MBS SQL Plugin Documentation

Christian Schmitz

May 2, 2026

0.1 Introduction

This is the PDF version of the documentation for the Xojo Plug-in from Monkeybread Software Germany.
Plugin part: MBS SQL Plugin

0.2 Content

Chapter 1

List of Topics

Chapter 2

List of all classes

Chapter 3

List of all modules

Chapter 4

List of all global methods

Chapter 5

SQL

5.1 class Database

5.1.1 class Database

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The Xojo Database class is the base class for the database subclasses that communicate with a variety of databases.
Notes: Use one of the subclasses to connect to your database.

5.1.2 Methods

5.1.3 AddRow(TableName as String, row as DatabaseRow)

Plugin Version: 20.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Inserts Data (a populated DatabaseRow) as a new row in TableName.
See also:

- 5.1.4 AddRow(TableName as String, row as DatabaseRow, idColumnName as string = "") as Integer 27

5.1.4 AddRow(TableName as String, row as DatabaseRow, idColumnName as string = "") as Integer

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Inserts Data (a populated DatabaseRow) as a new row in TableName.

Notes: Returns insert ID if possible or -1 if not found.
Implemented directly in SQLDatabaseMBS for CubeSQL, SQLite, mariaDB and MySQL. Other databases need column name for ID to run a select to get the value.
For PostgreSQL we use RETURNING, so you need to pass name of the ID column.
See also:

- 5.1.3 AddRow(TableName as String, row as DatabaseRow) 27

5.1.5 BeginTransaction

Plugin Version: 20.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new transaction.
Notes: Changes to the database made after this call can be saved with CommitTransaction or undone with RollbackTransaction.

5.1.6 Close

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Closes or disconnects the database.
Notes: Calling Close does not issue a Commit, but some databases will automatically Commit changes in a transaction when you Close the connection and some database will automatically Rollback changes in a transaction when the connection is closed. Refer to the documentation for your database to check what its behavior is.

For desktop applications, you will often Connect to the database when the app starts and Close it when the app quits.

For web applications, you usually Connect to the database when the Session starts and Close it when the Session quits.

5.1.7 Commit

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Commits an open transaction. This permanently saves changes to the database.
Notes: You have to have an open transaction to be able to use Commit. On SQLite (and other databases), you can start a transaction with this command:

BEGIN TRANSACTION

It can be sent using SQLExecute:

DB.SQLExecute("BEGIN TRANSACTION")

5.1.8 CommitTransaction

Plugin Version: 20.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Commits an open transaction. This permanently saves changes to the database.

5.1.9 Connect

Plugin Version: 20.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Connects to the database so that you can begin using it.
See also:

- 5.1.10 Connect as boolean 29

5.1.10 Connect as boolean

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Connects to the database so that you can begin using it.
Notes: Before proceeding with database operations, test to be sure that Connect returns True.
If Connect returns False, you should also check the ErrorCode and ErrorMessage.

See Option() for various options you can set before connecting.
e.g. c.Option("SQLiteVFSFlags") = "1" for SQLite for read only access.
See also:

- 5.1.9 Connect 29

5.1.11 ExecuteSQL(sql As String, ParamArray values As Variant)

Plugin Version: 20.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Used to execute an SQL command.
Notes: Use this for commands that do not return any data, such as CREATE TABLE or INSERT.
See also:

- 5.1.12 ExecuteSQL(sql As String, values() As Variant) 30

5.1.12 ExecuteSQL(sql As String, values() As Variant)

Plugin Version: 20.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Used to execute an SQL command.
Notes: Use this for commands that do not return any data, such as CREATE TABLE or INSERT.
See also:

- 5.1.11 ExecuteSQL(sql As String, ParamArray values As Variant) 29

5.1.13 InsertRecord(TableName as String, Data as DatabaseRecord)

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Inserts Data (a populated DatabaseRecord) as a new row in TableName.
Notes: Always check the Error property to verity that the data was added.

5.1.14 Prepare(statement as String) as PreparedSQLStatement

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a PreparedSQLStatement using the SQL statement for use with the various database prepared statement classes.
Notes: A prepared statement is an SQL statement with parameters that has been pre-processed by the database so that it can be executed more quickly if it is re-used with different parameters. Prepared statements also mitigate the risk of SQL injection in web apps.

5.1.15 Rollback

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Cancels an open transaction restoring the database to the state it was in before the transaction began.
Notes: You will generally want to rollback database changes if a database error occurs within the transaction.
You have to have an open transaction to be able to use Rollback. On SQLite (and other databases), you can start a transaction with this command:

BEGIN TRANSACTION

It can be sent using SQLExecute:

DB.SQLExecute("BEGIN TRANSACTION")

5.1.16 RollbackTransaction

Plugin Version: 20.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Cancels an open transaction restoring the database to the state it was in before the transaction began.

5.1.17 SelectSQL(sql As String, ParamArray values As Variant) as RowSet

Plugin Version: 20.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes a SQL SELECT statement and returns the results in a RowSet.
See also:

- 5.1.18 SelectSQL(sql As String, values() As Variant) as RowSet 31

5.1.18 SelectSQL(sql As String, values() As Variant) as RowSet

Plugin Version: 20.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes a SQL SELECT statement and returns the results in a RowSet.
See also:

- 5.1.17 SelectSQL(sql As String, ParamArray values As Variant) as RowSet 31

5.1.19 SQLExecute(ExecuteString as string)

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Used to execute an SQL command. Use this for commands that do not return any data, such as CREATE TABLE or INSERT. ExecuteString contains the SQL statement.
Notes: To avoid SQL Injection, be sure to use Prepared SQL Statements.

5.1.20 SQLSelect(SelectString as string) as RecordSet

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: This is an SQL statement that returns a RecordSet.
Notes: SelectString contains the SQL statement.

Typically only SQL SELECT statements return a RecordSet, but some databases return a RecordSet for SQL commands such as INSERT, UPDATE or stored procedures.

If the SQL does not return data then Nil is returned. Nil is also usually returned if there is an error in the SQL statement, but you should instead check Database.Error to check if an error occurred.

To avoid SQL Injection, be sure to use Prepared SQL Statements.

5.1.21 Properties

5.1.22 DatabaseName as String

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The name of the database to open.
Example:

```
Var db as new SQLDatabaseMBS
db.DatabaseName="PostgreSQL:127.0.0.1,5432@dbname=postgres connect_timeout=10 sslmode=require"
```

Notes: The DatabaseName is typically used with server databases (such as MySQL or PostgreSQL) to identify the specific database to use on the server.

Please set the DatabaseName, UserName and Password properties when using SQLDatabaseMBS class. The Host property is ignored.
The database name must contain the complete information and a prefix for the kind of database.
(Read and Write property)

5.1.23 Error as Boolean

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: True if an error is returned from the database engine.
Notes: See the values of the ErrorCode and ErrorMessage properties to learn the specifics of the error.

You should check the Error property after each database operation to see if there was an error. If there is an error, you can display or log the ErrorCode and ErrorMessage.
(Read and Write property)

5.1.24 ErrorCode as Integer

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The error code returned from the database.
Notes: Error codes and error messages are different for each database.
(Read and Write property)

5.1.25 ErrorMessage as String

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Text of the error message returned from the database.
Notes: Error codes and error messages are different for each database.
You should check the Error property after each database operation to see if there was an error. If there is an error, you can display or log the ErrorCode and ErrorMessage.
(Read and Write property)

5.1.26 Host as String

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The database host name or IP address of the database server.
Notes: (Read and Write property)

5.1.27 Password as String

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The password that is required for access to the database.
Notes: Typically used in conjunction with UserName.
(Read and Write property)

5.1.28 UserName as String

Plugin Version: 1.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The username that is required for access to the database.
Notes: (Read and Write property)

5.2 class DB2MBS

5.2.1 class DB2MBS

Plugin Version: 13.2, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** The class for some DB2 related API commands.
Example:

```
Var cmd as new SQLCommandMBS // your command
Var con as new SQLConnectionMBS // your connection
Var api as DB2MBS = con.NativeAPI

// now use an API function
MsgBox str(api.SQLRowCount(cmd))
```

Notes: More commands are added as requested.

Deprecated in favor of direct methods on SQLCommandMBS. Please let us know if you need more DB2 specific functions.
Subclass of the SQLAPIMBS class.
Blog Entries

- MBS Xojo Plugins, version 19.5pr1
- MBS Xojo / Real Studio Plugins, version 17.1pr1
- MonkeyBread Software Releases the MBS Real Studio plug-ins in version 13.2
- MBS Xojo / Real Studio Plugins, version 13.2pr11

5.2.2 Methods

5.2.3 SQLExecDirect(cmd as SQLCommandMBS, text as string)

Plugin Version: 13.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes an SQL command directly without any preprocessing in the plugin.
Example:

```
Var cmd as new SQLCommandMBS // your command
Var con as new SQLConnectionMBS // your connection
Var api as DB2MBS = con.NativeAPI

// now use an API function
const sql = "some sql command"
```

api.SQLExecDirect cmd, sql

Notes: Lasterror is set.

5.2.4 SQLRowCount(cmd as SQLCommandMBS) as Int64

Plugin Version: 13.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the affected number of rows for the last operation.
Notes: Lasterror is set.

5.2.5 Properties

5.2.6 Lasterror as Integer

Plugin Version: 13.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The last error code.
Notes: (Read and Write property)

5.3 class InformixMBS

5.3.1 class InformixMBS

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** The class for Informix specific functions.
Notes: Deprecated in favor of direct methods on SQLConnectionMBS. Please let us know if you need more Informix specific functions.
Subclass of the SQLAPIMBS class.
Blog Entries

- MBS Xojo Plugins, version 19.5pr1
- MBS Xojo / Real Studio Plugins, version 17.1pr1
- MonkeyBread Software Releases the MBS Real Studio plug-ins in version 13.1
- MonkeyBread Software Releases the MBS Real Studio plug-ins in version 13.0
- MBS Real Studio Plugins, version 13.0fc1

5.3.2 Methods

5.3.3 Error(cmd as SQLCommandMBS, byref SQLState as string, byref NativeError as Integer, byref ErrorMsg as string) as Integer

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the error status.

5.3.4 GetCursorName(cmd as SQLCommandMBS) as string

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries cursor name for given recordset.

5.3.5 HDBC as Integer

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns Database Connection handle.

5.3.6 HENV as Integer

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The environment handle.

5.3.7 HSTMT(cmd as SQLCommandMBS) as Integer

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the statement handle for the command object.
Notes: Returns 0 on any error.

5.3.8 SetCursorName(cmd as SQLCommandMBS, name as string) as boolean

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets cursor name for given recordset.
Notes: Returns true on success.

5.4 module InternalCubeSQLLibraryMBS

5.4.1 module InternalCubeSQLLibraryMBS

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The module for our internal CubeSQL client connector.
Example:

```
if InternalCubeSQLLibraryMBS.Use then
MsgBox "Using internal CubeSQL library."
else
MsgBox "Failed, so please use library file."
end if

Var db As SQLDatabaseMBS // or SQLConnectionMBS

db.Option("ConnectionTimeout") = "30" // 30 seconds timeout
db.Option("ConnectionEncryption") = "SSL_AES256"
db.Option("SSLCertificatePath") = SSLCertificatePath

// connect...
```

Notes: This is a CubeSQL client library built into a plugin, so you can decide with use of MBS SQL Plugin to use this plugin instead of providing your own external copy of CubeSQL shared library.

Version 19.4 or newer have CubeSQL with SSL included.
You can use Option("ConnectionEncryption") with SSL, SSL+AES128, SSL+AES192 and SSL+AES256 as values.

If ConnectionEncryption is SSL, we can pass path from Option("SSLCertificatePath") with the certificate file.
Blog Entries

- Connect to CubeSQL in Xojo
- CubeSQL Library for Mac
- MBS Xojo Plugins, version 18.3pr1
- CubeSQL support for MBS Xojo SQL Plugin

Videos

- Presentation from Xojo Developer Conference 2019 in Miami.

5.4.2 Methods

5.4.3 SSLVersion as String

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries SSL version of library.

5.4.4 Use as boolean

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Registers the built in CubeSQL client library for use.
Example:

```
if InternalCubeSQLLibraryMBS.Use then
MsgBox "Using internal CubeSQL library."
else
MsgBox "Failed, so please use library file."
end if
```

Notes: So instead of having SQL Plugin load CubeSQL shared library from file, we use the one built into this plugin.

5.4.5 Version as String

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries version of library.

5.5 module InternalSQLiteLibraryMBS

5.5.1 module InternalSQLiteLibraryMBS

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The module for our internal SQLite3 engine.
Example:

```
// enable our built-in SQLite library, which supports encryption
Call InternalSQLiteLibraryMBS.Use

// where to store?
Var f As FolderItem = SpecialFolder.Desktop.Child("test.db")
Var storage_database As New SQLDatabaseMBS ' SQLiteDatabase
storage_database.SQLiteEncryptionKey = "aes256:password" ' <- password with AES256 as prefix to pick algorithm
storage_database.DatabaseName = "sqlite:"+ f.NativePath

If storage_database.Connect Then

// create table if this is not yet here
storage_database.SQLExecute "Create table if not exists pics(pic_id integer PRIMARY KEY AUTOINCREMENT, name varchar(20), pic blob)"

// done
MsgBox "Ready"
Else
MsgBox storage_database.ErrorMessage
End If
```

Notes: This is a SQLite3 library built into a plugin, so you can decide with use of MBS SQL Plugin to use this plugin instead of providing your own external copy of SQLite shared library.

Our internal SQLite library includes the following extensions:

- JSON1, the extension to provide JSON functions
- FTS5, the full text search extension in version 5
- R*Tree index extension
- SOUNDEX function
- SQL math functions
- Geopoly extension

- ICU extension for unicode handling
- The RBU Extension
- SQLite Encryption Extension

You can turn on these SQLite extensions at runtime with MBS Plugin:

- UUID extension
- Base64 extension
- CSV extension

More extensions could be added on request.
Blog Entries

- MonkeyBread Software Releases the MBS Xojo Plugins in version 25.3
- SQLDatabase sample
- News from the MBS Xojo Plugins Version 24.1
- Did you know that you can load extensions in SQLite?
- SQLite and ICU Extension for Xojo
- CubeSQL support for MBS Xojo SQL Plugin
- Use JSON functions with SQLite
- MBS Releases the MBS Xojo / Real Studio plug-ins in version 16.4
- MBS Releases the MBS Xojo / Real Studio plug-ins in version 15.2
- SQL Plugin option to include SQLite Library

Xojo Developer Magazine

- 14.1, page 27: The MBS SQL Plugin, An alternative way to connect to databases by Christian Schmitz
- 13.5, page 8: News

5.5.2 Methods

5.5.3 CompileOption(index as Integer) as String

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries name of a compile option.
Example:

```
Var list() as string

for i as Integer = 0 to 100
Var s as string = InternalSQLiteLibraryMBS.CompileOption(i)
if s = "" then exit
Var b as Boolean = InternalSQLiteLibraryMBS.CompileOptionUsed(s)
Var t as string
if b then
t = ": yes"
else
t = ": no"
end if
list.Append s + t
next

MsgBox Join(list, EndOfLine)
```

Notes: Index index starting with zero until you get back an empty name.

5.5.4 CompileOptionUsed(optionName as String) as Boolean

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries if a given compile option was set on or off.
Example:

```
Var list() as string

for i as Integer = 0 to 100
Var s as string = InternalSQLiteLibraryMBS.CompileOption(i)
if s = "" then exit
Var b as Boolean = InternalSQLiteLibraryMBS.CompileOptionUsed(s)
Var t as string
if b then
t = ": yes"
else
t = ": no"
```

```
end if
list.Append s + t
next

MsgBox Join(list, EndOfLine)
```

Notes: If you need a specific option set, please contact MBS support.

5.5.5 DumpToFile(SqliteDBConectionHandle as Ptr, File as FolderItem, TableName as string = "", PreserveRowid as Boolean = false, Newlines as Boolean = false, DumpDataOnly as Boolean = false, DumpNoSys as Boolean = false)

Plugin Version: 22.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Dumps SQLite database to text file just like dump command in SQLite command line tool.
Example:

```
Call InternalSQLiteLibraryMBS.Use

Var con As New SQLConnectionMBS

// connect....

Var handle As ptr = con.SQLiteConnectionHandle
Var file As FolderItem = SpecialFolder.Desktop.Child("dump.txt")
InternalSQLiteLibraryMBS.DumpToFile(handle, file)
```

Notes: Must use the internal SQLite database, otherwise it will crash.
Raises exception if handle or file is nil.
SQL errors are output by SQLite to the output file.

5.5.6 DumpToString(SqliteDBConectionHandle as Ptr, byref Data as String, MaximumSize as Integer = 10000000, TableName as string = "", PreserveRowid as Boolean = false, Newlines as Boolean = false, DumpDataOnly as Boolean = false, DumpNoSys as Boolean = false)

Plugin Version: 22.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Dumps SQLite database to text file just like dump command in SQLite command line tool.
Example:

```
Call InternalSQLiteLibraryMBS.Use
```

```
Var con As New SQLConnectionMBS

// connect....

Var handle As ptr = con.SQLiteConnectionHandle
Var data as string
InternalSQLiteLibraryMBS.DumpToFile(handle, data, 1000000)
```

Notes: Must use the internal SQLite database, otherwise it will crash.
Raises exception if handle is nil or we run out of memory.
SQL errors are output by SQLite to the output file.

Please pass size of memory we should allocate. Then we can run the dump.
if the final COMMIT is missing or the size is close to the limit you gave, you may run again with higher limit.

5.5.7 DumpToStrings(SqliteDBConectionHandle as Ptr, SchemaName as String = "", TableName as string = "") as String()

Plugin Version: 25.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Convert an SQLite database into SQL statements that will recreate that database.
Notes: Schema: Which schema to dump. Usually "main".
Table: Which table to dump. Pass "" for all tables.

This subroutine converts the content of an SQLite database into UTF-8 text SQL statements that can be used to exactly recreate the original database. ROWID values are preserved.

The connection handle parameter is the SQLite database connection using our internal SQLite library. Schema is the schema within that database which is to be dumped. Usually the zSchema is "main" but can also be "temp" or any ATTACH-ed database. If Table is not empty, then only the content of that one table is dumped. If zTable is empty, then all tables are dumped.

The generate text is collected and returned as a string array.

5.5.8 isKeyword(name as string) as boolean

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks if an identifier is a keyword.
Example:

```
if InternalSQLiteLibraryMBS.isKeyword("TABLE") then
MsgBox "Table is a keyword"
else
MsgBox "Table is not a keyword!"
end if
```

5.5.9 Keywords as String()

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries list of keywords.
Example:

```
MsgBox Join(InternalSQLiteLibraryMBS.Keywords,EndOfLine)
```

5.5.10 LoadICU as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads ICU libraries now.
Example:

```
If InternalSQLiteLibraryMBS.ICULoaded Then
MessageBox "already loaded"
ElseIf InternalSQLiteLibraryMBS LoadICU Then
MessageBox "loaded successfully"
Else
MessageBox "failed to load the unicode libraries."
End If
```

Notes: Returns true on success.
Normally we load them when SQLite initializes and we then add the ICU extension if we find the library.
Call this function toe explicit load them now.

If MBS Plugin can find International Components for Unicode library files, we can load them and use them for proper unicode handling in SQLite.

5.5.11 SourceID as String

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The source code ID.
Example:

MsgBox InternalSQLiteLibraryMBS.SourceID

5.5.12 Use as boolean

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Registers the built in SQLite library for use.
Example:

```
if InternalSQLiteLibraryMBS.Use then
MsgBox "Using internal SQLite"
else
MsgBox "Failed, so please use library file."
end if
```

Notes: So instead of having SQL Plugin load sqlite3 shared library from file, we use the one built into this plugin.

5.5.13 Version as String

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the version number.
Example:

```
MsgBox InternalSQLiteLibraryMBS.Version
```

5.5.14 VersionNumber as Integer

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the sqlite version number.
Example:

```
MsgBox str(InternalSQLiteLibraryMBS.VersionNumber)
```

5.5.15 Properties

5.5.16 Base64ExtensionEnabled as Boolean

Plugin Version: 25.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether to enable the Base64 extensions.
Notes: This is a SQLite extension for converting in either direction between a (binary) blob and base64 text. Base64 can transit a sane USASCII channel unmolested. It also plays nicely in CSV or written as TCL brace-enclosed literals or SQL string literals, and can be used unmodified in XML-like documents.

This is an independent implementation of conversions specified in RFC 4648, done on the above date by the author (Larry Brasfield) who thereby has the right to put this into the public domain.

The conversions meet RFC 4648 requirements, provided that this C source specifies that line-feeds are included in the encoded data to limit visible line lengths to 72 characters and to terminate any encoded blob having non-zero length.

Length limitations are not imposed except that the runtime SQLite string or blob length limits are respected. Otherwise, any length binary sequence can be represented and recovered. Generated base64 sequences, with their line-feeds included, can be concatenated; the result converted back to binary will be the concatenation of the represented binary sequences.

This SQLite3 extension creates a function, base64(x), which either: converts text x containing base64 to a returned blob; or converts a blob x to returned text containing base64. An error will be thrown for other input argument types.
(Read and Write property)

5.5.17 CSVExtensionEnabled as Boolean

Plugin Version: 25.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether to enable the CSV extensions.
Notes: This file contains the implementation of an SQLite virtual table for reading CSV files.

Usage:

.load ./csv
CREATE VIRTUAL TABLE temp.csv USING csv(filename=FILENAME);
SELECT * FROM csv;

The columns are named "c1", "c2", "c3", ... by default. Or the application can define its own CREATE TABLE statement using the schema= parameter, like this:

CREATE VIRTUAL TABLE temp.csv2 USING csv(
filename = "../http.log",
schema = "CREATE TABLE x(date,ipaddr,url,referrer,userAgent)"

);

Instead of specifying a file, the text of the CSV can be loaded using the data= parameter.

If the columns=N parameter is supplied, then the CSV file is assumed to have N columns. If both the columns= and schema= parameters are omitted, then the number and names of the columns is determined by the first line of the CSV input.
(Read and Write property)

5.5.18 ICUEnabled as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether ICU libraries are enabled.
Notes: If MBS Plugin can find International Components for Unicode library files, we can load them and use them for proper unicode handling in SQLite.
Default is true to enable them, if possible.
(Read and Write property)

5.5.19 ICULoaded as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether ICU libraries are loaded.
Notes: If MBS Plugin can find International Components for Unicode library files, we can load them and use them for proper unicode handling in SQLite.
(Read only property)

5.5.20 ICUUsed as Boolean

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether ICU libraries have been used.
Notes: Once SQLite is initialized, we set this to 1 if we passed the functionality to SQLite.
Starts with false and should later turn true if ICU libraries are loaded and enabled for usage.
(Read only property)

5.5.21 MemoryHighwater as Int64

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries high water memory usage mark.
Example:

```
MsgBox "SQLite uses "+_
str(InternalSQLiteLibraryMBS.MemoryUsed)+_
" bytes and has a high mark of "+_
str(InternalSQLiteLibraryMBS.MemoryHighwater)+" bytes."
```

Notes: (Read only property)

5.5.22 MemoryUsed as Int64

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries current memory usage.
Example:

```
MsgBox "SQLite uses "+_
str(InternalSQLiteLibraryMBS.MemoryUsed)+_
" bytes and has a high mark of "+_
str(InternalSQLiteLibraryMBS.MemoryHighwater)+" bytes."
```

Notes: (Read only property)

5.5.23 Path as String

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries file path to the plugin library.
Notes: Allows you to query path for plugin library, so you can pass it to SQLite to load as extension.
(Read only property)

5.5.24 UUIDExtensionEnabled as Boolean

Plugin Version: 25.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether to enable the uuid extensions.
Notes: This SQLite extension implements functions that handling RFC-4122 UUIDs

Three SQL functions are implemented:

uuid()	generate a version 4 UUID as a string
uuid_str(X)	convert a UUID X into a well-formed UUID string
uuid_blob(X)	convert a UUID X into a 16-byte blob

The output from uuid() and uuid_str(X) are always well-formed RFC-4122 UUID strings in this format:

xxxxxxxx-xxxx-Mxxx-Nxxx-xxxxxxxxxxxx

All of the 'x', 'M', and 'N' values are lower-case hexadecimal digits. The M digit indicates the "version". For uuid()-generated UUIDs, the version is always "4" (a random UUID). The upper three bits of N digit are the "variant". This library only supports variant 1 (indicated by values of N between '8' and 'b') as those are overwhelming the most common. Other variants are for legacy compatibility only.

The output of uuid_blob(X) is always a 16-byte blob. The UUID input string is converted in network byte order (big-endian) in accordance with RFC-4122 specifications for variant-1 UUIDs. Note that network byte order is *always* used, even if the input self-identifies as a variant-2 UUID.

The input X to the uuid_str() and uuid_blob() functions can be either a string or a BLOB. If it is a BLOB it must be exactly 16 bytes in length or else a NULL is returned. If the input is a string it must consist of 32 hexadecimal digits, upper or lower case, optionally surrounded by { ... } and with optional "-" characters interposed in the middle. The flexibility of input is inspired by the PostgreSQL implementation of UUID functions that accept in all of the following formats:

- A0EEBC99-9C0B-4EF8-BB6D-6BB9BD380A11
- { a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11 }
- a0eebc999c0b4ef8bb6d6bb9bd380a11
- a0ee-bc99-9c0b-4ef8-bb6d-6bb9-bd38-0a11
- { a0eebc99-9c0b4ef8-bb6d6bb9-bd380a11 }

If any of the above inputs are passed into uuid_str(), the output will always be in the canonical RFC-4122 format:

- a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11

If the X input string has too few or too many digits or contains stray characters other than { , } , or -, then NULL is returned.
(Read and Write property)

5.6 class MySQLMBS

5.6.1 class MySQLMBS

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** The class for MySQL specific functionality.
Notes: Deprecated in favor of direct methods on SQLConnectionMBS. Please let us know if you need more MySQL or MariaDB specific functions.
Subclass of the SQLAPIMBS class.
Blog Entries

- MBS Xojo Plugins, version 19.5pr1
- MBS Xojo / Real Studio Plugins, version 17.1pr1
- MonkeyBread Software Releases the MBS Real Studio plug-ins in version 12.5
- MBS Real Studio Plugins, version 12.5pr7
- MBS Real Studio Plugins, version 12.5pr4

5.6.2 Methods

5.6.3 AffectedRows as UInt64

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the number of affected rows in the last statement.
Notes: see also
http://dev.mysql.com/doc/refman/5.1/en/mysql-affected-rows.html

5.6.4 Error as string

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the last error text.
Notes: see also
http://dev.mysql.com/doc/refman/5.1/en/mysql-error.html

5.6.5 ErrorNumber as UInt32

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the last error code.
Notes: see also
http://dev.mysql.com/doc/refman/5.1/en/mysql-errno.html

5.6.6 FieldCount as UInt32

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of columns for the most recent query on the connection.
Notes: see also
http://dev.mysql.com/doc/refman/5.1/en/mysql-field-count.html

5.6.7 Info as string

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Retrieves an info string providing information about the most recently executed statement.
Notes: see also
http://dev.mysql.com/doc/refman/5.1/en/mysql-info.html

5.6.8 InsertID as Int64

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the last auto increment value from last insert command.
Notes: Please query value right after doing Insert. This value is reset when you call commit.
For mySQL and MariaDB connections.

see also
http://dev.mysql.com/doc/refman/5.1/en/mysql-insert-id.html

5.6.9 NumberOfRows(cmd as SQLCommandMBS) as UInt64

Plugin Version: 12.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries number of records in a command.

5.6.10 SetSSL(keyPath as string, CertificatePath as string, AuthorityPath as string, authorityFolderPath as string, Cipher as string)

Plugin Version: 12.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets SSL connection parameters.
Notes: Calls mysql_ssl_set internally.
Used for establishing secure connections using SSL. It must be called before Connect(). It does nothing unless SSL support is enabled in the client library.

keyPath is the path name to the key file.
CertificatePath is the path name to the certificate file.
AuthorityPath is the path name to the certificate authority file.
authorityFolderPath is the path name to a directory that contains trusted SSL CA certificates in PEM format.
Cipher is a list of permissible ciphers to use for SSL encryption.

Any unused SSL parameters may be given as empty string.
For paths, please use folderitem.NativePath and not folderitem.ShellPath.

Please switch to using the following options on the connection:

MYSQL_SSL_KEY
MYSQL_SSL_CERT
MYSQL_SSL_CA
MYSQL_SSL_CAPATH
MYSQL_SSL_CIPHER

They should be specified before the connection is made.
e.g.

db.Option("MYSQL_SSL_CIPHER") = "DHE-RSA-AES256-SHA"

Allows to specify MySQL SSL parameters that will be used with mysql_ssl_set. MySQL API method called only when at least one parameter specified. See MySQL documentation for more information about these options.

5.7 class PostgreSQLAPIMBS

5.7.1 class PostgreSQLAPIMBS

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** The class for Postgre SQL specific functions.
Notes: The Listen method and Notification event are in the SQLDatabaseMBS/SQLConnectionMBS classes directly.

Deprecated in favor of direct methods on SQLCommandMBS. Please let us know if you need more postgreSQL specific functions.
Subclass of the SQLAPIMBS class.
Blog Entries

- MBS Xojo Plugins, version 19.5pr1
- MBS Xojo / Real Studio Plugins, version 17.1pr1
- Postgre SQL Database Extension

5.7.2 Methods

5.7.3 DB as string

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The database name used to create the connection.

5.7.4 ErrorMessage as string

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The last error message.

5.7.5 Field(cmd as SQLCommandMBS, RecordIndex as Integer, FieldIndex as Integer) as string

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries a field by index for the row with the RecordIndex.
See also:

5.7.6 Field(cmd as SQLCommandMBS, RecordIndex as Integer, FieldName as string) as string

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries a field by name for the row with the RecordIndex.
See also:

• 5.7.5 Field(cmd as SQLCommandMBS, RecordIndex as Integer, FieldIndex as Integer) as string 54

5.7.7 FieldCount(cmd as SQLCommandMBS) as Integer

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of fields in the result.

5.7.8 Host as string

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The host used to create the connection.

5.7.9 Options as string

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The options used to create the connection.

5.7.10 Password as string

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The password used to create the connection.

5.7.11 Port as string

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The port used to create the connection.

5.7.12 RecordCount(cmd as SQLCommandMBS) as Integer

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of records in the result.

5.7.13 TTY as string

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The tty used to create the connection.

5.7.14 User as string

Plugin Version: 9.8, Platforms: macOS, Linux, Windows, Targets: All.

Function: The user name used to create the connection.

5.8 Globals

5.8.1 BuildRecordSetMBS(fieldNames() as string, values() as string) as RecordSet

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Builds a recordset from strings.
Example:

```
Var names() as string = array("Firstname", "Lastname")
Var values() as string

values.append "Stefan"
values.append "Miller"
values.append "Patrick"
values.append "Maier"

Var r as RecordSet = BuildRecordSetMBS(names, values)
```

Notes: First array has field names. Second array has all values.
As plugin can't access multi dimensional arrays, we have to flatten it into one dimension and concat all rows.
Returns nil on low memory.
Array sizes should be like: Ubound(values)+1 = (ubound(fieldNames)+1) * RecordCount

See also BuildRowSetMBS for newer Xojo versions.
Blog Entries

- Cleanup Xojo Plugins
- RowSet in MBS Xojo SQL Plugin
- MBS Xojo Plugins, version 20.5pr6
- RecordSet to JSON and back
- MonkeyBread Software Releases the MBS Real Studio plug-ins in version 13.0
- MBS Real Studio Plugins, version 13.0pr6
- RecordSet news

5.8.2 BuildRowSetMBS(fieldNames() as string, values() as string) as RowSet

Plugin Version: 21.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Builds a recordset from strings.
Example:

```
Var names() as string = array("Firstname", "Lastname")
Var values() as string

values.append "Stefan"
values.append "Miller"
values.append "Patrick"
values.append "Maier"

Var r as RowSet = BuildRowSetMBS(names, values)
```

Notes: First array has field names. Second array has all values.
As plugin can't access multi dimensional arrays, we have to flatten it into one dimension and concat all rows.
Returns nil on low memory.
Array sizes should be like: Ubound(values)+1 = (ubound(fieldNames)+1) * RecordCount

RowSet requires Xojo 2019r2 or newer.
Blog Entries

- Cleanup Xojo Plugins
- MonkeyBread Software Releases the MBS Xojo Plugins in version 21.1
- MBS Xojo Plugins, version 21.1pr4
- RowSet in MBS Xojo SQL Plugin

5.8.3 CloneRecordSetMBS(rec as RecordSet) as RecordSet

Plugin Version: 12.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates an in memory copy of the RecordSet.
Notes: This copied record set can be used instead of the original one and even after the original database connection is closed.
Blog Entries

- Cleanup Xojo Plugins
- RecordSet news
- MBS Real Studio Plugins, version 12.5pr13

5.9 class RecordSet

5.9.1 class RecordSet

Plugin Version: 12.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in recordset class in Xojo.

5.9.2 Methods

5.9.3 CloneMBS as RecordSet

Plugin Version: 12.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates an in memory copy of the RecordSet.
Notes: This copied record set can be used instead of the original one and even after the original database connection is closed.
Blog Entries

- MBS Xojo / Real Studio Plugins, version 16.3pr5
- MonkeyBread Software Releases the MBS Real Studio plug-ins in version 12.5
- MBS Real Studio Plugins, version 12.5pr13
- Cloning RecordSets

Xojo Developer Magazine

- 11.1, page 9: News

5.10 class SQLAPIMBS

5.10.1 class SQLAPIMBS

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** This is a class for the native APIs.
Notes: The plugin does not implem
Blog Entries

- Cleanup Xojo Plugins
- MBS Xojo Plugins, version 19.5pr1
- MBS REALbasic plug-ins version 9.5

5.10.2 Properties

5.10.3 ClassName as String

Plugin Version: 14.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class name of the underlying C++ class.
Notes: Sometimes useful to see a which API currently is used.
(Read only property)

5.10.4 Connection as Variant

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The database connection this API is used with.
Notes: (Read only property)

5.11 class SQLBLobMBS

5.11.1 class SQLBLobMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: A class for a blob.
Notes: Basicly this is a SQLStringMBS which is always marked to contain binary data. You only need this class to use the constructor with dataprovider to stream data to the database.
Subclass of the SQLLongOrLobMBS class.
Blog Entries

- MBS Real Studio Plugins, version 12.4pr1

5.11.2 Methods

5.11.3 Constructor

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.
See also:

5.11.4 Constructor(Data as MemoryBlock)

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new string object with data, e.g. for blob.
See also:

5.11.5 Constructor(data as SQLStringMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new blob object from a string object.
See also:

- 5.11.3 Constructor 61
- 5.11.4 Constructor(Data as MemoryBlock) 61
- 5.11.6 Constructor(Data as string, isText as Boolean = True) 62
- 5.11.7 Constructor(dataProvider as SQLDataProviderMBS, BlockSize as UInt32) 62

5.11.6 Constructor(Data as string, isText as Boolean = True)

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new string object with data or text copied from the data string.
Notes: If isText is true, the data is interpreted as text and string encoding conversion may modify it. If isText is false the bytes are copied raw.
See also:

- 5.11.3 Constructor 61
- 5.11.4 Constructor(Data as MemoryBlock) 61
- 5.11.5 Constructor(data as SQLStringMBS) 62
- 5.11.7 Constructor(dataProvider as SQLDataProviderMBS, BlockSize as UInt32) 62

5.11.7 Constructor(dataProvider as SQLDataProviderMBS, BlockSize as UInt32)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new blob object from a data provider.
Notes: The blocksize specifies in which sizes data is requested from the provider.
You must make sure that the data provider and this new blob object life long enough. Because the actual data is requested later when you do the update on the database.

If BlockSize is 0, the default block size is used.
See also:

- 5.11.3 Constructor 61
- 5.11.4 Constructor(Data as MemoryBlock) 61
- 5.11.5 Constructor(data as SQLStringMBS) 62
- 5.11.6 Constructor(Data as string, isText as Boolean = True) 62

5.12 class SQLBytesMBS

5.12.1 class SQLBytesMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for a string of bytes.
Notes: Subclass of the SQLStringMBS class.
Blog Entries

- MBS Real Studio Plugins, version 12.4pr1

5.12.2 Methods

5.12.3 Constructor

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.
See also:

- 5.12.4 Constructor(Data as MemoryBlock) 63
- 5.12.5 Constructor(data as SQLStringMBS) 63
- 5.12.6 Constructor(Data as string, isText as Boolean = True) 64

5.12.4 Constructor(Data as MemoryBlock)

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new string object with data, e.g. for blob.
See also:

- 5.12.3 Constructor 63
- 5.12.5 Constructor(data as SQLStringMBS) 63
- 5.12.6 Constructor(Data as string, isText as Boolean = True) 64

5.12.5 Constructor(data as SQLStringMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new bytes object based on the given string object.
See also:

5.12.6 Constructor(Data as string, isText as Boolean = True)

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new string object with data or text copied from the data string.
Notes: If isText is true, the data is interpreted as text and string encoding conversion may modify it. If isText is false the bytes are copied raw.
See also:

5.13 class SQLCLobMBS

5.13.1 class SQLCLobMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: A class for a clob (character large object).
Notes: Basicly this is a SQLStringMBS which is always marked to contain text. You only need this class to use the constructor with dataprovider to stream data to the database.
Subclass of the SQLLongOrLobMBS class.
Blog Entries

- MBS Real Studio Plugins, version 12.4pr1

5.13.2 Methods

5.13.3 Constructor

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.
See also:

5.13.4 Constructor(data as SQLStringMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new clob object from a string object.
See also:

5.13.5 Constructor(Data as string, isText as boolean=true)

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new string object with data or text copied from the data string.
Notes: If isText is true, the data is interpreted as text and string encoding conversion may modify it. If isText is false the bytes are copied raw.
See also:

- 5.13.3 Constructor 65
- 5.13.4 Constructor(data as SQLStringMBS) 65
- 5.13.6 Constructor(dataProvider as SQLDataProviderMBS, BlockSize as UInt32) 66

5.13.6 Constructor(dataProvider as SQLDataProviderMBS, BlockSize as UInt32)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new clob object from a data provider.
Notes: The blocksize specifies in which sizes data is requested from the provider.
You must make sure that the data provider and this new clob object life long enough. Because the actual data is requested later when you do the update on the database.

