets a column to a boolean value.

SetByte Method [page 260]
: Sets a column to a byte value.

SetBytes Method [page 262]
: Sets a column to a binary value.

SetDateTime Method [page 263]
: Sets a column to a DateTime value.

SetDefault Method [page 264]
: Sets a column to its default value.

SetDouble Method [page 266]
: Sets a column to a double value.

SetFloat Method [page 267]
: Sets a column to a float value.

SetGuid Method [page 268]
: Sets a column to a GUID value.

SetInt16 Method [page 269]
: Sets a column to an integer value.

SetInt32 Method [page 270]
    Sets a column to an integer value.

SetInt64 Method [page 272]
    Sets a column to an integer value.

SetNull Method [page 273]
    Sets a column to null.

SetString Method [page 274]
    Sets a column to a string value.

SetUInt16 Method [page 275]
    Sets a column to an unsigned 16-bit integer value.

SetUInt32 Method [page 277]
    Sets a column to an unsigned 32-bit integer value.

SetUInt64 Method [page 278]
    Sets a column to an unsigned 64-bit integer value.

TruncateTable() Method [page 279]
    Truncates the table and temporarily activates STOP SYNCHRONIZATION DELETE.

Update() Method [page 280]
    Updates the current row.

UpdateBegin() Method [page 280]
    Selects the update mode for setting columns.

1.12.1 AfterLast() Method

Moves the cursor after the last row.

Syntax

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

1.12.2 AppendBytes Method

Appends bytes to a column.

Overload list

Modifier and Type	Overload name	Description
public virtual void	AppendBytes(String^, const Array< uint8 >^, int, int) [page 217]	Appends bytes to a column.
public virtual void	AppendBytes(uint16, const Array< uint8 >^, int, int) [page 218]	Appends bytes to a column.

In this section:

AppendBytes(String^, const Array< uint8 >^, int, int) Method [page 217]
Appends bytes to a column.

AppendBytes(uint16, const Array< uint8 >^, int, int) Method [page 218]
Appends bytes to a column.

1.12.2.1 AppendBytes(String^, const Array< uint8 >^, int, int) Method

Appends bytes to a column.

Syntax

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

Parameters

cname The name of the column.
value The byte chunk to append.
offset The offset into the byte chunk at which to start.
size The size of the byte chunk in bytes.

Remarks

The given bytes are appended to the end of the column written so far by AppendBytes method calls.

1.12.2.2 AppendBytes(uint16, const Array< uint8 >^, int, int) Method

Appends bytes to a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The byte chunk to append.
offset The offset into the byte chunk at which to start.
size The size of the byte chunk in bytes.

Remarks

The given bytes are appended to the end of the column written so far by AppendBytes method calls.

1.12.3 AppendChars Method

Appends a string chunk to a column.

Overload list

Modifier and Type	Overload name	Description
public virtual void	AppendChars(String^, const Array< wchar_t >^, int, int) [page 219]	Appends a string chunk to a column.

Modifier and Type	Overload name	Description
public virtual void	AppendChars(uint16, const Array< wchar_t >^, int, int) [page 220]	Appends a string chunk to a column.

In this section:

AppendChars(String^, const Array< wchar_t >^, int, int) Method [page 219]
Appends a string chunk to a column.

AppendChars(uint16, const Array< wchar_t >^, int, int) Method [page 220]
Appends a string chunk to a column.

1.12.3.1 AppendChars(String^, const Array< wchar_t >^, int, int) Method

Appends a string chunk to a column.

Syntax

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

Parameters

cname The name of the column.
value The string chunk to append.
offset The offset into the string chunk at which to start.
size The length of the string chunk in characters.

Remarks

This method appends the given string to the end of the string written so far by AppendChars method calls.

1.12.3.2 AppendChars(uint16, const Array< wchar_t >^, int, int) Method

Appends a string chunk to a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The string chunk to append.
offset The offset into the string chunk at which to start.
size The length of the string chunk in characters.

Remarks

This method appends the given string to the end of the string written so far by AppendChars method calls.

1.12.4 BeforeFirst() Method

Moves the cursor before the first row.

Syntax

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

1.12.5 CloseObject() Method

Destroys this object.

Syntax

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

1.12.6 Delete() Method

Deletes the current row and moves the cursor to the next valid row.

Syntax

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

1.12.7 DeleteAllRows() Method

Deletes all rows from a table.

Syntax

```
public void DeleteAllRows ()
```

Remarks

In some applications, you may need to delete all rows from a table before downloading a new set of data into the table. If you call the Connection.StopSynchronizationDelete on the connection, then the deleted rows are not synchronized.

i Note

Any uncommitted inserts from other connections are not deleted. They are also not deleted if the other connection performs a rollback after it calls the DeleteAllRows method.

If this table has been opened without an index, then it is considered read only and data cannot be deleted.

1.12.8 DeleteNamed(String^) Method

Deletes the current row and moves the cursor to the next valid row.

Syntax

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

Parameters

tableName A table name or its correlation to delete from.

1.12.9 Find(uint16) Method

Performs an exact match lookup based on the current index by scanning forward through the table.

Syntax

```
public bool Find (ncols)
```

Parameters

ncols For composite indexes, the number of columns to use during the search.

Returns

If no row matches the index value, then the cursor position is set after the last row and the method returns false.

Remarks

To specify the value to search for, set the column value for each column in the index. The cursor is positioned on the first row that exactly matches the index value.

1.12.10 FindBegin() Method

Prepares to perform a new Find call on a table by entering find mode.

Syntax

```
public void FindBegin ()
```

Remarks

You can only set columns in the index that the table was opened with. This method cannot be called if the table was opened without an index.

1.12.11 FindFirst(uint16) Method

Performs an exact match lookup based on the current index by scanning forward through the table.

Syntax

```
public bool FindFirst (ncols)
```

Parameters

ncols For composite indexes, the number of columns to use during the search.

Returns

If no row matches the index value, then the cursor position is set after the last row and the method returns false.

Remarks

To specify the value to search for, set the column value for each column in the index. The cursor is positioned on the first row that exactly matches the index value.

1.12.12 FindLast(uint16) Method

Performs an exact match lookup based on the current index by scanning backward through the table.

Syntax

```
public bool FindLast (ncols)
```

Parameters

ncols For composite indexes, the number of columns to use during the search.

Returns

If no row matches the index value, then the cursor position is set before the first row and the method returns false.

Remarks

To specify the value to search for, set the column value for each column in the index. The cursor is positioned on the first row that exactly matches the index value.

1.12.13 FindNext(uint16) Method

Gets the next row that exactly matches the index.

Syntax

```
public bool FindNext (ncols)
```

Parameters

ncols For composite indexes, the number of columns to use during the search.

Returns

False if no more rows match the index. In this case, the cursor is positioned after the last row.

1.12.14 FindPrevious(uint16) Method

Gets the previous row that exactly matches the index.

Syntax

```
public bool FindPrevious (ncols)
```

Parameters

ncols For composite indexes, the number of columns to use during the search.

Returns

False if no more rows match the index. In this case, the cursor is positioned before the first row.

1.12.15 First() Method

Moves the cursor to the first row.

Syntax

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

1.12.16 GetBinaryLength Method

Gets the binary length of the value of a column.

Overload list

Modifier and Type	Overload name	Description
public virtual int64	GetBinaryLength(String^) [page 226]	Gets the binary length of the value of a column.

Modifier and Type	Overload name	Description
public virtual int64	GetBinaryLength(uint16) [page 226]	Gets the binary length of the value of a column.

In this section:

GetBinaryLength(String^) Method [page 226]
Gets the binary length of the value of a column.

GetBinaryLength(uint16) Method [page 226]
Gets the binary length of the value of a column.

1.12.16.1 GetBinaryLength(String^) Method

Gets the binary length of the value of a column.

Syntax

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

Parameters

cname The name of the column.

Returns

The length of the column value as a binary.

1.12.16.2 GetBinaryLength(uint16) Method

Gets the binary length of the value of a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The length of the column value as a binary.

1.12.17 GetBool Method

Fetches a value from a column as a boolean.

Overload list

Modifier and Type	Overload name	Description
public virtual bool	GetBool(String^) [page 227]	Fetches a value from a column as a boolean.
public virtual bool	GetBool(uint16) [page 228]	Fetches a value from a column as a boolean.

In this section:

GetBool(String^) Method [page 227]
Fetches a value from a column as a boolean.

GetBool(uint16) Method [page 228]
Fetches a value from a column as a boolean.

1.12.17.1 GetBool(String^) Method

Fetches a value from a column as a boolean.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a boolean.

1.12.17.2 GetBool(uint16) Method

Fetches a value from a column as a boolean.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a boolean.

