# MBS Java 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 Java Plugin

## 0.2 Content

# Chapter 1

# List of Topics

# Chapter 2

# List of all classes

# Chapter 3

# Java

## 3.1 class JavaArrayMBS

### 3.1.1 class JavaArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for a java array object.
**Notes:** Subclass of the JavaObjectMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
**Blog Entries**

- Upgrading our Java Support for Xojo
- MBS Xojo Plugins, version 19.2pr5

### 3.1.2 Methods

### 3.1.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 3.1.4 Properties

### 3.1.5 Length as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The number of items in the array.
**Notes:** Returns 0 on any error.
(Read only property)

## 3.2 class JavaBooleanArrayMBS

### 3.2.1 class JavaBooleanArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for the java boolean array object.
**Notes:** Subclass of the JavaArrayMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

### 3.2.2 Methods

### 3.2.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 3.2.4 Elements as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** All the elements in this array as one memoryblock.
**Notes:** Use memoryblock.byte(index) to access.

### 3.2.5 Values as Boolean()

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns a Xojo array with values.

### 3.2.6 Properties

### 3.2.7 Region(start as Integer, len as Integer) as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set a region of the array.
**Notes:** Start is the starting index (0 based) and len the number of items to copy.
Throws ArrayIndexOutOfBoundsException if (start + len - 1) does not specify a valid index in the array.

(Read and Write computed property)

### 3.2.8 Value(index as Integer) as Boolean

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value with given index.
**Notes:** May raise OutOfBoundsException if index is out of range.
(Read and Write computed property)

## 3.3 class JavaByteArrayMBS

### 3.3.1 class JavaByteArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for the java byte array object.
**Notes:** Subclass of the JavaArrayMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

### 3.3.2 Methods

### 3.3.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 3.3.4 Elements as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** All the elements in this array as one memoryblock.
**Notes:** Use memoryblock.Byte(index) to access.

### 3.3.5 Values as Int8()

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns a Xojo array with values.

### 3.3.6 Properties

### 3.3.7 Region(start as Integer, len as Integer) as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set a region of the array.
**Notes:** Start is the starting index (0 based) and len the number of items to copy.
Throws ArrayIndexOutOfBoundsException if (start + len - 1) does not specify a valid index in the array.

(Read and Write computed property)

### 3.3.8 Value(index as Integer) as Int8

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value with given index.
**Notes:** May raise OutOfBoundsException if index is out of range.
(Read and Write computed property)

# 3.4 class JavaCharArrayMBS

## 3.4.1 class JavaCharArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for the java char array object.
**Notes:** Subclass of the JavaArrayMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

## 3.4.2 Methods

## 3.4.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

## 3.4.4 Elements as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** All the elements in this array as one memoryblock.
**Notes:** Use memoryblock.UShortMBS(index*2) to access.

## 3.4.5 Values as UInt16()

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns a Xojo array with values.

## 3.4.6 Properties

## 3.4.7 Region(start as Integer, len as Integer) as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set a region of the array.
**Notes:** Start is the starting index (0 based) and len the number of items to copy.
Throws ArrayIndexOutOfBoundsException if (start + len - 1) does not specify a valid index in the array.

(Read and Write computed property)

### 3.4.8 Value(index as Integer) as UInt16

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value with given index.
**Notes:** May raise OutOfBoundsException if index is out of range.
(Read and Write computed property)

# 3.5 class JavaClassMBS

## 3.5.1 class JavaClassMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for a Java class.
**Notes:** Subclass of the JavaObjectMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
**Blog Entries**

- News from the MBS Xojo Plugins Version 25.3
- Query Java classes in Xojo
- MBS Xojo Plugins, version 25.3pr1
- Trying Java 19 in Xojo
- MBS Xojo Plugins, version 19.4pr6
- Upgrading our Java Support for Xojo
- MBS Real Studio Plugins, version 12.5pr1

## 3.5.2 Methods

## 3.5.3 AllocateObject as JavaObjectMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Allocates a new Java object without invoking any of the constructors for the object. #
**Notes:** Returns a reference to the object or nil on any error.

Does not work for array classes.
Throws InstantiationException if the class is an interface or an abstract class.

## 3.5.4 CallStaticBooleanMethod(MethodID as JavaMethodMBS, args as memoryblock) as boolean

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a static method with a boolean return value.
**Notes:** This call invokes a static method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling JavaClassMBS.GetMethod().