If BlockSize is 0, the default block size is used.
See also:

- 5.13.3 Constructor 65
- 5.13.4 Constructor(data as SQLStringMBS) 65
- 5.13.5 Constructor(Data as string, isText as boolean=true) 65

5.14 class SQLCommandMBS

5.14.1 class SQLCommandMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This is the central class for the using the SQL database access.
Example:

```
Var con as SQLConnectionMBS
Var cmd as SQLCommandMBS

try

con = new SQLConnectionMBS // connection object
cmd = new SQLCommandMBS // create command object

// where is the library?
con.SetFileOption con.kOptionLibraryMySQL, SpecialFolder.UserHome.Child("libmysqlclient.dylib")

// connect to database (mySQL in our example)
// server: 192.168.1.80
// port: 3306
// database: test
// name: root
// no password
con.Connect("192.168.1.80,3306@test","root","",SQLConnectionMBS.kMySQLClient)
// associate a command with connection
// connection can also be specified in SACommand constructor
cmd.Connection=con

// create table
cmd.setCommandText("Create table test_tbl(fid integer, fvarchar20 varchar(20), fblob blob)")
cmd.Execute

// insert value
cmd.setCommandText("Insert into test_tbl(fid, fvarchar20) values (1, 'Some string (1)')")
cmd.Execute

// commit changes on success
con.Commit

MsgBox("Table created, row inserted!")

catch r as SQLErrorExceptionMBS
// SAConnection::Rollback()
// can also throw an exception
// (if a network error for example),
```

```
// we will be ready
try

// on error rollback changes
if con<>nil then
con.rollback
end if
catch x as SQLErrorExceptionMBS
// ignore
end try

// show error message
MsgBox r.message
end try
```

Notes: The plugin can cache the recordset locally. To enable you can call SQLCommandMBS.Cache or use the Option("AutoCache") = "true" on either command or connection or database objects. The plugin will than fetch all records and store them in memory. After this you can walk over the recordset and use FetchPos, FetchFirst, FetchLast, FetchPrev and FetchNext to locate the rows you need. When you call Field() you always get last row, but to read from cached result set, please use Value() function. When using RecordSet, the values are read via Value() functions automatically.

see also
https://www.sqlapi.com/ApiDoc/class_s_a_command.html

Blog Entries

- Connect to Postgres in Xojo
- MBS SQL Plugin Tips and Tricks
- MonkeyBread Software Releases the MBS Xojo Plugins in version 21.4
- RowSet in MBS Xojo SQL Plugin
- Multithreaded plugin functions can increase speed of Xojo application
- Prefetching records from databases
- Multiple recordsets with Microsoft SQL Server
- Wishes for little plug-in additions?
- Multithreaded plugin functions can increase speed of Real Studio application
- MBS REALbasic Plugins Version 10.4 release notes

Xojo Developer Magazine

- 14.1, pages 28 to 30: The MBS SQL Plugin, An alternative way to connect to databases by Christian Schmitz

- 14.1, pages 24 to 26: The MBS SQL Plugin, An alternative way to connect to databases by Christian Schmitz

- 12.2, page 10: News

5.14.2 Methods

5.14.3 AsRecordSet as RecordSet

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a recordset using the command to query fields.
Notes: You can use normal RecordSet functions to walk through fields and they simply control the command object.
This is for convenience like passing RecordSet to report functions in Xojo.

For this method to work, you need to have somewhere a property with SQLDatabaseMBS so Xojo includes our SQLDatabase plugin which provides the RecordSet functionality.

The record set may not have a valid RecordCount or have working movefirst/movelast/moveprev methods unless the underlaying database supports those and Scrollable result sets is enabled/supported.

5.14.4 AsRowSet as RowSet

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a RowSet using the command to query fields.
Notes: You can use normal RowSet functions to walk through fields and they simply control the command object.
This is for convenience like passing RowSet to other functions in Xojo.

For this method to work, you need to have somewhere a property with SQLDatabaseMBS so Xojo includes our SQLDatabase plugin which provides the RowSet functionality.

The RowSet may not have a valid RecordCount or have working movefirst/movelast/moveprev methods unless the underlaying database supports those and Scrollable result sets is enabled/supported.

Requires Xojo 2019r2 or newer.

5.14.5 Cache

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Caches values.
Notes: The plugin will load the whole recordset and store it in memory.
Now you can move forward/backward as needed to read data.

If you set Option("AutoCache") = "true", the plugin will call Cache automatically for all result sets.
We can only cache first result set.

5.14.6 Cancel

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Attempts to cancel the pending result set, or current statement execution.
Notes: Only if isExecuting is true, doing cancel makes sense.

Cancel can cancel the following types of processing on a statement:

A function running asynchronously on the statement.
A function running on the statement on another thread.
After an application calls a function asynchronously, it checks repeatedly to determine whether it has finished processing. While the function is processing, an application can call Cancel to cancel the function.

In a multithread application, the application can cancel a function that is running synchronously on a statement.

see also
https://www.sqlapi.com/ApiDoc/class_s_a_command.html

5.14.7 Close

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Closes the specified command object.
Notes: Use the Close method to close the command explicitly.

A command will be implicitly closed in destructor, so you don't have to call Close method explicitly.

5.14.8 Constructor

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new command object with no connection and no command text.
See also:

- 5.14.9 Constructor(connection as SQLConnectionMBS, SQLCommand as String, CommandType as Integer = 0) 71

5.14.9 Constructor(connection as SQLConnectionMBS, SQLCommand as String, CommandType as Integer = 0)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This constructor initializes a new SQLCommandMBS object.
Example:

```
// your connection
Var con as SQLConnectionMBS
Var SQL as string = "Insert into test_tbl(fid, fvarchar20) values(:id, :name)"

// create command object
Var cmd as new SQLCommandMBS(con, sql)

// assign values by name of parameter:
cmd.Param("id").setAsLong(2)
cmd.Param("name").setAsString(new SQLStringMBS("Some string (2)"))

// Insert first row
cmd.Execute
```

Notes: Connection: the connection to associated with the command.
SQLCommand: A string which represents command text string (an SQL statement or a stored procedure name). If it is an empty string, no command text is associated with the command, and you have to call setCommandText method later.
CommandType: The type of command like kCommandTypeUnknown, kCommandTypeSQLStatement, kCommandTypeSQLStatementRaw or kCommandTypeStoredProcedure.

All text strings sent to the plugin must have a defined encoding. Else the internal text encoding conversions will fail.
See also:

- 5.14.8 Constructor 71

5.14.10 CreateParam(name as string, ParamType as Integer, DirType as Integer=0) as SQLParamMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates parameter associated with the specified command.
Notes: Parameters

name:	A string representing the name of parameter.
ParamType:	Type of the parameter's value. Use the kDataType constants.
ParamSize:	An integer value represents parameter's value size.
ParamPrecision:	An integer value represents parameter's value precision.
ParamScale:	An integer value represents parameter's value scale.
DirType:	Type of the parameter. Use the kParamDirType* constants.

Returns a new SQLParamMBS object on success or nil on any error.

Normally you should not create parameters by yourself. The Library automatically detects whether the command has parameters in terms of the command text and implicitly creates a set of SAParam objects.

Nevertheless, if you call CreateParam explicitly you have to delete all SAParam objects created automatically by the Library before. Use DestroyParams method before the first call of CreateParam method.
See also:

- 5.14.11 CreateParam(name as string, ParamType as Integer, NativeType as Integer, ParamSize as Integer, ParamPrecision as Integer, ParamScale as Integer, DirType as Integer=0) as SQLParamMBS 72

5.14.11 CreateParam(name as string, ParamType as Integer, NativeType as Integer, ParamSize as Integer, ParamPrecision as Integer, ParamScale as Integer, DirType as Integer=0) as SQLParamMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates parameter associated with the specified command.
Notes: Parameters

Returns a new SQLParamMBS object on success or nil on any error.

Normally you should not create parameters by yourself. The Library automatically detects whether the command has parameters in terms of the command text and implicitly creates a set of SAParam objects.

name:	A string representing the name of parameter.
ParamType:	Type of the parameter's value. Use the kDataType constants.
ParamSize:	An integer value represents parameter's value size.
ParamPrecision:	An integer value represents parameter's value precision.
ParamScale:	An integer value represents parameter's value scale.
DirType:	Type of the parameter. Use the kParamDirType* constants.

Nevertheless, if you call CreateParam explicitly you have to delete all SAParam objects created automatically by the Library before. Use DestroyParams method before the first call of CreateParam method.
See also:

- 5.14.10 CreateParam(name as string, ParamType as Integer, DirType as Integer=0) as SQLParamMBS 72

5.14.12 DB2SQLExecDirect(sql as string)

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes an SQL command directly without any preprocessing in the plugin.
Example:

```
Var cmd as new SQLCommandMBS // your command
Var con as new SQLConnectionMBS // your connection

// now use an API function
const sql = "some sql command"
cmd.DB2SQLExecDirect sql
```

Notes: Lasterror is set.

5.14.13 DB2SQLRowCount as Int64

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the affected number of rows for the last operation.
Notes: Lasterror is set.

5.14.14 DestroyParams

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Destroys all parameters associated with the specified command.
Notes: DestroyParams method destroys all parameters either created automatically by the Library or by user.

Normally you should not create and delete parameters by yourself. The Library automatically detects whether the command has parameters, implicitly creates a set of SAParam objects and then deletes them in SACommanddestructor. But if you have some reason to create parameters explicitly use CreateParam method and then call DestroyParams method to delete all parameters after your work with parameters is over.

5.14.15 Execute

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes the current command.
Example:

```
// your connection
Var con as SQLConnectionMBS
Var SQL as string = "Insert into test_tbl(fid, fvarchar20) values(:id, :name)"

// create command object
Var cmd as new SQLCommandMBS(con, sql)

// assign values by name of parameter:
cmd.Param("id").setAsLong(2)
cmd.Param("name").setAsString(new SQLStringMBS("Some string (2)"))

// Insert first row
cmd.Execute
```

Notes: Use the Execute method to execute the query or stored procedure specified in the command text. Execute method calls Prepare method implicitly if needed. If the command has input parameters, they should be bound before calling Execute method. Input parameters represented by SAParam object. To bind input variables assign a value to SAParam object returning by Param or ParamByIndex methods.

A command (an SQL statement or procedure) can have a result set after executing. To check whether a result set exists use isResultSet method. If result set exists, a set of SAField objects is created after command execution. Rows from the result set can be fetched one by one using FetchNext method. To get field description or value use Field method.

Output parameters represented by SAParam objects. They are available after command execution. To get parameter description or value use Param or ParamByIndex methods.

5.14.16 ExecuteCommand(SQLCommand as string, CommandType as Integer=0)

Plugin Version: 10.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes the given command.
Notes: This is a convenience function.
Internally it calls setCommandText with the given command and calls Execute.

All text strings sent to the plugin must have a defined encoding. Else the internal text encoding conversions will fail.

5.14.17 ExecuteCommandMT(SQLCommand as string, CommandType as Integer=0)

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes the given command.
Notes: This is a convenience function.
Internally it calls setCommandText with the given command and calls Execute.

The work is performed on a preemptive thread, so this function does not block the application and can yield time to other Xojo threads. Must be called in a Xojo thread to enjoy benefits. If called in main thread will block, but keep other background threads running.

All text strings sent to the plugin must have a defined encoding. Else the internal text encoding conversions will fail.

5.14.18 ExecuteMT

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes the current command.
Notes: Use the Execute method to execute the query or stored procedure specified in the command text. Execute method calls Prepare method implicitly if needed. If the command has input parameters, they should be bound before calling Execute method. Input parameters represented by SAParam object. To bind input variables assign a value to SAParam object returning by Param or ParamByIndex methods.

A command (an SQL statement or procedure) can have a result set after executing. To check whether a result set exists use isResultSet method. If result set exists, a set of SAField objects is created after command execution. Rows from the result set can be fetched one by one using FetchNext method. To get field description or value use Field method.

Output parameters represented by SAParam objects. They are available after command execution. To get parameter description or value use Param or ParamByIndex methods.

The work is performed on a preemptive thread, so this function does not block the application and can yield time to other Xojo threads. Must be called in a Xojo thread to enjoy benefits. If called in main thread will block, but keep other background threads running.

5.14.19 FetchFirst as boolean

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fetches first row from a result set.
Notes: Same as FetchNext, but jumps to the first row.
Returns true if the row was fetched; otherwise false.

Not supported for Interbase and SQLite.

When you cache the result set, you can always move within the result set.

5.14.20 FetchLast as boolean

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fetches last row from a result set.
Notes: Same as FetchNext, but jumps to the last row.
Returns true if the row was fetched; otherwise false.

Not supported for Interbase and SQLite.

When you cache the result set, you can always move within the result set.

5.14.21 FetchNext as boolean

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fetches next row from a result set.
Notes: Returns true if the next row was fetched; otherwise false .

Use FetchNext method to fetch row by row from the result set.

Each column of fetched row is represented by SAField object. If a result set exists after the last command execution, a set of SAField objects is created implicitly. To check whether a result set exists use isResultSet method. FetchNext method updates value parts of SAField objects.

To get field description or value use Field method.

When you cache the result set, you can always move within the result set.

5.14.22 FetchPos(offset as Integer, relative as boolean = false) as boolean

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fetches a row by index.
Notes: Returns true if the row was fetched; otherwise false.
You may need to request recordset to be scrollable to have this work.
For that, please set Option("Scrollable") = "true" before doing the query.

When you cache the result set, you can always move within the result set.

5.14.23 FetchPrior as boolean

Plugin Version: 11.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Fetches previous row from a result set.
Notes: Returns true if the row was fetched; otherwise false.
Same as FetchNext, just going back inside the result set.

Not supported for Interbase and SQLite.

When you cache the result set, you can always move within the result set.

5.14.24 Field(index as Integer) as SQLFieldMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the column specified by its position in the result set.
Example:

```
Var c as SQLCommandMBS // your command object

// get field by name
Var f1 as SQLFieldMBS = c.Field("FirstName")

// get field by Index
Var f2 as SQLFieldMBS = c.Field(1)
```

Notes: index: A one-based field number in a result set.

Use Field method to access a field by its name or position in the result set.
For Cached result sets, please use Value() function to get values.

Using an index smaller than 1 and greater then the value returned by FieldCount method will result in a failed assertion.

A set of SAField objects creates implicitly after the command execution if the result set exists.SAField object contains full information about a column: name, type, size, value.

Raises OutOfBoundsException exception if index parameter is out of range.
See also:

- 5.14.25 Field(name as string) as SQLFieldMBS 78

5.14.25 Field(name as string) as SQLFieldMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the column specified by its name in the result set.
Example:

```
Var c as SQLCommandMBS // your command object

// get field by name
Var f1 as SQLFieldMBS = c.Field("FirstName")

// get field by Index
Var f2 as SQLFieldMBS = c.Field(1)
```

Notes: name: A string that represents a name of the requested field.

Returns a reference to a SAField object.

Use Field method to access a field by its name or position in the result set.
For Cached result sets, please use Value() function to get values.

Using a non-existent field name will throw an exception.

A set of SAField objects creates implicitly after the command execution if the result set exists.SAField object contains full information about a column: name, type, size, value.
See also:

- 5.14.24 Field(index as Integer) as SQLFieldMBS 77

5.14.26 FieldExists(name as string) as Boolean

Plugin Version: 21.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Checks whether a field exists.
Notes: Returns true if field is found or false if not.

5.14.27 FieldNames as String()

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns an array with all the field names for quick inspection.
Example:

```
Var cmd as SQLCommandMBS // your command

Var FieldNames() as String = cmd.FieldNames
```

5.14.28 Open

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Opens the specified command object.
Notes: Use the Open method to open the command explicitly.

A command will be implicitly opened by any method that needs an open command, therefore you don't have to call it explicitly.

To test whether a command is opened use isOpened method.

5.14.29 Param(ID as Integer) as SQLParamMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the command parameter specified by its position.
Notes: ID: A position of parameter specified in the command text. Normally position is a number stated in the command text after a colon (for example, 1 for :1, 5 for :5).

Returns a reference to a SAParam object which is only valid as long as the param object is not deleted by the library.

Use Param method to access a parameter by its name or position (in SQL statement). If, for example, you want to walk through all the parameters use ParamByIndex method.

If parameters were not created before calling Param method the Library creates them implicitly (can query native API if needed and therefore can throw exception on error) and then returns the specified parameter.

Passing a value of name or position which does not specified in the command text will throw an exception.

SAParam object contains full information about a parameter: name, type, size, etc. Values for the input parameters can be assigned to SAParam object.
See also:

- 5.14.30 Param(name as string) as SQLParamMBS 80

5.14.30 Param(name as string) as SQLParamMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the command parameter specified by its name.
Example:

```
// your connection
Var con as SQLConnectionMBS
Var SQL as string = "Insert into test_tbl(fid, fvarchar20) values(:id, :name)"

// create command object
Var cmd as new SQLCommandMBS(con, sql)

// assign values by name of parameter:
cmd.Param("id").setAsLong(2)
cmd.Param("name").setAsString(new SQLStringMBS("Some string (2)"))

// Insert first row
cmd.Execute
```

Notes: Name: A string that represents a name of the requested parameter. Normally name is a string stated in the command text after a colon (for example, 'city' for :city, 'my city' for :"my city") or a parameter name in a stored procedure or function.

Returns a reference to a SAParam object which is only valid as long as the param object is not deleted by the library.

Use Param method to access a parameter by its name or position (in SQL statement). If, for example, you want to walk through all the parameters use ParamByIndex method.

If parameters were not created before calling Param method the Library creates them implicitly (can query native API if needed and therefore can throw exception on error) and then returns the specified parameter.

Passing a value of name or position which does not specified in the command text will throw an exception.

SAParam object contains full information about a parameter: name, type, size, etc. Values for the input parameters can be assigned to SAParam object.
See also:

- 5.14.29 Param(ID as Integer) as SQLParamMBS 80

5.14.31 ParamByIndex(index as Integer) as SQLParamMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the command parameter specified by index.
Notes: Index: A zero-based index of the requested parameter in the array of SAParam objects. It must be greater than or equal to 0 and 1 less than the value returned by ParamCount method.

Returns a reference to a SAParam object.

Normally you should use Param method to access a parameter by its name or position (in SQL statement). ParamByIndex method can be used if, for example, you want to walk through all the parameters.

If parameters were not created before calling ParamByIndex method the Library creates them implicitly (can query native API if needed and therefore can throw exception on error) and then returns the specified parameter.

Passing a negative value of index or a value greater or equal than the value returned by ParamCount method will result in a failed assertion.

SAParam object contains full information about a parameter: name, type, size, etc. Values for the input parameters can be assigned to SAParam object.

Raises OutOfBoundsException exception if index parameter is out of range.

5.14.32 PostgreSQLField(RecordIndex as integer, FieldIndex as integer) as string

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries a field by index for the row with the RecordIndex.
See also:

- 5.14.33 PostgreSQLField(RecordIndex as integer, FieldName as string) as string 82

5.14.33 PostgreSQLField(RecordIndex as integer, FieldName as string) as string

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries a field by name for the row with the RecordIndex.
See also:

- 5.14.32 PostgreSQLField(RecordIndex as integer, FieldIndex as integer) as string 82

5.14.34 PostgreSQLFieldCount as Integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of fields in the result.

5.14.35 PostgreSQLRowCount as Integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of records in the result.

5.14.36 Prepare

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Prepares command before execution.
Notes: Prepare method compiles the command, but does not execute it. The method detects syntax errors in command text and verifies the existence of database objects.

Execute method calls Prepare method implicitly if needed, therefore you don't have to call it explicitly.

5.14.37 setCommandText(SQLCommand as string, CommandType as Integer = 0)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the command text.
Example:

```
Var s as new SQLCommandMBS

s.setCommandText "select * from test"

MsgBox s.CommandText
```

Notes: SQLCommand: A string which represents command text string (an SQL statement or a stored procedure name).
CommandType: The type of command like kCommandTypeUnknown, kCommandTypeSQLStatement, kCommandTypeSQLStatementRaw or kCommandTypeStoredProcedure.

It's not necessary to set a command type explicitly, because it is defined automatically in terms of command text string. But if you still have any reason to do it, use one of the kCommandType* constants. To get command type use CommandType method.

5.14.38 SetParameters(Params as dictionary)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the parameters based on the keys and values in the dictionary.
Example:

```
Var con as SQLConnectionMBS // your connection
Var pic as picture // some picture

// get picture data
Var jpegData as MemoryBlock = pic.GetData(Picture.FormatJPEG, 80)

// parse a SQL command
```

```
Var sql as string = "Insert into BlobTest(name, image) values (:name, :image)"
Var cmd as new SQLCommandMBS(con, sql)

Var d as new Dictionary
// set by param index
d.Value(0) = "test.jpg"
// set by param name
d.Value("image") = jpegData

// set all parameters together
cmd.SetParameters d
cmd.Execute
```

Notes: Keys can be String, Text or numeric types. Text and String are used to pick parameters by name.
Numeric values are used to pick parameter by index (zero based).
MemoryBlock and Strings without text encoding are converted to byte values (BLOB).
Texts and Strings with encoding are converted to text values.
Raises exceptions if you pass anything which is not recognized.
Other types are translated as good as possible.

Raises OutOfBoundsException exception if index parameter is out of range.

5.14.39 Value(index as Integer) as SQLValueReadMBS

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value specified by its position in the result set.
Example:

```
Var c as SQLCommandMBS // your command object

// get field by name
Var f1 as SQLValueReadMBS = c.Value("FirstName")

// get field by Index
Var f2 as SQLValueReadMBS = c.Value(1)
```

Notes: You can use Value() to get values for normal or cached result sets.

index: A one-based field number in a result set.

Use Value method to access a field by its name or position in the result set.
For Cached result sets, please use Value() function to get values.

Using an index smaller than 1 and greater then the value returned by FieldCount method will result in a failed assertion.

A set of SAField objects creates implicitly after the command execution if the result set exists. SAField object contains full information about a column: name, type, size, value.

Raises OutOfBoundsException exception if index parameter is out of range.
See also:

- 5.14.40 Value(name as string) as SQLValueReadMBS 85

5.14.40 Value(name as string) as SQLValueReadMBS

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value specified by its name in the result set.
Example:

```
Var c as SQLCommandMBS // your command object

// get value by name
Var f1 as SQLValueReadMBS = c.Value("FirstName")

// get value by Index
Var f2 as SQLValueReadMBS = c.Value(1)
```

Notes: You can use Value() to get values for normal or cached result sets.

name: A string that represents a name of the requested field.

Returns a reference to a SAValueRead object.

Use Value method to access a field by its name or position in the result set.
Using a non-existent field name will throw an exception.

A set of SAField objects creates implicitly after the command execution if the result set exists.SAField object contains full information about a column: name, type, size, value.
See also:

- 5.14.39 Value(index as Integer) as SQLValueReadMBS 84

5.14.41 Properties

5.14.42 CommandCount as Integer

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries number of current command objects.
Notes: This method should help you find leaked objects by keeping track of current count from the plugin perspective.
This includes SQLCommandMBS and RecordSet objects.
(Read only property)

5.14.43 CommandText as string

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the command text associated with the SACommand object.
Example:

```
Var s as new SQLCommandMBS(nil, "select * from test")

MsgBox s.CommandText
```

Notes: Use the CommandText method to return the command text declared in SACommand constructor or setCommandText method.

All text strings sent to the plugin must have a defined encoding. Else the internal text encoding conversions will fail.
(Read and Write property)

5.14.44 CommandType as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the command type currently associated with the SACommand object.
Notes: One of the following values from SACommandType_t enum:

- kCommandTypeUnknown Command type is not defined. Library will detect command type automatically when needed.
- kCommandTypeSQLStmt Command is an SQL statement.
- kCommandTypeSQLStmtRaw Command is an SQL statement that mustn't be interpreted by SQLAPI++.

- kCommandTypeStoredProc Command is a stored procedure or a function.

Remarks

The command type can be explicitly set in SACommand constructor and setCommandText method, but it's not necessary to do it.

The CommandTypemethod returns the command type value that was specified in SACommand constructor or setCommandText method. If you declared the command type value as kCommandTypeUnknown (the default value) then command type is detected by the Library and the CommandType method returns this detected value.
(Read only property)

5.14.45 Connection as SQLConnectionMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The connection for the command.
Notes: When you set the connection on a command object that already has associated connection, the previous association will be correctly discarded (with closing opened command if needed) and new connection will be set.

If you attempt to call any method on a SACommand object that requires database access with no valid connection, an error occurs.
(Read and Write property)

5.14.46 FieldCount as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of fields (columns) in a result set.
Notes: FieldCount method returns the number of fields created implicitly after the command execution if a result set exists.

A field is represented by SAField object. You can get field value and description using Field method.
(Read only property)

5.14.47 Fields as Dictionary

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Provides dictionary with all fields.
Notes: This dictionary should help for debugging to inspect all fields and their text value.
(Read only property)

5.14.48 hasCache as Boolean

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether cache is active.
Notes: (Read only property)

5.14.49 isBOF as Boolean

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether we are at the beginning of the result set.
Notes: Set to true when running the query and when FetchFirst succeeds.
Otherwise false.
(Read only property)

5.14.50 isEOF as Boolean

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether we are at the end of the result set.
Notes: True if we are at the end of the recordset, e.g. FetchNext failed.
Set to false by Execute.

We set it to true, when something goes wrong, so Xojo's RecordSet and RowSet end looping.
We can only know, that we are not at the last one, if FetchNext internally failed to get the next record.
(Read only property)

5.14.51 isExecuted as boolean

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this command was already executed.
Notes: (Read only property)

5.14.52 isExecuting as Boolean

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this command is executing.
Notes: You only see this true if you use threaded queries and look on the property from another thread.
(Read only property)

5.14.53 isOpened as boolean

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns true if the SACommand object is opened; otherwise false.
Notes: (Read only property)

5.14.54 isResultSet as boolean

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Tests whether a result set exists after the command execution.
Notes: Returns true if the result set exists; otherwise false.
(Read only property)

5.14.55 Options as Dictionary

Plugin Version: 18.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a dictionary with all options.
Notes: For debugging, it may be useful to inspect options in debugger.
(Read only property)

5.14.56 ParamCount as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of parameters associated with the SACommand object.
Notes: ParamCount method returns the number of parameters created explicitly by using CreateParam method or (if parameters were not created before) creates them implicitly (can query native API if needed and therefore can throw exception on error) and returns the number of created parameters.

Command parameter is represented by SAParam object. You can look SAParam objects through and assign their values with Param and ParamByIndex methods.
(Read only property)

5.14.57 Parameters as Dictionary

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Provides dictionary with all parameters.
Notes: This dictionary should help for debugging to inspect all parameters and their text value.
(Read only property)

5.14.58 RowsAffected as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of rows affected by the last insert/update/delete command execution.
Example:

```
Var con as SQLConnectionMBS // your connection

Var sql as string = "UPDATE Test SET MyField=1"
Var c as new SQLCommandMBS(con, sql)

c.Execute

MsgBox str(c.RowsAffected)
```

Notes: (Read only property)

5.14.59 Tag as Variant

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The tag property.
Example:

```
Var c as SQLCommandMBS // your command object

// store reference to window/control, so we have it available in events
c.Tag = self
```

Notes: You can store here whatever you like.
(Read and Write property)

5.14.60 Option(name as string) as string

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: A string value of a specific command option.
Example:

```
Var cmd as SQLCommandMBS // your command

// turn on auto cache
cmd.Option("AutoCache") = "true"
```

Notes: see also:
https://www.sqlapi.com/ApiDoc/class_s_a_command.html
(Read and Write computed property)

5.14.61 Events

5.14.62 Trace(traceInfo as Integer, SQL as string)

Plugin Version: 13.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The event to trace SQL commands.

5.14.63 Working

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The event called while the ExecuteMT and ExecuteCommandMT methods are running.

5.14.64 Constants

Constants

Constant	Value	Description
kOptionPreFetchRows	"PreFetchRows"	One of the option constants.
kParamDirTypeInput	0	One of the parameter direction type constants. Input parameter.
kParamDirTypeInputOutput	1	One of the parameter direction type constants. Input/output parameter.
kParamDirTypeOutput	2	One of the parameter direction type constants. Output parameter.
kParamDirTypeReturn	3	One of the parameter direction type constants. Returning parameter.

Command Types

Constant	Value	Description
kCommandTypeSQLStatement	1	Command is an SQL statement.
kCommandTypeSQLStatementRaw	2	Command is an SQL statement that mustn't be interpreted by SQLAPI.
kCommandTypeStoredProcedure	3	Command is a stored procedure or a function.
kCommandTypeUnknown	0	Used by default. Library detects command type automatically.

5.15 class SQLConnectionMBS

5.15.1 class SQLConnectionMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for a SQL Plugin Database connection.
Example:

```
Var con as new SQLConnectionMBS

try

// where is the library?
con.SetFileOption con.kOptionLibraryMySQL, SpecialFolder.UserHome.Child("libmysqlclient.dylib")

// connect to database
// in this example it is Oracle,
// but can also be Sybase, Informix, DB2
// SQLServer, InterBase, SQLBase and ODBC

Var server as string = "192.168.1.80,3306@test"

con.Connect(server,"root","",SQLConnectionMBS.kMySQLClient)

MsgBox "We are connected!"

// Disconnect is optional
// autodisconnect will ocur in destructor if needed
con.Disconnect

msgbox "We are disconnected!"

catch r as RuntimeException
MsgBox r.message

// SAConnection::Rollback()
// can also throw an exception
// (if a network error for example),
// we will be ready
try

// on error rollback changes
con.Rollback

catch rr as runtimeexception
MsgBox rr.message
end try
```

end try

Notes: Supported databases: CubeSQL, Centura SQLBase, DB2, DuckDB, Firebird, Informix, InterBase, MariaDB, Microsoft Access, Microsoft SQL Server, MySQL, ODBC, Oracle Database Server, PostgreSQL, SQL Anywhere, SQLite, SQLCipher and Sybase.
Connect to Microsoft Access, FileMaker Server (or Pro), Microsoft Visual FoxPro and others via ODBC.

With Xojo 2013r1, you only need a database server license from Xojo, Inc. if you use the SQLDatabaseMBS class. The SQLConnectionMBS class does not require this license. But some features like getting a recordset do need the license as they refer to the SQLDatabaseMBS class.

Please free all RecordSets and SQLCommand objects before you close the SQLConnection or the SQLDatabase. The plugin keeps references from RecordSets and SQLCommand to prevent automatic destruction of the database connection. If you close a database connection while you have RecordSets and SQLCommand in use, things may go wrong.

The plugin can cache the recordset locally. To enable you can call SQLCommandMBS.Cache or use the Option("AutoCache") = "true" on either command or connection or database objects. The plugin will than fetch all records and store them in memory. After this you can walk over the recordset and use FetchPos, FetchFirst, FetchLast, FetchPrev and FetchNext to locate the rows you need. When you call Field() you always get last row, but to read from cached result set, please use Value() function. When using RecordSet, the values are read via Value() functions automatically.

You can use InternalPostgreSQLLibraryMBS or InternalSQLiteLibraryMBS if you like to use our built in SQLite or PostgreSQL database libraries.

see also
https://www.sqlapi.com/ApiDoc/class_s_a_connection.html

The class pings the database every minute by checking whether it's alive and to avoid server dropping connection. This can be disabled by setting Option("Ping") = "false". Ping is not used for SQLite.

MBS Database connections are implemented via SQLConnectionMBS and SQLCommandMBS classes. We provide a thin layer on top with SQLDatabaseMBS class to make it compatible to the Xojo database class. And when you use SQLDatabaseMBS, you can always get the matching SQLConnectionMBS objet via Connection property. Instead of SQLCommandMBS class, you may just use SelectSQL/ExecuteSQL or older SQLSelect/SQLExecute functions.

We have a collection of library files here:
https://www.monkeybreadsoftware.de/xojo/download/plugin/Libs/
Blog Entries

- MonkeyBread Software Releases the MBS Xojo Plugins in version 25.4
- Connect to Postgres in Xojo
- News from the MBS Xojo Plugins Version 23.5
- Using MBS SQL Plugin with PostgreSQL
- Edit and Update for SQLDatabaseMBS class
- Crossplatform connection to Microsoft SQL Server in Xojo
- CubeSQL support for MBS Xojo SQL Plugin
- Connecting to Microsoft Visual FoxPro
- Multithreaded plugin functions can increase speed of Real Studio application
- Accessing Microsoft SQL Database from Mac/Linux

Videos

- MBS SQL Plugin Presentation
- Presentation from Munich conference about MBS Plugins.

Xojo Developer Magazine

- 22.1, page 9: News
- 16.1, page 10: News
- 14.1, pages 24 to 30: The MBS SQL Plugin, An alternative way to connect to databases by Christian Schmitz
- 12.4, page 9: News

5.15.2 Methods

5.15.3 BeginTransaction

Plugin Version: 22.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Begins a transaction.
Notes: This method does nothing. Why?
Well, if auto commit is on, you don't need to call this.
If auto commit is off, we call BeginTransaction for you, so if you call this method you would call it a second time and get an error.
We have this method for compatibility to other SQL database classes from Xojo.

5.15.4 CancelAllCommands

Plugin Version: 18.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Cancel all commands for the connection.
Notes: This loops over the list of commands associated with this connection and calls Cancel on them.

5.15.5 Commands as SQLCommandMBS()

Plugin Version: 18.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries list of all command objects related to the connection.

5.15.6 Commit

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Saves any changes and ends the current transaction.
Notes: Use Commit method to write transaction changes permanently to a database. It commits the work of all commands that associated with that connection.

All changes to the database since the last commit are made permanent and cannot be undone. Before a commit, all changes made since the start of the transaction can be rolled back using Rollback method.

5.15.7 Connect(DBString as string, UserID as string, Password as string, client as Integer = 0)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Opens the connection to a data source.
Example:

```
Var con as SQLConnectionMBS// your connection

// some calls for MS SQL Server:
con.Connect("srv2@pubs","", "", SQLConnectionMBS.kSQLServerClient)
con.Connect("@pubs", "", "", SQLConnectionMBS.kSQLServerClient)
con.Connect("BEDLAM\SQL2005EX_EN@pubs", "", "", SQLConnectionMBS.kSQLServerClient)
con.Connect("BEDLAM\SQLEXPRESS@master", "", "", SQLConnectionMBS.kSQLServerClient)

// for MySQL:

con.Connect("192.168.1.80,3306@test","root","password", SQLConnectionMBS.kMySQLClient)
```

```
// for Postgre SQL:

con.Connect("somedb", "name", "password",SQLConnectionMBS.kPostgreSQLClient)
// with options
con.Connect("127.0.0.1,5432@dbname=postgres connect_timeout=10 sslmode=require", "name", "password",SQL-
ConnectionMBS.kPostgreSQLClient)

// for SQLite:

con.Connect("/test.db", "", "",SQLConnectionMBS.kSQLiteClient)
```

Notes:

DBString:	Name of database this connection will connect to (see Server specific notes).
UserID:	A string containing a user name to use when establishing the connection (see Server specific notes).
Password:	A string containing a password to use when establishing the connection.
client:	Optional. One of the following values from k*Client constants.

Using the Connect method on a SAConnection object establishes the physical connection to a data source. After this method successfully completes, the connection is live and you can issue commands against it and process the results.

If you use the default value of Client parameter, you should set Client before using Connect.

To check whether a connection established use isConnected method. To check whether a connection is broken or not use isAlive method.

see also for server specific notes:
https://www.sqlapi.com/ApiDoc/class_s_a_connection.html

For IPv6 we changed plugin to use , instead of : for the port separator. So please use , to separate port from IP or host.

For Firebird, if you connect to a database and you have 32/64bit mismatch, you get error number 3.

See Option() for various options you can set before connecting.
e.g. c.Option("SQLiteVFSFlags") = "1" for SQLite for read only access.

Raises OutOfBoundsException exception if client parameter is out of range.

5.15.8 ConnectMT(DBString as string, UserID as string, Password as string, client as Integer = 0)

Plugin Version: 15.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Opens the connection to a data source.
Notes:

DBString:	Name of database this connection will connect to (see Server specific notes).
UserID:	A string containing a user name to use when establishing the connection (see Server specific notes).
Password:	A string containing a password to use when establishing the connection.
client:	Optional. One of the following values from k*Client constants.

Using the Connect method on a SAConnection object establishes the physical connection to a data source. After this method successfully completes, the connection is live and you can issue commands against it and process the results.

If you use the default value of Client parameter, you should set Client before using Connect.

To check whether a connection established use isConnected method. To check whether a connection is broken or not use isAlive method.

see also for server specific notes:
https://www.sqlapi.com/ApiDoc/class_s_a_connection.html

For IPv6 we changed plugin to use , instead of : for the port separator. So please use , to separate port from IP or host.