1.12.18 GetByte Method

Fetches a value from a column as a byte.

Overload list

Modifier and Type	Overload name	Description
public virtual uint8	GetByte(String^) [page 229]	Fetches a value from a column as a byte.
public virtual uint8	GetByte(uint16) [page 230]	Fetches a value from a column as a byte.

In this section:

GetByte(String^) Method [page 229]
Fetches a value from a column as a byte.

GetByte(uint16) Method [page 230]
Fetches a value from a column as a byte.

1.12.18.1 GetByte(String^) Method

Fetches a value from a column as a byte.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a byte.

1.12.18.2 GetByte(uint16) Method

Fetches a value from a column as a byte.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a byte.

1.12.19 GetBytes Method

Gets a binary chunk from the column.

Overload list

Modifier and Type	Overload name	Description
public virtual int64	GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) [page 231]	Gets a binary chunk from the column.
public virtual int64	GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) [page 231]	Gets a binary chunk from the column.

In this section:

GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) Method [page 231]
Gets a binary chunk from the column.

GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) Method [page 231]
Gets a binary chunk from the column.

1.12.19.1 GetBytes(String^, int64, WriteOnlyArray< uint8 >^, int, int) Method

Gets a binary chunk from the column.

Syntax

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

Parameters

cname The name of the column.
dst The buffer to hold the bytes.
count The size of the buffer in bytes.
srcOffset The offset into the value at which to start reading, or the BLOB_CONTINUE constant to continue from where the last read ended.
dstOffset The offset into the destination buffer at which to copy the bytes.

Returns

The number of bytes copied to the destination buffer. If the dst value is NULL, then the number of bytes left is returned. An empty string is returned in the dst parameter when the column is null.

Remarks

The end of the value has been reached if 0 is returned.

Use the IsNull method to differentiate between null and empty strings.

1.12.19.2 GetBytes(uint16, int64, WriteOnlyArray< uint8 >^, int, int) Method

Gets a binary chunk from the column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

dst The buffer to hold the bytes.

count The size of the buffer in bytes.

srcOffset The offset into the value at which to start reading, or the BLOB_CONTINUE constant to continue from where the last read ended.

dstOffset The offset into the destination buffer at which to copy the bytes.

Returns

The number of bytes copied to the destination buffer. If the dst value is NULL, then this method returns the number of bytes left. An empty string is returned in the dst parameter when the column is null.

Remarks

The end of the value has been reached if 0 is returned.

Use the IsNull method to differentiate between null and empty strings.

1.12.20 GetChars Method

Gets a wide string chunk from the column.

Overload list

Modifier and Type	Overload name	Description
public virtual int64	GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) [page 233]	Gets a wide string chunk from the column.
public virtual int64	GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) [page 233]	Gets a wide string chunk from the column.

In this section:

GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) Method [page 233]
Gets a wide string chunk from the column.

GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) Method [page 233]
Gets a wide string chunk from the column.

1.12.20.1 GetChars(String^, int64, WriteOnlyArray< wchar_t >^, int, int) Method

Gets a wide string chunk from the column.

Syntax

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

Parameters

cname The name of the column.

dst The buffer to hold the string chunk. The string is null terminated even if it is truncated.

count The size, in characters, of the buffer.

srcOffset The offset into the value at which to start reading, or the BLOB_CONTINUE constant to continue from where the last read ended.

dstOffset The offset into the destination buffer at which to copy the characters.

Returns

The number of characters copied to the destination buffer excluding the null terminator. If the dst value is NULL, then this method returns the number of characters left in the string. An empty string is returned in the dst parameter when the column is null.

Remarks

The end of the value has been reached if 0 is returned.

Use the IsNull method to differentiate between null and empty strings.

1.12.20.2 GetChars(uint16, int64, WriteOnlyArray< wchar_t >^, int, int) Method

Gets a wide string chunk from the column.

Syntax

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

Parameters

cid The zero-based ordinal column number.
dst The buffer to hold the string chunk. The string is null terminated even if it is truncated.
count The size, in characters, of the buffer.
srcOffset The offset into the value at which to start reading, or the BLOB_CONTINUE constant to continue from where the last read ended.
dstOffset The offset into the destination buffer at which to copy the characters.

Returns

The number of characters copied to the destination buffer excluding the null terminator. If the dst value is NULL, then this method returns the number of characters left in the string. An empty string is returned in the dst parameter when the column is null.

Remarks

The end of the value has been reached if 0 is returned.

Use the IsNull method to differentiate between null and empty strings.

1.12.21 GetDateTime Method

Fetches a value from a column as a DateTime.

Overload list

Modifier and Type	Overload name	Description
public virtual DateTime	GetDateTime(String^) [page 235]	Fetches a value from a column as a DateTime.
public virtual DateTime	GetDateTime(uint16) [page 235]	Fetches a value from a column as a DateTime.

In this section:

GetDateTime(String^) Method [page 235]
Fetches a value from a column as a DateTime.

GetDateTime(uint16) Method [page 235]

Fetches a value from a column as a DateTime.

1.12.21.1 GetDateTime(String^) Method

Fetches a value from a column as a DateTime.

Syntax

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

Parameters

cname The name of the column.

Returns

The DateTime value.

1.12.21.2 GetDateTime(uint16) Method

Fetches a value from a column as a DateTime.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The DateTime value.

1.12.22 GetDouble Method

Fetches a value from a column as a double.

Overload list

Modifier and Type	Overload name	Description
public virtual double	GetDouble(String^) [page 236]	Fetches a value from a column as a double.
public virtual double	GetDouble(uint16) [page 237]	Fetches a value from a column as a double.

In this section:

GetDouble(String^) Method [page 236]
Fetches a value from a column as a double.

GetDouble(uint16) Method [page 237]
Fetches a value from a column as a double.

1.12.22.1 GetDouble(String^) Method

Fetches a value from a column as a double.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a double.

1.12.22.2 GetDouble(uint16) Method

Fetches a value from a column as a double.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a double.

1.12.23 GetFloat Method

Fetches a value from a column as a float.

Overload list

Modifier and Type	Overload name	Description
public virtual float	GetFloat(String^) [page 238]	Fetches a value from a column as a float.
public virtual float	GetFloat(uint16) [page 238]	Fetches a value from a column as a float.

In this section:

GetFloat(String^) Method [page 238]
Fetches a value from a column as a float.

GetFloat(uint16) Method [page 238]
Fetches a value from a column as a float.

1.12.23.1 GetFloat(String^) Method

Fetches a value from a column as a float.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as a float.

1.12.23.2 GetFloat(uint16) Method

Fetches a value from a column as a float.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as a float.

1.12.24 GetGuid Method

Fetches a value from a column as a GUID.

Overload list

Modifier and Type	Overload name	Description
public virtual Guid	GetGuid(String^) [page 239]	Fetches a value from a column as a GUID.
public virtual Guid	GetGuid(uint16) [page 240]	Fetches a value from a column as a GUID.

In this section:

GetGuid(String^) Method [page 239]
Fetches a value from a column as a GUID.

GetGuid(uint16) Method [page 240]
Fetches a value from a column as a GUID.

1.12.24.1 GetGuid(String^) Method

Fetches a value from a column as a GUID.

Syntax

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

Parameters

cname The name of the column.

Returns

The GUID value.

1.12.24.2 GetGuid(uint16) Method

Fetches a value from a column as a GUID.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The GUID value.

1.12.25 GetInt16 Method

Fetches a value from a column as a 16-bit integer.

Overload list

Modifier and Type	Overload name	Description
public virtual int16	GetInt16(String^) [page 241]	Fetches a value from a column as a 16-bit integer.
public virtual int16	GetInt16(uint16) [page 241]	Fetches a value from a column as a 16-bit integer.

In this section:

GetInt16(String^) Method [page 241]
Fetches a value from a column as a 16-bit integer.

GetInt16(uint16) Method [page 241]
Fetches a value from a column as a 16-bit integer.

1.12.25.1 GetInt16(String^) Method

Fetches a value from a column as a 16-bit integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as an integer.

1.12.25.2 GetInt16(uint16) Method

Fetches a value from a column as a 16-bit integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as an integer.

1.12.26 GetInt32 Method

Fetches a value from a column as an integer.

Overload list

Modifier and Type	Overload name	Description
public virtual int	GetInt32(String^) [page 242]	Fetches a value from a column as an integer.
public virtual int	GetInt32(uint16) [page 243]	Fetches a value from a column as an integer.

In this section:

GetInt32(String^) Method [page 242]
Fetches a value from a column as an integer.

GetInt32(uint16) Method [page 243]
Fetches a value from a column as an integer.