The method ID must be derived from this class, not from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.5.5 CallStaticByteMethod(MethodID as JavaMethodMBS, args as memoryblock) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a static method with a byte return value.
**Notes:** This call invokes a static method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling JavaClassMBS.GetMethod().

The method ID must be derived from this class, not from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.5.6 CallStaticCharMethod(MethodID as JavaMethodMBS, args as memoryblock) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a static method with a char return value.
**Notes:** This call invokes a static method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling JavaClassMBS.GetMethod().

The method ID must be derived from this class, not from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.5.7 CallStaticDoubleMethod(MethodID as JavaMethodMBS, args as memoryblock) as Double

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a static method with a double return value.
**Notes:** This call invokes a static method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling JavaClassMBS.GetMethod().

The method ID must be derived from this class, not from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.5.8 CallStaticFloatMethod(MethodID as JavaMethodMBS, args as memoryblock) as single

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a static method with a float return value.
**Notes:** This call invokes a static method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling JavaClassMBS.GetMethod().

The method ID must be derived from this class, not from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.5.9 CallStaticIntMethod(MethodID as JavaMethodMBS, args as memoryblock) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a static method with an integer return value.
**Notes:** This call invokes a static method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling JavaClassMBS.GetMethod().

The method ID must be derived from this class, not from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.5.10 CallStaticLongMethod(MethodID as JavaMethodMBS, args as memoryblock) as Int64

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a static method with a long return value.
**Notes:** This call invokes a static method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling JavaClassMBS.GetMethod().

The method ID must be derived from this class, not from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.5.11 CallStaticMain(args() as string)

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a static main method and passes the String array.

### 3.5.12 CallStaticMethod(MethodID as JavaMethodMBS, args() as Variant) as Variant

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a static method.
**Example:**

```
Var vm As JavaVMMBS // your initialized VM

// 1. Find class
Var jclass As JavaClassMBS =vm.FindClass("stringtest")
```

```
if jclass=nil then
msgbox "Can't find stringtest class"
Return
end if

// 2. lookup method
Var jmethod3 As JavaMethodMBS = jclass.GetStaticMethod("setValue","(Ljava/lang/String;)V")
if jmethod3=nil then
MsgBox "Failed to find setValue method."
Return
end if

// 3. Call static method
Var param3() As Variant
param3.append "Hello World"
Call jclass.CallStaticMethod(jmethod3,param3)

MsgBox "done"
```

**Notes:** This call invokes a static method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling JavaClassMBS.GetMethod().

The method ID must be derived from this class, not from one of its superclasses.

Programmers place all arguments to the method in an args variant array that immediately follows the methodID argument.

This is generic version, where our plugin translates between native Xojo data types and Java data types. We support conversion of boolean, byte (integer), char (integer), short (integer), int (integer), long (int64), double, float (single) and Java objects. Objects can be JavaObjectMBS or subclasses including JavaStringMBS and the JavaArrayMBS subclasses. For your convenience you can pass in string and we convert to JavaStringMBS for you.

### 3.5.13 CallStaticObjectMethod(MethodID as JavaMethodMBS, args as memoryblock) as JavaObjectMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a static method with an object return value.
**Example:**

```
Var e as JavaVMMBS // global
```

```
// Call "public static String getMessage()" and "public static void setMessage(String theMessage)" from class "test".

Var jclass, sclass as JavaClassMBS
Var jmethod as JavaMethodMBS
Var jstring as JavaStringMBS
Var args as JavaObjectArrayMBS
Var m as MemoryBlock
Var s as String
Var jfield as JavaFieldMBS

jclass=e.FindClass("test")

if jclass=nil then
msgbox "Can't find test class"
else
jmethod = jclass.GetStaticMethod("setMessage", "(Ljava/lang/String;)V")

if jmethod=nil then
msgbox "Can't find HelloWorld.setMessage"
else
jstring = e.NewStringUTF8("Hello from Xojo!")
if jstring=nil then
msgbox "Out of memory"
else
m=NewMemoryBlock(8) // 8 bytes per parameter
m.Int64Value(0)=jstring.Handle

jclass.CallStaticVoidMethod(jmethod, m)

jmethod = jclass.GetStaticMethod("getMessage", "()Ljava/lang/String;")

if jmethod=nil then
msgbox "Can't find HelloWorld.getMessage"
else
m=NewMemoryBlock(8) // 8 bytes per parameter
m.Int64Value(0)=jstring.Handle

jstring=JavaStringMBS(jclass.CallStaticObjectMethod(jmethod,m))

MsgBox jstring.CopyStringUTF
end if
end if
end if
end if
```

**Notes:** This call invokes a static method on a Java object, according to the specified method ID. The

methodID argument must be obtained by calling JavaClassMBS.GetMethod().