The work is performed on a preemptive thread, so this function does not block the application and can yield time to other Xojo threads. Must be called in a Xojo thread to enjoy benefits. If called in main thread will block, but keep other background threads running.

The MT method will not trigger WillConnect and DidConnect events.
Raises OutOfBoundsException exception if client parameter is out of range.

5.15.9 CubeSQLLastInsertID as Int64

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the last auto increment value from last insert command.

Notes: Only for CubeSQL connections. May raise error if not available.

5.15.10 CubeSQLReceiveData(byref data as String, byref IsEndChunk as Boolean) as Boolean

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Receives a data chunk for file download.
Notes: Returns true on success.
Data is set with data and IsEndChunk is set to true for last chunk.

5.15.11 CubeSQLSendData(data as MemoryBlock)

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sends data chunk for file upload.
Notes: This is the sendchunk function in CubeSQL.
See also:

- 5.15.12 CubeSQLSendData(data as String) 99

5.15.12 CubeSQLSendData(data as String)

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sends data chunk for file upload.
Notes: This is the sendchunk function in CubeSQL.
See also:

- 5.15.11 CubeSQLSendData(data as MemoryBlock) 99

5.15.13 CubeSQLSendEndData

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sends end data packet.
Notes: This is the send_enddata function in CubeSQL.

5.15.14 Disconnect

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Disconnects the connection from the database.
Notes: Closes all commands objects and RecordSets.

5.15.15 InsertRecord(TableName as String, Record as Dictionary)

Plugin Version: 17.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Convenience function to insert a record.
Example:

```
Var con as SQLConnectionMBS // your database connection

Var d as new Dictionary

d.Value("ID")=2
d.Value("text")="test insert"
d.Value("other")="Just a test"

con.InsertRecord("test_tbl", d)
```

Notes: The plugin builds for you SQL statement with prepared statement and runs the insert command with values.
Lasterror is set or exception raised as with SQLExecute.

Internally this uses a prepared statement. You can check the generated statement via LastStatement property.

5.15.16 kOptionLibrarySeparator as String

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: One of the option constant to specify the platform specific path separator.
Notes: Use with Option() to specify multiple file paths for a library.
Has a different value on the different platforms.
Value is ";" on Windows and ":" on macOS/Linux.

5.15.17 Listen

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Start listening for notifications.
Notes: Works only for PostgresSQL Client.

Please set client or connect before calling this method.

5.15.18 MySQLInsertID as Int64

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the last auto increment value from last insert command.
Notes: Please query value right after doing Insert. This value is reset when you call commit.
For mySQL and MariaDB connections.

see also
http://dev.mysql.com/doc/refman/5.1/en/mysql-insert-id.html

5.15.19 Rollback

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Cancels any changes made during the current transaction and ends the transaction.
Notes: Rollback method rolls back the database to the state it was in at the completion of the last commit operation. All uncommitted work is undone.

Rollback method rolls back the work of all commands that associated with that connection.

To commit all changes made since the start of the transaction use Commit method.

5.15.20 SetFileOption(name as string, file as folderitem)

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets an option with passing a file path.
Example:

```
Var db as new SQLConnectionMBS

// where is the library?
db.SetFileOption SQLConnectionMBS.kOptionLibraryMySQL, SpecialFolder.UserHome.Child("libmysqlclient.dylib")
```

Notes: Allows you to specify a file path with a folderitem.

Makes sure the path is correct and you have a 32 or 64-bit library matching the architecture of your appli-

cation.

5.15.21 SQLExecute(command as string, CommandType as Integer = 0)

Plugin Version: 10.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes a SQL command and ignores result.
Notes: This is a convenience function.
Internally it creates a SQLCommandMBS with the given command and calls Execute.

All text strings sent to the plugin must have a defined encoding. Else the internal text encoding conversions will fail.

5.15.22 SQLExecuteMT(command as string, CommandType as Integer = 0)

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes a SQL command and ignores result.
Notes: This is a convenience function.
Internally it creates a SQLCommandMBS with the given command and calls Execute.

All text strings sent to the plugin must have a defined encoding. Else the internal text encoding conversions will fail.

The work is performed on a preemptive thread, so this function does not block the application and can yield time to other Xojo threads. Must be called in a Xojo thread to enjoy benefits. If called in main thread will block, but keep other background threads running.

5.15.23 SQLiteBackupFinish(Backup as SQLite3BackupMBS) as integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Finishes a backup run.
Notes: When BackupStep has returned kErrorDone, or when the application wishes to abandon the backup operation, the application should destroy the SQLite3BackupMBS by passing it to BackupFinish. The BackupFinish interfaces releases all resources associated with the SQLite3BackupMBS object. If BackupStep has not yet returned kErrorDone, then any active write-transaction on the destination database is rolled back. The SQLite3BackupMBS object is invalid and may not be used following a call to BackupFinish.
The value returned by BackupFinish is kErrorOK if no BackupStep errors occurred, regardless or whether or not BackupStep completed. If an out-of-memory condition or IO error occurred during any prior BackupStep call on the same SQLite3BackupMBS object, then BackupFinish returns the corresponding error code.

A return of kErrorBusy or kErrorLocked from BackupStep is not a permanent error and does not affect the return value of BackupFinish.

5.15.24 SQLiteBackupInit(Dest as Variant, DestName as String, Source as Variant, SourceName as String) as SQLite3BackupMBS

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Initializes a backup.
Notes: The backup API copies the content of one database into another. It is useful either for creating backups of databases or for copying in-memory databases to or from persistent files.

see also
http://www.sqlite.org/c3ref/backup_finish.html

Exclusive access is required to the destination database for the duration of the operation. However the source database is only read-locked while it is actually being read; it is not locked continuously for the entire backup operation. Thus, the backup may be performed on a live source database without preventing other users from reading or writing to the source database while the backup is underway.

To perform a backup operation:

- BackupInit is called once to initialize the backup,
- BackupStep is called one or more times to transfer the data between the two databases, and finally
- BackupFinish is called to release all resources associated with the backup operation.

There should be exactly one call to BackupFinish for each successful call to BackupInit.

The D and N arguments to BackupInit(D,N,S,M) are the database connection associated with the destination database and the database name, respectively. The database name is ”main” for the main database, ”temp” for the temporary database, or the name specified after the AS keyword in an ATTACH statement for an attached database. The S and M arguments passed to BackupInit(D,N,S,M) identify the database connection and database name of the source database, respectively. The source and destination database connections (parameters S and D) must be different or else BackupInit(D,N,S,M) will file with an error.
If an error occurs within BackupInit(D,N,S,M), then nil is returned and an error code and error message are store3d in the destination database connection D. The error code and message for the failed call to BackupInit can be retrieved using the ErrCode and ErrMessage functions. A successful call to BackupInit returns a SQLite3BackupMBS object. The SQLite3BackupMBS object may be used with the BackupStep and BackupFinish functions to perform the specified backup operation.

Concurrent Usage of Database Handles
The source database connection may be used by the application for other purposes while a backup operation is underway or being initialized. If SQLite is compiled and configured to support threadsafe database connections, then the source database connection may be used concurrently from within other threads.
However, the application must guarantee that the destination database connection is not passed to any other API (by any thread) after BackupInit is called and before the corresponding call to BackupFinish. SQLite does not currently check to see if the application incorrectly accesses the destination database connection and so no error code is reported, but the operations may malfunction nevertheless. Use of the destination database connection while a backup is in progress might also also cause a mutex deadlock.
If running in shared cache mode, the application must guarantee that the shared cache used by the destination database is not accessed while the backup is running. In practice this means that the application must guarantee that the disk file being backed up to is not accessed by any connection within the process, not just the specific connection that was passed to BackupInit.
The SQLite3BackupMBS object itself is partially threadsafe. Multiple threads may safely make multiple concurrent calls to BackupStep. However, the BackupRemaining and BackupPageCount APIs are not strictly speaking threadsafe. If they are invoked at the same time as another thread is invoking BackupStep it is possible that they return invalid values.

Source and Dest can be SQLConnectionMBS or SQLDatabaseMBS. You need to pass source and dest, even if one is self as we give you the option to decide where to pass the current database connection.

5.15.25 SQLiteBackupPageCount(Backup as SQLite3BackupMBS) as integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of pages in total.
Notes: Each call to BackupStep sets two values inside the SQLite3BackupMBS object: the number of pages still to be backed up and the total number of pages in the source databae file. The BackupRemaining and BackupPageCount interfaces retrieve these two values, respectively.
The values returned by these functions are only updated by BackupStep. If the source database is modified during a backup operation, then the values are not updated to account for any extra pages that need to be updated or the size of the source database file changing.

5.15.26 SQLiteBackupRemaining(Backup as SQLite3BackupMBS) as integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of pages remaining.
Notes: Each call to BackupStep sets two values inside the SQLite3BackupMBS object: the number of pages still to be backed up and the total number of pages in the source databae file. The BackupRemaining and BackupPageCount interfaces retrieve these two values, respectively.
The values returned by these functions are only updated by BackupStep. If the source database is modified during a backup operation, then the values are not updated to account for any extra pages that need to be updated or the size of the source database file changing.

5.15.27 SQLiteBackupStep(Backup as SQLite3BackupMBS, Pages as Integer) as integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies up to Pages pages between the source and destination databases specified by SQLite3BackupMBS object.
Notes: If N is negative, all remaining source pages are copied. If BackupStep(B,N) successfully copies N pages and there are still more pages to be copied, then the function resturns kErrorOK. If BackupStep(B,N) successfully finishes copying all pages from source to destination, then it returns kErrorDone. If an error occurs while running BackupStep(B,N), then an error code is returned. As well as kErrorOK and kErrorDone, a call to BackupStep may return kErrorReadOnly, kErrorNoMem, kErrorBusy, kErrorLocked, or an kErrorIOACCESS | kErrorIOXXX extended error code.
The BackupStep might return kErrorReadOnly if the destination database was opened read-only or if the destination is an in-memory database with a different page size from the source database.
If BackupStep cannot obtain a required file-system lock, then the sqlite3_busy_handler | busy-handler function is invoked (if one is specified). If the busy-handler returns non-zero before the lock is available, then kErrorBusy is returned to the caller. In this case the call to BackupStep can be retried later. If the source database connection is being used to write to the source database when BackupStep is called, then kErrorLocked is returned immediately. Again, in this case the call to BackupStep can be retried later on. (If kErrorIOACCESS | kErrorIOXXX, kErrorNoMem, or kErrorReadOnly is returned, then there is no point in retrying the call to BackupStep. These errors are considered fatal.) The application must accept that the backup operation has failed and pass the backup operation handle to the BackupFinish to release associated resources.
The first call to BackupStep obtains an exclusive lock on the destination file. The exclusive lock is not released until either BackupFinish is called or the backup operation is complete and BackupStep returns kErrorDone. Every call to BackupStep obtains a shared lock on the source database that lasts for the duration of the BackupStep call. Because the source database is not locked between calls to BackupStep, the source database may be modified mid-way through the backup process. If the source database is modified by an external process or via a database connection other than the one being used by the backup operation, then the backup will be automatically restarted by the next call to BackupStep. If the source database is modified by the using the same database connection as is used by the backup operation, then the backup database is automatically updated at the same time.

5.15.28 SQLiteConnectionHandle as Ptr

Plugin Version: 22.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the current connection reference for the database.
Notes: sqlite3 pointer for using in declares.

5.15.29 SQLiteEnableLoadExtension(OnOff as boolean)

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Enables/disables extension loading for the given connection.

5.15.30 SQLiteLastInsertRowID as Int64

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns Last Insert Rowid.
Notes: Each entry in an SQLite table has a unique 64-bit signed integer key called the ROWID. The rowid is always available as an undeclared column named ROWID, OID, or _ROWID_ as long as those names are not also used by explicitly declared columns. If the table has a column of type INTEGER PRIMARY KEY then that column is another alias for the rowid.

This routine returns the rowid of the most recent successful INSERT into the database from the database connection in the first argument. If no successful INSERTs have ever occurred on that database connection, zero is returned.

(If an INSERT occurs within a trigger, then the rowid of the inserted row is returned by this routine as long as the trigger is running. But once the trigger terminates, the value returned by this routine reverts to the last value inserted before the trigger fired.)

An INSERT that fails due to a constraint violation is not a successful INSERT and does not change the value returned by this routine. ^Thus INSERT OR FAIL, INSERT OR IGNORE, INSERT OR ROLLBACK, and INSERT OR ABORT make no changes to the return value of this routine when their insertion fails. ^(When INSERT OR REPLACE encounters a constraint violation, it does not fail. The INSERT continues to completion after deleting rows that caused the constraint problem so INSERT OR REPLACE will always change the return value of this interface.)^

For the purposes of this routine, an INSERT is considered to be successful even if it is subsequently rolled back.

This function is accessible to SQL statements via the last_insert_rowid() SQL function.

If a separate thread performs a new INSERT on the same database connection while the LastInsertRowID function is running and thus changes the last insert rowid, then the value returned by LastInsertRowID is unpredictable and might not equal either the old or the new last insert rowid.

5.15.31 SQLiteLibVersion as String

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the version string of the SQLite library.

5.15.32 SQLiteLoadExtension(file as FolderItem, ByRef ErrorMessage as String) as Integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads an SQLite extension library from the named file.
Notes: The LoadExtension interface attempts to load an SQLite extension library contained in the file.

Returns kErrorOk on success and kErrorError if something goes wrong.
Extension loading must be enabled using EnableLoadExtension prior to calling this API, otherwise an error will be returned.
See also:

- 5.15.33 SQLiteLoadExtension(path as String, ByRef ErrorMessage as String) as Integer 107

5.15.33 SQLiteLoadExtension(path as String, ByRef ErrorMessage as String) as Integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads an SQLite extension library from the named file.
Notes: The LoadExtension interface attempts to load an SQLite extension library contained in the file.

Returns kErrorOk on success and kErrorError if something goes wrong.
Extension loading must be enabled using EnableLoadExtension prior to calling this API, otherwise an error will be returned.
See also:

- 5.15.32 SQLiteLoadExtension(file as FolderItem, ByRef ErrorMessage as String) as Integer 107

5.15.34 SQLiteMemoryHighwater(reset as boolean = false) as Int64

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries maximum memory usage so far.
Notes: Can be reset with reset parameter being true.

5.15.35 SQLiteMemoryUsed as Int64

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries memory in use by SQLite.
Notes: This is memory allocated, but not yet freed.
Value is zero until SQLite3 initialized.

5.15.36 SQLiteReKey(Key as String) as Integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: You can change the key on a database using the Rekey Function.
Notes: An empty key decrypts the database.

Rekeying requires that every page of the database file be read, decrypted, reencrypted with the new key, then written out again. Consequently, rekeying can take a long time on a larger database.

Most SEE variants allow you to encrypt an existing database that was created using the public domain version of SQLite. This is not possible when using the authenticating version of the encryption extension in see-aes128-ccm. If you do encrypt a database that was created with the public domain version of SQLite, no nonce will be used and the file will be vulnerable to a chosen-plaintext attach. If you call SetKey() immediately after Open when you are first creating the database, space will be reserved in the database for a nonce and the encryption will be much stronger. If you do not want to encrypt right away, call SetKey() anyway, with an empty key, and the space for the nonce will be reserved in the database even though no encryption is done initially.

A public domain version of the SQLite library can read and write an encrypted database with an empty key. You only need the encryption extension if the key is non-empty.

Returns a SQLite error code.

5.15.37 SQLiteSetBusyHandler(MaxAttempts as Integer = 5)

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Installs busy handler for this connection.
Notes: This routine sets a callback function that might be invoked whenever an attempt is made to open a database table that another thread or process has locked.

The plugin has an busy handler which will wait up to MaxAttemps and yield to other Xojo threads while waiting.

Passing 5 should wait up to 100ms.

There can only be a single busy handler defined for each [database connection] . Setting a new busy handler clears any previously set handler.) Note that calling SetBusyTimeout will also set or clear the busy handler.

The busy callback should not take any actions which modify the database connection that invoked the busy handler. Any such actions result in undefined behavior.

5.15.38 SQLiteSetBusyTimeout(TimeOutMS as Integer = 20)

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: This routine sets a busy handler that sleeps for a specified amount of time when a table is locked.
Notes: The handler will sleep multiple times until at least "ms" milliseconds of sleeping have accumulated. ^After at least "ms" milliseconds of sleeping, the handler returns 0 which causes SQLite query to return SQLite Busy or IO Blocked error.

Calling this routine with an argument less than or equal to zero turns off all busy handlers.

(There can only be a single busy handler for a particular database connection any any given moment. If another busy handler was defined (using SetBusyHandler prior to calling this routine, that other busy handler is cleared.)

5.15.39 SQLiteSetKey(Key as String) as Integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Applies encryption to a database connection.
Notes: Returns a SQLite error code.

The amount of key material actually used by the encryption extension depends on which variant of SEE you are using. With RC4, the first 256 byte of key are used. With the AES128, the first 16 bytes of the key are used. With AES256, the first 32 bytes of key are used.

If you specify a key that is shorter than the maximum key length, then the key material is repeated as many times as necessary to complete the key. If you specify a key that is larger than the maximum key length, then the excess key material is silently ignored.

The key must begin with an ASCII prefix to specify which algorithm to use. The prefix must be one of "rc4:", "aes128:", or "aes256:". The prefix is not used as part of the key sent into the encryption algorithm.

So the real key should begin on the first byte after the prefix.

The string provided to the plugin is used with it's current encoding. So be sure you use right text encoding for what you want. e.g. using "M√ºller" as key in text encoding Windows ANSI will not open a database which used that key in UTF-8 encoding.

The Xojo database encryption in SQLiteDatabase class uses AES-128 OFB.

5.15.40 SQLiteTableColumnMetaData(DBName as string, TableName as string, ColumnName as string, byref DataType as string, byref CollationSequence as string, byref NotNull as boolean, byref PrimaryKey as boolean, byref AutoIncrement as Boolean) as integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Extract Metadata About A Column Of A Table
Notes: Not available in all sqlite libraries!

This routine returns metadata about a specific column of a specific database table accessible using the database connection handle passed as the first function argument.
The column is identified by the second, third and fourth parameters to this function. The second parameter is either the name of the database (i.e. "main", "temp", or an attached database) containing the specified table or NULL. If it is NULL, then all attached databases are searched for the table using the same algorithm used by the database engine to resolve unqualified table references.
The third and fourth parameters to this function are the table and column name of the desired column, respectively. Neither of these parameters may be NULL.
Metadata is returned by writing to the memory locations passed as the 5th and subsequent parameters to this function. Any of these arguments may be NULL, in which case the corresponding element of metadata is omitted.

CollationSequence is assigned the Name of default collation sequence. NotNull is set to true if column has a NOT NULL constraint. PrimaryKey is set to true if column is part of the PRIMARY KEY and AutoIncrement is set to true if column is AUTOINCREMENT.

If the specified table is actually a view, an error code is returned.

If the specified column is "rowid", "oid" or "_rowid_" and an INTEGER PRIMARY KEY column has been explicitly declared, then the output parameters are set for the explicitly declared column. (If there is no explicitly declared INTEGER PRIMARY KEY column, then the output parameters are set as follows:

data type: "INTEGER"
collation sequence: "BINARY"

not null: false
primary key: true
auto increment: false

(This function may load one or more schemas from database files. If an error occurs during this process, or if the requested table or column cannot be found, an error code is returned and an error message left in the database connection (to be retrieved using ErrMessage).)

This API is only available if the library was compiled with the SQLITE_ENABLE_COLUMN_METADATA C-preprocessor symbol defined.

5.15.41 SQLiteThreadsafe as integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Test To See If The Library Is Threadsafe.
Notes: The threadsafe() function returns zero if and only if SQLite was compiled mutexing code omitted due to the SQLITE_THREADSAFE compile-time option being set to 0.

SQLite can be compiled with or without mutexes. When the SQLITE_THREADSAFE C preprocessor macro is 1 or 2, mutexes are enabled and SQLite is threadsafe. When the SQLITE_THREADSAFE macro is 0, the mutexes are omitted. Without the mutexes, it is not safe to use SQLite concurrently from more than one thread.

Enabling mutexes incurs a measurable performance penalty. So if speed is of utmost importance, it makes sense to disable the mutexes. But for maximum safety, mutexes should be enabled. The default behavior is for mutexes to be enabled.

This interface can be used by an application to make sure that the version of SQLite that it is linking against was compiled with the desired setting of the SQLITE_THREADSAFE macro.

This interface only reports on the compile-time mutex setting of the SQLITE_THREADSAFE flag. If SQLite is compiled with SQLITE_THREADSAFE=1 or =2 then mutexes are enabled by default but can be fully or partially disabled using a call to sqlite3_config() with the verbs SQLITE_CONFIG_SINGLETHREAD, SQLITE_CONFIG_MULTITHREAD, or SQLITE_CONFIG_MUTEX. ^(The return value of the sqlite3_threadsafe() function shows only the compile-time setting of thread safety, not any run-time changes to that setting made by sqlite3_config(). In other words, the return value from sqlite3_threadsafe() is unchanged by calls to sqlite3_config().)^

See the threading mode documentation for additional information.

5.15.42 SQLSelect(command as string, CommandType as Integer = 0) as string

Plugin Version: 10.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes a SQL command and returns the first field's string value.
Notes: This is a convenience function.
Internally it creates a SQLCommandMBS with the given command and calls Execute.
If the result is a record set, the first field from the first row is returned.
This is basicly useful for commands like "select sqlite_version()".

All text strings sent to the plugin must have a defined encoding. Else the internal text encoding conversions will fail.

5.15.43 SQLSelectAsRecordSet(command as string, CommandType as Integer = 0) as RecordSet

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes a SQL command and returns the result as RecordSet object.
Notes: This is a convenience function.
Internally it creates a SQLCommandMBS with the given command and calls Execute.

For this method to work, you need to have somewhere a property with SQLDatabaseMBS so Xojo includes our SQLDatabase plugin which provides the RecordSet functionality.

If Scrollable property is true, the recordset will be requested to be scrollable.

The record set may not have a valid RecordCount or have working movefirst/movelast/moveprev methods unless the underlaying database supports those and Scrollable result sets is enabled/supported.

5.15.44 SQLSelectAsRecordSetMT(command as string, CommandType as Integer = 0) as RecordSet

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes a SQL command and returns the result as RecordSet object.
Notes: This is a convenience function.
Internally it creates a SQLCommandMBS with the given command and calls Execute.

For this method to work, you need to have somewhere a property with SQLDatabaseMBS so Xojo includes our SQLDatabase plugin which provides the RecordSet functionality.

If Scrollable property is true, the recordset will be requested to be scrollable.

The record set may not have a valid RecordCount or have working movefirst/movelast/moveprev methods unless the underlaying database supports those and Scrollable result sets is enabled/supported.

The work is performed on a preemptive thread, so this function does not block the application and can yield time to other Xojo threads. Must be called in a Xojo thread to enjoy benefits. If called in main thread will block, but keep other background threads running.

5.15.45 SQLSelectAsRowSet(command as string, CommandType as integer = 0) as RowSet

Plugin Version: 21.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes a SQL command and returns the result as RowSet object.
Notes: This is a convenience function.
Internally it creates a SQLCommandMBS with the given command and calls Execute.

For this method to work, you need to have somewhere a property with SQLDatabaseMBS so Xojo includes our SQLDatabase plugin which provides the RowSet functionality.

If Scrollable property is true, the RowSet will be requested to be scrollable.

The RowSet may not have a valid RecordCount or have working movefirst/movelast/moveprev methods unless the underlaying database supports those and Scrollable result sets is enabled/supported.

For Xojo 2019r2 and newer. See SQLSelectAsRecordSet for older versions.

5.15.46 SQLSelectAsRowSetMT(command as string, CommandType as integer = 0) as RowSet

Plugin Version: 21.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes a SQL command and returns the result as RowSet object.
Notes: This is a convenience function.
Internally it creates a SQLCommandMBS with the given command and calls Execute.

For this method to work, you need to have somewhere a property with SQLDatabaseMBS so Xojo includes our SQLDatabase plugin which provides the RowSet functionality.

If Scrollable property is true, the RowSet will be requested to be scrollable.

The RowSet may not have a valid RecordCount or have working movefirst/movelast/moveprev methods unless the underlaying database supports those and Scrollable result sets is enabled/supported.

The work is performed on a preemptive thread, so this function does not block the application and can yield time to other Xojo threads. Must be called in a Xojo thread to enjoy benefits. If called in main thread will block, but keep other background threads running.

For Xojo 2019r2 and newer. See SQLSelectAsRecordSetMT for older versions.

5.15.47 SQLSelectMT(command as string, CommandType as Integer = 0) as string

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Executes a SQL command and returns the first field's string value.
Notes: This is a convenience function.
Internally it creates a SQLCommandMBS with the given command and calls Execute.
If the result is a record set, the first field from the first row is returned.
This is basicly useful for commands like "select sqlite_version()".

All text strings sent to the plugin must have a defined encoding. Else the internal text encoding conversions will fail.

The work is performed on a preemptive thread, so this function does not block the application and can yield time to other Xojo threads. Must be called in a Xojo thread to enjoy benefits. If called in main thread will block, but keep other background threads running.

5.15.48 UpdateRecord(TableName as String, Record as Dictionary, Keys as Dictionary)

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Convenience function to update a record.
Example:

```
Var con as SQLConnectionMBS // your database connection

Var d as new Dictionary

d.Value("text")="new text"
```

```
d.Value("other")="second value"

con.UpdateRecord("test_tbl", d, new dictionary("ID":2))
```

Notes: The plugin builds for you SQL statement with prepared statement and runs the update command with given values for records with given key values.

You can put multiple field names in the keys dictionary.

Lasterror is set or exception raised as with SQLExecute.
Internally this uses a prepared statement. You can check the generated statement via LastStatement property.

5.15.49 Properties

5.15.50 AutoCommit as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether autocommit is enabled or disabled for the current connection.
Example:

```
Var con as SQLConnectionMBS // your database connection
con.AutoCommit = SQLConnectionMBS.kAutoCommitOn
```

Notes: If autocommit is on, the database is committed automatically after each SQL command. Otherwise, transaction is committed only after Commit calling.
(Read and Write property)

5.15.51 Client as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The current DBMS client assigned for the connection.
Example:

```
Var con as new SQLConnectionMBS
con.client = SQLConnectionMBS.kSQLiteClient
```

Notes: Raises OutOfBoundsException exception if value parameter is out of range on setting.
(Read and Write property)

5.15.52 ClientVersion as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the DBMS client API version number.
Example:

```
Call InternalSQLiteLibraryMBS.Use

Var con As New SQLConnectionMBS
con.Client = con.kSQLiteClient

Var version As Integer = con.ClientVersion
Var HighVersion As Integer = Bitwise.ShiftRight(version, 16)
Var LowVersion As Integer = Bitwise.BitAnd(version, 65535)

var v as string = InternalSQLiteLibraryMBS.Version

MessageBox HighVersion.ToString+"."+LowVersion.ToString+EndOfLine+v
```

Notes: The higher word contains the major client version (the XX value in the XX.YY version number); the lower word contains the minor client version (the YY value in the XX.YY version number).

If an DBMS client was not set calling ClientVersion method will throw an exception.
(Read only property)

5.15.53 ConnectionCount as Integer

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries number of current connection objects.
Notes: This method should help you find leaked objects by keeping track of current count from the plugin perspective.
This includes SQLConnectionMBS and SQLDatabaseMBS objects.
(Read only property)

5.15.54 Error as Boolean

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether an error occurred.
Notes: This is set always when an error occurs and cleared if no error happened.
Be aware that the next call to a plugin function may reset error status.
If you look on this property in debugger, it's probably already cleared by the debugger querying a property.

(Read only property)

5.15.55 ErrorCode as Integer

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The last error code.
Notes: This is set always when an error occurs and cleared if no error happened.
(Read only property)

5.15.56 ErrorMessage as string

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The last error message.
Notes: This is set always when an error occurs and cleared if no error happened.
(Read only property)

5.15.57 isAlive as boolean

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the database server connection status for a particular connection object.
Notes: Returns true if the database server is active and accessible; otherwise false.
This method uses the safe query execution for most supported DBMS-es. The query uses the well known database table or procedure (mysql_ping is used for MySQL or MariaDB). If the query fails the method returns false.
(Read only property)

5.15.58 isConnected as boolean

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the connection state for a particular connection object.
Notes: Returns true if connected; otherwise false.
If you connect over a network, please check with isAlive whether the connection is still alive.
(Read only property)

5.15.59 IsolationLevel as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The transaction isolation level.
Example:

```
Var con as SQLConnectionMBS // your connection
con.IsolationLevel = SQLConnectionMBS.kReadUncommitted
```

Notes: Use the kReadCommitted, kReadUncommitted, kRepeatableRead, kSerializable and kLevelUnknown constants.
(Read and Write property)

5.15.60 LastStatement as String

Plugin Version: 15.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The last executed SQL Statement.
Notes: (Read only property)

5.15.61 NativeAPI as Variant

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Returns a set of functions of native DBMS client API.
Notes: Returns a SQLAPIMBS object.
Deprecated in version 19.5 in favor of direct methods in SQLDatabaseMBS and SQLConnectionMBS classes.
(Read only property)

5.15.62 Options as Dictionary

Plugin Version: 18.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a dictionary with all options.
Notes: For debugging, it may be useful to inspect options in debugger.
(Read only property)

5.15.63 RaiseExceptions as Boolean

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether to raise exceptions.
Notes: Default is true which means we set error, ErrorCode and ErrorMessage properties and than raise SQLErrorExceptionMBS exception.
If you set to false, we don't raise the exception and you have similar behavior as with database class.
We recommend to use exceptions as they are not so easily ignored like an error property being true.
(Read and Write property)

5.15.64 RowsAffected as Integer

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns number of rows affected by last DML operation.
Notes: Returns the number of rows affected by the last insert/update/delete command execution.

Can also return the result set rows count, usually when the result set is cached at the client side.
(Read only property)

5.15.65 Scrollable as Boolean

Plugin Version: 14.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether to make internally created SQLCommand objects scrollable.
Notes: Since plugin version 15.0, Scrollable is false by default.
(Read and Write property)

5.15.66 ServerVersion as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the currently connected DBMS server version number.
Notes: The higher word contains the major server version (the XX value in the XX.YY version number); the lower word contains the minor server version (the YY value in the XX.YY version number).
(Read only property)

5.15.67 ServerVersionString as string

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the currently connected DBMS server version string.
Notes: A server version string may contain some useful information about server brand, configuration and so on. It is a good idea to display this information in all your applications.
(Read only property)

5.15.68 SQLiteEncryptionKey as String

Plugin Version: 15.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The encryption key to use.
Example:

```
Var db as new SQLConnectionMBS
db.SQLiteEncryptionKey = "Hello"
```

Notes: This key is applied to the database after connecting. In case of an error, the plugin raises an exception. An empty key can be used for having no encryption.
Alternatively you can use SQLite3MBS.SetKey yourself.

The amount of key material actually used by the encryption extension depends on which variant of SEE you are using. With RC4, the first 256 byte of key are used. With the AES128, the first 16 bytes of the key are used. With AES256, the first 32 bytes of key are used.

If you specify a key that is shorter than the maximum key length, then the key material is repeated as many times as necessary to complete the key. If you specify a key that is larger than the maximum key length, then the excess key material is silently ignored.

The key must begin with an ASCII prefix to specify which algorithm to use. The prefix must be one of "rc4:", "aes128:", or "aes256:". The prefix is not used as part of the key sent into the encryption algorithm. So the real key should begin on the first byte after the prefix. If no prefix is given, we default to AES 128. To be compatible to Xojo, you can use AES128.

The string provided to the plugin is used with it's current encoding. So be sure you use right text encoding for what you want. e.g. using "M√ºller" as key in text encoding Windows ANSI will not open a database which used that key in UTF-8 encoding.

The Xojo database encryption in SQLiteDatabase class uses AES-128 OFB.
(Read and Write property)

5.15.69 Tag as Variant

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The tag property.
Notes: You can store here whatever you like.
(Read and Write property)

5.15.70 VariantsKeepSQLObjects as Boolean

Plugin Version: 16.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether variants should use SQL types.
Notes: If set to true, we return datetime and numeric as SQLDateTimeMBS and SQLNumericMBS objects.
If false, we return them as date and double.
(Read and Write property)

5.15.71 Option(name as string) as string

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: A string value of a specific connection or command option.
Example:

```
Var c as SQLConnectionMBS // your connection

// for Microsoft SQL use OLEDB, so you don't need native SQL drivers installed...
c.Option("UseAPI") = "OLEDB"
c.Option("SQLNCLI.LIBS") = "sqlsrv32.dll" // Library included in Windows Vista and newer

// for SQLite, set flag to open database file read only:
c.Option("SQLiteVFSFlags") = "1"

// set 10 seconds timeout for MySQL
c.Option("MYSQL_OPT_CONNECT_TIMEOUT") = "10"

// turn on auto cache
c.Option("AutoCache") = "true"

// set connection timeout for ODBC:
c.Option("SQL_ATTR_CONNECTION_TIMEOUT") = "10"
```

Notes: see also:
https://www.sqlapi.com/ApiDoc/class_s_a_connection.html

We have a collection of library files here:
https://www.monkeybreadsoftware.de/xojo/download/plugin/Libs/
(Read and Write computed property)

5.15.72 Events

5.15.73 DidConnect

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Event called after connection was made.
Notes: After we got connected, you can apply various options on the new connection here.

5.15.74 PostgresNotification(NotificationName as string, PID as Integer, Extras as String)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: A postgresSQL notification was received.
Notes: Provides notification name, process ID of app and optional extra information.

5.15.75 Trace(traceInfo as Integer, SQL as string, Command as SQLCommandMBS)

Plugin Version: 13.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The event to trace SQL commands.

5.15.76 WillConnect

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Event called before connection is established.
Notes: Last chance to set connection options.

5.15.77 Working

Plugin Version: 10.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The event called while the SQLExecuteMT and SQLSelectMT methods are running.

5.15.78 Constants

Constants

Constant	Value	Description
kOptionAPPNAME	"APPNAME"	A constant for the options.
kOptionLibraryCubeSQL	"CUBESQL.LIBS"	One of the option constant to specify the library with the SetFileO method. Tells the plugin where to find the library for CubeSQL. Only needed don't use InternalCubeSQLLibraryMBS module! The value can contain multiple names and paths separated with ";" on Wi and ":" on macOS/Linux. The library extension on Mac is ".dylib", on Linux ".so" and on Windows You get this library with the CubeSQL download on their homepage. If no value is given, we default to Windows: cubeSQL_64bit.dll;cubesql.dll macOS: libcubesql_r.dylib Linux: libcubesql_r.so The default is set to work in most cases and try various possible library r
kOptionWSID	"WSID"	A constant for the options.
SQLiteInMemory	":memory:"	Connection string for SQLite for a new in-memory database.

Isolation Levels

Constant	Value	Description
kANSILevel0	0	ANSI Level 0
kANSILevel1	1	ANSI Level 1
kANSILevel2	2	ANSI Level 2
kANSILevel3	3	ANSI Level 3
kLevelUnknown	-1	Unknown
kReadCommitted	1	Read committed.
kReadUncommitted	0	Read uncommitted.
kRepeatableRead	2	Repeatable read.
kSerializable	3	Serializable.
kSnapshot	4	Changes made in other transactions can not be seen. For Microsoft SQL Server.

Values for autocommit property

Constant	Value	Description
kAutoCommitOff	0	Autocommit is off.
kAutoCommitOn	1	Autocommit is on.
kAutoCommitUnknown	-1	Autocommit unknown

The database client constants

Constant	Value	Description
kClientNotSpecified	0	Client is not specified.
kCubeSQLClient	13	CubeSQL client. (coming soon)
kDB2Client	6	DB2 client.
kDuckDBClient	14	DuckDB client
kFirebirdClient	4	InterBase/Firebird client.
kInformixClient	7	Informix client.
kInterBaseClient	4	InterBase/Firebird client.
kMariaDBClient	15	MariaDB client.
kMySQLClient	9	MySQL or MariaDB client.
kODBCClient	1	ODBC client.
kOLEDBClient	16	The OLEDB client.
kOracleClient	2	Oracle client. For Windows the file is "oci.dll", for Linux libclntsh.so and for Mac OS X libclntsh.dylib.
kPostgreSQLClient	10	PostgreSQL client.
kSQLAnywhereClient	12	SQL Anywhere client.
kSQLBaseClient	5	SQLbase client.
kSQLiteClient	11	SQLite client. Or spatialite.
kSQLServerClient	3	Mircosoft SQL Server client. You may need to download the client packages for accessing SQL Server. Files like the SQLNCLI dll may be missing. You can download for example the Feature Pack for Microsoft SQL Server 2005 from the microsoft download page.
kSybaseClient	8	Sybase client.

Error Codes

Constant	Value	Description
kErrorBindVarNotFound	7	Bind variable not found.
kErrorClientInitFails	6	Initialization failed for client.
kErrorClientNotSet	1	Client not set.
kErrorClientNotSupported	2	Unsupported client type for this platform.
kErrorClientVersionOld	5	Library file is too old.
kErrorFieldNotFound	8	Field not found.
kErrorGetLibraryVersionFails	4	Failed to query library version.
kErrorLoadLibraryFails	3	Failed to load a library. For example path could be wrong or 32/64bit mismatch.
kErrorNoMemory	0	Out of memory.
kErrorUnknownColumnType	11	Unknown column type.
kErrorUnknownDataType	9	Unknown data type.
kErrorUnknownParameterType	10	Unknown parameter type.
kErrorWrongConversion	12	Failed to convert a value, e.g. string to number.
kErrorWrongDatetime	13	Can't convert text to date.