1.12.26.1 GetInt32(String^) Method

Fetches a value from a column as an integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as an integer.

1.12.26.2 GetInt32(uint16) Method

Fetches a value from a column as an integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as an integer.

1.12.27 GetInt64 Method

Fetches a value from a column as a 64-bit integer.

Overload list

Modifier and Type	Overload name	Description
public virtual int64	GetInt64(String^) [page 244]	Fetches a value from a column as a 64-bit integer.
public virtual int64	GetInt64(uint16) [page 244]	Fetches a value from a column as a 64-bit integer.

In this section:

GetInt64(String^) Method [page 244]
Fetches a value from a column as a 64-bit integer.

GetInt64(uint16) Method [page 244]
Fetches a value from a column as a 64-bit integer.

1.12.27.1 GetInt64(String^) Method

Fetches a value from a column as a 64-bit integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as an integer.

1.12.27.2 GetInt64(uint16) Method

Fetches a value from a column as a 64-bit integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as an integer.

1.12.28 GetRowCount(unsigned int) Method

Gets the number of rows in the table.

Syntax

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

Parameters

threshold The limit on the number of rows to count. Set to 0 to indicate no limit.

Returns

The number of rows in the table.

Remarks

This method is equivalent to executing the following statement:

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

1.12.29 GetState() Method

Gets the internal state of the cursor.

Syntax

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

Returns

The state of the cursor.

1.12.30 GetString Method

Fetches a value from a column as a string.

Overload list

Modifier and Type	Overload name	Description
public virtual String	GetString(String^) [page 246]	Fetches a value from a column as a string.
public virtual String	GetString(uint16) [page 247]	Fetches a value from a column as a string.

In this section:

GetString(String^) Method [page 246]
Fetches a value from a column as a string.

GetString(uint16) Method [page 247]
Fetches a value from a column as a string.

1.12.30.1 GetString(String^) Method

Fetches a value from a column as a string.

Syntax

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

Parameters

cname The name of the column.

Returns

The string value.

1.12.30.2 GetString(uint16) Method

Fetches a value from a column as a string.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The string value.

1.12.31 GetStringLength Method

Gets the string length of the value of a column.

Overload list

Modifier and Type	Overload name	Description
public virtual int64	GetStringLength(String^) [page 248]	Gets the string length of the value of a column.
public virtual int64	GetStringLength(uint16) [page 248]	Gets the string length of the value of a column.

In this section:

GetStringLength(String^) Method [page 248]
Gets the string length of the value of a column.

GetStringLength(uint16) Method [page 248]
Gets the string length of the value of a column.

1.12.31.1 GetStringLength(String^) Method

Gets the string length of the value of a column.

Syntax

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

Parameters

cname The name of the column.

Returns

The number of characters of a string type column value.

1.12.31.2 GetStringLength(uint16) Method

Gets the string length of the value of a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The number of characters of a string type column value.

1.12.32 GetTableSchema() Method

Returns a TableSchema object that can be used to get schema information about the table.

Syntax

```
public TableSchema GetTableSchema ()
```

Returns

A TableSchema object that can be used to get schema information about the table.

1.12.33 GetUInt16 Method

Fetches a value from a column as a 16-bit unsigned integer.

Overload list

Modifier and Type	Overload name	Description
public virtual uint16	GetUInt16(String^) [page 249]	Fetches a value from a column as a 16-bit unsigned integer.
public virtual uint16	GetUInt16(uint16) [page 250]	Fetches a value from a column as a 16-bit unsigned integer.

In this section:

GetUInt16(String^) Method [page 249]
Fetches a value from a column as a 16-bit unsigned integer.

GetUInt16(uint16) Method [page 250]
Fetches a value from a column as a 16-bit unsigned integer.

1.12.33.1 GetUInt16(String^) Method

Fetches a value from a column as a 16-bit unsigned integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as an unsigned integer.

1.12.33.2 GetUInt16(uint16) Method

Fetches a value from a column as a 16-bit unsigned integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as an unsigned integer.

1.12.34 GetUInt32 Method

Fetches a value from a column as a 32-bit unsigned integer.

Overload list

Modifier and Type	Overload name	Description
public virtual unsigned int	GetUInt32(String^) [page 251]	Fetches a value from a column as a 32-bit unsigned integer.
public virtual unsigned int	GetUInt32(uint16) [page 252]	Fetches a value from a column as a 32-bit unsigned integer.

In this section:

GetUInt32(String^) Method [page 251]
Fetches a value from a column as a 32-bit unsigned integer.

GetUInt32(uint16) Method [page 252]
Fetches a value from a column as a 32-bit unsigned integer.

1.12.34.1 GetUInt32(String^) Method

Fetches a value from a column as a 32-bit unsigned integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as an unsigned integer.

1.12.34.2 GetUInt32(uint16) Method

Fetches a value from a column as a 32-bit unsigned integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as an unsigned integer.

1.12.35 GetUInt64 Method

Fetches a value from a column as a 64-bit unsigned integer.

Overload list

Modifier and Type	Overload name	Description
public virtual uint64	GetUInt64(String^) [page 253]	Fetches a value from a column as a 64-bit unsigned integer.
public virtual uint64	GetUInt64(uint16) [page 253]	Fetches a value from a column as a 64-bit unsigned integer.

In this section:

GetUInt64(String^) Method [page 253]
Fetches a value from a column as a 64-bit unsigned integer.

GetUInt64(uint16) Method [page 253]
Fetches a value from a column as a 64-bit unsigned integer.

1.12.35.1 GetUInt64(String^) Method

Fetches a value from a column as a 64-bit unsigned integer.

Syntax

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

Parameters

cname The name of the column.

Returns

The column value as an unsigned integer.

1.12.35.2 GetUInt64(uint16) Method

Fetches a value from a column as a 64-bit unsigned integer.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column value as an unsigned integer.

1.12.36 Insert() Method

Inserts a new row into the table.

Syntax

```
public void Insert ()
```

1.12.37 InsertBegin() Method

Selects the insert mode for setting columns.

Syntax

```
public void InsertBegin ()
```

Remarks

All columns are set to their default value during an insert unless an alternative value is supplied via Set method calls.

1.12.38 IsNull Method

Checks whether a column value is NULL.

Overload list

Modifier and Type	Overload name	Description
public virtual bool	IsNull(String^) [page 255]	Checks whether a column value is NULL.
public virtual bool	IsNull(uint16) [page 255]	Checks whether a column value is NULL.

In this section:

IsNull(String^) Method [page 255]
Checks whether a column value is NULL.

[IsNull(uint16) Method [page 255]](#)
    Checks whether a column value is NULL.

1.12.38.1 IsNull(String^) Method

Checks whether a column value is NULL.

Syntax

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

Parameters

cname The name of the column.

Returns

True if the column value is NULL.

1.12.38.2 IsNull(uint16) Method

Checks whether a column value is NULL.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

True if the column value is NULL.

1.12.39 Last() Method

Moves the cursor to the last row.

Syntax

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

1.12.40 Lookup(uint16) Method

Performs a lookup based on the current index by scanning forward through the table.

Syntax

```
public bool Lookup (ncols)
```

Parameters

ncols For composite indexes, the number of columns to use in the lookup.

Returns

False if the resulting cursor position is set after the last row.

Remarks

To specify the value to search for, set the column value for each column in the index. The cursor is positioned on the last row that matches or is less than the index value. For composite indexes, the ncols parameter specifies the number of columns to use in the lookup.

1.12.41 LookupBackward(uint16) Method

Performs a lookup based on the current index by scanning backward through the table.

Syntax

```
public bool LookupBackward (ncols)
```

Parameters

ncols For composite indexes, the number of columns to use in the lookup.

Returns

False if the resulting cursor position is set before the first row.

Remarks

To specify the value to search for, set the column value for each column in the index. The cursor is positioned on the last row that matches or is less than the index value. For composite indexes, the ncols parameter specifies the number of columns to use in the lookup.

1.12.42 LookupBegin() Method

Prepares to perform a new lookup on a table.

Syntax

```
public void LookupBegin ()
```

Remarks

You can only set columns in the index that the table was opened with. If the table was opened without an index, then this method cannot be called.

1.12.43 LookupForward(uint16) Method

Performs a lookup based on the current index by scanning forward through the table.

Syntax

```
public bool LookupForward (ncols)
```

Parameters

ncols For composite indexes, the number of columns to use in the lookup.

Returns

False if the resulting cursor position is set after the last row.

Remarks

To specify the value to search for, set the column value for each column in the index. The cursor is positioned on the last row that matches or is less than the index value. For composite indexes, the ncols parameter specifies the number of columns to use in the lookup.

1.12.44 Next() Method

Moves the cursor forward one row.