The method ID must be derived from this class, not from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.5.14 CallStaticShortMethod(MethodID as JavaMethodMBS, args as memoryblock) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a static method with a short return value.
**Notes:** This call invokes a static method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling JavaClassMBS.GetMethod().

The method ID must be derived from this class, not from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.5.15 CallStaticVoidMethod(MethodID as JavaMethodMBS, args as memoryblock)

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a static method with no return value.
**Notes:** This call invokes a static method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling JavaClassMBS.GetMethod().

The method ID must be derived from this class, not from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.5.16 Constructor

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 3.5.17 Fields as JavaFieldMBS()

Plugin Version: 25.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Queries the list of fields in the class.
**Example:**

```
// List field names

Var vm as JavaVMMBS // your VM
Var jclass As JavaClassMBS = vm.FindClass("test")
Var fields() As JavaFieldMBS = jclass.Fields
Var fieldNames() As String

For Each field As JavaFieldMBS In fields
fieldNames.add field.Name
Next

MessageBox String.FromArray(fieldNames, EndOfLine)
```

**Notes:** This includes static fields.
Does not include items from super class.

### 3.5.18 GetField(name as string, sig as string) as JavaFieldMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Searches for the Field with the given Name and Signature.
**Notes:** Nil on any error.

The signature is a string derived from the field's type or method's arguments and return type, as shown here:

### 3.5.19 GetMethod(name as string, sig as string) as JavaMethodMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

| Java Type | Signature |
|---|---|
| boolean | Z |
| byte | B |
| char | C |
| short | S |
| int | I |
| long | L |
| float | F |
| double | D |
| void | V |
| objects | Lfully-qualified-class-name; |
| arrays | [ array-type |
| methods | (argument-types)return-type |

**Function:** Searches the method with the given name and signature.
**Example:**

```
Var jclass as JavaClassMBS
Var method as JavaMethodMBS

method=jclass.GetMethod("mymethod", "( [ Ljava/lang/String;)V")
```

**Notes:** Returns the method ID for an instance (non-static) method of a class or interface. The method may be defined in one of the the class's super classes and inherited by the class. The method is determined by its name and signature.

To obtain the method ID of a constructor, supply <init>as the method name and void (V) as the return type.

Nil on any error.

Throws NoSuchMethodError if the specified method cannot be found.

The signature is a string derived from the field's type or method's arguments and return type, as shown here:

### 3.5.20 GetStaticField(name as string, sig as string) as JavaFieldMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Searches the static field with the given name and signature.
**Notes:** Nil on any error.

| Java Type | Signature |
|---|---|
| boolean | Z |
| byte | B |
| char | C |
| short | S |
| int | I |
| long | L |
| float | F |
| double | D |
| void | V |
| objects | Lfully-qualified-class-name; |
| arrays | [ array-type |
| methods | (argument-types)return-type |

### 3.5.21 GetStaticMethod(name as string, sig as string) as JavaMethodMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns the method ID for a static method of a class. The method is specified by its name and signature.
**Example:**

```
Var jclass as JavaClassMBS
Var method as JavaMethodMBS

method=jclass.GetStaticMethod("main", "( [ Ljava/lang/String;)V")
```

**Notes:** Nil on any error.
e.g. the signature of the default static main method is "( [ Ljava/lang/String;)V" which means return type void at the end and before an array of string.

### 3.5.22 Methods as JavaMethodMBS()

Plugin Version: 25.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Queries the list of methods in this class.
**Example:**

```
// List method names
```

```
Var vm as JavaVMMBS // your VM
Var jclass As JavaClassMBS = vm.FindClass("test")
Var Methods() As JavaMethodMBS = jclass.Methods
Var MethodNames() As String

For Each Method As JavaMethodMBS In Methods
MethodNames.add Method.Name
Next

MessageBox String.FromArray(MethodNames, EndOfLine)
```

**Notes:** This includes constructors with name "<init>", private, protected, public and static methods.
Does not include items from super class.

### 3.5.23 NewObject(methodID as JavaMethodMBS, args as memoryblock) as JavaObjectMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Constructs a new Java object.
**Notes:** The method ID indicates which constructor method to invoke. This ID must be obtained by calling GetMethod with <init>as the method name and void (V) as the return type.