Options to specify the library with SetFileOption

Constant	Value	Description
kOptionLibraryDB2	"DB2CLI.LIBS"	Tells the plugin where to find the library for DB2. The value can contain multiple names and paths separated with ";" o and ":" on macOS/Linux. The library extension on Mac is ".dylib", on Linux ".so" and on Win You get this library with the DB2 download on their homepage. If no value is given, we default to Windows: db2clio.dll macOS: libdb2.dylib Linux: libdb2.so
kOptionLibraryDuckDB	"DUCKDB.LIBS"	Tells the plugin where to find the library for DuckDB. The value can contain multiple names and paths separated with ";" o and ":" on macOS/Linux. The library extension on Mac is ".dylib", on Linux ".so" and on Win You get this library with the DuckDB download on their homepage If no value is given, we default to Windows: duckdb.dll macOS: libduckdb.dylib Linux: libduckdb.so
kOptionLibraryFirebird	"IBASE.LIBS"	Tells the plugin where to find the library for FireBird (or Interbase The value can contain multiple names and paths separated with ";" o and ":" on macOS/Linux. The library extension on Mac is ".dylib", on Linux ".so" and on Win You get this library with the FireBird download on their homepage If no value is given, we default to Windows: ibclient64.dll;fbclient.dll;gds32.dll macOS: libgds.dylib:libfbclient.dylib Linux: libgds.so:libfbclient.so The default is set to work in most cases and try various possible libr
kOptionLibraryInformix	"INFCLI.LIBS"	Tells the plugin where to find the library for Informix. The value can contain multiple names and paths separated with ";" o and ":" on macOS/Linux. The library extension on Mac is ".dylib", on Linux ".so" and on Win You get this library with the Informix download on their homepage If no value is given, we default to Windows: ICLIT09B.DLL;ICLIT09A.DLL macOS: iclit09b.dylib:iclit09a.dylib:libifcli.dylib Linux: iclit09b.so:iclit09a.so:libifcli.so The default is set to work in most cases and try various possible libr
kOptionLibraryInterbase	"IBASE.LIBS"	Tells the plugin where to find the library for FireBird (or Interbase The value can contain multiple names and paths separated with ";" o and ":" on macOS/Linux. The library extension on Mac is ".dylib", on Linux ".so" and on Win You get this library with the FireBird download on their homepage If no value is given, we default to Windows: ibclient64.dll;fbclient.dll;gds32.dll macOS: libgds.dylib:libfbclient.dylib Linux: libgds.so:libfbclient.so The default is set to work in most cases and try various possible libr
kOptionLibraryMySQL	"MYSQL.LIBS"	Tells the plugin where to find the library for MySQL (or MariaDB The value can contain multiple names and paths separated with ";" o and ":" on macOS/Linux. The library extension on Mac is ".dylib", on Linux ".so" and on Win You get this library with the MySQL download on their homepage Libs folder on our website. If no value is given, we default to Windows: libmySQL.dll;libmariadb.dll macOS: libmysqlclient.dylib.21:libmysqlclient_r.dyli client_r.18.dylib:libmysqlclient_r.16.dylib:libmysqlclient_r.15.dylib

5.16 class SQLDatabaseMBS

5.16.1 class SQLDatabaseMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The database class for the SQL plugin
Example:

```
Var db as new SQLDatabaseMBS

// where is the library?
db.SetFileOption SQLConnectionMBS.kOptionLibrarySQLite, getfolderitem("/usr/lib/libsqlite3.0.dylib", folderitem.PathTy
Shell)

// connect to database
// in this example it is SQLite,
// but can also be Sybase, Oracle, Informix, DB2, SQLServer, InterBase, MySQL, SQLBase and ODBC

Var path as string

if TargetMacOS then
path = "/tmp/test.db" // put the database in the temporary folder
else
path = "test.db" // for Windows and Linux in the current folder the application is inside.
end if

db.DatabaseName = "sqlite:"+path

if db.Connect then

MsgBox "We are connected!"

// Disconnect is optional
// autodisconnect will ocur in destructor if needed
db.close

msgbox "We are disconnected!"
end if
```

Notes: You can use the SQL plugin without using Xojo built in database classes if you use the SQLConnectionMBS and SQLCommandMBS classes.

Or you use the SQLDatabaseMBS class which is a subclass of the database class and can be used with Xojo's RecordSet class. The current implementation is not complete. You can connect with passing the database URL in the DatabaseName property of the SQLDatabaseMBS class. You prefix this URL with the database type you are using.

You can use Execute and Select to run SQL statements. Errors can be queries with the lasterror properties. For the RecordSet, you can get the column count, the column names and values and move to the next row. All the other methods like deleting a record or updating a value are not implemented and you need to use SQL commands to do this.

Supported databases: CubeSQL, Centura SQLBase, DB2, DuckDB, Firebird, Informix, InterBase, MariaDB, Microsoft Access, Microsoft SQL Server, MySQL, ODBC, Oracle Database Server, PostgreSQL, SQL Anywhere, SQLite, SQLCipher and Sybase.
Connect to Microsoft Access, FileMaker Server (or Pro), Microsoft Visual FoxPro and others via ODBC.

As field and table schema functions are not implemented, you can't use this database with the database browser features in the Xojo IDE.
The plugin does not provide RecordCount on RecordSet class. For that you need to make a extra SELECT count(*) query.

With Xojo 2013r1, you only need a database server license from Xojo, Inc. if you use the SQLDatabaseMBS class. The SQLConnectionMBS class does not require this license. But some features like getting a recordset do need the license as they refer to the SQLDatabaseMBS class.

Please free all RecordSets and SQLCommand objects before you close the SQLConnection or the SQLDatabase. The plugin keeps references from RecordSets and SQLCommand to prevent automatic destruction of the database connection. If you close a database connection while you have RecordSets and SQLCommand in use, things may go wrong.

The plugin can cache the recordset locally. To enable you can call SQLCommandMBS.Cache or use the Option("AutoCache") = "true" on either command or connection or database objects. The plugin will than fetch all records and store them in memory. After this you can walk over the recordset and use FetchPos, FetchFirst, FetchLast, FetchPrev and FetchNext to locate the rows you need. When you call Field() you always get last row, but to read from cached result set, please use Value() function. When using RecordSet, the values are read via Value() functions automatically.

You can use InternalPostgreSQLLibraryMBS or InternalSQLiteLibraryMBS if you like to use our built in SQLite or PostgreSQL database libraries.

The class pings the database every minute by checking whether it's alive and to avoid server dropping connection. This can be disabled by setting Option("Ping") = "false". Ping is not used for SQLite.

MBS Plugin 21.1 adds support for Edit/Update methods in RecordSet and RowSet classes.

MBS Database connections are implemented via SQLConnectionMBS and SQLCommandMBS classes. We provide a thin layer on top with SQLDatabaseMBS class to make it compatible to the Xojo database class.

And when you use SQLDatabaseMBS, you can always get the matching SQLConnectionMBS objet via Connection property. Instead of SQLCommandMBS class, you may just use SelectSQL/ExecuteSQL or older SQLSelect/SQLExecute functions.

We have a collection of library files here:
https://www.monkeybreadsoftware.de/xojo/download/plugin/Libs/
Subclass of the Database class.
Blog Entries

- News from the MBS Xojo Plugins Version 23.5
- News from the MBS Xojo Plugins Version 22.2
- RowSet in MBS Xojo SQL Plugin
- Connecting to PostgreSQL with SSL and client certificate file in Xojo
- MBS Xojo Plugins 18.3
- MBS Xojo Web Starter Kit in version 1.3
- Upcoming Changes for our SQL Plugin
- MBS Releases the MBS Xojo / Real Studio plug-ins in version 16.1
- MBS Releases the MBS Xojo / Real Studio plug-ins in version 15.1
- Connecting to Microsoft Visual FoxPro

Videos

- MBS SQL Plugin Presentation
- Presentation from Munich conference about MBS Plugins.

Xojo Developer Magazine

- 22.1, page 9: News
- 21.1, page 9: News
- 18.5, page 10: News
- 14.1, pages 24 to 30: The MBS SQL Plugin, An alternative way to connect to databases by Christian Schmitz
- 13.4, page 11: News
- 12.4, page 9: News
- 12.3, page 10: News
- 11.6, page 40: ChartPart 3.0, Create charts and graphs within your Xojo applications by Kevin Cully
- 11.2, page 8: News
- 11.1, page 9: News

5.16.2 Methods

5.16.3 BeginTransaction

Plugin Version: 22.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Begins a transaction.
Notes: This method does nothing. Why?
Well, if auto commit is on, you don't need to call this.
If auto commit is off, we call BeginTransaction for you, so if you call this method you would call it a second time and get an error.
We have this method for compatibility to other SQL database classes from Xojo.

5.16.4 CancelAllCommands

Plugin Version: 18.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Cancel all commands for the connection.
Notes: This loops over the list of commands associated with this connection and calls Cancel on them.

5.16.5 Commands as SQLCommandMBS()

Plugin Version: 18.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries list of all command objects related to the connection.

5.16.6 Connect as boolean

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Connects to the database.
Example:

```
Var db as new SQLDatabaseMBS

// where is the library?
db.SetFileOption SQLConnectionMBS.kOptionLibraryMySQL, SpecialFolder.UserHome.Child("libmysqlclient.dylib")

// connect to database
// in this example it is MySQL,
// but can also be Sybase, Informix, Oracle, DB2
// SQLServer, InterBase, SQLBase and ODBC
```

```
db.DatabaseName="mysql:192.168.1.80,3306@test"

db.UserName="root"
db.Password=""

// or postgreSQL with timeout and ssl mode
db.DatabaseName="PostgreSQL:127.0.0.1,5432@dbname=postgres connect_timeout=10 sslmode=require"

if db.Connect then

MsgBox "We are connected!"

MsgBox "Server Version: "+db.Connection.ServerVersionString

// Disconnect is optional
// autodisconnect will ocur in destructor if needed

else
MsgBox db.ErrorMessage
end if
```

Notes: Returns true on success and false on failure.

Please set the DatabaseName, UserName and Password properties. The Host property is ignored.
The database name must contain the complete information and a prefix for the kind of database.

Use this prefixes: "CubeSQL:", "SQLAnywhere:", "ODBC:", "Oracle:", "SQLServer:", "Firebird:", "DuckDB:", "InterBase:", "SQLBase:", "DB2:", "Informix:", "Sybase:", "MySQL:", "MariaDB:", "PostgreSQL:" or "SQLite:". Connect to Microsoft Access, FileMaker Server (or Pro), Microsoft Visual FoxPro and others via ODBC.

For IPv6 we changed plugin to use , instead of : for the port separator. So please use , to separate port from IP or host.

For Firebird, if you connect to a database and you have 32/64bit mismatch, you get error number 3.

5.16.7 ConnectMT as Boolean

Plugin Version: 15.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Connects to the database.
Notes: Returns true on success and false on failure.

Please set the DatabaseName, UserName and Password properties. The Host property is ignored.

The database name must contain the complete information and a prefix for the kind of database.

Use this prefixes: ”CubeSQL:”, ”SQLAnywhere:”, ”ODBC:”, ”Oracle:”, ”SQLServer:”, ”Firebird:”, ”DuckDB:”, ”InterBase:”, ”SQLBase:”, ”DB2:”, ”Informix:”, ”Sybase:”, ”MySQL:”, ”MariaDB:”, ”PostgreSQL:” or ”SQLite:”. Connect to Microsoft Access, FileMaker Server (or Pro), Microsoft Visual FoxPro and others via ODBC.

For IPv6 we changed plugin to use , instead of : for the port separator. So please use , to separate port from IP or host.

Same as Connect, but if you run this on a thread, the plugin gives time to other threads so the rest of your application runs just fine.

The MT method will not trigger WillConnect and DidConnect events.

The work is performed on a preemptive thread, so this function does not block the application and can yield time to other Xojo threads. Must be called in a Xojo thread to enjoy benefits. If called in main thread will block, but keep other background threads running.

5.16.8 Constructor(globals as SQLGlobalsMBS = nil)

Plugin Version: 13.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The constructor.
Notes: Please don’t call this directly as it’s called automatically with using new command.

5.16.9 CubeSQLLastInsertID as Int64

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the last auto increment value from last insert command.
Notes: Only for CubeSQL connections. May raise error if not available.

5.16.10 CubeSQLReceiveData(byref data as String, byref IsEndChunk as Boolean) as Boolean

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Receives a data chunk for file download.
Notes: Returns true on success.

Data is set with data and IsEndChunk is set to true for last chunk.

5.16.11 CubeSQLSendData(data as MemoryBlock)

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sends data chunk for file upload.
Notes: This is the sendchunk function in CubeSQL.
See also:

- 5.16.12 CubeSQLSendData(data as String) 132

5.16.12 CubeSQLSendData(data as String)

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sends data chunk for file upload.
Notes: This is the sendchunk function in CubeSQL.
See also:

- 5.16.11 CubeSQLSendData(data as MemoryBlock) 132

5.16.13 CubeSQLSendEndData

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sends end data packet.
Notes: This is the send_enddata function in CubeSQL.

5.16.14 InsertRecord(TableName as String, Record as Dictionary)

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Convenience function to insert a record.
Example:

```
Var db as SQLDatabaseMBS // your database connection

Var d as new Dictionary

d.Value("ID")=2
d.Value("text")="test insert"
d.Value("other")="Just a test"
```

```
db.InsertRecord("test_tbl", d)

if db.Error then
MsgBox db.ErrorMessage
end if
```

Notes: The plugin builds for you SQL statement with prepared statement and runs the insert command with values.
Lasterror is set or exception raised as with SQLExecute.
Internally this uses a prepared statement. You can check the generated statement via LastStatement property.

5.16.15 Listen

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Start listening for notifications.
Notes: Works only for PostgresSQL Client.
Please set client or connect before calling this method.

5.16.16 MySQLInsertID as Int64

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the last auto increment value from last insert command.
Notes: Please query value right after doing Insert. This value is reset when you call commit.
For mySQL and MariaDB connections.

see also
http://dev.mysql.com/doc/refman/5.1/en/mysql-insert-id.html

5.16.17 Prepare(statement as string) as SQLPreparedStatementMBS

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Prepares a statement.
Notes: Returns prepared statement or nil in case of error.
Please check ErrorMessage property for errors.

5.16.18 SetFileOption(name as string, file as folderitem)

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets an option with passing a file path.
Example:

```
Var db as new SQLDatabaseMBS

// where is the library?
db.SetFileOption SQLConnectionMBS.kOptionLibraryMySQL, SpecialFolder.UserHome.Child("libmysqlclient.dylib")
```

Notes: Allows you to specify a file path with a folderitem.

Makes sure the path is correct and you have a 32 or 64-bit library matching the architecture of your application.

5.16.19 SQLExecute(ExecuteString as string, CommandType as Integer)

Plugin Version: 13.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Runs the SQLExecute threaded.
Notes: Same as SQLExecute, but with additional CommandType parameter.

5.16.20 SQLExecuteMT(ExecuteString as string, CommandType as Integer = 0)

Plugin Version: 12.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Runs the SQLExecute threaded.
Notes: Same as SQLExecute, but if you run this on a thread, the plugin gives time to other threads so the rest of your application runs just fine.

The work is performed on a preemptive thread, so this function does not block the application and can yield time to other Xojo threads. Must be called in a Xojo thread to enjoy benefits. If called in main thread will block, but keep other background threads running.

5.16.21 SQLiteBackupFinish(Backup as SQLite3BackupMBS) as integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Finishes a backup run.

Notes: When BackupStep has returned kErrorDone, or when the application wishes to abandon the backup operation, the application should destroy the SQLite3BackupMBS by passing it to BackupFinish. The BackupFinish interfaces releases all resources associated with the SQLite3BackupMBS object. If BackupStep has not yet returned kErrorDone, then any active write-transaction on the destination database is rolled back. The SQLite3BackupMBS object is invalid and may not be used following a call to BackupFinish.
The value returned by BackupFinish is kErrorOK if no BackupStep errors occurred, regardless or whether or not BackupStep completed. If an out-of-memory condition or IO error occurred during any prior BackupStep call on the same SQLite3BackupMBS object, then BackupFinish returns the corresponding error code.
A return of kErrorBusy or kErrorLocked from BackupStep is not a permanent error and does not affect the return value of BackupFinish.

5.16.22 SQLiteBackupInit(Dest as Variant, DestName as String, Source as Variant, SourceName as String) as SQLite3BackupMBS

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Initializes a backup.
Notes: The backup API copies the content of one database into another. It is useful either for creating backups of databases or for copying in-memory databases to or from persistent files.

see also
http://www.sqlite.org/c3ref/backup_finish.html

Exclusive access is required to the destination database for the duration of the operation. However the source database is only read-locked while it is actually being read; it is not locked continuously for the entire backup operation. Thus, the backup may be performed on a live source database without preventing other users from reading or writing to the source database while the backup is underway.

To perform a backup operation:

- BackupInit is called once to initialize the backup,
- BackupStep is called one or more times to transfer the data between the two databases, and finally
- BackupFinish is called to release all resources associated with the backup operation.

There should be exactly one call to BackupFinish for each successful call to BackupInit.

The D and N arguments to BackupInit(D,N,S,M) are the database connection associated with the destination database and the database name, respectively. The database name is ”main” for the main database, ”temp” for the temporary database, or the name specified after the AS keyword in an ATTACH statement for an attached database. The S and M arguments passed to BackupInit(D,N,S,M) identify the database connection and database name of the source database, respectively. The source and destination database

connections (parameters S and D) must be different or else BackupInit(D,N,S,M) will file with an error.
If an error occurs within BackupInit(D,N,S,M), then nil is returned and an error code and error message are store3d in the destination database connection D. The error code and message for the failed call to BackupInit can be retrieved using the ErrCode and ErrMessage functions. A successful call to BackupInit returns a SQLite3BackupMBS object. The SQLite3BackupMBS object may be used with the BackupStep and BackupFinish functions to perform the specified backup operation.

Concurrent Usage of Database Handles
The source database connection may be used by the application for other purposes while a backup operation is underway or being initialized. If SQLite is compiled and configured to support threadsafe database connections, then the source database connection may be used concurrently from within other threads.
However, the application must guarantee that the destination database connection is not passed to any other API (by any thread) after BackupInit is called and before the corresponding call to BackupFinish. SQLite does not currently check to see if the application incorrectly accesses the destination database connection and so no error code is reported, but the operations may malfunction nevertheless. Use of the destination database connection while a backup is in progress might also also cause a mutex deadlock.
If running in shared cache mode, the application must guarantee that the shared cache used by the destination database is not accessed while the backup is running. In practice this means that the application must guarantee that the disk file being backed up to is not accessed by any connection within the process, not just the specific connection that was passed to BackupInit.
The SQLite3BackupMBS object itself is partially threadsafe. Multiple threads may safely make multiple concurrent calls to BackupStep. However, the BackupRemaining and BackupPageCount APIs are not strictly speaking threadsafe. If they are invoked at the same time as another thread is invoking BackupStep it is possible that they return invalid values.

Source and Dest can be SQLConnectionMBS or SQLDatabaseMBS. You need to pass source and dest, even if one is self as we give you the option to decide where to pass the current database connection.

5.16.23 SQLiteBackupPageCount(Backup as SQLite3BackupMBS) as integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of pages in total.
Notes: Each call to BackupStep sets two values inside the SQLite3BackupMBS object: the number of pages still to be backed up and the total number of pages in the source databae file. The BackupRemaining and BackupPageCount interfaces retrieve these two values, respectively.
The values returned by these functions are only updated by BackupStep. If the source database is modified during a backup operation, then the values are not updated to account for any extra pages that need to be updated or the size of the source database file changing.

5.16.24 SQLiteBackupRemaining(Backup as SQLite3BackupMBS) as integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of pages remaining.
Notes: Each call to BackupStep sets two values inside the SQLite3BackupMBS object: the number of pages still to be backed up and the total number of pages in the source databae file. The BackupRemaining and BackupPageCount interfaces retrieve these two values, respectively.
The values returned by these functions are only updated by BackupStep. If the source database is modified during a backup operation, then the values are not updated to account for any extra pages that need to be updated or the size of the source database file changing.

5.16.25 SQLiteBackupStep(Backup as SQLite3BackupMBS, Pages as Integer) as integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies up to Pages pages between the source and destination databases specified by SQLite3BackupMBS object.
Notes: If N is negative, all remaining source pages are copied. If BackupStep(B,N) successfully copies N pages and there are still more pages to be copied, then the function resturns kErrorOK. If BackupStep(B,N) successfully finishes copying all pages from source to destination, then it returns kErrorDone. If an error occurs while running BackupStep(B,N), then an error code is returned. As well as kErrorOK and kErrorDone, a call to BackupStep may return kErrorReadOnly, kErrorNoMem, kErrorBusy, kErrorLocked, or an kErrorIOACCESS | kErrorIOXXX extended error code.
The BackupStep might return kErrorReadOnly if the destination database was opened read-only or if the destination is an in-memory database with a different page size from the source database.
If BackupStep cannot obtain a required file-system lock, then the sqlite3_busy_handler | busy-handler function is invoked (if one is specified). If the busy-handler returns non-zero before the lock is available, then kErrorBusy is returned to the caller. In this case the call to BackupStep can be retried later. If the source database connection is being used to write to the source database when BackupStep is called, then kErrorLocked is returned immediately. Again, in this case the call to BackupStep can be retried later on. (If kErrorIOACCESS | kErrorIOXXX, kErrorNoMem, or kErrorReadOnly is returned, then there is no point in retrying the call to BackupStep. These errors are considered fatal.) The application must accept that the backup operation has failed and pass the backup operation handle to the BackupFinish to release associated resources.
The first call to BackupStep obtains an exclusive lock on the destination file. The exclusive lock is not released until either BackupFinish is called or the backup operation is complete and BackupStep returns kErrorDone. Every call to BackupStep obtains a shared lock on the source database that lasts for the duration of the BackupStep call. Because the source database is not locked between calls to BackupStep, the source database may be modified mid-way through the backup process. If the source database is modified by an external process or via a database connection other than the one being used by the backup operation, then the backup will be automatically restarted by the next call to BackupStep. If the source database is modified by the using the same database connection as is used by the backup operation, then the backup database is automatically updated at the same time.

5.16.26 SQLiteConnectionHandle as Ptr

Plugin Version: 22.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the current connection reference for the database.
Notes: sqlite3 pointer for using in declares.

5.16.27 SQLiteEnableLoadExtension(OnOff as boolean)

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Enables/disables extension loading for the given connection.

5.16.28 SQLiteLastInsertRowID as Int64

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns Last Insert Rowid.
Notes: Each entry in an SQLite table has a unique 64-bit signed integer key called the ROWID. The rowid is always available as an undeclared column named ROWID, OID, or _ROWID_ as long as those names are not also used by explicitly declared columns. If the table has a column of type INTEGER PRIMARY KEY then that column is another alias for the rowid.

This routine returns the rowid of the most recent successful INSERT into the database from the database connection in the first argument. If no successful INSERTs have ever occurred on that database connection, zero is returned.

(If an INSERT occurs within a trigger, then the rowid of the inserted row is returned by this routine as long as the trigger is running. But once the trigger terminates, the value returned by this routine reverts to the last value inserted before the trigger fired.)

An INSERT that fails due to a constraint violation is not a successful INSERT and does not change the value returned by this routine. ^Thus INSERT OR FAIL, INSERT OR IGNORE, INSERT OR ROLLBACK, and INSERT OR ABORT make no changes to the return value of this routine when their insertion fails. ^(When INSERT OR REPLACE encounters a constraint violation, it does not fail. The INSERT continues to completion after deleting rows that caused the constraint problem so INSERT OR REPLACE will always change the return value of this interface.)^

For the purposes of this routine, an INSERT is considered to be successful even if it is subsequently rolled back.

This function is accessible to SQL statements via the last_insert_rowid() SQL function.

If a separate thread performs a new INSERT on the same database connection while the LastInsertRowID function is running and thus changes the last insert rowid, then the value returned by LastInsertRowID is unpredictable and might not equal either the old or the new last insert rowid.

5.16.29 SQLiteLibVersion as String

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the version string of the SQLite library.

5.16.30 SQLiteLoadExtension(file as FolderItem, ByRef ErrorMessage as String) as Integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads an SQLite extension library from the named file.
Notes: The LoadExtension interface attempts to load an SQLite extension library contained in the file.

Returns kErrorOk on success and kErrorError if something goes wrong.
Extension loading must be enabled using EnableLoadExtension prior to calling this API, otherwise an error will be returned.
See also:

- 5.16.31 SQLiteLoadExtension(path as String, ByRef ErrorMessage as String) as Integer 139

5.16.31 SQLiteLoadExtension(path as String, ByRef ErrorMessage as String) as Integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads an SQLite extension library from the named file.
Notes: The LoadExtension interface attempts to load an SQLite extension library contained in the file.

Returns kErrorOk on success and kErrorError if something goes wrong.
Extension loading must be enabled using EnableLoadExtension prior to calling this API, otherwise an error will be returned.
See also:

- 5.16.30 SQLiteLoadExtension(file as FolderItem, ByRef ErrorMessage as String) as Integer 139

5.16.32 SQLiteMemoryHighwater(reset as boolean = false) as Int64

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries maximum memory usage so far.
Notes: Can be reset with reset parameter being true.

5.16.33 SQLiteMemoryUsed as Int64

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries memory in use by SQLite.
Notes: This is memory allocated, but not yet freed.
Value is zero until SQLite3 initialized.

5.16.34 SQLiteReKey(Key as String) as Integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: You can change the key on a database using the Rekey Function.
Notes: An empty key decrypts the database.

Rekeying requires that every page of the database file be read, decrypted, reencrypted with the new key, then written out again. Consequently, rekeying can take a long time on a larger database.

Most SEE variants allow you to encrypt an existing database that was created using the public domain version of SQLite. This is not possible when using the authenticating version of the encryption extension in see-aes128-ccm. If you do encrypt a database that was created with the public domain version of SQLite, no nonce will be used and the file will be vulnerable to a chosen-plaintext attach. If you call SetKey() immediately after Open when you are first creating the database, space will be reserved in the database for a nonce and the encryption will be much stronger. If you do not want to encrypt right away, call SetKey() anyway, with an empty key, and the space for the nonce will be reserved in the database even though no encryption is done initially.

A public domain version of the SQLite library can read and write an encrypted database with an empty key. You only need the encryption extension if the key is non-empty.

Returns a SQLite error code.

5.16.35 SQLiteSetBusyHandler(MaxAttempts as Integer = 5)

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Installs busy handler for this connection.
Notes: This routine sets a callback function that might be invoked whenever an attempt is made to open a database table that another thread or process has locked.

The plugin has an busy handler which will wait up to MaxAttemps and yield to other Xojo threads while waiting.
Passing 5 should wait up to 100ms.

There can only be a single busy handler defined for each [database connection] . Setting a new busy handler clears any previously set handler.) Note that calling SetBusyTimeout will also set or clear the busy handler.

The busy callback should not take any actions which modify the database connection that invoked the busy handler. Any such actions result in undefined behavior.

5.16.36 SQLiteSetBusyTimeout(TimeOutMS as Integer = 20)

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: This routine sets a busy handler that sleeps for a specified amount of time when a table is locked.
Notes: The handler will sleep multiple times until at least "ms" milliseconds of sleeping have accumulated. ^After at least "ms" milliseconds of sleeping, the handler returns 0 which causes SQLite query to return SQLite Busy or IO Blocked error.

Calling this routine with an argument less than or equal to zero turns off all busy handlers.

(There can only be a single busy handler for a particular database connection any any given moment. If another busy handler was defined (using SetBusyHandler prior to calling this routine, that other busy handler is cleared.)

5.16.37 SQLiteSetKey(Key as String) as Integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Applies encryption to a database connection.
Notes: Returns a SQLite error code.

The amount of key material actually used by the encryption extension depends on which variant of SEE you

are using. With RC4, the first 256 byte of key are used. With the AES128, the first 16 bytes of the key are used. With AES256, the first 32 bytes of key are used.

If you specify a key that is shorter than the maximum key length, then the key material is repeated as many times as necessary to complete the key. If you specify a key that is larger than the maximum key length, then the excess key material is silently ignored.

The key must begin with an ASCII prefix to specify which algorithm to use. The prefix must be one of "rc4:", "aes128:", or "aes256:". The prefix is not used as part of the key sent into the encryption algorithm. So the real key should begin on the first byte after the prefix.

The string provided to the plugin is used with it's current encoding. So be sure you use right text encoding for what you want. e.g. using "M√ºller" as key in text encoding Windows ANSI will not open a database which used that key in UTF-8 encoding.

The Xojo database encryption in SQLiteDatabase class uses AES-128 OFB.

5.16.38 SQLiteTableColumnMetaData(DBName as string, TableName as string, ColumnName as string, byref DataType as string, byref CollationSequence as string, byref NotNull as boolean, byref PrimaryKey as boolean, byref AutoIncrement as Boolean) as integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Extract Metadata About A Column Of A Table
Notes: Not available in all sqlite libraries!

This routine returns metadata about a specific column of a specific database table accessible using the database connection handle passed as the first function argument.
The column is identified by the second, third and fourth parameters to this function. The second parameter is either the name of the database (i.e. "main", "temp", or an attached database) containing the specified table or NULL. If it is NULL, then all attached databases are searched for the table using the same algorithm used by the database engine to resolve unqualified table references.
The third and fourth parameters to this function are the table and column name of the desired column, respectively. Neither of these parameters may be NULL.
Metadata is returned by writing to the memory locations passed as the 5th and subsequent parameters to this function. Any of these arguments may be NULL, in which case the corresponding element of metadata is omitted.

CollationSequence is assigned the Name of default collation sequence. NotNull is set to true if column has a NOT NULL constraint. PrimaryKey is set to true if column is part of the PRIMARY KEY and AutoIncrement is set to true if column is AUTOINCREMENT.

If the specified table is actually a view, an error code is returned.

If the specified column is "rowid", "oid" or "_rowid_" and an INTEGER PRIMARY KEY column has been explicitly declared, then the output parameters are set for the explicitly declared column. (If there is no explicitly declared INTEGER PRIMARY KEY column, then the output parameters are set as follows:

data type: "INTEGER"
collation sequence: "BINARY"
not null: false
primary key: true
auto increment: false

(This function may load one or more schemas from database files. If an error occurs during this process, or if the requested table or column cannot be found, an error code is returned and an error message left in the database connection (to be retrieved using ErrMessage).)

This API is only available if the library was compiled with the SQLITE_ENABLE_COLUMN_METADATA C-preprocessor symbol defined.

5.16.39 SQLiteThreadsafe as integer

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Test To See If The Library Is Threadsafe.
Notes: The threadsafe() function returns zero if and only if SQLite was compiled mutexing code omitted due to the SQLITE_THREADSAFE compile-time option being set to 0.

SQLite can be compiled with or without mutexes. When the SQLITE_THREADSAFE C preprocessor macro is 1 or 2, mutexes are enabled and SQLite is threadsafe. When the SQLITE_THREADSAFE macro is 0, the mutexes are omitted. Without the mutexes, it is not safe to use SQLite concurrently from more than one thread.

Enabling mutexes incurs a measurable performance penalty. So if speed is of utmost importance, it makes sense to disable the mutexes. But for maximum safety, mutexes should be enabled. The default behavior is for mutexes to be enabled.

This interface can be used by an application to make sure that the version of SQLite that it is linking against was compiled with the desired setting of the SQLITE_THREADSAFE macro.

This interface only reports on the compile-time mutex setting of the SQLITE_THREADSAFE flag. If SQLite

is compiled with SQLITE_THREADSAFE=1 or =2 then mutexes are enabled by default but can be fully or partially disabled using a call to sqlite3_config() with the verbs SQLITE_CONFIG_SINGLETHREAD, SQLITE_CONFIG_MULTITHREAD, or SQLITE_CONFIG_MUTEX. ^(The return value of the sqlite3_threadsafe() function shows only the compile-time setting of thread safety, not any run-time changes to that setting made by sqlite3_config(). In other words, the return value from sqlite3_threadsafe() is unchanged by calls to sqlite3_config().)^

See the threading mode documentation for additional information.

5.16.40 SQLSelect(SelectString as string, CommandType as Integer) as RecordSet

Plugin Version: 13.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Runs the SQLSelect threaded.
Notes: Same as SQLSelect, but with additional CommandType parameter.

For this method to work, you need to have somewhere a property with SQLDatabaseMBS so Xojo includes our SQLDatabase plugin which provides the RecordSet functionality.

If Scrollable property is true, the recordset will be requested to be scrollable.

The record set may not have a valid RecordCount or have working movefirst/movelast/moveprev methods unless the underlaying database supports those and Scrollable result sets is enabled/supported.

5.16.41 SQLSelectMT(SelectString as string, CommandType as Integer = 0) as RecordSet

Plugin Version: 12.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Runs the SQLSelect threaded.
Notes: Same as SQLSelect, but if you run this on a thread, the plugin gives time to other threads so the rest of your application runs just fine.

The work is performed on a preemptive thread, so this function does not block the application and can yield time to other Xojo threads. Must be called in a Xojo thread to enjoy benefits. If called in main thread will block, but keep other background threads running.

If Scrollable property is true, the recordset will be requested to be scrollable.

The record set may not have a valid RecordCount or have working movefirst/movelast/moveprev methods unless the underlaying database supports those and Scrollable result sets is enabled/supported.

5.16.42 UpdateRecord(TableName as String, Record as Dictionary, Keys as Dictionary)

Plugin Version: 18.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Convenience function to update a record.
Example:

```
Var db as SQLDatabaseMBS // your database connection

Var d as new Dictionary

d.Value("text")="new text"
d.Value("other")="second value"

db.UpdateRecord("test_tbl", d, new dictionary("ID":2))

if db.Error then
MsgBox db.ErrorMessage
end if
```

Notes: The plugin builds for you SQL statement with prepared statement and runs the update command with given values for records with given key values.

You can put multiple field names in the keys dictionary.

Lasterror is set or exception raised as with SQLExecute.
Internally this uses a prepared statement. You can check the generated statement via LastStatement property.

5.16.43 Properties

5.16.44 AutoCommit as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether autocommit is enabled or disabled for the current connection.
Example:

```
Var db as SQLDatabaseMBS // your database connection
```

```
db.AutoCommit = SQLDatabaseMBS.kAutoCommitOn
```

Notes: If autocommit is on, the database is committed automatically after each SQL command. Otherwise, transaction is committed only after Commit calling.
(Read and Write property)

5.16.45 Client as Integer

Plugin Version: 14.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The current DBMS client assigned for the connection.
Example:

```
Var con as new SQLDatabaseMBS
db.client = SQLConnectionMBS.kSQLiteClient
```

Notes: Raises OutOfBoundsException exception if value parameter is out of range on setting.
With SQLDatabaseMBS, the client is usually taken from the DatabaseName property with a prefix, e.g. "ODBC:test".
(Read and Write property)

5.16.46 ClientVersion as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the DBMS client API version number.
Example:

```
Call InternalSQLiteLibraryMBS.Use

Var db As New SQLDatabaseMBS
db.Client = SQLConnectionMBS.kSQLiteClient

Var version As Integer = db.ClientVersion
Var HighVersion As Integer = Bitwise.ShiftRight(version, 16)
Var LowVersion As Integer = Bitwise.BitAnd(version, 65535)

var v as string = InternalSQLiteLibraryMBS.Version

MessageBox HighVersion.ToString+"."+LowVersion.ToString+EndOfLine+v
```

Notes: The higher word contains the major client version (the XX value in the XX.YY version number); the lower word contains the minor client version (the YY value in the XX.YY version number).

If an DBMS client was not set calling ClientVersion method will throw an exception.
(Read only property)

5.16.47 Connection as SQLConnectionMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The connection for this database used in the background.
Notes: Note that methods on this connection object can raise exceptions while methods on the SQLDatabaseMBS class sets the error properties.
(Read only property)

5.16.48 isAlive as boolean

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the database server connection status for a particular connection object.
Notes: Returns true if the database server is active and accessible; otherwise false.
This method uses the safe query execution for most supported DBMS-es. The query uses the well known database table or procedure (mysql_ping is used for MySQL or MariaDB). If the query fails the method returns false.

For Sybase, SQLBase, MS SQL Server, Oracle, ODBC, MySQL, MariaDB, DB2, CubeSQL, Informix, Interbase, FireBase and PostgreSQL we check the status of the connection. This may send a query over the network and sees if server responds properly. If network connection breaks, this function returns false.

For SQLite, DuckDB and SQL Anywhere we return true if we are connected, just like isConnected property.
(Read only property)

5.16.49 isConnected as boolean

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the connection state for a particular connection object.
Notes: Returns true if connected; otherwise false.
If you connect over a network, please check with isAlive whether the connection is still alive.
(Read only property)

5.16.50 IsolationLevel as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The transaction isolation level.
Notes: Use the kReadCommitted, kReadUncommitted, kRepeatableRead, kSerializable and kLevelUnknown constants.
(Read and Write property)

5.16.51 LastStatement as String

Plugin Version: 13.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The last executed SQL Statement.
Notes: (Read only property)

5.16.52 NativeAPI as Variant

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Returns a set of functions of native DBMS client API.
Notes: Returns a SQLAPIMBS object.
Deprecated in version 19.5 in favor of direct methods in SQLDatabaseMBS and SQLConnectionMBS classes.
(Read only property)

5.16.53 Options as Dictionary

Plugin Version: 18.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a dictionary with all options.
Notes: For debugging, it may be useful to inspect options in debugger.
(Read only property)

5.16.54 RaiseExceptions as Boolean

Plugin Version: 14.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether to raise exceptions.
Notes: Default is false which means we set error, ErrorCode and ErrorMessage properties and not raise SQLErrorExceptionMBS exception.
If you set to true, we do raise the exception and you have similar behavior as with SQLConnection class.