Syntax

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

Returns

True if the cursor successfully moves forward. An error can still be signaled when the cursor moves successfully to the next row. For example, there could be conversion errors while evaluating the SELECT expressions. In this case, errors are also returned when retrieving the column values. False is returned if the cursor fails to move forward. For example, there is not a next row. In this case, the resulting cursor position is set after the last row.

1.12.45 Previous() Method

Moves the cursor back one row.

Syntax

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

Returns

True if the cursor successfully moves back one row. False if it fails to move backward. The resulting cursor position is set before the first row.

1.12.46 Relative(int) Method

Moves the cursor by offset rows from the current cursor position.

Syntax

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

Parameters

offset The number of rows to move.

1.12.47 SetBool Method

Sets a column to a boolean value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetBool(String^, bool) [page 260]	Sets a column to a boolean value.
public virtual void	SetBool(uint16, bool) [page 260]	Sets a column to a boolean value.

In this section:

SetBool(String^, bool) Method [page 260]
Sets a column to a boolean value.

SetBool(uint16, bool) Method [page 260]
Sets a column to a boolean value.

1.12.47.1 SetBool(String^, bool) Method

Sets a column to a boolean value.

Syntax

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

Parameters

cname The name of the column.
value The boolean value.

1.12.47.2 SetBool(uint16, bool) Method

Sets a column to a boolean value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The boolean value.

1.12.48 SetByte Method

Sets a column to a byte value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetByte(String^, uint8) [page 261]	Sets a column to a byte value.

Modifier and Type	Overload name	Description
public virtual void	SetByte(uint16, uint8) [page 261]	Sets a column to a byte value.

In this section:

SetByte(String^, uint8) Method [page 261]
    Sets a column to a byte value.

SetByte(uint16, uint8) Method [page 261]
    Sets a column to a byte value.

1.12.48.1 SetByte(String^, uint8) Method

Sets a column to a byte value.

Syntax

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

Parameters

cname The name of the column.

value The byte value.

1.12.48.2 SetByte(uint16, uint8) Method

Sets a column to a byte value.

Syntax

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

Parameters

cid The zero-based ordinal column number.

value The byte value.

1.12.49 SetBytes Method

Sets a column to a binary value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetBytes(String^, const Array< uint8 >^, int) [page 262]	Sets a column to a binary value.
public virtual void	SetBytes(uint16, const Array< uint8 >^, int) [page 263]	Sets a column to a binary value.

In this section:

SetBytes(String^, const Array< uint8 >^, int) Method [page 262]
  Sets a column to a binary value.

SetBytes(uint16, const Array< uint8 >^, int) Method [page 263]
  Sets a column to a binary value.

1.12.49.1 SetBytes(String^, const Array< uint8 >^, int) Method

Sets a column to a binary value.

Syntax

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

Parameters

cname The name of the column.
value The binary value. Passing NULL is equivalent to calling the SetNull method.
length The length in bytes to get from the byte array value.

1.12.49.2 SetBytes(uint16, const Array< uint8 >^, int) Method

Sets a column to a binary value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The binary value. Passing NULL is equivalent to calling the SetNull method.
length The length in bytes to get from the byte array value.

1.12.50 SetDateTime Method

Sets a column to a DateTime value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetDateTime(String^, DateTime) [page 264]	Sets a column to a DateTime value.
public virtual void	SetDateTime(uint16, DateTime) [page 264]	Sets a column to a DateTime value.

In this section:

SetDateTime(String^, DateTime) Method [page 264]
Sets a column to a DateTime value.

SetDateTime(uint16, DateTime) Method [page 264]
Sets a column to a DateTime value.

1.12.50.1 SetDateTime(String^, DateTime) Method

Sets a column to a DateTime value.

Syntax

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

Parameters

cname The name of the column.
value The DateTime value.

1.12.50.2 SetDateTime(uint16, DateTime) Method

Sets a column to a DateTime value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The DateTime value.

1.12.51 SetDefault Method

Sets a column to its default value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetDefault(String^) [page 265]	Sets a column to its default value.

Modifier and Type	Overload name	Description
public virtual void	SetDefault(uint16) [page 265]	Sets a column to its default value.

In this section:

SetDefault(String^) Method [page 265]
Sets a column to its default value.

SetDefault(uint16) Method [page 265]
Sets a column to its default value.

1.12.51.1 SetDefault(String^) Method

Sets a column to its default value.

Syntax

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

Parameters

cname The name of the column.

1.12.51.2 SetDefault(uint16) Method

Sets a column to its default value.

Syntax

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

Parameters

cid The zero-based ordinal column number.

1.12.52 SetDouble Method

Sets a column to a double value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetDouble(String^, double) [page 266]	Sets a column to a double value.
public virtual void	SetDouble(uint16, double) [page 266]	Sets a column to a double value.

In this section:

SetDouble(String^, double) Method [page 266]
    Sets a column to a double value.

SetDouble(uint16, double) Method [page 266]
    Sets a column to a double value.

1.12.52.1 SetDouble(String^, double) Method

Sets a column to a double value.

Syntax

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

Parameters

cname The name of the column.

value The double value.

1.12.52.2 SetDouble(uint16, double) Method

Sets a column to a double value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The double value.

1.12.53 SetFloat Method

Sets a column to a float value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetFloat(String^, float) [page 267]	Sets a column to a float value.
public virtual void	SetFloat(uint16, float) [page 268]	Sets a column to a float value.

In this section:

SetFloat(String^, float) Method [page 267]
Sets a column to a float value.

SetFloat(uint16, float) Method [page 268]
Sets a column to a float value.

1.12.53.1 SetFloat(String^, float) Method

Sets a column to a float value.

Syntax

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

Parameters

cname The name of the column.
value The float value.

1.12.53.2 SetFloat(uint16, float) Method

Sets a column to a float value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The float value.

1.12.54 SetGuid Method

Sets a column to a GUID value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetGuid(String^, Guid) [page 268]	Sets a column to a GUID value.
public virtual void	SetGuid(uint16, Guid) [page 269]	Sets a column to a GUID value.

In this section:

SetGuid(String^, Guid) Method [page 268]
Sets a column to a GUID value.

SetGuid(uint16, Guid) Method [page 269]
Sets a column to a GUID value.

1.12.54.1 SetGuid(String^, Guid) Method

Sets a column to a GUID value.

Syntax

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

Parameters

cname The name of the column.
value The GUID value.

1.12.54.2 SetGuid(uint16, Guid) Method

Sets a column to a GUID value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The GUID value.

1.12.55 SetInt16 Method

Sets a column to an integer value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetInt16(String^, int16) [page 270]	Sets a column to an integer value.
public virtual void	SetInt16(uint16, int16) [page 270]	Sets a column to an integer value.

In this section:

SetInt16(String^, int16) Method [page 270]
Sets a column to an integer value.

SetInt16(uint16, int16) Method [page 270]
Sets a column to an integer value.

1.12.55.1 SetInt16(String^, int16) Method

Sets a column to an integer value.

Syntax

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

Parameters

cname The name of the column.

value The signed integer value.

1.12.55.2 SetInt16(uint16, int16) Method

Sets a column to an integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.

value The signed integer value.

1.12.56 SetInt32 Method

Sets a column to an integer value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetInt32(String^, int) [page 271]	Sets a column to an integer value.

Modifier and Type	Overload name	Description
public virtual void	SetInt32(uint16, int) [page 271]	Sets a column to an integer value.

In this section:

SetInt32(String^, int) Method [page 271]
    Sets a column to an integer value.

SetInt32(uint16, int) Method [page 271]
    Sets a column to an integer value.

1.12.56.1 SetInt32(String^, int) Method

Sets a column to an integer value.

Syntax

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

Parameters

cname The name of the column.

value The signed integer value.

1.12.56.2 SetInt32(uint16, int) Method

Sets a column to an integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.

value The signed integer value.

1.12.57 SetInt64 Method

Sets a column to an integer value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetInt64(String^, int64) [page 272]	Sets a column to an integer value.
public virtual void	SetInt64(uint16, int64) [page 272]	Sets a column to an integer value.

In this section:

SetInt64(String^, int64) Method [page 272]
Sets a column to an integer value.

SetInt64(uint16, int64) Method [page 272]
Sets a column to an integer value.

1.12.57.1 SetInt64(String^, int64) Method

Sets a column to an integer value.

Syntax

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

Parameters

cname The name of the column.
value The signed integer value.

1.12.57.2 SetInt64(uint16, int64) Method

Sets a column to an integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The signed integer value.

1.12.58 SetNull Method

Sets a column to null.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetNull(String^) [page 273]	Sets a column to null.
public virtual void	SetNull(uint16) [page 274]	Sets a column to null.