Returns nil on any error.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

Throws Java InstantiationException if the class is an interface or an abstract class.
See also:

- 3.5.24 NewObject(MethodID as JavaMethodMBS, args() as Variant) as JavaObjectMBS 53

### 3.5.24 NewObject(MethodID as JavaMethodMBS, args() as Variant) as JavaObjectMBS

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Constructs a new Java object.
**Example:**

```
Var vm as JavaVMMBS // your initialized VM

// 1. Find class
Var jclass As JavaClassMBS=vm.FindClass("stringtest")

If jclass = Nil Then
MsgBox "Can't find stringtest class"
Return
End If

// 2. Call constructor
Var jmethod As JavaMethodMBS = jclass.GetMethod("<init>", "(ILjava/lang/String;D)V")

If jmethod = Nil Then
MsgBox "Can't find constructor"
Return
End If

Var param() As Variant
// 1. Parameter with Integer: I
param.Append 123
// 2. Parameter with string: Ljava/lang/String;
param.Append "Hello World"
// 3. Parameter with double: D
param.Append 3.14
Var jobject As JavaObjectMBS = jclass.NewObject(jmethod, param)

If jobject = Nil Then
MsgBox "Constructor failed!?"
Else
MsgBox "OK: "+jobject.ClassName
End If
```

**Notes:** The method ID indicates which constructor method to invoke. This ID must be obtained by calling GetMethod with <init>as the method name and void (V) as the return type.

Returns nil on any error.

Programmers place all arguments to the method in an args variant array that immediately follows the methodID argument.

Throws Java InstantiationException if the class is an interface or an abstract class.

This is generic version, where our plugin translates between native Xojo data types and Java data types. We support conversion of boolean, byte (integer), char (integer), short (integer), int (integer), long (int64), dou-

ble, float (single) and Java objects. Objects can be JavaObjectMBS or subclasses including JavaStringMBS and the JavaArrayMBS subclasses. For your convenience you can pass in string and we convert to JavaStringMBS for you.
See also:

- 3.5.23 NewObject(methodID as JavaMethodMBS, args as memoryblock) as JavaObjectMBS 53

### 3.5.25 Superclass as JavaClassMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Points to the superclass of this class.
**Notes:** Nil if no superclass exists.

### 3.5.26 Properties

### 3.5.27 StaticBooleanField(TheField as JavaFieldMBS) as boolean

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a static boolean field in this class.
**Notes:** (Read and Write computed property)

### 3.5.28 StaticByteField(TheField as JavaFieldMBS) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a static byte field in this class.
**Notes:** (Read and Write computed property)

### 3.5.29 StaticCharField(TheField as JavaFieldMBS) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a static char field in this class.
**Notes:** (Read and Write computed property)

### 3.5.30 StaticDoubleField(TheField as JavaFieldMBS) as Double

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a static double field in this class.
**Notes:** (Read and Write computed property)

### 3.5.31 StaticField(TheField as JavaFieldMBS) as Variant

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a static field in this class.
**Example:**

```
Var jclass As JavaClassMBS=vm.FindClass("stringtest")

If jclass = Nil Then
MsgBox "Can't find stringtest class"
Return
End If

Var field As JavaFieldMBS = jclass.GetField("Name", "Ljava/lang/String;")
Var v As Variant = jclass.StaticField(field)
Var js As JavaStringMBS = v
MsgBox js
```

**Notes:** This is generic version, where our plugin translates between native Xojo data types and Java data types. We support conversion of boolean, byte (integer), char (integer), short (integer), int (integer), long (int64), double, float (single) and Java objects. Objects can be JavaObjectMBS or subclasses including JavaStringMBS and the JavaArrayMBS subclasses. For your convenience you can pass in string and we convert to JavaStringMBS for you.

Do not use for generic fields!
(Read and Write computed property)

### 3.5.32 StaticFloatField(TheField as JavaFieldMBS) as single

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a static float field in this class.
**Notes:** (Read and Write computed property)

### 3.5.33 StaticIntField(TheField as JavaFieldMBS) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a static integer field in this class.
**Example:**

```
Var f as FolderItem

f=SpecialFolder.Desktop.Child("ojdbc14.jar")

Var j as new JavaVMMBS(f)

Var c as JavaClassMBS = j.FindClass("oracle/jdbc/driver/OracleTypes")

Var field as JavaFieldMBS

// this are all static integer fields in this class:

field = c.GetStaticField("CURSOR","I")
MsgBox str(c.StaticIntField(field))

field = c.GetStaticField("BLOB","I")
MsgBox str(c.StaticIntField(field))

field = c.GetStaticField("DOUBLE","I")
MsgBox str(c.StaticIntField(field))
```