We recommend to use exceptions as they are not so easily ignored like an error property being true.
(Read and Write property)

5.16.55 RowsAffected as Integer

Plugin Version: 23.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns number of rows affected by last DML operation.
Notes: Returns the number of rows affected by the last insert/update/delete command execution.

Can also return the result set rows count, usually when the result set is cached at the client side.
(Read only property)

5.16.56 Scrollable as Boolean

Plugin Version: 14.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether to make internally created SQLCommand objects scrollable.
Notes: Since plugin version 15.0, Scrollable is false by default.
(Read and Write property)

5.16.57 ServerVersion as Integer

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the currently connected DBMS server version number.
Notes: The higher word contains the major server version (the XX value in the XX.YY version number); the lower word contains the minor server version (the YY value in the XX.YY version number).
(Read only property)

5.16.58 ServerVersionString as string

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets the currently connected DBMS server version string.
Notes: A server version string may contain some useful information about server brand, configuration and so on. It is a good idea to display this information in all your applications.
(Read only property)

5.16.59 SQLiteEncryptionKey as String

Plugin Version: 15.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The encryption key to use.
Example:

```
// enable our built-in SQLite library, which supports encryption
Call InternalSQLiteLibraryMBS.Use

// where to store?
Var f As FolderItem = SpecialFolder.Desktop.Child("test.db")
Var storage_database As New SQLDatabaseMBS ' SQLiteDatabase
storage_database.SQLiteEncryptionKey = "aes256:password" ' <- password with AES256 as prefix to pick algorithm
storage_database.DatabaseName = "sqlite:"+ f.NativePath

If storage_database.Connect Then

// create table if this is not yet here
storage_database.SQLExecute "Create table if not exists pics(pic_id integer PRIMARY KEY AUTOINCREMENT, name varchar(20), pic blob)"

// done
MsgBox "Ready"
Else
MsgBox storage_database.ErrorMessage
End If
```

Notes: This key is applied to the database after connecting. In case of an error, the plugin raises an exception. An empty key can be used for having no encryption.
Alternatively you can use SQLite3MBS.SetKey yourself.

The amount of key material actually used by the encryption extension depends on which variant of SEE you are using. With RC4, the first 256 byte of key are used. With the AES128, the first 16 bytes of the key are used. With AES256, the first 32 bytes of key are used.

If you specify a key that is shorter than the maximum key length, then the key material is repeated as many times as necessary to complete the key. If you specify a key that is larger than the maximum key length, then the excess key material is silently ignored.

The key must begin with an ASCII prefix to specify which algorithm to use. The prefix must be one of "rc4:", "aes128:", or "aes256:". The prefix is not used as part of the key sent into the encryption algorithm. So the real key should begin on the first byte after the prefix. If no prefix is given, we default to AES 128. To be compatible to Xojo, you can use AES128.

The string provided to the plugin is used with it's current encoding. So be sure you use right text encoding for what you want. e.g. using "M√∂ller" as key in text encoding Windows ANSI will not open a database which used that key in UTF-8 encoding.

The Xojo database encryption in SQLiteDatabase class uses AES-128 OFB.
(Read and Write property)

5.16.60 Tag as Variant

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: The tag property.
Notes: You can store here whatever you like.
(Read and Write property)

5.16.61 Option(name as string) as string

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets an option for the connection.
Example:

```
Var c as SQLDatabaseMBS // your database connection

// for Microsoft SQL use OLEDB, so you don't need native SQL drivers installed...
c.Option("UseAPI") = "OLEDB"
c.Option("SQLNCLI.LIBS") = "sqlsrv32.dll" // Library included in Windows Vista and newer

// for SQLite, set flag to open database file read only:
c.Option("SQLiteVFSFlags") = "1"

// turn on auto cache
c.Option("AutoCache") = "true"

// set connection timeout for ODBC:
c.Option("SQL_ATTR_CONNECTION_TIMEOUT") = "10"

// cache insert statements for AddRow/InsertRecord
c.Option("CacheInsertStatement") = "true"
```

Notes: We have a collection of library files here:
https://www.monkeybreadsoftware.de/xojo/download/plugin/Libs/
(Read and Write computed property)

5.16.62 Events

5.16.63 DidConnect

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Event called after connection was made.
Notes: After we got connected, you can apply various options on the new connection here.

5.16.64 PostgresNotification(NotificationName as string, PID as Integer, Extras as String)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: A postgresSQL notification was received.
Notes: Provides notification name, process ID of app and optional extra information.

5.16.65 Trace(traceInfo as Integer, SQL as string, Command as SQLCommandMBS)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The event to trace SQL commands.

5.16.66 WillConnect

Plugin Version: 14.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Event called before connection is established.
Notes: Last chance to set connection options.

5.16.67 Constants

Constants

Constant	Value	Description
kOptionLibraryCubeSQL	"CUBESQL.LIBS"	One of the option constant to specify the library with the SetFileO method. Tells the plugin where to find the library for CubeSQL. Only needed don't use InternalCubeSQLLibraryMBS module! The value can contain multiple names and paths separated with ";" on Wi and ":" on macOS/Linux. The library extension on Mac is ".dylib", on Linux ".so" and on Windows You get this library with the CubeSQL download on their homepage. If no value is given, we default to Windows: cubeSQL_64bit.dll;cubesql.dll macOS: libcubesql_r.dylib Linux: libcubesql_r.so The default is set to work in most cases and try various possible library r
SQLiteInMemory	"Sqlite::memory:"	Connection string for SQLite for a new in-memory database. Prefixed with SQLite to use directly.

Isolation Levels

Constant	Value	Description
kANSILevel0	0	ANSI Level 0
kANSILevel1	1	ANSI Level 1
kANSILevel2	2	ANSI Level 2
kANSILevel3	3	ANSI Level 3
kLevelUnknown	-1	Unknown
kReadCommitted	1	Read committed.
kReadUncommitted	0	Read uncommitted.
kRepeatableRead	2	Repeatable read.
kSerializable	3	Serializable.
kSnapshot	4	Changes made in other transactions can not be seen. For Microsoft SQL Server.

Values for autocommit property

Constant	Value	Description
kAutoCommitOff	0	Autocommit is off.
kAutoCommitOn	1	Autocommit is on.
kAutoCommitUnknown	-1	Autocommit unknown

Command Types

Constant	Value	Description
kCommandTypeSQLStatement	1	Command is an SQL statement.
kCommandTypeSQLStatementRaw	2	Command is an SQL statement that mustn't be interpreted by SQLAPI.
kCommandTypeStoredProcedure	3	Command is a stored procedure or a function.
kCommandTypeUnknown	0	Used by default. Library detects command type automatically.

Error Codes

Constant	Value	Description
kErrorBindVarNotFound	7	Bind variable not found.
kErrorClientInitFails	6	Initialization failed for client.
kErrorClientNotSet	1	Client not set.
kErrorClientNotSupported	2	Unsupported client type for this platform.
kErrorClientVersionOld	5	Library file is too old.
kErrorFieldNotFound	8	Field not found.
kErrorGetLibraryVersionFails	4	Failed to query library version.
kErrorLoadLibraryFails	3	Failed to load a library. For example path could be wrong or 32/64bit mismatch.
kErrorNoMemory	0	Out of memory.
kErrorUnknownColumnType	11	Unknown column type.
kErrorUnknownDataType	9	Unknown data type.
kErrorUnknownParameterType	10	Unknown parameter type.
kErrorWrongConversion	12	Failed to convert a value, e.g. string to number.
kErrorWrongDatetime	13	Can't convert text to date.

Options to specify the library with SetFileOption

Constant	Value	Description
kOptionLibraryDB2	"DB2CLI.LIBS"	Tells the plugin where to find the library for DB2. The value can contain multiple names and paths separated with ";" o and ":" on macOS/Linux. The library extension on Mac is ".dylib", on Linux ".so" and on Win You get this library with the DB2 download on their homepage. If no value is given, we default to Windows: db2clio.dll macOS: libdb2.dylib Linux: libdb2.so
kOptionLibraryDuckDB	"DUCKDB.LIBS"	Tells the plugin where to find the library for DuckDB. The value can contain multiple names and paths separated with ";" o and ":" on macOS/Linux. The library extension on Mac is ".dylib", on Linux ".so" and on Win You get this library with the DuckDB download on their homepage If no value is given, we default to Windows: duckdb.dll macOS: libduckdb.dylib Linux: libduckdb.so
kOptionLibraryFirebird	"IBASE.LIBS"	Tells the plugin where to find the library for FireBird (or Interbase The value can contain multiple names and paths separated with ";" o and ":" on macOS/Linux. The library extension on Mac is ".dylib", on Linux ".so" and on Win You get this library with the Firebird download on their homepage If no value is given, we default to Windows: ibclient64.dll;fbclient.dll;gds32.dll macOS: libgds.dylib:libfbclient.dylib Linux: libgds.so:libfbclient.so The default is set to work in most cases and try various possible libr
kOptionLibraryInformix	"INFCLI.LIBS"	Tells the plugin where to find the library for Informix. The value can contain multiple names and paths separated with ";" o and ":" on macOS/Linux. The library extension on Mac is ".dylib", on Linux ".so" and on Win You get this library with the Informix download on their homepage If no value is given, we default to Windows: ICLIT09B.DLL;ICLIT09A.DLL macOS: iclit09b.dylib:iclit09a.dylib:libifcli.dylib Linux: iclit09b.so:iclit09a.so:libifcli.so The default is set to work in most cases and try various possible libr
kOptionLibraryInterbase	"IBASE.LIBS"	Tells the plugin where to find the library for FireBird (or Interbase The value can contain multiple names and paths separated with ";" o and ":" on macOS/Linux. The library extension on Mac is ".dylib", on Linux ".so" and on Win You get this library with the FireBird download on their homepage If no value is given, we default to Windows: ibclient64.dll;fbclient.dll;gds32.dll macOS: libgds.dylib:libfbclient.dylib Linux: libgds.so:libfbclient.so The default is set to work in most cases and try various possible libr
kOptionLibraryMySQL	"MYSQL.LIBS"	Tells the plugin where to find the library for MySQL (or MariaDB The value can contain multiple names and paths separated with ";" o and ":" on macOS/Linux. The library extension on Mac is ".dylib", on Linux ".so" and on Win You get this library with the MySQL download on their homepage Libs folder on our website. If no value is given, we default to Windows: libmySQL.dll;libmariadb.dll macOS: libmysqlclient.dylib.21:libmysqlclient_r.dyli client_r.18.dylib:libmysqlclient_r.16.dylib:libmysqlclient_r.15.dylib

5.17 class SQLDataConsumerMBS

5.17.1 class SQLDataConsumerMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for a data consumer.
Notes: If you query a clob/blob value, that value may not fit into memory, so you may prefer to get a callback for data and write it to a file in small chunks.
Blog Entries

- MBS Xojo / Real Studio Plugins, version 16.1pr4

Xojo Developer Magazine

- 14.1, page 30: The MBS SQL Plugin, An alternative way to connect to databases by Christian Schmitz

5.17.2 Events

5.17.3 Write(PieceType as Integer, data as string, Length as UInt32, BlobSize as UInt32)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The event called to process data.
Notes: PieceType is kOnePiece, kFirstPiece, kLastPiece or kNextPiece.
Data is the raw data in a binary string.
Length is the number of bytes and BlobSize the size of data blocks used.

5.17.4 Constants

Constants

Constant	Value	Description
kFirstPiece	1	One of the piece type constants. The first piece is processed. You may setup everything you need to handle the data.
kLastPiece	3	One of the piece type constants. The last piece is processed. You can close files/network connections.
kNextPiece	2	One of the piece type constants. The next piece is processed. Not the first one or the last one, but one between.
kOnePiece	4	One of the piece type constants. The whole data stream is delivered in one call of the event.

5.18 class SQLDataProviderMBS

5.18.1 class SQLDataProviderMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for a data provider.
Notes: Use this to set a blob/clob object with streaming data. For example if you want to add a 1 GB big file to the database without loading it into RAM in one piece, you can use this class to read it in small chunks.
Blog Entries

- MBS Xojo / Real Studio Plugins, version 16.1pr4

Xojo Developer Magazine

- 14.1, page 30: The MBS SQL Plugin, An alternative way to connect to databases by Christian Schmitz

5.18.2 Events

5.18.3 Read(byref PieceType as Integer, Length as UInt32) as string

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The event called whenever new data is needed.
Notes: PieceType is kOnePiece, kFirstPiece, kLastPiece or kNextPiece.
If your stream is on the end, you set this to kLastPiece.
Return the raw data in a string.
Length is the number of bytes.

5.18.4 Constants

Constants

Constant	Value	Description
kFirstPiece	1	One of the piece type constants. The first piece is processed. You may setup everything you need to handle the data.
kLastPiece	3	One of the piece type constants. The last piece is processed. You can close files/network connections.
kNextPiece	2	One of the piece type constants. The next piece is processed. Not the first one or the last one, but one between.
kOnePiece	4	One of the piece type constants. The whole data stream is delivered in one call of the event.

5.19 class SQLDateTimeMBS

5.19.1 class SQLDateTimeMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The SQL date/time value class.
Example:

```
Var d as new SQLDateTimeMBS(2008, 3, 4, 23, 10, 20)

MsgBox d.StringValue // shows "2008-03-04T23:10:20"
```

Notes: see also
https://www.sqlapi.com/ApiDoc/class_s_a_date_time.html
Blog Entries

- MBS Xojo Plugins, version 21.2pr3
- MBS Xojo Plugins, version 20.4pr8
- MBS Xojo Plugins, version 19.6pr1
- MonkeyBread Software Releases the MBS Xojo / Real Studio plug-ins in version 14.1
- MBS Xojo / Real Studio Plugins, version 14.1pr1

Xojo Developer Magazine

- 19.4, page 53: Maps,ÄîPart 10, Mapping GPS data with the MapKitMBS plugin by Markus Winter

5.19.2 Methods

5.19.3 Constructor(DateTimeValue as DateTime)

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Initializes the datetime value with the given DateTime object.
See also:

5.19.4 Constructor(DateValue as Date)

Plugin Version: 14.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Initializes the datetime value with the given date.
See also:

5.19.5 Constructor(Hour as Integer, Minute as Integer, Second as Integer = 0, Fraction as Integer = 0)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new SQL Datetime with the given values.
See also:

5.19.6 Constructor(other as SQLDateTimeMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a copy of the SQL Date Time.
Example:

```
Var d as new SQLDateTimeMBS(2008, 3, 4, 23, 10, 20)
Var e as new SQLDateTimeMBS(d)

MsgBox e.StringValue // shows "2008-03-04T23:10:20"
```

See also:

5.19.7 Constructor(StringValue as String)

Plugin Version: 21.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Initialized a date time value from a string.
Example:

```
Var d As New SQLDateTimeMBS("2021-04-12T18:46:16.242")
msgbox d.StringValue

Var d2 As New SQLDateTimeMBS("now")
```

MsgBox d2.StringValue

Notes: You can pass "now" for current time, a string containing a double value for seconds since epoche or a string formatted with SQL datetime. Either with T or space as separator. If fraction is given, it should be 3 digits for milliseconds.
See also:

- 5.19.3 Constructor(DateTimeValue as DateTime) 158
- 5.19.4 Constructor(DateValue as Date) 159
- 5.19.5 Constructor(Hour as Integer, Minute as Integer, Second as Integer = 0, Fraction as Integer = 0) 159
- 5.19.6 Constructor(other as SQLDateTimeMBS) 160
- 5.19.8 Constructor(value as Double) 161
- 5.19.9 Constructor(Year as Integer, Month as Integer, Day as Integer, Hour as Integer, Minute as Integer, Second as Integer = 0, Fraction as Integer = 0, TimeZone as String = "") 162
- 5.19.10 Constructor(Year as Integer, Month as Integer, Day as Integer, Hour as Integer, Minute as Integer, Second as Integer, TimeZone as String) 162

5.19.8 Constructor(value as Double)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new SQL date time value based on the double value.
Example:

```
Var d as new SQLDateTimeMBS(2008, 3, 4, 23, 10, 20)
Var e as new SQLDateTimeMBS(d.DoubleValue+1) // clone with one day more

MsgBox e.StringValue // shows "2008-03-05T23:10:20"
```

See also:

- 5.19.3 Constructor(DateTimeValue as DateTime) 158
- 5.19.4 Constructor(DateValue as Date) 159
- 5.19.5 Constructor(Hour as Integer, Minute as Integer, Second as Integer = 0, Fraction as Integer = 0) 159
- 5.19.6 Constructor(other as SQLDateTimeMBS) 160
- 5.19.7 Constructor(StringValue as String) 160
- 5.19.9 Constructor(Year as Integer, Month as Integer, Day as Integer, Hour as Integer, Minute as Integer, Second as Integer = 0, Fraction as Integer = 0, TimeZone as String = "") 162

5.19.9 Constructor(Year as Integer, Month as Integer, Day as Integer, Hour as Integer, Minute as Integer, Second as Integer = 0, Fraction as Integer = 0, TimeZone as String = "")

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new SQL Datetime with the given values.
Example:

```
Var d as new SQLDateTimeMBS(2008, 3, 4, 23, 10, 20)

MsgBox d.StringValue // shows "2008-03-04T23:10:20"
```

See also:

5.19.10 Constructor(Year as Integer, Month as Integer, Day as Integer, Hour as Integer, Minute as Integer, Second as Integer, TimeZone as String)

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new SQL Datetime with the given values.
See also:

5.19.11 Set(DateTimeValue as DateTime)

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Assigns the datetime value with the given dateTime object.
See also:

5.19.12 Set(value as Date)

Plugin Version: 14.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Assigns the datetime value with the given date.
See also:

5.19.13 Properties

5.19.14 DateTimeValue as DateTime

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries the value of this datetime object as a date object.
Notes: (Read only property)

5.19.15 DateValue as Date

Plugin Version: 14.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Queries the value of this datetime object as a date object.
Notes: (Read only property)

5.19.16 Day as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the day this SADateTime object represents (1 –31).
Notes: (Read only property)

5.19.17 DayOfWeek as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the day of the week this SADateTime object represents (Sunday = 1).
Notes: (Read only property)

5.19.18 DayOfYear as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the day of the year this SADateTime object represents (Jan 1 = 1).
Notes: (Read only property)

5.19.19 DoubleValue as Double

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The double value of this date/time.
Notes: Use these operators to get current date/time value using standard double representation. Days are represented by whole number increments starting with 30 December 1899, midnight as time zero. Hour values are expressed as the absolute value of the fractional part of the number.

Date and time	Representation
30 December 1899, midnight	0.00
1 January 1900, midnight	2.00
4 January 1900, midnight	5.00
4 January 1900, 6 A.M.	5.25
4 January 1900, noon	5.50
4 January 1900, 9 P.M.	5.875

(Read only property)

5.19.20 Fraction as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value of the fraction of second (0 to 999,999,999) this SQLDateTime object represents.
Notes: The value of the fraction field is the number of billionths of a second and ranges from 0 through 999,999,999 (1 less than 1 billion). For example, the value of the fraction field for a half-second is 500,000,000, for a thousandth of a second (one millisecond) is 1,000,000, for a millionth of a second (one microsecond) is 1,000, and for a billionth of a second (one nanosecond) is 1.
(Read only property)

5.19.21 hasDate as Boolean

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this date & time object contains a date.
Notes: (Read only property)

5.19.22 hasTime as Boolean

Plugin Version: 20.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this date & time object contains a time.
Notes: (Read only property)

5.19.23 Hour as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the hour this SADateTime object represents (0 –23).
Notes: (Read only property)

5.19.24 Minute as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the minute this SADateTime object represents (0 –59).
Notes: (Read only property)

5.19.25 Month as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the month this SADateTime object represents (1 –12).
Notes: (Read only property)

5.19.26 Second as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the second this SADateTime object represents (0 –59).
Notes: (Read only property)

5.19.27 StringValue as string

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The string value for this date/time.
Notes: (Read only property)

5.19.28 TimeZone as String

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The time zone.
Notes: Should be in format ”00:00” and should be supported for Oracle and Postgres.
(Read and Write property)

5.19.29 Year as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the year this SADateTime object represents.
Notes: (Read only property)

5.20 class SQLErrorExceptionMBS

5.20.1 class SQLErrorExceptionMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The error exception class to report SQL errors.
Notes: The SQLDatabaseMBS class sets its error properties on an error. All other SQL classes raise exceptions where you can check the message property.
Subclass of the RuntimeException class.
Blog Entries

- Load PDF from MS SQL Server and display it
- MonkeyBread Software Releases the MBS Xojo Plugins in version 19.1
- MBS Xojo Plugins, version 19.1pr2
- MonkeyBread Software Releases the MBS Xojo / Real Studio plug-ins in version 14.0
- MBS Real Studio Plugins, version 13.1pr4

Xojo Developer Magazine

- 14.1, page 27: The MBS SQL Plugin, An alternative way to connect to databases by Christian Schmitz

5.20.2 Properties

5.20.3 ErrorClass as Integer

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a class of error.
Notes: Returns one of the following values:

SA_No_Error	0	No error occurred.
SA_UserGenerated_Error	1	User-generated error.
SA_Library_Error	2	The Library error occurred.
SA_DBMS_API_Error	3	DBMS API error occurred.

A SQLErrorExceptionMBS object handles the next error classes:

- User-generated errors
- Library errors

- DBMS API errors

The Library errors are generated by the Library itself. It can be like detecting some mistake in passing arguments to the function or referencing the parameter with an inappropriate name. To get a Library-defined error text call ErrorMessage method.

The DBMS API errors come to the Library from the DBMS Client or Server. In this case the Library returns an error code and text Client- or Server-defined. To get error code and error text returned by the server call NativeError and ErrorMessage methods.

The User-generated exception is "SQLAPI++ compatible" exception thrown by the user. To throw user exception use throwUserException method.
(Read only property)

5.20.4 ErrorMessage as String

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets an error text.
Notes: A error text depends on a class of error.

If error class is Library error the ErrorMessage method returns Library-defined error text. If error class is DBMS API error the ErrorMessage method returns an error text gotten from DBMS Server or Client. If error class is User-defined error the ErrorMessage method returns an error text specified by user (see throwUserException method).

To get the error class call ErrorClass method.
(same as Message property)
(Read only property)

5.20.5 ErrorPosition as Integer

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets an error position in SQL statement.
Notes: Returns an integer value representing error position.
Not all DBMS servers allow to get error position. See Server specific notes to get detailed information about returned value.

If a command object's associated SQL statement contains any syntax errors, an exception will be thrown when you try to compile. ErrorPosition method returns the error position within the command string.

Server specific notes

DBMS server	ErrorPosition method returned value
Oracle	ErrorPosition returns parse error offset.
SQL Server	ErrorPosition returns the number of line within SQL statement where error occurred.
Sybase	ErrorPosition returns the number of line within SQL statement where error occurred.
DB2	ErrorPosition returns -1. DB2 does not support this function.
Informix	ErrorPosition returns -1. Informix does not support this function.
InterBase	ErrorPosition returns -1. InterBase does not support this function.
SQLBase	ErrorPosition returns character position of the syntax error within an SQL statement. The first character is position 0.
MySQL	ErrorPosition returns -1. MySQL does not support this function.
PostgreSQL	ErrorPosition returns -1. PostgreSQL does not support this function.
ODBC	ErrorPosition returns -1. ODBC does not support this function.

(Read only property)

5.20.6 NativeError as Integer

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Gets a native code associated with current error.
Notes: Returns an integer value represents a native code associated with current error.

If error class is DBMS API error the NativeError method returns error code received from DBMS Server or Client. If error class is User-defined error the NativeError method returns an error code specified by user (see throwUserException method). If error class is Library error the NativeError method returns -1.

To get the error class call ErrorClass method.

See server specific documentation to get more information about DBMS API error code.
(same as ErrorNumber property)
(Read only property)

5.20.7 SQL as String

Plugin Version: 19.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The SQL string set when this error occurred.
Notes: (Read only property)

5.21 class SQLFieldMBS

5.21.1 class SQLFieldMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: This is the class for a SQL field in a record.
Notes: Be aware that field objects exists only as long as their SQLCommand exists.

see also
https://www.sqlapi.com/ApiDoc/class_s_a_field.html
Subclass of the SQLValueReadMBS class.
Blog Entries

- MBS Xojo Plugins, version 19.5pr4
- MBS Xojo Plugins, version 19.2pr1
- MBS Xojo Plugins, version 18.4pr10
- MBS Xojo / Real Studio Plugins, version 16.4pr5
- Upcoming Changes for our SQL Plugin

5.21.2 Methods

5.21.3 ReadLongOrLob(toConsumer as SQLDataConsumerMBS, BlockSize as Integer)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Starts reading of Long or BLob(CLob) value using the given data consumer.
Notes: BlockSize: Size of piece of data you want to get to the consumer event.

After a command execution all output parameters are updated by their values, including Long and BLob(CLob) parameters. If you want to control piecewise reading of Long or BLob(CLob) data you should do the following:

Before a command execution set kLongOrLobReaderManual reading mode (see LongOrLobReaderMode) for Long or BLob(CLob) parameters you want to process by a data consumer. After that SQLAPI++ will skip reading output Long and BLob(CLob) parameters that you set to be read manually.
After command execution use ReadLongOrLob method for each output parameter defined to be read manually.
Note, that if the command has result set(s) (it is possible in some servers, see Server specific notes) then output parameters are available only after all result sets are completely processed using FetchNext method.
See also:

5.21.4 ReadLongOrLob(toFile as FolderItem)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Starts reading of Long or BLob(CLob) value to the given file.
Example:

```
Var cmd as SQLCommandMBS // your command
Var f as FolderItem = SpecialFolder.Desktop.Child("test.jpg")
Var field as SQLFieldMBS = cmd.Field("image")
// read blob content to binarystream
field.ReadLongOrLob(f)
```

Notes: May raise IOExceptions if things go wrong.
See also:

- 5.21.3 ReadLongOrLob(toConsumer as SQLDataConsumerMBS, BlockSize as Integer) 171
- 5.21.5 ReadLongOrLob(toStream as Writeable) 172

5.21.5 ReadLongOrLob(toStream as Writeable)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Starts reading of Long or BLob(CLob) value to the given writeable stream.
Example:

```
Var cmd as SQLCommandMBS // your command
Var f as FolderItem = SpecialFolder.Desktop.Child("test.jpg")
Var b as BinaryStream = BinaryStream.Create(f, true)
Var field as SQLFieldMBS = cmd.Field("image")
// read blob content to binarystream
field.ReadLongOrLob(b)
```

Notes: This allows you to read in chunks the data to a stream, e.g. binarystream, textoutputstream or socket.
See also:

- 5.21.3 ReadLongOrLob(toConsumer as SQLDataConsumerMBS, BlockSize as Integer) 171
- 5.21.4 ReadLongOrLob(toFile as FolderItem) 172

5.21.6 Properties

5.21.7 FieldNativeType as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Returns native type code of the field.
Notes: Deprecated. Please use NativeType property instead.
(Read only property)

5.21.8 FieldPrecision as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Returns precision of the field value (the total number of allowable digits).
Notes: Deprecated. Please use Precision property instead.
(Read only property)

5.21.9 FieldScale as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Returns scale of the field value (the number of digits to the right of the decimal point).
Notes: Deprecated. Please use Scale property instead.
(Read and Write property)

5.21.10 FieldSize as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Returns field data size.
Notes: Deprecated. Please use Size property instead.
(Read only property)

5.21.11 FieldType as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** Returns field data type.
Notes: Value is one of the kDataType* constants.
Deprecated. Please use Type property instead.
(Read and Write property)

5.21.12 isFieldRequired as boolean

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Shows if it is possible for the field value to be null.
Notes: Returns true if the field value can be null; false otherwise.
(Read only property)

5.21.13 Name as string

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns name of the field.
Notes: (Read only property)

5.21.14 NativeType as Integer

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns native type code of the field.
Notes: (Read only property)

5.21.15 Options as Dictionary

Plugin Version: 18.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a dictionary with all options.
Notes: For debugging, it may be useful to inspect options in debugger.
(Read only property)

5.21.16 Pos as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a one-based position of the field in a result set.
Notes: (Read only property)

5.21.17 Precision as Integer

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns precision of the field value (the total number of allowable digits).
Notes: (Read only property)

5.21.18 Scale as Integer

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns scale of the field value (the number of digits to the right of the decimal point).
Notes: (Read and Write property)

5.21.19 Size as Integer

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns field data size.
Notes: (Read only property)

5.21.20 Type as Integer

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns field data type.
Example:

```
Var db as SQLConnectionMBS
Var cmd as new SQLCommandMBS(db, "select * from test")

cmd.Execute

Var f as SQLFieldMBS = cmd.Field("test")

if f.Type = f.kDataTypeLong then
MsgBox "type is long"
end if
```

Notes: Value is one of the kDataType* constants.
(Read and Write property)

5.21.21 Option(name as string) as string

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The string value of a specific field option.
Notes: See for more details:
https://www.sqlapi.com/ApiDoc/class_s_a_field.html
(Read and Write computed property)

5.22 class SQLGlobalsMBS

5.22.1 class SQLGlobalsMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for the global methods of the SQL plugin.
Blog Entries

- News from the MBS Xojo Plugins in Version 25.1
- MBS Xojo Plugins, version 25.1pr4
- MBS Xojo Plugins, version 25.1pr3
- News from the MBS Xojo Plugins Version 21.1
- MBS Xojo Plugins, version 21.1pr2
- MBS Xojo Plugins, version 21.1pr1
- Edit and Update for SQLDatabaseMBS class
- Register MBS Xojo Plugins
- Accessing Microsoft SQL Database from Mac/Linux
- SQL Plugin and Textencoding

5.22.2 Methods

5.22.3 FindTableName(SQL as String) as String

Plugin Version: 21.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries table name used in a SQL SELECT statement.
Example:

```
Var sql As String = "SELECT First, Last FROM MyTable WHERE ID = 1"
Var TableName As String = SQLGlobalsMBS.FindTableName(sql)

MessageBox TableName // shows MyTable
```

Notes: This is a helper function used by our Edit/Update methods in RecordSet.
Returns either the found table name or empty text if something got wrong. e.g. multiple tables affected due to JOIN operation.

If you find a case where the parser crashes or reports wrong result, please contact us.

5.22.4 GetEnv(name as string) as string

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries environment variable.
Notes: Returns empty string if variable is undefined.

5.22.5 GetVersion as String

Plugin Version: 14.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The version string of the SQLAPI library.

5.22.6 GetVersionBuild as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The build number of the SQLAPI library.

5.22.7 GetVersionMajor as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The major version number of the SQLAPI library.

5.22.8 GetVersionMinor as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The minor version number of the SQLAPI library.

5.22.9 PutEnv(line as string) as boolean

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets environment variable.
Notes: Line format should be ”key=value” and returns true on success and false on failure.

5.22.10 RaiseException(message as string)

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Raises an exception to test exception handling in SQL Plugin.
Notes: Raises directly a SQLErrorExceptionMBS with the given message.

5.22.11 RaiseSQLErrorException(UserCode as Integer, message as string)

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Raises an exception to test exception handling in SQL Plugin.
Notes: Raises a SAUserException which the plugin catches and translates to a SQLErrorExceptionMBS in Xojo.

5.22.12 SetCurrentWorkingDirectory(path as folderitem) as boolean

Plugin Version: 16.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the current working directory.
Notes: This is often useful to make sure the DLLs in that folder are found.
See also:

- 5.22.13 SetCurrentWorkingDirectory(path as String) as boolean 179

5.22.13 SetCurrentWorkingDirectory(path as String) as boolean

Plugin Version: 16.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the current working directory.
Notes: This is often useful to make sure the DLLs in that folder are found.
See also:

- 5.22.12 SetCurrentWorkingDirectory(path as folderitem) as boolean 179

5.22.14 SetDllDirectory(path as folderitem) as boolean

Plugin Version: 25.1, Platform: Windows, Targets: All.

Function: Adds a directory to the search path used to locate DLLs for the application.
Notes: The directory to be added to the search path. If this parameter is an empty string (""), the call removes the current directory from the default DLL search order. If this parameter is nil, the function

restores the default search order.

If the function succeeds, the return value is true, otherwise false.

The SetDllDirectory function affects all subsequent calls to the LoadLibrary and LoadLibraryEx functions. It also effectively disables safe DLL search mode while the specified directory is in the search path.

After calling SetDllDirectory, the standard DLL search path is:

- The directory from which the application loaded.
- The directory specified by the SetDllDirectory function.
- The system directory. Use the GetSystemDirectory function to get the path of this directory. The name of this directory is System32.
- The Windows directory. Use the GetWindowsDirectory function to get the path of this directory.
- The directories that are listed in the PATH environment variable.

Each time the SetDllDirectory function is called, it replaces the directory specified in the previous SetDllDirectory call.

To revert to the standard search path used by LoadLibrary and LoadLibraryEx, call SetDllDirectory with nil. This also restores safe DLL search mode based on the SafeDllSearchMode registry value.
See also:

- 5.22.15 SetDllDirectory(path as String) as boolean 180

5.22.15 SetDllDirectory(path as String) as boolean

Plugin Version: 25.1, Platform: Windows, Targets: All.

Function: Adds a directory to the search path used to locate DLLs for the application.
Notes: The directory to be added to the search path. If this parameter is an empty string (""), the call removes the current directory from the default DLL search order. If this parameter is nil, the function restores the default search order.

If the function succeeds, the return value is true, otherwise false.

The SetDllDirectory function affects all subsequent calls to the LoadLibrary and LoadLibraryEx functions. It also effectively disables safe DLL search mode while the specified directory is in the search path.

After calling SetDllDirectory, the standard DLL search path is:

- The directory from which the application loaded.
- The directory specified by the SetDllDirectory function.
- The system directory. Use the GetSystemDirectory function to get the path of this directory. The name of this directory is System32.
- The Windows directory. Use the GetWindowsDirectory function to get the path of this directory.
- The directories that are listed in the PATH environment variable.

Each time the SetDllDirectory function is called, it replaces the directory specified in the previous SetDllDirectory call.

To revert to the standard search path used by LoadLibrary and LoadLibraryEx, call SetDllDirectory with nil. This also restores safe DLL search mode based on the SafeDllSearchMode registry value.
See also:

- 5.22.14 SetDllDirectory(path as folderitem) as boolean 179

5.22.16 SetEnv(name as string, value as string) as boolean

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets an environment variable.
Notes: Existing variable with same name will be overwritten.
Returns true on success and false on failure.

5.22.17 SetLicenseCode(n as string, enddate as Integer, v1 as Integer, v2 as Integer)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Registeres the SQL plugin and library.
Notes: Once you ordered a license, you receive details on how to call this method.

5.22.18 Setlocale(category as Integer, locale as string)

Plugin Version: 10.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the locale to use.
Example:

```
// ask for English for USA with 1252 codepage
SQLGlobalsMBS.Setlocale SQLGlobalsMBS.LocaleAll, "English_United States.1252"
```

Notes: The Setlocale function sets the C library's notion of natural language formatting style for particular sets of routines. Each such style is called a 'locale' and is invoked using an appropriate name passed as a C string.

The setlocale() function recognizes several categories of routines. These are the categories and the sets of routines they select:

LocaleAll	Set the entire locale generically.
LocaleCollate	Set a locale for string collation routines. This controls alphabetic ordering in strcoll() and strxfrm().
LocaleCType	Set a locale for the ctype and multibyte functions. This controls recognition of upper and lower case, alphabetic or non-alphabetic characters, and so on.
LocaleMessages	Set a locale for message catalogs, see catopen function.
LocaleMonetary	Set a locale for formatting monetary values; this affects the localeconv() function.
LocaleNumeric	Set a locale for formatting numbers. This controls the formatting of decimal points in input and output of floating point numbers in functions such as printf() and scanf(), as well as values returned by localeconv().
LocaleTime	Set a locale for formatting dates and times using the strftime() function.

Only three locales are defined by default: the empty string "" (which denotes the native environment) and the "C" and "POSIX" locales (which denote the C-language environment). By default, C programs start in the "C" locale.

5.22.19 UnInitialize

Plugin Version: 21.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Explicitly shutdown SQL engine.
Notes: You can call this in App Close or Destructor, but only after you destroyed all Connections, e.g. all SQLDatabaseMBS and SQLConnectionMBS objects.

5.22.20 UnSetEnv(name as string) as boolean

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Removes environment variable.
Notes: Returns true on success and false on failure.

5.22.21 Events

5.22.22 Trace(traceInfo as Integer, SQL as string, Connection as SQLConnectionMBS, Command as SQLCommandMBS)

Plugin Version: 13.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The event to trace SQL commands.

5.22.23 Constants

Constants

Constant	Value	Description
LocaleAll	0	One of the locale category constants for SetLocale. Set the entire locale generically.
LocaleCollate	1	One of the locale category constants for SetLocale. Set a locale for string collation routines. This controls alphabetic ordering in strcoll() and strxfrm().
LocaleCType	2	One of the locale category constants for SetLocale. et a locale for the ctype(3) and multibyte(3) functions. This controls recognition of upper and lower case, alphabetic or non-alphabetic characters, and so on.
LocaleMessages	6	One of the locale category constants for SetLocale. Set a locale for message catalogs, see catopen(3) function.
LocaleMonetary	3	One of the locale category constants for SetLocale. Set a locale for formatting monetary values; this affects the localeconv() function.
LocaleNumeric	4	One of the locale category constants for SetLocale. Set a locale for formatting numbers. This controls the formatting of decimal points in input and output of floating point numbers in functions such as printf() and scanf(), as well as values returned by localeconv().
LocaleTime	5	One of the locale category constants for SetLocale. Set a locale for formatting dates and times using the strftime() function.