In this section:

SetNull(String^) Method [page 273]
Sets a column to null.

SetNull(uint16) Method [page 274]
Sets a column to null.

1.12.58.1 SetNull(String^) Method

Sets a column to null.

Syntax

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

Parameters

cname The name of the column.

1.12.58.2 SetNull(uint16) Method

Sets a column to null.

Syntax

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

Parameters

cid The zero-based ordinal column number.

1.12.59 SetString Method

Sets a column to a string value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetString(String^, String^) [page 274]	Sets a column to a string value.
public virtual void	SetString(uint16, String^) [page 275]	Sets a column to a string value.

In this section:

SetString(String^, String^) Method [page 274]
Sets a column to a string value.

SetString(uint16, String^) Method [page 275]
Sets a column to a string value.

1.12.59.1 SetString(String^, String^) Method

Sets a column to a string value.

Syntax

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

Parameters

cname The name of the column.
value The string value.

1.12.59.2 SetString(uint16, String^) Method

Sets a column to a string value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The string value.

1.12.60 SetUInt16 Method

Sets a column to an unsigned 16-bit integer value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetUInt16(String^, uint16) [page 276]	Sets a column to an unsigned 16-bit integer value.
public virtual void	SetUInt16(uint16, uint16) [page 276]	Sets a column to an unsigned 16-bit integer value.

In this section:

SetUInt16(String^, uint16) Method [page 276]
Sets a column to an unsigned 16-bit integer value.

SetUInt16(uint16, uint16) Method [page 276]
Sets a column to an unsigned 16-bit integer value.

1.12.60.1 SetUInt16(String^, uint16) Method

Sets a column to an unsigned 16-bit integer value.

Syntax

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

Parameters

cname The name of the column.
value The unsigned integer value.

1.12.60.2 SetUInt16(uint16, uint16) Method

Sets a column to an unsigned 16-bit integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The unsigned integer value.

1.12.61 SetUInt32 Method

Sets a column to an unsigned 32-bit integer value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetUInt32(String^, unsigned int) [page 277]	Sets a column to an unsigned 32-bit integer value.
public virtual void	SetUInt32(uint16, unsigned int) [page 278]	Sets a column to an unsigned 32-bit integer value.

In this section:

SetUInt32(String^, unsigned int) Method [page 277]
Sets a column to an unsigned 32-bit integer value.

SetUInt32(uint16, unsigned int) Method [page 278]
Sets a column to an unsigned 32-bit integer value.

1.12.61.1 SetUInt32(String^, unsigned int) Method

Sets a column to an unsigned 32-bit integer value.

Syntax

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

Parameters

cname The name of the column.
value The unsigned integer value.

1.12.61.2 SetUInt32(uint16, unsigned int) Method

Sets a column to an unsigned 32-bit integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.

value The unsigned integer value.

1.12.62 SetUInt64 Method

Sets a column to an unsigned 64-bit integer value.

Overload list

Modifier and Type	Overload name	Description
public virtual void	SetUInt64(String^, uint64) [page 279]	Sets a column to an unsigned 64-bit integer value.
public virtual void	SetUInt64(uint16, uint64) [page 279]	Sets a column to an unsigned 64-bit integer value.

In this section:

SetUInt64(String^, uint64) Method [page 279]
Sets a column to an unsigned 64-bit integer value.

SetUInt64(uint16, uint64) Method [page 279]
Sets a column to an unsigned 64-bit integer value.

1.12.62.1 SetUInt64(String^, uint64) Method

Sets a column to an unsigned 64-bit integer value.

Syntax

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

Parameters

cname The name of the column.
value The unsigned integer value.

1.12.62.2 SetUInt64(uint16, uint64) Method

Sets a column to an unsigned 64-bit integer value.

Syntax

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

Parameters

cid The zero-based ordinal column number.
value The unsigned integer value.

1.12.63 TruncateTable() Method

Truncates the table and temporarily activates STOP SYNCHRONIZATION DELETE.

Syntax

```
public void TruncateTable ()
```

1.12.64 Update() Method

Updates the current row.

Syntax

```
public virtual void Update ()
```

1.12.65 UpdateBegin() Method

Selects the update mode for setting columns.

Syntax

```
public virtual void UpdateBegin ()
```

Remarks

Columns in the primary key cannot be modified when UltraLite is in update mode. Use Connection.Commit() or Rollback() to commit or rollback a transaction.

1.13 TableSchema Class

Represents the schema of an UltraLite table. DatabaseSchema.GetTables() or GetTableSchema() or Table.GetTableSchema() return a TableSchema.

Namespace

```
UltraLite
```

Syntax

```
public ref class  sealed : CursorSchema
```

Members

All members of TableSchema, including inherited members.

Methods

Modifier and Type	Method	Description
public void	Close() [page 283]	Destroys this object.
public virtual uint16	GetColumnCount() [page 283]	Gets the number of columns in the result set or table.
public String	GetColumnDefault(uint16) [page 284]	Gets the default value for the column if it exists.
public uint16	GetColumnDefaultType(uint16) [page 284]	Gets the type of column default.
public virtual uint16	GetColumnID(String^) [page 285]	Gets the zero-based column ID from its name.
public virtual String	GetColumnName(uint16, uint16type) [page 285]	Gets the name of a column given its zero-based ID.
public virtual uint16	GetColumnPrecision(uint16) [page 286]	Gets the precision of a numeric column.
public virtual uint16	GetColumnScale(uint16) [page 286]	Gets the scale of a numeric column.
public virtual int	GetColumnSize(uint16) [page 287]	Gets the size of the column.
public virtual uint16	GetColumnSQLType(uint16) [page 287]	Gets the SQL type of a column.
public virtual uint16	GetColumnType(uint16) [page 288]	Gets the storage/host variable type of a column.
public uint64	GetGlobalAutoincPartitionSize(uint16) [page 288]	Gets the partition size.
public uint16	GetIndexCount() [page 289]	Gets the number of indexes in the table.
public IIterator< IndexSchema^>	GetIndexes() [page 289]	Gets an iterator over a collection of all indexes (schema) in the table.
public IndexSchema	GetIndexSchema(String^) [page 290]	Gets the schema of an index when given the name.
public String	GetName() [page 290]	Gets the name of the table.
public String	GetOptimalIndex(uint16) [page 290]	Determines the best index to use for searching for a column value.
public IndexSchema	GetPrimaryKey() [page 291]	Gets the primary key for the table.
public String	GetPublicationPredicate(String^) [page 291]	Gets the publication predicate as a string.
public uint16	GetTableSyncType() [page 292]	Gets the table synchronization type.
public bool	InPublication(String^) [page 292]	Checks whether the table is contained in the named publication.
public virtual bool	IsAliased(uint16) [page 293]	Indicates whether the column in a result set was given an alias.

Modifier and Type	Method	Description
public bool	IsColumnInIndex(uint16, String^) [page 293]	Checks whether the column is contained in the named index.
public bool	IsColumnNullable(uint16) [page 294]	Checks whether the specified column is nullable.

In this section:

GetColumnCount() Method [page 283]
: Gets the number of columns in the result set or table.

GetPrimaryKey() Method [page 291]
    Gets the primary key for the table.

GetPublicationPredicate(String^) Method [page 291]
    Gets the publication predicate as a string.

GetTableSyncType() Method [page 292]
    Gets the table synchronization type.

InPublication(String^) Method [page 292]
    Checks whether the table is contained in the named publication.

IsAliased(uint16) Method [page 293]
    Indicates whether the column in a result set was given an alias.

IsColumnInIndex(uint16, String^) Method [page 293]
    Checks whether the column is contained in the named index.

IsColumnNullable(uint16) Method [page 294]
    Checks whether the specified column is nullable.

1.13.1 CloseObject() Method

Destroys this object.

Syntax

```
public void CloseObject ()
```

1.13.2 GetColumnCount() Method

Gets the number of columns in the result set or table.

Syntax

```
public virtual uint16 GetColumnCount ()
```

Returns

The number of columns in the result set or table.

1.13.3 GetColumnDefault(uint16) Method

Gets the default value for the column if it exists.

Syntax

```
public String GetColumnDefault (cid)
```

Parameters

cid A zero-based ordinal column number.

Returns

The default value. An empty string is returned if the column has no default value.

1.13.4 GetColumnDefaultType(uint16) Method

Gets the type of column default.

Syntax

```
public uint16 GetColumnDefaultType (cid)
```

Parameters

cid A zero-based ordinal column number.

Returns

The type of column default.

1.13.5 GetColumnID(String^) Method

Gets the zero-based column ID from its name.

Syntax

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

Parameters

columnName The column name.