**Notes:** (Read and Write computed property)

### 3.5.34 StaticLongField(TheField as JavaFieldMBS) as Int64

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a static long field in this class.
**Notes:** (Read and Write computed property)

### 3.5.35 StaticObjectField(TheField as JavaFieldMBS) as JavaObjectMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a static object field in this class.
**Notes:** (Read and Write computed property)

### 3.5.36 StaticShortField(TheField as JavaFieldMBS) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a static short field in this class.
**Notes:** (Read and Write computed property)

# 3.6 class JavaDoubleArrayMBS

## 3.6.1 class JavaDoubleArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for the java double array object.
**Notes:** Subclass of the JavaArrayMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

## 3.6.2 Methods

## 3.6.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

## 3.6.4 Elements as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** All the elements in this array as one memoryblock.
**Notes:** Use memoryblock.DoubleValue(index*4) to access.

## 3.6.5 Values as Double()

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns a Xojo array with values.

## 3.6.6 Properties

## 3.6.7 Region(start as Integer, len as Integer) as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set a region of the array.
**Notes:** Start is the starting index (0 based) and len the number of items to copy.
Throws ArrayIndexOutOfBoundsException if (start + len - 1) does not specify a valid index in the array.

(Read and Write computed property)

### 3.6.8 Value(index as Integer) as Double

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value with given index.
**Notes:** May raise OutOfBoundsException if index is out of range.
(Read and Write computed property)

# 3.7 class JavaFieldMBS

## 3.7.1 class JavaFieldMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for the java field ID.
**Example:**

```
Var f as FolderItem

f=SpecialFolder.Desktop.Child("ojdbc14.jar")

Var j as new JavaVMMBS(f)

Var c as JavaClassMBS = j.FindClass("oracle/jdbc/driver/OracleTypes")

Var field as JavaFieldMBS

// this are all static integer fields in this class:

field = c.GetStaticField("CURSOR","I")
MsgBox str(c.StaticIntField(field))

field = c.GetStaticField("BLOB","I")
MsgBox str(c.StaticIntField(field))

field = c.GetStaticField("DOUBLE","I")
MsgBox str(c.StaticIntField(field))
```

**Notes:** This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
**Blog Entries**

- News from the MBS Xojo Plugins Version 25.3
- Query Java classes in Xojo
- MBS Xojo Plugins, version 19.4pr6
- Upgrading our Java Support for Xojo

## 3.7.2 Methods

## 3.7.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 3.7.4 Properties

### 3.7.5 Handle as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The fieldID of this field.
**Notes:** (Read only property)

### 3.7.6 Name as String

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The name of the field.
**Notes:** This is the name you passed in to GetField or GetStaticField functions.
(Read only property)

### 3.7.7 Signature as String

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The signature of the field.
**Notes:** This is the signature you passed in to GetField or GetStaticField functions.
(Read only property)

# 3.8 class JavaFloatArrayMBS

## 3.8.1 class JavaFloatArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for the java float array object.
**Notes:** Subclass of the JavaArrayMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

## 3.8.2 Methods

## 3.8.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

## 3.8.4 Elements as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** All the elements in this array as one memoryblock.
**Notes:** Use memoryblock.SingleValue(index*4) to access.

## 3.8.5 Values as Single()

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns a Xojo array with values.

## 3.8.6 Properties

## 3.8.7 Region(start as Integer, len as Integer) as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set a region of the array.
**Notes:** Start is the starting index (0 based) and len the number of items to copy.
Throws ArrayIndexOutOfBoundsException if (start + len - 1) does not specify a valid index in the array.

(Read and Write computed property)

### 3.8.8 Value(index as Integer) as Single

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value with given index.
**Notes:** May raise OutOfBoundsException if index is out of range.
(Read and Write computed property)

# 3.9 class JavaHandleNilExceptionMBS

## 3.9.1 class JavaHandleNilExceptionMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The exception which is raised whenever a Java function is operations on an object and there is none.
**Notes:** For example if you call a method on the JavaObjectMBS class which requires the handle value being not zero, the exception raises if the handle value is zero.
Subclass of the RuntimeException class.

## 3.10 class JavaIntArrayMBS

### 3.10.1 class JavaIntArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for the java int array object.
**Notes:** Subclass of the JavaArrayMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

### 3.10.2 Methods

### 3.10.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 3.10.4 Elements as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** All the elements in this array as one memoryblock.
**Notes:** Use memoryblock.Long(index*4) to access.