5.23 class SQLIntervalMBS

5.23.1 class SQLIntervalMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class in the SQL Plugin for an interval.

5.23.2 Methods

5.23.3 Constructor

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new zero interval.
See also:

- 5.23.4 Constructor(days as Integer, hours as Integer, minutes as Integer, seconds as Integer = 0, NanoSeconds as Integer = 0) 184
- 5.23.5 Constructor(value as Double) 184

5.23.4 Constructor(days as Integer, hours as Integer, minutes as Integer, seconds as Integer = 0, NanoSeconds as Integer = 0)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new interval with the given values.
See also:

- 5.23.3 Constructor 184
- 5.23.5 Constructor(value as Double) 184

5.23.5 Constructor(value as Double)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new interval with the given time delta.
Example:

```
Var n as new SQLIntervalMBS(5)

MsgBox n.StringValue // shows "120:00:00" for 120 hours
```

See also:

- 5.23.3 Constructor 184
- 5.23.4 Constructor(days as Integer, hours as Integer, minutes as Integer, seconds as Integer = 0, NanoSeconds as Integer = 0) 184

5.23.6 Dec(interval as SQLIntervalMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Decrements the interval.

5.23.7 Inc(interval as SQLIntervalMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Increments the interval.

5.23.8 SetInterval(days as Integer, hours as Integer, minutes as Integer, seconds as Integer = 0, NanoSeconds as Integer = 0)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the interval values.

5.23.9 Properties

5.23.10 Days as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The days in this interval.
Notes: (Read only property)

5.23.11 DoubleValue as Double

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The value of this interval.
Example:

```
Var n as new SQLIntervalMBS(1,2,3,4)

MsgBox str(n.DoubleValue) // shows "1.085463" for 1 day, 2 hours, 3 minutes and 4 seconds
```

Notes: (Read only property)

5.23.12 Fraction as Integer

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: The fraction of a second in nano seconds.
Notes: Range 0..999999999.
(Read only property)

5.23.13 Hours as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The hours value.
Notes: (Read only property)

5.23.14 Minutes as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The minutes value.
Notes: (Read only property)

5.23.15 Seconds as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The seconds value.
Notes: (Read only property)

5.23.16 StringValue as string

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The interval as a string.
Example:

```
Var n as new SQLIntervalMBS(5)

MsgBox n.StringValue // shows "120:00:00" for 120 hours
```

Notes: (Read only property)

5.23.17 TotalDays as Double

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The total days value.
Example:

```
Var n as new SQLIntervalMBS(1,2,3,4)

MsgBox str(n.TotalDays) // shows "1"
```

Notes: (Read only property)

5.23.18 TotalHours as Double

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The total hours value.
Example:

```
Var n as new SQLIntervalMBS(1,2,3,4)

MsgBox str(n.TotalHours) // shows "26"
```

Notes: (Read only property)

5.23.19 TotalMinutes as Double

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The total minutes value.
Example:

```
Var n as new SQLIntervalMBS(1,2,3,4)
```

MsgBox str(n.TotalMinutes) // shows ”1563”

Notes: (Read only property)

5.23.20 TotalSeconds as Double

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The total seconds value.
Notes: Var n as new SQLIntervalMBS(1,2,3,4)

MsgBox str(n.GetTotalSeconds) // shows ”93784”
(Read only property)

5.24 class SQLite3BackupMBS

5.24.1 class SQLite3BackupMBS

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The object for a running backup.
Notes: The backup object records state information about an ongoing online backup operation. The sqlite3_backup object is created by a call to BackupInit() and is destroyed by a call to BackupFinish().
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
Blog Entries

- SQLite Backup Functions

Xojo Developer Magazine

- 14.1, page 30: The MBS SQL Plugin, An alternative way to connect to databases by Christian Schmitz

5.24.2 Methods

5.24.3 Constructor

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.

5.24.4 Properties

5.24.5 Handle as Integer

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The internal object reference.
Notes: (Read only property)

5.25 class SQLite3MBS

5.25.1 class SQLite3MBS

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. **Function:** The class for the native SQLite API.
Example:

```
Var con as new SQLConnectionMBS
Var path as string = "/tmp/test.db" // change path for Windows!

con.Connect(path,"","",SQLConnectionMBS.kSQLiteClient)

Var api as SQLAPIMBS = con.NativeAPI
if api isa SQLite3MBS then
Var s as SQLite3MBS = SQLite3MBS(api)
MsgBox s.Version
end if
```

Notes: Deprecated in favor of direct methods on SQLConnectionMBS. Please let us know if you need more SQLite specific functions.
Subclass of the SQLAPIMBS class.
Blog Entries

- News from the MBS Xojo Plugins Version 22.2
- MBS Xojo Plugins, version 22.2pr1
- MBS Xojo Plugins, version 19.5pr1
- MonkeyBread Software Releases the MBS Xojo Plugins in version 17.4
- MBS Releases the MBS Xojo / Real Studio plug-ins in version 16.4
- MBS Releases the MBS Xojo / Real Studio plug-ins in version 16.1
- Embedded SQLite and encryption
- [ANN] MonkeyBread Software Releases the MBS Xojo / Real Studio plug-ins in version 14.4
- SQLite Backup Functions
- MBS REALbasic plug-ins version 9.5

Videos

- Presentation from London conference about MBS Plugins.

5.25.2 Methods

5.25.3 BackupFinish(Backup as SQLite3BackupMBS) as Integer

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Finishes a backup run.
Notes: When BackupStep has returned kErrorDone, or when the application wishes to abandon the backup operation, the application should destroy the SQLite3BackupMBS by passing it to BackupFinish. The BackupFinish interfaces releases all resources associated with the SQLite3BackupMBS object. If BackupStep has not yet returned kErrorDone, then any active write-transaction on the destination database is rolled back. The SQLite3BackupMBS object is invalid and may not be used following a call to BackupFinish.
The value returned by BackupFinish is kErrorOK if no BackupStep errors occurred, regardless or whether or not BackupStep completed. If an out-of-memory condition or IO error occurred during any prior BackupStep call on the same SQLite3BackupMBS object, then BackupFinish returns the corresponding error code.
A return of kErrorBusy or kErrorLocked from BackupStep is not a permanent error and does not affect the return value of BackupFinish.

5.25.4 BackupInit(Dest as Variant, DestName as String, Source as Variant, SourceName as String) as SQLite3BackupMBS

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Initializes a backup.
Notes: The backup API copies the content of one database into another. It is useful either for creating backups of databases or for copying in-memory databases to or from persistent files.

see also
http://www.sqlite.org/c3ref/backup_finish.html

Exclusive access is required to the destination database for the duration of the operation. However the source database is only read-locked while it is actually being read; it is not locked continuously for the entire backup operation. Thus, the backup may be performed on a live source database without preventing other users from reading or writing to the source database while the backup is underway.

To perform a backup operation:

- BackupInit is called once to initialize the backup,
- BackupStep is called one or more times to transfer the data between the two databases, and finally
- BackupFinish is called to release all resources associated with the backup operation.

There should be exactly one call to BackupFinish for each successful call to BackupInit.

The D and N arguments to BackupInit(D,N,S,M) are the database connection associated with the destination database and the database name, respectively. The database name is ”main” for the main database, ”temp” for the temporary database, or the name specified after the AS keyword in an ATTACH statement for an attached database. The S and M arguments passed to BackupInit(D,N,S,M) identify the database connection and database name of the source database, respectively. The source and destination database connections (parameters S and D) must be different or else BackupInit(D,N,S,M) will file with an error.
If an error occurs within BackupInit(D,N,S,M), then nil is returned and an error code and error message are store3d in the destination database connection D. The error code and message for the failed call to BackupInit can be retrieved using the ErrCode and ErrMessage functions. A successful call to BackupInit returns a SQLite3BackupMBS object. The SQLite3BackupMBS object may be used with the BackupStep and BackupFinish functions to perform the specified backup operation.

Concurrent Usage of Database Handles
The source database connection may be used by the application for other purposes while a backup operation is underway or being initialized. If SQLite is compiled and configured to support threadsafe database connections, then the source database connection may be used concurrently from within other threads.
However, the application must guarantee that the destination database connection is not passed to any other API (by any thread) after BackupInit is called and before the corresponding call to BackupFinish. SQLite does not currently check to see if the application incorrectly accesses the destination database connection and so no error code is reported, but the operations may malfunction nevertheless. Use of the destination database connection while a backup is in progress might also also cause a mutex deadlock.
If running in shared cache mode, the application must guarantee that the shared cache used by the destination database is not accessed while the backup is running. In practice this means that the application must guarantee that the disk file being backed up to is not accessed by any connection within the process, not just the specific connection that was passed to BackupInit.
The SQLite3BackupMBS object itself is partially threadsafe. Multiple threads may safely make multiple concurrent calls to BackupStep. However, the BackupRemaining and BackupPageCount APIs are not strictly speaking threadsafe. If they are invoked at the same time as another thread is invoking BackupStep it is possible that they return invalid values.

Source and Dest can be SQLConnectionMBS or SQLDatabaseMBS. You need to pass source and dest, even if one is self as we give you the option to decide where to pass the current database connection.

5.25.5 BackupPageCount(Backup as SQLite3BackupMBS) as Integer

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of pages in total.
Notes: Each call to BackupStep sets two values inside the SQLite3BackupMBS object: the number of pages still to be backed up and the total number of pages in the source databae file. The BackupRemaining and BackupPageCount interfaces retrieve these two values, respectively.
The values returned by these functions are only updated by BackupStep. If the source database is modified

during a backup operation, then the values are not updated to account for any extra pages that need to be updated or the size of the source database file changing.

5.25.6 BackupRemaining(Backup as SQLite3BackupMBS) as Integer

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of pages remaining.
Notes: Each call to BackupStep sets two values inside the SQLite3BackupMBS object: the number of pages still to be backed up and the total number of pages in the source databae file. The BackupRemaining and BackupPageCount interfaces retrieve these two values, respectively.
The values returned by these functions are only updated by BackupStep. If the source database is modified during a backup operation, then the values are not updated to account for any extra pages that need to be updated or the size of the source database file changing.

5.25.7 BackupStep(Backup as SQLite3BackupMBS, Pages as Integer) as Integer

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies up to Pages pages between the source and destination databases specified by SQLite3BackupMBS object.
Notes: If N is negative, all remaining source pages are copied. If BackupStep(B,N) successfully copies N pages and there are still more pages to be copied, then the function resturns kErrorOK. If BackupStep(B,N) successfully finishes copying all pages from source to destination, then it returns kErrorDone. If an error occurs while running BackupStep(B,N), then an error code is returned. As well as kErrorOK and kErrorDone, a call to BackupStep may return kErrorReadOnly, kErrorNoMem, kErrorBusy, kErrorLocked, or an kErrorIOACCESS | kErrorIOXXX extended error code.
The BackupStep might return kErrorReadOnly if the destination database was opened read-only or if the destination is an in-memory database with a different page size from the source database.
If BackupStep cannot obtain a required file-system lock, then the sqlite3_busy_handler | busy-handler function is invoked (if one is specified). If the busy-handler returns non-zero before the lock is available, then kErrorBusy is returned to the caller. In this case the call to BackupStep can be retried later. If the source database connection is being used to write to the source database when BackupStep is called, then kErrorLocked is returned immediately. Again, in this case the call to BackupStep can be retried later on. (If kErrorIOACCESS | kErrorIOXXX, kErrorNoMem, or kErrorReadOnly is returned, then there is no point in retrying the call to BackupStep. These errors are considered fatal.) The application must accept that the backup operation has failed and pass the backup operation handle to the BackupFinish to release associated resources.
The first call to BackupStep obtains an exclusive lock on the destination file. The exclusive lock is not released until either BackupFinish is called or the backup operation is complete and BackupStep returns kErrorDone. Every call to BackupStep obtains a shared lock on the source database that lasts for the duration of the BackupStep call. Because the source database is not locked between calls to BackupStep, the source database may be modified mid-way through the backup process. If the source database is modified

by an external process or via a database connection other than the one being used by the backup operation, then the backup will be automatically restarted by the next call to BackupStep. If the source database is modified by the using the same database connection as is used by the backup operation, then the backup database is automatically updated at the same time.

5.25.8 EnableLoadExtension(OnOff as boolean)

Plugin Version: 14.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Enables/disables extension loading for the given connection.

5.25.9 ErrCode as Integer

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The ErrCode function returns the numeric result code or extended result code for the most recent failed sqlite3 API call associated with a database connection.
Notes: If a prior API call failed but the most recent API call succeeded, the return value from ErrCode is undefined.

5.25.10 ErrMessage as string

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the most recent error message in english for the given connection.

5.25.11 LastInsertRowID as Int64

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns Last Insert Rowid.
Notes: Each entry in an SQLite table has a unique 64-bit signed integer key called the ROWID. The rowid is always available as an undeclared column named ROWID, OID, or _ROWID_ as long as those names are not also used by explicitly declared columns. If the table has a column of type INTEGER PRIMARY KEY then that column is another alias for the rowid.

This routine returns the rowid of the most recent successful INSERT into the database from the database connection in the first argument. If no successful INSERTs have ever occurred on that database connection, zero is returned.

(If an INSERT occurs within a trigger, then the rowid of the inserted row is returned by this routine as long as the trigger is running. But once the trigger terminates, the value returned by this routine reverts to the last value inserted before the trigger fired.)

An INSERT that fails due to a constraint violation is not a successful INSERT and does not change the value returned by this routine. ^Thus INSERT OR FAIL, INSERT OR IGNORE, INSERT OR ROLLBACK, and INSERT OR ABORT make no changes to the return value of this routine when their insertion fails. ^(When INSERT OR REPLACE encounters a constraint violation, it does not fail. The INSERT continues to completion after deleting rows that caused the constraint problem so INSERT OR REPLACE will always change the return value of this interface.)^

For the purposes of this routine, an INSERT is considered to be successful even if it is subsequently rolled back.

This function is accessible to SQL statements via the last_insert_rowid() SQL function.

If a separate thread performs a new INSERT on the same database connection while the LastInsertRowID function is running and thus changes the last insert rowid, then the value returned by LastInsertRowID is unpredictable and might not equal either the old or the new last insert rowid.

5.25.12 LoadExtension(file as FolderItem, ByRef ErrorMessage as String) as Integer

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads an SQLite extension library from the named file.
Notes: The LoadExtension interface attempts to load an SQLite extension library contained in the file.

Returns kErrorOk on success and kErrorError if something goes wrong.
Extension loading must be enabled using EnableLoadExtension prior to calling this API, otherwise an error will be returned.
See also:

- 5.25.13 LoadExtension(path as String, ByRef ErrorMessage as String) as Integer 195

5.25.13 LoadExtension(path as String, ByRef ErrorMessage as String) as Integer

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Loads an SQLite extension library from the named file.
Notes: The LoadExtension interface attempts to load an SQLite extension library contained in the file.

Returns kErrorOk on success and kErrorError if something goes wrong.
Extension loading must be enabled using EnableLoadExtension prior to calling this API, otherwise an error will be returned.
See also:

- 5.25.12 LoadExtension(file as FolderItem, ByRef ErrorMessage as String) as Integer 195

5.25.14 MemoryHighwater(reset as boolean) as Int64

Plugin Version: 17.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries maximum memory usage so far.
Notes: Can be reset with reset parameter being true.
See also:

- 5.25.23 MemoryHighwater as Int64 200

5.25.15 ReKey(Key as String) as Integer

Plugin Version: 15.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: You can change the key on a database using the Rekey Function.
Notes: An empty key decrypts the database.

Rekeying requires that every page of the database file be read, decrypted, reencrypted with the new key, then written out again. Consequently, rekeying can take a long time on a larger database.

Most SEE variants allow you to encrypt an existing database that was created using the public domain version of SQLite. This is not possible when using the authenticating version of the encryption extension in see-aes128-ccm. If you do encrypt a database that was created with the public domain version of SQLite, no nonce will be used and the file will be vulnerable to a chosen-plaintext attach. If you call SetKey() immediately after Open when you are first creating the database, space will be reserved in the database for a nonce and the encryption will be much stronger. If you do not want to encrypt right away, call SetKey() anyway, with an empty key, and the space for the nonce will be reserved in the database even though no encryption is done initially.

A public domain version of the SQLite library can read and write an encrypted database with an empty key. You only need the encryption extension if the key is non-empty.

Returns a SQLite error code.

5.25.16 SetBusyHandler(MaxAttempts as Integer = 5)

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Installs busy handler for this connection.
Notes: This routine sets a callback function that might be invoked whenever an attempt is made to open a database table that another thread or process has locked.

The plugin has an busy handler which will wait up to MaxAttemps and yield to other Xojo threads while waiting.
Passing 5 should wait up to 100ms.

There can only be a single busy handler defined for each [database connection] . Setting a new busy handler clears any previously set handler.) Note that calling SetBusyTimeout will also set or clear the busy handler.

The busy callback should not take any actions which modify the database connection that invoked the busy handler. Any such actions result in undefined behavior.

5.25.17 SetBusyTimeout(TimeOutMS as Integer = 20)

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: This routine sets a busy handler that sleeps for a specified amount of time when a table is locked.
Notes: The handler will sleep multiple times until at least "ms" milliseconds of sleeping have accumulated. ^After at least "ms" milliseconds of sleeping, the handler returns 0 which causes SQLite query to return SQLite Busy or IO Blocked error.

Calling this routine with an argument less than or equal to zero turns off all busy handlers.

(There can only be a single busy handler for a particular database connection any any given moment. If another busy handler was defined (using SetBusyHandler prior to calling this routine, that other busy handler is cleared.)

5.25.18 SetKey(Key as String) as Integer

Plugin Version: 15.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Applies encryption to a database connection.
Notes: Returns a SQLite error code.

The amount of key material actually used by the encryption extension depends on which variant of SEE you

are using. With RC4, the first 256 byte of key are used. With the AES128, the first 16 bytes of the key are used. With AES256, the first 32 bytes of key are used.

If you specify a key that is shorter than the maximum key length, then the key material is repeated as many times as necessary to complete the key. If you specify a key that is larger than the maximum key length, then the excess key material is silently ignored.

The key must begin with an ASCII prefix to specify which algorithm to use. The prefix must be one of "rc4:", "aes128:", or "aes256:". The prefix is not used as part of the key sent into the encryption algorithm. So the real key should begin on the first byte after the prefix.

The string provided to the plugin is used with it's current encoding. So be sure you use right text encoding for what you want. e.g. using "M√ºller" as key in text encoding Windows ANSI will not open a database which used that key in UTF-8 encoding.

The Xojo database encryption in SQLiteDatabase class uses AES-128 OFB.

5.25.19 TableColumnMetaData(DBName as string, TableName as string, ColumnName as string, byref DataType as string, byref CollationSequence as string, byref NotNull as boolean, byref PrimaryKey as boolean, byref AutoIncrement as Boolean) as Integer

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Extract Metadata About A Column Of A Table
Notes: Not available in all sqlite libraries!

This routine returns metadata about a specific column of a specific database table accessible using the database connection handle passed as the first function argument.
The column is identified by the second, third and fourth parameters to this function. The second parameter is either the name of the database (i.e. "main", "temp", or an attached database) containing the specified table or NULL. If it is NULL, then all attached databases are searched for the table using the same algorithm used by the database engine to resolve unqualified table references.
The third and fourth parameters to this function are the table and column name of the desired column, respectively. Neither of these parameters may be NULL.
Metadata is returned by writing to the memory locations passed as the 5th and subsequent parameters to this function. Any of these arguments may be NULL, in which case the corresponding element of metadata is omitted.

CollationSequence is assigned the Name of default collation sequence. NotNull is set to true if column has a NOT NULL constraint. PrimaryKey is set to true if column is part of the PRIMARY KEY and AutoIncrement is set to true if column is AUTOINCREMENT.

If the specified table is actually a view, an error code is returned.

If the specified column is "rowid", "oid" or "_rowid_" and an INTEGER PRIMARY KEY column has been explicitly declared, then the output parameters are set for the explicitly declared column. (If there is no explicitly declared INTEGER PRIMARY KEY column, then the output parameters are set as follows:

data type: "INTEGER"
collation sequence: "BINARY"
not null: false
primary key: true
auto increment: false

(This function may load one or more schemas from database files. If an error occurs during this process, or if the requested table or column cannot be found, an error code is returned and an error message left in the database connection (to be retrieved using ErrMessage).)

This API is only available if the library was compiled with the SQLITE_ENABLE_COLUMN_METADATA C-preprocessor symbol defined.

5.25.20 Threadsafe as Integer

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Test To See If The Library Is Threadsafe.
Notes: The threadsafe() function returns zero if and only if SQLite was compiled mutexing code omitted due to the SQLITE_THREADSAFE compile-time option being set to 0.

SQLite can be compiled with or without mutexes. When the SQLITE_THREADSAFE C preprocessor macro is 1 or 2, mutexes are enabled and SQLite is threadsafe. When the SQLITE_THREADSAFE macro is 0, the mutexes are omitted. Without the mutexes, it is not safe to use SQLite concurrently from more than one thread.

Enabling mutexes incurs a measurable performance penalty. So if speed is of utmost importance, it makes sense to disable the mutexes. But for maximum safety, mutexes should be enabled. The default behavior is for mutexes to be enabled.

This interface can be used by an application to make sure that the version of SQLite that it is linking against was compiled with the desired setting of the SQLITE_THREADSAFE macro.

This interface only reports on the compile-time mutex setting of the SQLITE_THREADSAFE flag. If SQLite

is compiled with SQLITE_THREADSAFE=1 or =2 then mutexes are enabled by default but can be fully or partially disabled using a call to sqlite3_config() with the verbs SQLITE_CONFIG_SINGLETHREAD, SQLITE_CONFIG_MULTITHREAD, or SQLITE_CONFIG_MUTEX. ^(The return value of the sqlite3_threadsafe() function shows only the compile-time setting of thread safety, not any run-time changes to that setting made by sqlite3_config(). In other words, the return value from sqlite3_threadsafe() is unchanged by calls to sqlite3_config().)^

See the threading mode documentation for additional information.

5.25.21 Properties

5.25.22 ConnectionHandle as Ptr

Plugin Version: 22.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the current connection reference for the database.
Notes: sqlite3 pointer for using in declares.
(Read only property)

5.25.23 MemoryHighwater as Int64

Plugin Version: 17.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries maximum memory usage so far.
Notes: Can be reset with reset parameter.
(Read only property)
See also:

- 5.25.14 MemoryHighwater(reset as boolean) as Int64 196

5.25.24 MemoryUsed as Int64

Plugin Version: 17.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Queries memory in use by SQLite.
Notes: This is memory allocated, but not yet freed.
Value is zero until SQLite3 initialized.
(Read only property)

5.25.25 Version as string

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The version string of the SQLite library.
Notes: (Read only property)

5.25.26 VersionNumber as Integer

Plugin Version: 9.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: The version number of the SQLite library.
Notes: (Read only property)

5.25.27 Constants

Error Codes

Constant	Value	Description
kErrorAbort	4	Callback routine requested an abort.
kErrorAuth	23	Authorization denied
kErrorBusy	5	The database file is locked.
kErrorCantopen	14	Unable to open the database file.
kErrorConstraint	19	Abort due to constraint violation.
kErrorCorrupt	11	The database disk image is malformed.
kErrorDone	101	sqlite3_step() has finished executing.
kErrorEmpty	16	Database is empty
kErrorError	1	SQL error or missing database.
kErrorFormat	24	Auxiliary database format error.
kErrorFull	13	Insertion failed because database is full.
kErrorInternal	2	Internal logic error in SQLite
kErrorInterrupt	9	Operation terminated by sqlite3_interrupt().
kErrorIoerr	10	Some kind of disk I/O error occurred.
kErrorLocked	6	A table in the database is locked.
kErrorMismatch	20	Data type mismatch.
kErrorMisuse	21	Library used incorrectly.
kErrorNolfs	22	Uses OS features not supported on host.
kErrorNoMem	7	Out of memory.
kErrorNotaDB	26	File opened that is not a database file.
kErrorNotFound	12	NOT USED. Table or record not found.
kErrorOk	0	Successful result
kErrorPerm	3	Access permission denied.
kErrorProtocol	15	NOT USED. Database lock protocol error.
kErrorRange	25	2nd parameter to sqlite3_bind out of range.
kErrorReadonly	8	Attempt to write a readonly database.
kErrorRow	100	sqlite3_step() has another row ready.
kErrorSchema	17	The database schema changed.
kErrorToobig	18	String or BLOB exceeds size limit.

5.26 class SQLiteFunctionMBS

5.26.1 class SQLiteFunctionMBS

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The class for a custom SQLite function implemented in Xojo.
Example:

```
// Create a custom function with a handler function:
Var f As New SQLiteFunctionMBS
f.name = "SHA256"
f.ArgumentCount = 1
f.Flags = f.kFlagDeterministic OR f.kFlagUTF8 OR f.kFlagInnocuous
// Text encoding UTF8, function is deterministic and can be cached, function is innocuous as it depends only on parameters
AddHandler f.Perform, AddressOf PerformSHA256
functions.Append f // keep reference
```

Notes: You can use this with our InternalSQLiteLibraryMBS module and the SQL classes within MBS Plugin to work on SQLite databases and as extension for SQLiteDatabase in Xojo.
Blog Entries

- 15th birthday of MBS SQL Plugin
- News from the MBS Xojo Plugins Version 24.1
- MonkeyBread Software Releases the MBS Xojo Plugins in version 24.1
- Did you know that you can load extensions in SQLite?
- MBS Xojo Plugins, version 24.1pr1

5.26.2 Methods

5.26.3 Constructor

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The constructor.
Example:

```
Var f As New SQLiteFunctionMBS
f.name = "SHA256"
```

Notes: Please keep a reference to the function in a global array to avoid it being destroyed.

5.26.4 Destructor

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The destructor.

5.26.5 ResultBlob(data as MemoryBlock)

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Returns a BLOB value to SQLite.
Notes: We have three variants, so you can pass string, MemoryBlock or ptr.
See also:

- 5.26.6 ResultBlob(data as ptr, size as Integer) 204
- 5.26.7 ResultBlob(text as string) 204

5.26.6 ResultBlob(data as ptr, size as Integer)

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Returns a BLOB value to SQLite.
Notes: We have three variants, so you can pass string, MemoryBlock or ptr.
See also:

- 5.26.5 ResultBlob(data as MemoryBlock) 204
- 5.26.7 ResultBlob(text as string) 204

5.26.7 ResultBlob(text as string)

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Returns a BLOB value to SQLite.
Example:

me.ResultBlob "Hello World"

Notes: We have three variants, so you can pass string, MemoryBlock or ptr.
See also:

- 5.26.5 ResultBlob(data as MemoryBlock) 204
- 5.26.6 ResultBlob(data as ptr, size as Integer) 204

5.26.8 ResultDouble(value as Double)

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Returns a double value back to SQLite.
Example:

```
Sub Perform(ArgumentCount as Integer, Arguments() as Variant) Handles Perform
#Pragma BackgroundTasks False
#Pragma BoundsChecking False
#Pragma StackOverflowChecking False

// square function
Var a as double = arguments(0).DoubleValue
ResultDouble a * a

End Sub
```

5.26.9 ResultError(ErrorMessage as string)

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Returns an error message for the function call.
Example:

```
Sub Perform(ArgumentCount as Integer, Arguments() as Variant) Handles Perform
#Pragma BackgroundTasks False
#Pragma BoundsChecking False
#Pragma StackOverflowChecking False

Var v as string = arguments(0)
if v.length = 0 then
ResultError "Please pass text"
ResultErrorCode 12345 // custom error code
else
ResultText v + v
end if
End Sub
```

Notes: Sets error code to 1. Use ResultErrorCode to change this.

5.26.10 ResultErrorCode(ErrorCode as Integer)

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Sets error code for the function call.

5.26.11 ResultInteger(value as Integer)

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Returns an integer value back to SQLite.
Example:

```
Sub Perform(ArgumentCount as Integer, Arguments() as Variant) Handles Perform
#Pragma BackgroundTasks False
#Pragma BoundsChecking False
#Pragma StackOverflowChecking False

// square function
Var a as Integer = arguments(0).IntegerValue
ResultInteger a * a

End Sub
```

5.26.12 ResultNull

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Returns a null value back to SQLite.
Example:

```
Sub Perform(ArgumentCount as Integer, Arguments() as Variant) Handles Perform
#Pragma BackgroundTasks False
#Pragma BoundsChecking False
#Pragma StackOverflowChecking False

// return NULL value
ResultNull

End Sub
```

5.26.13 ResultText(text as string)

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Returns a string value back to SQLite.
Example:

```
Sub Perform(ArgumentCount as Integer, Arguments() as Variant) Handles Perform
#Pragma BackgroundTasks False
#Pragma BoundsChecking False
#Pragma StackOverflowChecking False

ResultText "Hello World"

End Sub
```

5.26.14 ResultZeroBlob(Length as Integer)

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Returns a zero filled BLOB value back to SQLite.

5.26.15 Properties

5.26.16 ArgumentCount as Integer

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The number of arguments expected for this function.
Notes: Pass 0 to 127 for number of arguments or -1 for variable number of arguments.
(Read and Write property)

5.26.17 CallCounter as Integer

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The number of times the function was called.
Notes: (Read only property)

5.26.18 DatabaseCount as Integer

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The number of times the function was registered to a database connection.
Notes: (Read only property)

5.26.19 Enabled as Boolean

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: Whether this function is enabled.
Notes: The value is checked when connecting to a new SQLite database.
Defaults to true.
(Read and Write property)

5.26.20 Flags as Integer

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The flags for the function.
Example:

```
Var f As New SQLiteFunctionMBS

f.Flags = f.kFlagDeterministic OR f.kFlagUTF8 OR f.kFlagInnocuous
// Text encoding UTF8, function is deterministic and can be cached, function is innocuous as it depends only on parameters
```

Notes: See kFlag* constants.
(Read and Write property)

5.26.21 Name as String

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The name of the function.
Notes: (Read and Write property)

5.26.22 Events

5.26.23 Perform(ArgumentCount as Integer, Arguments() as Variant)

Plugin Version: 24.1, Platforms: macOS, Linux, Windows, iOS, Targets: All.

Function: The event called to perform the function.
Example:

```
Sub Perform(ArgumentCount as Integer, Arguments() as Variant) Handles Perform
#Pragma BackgroundTasks False
```

```
#Pragma BoundsChecking False
#Pragma StackOverflowChecking False

ResultText "Hello World"

End Sub
```

Notes: ArgumentCount is the number of arguments passed in the array.
Arguments contains values, which may be nil, double, integer, string, Memoryblock for BLOB.

Please use #pragmas like in the sample project to disable stack checking and background tasks for best performance. Also catch all exceptions that may happen to prevent trouble.

Call one of the result functions to return the result.

5.26.24 Constants

Flags

Constant	Value	Description
kFlagDeterministic	&h800	The function always gives the same output when the input parameters are the same.
kFlagDirectOnly	&h80000	The function may only be invoked from top-level SQL, and cannot be used in VIEWs or TRIGGERs nor in schema structures.
kFlagInnocuous	&h200000	The function is unlikely to cause problems even if misused. An innocuous function should have no side effects and should not depend on any values other than its input parameters.
kFlagUTF8	1	Specifies text encoding to be UTF-8.

5.27 class SQLLongBinaryMBS

5.27.1 class SQLLongBinaryMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: A class for a long binary object.
Notes: Basicly this is a SQLStringMBS which is always marked to contain binary data. You only need this class to use the constructor with dataprovider to stream data to the database.
Subclass of the SQLLongOrLobMBS class.
Blog Entries

- MBS Real Studio Plugins, version 12.4pr1

5.27.2 Methods

5.27.3 Constructor

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.
See also:

5.27.4 Constructor(Data as MemoryBlock)

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new string object with data, e.g. for blob.
See also:

5.27.5 Constructor(data as SQLStringMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new long binary object from a string object.
See also:

- 5.27.3 Constructor 210
- 5.27.4 Constructor(Data as MemoryBlock) 210
- 5.27.6 Constructor(Data as string, isText as Boolean = True) 211
- 5.27.7 Constructor(dataProvider as SQLDataProviderMBS, BlockSize as UInt32) 211

5.27.6 Constructor(Data as string, isText as Boolean = True)

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new string object with data or text copied from the data string.
Notes: If isText is true, the data is interpreted as text and string encoding conversion may modify it. If isText is false the bytes are copied raw.
See also:

- 5.27.3 Constructor 210
- 5.27.4 Constructor(Data as MemoryBlock) 210
- 5.27.5 Constructor(data as SQLStringMBS) 211
- 5.27.7 Constructor(dataProvider as SQLDataProviderMBS, BlockSize as UInt32) 211

5.27.7 Constructor(dataProvider as SQLDataProviderMBS, BlockSize as UInt32)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new long binary object from a data provider.
Notes: The blocksize specifies in which sizes data is requested from the provider.
You must make sure that the data provider and this new blob object life long enough. Because the actual data is requested later when you do the update on the database.

If BlockSize is 0, the default block size is used.
See also:

- 5.27.3 Constructor 210
- 5.27.4 Constructor(Data as MemoryBlock) 210
- 5.27.5 Constructor(data as SQLStringMBS) 211
- 5.27.6 Constructor(Data as string, isText as Boolean = True) 211

5.28 class SQLLongCharMBS

5.28.1 class SQLLongCharMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: A class for the long character data type.
Notes: Basicly this is a SQLStringMBS which is always marked to contain text. You only need this class to use the constructor with dataprovider to stream data to the database.
Subclass of the SQLLongOrLobMBS class.
Blog Entries

- MBS Real Studio Plugins, version 12.4pr1

5.28.2 Methods

5.28.3 Constructor

Plugin Version: 12.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.
See also:

- 5.28.4 Constructor(data as SQLStringMBS) 212
- 5.28.5 Constructor(Data as string, isText as boolean=true) 212
- 5.28.6 Constructor(dataProvider as SQLDataProviderMBS, BlockSize as UInt32) 213

5.28.4 Constructor(data as SQLStringMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new long character object from a string object.
See also:

- 5.28.3 Constructor 212
- 5.28.5 Constructor(Data as string, isText as boolean=true) 212
- 5.28.6 Constructor(dataProvider as SQLDataProviderMBS, BlockSize as UInt32) 213

5.28.5 Constructor(Data as string, isText as boolean=true)

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new string object with data or text copied from the data string.
Notes: If isText is true, the data is interpreted as text and string encoding conversion may modify it. If isText is false the bytes are copied raw.
See also:

- 5.28.3 Constructor 212
- 5.28.4 Constructor(data as SQLStringMBS) 212
- 5.28.6 Constructor(dataProvider as SQLDataProviderMBS, BlockSize as UInt32) 213

5.28.6 Constructor(dataProvider as SQLDataProviderMBS, BlockSize as UInt32)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new long character object from a data provider.
Notes: The blocksize specifies in which sizes data is requested from the provider.
You must make sure that the data provider and this new long character object life long enough. Because the actual data is requested later when you do the update on the database.

If BlockSize is 0, the default block size is used.
See also:

- 5.28.3 Constructor 212
- 5.28.4 Constructor(data as SQLStringMBS) 212
- 5.28.5 Constructor(Data as string, isText as boolean=true) 212

5.29 class SQLLongOrLobMBS

5.29.1 class SQLLongOrLobMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The super class for Long Binary/Text and BLOB/CLOB classes.
Notes: Subclass of the SQLStringMBS class.

5.30 class SQLNotInitializedExceptionMBS

5.30.1 class SQLNotInitializedExceptionMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The exception raised if you call a method on an object which was not properly initialized.
Notes: Subclass of the RuntimeException class.
Blog Entries

- MBS Xojo Plugins, version 19.2pr1

5.31 class SQLNullMBS

5.31.1 class SQLNullMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class used internally for null values.