Returns

The column ID.

1.13.6 GetColumnName(uint16, uint16type) Method

Gets the name of a column given its zero-based ID.

Syntax

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

Parameters

cid The zero-based ordinal column number.
type The desired column name type.

Returns

A string containing the column name, if found; otherwise, an error is thrown.

Remarks

Depending on the type selected and how the column was declared in the SELECT statement, the column name may be returned in the form [table-name].[column-name].

The type parameter specifies what type of column name to return.

1.13.7 GetColumnPrecision(uint16) Method

Gets the precision of a numeric column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The precision of the numeric column.

1.13.8 GetColumnScale(uint16) Method

Gets the scale of a numeric column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The scale of the numeric column.

1.13.9 GetColumnSize(uint16) Method

Gets the size of the column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

The column size.

1.13.10 GetColumnSQLType(uint16) Method

Gets the SQL type of a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

A value from the ColumnSQLType enumeration or SQLTYPE_BAD_INDEX if the column does not exist.

1.13.11 GetColumnType(uint16) Method

Gets the storage/host variable type of a column.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

TYPE_BAD_INDEX if the column does not exist.

1.13.12 GetGlobalAutoincPartitionSize(uint16) Method

Gets the partition size.

Syntax

```
public uint64 GetGlobalAutoincPartitionSize (cid)
```

Parameters

cid A zero-based ordinal column number.

Returns

The partition size for the column. All global autoincrement columns in a given table share the same global autoincrement partition.

1.13.13 GetIndexCount() Method

Gets the number of indexes in the table.

Syntax

```
public uint16 GetIndexCount ()
```

Returns

The number of indexes in the table.

Remarks

Index IDs and counts may change during a schema upgrade. To correctly identify an index, access it by name or refresh any cached IDs and counts after a schema upgrade.

1.13.14 GetIndexes() Method

Gets an iterator over a collection of all indexes (schema) in the table.

Syntax

```
public IIterator< IndexSchema^> GetIndexes ()
```

Returns

An iterator over a collection of IndexSchema objects.

1.13.15 GetIndexSchema(String^) Method

Gets the schema of an index when given the name.

Syntax

```
public IndexSchema GetIndexSchema (indexName)
```

Parameters

indexName The name of the index.

Returns

An IndexSchema object for the specified index, or NULL if the object does not exist.

1.13.16 GetName() Method

Gets the name of the table.

Syntax

```
public String GetName ()
```

Returns

The name of the table.

1.13.17 GetOptimalIndex(uint16) Method

Determines the best index to use for searching for a column value.

Syntax

```
public String GetOptimalIndex (cid)
```

Parameters

cid A zero-based ordinal column number.

Returns

The name of the index or NULL if the column isn't indexed.

1.13.18 GetPrimaryKey() Method

Gets the primary key for the table.

Syntax

```
public IndexSchema GetPrimaryKey ()
```

Returns

An IndexSchema object for the table's primary key.

1.13.19 GetPublicationPredicate(String^) Method

Gets the publication predicate as a string.

Syntax

```
public String GetPublicationPredicate (pubName)
```

Parameters

pubName The name of the publication.

Returns

The publication predicate string for the specified publication.

1.13.20 GetTableSyncType() Method

Gets the table synchronization type.

Syntax

```
public uint16 GetTableSyncType ()
```

Returns

The table synchronization type.

Remarks

This method indicates how the table participates in synchronization and is defined when the table is created with the SYNCHRONIZE constraint clause of the CREATE TABLE statement.

1.13.21 InPublication(String^) Method

Checks whether the table is contained in the named publication.

Syntax

```
public bool InPublication (pubName)
```

Parameters

pubName The name of the publication.

Returns

True if the table is contained in the publication; otherwise, returns false.

1.13.22 IsAliased(uint16) Method

Indicates whether the column in a result set was given an alias.

Syntax

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

Parameters

cid The zero-based ordinal column number.

Returns

True if the column is aliased; otherwise, returns false.

1.13.23 IsColumnInIndex(uint16, String^) Method

Checks whether the column is contained in the named index.

Syntax

```
public bool IsColumnInIndex (cid, indexName)
```

Parameters

cid A zero-based ordinal column number.
indexName The name of the index.

Returns

True if the column is contained in the index; otherwise, returns false.

1.13.24 IsColumnNullable(uint16) Method

Checks whether the specified column is nullable.

Syntax

```
public bool IsColumnNullable (cid)
```

Parameters

cid A zero-based ordinal column number.

Returns

True if the column is nullable; otherwise, returns false.

1.14 FileTransferObserver(FileTransferStatus^) Delegate

The definition for a delegate that is called at various stages of the file transfer.

Syntax

```
public delegate bool FileTransferObserver (status);
```

1.15 SyncObserver(SyncStatus^) Delegate

The definition for a delegate that is called at various stages of the synchronization.

Syntax

```
public delegate bool SyncObserver (status);
```

1.16 ValidateCallback(ValidateData^) Delegate

The definition for a delegate that is called for validation feedback.

Syntax

```
public delegate bool ValidateCallback (vdata);
```

1.17 AuthStatusCode Enumeration

Specifies MobiLink authentication status codes. SyncResult and FileTransferResult have AuthStatus fields that reuturn an AuthStatusCode value.

Syntax

```
enum AuthStatusCode
```

Members

Member name	Description	Value
UNKNOWN	Authorization status is unknown, possibly because the connection has not yet synchronized.	0
VALID	User ID and password were valid at the time of synchronization.	1
VALID_BUT_EXPIRES_SOON	User ID and password were valid at the time of synchronization but will expire soon.	2
EXPIRED	Authorization failed: user ID or password have expired.	3
INVALID	Authorization failed: bad user ID or password.	4
IN_USE	Authorization failed: user ID already in use.	5

1.18 ColumnDefaultType Enumeration

Identifies a column default type.

Syntax

```
enum ColumnDefaultType
```

Members

Member name	Description
column_default_none	The column has no default value.
column_default_autoincrement	The column default is AUTOINCREMENT.
column_default_global_autoincrement	The column default is GLOBAL AUTOINCREMENT.
column_default_current_timestamp	The column default is CURRENT TIMESTAMP.
column_default_current_utc_timestamp	The column default is CURRENT UTC TIMESTAMP.
column_default_current_time	The column default is CURRENT TIME.
column_default_current_date	The column default is CURRENT DATE.
column_default_newid	The column default is NEWID().
column_default_other	The column default is a user-specified constant.

Returns

TableSchema::GetColumnDefaultType

1.19 ColumnNameType Enumeration

Specifies values that control how a column name is retrieved when describing a result set.

Syntax

```
enum ColumnNameType
```

Members

Member name	Description
name_type_sql	For SELECT statements, returns the alias or correlation name. For tables, returns the column name.
name_type_sql_column_only	For SELECT statements, returns the alias or correlation name and excludes any table names that were specified. For tables, returns the column name.
name_type_base_table	Returns the underlying table name if it can be determined. If the table does not exist in the database schema, returns an empty string.
name_type_base_column	Returns the underlying column name if it can be determined. If the column does not exist in the database schema, returns an empty string.
name_type_qualified	Returns the underlying qualified column name, if it can be determined, when used in conjunction with the ULResultSet-Schema.GetColumnName method. The returned name can be one of the following values, and is determined in this order: 1. The represented correlated table 2. The name of the represented table column 3. The alias name of the column 4. An empty string
name_type_base	Indicates that a column name qualified with its table name should be returned when used with the GetColumnName method. If the column name being retrieved is associated with a base table in the query, then the base table name is used as the column qualifier (that is, the base_table_name.col-umn_name value is returned). If the column name being re-trieved refers to a column in a correlated table in the query, then the correlation name is used as the column qualifier (that is, the correl_table_name.col_name value is returned). If the column has an alias, then the qualified name of the col-umn being aliased is returned; the alias is not part of the qualified name. Otherwise, an empty string is returned.

Remarks

ResultSetSchema::GetColumnName

1.20 ColumnSQLType Enumeration

Specifies the SQL types for a column. CursorSchema.GetColumnSQLType() returns a ColumnSQLType.