### 3.10.5 Values as Integer()

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns a Xojo array with values.

### 3.10.6 Properties

### 3.10.7 Region(start as Integer, len as Integer) as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set a region of the array.
**Notes:** Start is the starting index (0 based) and len the number of items to copy.
Throws ArrayIndexOutOfBoundsException if (start + len - 1) does not specify a valid index in the array.

(Read and Write computed property)

### 3.10.8 Value(index as Integer) as Int32

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value with given index.
**Notes:** May raise OutOfBoundsException if index is out of range.
(Read and Write computed property)

## 3.11 class JavaLongArrayMBS

### 3.11.1 class JavaLongArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for the java long array object.
**Notes:** Subclass of the JavaArrayMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

### 3.11.2 Methods

### 3.11.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 3.11.4 Elements as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** All the elements in this array as one memoryblock.
**Notes:** Use memoryblock.Int64DoubleMBS(index*8) to access.

### 3.11.5 Values as Int64()

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns a Xojo array with values.

### 3.11.6 Properties

### 3.11.7 Region(start as Integer, len as Integer) as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set a region of the array.
**Notes:** Start is the starting index (0 based) and len the number of items to copy.
Throws ArrayIndexOutOfBoundsException if (start + len - 1) does not specify a valid index in the array.

(Read and Write computed property)

### 3.11.8 Value(index as Integer) as Int64

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value with given index.
**Notes:** May raise OutOfBoundsException if index is out of range.
(Read and Write computed property)

## 3.12 class JavaMethodMBS

### 3.12.1 class JavaMethodMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for the java method ID.
**Notes:** In terminal you can use "javap -s <classname>" to display the class with the method names and parameters.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
**Blog Entries**

- News from the MBS Xojo Plugins Version 25.3
- Query Java classes in Xojo
- Trying Java 19 in Xojo
- MBS Xojo Plugins, version 19.4pr6
- Upgrading our Java Support for Xojo

### 3.12.2 Methods

### 3.12.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 3.12.4 Properties

### 3.12.5 Handle as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The methodID of this method.
**Notes:** (Read only property)

### 3.12.6 Name as String

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The name of the method.
**Notes:** This is the name you passed in to GetMethod or GetStaticMethod functions.

(Read only property)

### 3.12.7 Signature as String

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The signature of the method.
**Notes:** This is the signature you passed in to GetMethod or GetStaticMethod functions.
(Read only property)

## 3.13 class JavaNotInitializedExceptionMBS

### 3.13.1 class JavaNotInitializedExceptionMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The exception which is raised whenever a Java function is called which requires you to initialize the JVM before calling it.
**Notes:** Subclass of the RuntimeException class.

# 3.14 class JavaObjectArrayMBS

## 3.14.1 class JavaObjectArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for the java object array object.
**Notes:** Subclass of the JavaArrayMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

## 3.14.2 Methods

## 3.14.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

## 3.14.4 Values as JavaObjectMBS()

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns a Xojo array with values.

## 3.14.5 Properties

## 3.14.6 ArrayElement(index as Integer) as JavaObjectMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Deprecated:** This item is deprecated and should no longer be used. You can use Value instead. **Function:** The object at the given index in the array.
**Notes:** May return nil if the array is empty on the given index or if the index is out of bounds or any other error occurs.

Index is 0 based.

Throws
ArrayIndexOutOfBoundsException: if index does not specify a valid index in the array.
ArrayStoreException: if the class of value is not a subclass of the element class of the array.

(Read and Write computed property)

### 3.14.7 Value(index as Integer) as JavaObjectMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value with given index.
**Notes:** May raise OutOfBoundsException if index is out of range.
(Read and Write computed property)

# 3.15 class JavaObjectMBS

## 3.15.1 class JavaObjectMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for the java object.
**Notes:** This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
**Blog Entries**

- Trying Java 19 in Xojo
- MBS Xojo Plugins, version 19.4pr6
- Upgrading our Java Support for Xojo
- MBS Xojo / Real Studio Plugins, version 16.5pr1
- MBS Real Studio Plugins, version 12.1pr1

## 3.15.2 Methods

## 3.15.3 CallBooleanMethod(MethodID as JavaMethodMBS, args as memoryblock) as boolean

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a virtual method with a boolean value.
**Notes:** This family of operations invokes an instance (non-static) method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling GetMethodID.