5.32 class SQLNumericMBS

5.32.1 class SQLNumericMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for numeric values.
Notes: see also
https://www.sqlapi.com/ApiDoc/class_s_a_numeric.html
Blog Entries

- MBS Xojo Plugins, version 23.4pr6
- MBS Xojo Plugins, version 23.3pr7
- MBS Xojo Plugins, version 17.2pr4
- MBS Xojo / Real Studio Plugins, version 16.5pr9
- MBS Xojo / Real Studio plug-ins in version 14.2
- MBS Xojo / Real Studio Plugins, version 14.2pr1

Xojo Developer Magazine

- 12.4, page 9: News

5.32.2 Methods

5.32.3 Constructor

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates an empty numeric object.
See also:

- 5.32.4 Constructor(value as Double) 217
- 5.32.5 Constructor(value as string) 218

5.32.4 Constructor(value as Double)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new numeric object based on the given double value.
See also:

- 5.32.3 Constructor 217
- 5.32.5 Constructor(value as string) 218

5.32.5 Constructor(value as string)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new numeric object based on the given string.
See also:

- 5.32.3 Constructor 217
- 5.32.4 Constructor(value as Double) 217

5.32.6 NumericWithCurrency(value as Currency) as SQLNumericMBS

Plugin Version: 14.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new numeric value object with given currency value.
Example:

```
// test code for currency
Var c1 as Currency = 12345678.
Var c2 as Currency = 1234567.8
Var c3 as Currency = 123456.78
Var c4 as Currency = 12345.678
Var c5 as Currency = 1234.5678
Var c6 as Currency = 123.45678

Var n1 as SQLNumericMBS = SQLNumericMBS.NumericWithCurrency(c1)
Var n2 as SQLNumericMBS = SQLNumericMBS.NumericWithCurrency(c2)
Var n3 as SQLNumericMBS = SQLNumericMBS.NumericWithCurrency(c3)
Var n4 as SQLNumericMBS = SQLNumericMBS.NumericWithCurrency(c4)
Var n5 as SQLNumericMBS = SQLNumericMBS.NumericWithCurrency(c5)
Var n6 as SQLNumericMBS = SQLNumericMBS.NumericWithCurrency(c6)

Var s1 as string = n1.StringValue
Var s2 as string = n2.StringValue
Var s3 as string = n3.StringValue
Var s4 as string = n4.StringValue
Var s5 as string = n5.StringValue
Var s6 as string = n6.StringValue

Var d1 as Double = n1.DoubleValue
Var d2 as Double = n2.DoubleValue
Var d3 as Double = n3.DoubleValue
Var d4 as Double = n4.DoubleValue
Var d5 as Double = n5.DoubleValue
Var d6 as Double = n6.DoubleValue

Var x1 as Currency = n1.CurrencyValue
```

```
Var x2 as Currency = n2.CurrencyValue
Var x3 as Currency = n3.CurrencyValue
Var x4 as Currency = n4.CurrencyValue
Var x5 as Currency = n5.CurrencyValue
Var x6 as Currency = n6.CurrencyValue

// check for errors
if x1<>c1 then break
if x2<>c2 then break
if x3<>c3 then break
if x4<>c4 then break
if x5<>c5 then break
if x6<>c6 then break

if x1*10000 <>round(d1 * 10000) then Break
if x2*10000 <>round(d2 * 10000) then Break
if x3*10000 <>round(d3 * 10000) then Break
if x4*10000 <>round(d4 * 10000) then Break
if x5*10000 <>round(d5 * 10000) then Break
if x6*10000 <>round(d6 * 10000) then Break

Break // if no break before, it's okay.
```

5.32.7 NumericWithDouble(value as Double) as SQLNumericMBS

Plugin Version: 14.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new number with the given double value.

5.32.8 NumericWithInt64(value as Int64) as SQLNumericMBS

Plugin Version: 14.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new number with the given Int64 value.

5.32.9 NumericWithString(value as string) as SQLNumericMBS

Plugin Version: 14.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new number with the given string value.
Notes: If string is empty, we return a number with zero as value.

5.32.10 NumericWithUInt64(value as UInt64) as SQLNumericMBS

Plugin Version: 14.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new number with the given unsigned integer value.

5.32.11 Properties

5.32.12 CurrencyValue as Currency

Plugin Version: 14.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The currency value.
Example:

```
// test code for currency
Var c1 as Currency = 12345678.
Var c2 as Currency = 1234567.8
Var c3 as Currency = 123456.78
Var c4 as Currency = 12345.678
Var c5 as Currency = 1234.5678
Var c6 as Currency = 123.45678

Var n1 as SQLNumericMBS = SQLNumericMBS.NumericWithCurrency(c1)
Var n2 as SQLNumericMBS = SQLNumericMBS.NumericWithCurrency(c2)
Var n3 as SQLNumericMBS = SQLNumericMBS.NumericWithCurrency(c3)
Var n4 as SQLNumericMBS = SQLNumericMBS.NumericWithCurrency(c4)
Var n5 as SQLNumericMBS = SQLNumericMBS.NumericWithCurrency(c5)
Var n6 as SQLNumericMBS = SQLNumericMBS.NumericWithCurrency(c6)

Var s1 as string = n1.StringValue
Var s2 as string = n2.StringValue
Var s3 as string = n3.StringValue
Var s4 as string = n4.StringValue
Var s5 as string = n5.StringValue
Var s6 as string = n6.StringValue

Var d1 as Double = n1.DoubleValue
Var d2 as Double = n2.DoubleValue
Var d3 as Double = n3.DoubleValue
Var d4 as Double = n4.DoubleValue
Var d5 as Double = n5.DoubleValue
Var d6 as Double = n6.DoubleValue

Var x1 as Currency = n1.CurrencyValue
Var x2 as Currency = n2.CurrencyValue
Var x3 as Currency = n3.CurrencyValue
```

```
Var x4 as Currency = n4.CurrencyValue
Var x5 as Currency = n5.CurrencyValue
Var x6 as Currency = n6.CurrencyValue

// check for errors
if x1<>c1 then break
if x2<>c2 then break
if x3<>c3 then break
if x4<>c4 then break
if x5<>c5 then break
if x6<>c6 then break

if x1*10000 <>round(d1 * 10000) then Break
if x2*10000 <>round(d2 * 10000) then Break
if x3*10000 <>round(d3 * 10000) then Break
if x4*10000 <>round(d4 * 10000) then Break
if x5*10000 <>round(d5 * 10000) then Break
if x6*10000 <>round(d6 * 10000) then Break

Break // if no break before, it's okay.
```

Notes: (Read and Write property)

5.32.13 DoubleValue as Double

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The double value for this number.
Notes: (Read and Write property)

5.32.14 Int64Value as Int64

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number value as an int64.
Notes: (Read and Write property)

5.32.15 precision as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The maximum number of digits in base 10.

Notes: (Read only property)

5.32.16 scale as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number of digits to the right of the decimal point.
Notes: (Read only property)

5.32.17 sign as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The sign: 1 for positive numbers, 0 for negative numbers.
Notes: (Read only property)

5.32.18 StringValue as string

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The string value of this number.
Notes: (Read and Write property)

5.32.19 UInt64Value as UInt64

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The number value as an uint64.
Notes: (Read and Write property)

5.33 class SQLParamMBS

5.33.1 class SQLParamMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The SQL class for parameters.
Notes: see also
https://www.sqlapi.com/ApiDoc/class_s_a_param.html
Subclass of the SQLValueMBS class.
Blog Entries

- MBS SQL Plugin Tips and Tricks
- MBS Xojo Plugins, version 20.3pr3
- MBS Xojo Plugins, version 19.5pr4
- MBS Xojo Plugins, version 19.2pr1
- MBS Xojo Plugins, version 18.4pr10
- MBS Xojo / Real Studio Plugins, version 16.4pr5
- Upcoming Changes for our SQL Plugin

5.33.2 Methods

5.33.3 ReadLongOrLob(toConsumer as SQLDataConsumerMBS, BlockSize as Integer)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The Long or Lob data reading mode.
Notes: SQLAPI++ Library provides two ways to read Long or BLob(CLob) object's value (usually SQLField or SQLParam objects):

1. reading of Long or Lob data at once into an internal buffer (like ordinary string or binary values);
2. piecewise reading of Long or Lob data using user defined callback.
kLongOrLobReaderDefault reading mode used by default.

If you want to control piecewise reading of Long or BLob(CLob) data you should set LongOrLobReaderMode and use kLongOrLobReaderManual reading mode for Long or BLob(CLob) parameters or fields you want to process with your data consumer. After that each fetch will skip reading Long and BLob(CLob) parameters that you set to be read manually. To read field or parameter defined to be read manually you should call ReadLongOrLob method for each of them after the fetch. ReadLongOrLob method will repeatedly call the data consumer Write event.
See also:

5.33.4 ReadLongOrLob(toFile as FolderItem)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Starts reading of Long or BLob(CLob) value to the given file.
Example:

```
Var cmd as SQLCommandMBS // your command
Var f as FolderItem = SpecialFolder.Desktop.Child("test.jpg")
Var Param as SQLParamMBS = cmd.Param("image")
// read blob content to binarystream
Param.ReadLongOrLob(f)
```

Notes: May raise IOExceptions if things go wrong.
See also:

5.33.5 ReadLongOrLob(toStream as Writeable)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Starts reading of Long or BLob(CLob) value to the given writeable stream.
Example:

```
Var cmd as SQLCommandMBS // your command
Var f as FolderItem = SpecialFolder.Desktop.Child("test.jpg")
Var b as BinaryStream = BinaryStream.Create(f, true)
Var param as SQLParamMBS = cmd.param("image")
// read blob content to binarystream
param.ReadLongOrLob(b)
```

Notes: This allows you to read in chunks the data to a stream, e.g. binarystream, textoutputstream or socket.
See also:

5.33.6 Properties

5.33.7 DirType as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The direction type of parameter (input, output, etc.).
Notes: Use the kParamDirType* constants.

Usually the Library automatically detects parameter's direction type and implicitly creates an appropriate SAParam object. But not all of DBMS clients/servers provide complete parameters information. In that situation programmer need to describe parameter's direction type explicitly. See Server specific notes for details.

https://www.sqlapi.com/ApiDoc/class_s_a_param.html
(Read and Write property)

5.33.8 IsInput as Boolean

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this is an input paramater.
Notes: Checks ParamType property internally.
(Read only property)

5.33.9 IsOutput as Boolean

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether this is an output paramater.
Notes: Checks ParamType property internally.
(Read only property)

5.33.10 Name as string

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The name of the parameter.
Notes: (Read only property)

5.33.11 NativeType as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The native type code of the parameter.
Notes: (Read and Write property)

5.33.12 Options as Dictionary

Plugin Version: 18.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a dictionary with all options.
Notes: For debugging, it may be useful to inspect options in debugger.
(Read only property)

5.33.13 Precision as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The precision of the parameter value (the total number of allowable digits).
Notes: (Read and Write property)

5.33.14 Scale as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The scale of the parameter value (the number of digits to the right of the decimal point).
Notes: (Read and Write property)

5.33.15 Size as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The parameter's data size.
Notes: (Read and Write property)

5.33.16 Type as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The parameter's data type.
Notes: See the kDataType constants.
(Read and Write property)

5.33.17 Option(name as string) as string

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The string value of a specific parameter option.
Notes: see also:
https://www.sqlapi.com/ApiDoc/class_s_a_param.html
(Read and Write computed property)

5.33.18 Constants

Constants

Constant	Value	Description
kParamDirTypeInput	0	One of the parameter direction type constants. Input parameter.
kParamDirTypeInputOutput	1	One of the parameter direction type constants. Input/output parameter.
kParamDirTypeOutput	2	One of the parameter direction type constants. Output parameter.
kParamDirTypeReturn	3	One of the parameter direction type constants. Returning parameter.

5.34 class SQLPositionMBS

5.34.1 class SQLPositionMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for a position value.

5.34.2 Methods

5.34.3 Constructor(withID as Integer)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new Position value with an ID.
See also:

- 5.34.4 Constructor(withName as string) 228

5.34.4 Constructor(withName as string)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new Position value with a name.
See also:

- 5.34.3 Constructor(withID as Integer) 228

5.35 class SQLPreparedStatementMBS

5.35.1 class SQLPreparedStatementMBS

Plugin Version: 11.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for prepared statements if you work with SQLDatabaseMBS class.
Notes: If you work with SQLCommandMBS class, you can set parameters there directly.

For the SQL string you number parameters with colon and number. Like this: :1, :2, :3.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
Blog Entries

- SQLDatabase sample
- MBS SQL Plugin Tips and Tricks
- Using MBS SQL Plugin with PostgreSQL
- Multithreaded plugin functions can increase speed of Xojo application
- MBS Plugins updated for Xojo 2019r2.1
- MBS Releases the MBS Xojo / Real Studio plug-ins in version 16.4
- Upcoming Changes for our SQL Plugin
- MBS Xojo / Real Studio plug-ins in version 16.3
- MonkeyBread Software Releases the MBS Xojo / Real Studio plug-ins in version 14.1
- SQLPreparedStatementMBS improvements

Videos

- MBS SQL Plugin Presentation

Xojo Developer Magazine

- 14.1, page 30: The MBS SQL Plugin, An alternative way to connect to databases by Christian Schmitz
- 12.3, page 10: News

Interfaces: PreparedSQLStatement

5.35.2 Methods

5.35.3 Bind(name As String, value as Variant)

Plugin Version: 14.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Defines the value for one parameter.
Example:

```
Var db as SQLDatabaseMBS // your db connection
Var sql as string = "Insert into test_tbl(fid, fvarchar20) values(:fid, :fvarchar20)"
Var v as Variant = db.Prepare(sql)
Var p as SQLPreparedStatementMBS = v

p.BindType("fid", SQLPreparedStatementMBS.kTypeLong)
p.BindType("fvarchar20", SQLPreparedStatementMBS.kTypeString)
p.Bind("fid", 2345)
p.Bind("fvarchar20", "Hello World by name")

p.SQLExecute
```

Notes: Version 16.4 and newer allow you to bind BLOB fields using a Memoryblock or a String value. Older versions only accepted string.

When passing variant for value, MemoryBlock and Strings without text encoding are converted to byte values (BLOB). Texts and Strings with encoding are converted to text values. Other types are translated as good as possible. Raises exceptions if you pass anything which is not recognized.

With version 19.0 or later, you can pass folderitem to stream blob from file. This may raise exception if file can't be opened.
See also:

5.35.4 Bind(name As String, value as Variant, type as Integer)

Plugin Version: 14.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Defines one paramater with value and type.
Example:

```
Var db as SQLDatabaseMBS // your db connection
Var sql as string = "Insert into test_tbl(fid, fvarchar20) values(:fid, :fvarchar20)"
Var v as Variant = db.Prepare(sql)
Var p as SQLPreparedStatementMBS = v

p.Bind("fid", 2345, SQLPreparedStatementMBS.kTypeLong)
p.Bind("fvarchar20", "Hello World by name", SQLPreparedStatementMBS.kTypeString)

p.SQLExecute
```

Notes: Version 16.4 and newer allow you to bind BLOB fields using a Memoryblock or a String value. Older versions only accepted string.

When passing variant for value, MemoryBlock and Strings without text encoding are converted to byte values (BLOB). Texts and Strings with encoding are converted to text values. Other types are translated as good as possible. Raises exceptions if you pass anything which is not recognized.

With version 19.0 or later, you can pass folderitem to stream blob from file. This may raise exception if file can't be opened.

With plugin version 16.4 and newer binding type is optional. In that case the type is determinated from the value type.
See also:

5.35.5 Bind(Values as Dictionary)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the parameters based on the keys and values in the dictionary.
Example:

```
Var db as SQLDatabaseMBS // your database
Var pic as Picture // some picture
Var jpegData as MemoryBlock = pic.GetData(Picture.FormatJPEG, 80)

// get an insert command
Var sql as string = "Insert into BlobTest(name, image) values (:name, :image)"
```

```
Var p as SQLPreparedStatementMBS = db.Prepare(sql)

// put parameter values in a dictionary
Var d as new Dictionary

// by param index
d.Value(0) = "logo.jpg"
// by param name
d.Value("image") = jpegData

// bind values and run it
p.Bind(d)
p.SQLExecute
```

Notes: The dictionary is saved to fill parameters later.
Keys can be String, Text or numeric types. Text and String are used to pick parameters by name. Numeric values are used to pick parameter by index (zero based).
MemoryBlock and Strings without text encoding are converted to byte values (BLOB).
Texts and Strings with encoding are converted to text values.
Raises exceptions if you pass anything which is not recognized.
Other types are translated as good as possible.
See also:

- 5.35.3 Bind(name As String, value as Variant) 230
- 5.35.4 Bind(name As String, value as Variant, type as Integer) 230
- 5.35.6 Bind(values() as Variant) 232
- 5.35.7 Bind(zeroBasedIndex as Integer, value as Variant) 233
- 5.35.8 Bind(zeroBasedIndex as Integer, value as Variant, type as Integer) 234

5.35.6 Bind(values() as Variant)

Plugin Version: 11.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the value list with the given values.
Notes: You can either pass values to the SQLExecute/SQLSelect method or call Bind methods to set values. You have to define for each parameter both the type and the value.

Version 16.4 and newer allow you to bind BLOB fields using a Memoryblock or a String value.
Older versions only accepted string.

When passing variant for value, MemoryBlock and Strings without text encoding are converted to byte values (BLOB). Texts and Strings with encoding are converted to text values. Other types are translated as

good as possible. Raises exceptions if you pass anything which is not recognized.

With version 19.0 or later, you can pass folderitem to stream blob from file. This may raise exception if file can't be opened.
See also:

- 5.35.3 Bind(name As String, value as Variant) 230
- 5.35.4 Bind(name As String, value as Variant, type as Integer) 230
- 5.35.5 Bind(Values as Dictionary) 231
- 5.35.7 Bind(zeroBasedIndex as Integer, value as Variant) 233
- 5.35.8 Bind(zeroBasedIndex as Integer, value as Variant, type as Integer) 234

5.35.7 Bind(zeroBasedIndex as Integer, value as Variant)

Plugin Version: 11.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Defines the value for one parameter.
Notes: You can either pass values to the SQLExecute/SQLSelect method or call Bind methods to set values. You have to define for each parameter both the type and the value.

Version 16.4 and newer allow you to bind BLOB fields using a Memoryblock or a String value.
Older versions only accepted string.

When passing variant for value, MemoryBlock and Strings without text encoding are converted to byte values (BLOB). Texts and Strings with encoding are converted to text values. Other types are translated as good as possible. Raises exceptions if you pass anything which is not recognized.

With version 19.0 or later, you can pass folderitem to stream blob from file. This may raise exception if file can't be opened.

Raises OutOfBoundsException exception if index parameter is out of range.
See also:

- 5.35.3 Bind(name As String, value as Variant) 230
- 5.35.4 Bind(name As String, value as Variant, type as Integer) 230
- 5.35.5 Bind(Values as Dictionary) 231
- 5.35.6 Bind(values() as Variant) 232
- 5.35.8 Bind(zeroBasedIndex as Integer, value as Variant, type as Integer) 234

5.35.8 Bind(zeroBasedIndex as Integer, value as Variant, type as Integer)

Plugin Version: 11.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Defines one paramater with value and type.
Notes: You can either pass values to the SQLExecute/SQLSelect method or call Bind methods to set values. You have to define for each parameter both the type and the value.

Version 16.4 and newer allow you to bind BLOB fields using a Memoryblock or a String value.
Older versions only accepted string.

When passing variant for value, MemoryBlock and Strings without text encoding are converted to byte values (BLOB). Texts and Strings with encoding are converted to text values. Other types are translated as good as possible. Raises exceptions if you pass anything which is not recognized.

With plugin version 16.4 and newer binding type is optional. In that case the type is determinated from the value type.

With version 19.0 or later, you can pass folderitem to stream blob from file. This may raise exception if file can't be opened.

Raises OutOfBoundsException exception if index parameter is out of range.
See also:

- 5.35.3 Bind(name As String, value as Variant) 230
- 5.35.4 Bind(name As String, value as Variant, type as Integer) 230
- 5.35.5 Bind(Values as Dictionary) 231
- 5.35.6 Bind(values() as Variant) 232
- 5.35.7 Bind(zeroBasedIndex as Integer, value as Variant) 233

5.35.9 BindType(name As String, type as Integer)

Plugin Version: 14.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Defines the type of one value.
Example:

```
Var db as SQLDatabaseMBS // your db connection
Var sql as string = "Insert into test_tbl(fid, fvarchar20) values(:fid, :fvarchar20)"
Var v as Variant = db.Prepare(sql)
Var p as SQLPreparedStatementMBS = v
```

```
p.BindType("fid", SQLPreparedStatementMBS.kTypeLong)
p.BindType("fvarchar20", SQLPreparedStatementMBS.kTypeString)
p.Bind("fid", 2345)
p.Bind("fvarchar20", "Hello World by name")

p.SQLExecute
```

Notes: With plugin version 16.4 and newer binding type is optional. In that case the type is determinated from the value type.
See also:

- 5.35.10 BindType(types() as Integer) 235
- 5.35.11 BindType(zeroBasedIndex as Integer, type as Integer) 235

5.35.10 BindType(types() as Integer)

Plugin Version: 11.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Defines the types for all values.
Notes: You can either pass values to the SQLExecute/SQLSelect method or call Bind methods to set values. You have to define for each parameter both the type and the value.

With plugin version 16.4 and newer binding type is optional. In that case the type is determinated from the value type.
See also:

- 5.35.9 BindType(name As String, type as Integer) 234
- 5.35.11 BindType(zeroBasedIndex as Integer, type as Integer) 235

5.35.11 BindType(zeroBasedIndex as Integer, type as Integer)

Plugin Version: 11.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Defines the type of one value.
Notes: You can either pass values to the SQLExecute/SQLSelect method or call Bind methods to set values. You have to define for each parameter both the type and the value.

With plugin version 16.4 and newer binding type is optional. In that case the type is determinated from the value type.
See also:

- 5.35.9 BindType(name As String, type as Integer) 234
- 5.35.10 BindType(types() as Integer) 235

5.35.12 Clear

Plugin Version: 14.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Clears all parameters for reusing the SQL Prepared statement.

5.35.13 Constructor

Plugin Version: 11.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: The private constructor.
Notes: This constructor makes sure you don't create useless SQLPreparedStatementMBS objects by error. The only way to create an object is to use the prepare method in the database class.
This constructor is private to make sure you don't create an object from this class by error. Please use designated functions to create objects.

5.35.14 ExecuteSQL(ParamArray bindItems As Variant)

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Runs the SQL command with the given parameters.
Notes: You can decide whether you pass values here or call Bind methods.

When passing variant for value, MemoryBlock and Strings without text encoding are converted to byte values (BLOB). Texts and Strings with encoding are converted to text values. Other types are translated as good as possible. Raises exceptions if you pass anything which is not recognized.

With version 19.0 or later, you can pass folderitem to stream blob from file. This may raise exception if file can't be opened.

If bindItems is not nil, we bind parameters using it.
If you like to pass an array of variant for the values, please pass this to Bind() and no parameters here. Otherwise passing an array here would create a variant array with your array as content.

This version of the function raises always exceptions, while SQLExecute only if you set RaiseException property to true.

5.35.15 ExecuteSQLMT(ParamArray bindItems As Variant)

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Runs the SQL command with the given parameters.
Notes: You can decide whether you pass values here or call Bind methods.

When passing variant for value, MemoryBlock and Strings without text encoding are converted to byte values (BLOB). Texts and Strings with encoding are converted to text values. Other types are translated as good as possible. Raises exceptions if you pass anything which is not recognized.

If bindItems is not nil, we bind parameters using it.

With version 19.0 or later, you can pass folderitem to stream blob from file. This may raise exception if file can't be opened.

This version of the function raises always exceptions, while SQLExecuteMT only if you set RaiseException property to true.

If you like to pass an array of variant for the values, please pass this to Bind() and no parameters here. Otherwise passing an array here would create a variant array with your array as content.

The work is performed on a preemptive thread, so this function does not block the application and can yield time to other Xojo threads. Must be called in a Xojo thread to enjoy benefits. If called in main thread will block, but keep other background threads running.

5.35.16 SelectSQL(ParamArray bindItems As Variant) As RowSet

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Runs the query with the given parameters.
Notes: Returns the RowSet object or nil on error.
You can decide whether you pass values here or call Bind methods.

For this method to work, you need to have somewhere a property with SQLDatabaseMBS so Xojo includes our SQLDatabase plugin which provides the RecordSet functionality.

The record set may not have a valid RecordCount or have working movefirst/movelast/moveprev methods unless the underlaying database supports those and Scrollable result sets is enabled/supported.

If bindItems is not nil, we bind parameters using it.
If you like to pass an array of variant for the values, please pass this to Bind() and no parameters here. Otherwise passing an array here would create a variant array with your array as content.

When passing variant for value, MemoryBlock and Strings without text encoding are converted to byte values (BLOB). Texts and Strings with encoding are converted to text values. Other types are translated as good as possible. Raises exceptions if you pass anything which is not recognized.

This version of the function raises always exceptions, while SQLSelect only if you set RaiseException property to true.

5.35.17 SelectSQLMT(ParamArray bindItems As Variant) As Rowset

Plugin Version: 19.5, Platforms: macOS, Linux, Windows, Targets: All.

Function:

Returns the RowSet object or nil on error.
You can decide whether you pass values here or call Bind methods.

For this method to work, you need to have somewhere a property with SQLDatabaseMBS so Xojo includes our SQLDatabase plugin which provides the RecordSet functionality.

The record set may not have a valid RecordCount or have working movefirst/movelast/moveprev methods unless the underlaying database supports those and Scrollable result sets is enabled/supported.

If bindItems is not nil, we bind parameters using it.
If you like to pass an array of variant for the values, please pass this to Bind() and no parameters here. Otherwise passing an array here would create a variant array with your array as content.

When passing variant for value, MemoryBlock and Strings without text encoding are converted to byte values (BLOB). Texts and Strings with encoding are converted to text values. Other types are translated as good as possible. Raises exceptions if you pass anything which is not recognized.

This version of the function raises always exceptions, while SQLSelectMT only if you set RaiseException property to true.

The work is performed on a preemptive thread, so this function does not block the application and can yield time to other Xojo threads. Must be called in a Xojo thread to enjoy benefits. If called in main thread will block, but keep other background threads running.

5.35.18 SQLExecute(ParamArray bindItems as Variant)

Plugin Version: 11.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Runs the SQL command with the given parameters.
Notes: You can decide whether you pass values here or call Bind methods.

When passing variant for value, MemoryBlock and Strings without text encoding are converted to byte values (BLOB). Texts and Strings with encoding are converted to text values. Other types are translated as good as possible. Raises exceptions if you pass anything which is not recognized.

If you like to pass an array of variant for the values, please pass this to Bind() and no parameters here. Otherwise passing an array here would create a variant array with your array as content.

With version 19.0 or later, you can pass folderitem to stream blob from file. This may raise exception if file can't be opened.

5.35.19 SQLExecuteMT(ParamArray bindItems as Variant)

Plugin Version: 16.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Runs the SQL command with the given parameters.
Notes: You can decide whether you pass values here or call Bind methods.

The work is performed on a preemptive thread, so this function does not block the application and can yield time to other Xojo threads. Must be called in a Xojo thread to enjoy benefits. If called in main thread will block, but keep other background threads running.

When passing variant for value, MemoryBlock and Strings without text encoding are converted to byte values (BLOB). Texts and Strings with encoding are converted to text values. Other types are translated as good as possible. Raises exceptions if you pass anything which is not recognized.

If you like to pass an array of variant for the values, please pass this to Bind() and no parameters here. Otherwise passing an array here would create a variant array with your array as content.

With version 19.0 or later, you can pass folderitem to stream blob from file. This may raise exception if file can't be opened.

5.35.20 SQLSelect(ParamArray bindItems as Variant) As RecordSet

Plugin Version: 11.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Runs the query with the given parameters.
Notes: Returns the recordset object or nil on error.

You can decide whether you pass values here or call Bind methods.

For this method to work, you need to have somewhere a property with SQLDatabaseMBS so Xojo includes our SQLDatabase plugin which provides the RecordSet functionality.

The record set may not have a valid RecordCount or have working movefirst/movelast/moveprev methods unless the underlaying database supports those and Scrollable result sets is enabled/supported.

If you like to pass an array of variant for the values, please pass this to Bind() and no parameters here. Otherwise passing an array here would create a variant array with your array as content.

When passing variant for value, MemoryBlock and Strings without text encoding are converted to byte values (BLOB). Texts and Strings with encoding are converted to text values. Other types are translated as good as possible. Raises exceptions if you pass anything which is not recognized.

With version 19.0 or later, you can pass folderitem to stream blob from file. This may raise exception if file can't be opened.

5.35.21 SQLSelectMT(ParamArray bindItems as Variant) As RecordSet

Plugin Version: 16.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Runs the query with the given parameters.
Notes: Returns the recordset object or nil on error.
You can decide whether you pass values here or call Bind methods.

For this method to work, you need to have somewhere a property with SQLDatabaseMBS so Xojo includes our SQLDatabase plugin which provides the RecordSet functionality.

The record set may not have a valid RecordCount or have working movefirst/movelast/moveprev methods unless the underlaying database supports those and Scrollable result sets is enabled/supported.

If you like to pass an array of variant for the values, please pass this to Bind() and no parameters here. Otherwise passing an array here would create a variant array with your array as content.

When passing variant for value, MemoryBlock and Strings without text encoding are converted to byte values (BLOB). Texts and Strings with encoding are converted to text values. Other types are translated as good as possible. Raises exceptions if you pass anything which is not recognized.

With version 19.0 or later, you can pass folderitem to stream blob from file. This may raise exception if file

can't be opened.

The work is performed on a preemptive thread, so this function does not block the application and can yield time to other Xojo threads. Must be called in a Xojo thread to enjoy benefits. If called in main thread will block, but keep other background threads running.

5.35.22 Properties

5.35.23 BoundTypes as Dictionary

Plugin Version: 19.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The binded types to inspect in debugger.
Notes: When queried, the plugin creates a copy of the dictionary for you to inspect in debugger. Changes to the dictionary do not bind types.

Renamed to BoundTypes in version 19.4. Was BindedTypes, but that was not proper english.
(Read only property)

5.35.24 BoundValues as Dictionary

Plugin Version: 19.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The binded values to inspect in debugger.
Notes: When queried, the plugin creates a copy of the dictionary for you to inspect in debugger. Changes to the dictionary do not bind values.

Renamed to BoundValues in version 19.4. Was BindedValues, but that was not proper english.
(Read only property)

5.35.25 Scrollable as Boolean

Plugin Version: 14.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Whether the plugin will ask for a scrollable recordset when doing SQLSelect.
Notes: Since plugin version 15.0, Scrollable is false by default.
(Read and Write property)

5.35.26 SQL as String

Plugin Version: 12.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The SQL command for this prepared statement.
Notes: (Read and Write property)

5.35.27 Constants

Data Type Constants

Constant	Value	Description
kTypeBlob	16	Binary large Object. Pass a string or memoryblock.
kTypeBool	1	Boolean
kTypeBytes	13	a binary string or MemoryBlock. (which is a string without text encoding) This is usually a varbinary field. kTypeLongBinary is used for bigger binary data and streamed. And kTypeBlob is used for huge streams of bytes and also transferred in chunks. In most data base systems the varchar field is stored within the record, while BLOB and CLOB are stored separately.
kTypeClob	17	Character Large Object.
kTypeDateTime	10	Date and/or Time.
kTypeDouble	8	double float value.
kTypeInt32	4	signed 32 bit integer
kTypeInt64	6	Signed 64-bit integer.
kTypeInterval	11	An interval. Please pass SQLIntervalMBS in the variant. If the variant contains anything else, the plugin will pass nil value. When passing variant for value, MemoryBlock and Strings without text encoding are converted to byte values (BLOB). Texts and Strings with encoding are converted to text values. Other types are translated as good as possible. Raises exceptions if you pass anything which is not recognized.
kTypeLong	4	signed 32 bit integer
kTypeLongBinary	14	Long binary.
kTypeLongChar	15	Long string.
kTypeNull	99	NULL value
kTypeNumeric	9	A number (Int64 or double). This can be used for Int64 or Double values. Depending of the type of number in the variant, the plugin will either make an Int64 or a double internally. When passing variant for value, MemoryBlock and Strings without text encoding are converted to byte values (BLOB). Texts and Strings with encoding are converted to text values. Other types are translated as good as possible. Raises exceptions if you pass anything which is not recognized.
kTypeShort	2	signed 16 bit integer
kTypeString	12	String. This is usually a varchar field. kTypeLongChar is used for bigger varchars and streamed. And kTypeClob is used for huge streams of characters (BLOB for text) and also transferred in chunks. In most data base systems the varchar field is stored within the record, while BLOB and CLOB are stored separately.
kTypeUInt32	5	unsigned 32 bit integer
kTypeUInt64	7	Unsigned 64-bit integer.
kTypeULong	5	unsigned 32 bit integer
kTypeUnknown	0	unknown type
kTypeUShort	3	unsigned 16 bit integer

5.36 class SQLStringMBS

5.36.1 class SQLStringMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for strings in this plugin.
Example:

```
Var s as new SQLStringMBS("Hello √§√√º")

MsgBox "Characters: "+str(s.GetLength)+" Bytes: "+str(s.GetBinaryLength)

Var a as string= s.CopyBinaryData
Var b as string= s.CopyText

MsgBox a // RB shows garbage as it tries to display bytes as UTF8 which does not work
MsgBox b // displays correct
```

Notes: A string can be text (with text encoding) or bytes (raw binary data).

see also
https://www.sqlapi.com/ApiDoc/class_s_a_string.html
Blog Entries

- MBS Xojo Plugins, version 24.0pr6
- MBS Real Studio Plugins, version 12.4pr9

5.36.2 Methods

5.36.3 Compare(text as SQLStringMBS) as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Compares this string object with another string.
Notes: Function performs a case-sensitive comparison of the strings, and is not affected by locale.

Returns zero if the strings are identical, <0 if this string object is less than text, or >0 if this string object is greater than text.
See also:

- 5.36.4 Compare(text as string) as Integer

5.36.4 Compare(text as string) as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Compares this string object with another string.
Notes: Function performs a case-sensitive comparison of the strings, and is not affected by locale.

Returns zero if the strings are identical, <0 if this string object is less than text, or >0 if this string object is greater than text.
See also:

- 5.36.3 Compare(text as SQLStringMBS) as Integer 244

5.36.5 CompareNoCase(text as SQLStringMBS) as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Compares this string object with another string.
Notes: Function performs a case-insensitive comparison of the strings, and is not affected by locale.

Returns zero if the strings are identical (ignoring case), <0 if this string object is less than text (ignoring case), or >0 if this string object is greater than text (ignoring case).
See also:

- 5.36.6 CompareNoCase(text as string) as Integer 245

5.36.6 CompareNoCase(text as string) as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Compares this string object with another string.
Notes: Function performs a case-insensitive comparison of the strings, and is not affected by locale.

Returns zero if the strings are identical (ignoring case), <0 if this string object is less than text (ignoring case), or >0 if this string object is greater than text (ignoring case).
See also:

- 5.36.5 CompareNoCase(text as SQLStringMBS) as Integer 245

5.36.7 Constructor

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new empty string object.
See also:

5.36.8 Constructor(Data as MemoryBlock)

Plugin Version: 15.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new string object with data, e.g. for blob.
See also:

5.36.9 Constructor(Data as string, isText as Boolean = True)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new string object with data or text copied from the data string.
Notes: If isText is true, the data is interpreted as text and string encoding conversion may modify it. If isText is false the bytes are copied raw.
See also:

5.36.10 Constructor(other as SQLStringMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new string object with data copied from the other string object.
See also:

5.36.11 CopyBinaryData as string

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies the bytes from the internal buffer ignoring any text encoding.

5.36.12 CopyMemoryBlock as MemoryBlock

Plugin Version: 24.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies the content of the string/BLOB as MemoryBlock.

5.36.13 CopyText as string

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Copies the characters of this string as text.
Notes: Text encoding conversion may happen.

5.36.14 Empty

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Forces a string to have 0 length.

5.36.15 GetBinaryLength as UInt32

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use BinaryLength instead.
Function: Returns a count of the bytes in the binary data buffer.
Notes: Deprecated. Please use BinaryLength property.

5.36.16 GetLength as UInt32

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Deprecated: This item is deprecated and should no longer be used. You can use Length instead. **Function:** Returns the number of characters in a SAString object.
Notes: Deprecated. Please use Length property.

For multibyte character sets, GetLength counts each 8-bit character; that is, a lead and trail byte in one multibyte character are counted as two bytes.

5.36.17 Left(count as Integer) as SQLStringMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Extracts the left part of a string.

5.36.18 MakeLower

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Changes all characters in the string to lower case.

5.36.19 MakeUpper

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Changes all characters in the string to upper case.

5.36.20 Mid(first as Integer) as SQLStringMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Extracts the middle part of a string.
Notes: first: The zero-based index of the first character in this string object that is to be included in the extracted substring.
See also:

- 5.36.21 Mid(first as Integer, Count as Integer) as SQLStringMBS 248

5.36.21 Mid(first as Integer, Count as Integer) as SQLStringMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Extracts the middle part of a string.
Notes: first: The zero-based index of the first character in this string object that is to be included in the extracted substring.
count: The number of characters to extract from this string object. If this parameter is not supplied, then

the remainder of the string is extracted.
See also:

- 5.36.20 Mid(first as Integer) as SQLStringMBS 248

5.36.22 Operator_Convert as string

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Convert operation for assignment to string.
See also:

- 5.36.23 Operator_Convert(text as string) 249

5.36.23 Operator_Convert(text as string)

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Convert operation for assigning text string.
See also:

- 5.36.22 Operator_Convert as string 249

5.36.24 Right(count as Integer) as SQLStringMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Extracts the right part of a string.

5.36.25 TrimLeft

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Trim leading whitespace characters from the string.

5.36.26 TrimRight

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Trim trailing whitespace characters from the string.

5.36.27 Properties

5.36.28 BinaryLength as UInt32

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns a count of the bytes in the binary data buffer.
Notes: (Read only property)

5.36.29 DebugText as String

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Text content for debugging.
Notes: We show up to 1000 characters here for debugger.
(Read only property)

5.36.30 IsEmpty as boolean

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Tests whether a String object contains no characters.
Notes: Returns true if length is zero.
(Read only property)

5.36.31 Length as UInt32

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the number of characters in a SAString object.
Notes: For multibyte character sets, GetLength counts each 8-bit character; that is, a lead and trail byte in one multibyte character are counted as two bytes.
(Read only property)

5.37 class SQLUnsupportedExceptionMBS

5.37.1 class SQLUnsupportedExceptionMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class for an exception to report that the function is not supported on this platform.
Notes: This one raises only if the plugin is compiled for Mac OS Classic.
Subclass of the RuntimeException class.

5.38 class SQLValueMBS

5.38.1 class SQLValueMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The SQL class for mutable values.
Notes: Subclass of the SQLValueReadMBS class.
Blog Entries

- MBS Xojo Plugins, version 20.5pr2
- MBS Xojo Plugins, version 19.2pr1
- MBS Releases the MBS Xojo / Real Studio plug-ins in version 16.4
- MBS Xojo / Real Studio Plugins, version 16.4pr5
- Upcoming Changes for our SQL Plugin
- Text data type
- MBS Xojo / Real Studio Plugins, version 14.0pr6
- Release notes for SQL or ChartDirector?
- MBS Real Studio Plugins, version 11.2pr3

Videos

- Presentation from London conference about MBS Plugins.