Syntax

```
enum ColumnSQLType
```

Members

Member name	Description
SQLTYPE_BAD_INDEX	Represents that the column at the specified index does not exist.
SQLTYPE_S_LONG	Represents that the column contains a signed long.
SQLTYPE_U_LONG	Represents that the column contains an unsigned long.
SQLTYPE_S_SHORT	Represents that the column contains a signed short.
SQLTYPE_U_SHORT	Represents that the column contains an unsigned short.
SQLTYPE_S_BIG	Represents that the column contains a signed 64-bit integer.
SQLTYPE_U_BIG	Represents that the column contains an unsigned 64-bit integer.
SQLTYPE_TINY	Represents that the column contains an unsigned 8-bit integer.
SQLTYPE_BIT	Represents that the column contains a 1-bit flag.
SQLTYPE_TIMESTAMP	Represents that the column contains timestamp information.
SQLTYPE_DATE	Represents that the column contains date information.
SQLTYPE_TIME	Represents that the column contains time information.
SQLTYPE_DOUBLE	Represents that the column contains a double precision floating-point number (8 bytes).
SQLTYPE_REAL	Represents that the column contains a single precision floating-point number (4 bytes).
SQLTYPE_NUMERIC	Represents that the column contains exact numerical data, with specified precision and scale.
SQLTYPE_BINARY	Represents that the column contains binary data with a specified maximum length.
SQLTYPE_CHAR	Represents that the column contains character data with a specified length.
SQLTYPE_LONGVARCHAR	Represents that the column contains character data with variable length.

Member name	Description
SQLTYPE_LONGBINARY	Represents that the column contains binary data with variable length.
SQLTYPE_UUID	Represents that the column contains a UUID.
SQLTYPE_ST_GEOMETRY	Represents that the column contains spatial data in the form of points.
SQLTYPE_TIMESTAMP_WITH_TIME_ZONE	Represents that the column contains timestamp and time zone information.

Remarks

These values correspond to SQL column types.

1.21 ColumnStorageType Enumeration

Specifies the host variable types for a column.

Syntax

```
enum ColumnStorageType
```

Members

Member name	Description
TYPE_BAD_INDEX	Represents an invalid value.
TYPE_S_LONG	Represents a ul_s_long (32-bit signed int).
TYPE_U_LONG	Represents a ul_u_long (32-bit unsigned int).
TYPE_S_SHORT	Represents a ul_s_short (16-bit signed int).
TYPE_U_SHORT	Represents a ul_u_short (16-bit unsigned int).
TYPE_S_BIG	Represents a ul_s_big (64-bit signed int).
TYPE_U_BIG	Represents a ul_u_big (64-bit unsigned int).
TYPE_TINY	Represents a ul_byte (8-bit unsigned).
TYPE_BIT	Represents a ul_byte (8-bit unsigned, 1-bit used).
TYPE_DOUBLE	Represents a ul_double (double).

Member name	Description
TYPE_REAL	Represents a ul_real (float).
TYPE_BINARY	Represents a ul_binary (2 byte length followed by byte array).
TYPE_TIMESTAMP_STRUCT	Represents a DECL_DATETIME.
TYPE_TCHAR	Represents a character array (string buffer).
TYPE_CHAR	Represents a char array (string buffer).
TYPE_WCHAR	Represents a ul_wchar (UTF16) array.
TYPE_GUID	Represents a GUID structure.

Remarks

These values identify the host variable type required for a column and indicate how UltraLite should fetch values. CursorSchema.GetColumnType() returns a ColumnStorageType.

1.22 CursorState Enumeration

Specifies a possible result set or cursor states. Cursor.GetState() returns a CursorState.

Syntax

```
enum CursorState
```

Members

Member name	Description
RS_STATE_ERROR	An error.
RS_STATE_UNPREPARED	Not prepared.
RS_STATE_ON_ROW	On a valid row.
RS_STATE_BEFORE_FIRST	Before the first row.
RS_STATE_AFTER_LAST	After the last row.
RS_STATE_COMPLETED	Closed.

1.23 ErrorCodes Enumeration

Specifies the Platform::COMException HResult code that indicates an UltraLite exception.

Syntax

```
enum ErrorCodes
```

Members

Member name	Description	Value
E_ULTRALITE_ERROR	HResult code indicating an UltraLite error.	(-10000)

Remarks

Use a GetLastError method to get the UltraLite-specific error information.

DatabaseManager::GetLastError Connection::GetLastError

1.24 IndexFlag Enumeration

Specifies the flags (bit fields) that identify properties of an index.

Syntax

```
enum IndexFlag
```

Members

Member name	Description	Value
index_flag_primary_key	Represents that the index is a primary key.	0x0001

Member name	Description	Value
index_flag_unique_key	Represents that The index is a primary key or an index created for a unique constraint (nulls not allowed).	0x0002
index_flag_unique_index	Represents that the index was created with the UNIQUE flag (or is a primary key).	0x0004
index_flag_foreign_key	Represents that the index is a foreign key.	0x0010
index_flag_foreign_key_nullable	Represents that the foreign key allows nulls.	0x0020
index_flag_foreign_key_check_on_commit	Represents that referential integrity checks are performed on commit (rather than on insert/update).	0x0040

Remarks

IndexSchema::GetIndexFlags

1.25 StreamType Enumeration

Specifies the possible values for the FileTransfer.Stream property.

Syntax

```
enum StreamType
```

Members

Member name	Description
TCPIP	Represents a TCP/IP stream type.
HTTP	Represents an HTTP stream type.
HTTPS	Represents an HTTPS stream type.
TLS	Represents a TLS stream type.

1.26 SyncState Enumeration

Indicates the current stage of synchronization.

Syntax

```
enum SyncState
```

Members

Member name	Description
STATE_STARTING	The synchronization is starting; initial parameter validation is complete and the synchronization result will be saved.
STATE_CONNECTING	Connecting to the MobiLink server.
STATE_RESUMING_DOWNLOAD	An optional state that is entered when we are attempting to resume a partial download. On success, sync will proceed to the STATE_RECEIVING_TABLE state, and STATE_ERROR if we are unable to resume.
STATE_SENDING_HEADER	The synchronization connection is established and initial data is about to be sent.
STATE_SENDING_CHECK_SYNC_REQUEST	A table is about to be sent. The state of the last upload is unknown, so a request to check its status is being sent.
STATE_WAITING_FOR_CHECK_SYNC_RESPONSE	Waiting for the server to respond to the check sync request.
STATE_PROCESSING_CHECK_SYNC_RESPONSE	The response to the check sync request has been received and is being processed.
STATE_SENDING_TABLE	A table is about to be sent.
STATE_SENDING_DATA	Schema information or row data is being sent.
STATE_FINISHING_UPLOAD	The upload stage is complete and state information is about to be committed.
STATE_WAITING_FOR_UPLOAD_ACK	Waiting for the server to acknowledge receiving our upload.
STATE_PROCESSING_UPLOAD_ACK	The server has acknowledged receiving our upload.
STATE_WAITING_FOR_DOWNLOAD	Waiting for the server to start sending the download.
STATE_RECEIVING_TABLE	A table is about to be received.
STATE_RECEIVING_DATA	Data for the most recently identified table is being received.
STATE_COMMITTING_DOWNLOAD	The download stage is complete and downloaded rows are about to be committed.

Member name	Description
STATE_ROLLING_BACK_DOWNLOAD	An error occurred during download and the download is being rolled back.
STATE_SENDING_DOWNLOAD_ACK	Sending an acknowledgement that the download is complete.
STATE_DISCONNECTING	About to disconnect from the MobiLink server.
STATE_DONE	Synchronization has completed successfully.
STATE_ERROR	Synchronization has completed, but with an error.

Remarks

You should not assume that the synchronization states occur in the order listed. SyncStatus.State contains a SyncState.

1.27 TableSyncType Enumeration

Identifies a table synchronization type.

Syntax

```
enum TableSyncType
```

Members

Member name	Description
table_sync_on	Indicates that all changed rows are synchronized, which is the default behavior. This initializer corresponds to the SYNCHRONIZE ON clause in a CREATE TABLE statement.
table_sync_off	Indicates that the table is never synchronized. This initializer corresponds to the SYNCHRONIZE OFF clause in a CREATE TABLE statement.

Member name	Description
table_sync_upload_all_rows	Indicates that every row is always uploaded, including unchanged rows. This initializer corresponds to the SYNCHRONIZE ALL clause in a CREATE TABLE statement.
table_sync_download_only	Indicates that changes are never uploaded. This initializer corresponds to the SYNCHRONIZE DOWNLOAD clause in a CREATE TABLE statement.

Remarks

TableSchema::GetTableSyncType

1.28 ValidateFlags Enumeration

Represents validation input flags. The ValidateDatabase methods on Connection and DatabaseManager use ValidateFlags as the flags parameter.

Syntax

```
enum ValidateFlags
```

Members

Member name	Description	Value
VF_TABLE	Validate tables. Check that table and index row counts match.	0x0001
VF_INDEX	Validate indexes. Check the integrity of the index.	0x0002
VF_DATABASE	Validate database. Verify all database pages by using page checksums and additional checks.	0x0004

Member name	Description	Value
VF_EXPRESS	Performs a faster, less thorough, validation. This flag modifies other specified flags.	0x8000
VF_FULL_VALIDATE	Performs all types of validation on the database.	(ULVF_TABLE|ULVF_INDEX|ULVF_DATABASE)

1.29 ValidateStatusId Enumeration

Specifies possible status IDs for validate progress callbacks.