When this function is used to call private methods and constructors, the method ID must be derived from the real class of obj, not from one of its super classes.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

## 3.15.4 CallByteMethod(MethodID as JavaMethodMBS, args as memoryblock) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a virtual method with a byte value.
**Notes:** This family of operations invokes an instance (non-static) method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling GetMethodID.

When this function is used to call private methods and constructors, the method ID must be derived from the real class of obj, not from one of its super classes.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.5 CallCharMethod(MethodID as JavaMethodMBS, args as memoryblock) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a virtual method with a char value.
**Notes:** This family of operations invokes an instance (non-static) method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling GetMethodID.

When this function is used to call private methods and constructors, the method ID must be derived from the real class of obj, not from one of its super classes.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.6 CallDoubleMethod(MethodID as JavaMethodMBS, args as memoryblock) as Double

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a virtual method with a double value.
**Notes:** This family of operations invokes an instance (non-static) method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling GetMethodID.

When this function is used to call private methods and constructors, the method ID must be derived from the real class of obj, not from one of its super classes.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.7 CallFloatMethod(MethodID as JavaMethodMBS, args as memoryblock) as single

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a virtual method with a float value.
**Notes:** This family of operations invokes an instance (non-static) method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling GetMethodID.

When this function is used to call private methods and constructors, the method ID must be derived from the real class of obj, not from one of its super classes.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.8 CallIntMethod(MethodID as JavaMethodMBS, args as memoryblock) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a virtual method with an integer value.
**Notes:** This family of operations invokes an instance (non-static) method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling GetMethodID.

When this function is used to call private methods and constructors, the method ID must be derived from the real class of obj, not from one of its super classes.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.9 CallLongMethod(MethodID as JavaMethodMBS, args as memoryblock) as Int64

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a virtual method with a long value.
**Notes:** This family of operations invokes an instance (non-static) method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling GetMethodID.

When this function is used to call private methods and constructors, the method ID must be derived from the real class of obj, not from one of its super classes.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.10 CallMethod(MethodID as JavaMethodMBS, args() as Variant) as Variant

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a non virtual method.
**Example:**

```
Var vm As JavaVMMBS // your initialized VM

// 1. Find class
Var jclass As JavaClassMBS =vm.FindClass("stringtest")

if jclass=nil then
msgbox "Can't find stringtest class"
Return
end if

// 2. Call constructor
Var jmethod1 as JavaMethodMBS = jclass.GetMethod("<init>", "()V")

if jmethod1=nil then
msgbox "Can't find constructor"
Return
end if

// 3. Create object
Var param1() As Variant
```

```
Var jobject1 as JavaObjectMBS=jclass.NewObject(jmethod1, param1)

if jobject1=nil then
MsgBox "Constructor failed!?"
Return
end if

// 4. lookup method
Var jmethod3 As JavaMethodMBS = jclass.GetMethod("setValue","(Ljava/lang/String;)V")
if jmethod3=nil then
MsgBox "Failed to find setValue method."
Return
end if

// 5. Call method
Var param3() As Variant
param3.append "Hello World"
Call jobject1.CallMethod(jmethod3,param3)

MsgBox "done"
```

**Notes:** This call invokes an instance (non-static) method on a Java object, according to the specified class and method ID. The methodID argument must be obtained by calling GetMethodID on the class TheClass.

The CallNonvirtualMethod family of routines and the CallMethod family of routines are different. CallMethod routines invoke the method based on the class of the object, while CallNonvirtualMethod routines invoke the method based on the class, designated by the TheClass parameter, from which the method ID is obtained. The method ID must be obtained from the real class of the object or from one of its superclasses.

Programmers place all arguments to the method in an args variant array that immediately follows the methodID argument.

This is generic version, where our plugin translates between native Xojo data types and Java data types. We support conversion of boolean, byte (integer), char (integer), short (integer), int (integer), long (int64), double, float (single) and Java objects. Objects can be JavaObjectMBS or subclasses including JavaStringMBS and the JavaArrayMBS subclasses. For your convenience you can pass in string and we convert to JavaStringMBS for you.

### 3.15.11 CallNonvirtualBooleanMethod(TheClass as JavaClassMBS, MethodID as JavaMethodMBS, args as memoryblock) as boolean

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a non virtual method with a boolean value.

**Notes:** This call invokes an instance (non-static) method on a Java object, according to the specified class and method ID. The methodID argument must be obtained by calling GetMethodID on the class TheClass.