5.38.2 Methods

5.38.3 Constructor(DataType as Integer)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new empty value object with the given data type.

5.38.4 setAsBlob(data as MemoryBlock)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as BLob data using a memoryblock.
See also:

5.38.5 setAsBlob(data as SQLDataProviderMBS, BlockSize as UInt32)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as BLob data (SQLString)
Notes: When you call the SQLCommandMBS.Execute method all input parameters are bound with their values, including Long and BLob(CLob) parameters.
That is the time when the data provider Read event runs to get the values in the block size you specify.

The default value for the block size is 0. If you use the default value, SQLAPI++ Library will automatically use the most appropriate size for current DBMS.
See also:

5.38.6 setAsBlob(data as SQLStringMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as BLob data (SQLString)
See also:

5.38.7 setAsBlob(data as string)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as BLob data (SQLString)
Example:

```
Var c as SQLCommandMBS // your command object
Var JPEGData as string // some data

c.Param("imageData").setAsBlob JPEGData
```

See also:

- 5.38.4 setAsBlob(data as MemoryBlock) 252
- 5.38.5 setAsBlob(data as SQLDataProviderMBS, BlockSize as UInt32) 253
- 5.38.6 setAsBlob(data as SQLStringMBS) 253
- 5.38.8 setAsBlob(file as folderItem) 254
- 5.38.9 setAsBlob(stream as Readable) 255

5.38.8 setAsBlob(file as folderItem)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets field with content of file.
Example:

```
Var cmd as SQLCommandMBS // your command object

// pass folderitem to BLOB field
Var f as FolderItem = SpecialFolder.Desktop.Child("test.jpg")
cmd.Param("image").setAsBlob(f)
```

Notes: The file will be read later when statement is executed.
May not work when using preemptive threads.
See also:

- 5.38.4 setAsBlob(data as MemoryBlock) 252
- 5.38.5 setAsBlob(data as SQLDataProviderMBS, BlockSize as UInt32) 253
- 5.38.6 setAsBlob(data as SQLStringMBS) 253
- 5.38.7 setAsBlob(data as string) 254
- 5.38.9 setAsBlob(stream as Readable) 255

5.38.9 setAsBlob(stream as Readable)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets field with content of stream.
Example:

```
Var cmd as SQLCommandMBS // your command object

// pass BinaryStream to BLOB field
Var f as FolderItem = SpecialFolder.Desktop.Child("test.jpg")
Var b as BinaryStream = BinaryStream.open(f)
cmd.Param("image").setAsBlob(b)
```

Notes: The stream will be read later when statement is executed.
May not work when using preemptive threads.
See also:

- 5.38.4 setAsBlob(data as MemoryBlock) 252
- 5.38.5 setAsBlob(data as SQLDataProviderMBS, BlockSize as UInt32) 253
- 5.38.6 setAsBlob(data as SQLStringMBS) 253
- 5.38.7 setAsBlob(data as string) 254
- 5.38.8 setAsBlob(file as folderItem) 254

5.38.10 setAsBool(value as boolean)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as bool data.

5.38.11 setAsBytes(data as MemoryBlock)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as binary data from memoryblock.
See also:

- 5.38.12 setAsBytes(data as string) 256
- 5.38.13 setAsBytes(value as SQLBytesMBS) 256
- 5.38.14 setAsBytes(value as SQLStringMBS) 256

5.38.12 setAsBytes(data as string)

Plugin Version: 14.0, Platforms: macOS, Linux, Windows, Targets: All.

Function: Set the value to be bytes with content of given string.
Notes: For BLOB fields or parameters.
Same as the other variant, but avoids creating extra SQLStringMBS object.
See also:

- 5.38.11 setAsBytes(data as MemoryBlock) 255
- 5.38.13 setAsBytes(value as SQLBytesMBS) 256
- 5.38.14 setAsBytes(value as SQLStringMBS) 256

5.38.13 setAsBytes(value as SQLBytesMBS)

Plugin Version: 16.1, Platforms: macOS, Linux, Windows, Targets: All.

Function: Set the value to be bytes with content of given Bytes object.
Notes: For BLOB fields or parameters.
Same as the other variant, but avoids creating extra SQLStringMBS object.
See also:

- 5.38.11 setAsBytes(data as MemoryBlock) 255
- 5.38.12 setAsBytes(data as string) 256
- 5.38.14 setAsBytes(value as SQLStringMBS) 256

5.38.14 setAsBytes(value as SQLStringMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as binary string data (SQLString).
See also:

- 5.38.11 setAsBytes(data as MemoryBlock) 255
- 5.38.12 setAsBytes(data as string) 256
- 5.38.13 setAsBytes(value as SQLBytesMBS) 256

5.38.15 setAsClob(data as MemoryBlock)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Private method.
Notes: This method should make sure you don't accidentally a memoryblock instead of a text.
See also:

5.38.16 setAsClob(data as SQLDataProviderMBS, BlockSize as UInt32)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as CLob data (SQLString)
Notes: nWhen you call the SQLCommandMBS.Execute method all input parameters are bound with their values, including Long and BLob(CLob) parameters.
That is the time when the data provider Read event runs to get the values in the block size you specify.

The default value for the block size is 0. If you use the default value, SQLAPI++ Library will automatically use the most appropriate size for current DBMS.
See also:

5.38.17 setAsClob(file as folderItem)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets field with content of file.
Notes: The file will be read later when statement is executed.
May not work when using preemptive threads.
See also:

5.38.18 setAsClob(stream as Readable)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets field with content of stream.
Notes: The stream will be read later when statement is executed.
May not work when using preemptive threads.
See also:

5.38.19 setAsClob(text as SQLStringMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as CLob data (SQLString)
See also:

5.38.20 setAsClob(text as string)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as CLob data (SQLString)
See also:

5.38.21 setAsDate(value as date)

Plugin Version: 11.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Sets parameter's value with a date.
Example:

```
Var cmd as SQLCommandMBS // your command object
Var d as new date(2012,12,24,16,0,0) // some date
cmd.Param(3).setAsDate(d) // set third parameter
```

Notes: Same as setAsDateTime, but here we take a date object to make it more convenient.

5.38.22 setAsDateTime(value as dateTime)

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value with a dateTime.
Notes: Same as setAsDate, but here we take a dateTime object to make it more convenient.
See also:

5.38.23 setAsDateTime(value as SQLDateTimeMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as SQLDateTime data.
See also:

5.38.24 setAsDefault

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Forces to use the default parameter's value.
Notes: Forces DBMS API to use the default parameter value as the input value for an input or input/output parameter in a procedure.

If DBMS API does not support the concept of "default parameter values" in stored procedures, this setting will be ignored.

If you set this flag for the parameter that doesn't have a default value, the effect is DBMS defined (e.g. an error can be returned or NULL can be bound).

To cancel using the default parameter value you should call any other SQLValue::setAs... method to bind a parameter value.

To check whether this flag is set or not use isDefault method.

5.38.25 setAsDouble(value as Double)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as Double data.

5.38.26 setAsInt32(value as Int32)

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as Int32 data.

5.38.27 setAsInt64(value as Int64)

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as Int64 data.

5.38.28 setAsInteger(value as Integer)

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as Integer data.
Notes: Int32 or Int64 depending whether app is 32 or 64 bit.

5.38.29 setAsInterval(value as SQLIntervalMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets an interval value.

5.38.30 setAsLong(value as Int32)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as Int32 data.

5.38.31 setAsLongBinary(data as MemoryBlock)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as long binary data from memoryblock.
See also:

5.38.32 setAsLongBinary(data as SQLDataProviderMBS, BlockSize as UInt32)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as long binary data (SQLString)
Notes: When you call the SQLCommandMBS.Execute method all input parameters are bound with their values, including Long and BLob(CLob) parameters.
That is the time when the data provider Read event runs to get the values in the block size you specify.

The default value for the block size is 0. If you use the default value, SQLAPI++ Library will automatically use the most appropriate size for current DBMS.
See also:

5.38.33 setAsLongBinary(data as SQLStringMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as long binary data (SQLString)
See also:

5.38.34 setAsLongBinary(data as string)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as long binary data (SQLString)
See also:

5.38.35 setAsLongBinary(file as folderItem)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets field with content of file.
Notes: The file will be read later when statement is executed.
May not work when using preemptive threads.
See also:

- 5.38.31 setAsLongBinary(data as MemoryBlock) 261
- 5.38.32 setAsLongBinary(data as SQLDataProviderMBS, BlockSize as UInt32) 261
- 5.38.33 setAsLongBinary(data as SQLStringMBS) 262
- 5.38.34 setAsLongBinary(data as string) 262
- 5.38.36 setAsLongBinary(stream as Readable) 263

5.38.36 setAsLongBinary(stream as Readable)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets field with content of stream.
Notes: The stream will be read later when statement is executed.
May not work when using preemptive threads.
See also:

- 5.38.31 setAsLongBinary(data as MemoryBlock) 261
- 5.38.32 setAsLongBinary(data as SQLDataProviderMBS, BlockSize as UInt32) 261
- 5.38.33 setAsLongBinary(data as SQLStringMBS) 262
- 5.38.34 setAsLongBinary(data as string) 262
- 5.38.35 setAsLongBinary(file as folderItem) 263

5.38.37 setAsLongChar(data as MemoryBlock)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Private method.
Notes: This method should make sure you don't accidentally a memoryblock instead of a text.
See also:

- 5.38.38 setAsLongChar(data as SQLDataProviderMBS, BlockSize as UInt32) 264
- 5.38.39 setAsLongChar(file as folderItem) 264

5.38.38 setAsLongChar(data as SQLDataProviderMBS, BlockSize as UInt32)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as long character data (SQLString)
Notes: When you call the SQLCommandMBS.Execute method all input parameters are bound with their values, including Long and BLob(CLob) parameters.
That is the time when the data provider Read event runs to get the values in the block size you specify.

The default value for the block size is 0. If you use the default value, SQLAPI++ Library will automatically use the most appropriate size for current DBMS.
See also:

5.38.39 setAsLongChar(file as folderItem)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets field with content of file.
Notes: The file will be read later when statement is executed.
May not work when using preemptive threads.
See also:

5.38.40 setAsLongChar(stream as Readable)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets field with content of stream.
Notes: The stream will be read later when statement is executed.
May not work when using preemptive threads.
See also:

- 5.38.37 setAsLongChar(data as MemoryBlock) 263
- 5.38.38 setAsLongChar(data as SQLDataProviderMBS, BlockSize as UInt32) 264
- 5.38.39 setAsLongChar(file as folderItem) 264
- 5.38.41 setAsLongChar(text as SQLStringMBS) 265
- 5.38.42 setAsLongChar(text as string) 265

5.38.41 setAsLongChar(text as SQLStringMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as long character data (SQLString)
See also:

- 5.38.37 setAsLongChar(data as MemoryBlock) 263
- 5.38.38 setAsLongChar(data as SQLDataProviderMBS, BlockSize as UInt32) 264
- 5.38.39 setAsLongChar(file as folderItem) 264
- 5.38.40 setAsLongChar(stream as Readable) 265
- 5.38.42 setAsLongChar(text as string) 265

5.38.42 setAsLongChar(text as string)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as long character data (SQLString)
See also:

- 5.38.37 setAsLongChar(data as MemoryBlock) 263
- 5.38.38 setAsLongChar(data as SQLDataProviderMBS, BlockSize as UInt32) 264
- 5.38.39 setAsLongChar(file as folderItem) 264
- 5.38.40 setAsLongChar(stream as Readable) 265
- 5.38.41 setAsLongChar(text as SQLStringMBS) 265

5.38.43 setAsNull

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets value to null.

5.38.44 setAsNumeric(value as SQLNumericMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as SQLNumeric data.

5.38.45 setAsShort(value as Int16)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as short data.

5.38.46 setAsString(data as MemoryBlock)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Private method.
Notes: This method should make sure you don't accidentally a memoryblock instead of a text.
See also:

- 5.38.47 setAsString(value as SQLStringMBS) 266
- 5.38.48 setAsString(value as string) 267

5.38.47 setAsString(value as SQLStringMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as character string data (SQLString)
See also:

- 5.38.46 setAsString(data as MemoryBlock) 266
- 5.38.48 setAsString(value as string) 267

5.38.48 setAsString(value as string)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as character string data.
Notes: Same as setAsString, but for your convenience with a Xojo string instead of a SQLStringMBS object.
See also:

- 5.38.46 setAsString(data as MemoryBlock) 266
- 5.38.47 setAsString(value as SQLStringMBS) 266

5.38.49 setAsUInt32(value as UInt32)

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as UInt32 data.

5.38.50 setAsULong(value as UInt32)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as unsigned long data.

5.38.51 setAsUnknown

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's type as unknown.

5.38.52 setAsUShort(value as UInt16)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value as unsigned short data.
Notes: Sets value as unsigned short data.

5.38.53 setAsValueRead(value as SQLValueReadMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets parameter's value from SQLParam or SQLField objects.
Notes: This method allows using SQLField or SQLParam object received from one SQL statement as a parameter for another SQL statement.

5.38.54 setVariant(value as Variant)

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Sets the value based on a variant.
Example:

```
Var con as SQLConnectionMBS // your connection
Var pic as picture // some picture

// get picture data
Var jpegData as MemoryBlock = pic.GetData(Picture.FormatJPEG, 80)

// get a command for an Insert command
Var sql as string = "Insert into BlobTest(name, image) values (:name, :image)"
Var cmd as new SQLCommandMBS(con, sql)

// set values by variant
cmd.Param("name").setVariant "logo.jpg"
cmd.Param("image").setVariant jpegData

// do the insert
cmd.Execute
```

Notes: MemoryBlock and Strings without text encoding are converted to byte values (BLOB).
Texts and Strings with encoding are converted to text values.
Raises exceptions if you pass anything which is not recognized.
Other types are translated as good as possible.

With version 19.0 or later, you can pass folderitem to stream blob from file. This may raise exception if file can't be opened.

5.38.55 Properties

5.38.56 isDefault as boolean

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns true if the plugin has been forced to use parameter's default value by calling setAsDefault method; false otherwise.

Notes: (Read only property)

5.39 class SQLValueReadMBS

5.39.1 class SQLValueReadMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The class used in the SQL Plugin for value objects which can be read.
Blog Entries

- MBS Xojo Plugins, version 20.5pr9
- MBS Xojo Plugins, version 20.5pr7
- MBS Xojo Plugins, version 19.5pr2
- MBS Xojo Plugins in version 19.2
- MBS Xojo Plugins, version 19.2pr1
- MBS Xojo / Real Studio Plugins, version 16.4pr5
- Text data type
- MBS Real Studio Plugins, version 12.5pr4
- Release notes for SQL or ChartDirector?
- MBS Real Studio Plugins, version 11.2pr3

5.39.2 Methods

5.39.3 asBLob as SQLStringMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as BLob (SQLString) data.
Notes: If the value of current object is NULL, asBLob method returns an empty string. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is string (kDataTypeString), bytes (kDataTypeBytes), long binary (kDataTypeLongBinary), long character (kDataTypeLongChar), BLob (kDataTypeBLob) or CLob (kDataTypeCLob), asBLob method returns the object's value as SQLString object.

If the value's type of current object is bool (kDataTypeBool), short (kDataTypeShort), long (kDataTypeLong), double (kDataTypeDouble), numeric (kDataTypeNumeric), date-time (kDataTypeDateTime) or cursor (kDataTypeCursor), the result is undefined and debug version asserts.

Use DataType method to get the value's type of SQLValueRead object.

5.39.4 asBLobMemory as MemoryBlock

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as BLob data in a memoryblock.

5.39.5 asBLobString as String

Plugin Version: 16.4, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as BLob data in a string.

5.39.6 asBytes as SQLStringMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as binary string data (SQLString).
Notes: If the value of current object is NULL, asBytes method returns an empty string. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is string (kDataTypeString), bytes (kDataTypeBytes), long binary (kDataTypeLongBinary), long character (kDataTypeLongChar), BLob (kDataTypeBLob) or CLob (kDataTypeCLob), asBytes method returns the object's value as SQLString object.

If the value's type of current object is bool (kDataTypeBool), short (kDataTypeShort), long (kDataTypeLong), double (kDataTypeDouble), numeric (kDataTypeNumeric) or date-time (kDataTypeDateTime), asBytes method returns a block of data with size sizeof(value's type) as SQLString object.

If the value's type of current object is cursor (kDataTypeCursor), the result is undefined and debug version asserts.

Use DataType method to get the value's type of SQLValueRead object.

5.39.7 asCLob as SQLStringMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as CLob (SQLString) data.
Notes: If the value of current object is NULL, asCLob method returns an empty string. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is string (kDataTypeString), bytes (kDataTypeBytes), long binary (kDataTypeLongBinary), long character (kDataTypeLongChar), BLob (kDataTypeBLob) or CLob (kDataTypeCLob), asCLob method returns the object's value as SQLString object.

If the value's type of current object is bool (kDataTypeBool), short (kDataTypeShort), long (kDataTypeLong), double (kDataTypeDouble), numeric(kDataTypeNumeric), date-time (kDataTypeDateTime) or cursor (kDataTypeCursor), the result is undefined and debug version asserts.

Use DataType method to get the value's type of SQLValueRead object.

5.39.8 asDate as Date

Plugin Version: 11.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

Function: Returns the value as date.
Example:

```
Var cmd as SQLCommandMBS // your command object
Var d as date = cmd.Field("mydate").asDate // read date value
```

Notes: If the value of current object is NULL, asDate method returns an empty date object. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is kDataTypeDateTime, asDate method returns date object.

If the value's type of current object is any data type except kDataTypeDateTime, the result is undefined and debug version asserts.

Use DataType method to get the value's type of SQLValueRead object.

5.39.9 asDateTime as SQLDateTimeMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as SQLDateTimeMBS object.
Notes: If the value of current object is NULL, asDateTime method returns an empty SQLDateTime object. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is kDataTypeDateTime, asDateTime method returns SQLDateTime object.

If the value's type of current object is any data type except kDataTypeDateTime, the result is undefined and debug version asserts.

Use DataType method to get the value's type of SQLValueRead object.

5.39.10 asDateTimeValue as DateTime

Plugin Version: 20.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as date.
Notes: If the value of current object is NULL, asDate method returns an empty date object. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is kDataTypeDateTime, asDate method returns date object.

If the value's type of current object is any data type except kDataTypeDateTime, the result is undefined and debug version asserts.

Use DataType method to get the value's type of SQLValueRead object.

5.39.11 asInterval as SQLIntervalMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as interval (SQLIntervalMBS).

5.39.12 asLongBinary as SQLStringMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as long binary (SQLString) data.
Notes: If the value of current object is NULL, asLongBinary method returns an empty string. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is string (kDataTypeString), bytes (kDataTypeBytes), long binary (kDataTypeLongBinary), long character (kDataTypeLongChar), BLob (kDataTypeBLob) or CLob (kDataTypeCLob), asLongBinary method returns the object's value as SQLString object.

If the value's type of current object is bool (kDataTypeBool), short (kDataTypeShort), long (kDataType-

Long), double (kDataTypeDouble), numeric (kDataTypeNumeric), date-time (kDataTypeDateTime) or cursor (kDataTypeCursor), the result is undefined and debug version asserts.

Use DataType method to get the value's type of SQLValueRead object.

5.39.13 asLongChar as SQLStringMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as long character (SQLString) data.
Notes: If the value of current object is NULL, asLongChar method returns an empty string. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is string (kDataTypeString), bytes (kDataTypeBytes), long binary (kDataTypeLongBinary), long character (kDataTypeLongChar), BLob (kDataTypeBLob) or CLob (kDataTypeCLob), asLongChar method returns the object's value as SQLString object.

If the value's type of current object is bool (kDataTypeBool), short (kDataTypeShort), long (kDataTypeLong), double (kDataTypeDouble), numeric (kDataTypeNumeric), date-time (kDataTypeDateTime) or cursor (kDataTypeCursor), the result is undefined and debug version asserts.

Use DataType method to get the value's type of SQLValueRead object.

5.39.14 asNumeric as SQLNumericMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as SQLNumeric data.
Notes: If the value of current object is NULL, asNumeric method returns 0. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is exact numeric value (kDataTypeNumeric), asNumeric method returns the original value with no conversion.

If the value's type of current object is bool (kDataTypeBool), short (kDataTypeShort), double (kDataTypeDouble) or long (kDataTypeLong), asNumeric method converts it to SQLNumeric .

If the value's type of current object is string (kDataTypeString), asNumeric method tries to convert it from SQLChar* value. If the conversion is possible and correct, asNumeric converts to SQLNumeric from SQLChar* value. If conversion is incorrect asNumeric method throws an exception.

If the value's type of current object is any data type except the described above, the result is undefined and debug version asserts.

Use DataType method to get the value's type of SQLValueRead object.

5.39.15 asString as SQLStringMBS

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: returns the value as string (SQLString) data.
Notes: If the value of current object is NULL, asString method returns an empty string. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is bool (kDataTypeBool), asString method returns "true" or "false" string (SQLString object).

If the value's type of current object is short (kDataTypeShort), asString method converts it to decimal string (SQLString object) like function printf("%hd", ...) does.

If the value's type of current object is long (kDataTypeLong), asString method converts it to decimal string (SQLString object) like function printf("%ld", ...) does.

If the value's type of current object is double (kDataTypeDouble), asString method converts it to decimal string (SQLString object) like function printf("%g", ...) does.

If the value's type of current object is numeric (kDataTypeNumeric), asString method converts it to decimal string (SQLString object) without precision loss.

If the value's type of current object is date-time (kDataTypeDateTime), asString method converts it to string (SQLString object) like function asctime(...) does.

If the value's type of current object is string (kDataTypeString, kDataTypeLongChar, kDataTypeCLob), asString method returns the original object's value as SQLString object.

If the value's type of current object is binary (kDataTypeBytes, kDataTypeLongBinary, kDataTypeBLob), asString method converts it to hexadecimal string (SQLString object).

If the value's type of current object is cursor (kDataTypeCursor), the result is undefined and debug version asserts.

Use DataType method to get the value's type of SQLValueRead object.
For numbers, this may give english decimal separator. For getting localized one, please use AsDoubleValue and use cstr() function.

5.39.16 Constructor(DataType as Integer)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new empty value object for the given data type.
See also:

- 5.39.17 Constructor(value as SQLValueReadMBS) 276

5.39.17 Constructor(value as SQLValueReadMBS)

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Creates a new value object by copying the given one.
See also:

- 5.39.16 Constructor(DataType as Integer) 276

5.39.18 Properties

5.39.19 asBool as boolean

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value of current object as bool data.
Notes: f the value of current object is NULL, asBool method returns false. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is bool (kDataTypeBool), asBool method returns the original value with no conversion.

If the value's type of current object is short (kDataTypeShort), long (kDataTypeLong) or double (kDataTypeDouble), asBool method converts it to bool data type. Conversion returns false if the value is 0; true otherwise.

If the value's type of current object is any data type except the described above, the result is undefined and debug version asserts.

Use DataType method to get the value's type of SQLValueRead object.
(Read only property)

5.39.20 asDouble as Double

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: returns the value as Double data.
Notes: If the value of current object is NULL, asDouble method returns 0. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is double (kDataTypeDouble), asDouble method returns the original value with no conversion.

If the value's type of current object is bool (kDataTypeBool), short (kDataTypeShort), long (kDataTypeLong) or numeric (kDataTypeNumeric), asDouble method converts it to double (kDataTypeDouble) data type.

If the value's type of current object is string (kDataTypeString), asDouble method tries to convert it to double value. If the conversion is possible and correct, asDouble returns kDataTypeDouble value. If conversion is incorrect asDouble method throws an exception.

If the value's type of current object is kDataTypeDateTime, asDouble method converts it to standard double representation. Days are represented by whole number increments starting with 30 December 1899, midnight as time zero. Hour values are expressed as the absolute value of the fractional part of the number. See SQLDateTime::operator double() for more detailes.

If the value's type of current object is any data type except the described above, the result is undefined and debug version asserts.

Use DataType method to get the value's type of SQLValueRead object.
(Read only property)

5.39.21 asInt32 as Int32

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as Int32 data.
Notes: (Read only property)

5.39.22 asInt64 as Int64

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as Int64 data.
Notes: (Read only property)

5.39.23 asInteger as Integer

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as Integer data.
Notes: Either Int32 or Int64 depending on whether the application is 32 or 64 bit.
(Read only property)

5.39.24 asLong as Int32

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as Int32 data.
Notes: If the value of current object is NULL, asLong method returns 0. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is long (kDataTypeLong), asLong method returns the original value with no conversion.

If the value's type of current object is bool (kDataTypeBool), short (kDataTypeShort), double (kDataTypeDouble) or numeric (kDataTypeNumeric), asLong method converts it to long data type. Note, that in this case the returned value can be truncated.

If the value's type of current object is string (kDataTypeString), asLong method tries to convert it to long (kDataTypeLong) value. If the conversion is possible and correct, asLong returns kDataTypeLong value. If conversion is incorrect asLong method throws an exception.

If the value's type of current object is any data type except the described above, the result is undefined and debug version asserts.

Use DataType method to get the value's type of SQLValueRead object.
(Read only property)

5.39.25 asShort as Int16

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as short.
Notes: If the value of current object is NULL, asShort method returns 0. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is short, asShort method returns the original value with no conversion.

If the value's type of current object is bool (kDataTypeBool), long (kDataTypeLong), unsigned long (SkDataTypeULong), double (kDataTypeDouble) or numeric (kDataTypeNumeric), asShort method converts it to short data type. Note, that in this case the returned value can be truncated.

If the value's type of current object is string (kDataTypeString), asShort method tries to convert it to short value. If the conversion is possible and correct, asShort returns the value. If conversion is incorrect asShort method throws an exception.

If the value's type of current object is any data type except the described above, the result is undefined and debug version asserts.

Use DataType method to get the value's type of SAValueRead object.
(Read only property)

5.39.26 asStringValue as String

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value string.
Notes: Same as asString but returns a Xojo string.
Please use SQLStringMBS and CopyBinaryData if the string you want to read is a BLOB value, else text encoding will change your data!
For numbers, this may give english decimal separator. For getting localized one, please use AsDoubleValue and use cstr() function.
(Read only property)

5.39.27 asUInt32 as UInt32

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as UInt32 data.
Notes: (Read only property)

5.39.28 asULong as UInt32

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as an unsigned 32 bit integer value.
Notes: If the value of current object is NULL, asULong method returns 0. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is long (kDataTypeLong), asULong method returns the original value with no conversion.

If the value's type of current object is bool (kDataTypeBool), short (kDataTypeShort), double (kDataTypeDouble) or numeric (kDataTypeNumeric), asULong method converts it to long data type. Note, that in this case the returned value can be truncated.

If the value's type of current object is string (kDataTypeString), asULong method tries to convert it to long (kDataTypeLong) value. If the conversion is possible and correct, asULong returns kDataTypeLong value. If conversion is incorrect asULong method throws an exception.

If the value's type of current object is any data type except the described above, the result is undefined and debug version asserts.

Use DataType method to get the value's type of SQLValueRead object.
(Read only property)

5.39.29 asUShort as UInt16

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as unsigned short.
Notes: If the value of current object is NULL, asUShort method returns 0. Use isNull method to make sure if the value is NULL or not.

If the value's type of current object is unsigned short, asUShort method returns the original value with no conversion.

If the value's type of current object is bool (kDataTypeBool), long (kDataTypeLong), unsigned long (Sk-

DataTypeULong), double (kDataTypeDouble) or numeric (kDataTypeNumeric), asUShort method converts it to unsigned short data type. Note, that in this case the returned value can be truncated.

If the value's type of current object is string (kDataTypeString), asUShort method tries to convert it to unsigned short value. If the conversion is possible and correct, asUShort returns the value. If conversion is incorrect asUShort method throws an exception.

If the value's type of current object is any data type except the described above, the result is undefined and debug version asserts.

Use DataType method to get the value's type of SAValueRead object.
(Read only property)

5.39.30 asVariant as Variant

Plugin Version: 12.5, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns the value as a variant.
Notes: This is a convenience function.
May return nil, date, number or string.
BLOB strings without encoding and text strings as UTF-8.
(Read only property)

5.39.31 DataType as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns SAValueRead object's data type.
Notes: One of the kDataType constants.
(Read only property)

5.39.32 isNull as boolean

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: Returns true if the value of current object is NULL; otherwise false.
Notes: (Read only property)

5.39.33 LongOrLobReaderMode as Integer

Plugin Version: 9.3, Platforms: macOS, Linux, Windows, Targets: All.

Function: The Long or Lob data reading mode.
Notes: SQLAPI++ Library provides two ways to read Long or BLob(CLob) object's value (usually SQLField or SQLParam objects):

1. reading of Long or Lob data at once into an internal buffer (like ordinary string or binary values);
2. piecewise reading of Long or Lob data using user defined callback.
kLongOrLobReaderDefault reading mode used by default.

If you want to control piecewise reading of Long or BLob(CLob) data you should set LongOrLobReaderMode and use kLongOrLobReaderManual reading mode for Long or BLob(CLob) parameters or fields you want to process with your data consumer. After that each fetch will skip reading Long and BLob(CLob) parameters that you set to be read manually. To read field or parameter defined to be read manually you should call ReadLongOrLob method for each of them after the fetch. ReadLongOrLob method will repeatedly call the data consumer Write event.
(Read and Write property)

5.39.34 Constants

Constants

Constant	Value	Description
kDataTypeInt64	6	One of the field type constants Signed 64-bit integer.
kDataTypeUInt64	7	One of the field type constants Unsigned 64-bit integer.

Data Types

Constant	Value	Description
kDataTypeBlob	16	Data type is BLob data (SQLStringMBS).
kDataTypeBool	1	Data type is a boolean.
kDataTypeBytes	13	Data type is binary string (SQLStringMBS).
kDataTypeClob	17	Data type is CLob data (SQLStringMBS).
kDataTypeCursor	18	Data type is Oracle REF CURSOR (SQLCommand).
kDataTypeDateTime	10	Data type is SQLDateTime.
kDataTypeDouble	8	This is a normal double variable.
kDataTypeInterval	11	Data type is an interval (SQLIntervalMBS).
kDataTypeLong	4	A 32 bit integer.
kDataTypeLongBinary	14	Data type is long binary data (SQLStringMBS).
kDataTypeLongChar	15	Data type is long character data (SQLStringMBS).
kDataTypeNumeric	9	Data type is SQLNumeric (used internally).
kDataTypeShort	2	Data type is a 16 bit signed integer.
kDataTypeSpecificToDBMS	19	Data type is server-specific.
kDataTypeString	12	Data type is character string (SQLString).
kDataTypeULong	5	Data type is a 32 bit unsigned integer.
kDataTypeUnknown	0	Data type is unknown.
kDataTypeUShort	3	A 16 bit unsigned integer.

Read Modes

Constant	Value	Description
kLongOrLobReaderModeDefault	0	Long or Lob(CLob) data reading mode is default (automatic).
kLongOrLobReaderModeManual	1	Long or Lob(CLob) data reading mode is manual.

Chapter 6

List of Questions in the FAQ

Chapter 7

The FAQ

7.0.1 API client not supported?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: If you get this exception message on SQLConnectionMBS.Connect, we may have a problem.
Notes: First case is that the given thing is not supported (e.g. MS SQL directly on Mac).
Second case is that the plugin compilation went wrong and the support for the database was not linked into the plugin. Like MySQL missing or MS SQL on Windows missing. In that case please contact us to fix the plugin.

7.0.2 Can I access Access Database with Java classes?

Plugin Version: all, Platform: Windows.

Answer: You can use ucanaccess to access databases created with Microsoft
Example:

```
Var options(-1) as string

// load all the jar files we have in a folder called java:

Var appFolder as FolderItem = GetFolderItem("")

Var count as Integer = appFolder.Parent.Child("java").Count
Var libjs() as string
For i as Integer = 1 to count
Var f As FolderItem = appFolder.Parent.Child("java").item(i)
If f <>Nil and f.Exists Then
libjs.append f.NativePath+";"
End If
```

```
Next

// now init virtual machine
Var librery as string = Join(libjs, "")
Var vm as new JavaVMMBS(librery)

if vm.Handle = 0 then
MsgBox "Failed to initialize virtual machine"
else
// now make a new database connection with ucanaccess
Var d as new JavaDatabaseMBS(vm,"net.ucanaccess.jdbc.UcanaccessDriver")
Var DbFile as FolderItem = appFolder.Parent.Child("Database11.accdb")
Var j as JavaConnectionMBS = d.getConnection("jdbc:ucanaccess://"+DbFile.NativePath)

// select and show values
Var r as JavaResultSetMBS = j.MySelectSQL("Select * From test")
while r.NextRecord
MsgBox r.getString("FirstName") +" "+ r.getString("LastName")
wend

end if

Exception e as JavaExceptionMBS
MsgBox e.message+" errorcode: "+str(e.ErrorNumber)
```

Notes: see website:
http://ucanaccess.sourceforge.net/site.html

7.0.3 Does SQL Plugin handle stored procedures with multiple result sets?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, the plugin can work with multiple recordsets.
Notes: You need to use SQLCommandMBS class. When you get back results, you use FetchNext to walk over all records in the first result set. Than you simply start again with FetchNext to get the second record set.
Even the RecordSet functions should work, just use them twice to get all records from both record sets.

7.0.4 How to create a birthday like calendar event?

Plugin Version: all, Platform: macOS.

Answer: Try this code:
Example:

```
// start a connection to the calendar database
Var s as new CalCalendarStoreMBS

// needed for the error details
Var e as NSErrorMBS

Var r as CalRecurrenceRuleMBS = CalRecurrenceRuleMBS.initYearlyRecurrence(1, nil) // repeat every year without end

Var a as new CalAlarmMBS // add alarm
a.action = a.CalAlarmActionDisplay
a.relativeTrigger = -3600*24 // 24 Hours before

// create a new calendar
Var c as new CalEventMBS

Var d as new date(2011, 04, 20) // the date

Var calendars() as CalCalendarMBS = s.calendars

// set properties
c.Title="Test Birthday"
c.startDate=d
c.recurrenceRule = r
c.calendar=calendars(0) // add to first calendar
c.addAlarm(a)
c.endDate = d
c.isAllDay = true

// save event
call s.saveEvent(c,s.CalSpanAllEvents, e)
if e<>nil then
MsgBox e.localizedDescription
else
MsgBox "New event was created."
end if
```

Notes: This adds an event to iCal for the given date with alarm to remember you and repeats it every year.

7.0.5 How to make a lot of changes to a REAL SQL Database faster?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You may try to embed your changes to the database between two transaction calls.
Example:

```
Var db as Database // some database

db.SQLExecute "BEGIN TRANSACTION"
// Do some Stuff
db.SQLExecute "END TRANSACTION"
```

Notes: This can increase speed by some factors.

7.0.6 How to set cache size for SQLite or REALSQLDatabase?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You use the pragma cache_size command on the database.
Example:

```
// set cache size to 20000 pages which is about 20 MB for default page size
Var db as REALSQLDatabase
db.SQLExecute "PRAGMA cache_size = 20000"
```

Notes: Default cache size is 2000 pages which is not much.
You get best performance if whole database fits in memory.
At least you should try to have a cache big enough so you can do queries in memory.
You only need to call this pragma command once after you opened the database.

7.0.7 How to use Sybase in Web App?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use our MBS Xojo SQL Plugin to connect to a Sybase Database in your web application.
Notes: If you see db.Connect giving the error message "cs_ctx_alloc ->CS_MEM_ERROR", than some things are not setup right for Sybase.
The Apache process may not have all the SYBASE environment variables being set when the CGI was launched.

Adding these lines to /etc/httpd/conf/httpd.conf stopped the faux memory errors for us:

```
SetEnv LD_LIBRARY_PATH /opt/sybase/OCS-15_0/lib:/opt/sybase/OCS-15_0/lib3p64:/opt/sybase/OCS-15_0/lib3p:
SetEnv SYBROOT /opt/sybase
SetEnv SYBASE_OCS /opt/sybase
SetEnv SYBASE /opt/sybase
```

7.0.8 SQLDatabase not initialized error?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Before you can use SQLDatabaseMBS, it must be initialized.
Example:

```
Var d as new SQLDatabaseMBS
```

Notes: This happens normally when you use "new SQLDatabaseMBS".
But if you just have a SQLConnectionMBS and get a recordset there, the initialization may not have happend, yet.
So please simply add a line "dim d as new SQLDatabaseMBS" to your app.open code after registration, so the plugin part can initialize and late provide recordsets.

7.0.9 What is the difference between Timer and WebTimer?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Time is server side and WebTimer client side.
Notes: Timer is the normal timer class in Xojo. It runs on the server. On the side the WebTimer runs on the client. It triggers a request to the server to perform the action. So a WebTimer is good to keep the connection running and the website updated regularly. A timer on the server is good to make regular jobs like starting a database backup every 24 hours.

7.0.10 What to do with SQL Plugin reporting Malformed string as error?

Plugin Version: all, Platform: macOS.

Answer: Please make sure the table and/or database fields have a text encoding set.
Notes: For Firebird our plugin tries to use UTF-8 encoding if possible and to correctly convert between various tables, the tables and their fields need to have a text encoding defined.
e.g. if the text field in the table is windows-1252 and the other ISO 8859-5, then the Firebird database can convert them to UTF-8 and deliver texts to the plugin.
If encoding is set to none, it may get confused for non-ascii text.