Syntax

```
enum ValidateStatusId
```

Members

Member name	Description	Value
VALID_NO_ERROR	No error occurred.	0
VALID_START	Start validation.	1
VALID_END	End validation. Parm1 tracks the resulting sqlcode, which indicates success or failure.	2
VALID_CHECKING_PAGE	Send a periodic status message while checking database pages. Parm1 tracks a number associated with the page. The order is not defined.	10
VALID_CHECKING_TABLE	Checking a table. Parm1 tracks the table name.	20
VALID_DUPLICATE_INDEX	Checking an index. Parm1 stores the table name and parm2 stores the index name.	32

Member name	Description	Value
VALID_DATABASE_ERROR	An error occurred accessing the database. Check the SQLCODE for more information.	100
VALID_STARTUP_ERROR	Error starting the database. (for low-level access)	101
VALID_CORRUPT_PAGE_TABLE	Page table is corrupt.	110
VALID_FAILED_CHECKSUM	Page checksum failed. Parm1 tracks a number associated with the page.	111
VALID_CORRUPT_PAGE	A page is corrupt. Parm1 tracks a number associated with the page.	112
VALID_ROWCOUNT_MISMATCH	The number of rows in the index is different from the table row count. Parm1 tracks the table name, and parm2 tracks index name.	120
VALID_BAD_ROWID	There is an invalid row identifier in the index. Parm1 tracks the table name, and parm2 tracks the index name.	121

1.30 FileTransferResult Structure

Stores the file transfer result so that appropriate action can be taken in the application.

Syntax

```
typedef struct  sealed
```

Members

All members of FileTransferResult, including inherited members.

Variables

Modifier and Type	Variable	Description
public property uint16	AuthCode	Contains the return code of the optional authenticate_file_transfer script on the MobiLink server.
public property int16	AuthStatus	Supplies parameters to authentication parameters in MobiLink events.
public property int64	AuthValue	Reports results of a custom MobiLink user authentication script. The MobiLink server provides this information to the client.
public property uint16	TransferredFile	Returns 1 if the file was successfully transferred, and 0 if an error occurs.
public property int16	StreamErrorCode	Represents the specific stream error. `%SQLANY17%\\SDK\\Include\\sserror.h`
public property int32	StreamErrorSystem	Represents a system-specific error code.

1.31 FileTransferStatus Structure

Stores status/progress information while the file upload/download is in progress.

Syntax

```
typedef struct  sealed
```

Members

All members of FileTransferStatus, including inherited members.

Variables

Modifier and Type	Variable	Description
public property uint64	FileSize	Indicates the total size, in bytes, of the file being downloaded.
public property uint64	BytesReceived	Indicates how much of the file has been downloaded so far, including previous synchronizations, if the download is resumed.

Modifier and Type	Variable	Description
public property uint64	ResumedAtSize	Indicates at what point the current download resumes.
public property int	Flags	Provides additional information about the file transfer status. The MLFT_STATUS_FLAG_IS_BLOCKING value is set when the MLFileDownload method is blocking on a network call and the download status has not changed since the last time the observer method was called.

Remarks

The FileTransferObserver delegate of FileTransfer.UploadFile() and DownloadFile() is passed a FileTransferStatus object. Also FileTransfer.UploadFileAsync() and DownloadFileAsync() supply a FileTransferStatus object in the progress callback.

1.32 SyncResult Structure

Stores the synchronization result so that appropriate action can be taken in the application. Connection.GetSyncResult() returns a SyncResult.

Syntax

```
typedef struct  sealed
```

Members

All members of SyncResult, including inherited members.

Variables

Modifier and Type	Variable	Description
public property ULSqlCode	ErrorCode	Indicates the SQLCODE value.
public property String	ErrorParms	Indicates error parameters.
public property int	SqlCount	Indicates the SQLCOUNT value.

Modifier and Type	Variable	Description
public property int16	StreamErrorCode	Indicates the specific stream error. See the ss_error_code enumeration for possible values.
public property int32	StreamErrorSystem	Indicates a system-specific error code. For more information about error codes, see your platform documentation.
public property String	StreamErrorParms	Indicates a string with additional information, if available, for the stream_error_code value.
public property bool	UploadOK	Returns true if the upload was successful; false otherwise.
public property bool	IgnoredRows	Returns true if uploaded rows were ignored; false otherwise.
public property int16	AuthStatus	Indicates the synchronization authentication status.
public property int64	AuthValue	Indicates the value used by the MobiLink server to determine the auth_status result.
public property bool	PartialDownloadRetained	Indicates the value that tells you that a partial download was retained. See keep_partial_download.
public property DateTime	Timestamp	Indicates the time and date of the last synchronization.
public property int	SentBytes	Indicates the number of bytes currently sent.
public property int	SentInserts	Indicates the number of rows currently inserted.
public property int	SentUpdates	Indicates the number of updated rows currently sent.
public property int	SentDeletes	Indicates the number of deleted rows currently sent.
public property int	RecvBytes	Indicates the number of bytes currently received.
public property int	RecvInserts	Indicates the number of rows currently inserted.
public property int	RecvUpdates	Indicates the number of rows currently received that have been updated.
public property int	IgnoredUpdates	Indicates the number of rows in the current download that have been received and are duplicates of rows that already exist in the table.

Modifier and Type	Variable	Description
public property int	RecvDeletes	Indicates the number of rows currently received that have been deleted.
public property int	IgnoredDeletes	Indicates the number of rows in the current download that have been received and do not exist in the table.
public property int	TruncateDeletes	Indicates the number of rows that have been deleted by a truncate operation.

1.33 SyncStatus Structure

Stores status/progress information while the synchronization is in progress.

Syntax

```
typedef struct   sealed
```

Members

All members of SyncStatus, including inherited members.

Variables

Modifier and Type	Variable	Description
public property short	State	Indicates one of the many supported states of the SyncState enumeration.
public property short	TableCount	Indicates the number of tables in database.
public property short	TableIndex	Indicates an index ranging from 1 to TableCount.
public property short	TableID	Indicates the current table ID that is being uploaded or downloaded (one-based). This number may skip values when not all tables are being synchronized, and is not necessarily increasing.
public property String	TableName	Indicates the table name corresponding to Table ID.
public property int	SentBytes	Indicates the number of bytes currently sent.

Modifier and Type	Variable	Description
public property int	SentInserts	Indicates the number of rows currently inserted.
public property int	SentUpdates	Indicates the number of updated rows currently sent.
public property int	SentDeletes	Indicates the number of deleted rows currently sent.
public property int	RecvBytes	Indicates the number of bytes currently received.
public property int	RecvInserts	Indicates the number of rows currently inserted.
public property int	RecvUpdates	Indicates the number of rows currently received that have been updated.
public property int	IgnoredUpdates	Indicates the number of rows in the current download that have been received and are duplicates of rows that already exist in the table.
public property int	RecvDeletes	Indicates the number of rows currently received that have been deleted.
public property int	IgnoredDeletes	Indicates the number of rows in the current download that have been received and do not exist in the table.
public property int	TruncateDeletes	Indicates the number of rows that have been deleted by a truncate operation.
public property unsigned short	Flags	Indicates the current synchronization flags indicating additional information relating to the current state.
public property int	CurrentDownloadRowCount	Indicates the number of rows that have been downloaded so far. This number includes duplicate rows that aren't included in recvInserts, recvUpdates, or recvDeletes.
public property int	TotalDownloadRowCount	Indicates the total number of rows to be received in the download. This number includes duplicate rows that aren't included in recvInserts, recvUpdates, or recvDeletes. This field is not set until the synchronization enters the STATE_RECEIVING_TABLE state for the first table.

Remarks

The SyncObserver delegate of Connection.Synchronize() is passed a SyncStatus object. Also Connection.SynchronizeAsync() supplies a SyncStatus object in the progress callback.

1.34 ValidateData Structure

Stores validation status information while the validation is in progress.

Syntax

```
typedef struct   sealed
```

Members

All members of ValidateData, including inherited members.

Variables

Modifier and Type	Variable	Description
public property uint8	StatusId	Indicates what is being reported in the validation process.
public property Array< String^>	Parms	Represents parameters for this validation status.

Remarks

The SyncObserver delegate of Connection.Synchronize() is passed a SyncStatus object. Also Connection.SynchronizeAsync() supplies a SyncStatus object in the progress callback.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

THE BEST RUN