The CallNonvirtualMethod family of routines and the CallMethod family of routines are different. CallMethod routines invoke the method based on the class of the object, while CallNonvirtualMethod routines invoke the method based on the class, designated by the TheClass parameter, from which the method ID is obtained. The method ID must be obtained from the real class of the object or from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.12 CallNonvirtualByteMethod(TheClass as JavaClassMBS, MethodID as JavaMethodMBS, args as memoryblock) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a non virtual method with a byte value.
**Notes:** This call invokes an instance (non-static) method on a Java object, according to the specified class and method ID. The methodID argument must be obtained by calling GetMethodID on the class TheClass.

The CallNonvirtualMethod family of routines and the CallMethod family of routines are different. CallMethod routines invoke the method based on the class of the object, while CallNonvirtualMethod routines invoke the method based on the class, designated by the TheClass parameter, from which the method ID is obtained. The method ID must be obtained from the real class of the object or from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.13 CallNonvirtualCharMethod(TheClass as JavaClassMBS, MethodID as JavaMethodMBS, args as memoryblock) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a non virtual method with a char value.
**Notes:** This call invokes an instance (non-static) method on a Java object, according to the specified class and method ID. The methodID argument must be obtained by calling GetMethodID on the class TheClass.

The CallNonvirtualMethod family of routines and the CallMethod family of routines are different. CallMethod routines invoke the method based on the class of the object, while CallNonvirtualMethod routines invoke the method based on the class, designated by the TheClass parameter, from which the method ID is obtained. The method ID must be obtained from the real class of the object or from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.14 CallNonvirtualDoubleMethod(TheClass as JavaClassMBS, MethodID as JavaMethodMBS, args as memoryblock) as Double

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a non virtual method with a double value.
**Notes:** This call invokes an instance (non-static) method on a Java object, according to the specified class and method ID. The methodID argument must be obtained by calling GetMethodID on the class TheClass.

The CallNonvirtualMethod family of routines and the CallMethod family of routines are different. CallMethod routines invoke the method based on the class of the object, while CallNonvirtualMethod routines invoke the method based on the class, designated by the TheClass parameter, from which the method ID is obtained. The method ID must be obtained from the real class of the object or from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.15 CallNonvirtualFloatMethod(TheClass as JavaClassMBS, MethodID as JavaMethodMBS, args as memoryblock) as single

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a non virtual method with a float value.
**Notes:** This call invokes an instance (non-static) method on a Java object, according to the specified class and method ID. The methodID argument must be obtained by calling GetMethodID on the class TheClass.

The CallNonvirtualMethod family of routines and the CallMethod family of routines are different. CallMethod routines invoke the method based on the class of the object, while CallNonvirtualMethod routines invoke the method based on the class, designated by the TheClass parameter, from which the method ID is obtained. The method ID must be obtained from the real class of the object or from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.16 CallNonvirtualIntMethod(TheClass as JavaClassMBS, MethodID as JavaMethodMI args as memoryblock) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a non virtual method with an integer value.
**Notes:** This call invokes an instance (non-static) method on a Java object, according to the specified class and method ID. The methodID argument must be obtained by calling GetMethodID on the class TheClass.

The CallNonvirtualMethod family of routines and the CallMethod family of routines are different. CallMethod routines invoke the method based on the class of the object, while CallNonvirtualMethod routines invoke the method based on the class, designated by the TheClass parameter, from which the method ID is obtained. The method ID must be obtained from the real class of the object or from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.17 CallNonvirtualLongMethod(TheClass as JavaClassMBS, MethodID as JavaMethodMBS, args as memoryblock) as Int64

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a non virtual method with a long value.
**Notes:** This call invokes an instance (non-static) method on a Java object, according to the specified class and method ID. The methodID argument must be obtained by calling GetMethodID on the class TheClass.

The CallNonvirtualMethod family of routines and the CallMethod family of routines are different. CallMethod routines invoke the method based on the class of the object, while CallNonvirtualMethod routines invoke the method based on the class, designated by the TheClass parameter, from which the method ID is obtained. The method ID must be obtained from the real class of the object or from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.

In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.18 CallNonvirtualMethod(TheClass as JavaClassMBS, MethodID as JavaMethodMBS, args() as Variant) as Variant

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a non virtual method.
**Notes:** This call invokes an instance (non-static) method on a Java object, according to the specified class and method ID. The methodID argument must be obtained by calling GetMethodID on the class TheClass.

The CallNonvirtualMethod family of routines and the CallMethod family of routines are different. CallMethod routines invoke the method based on the class of the object, while CallNonvirtualMethod routines invoke the method based on the class, designated by the TheClass parameter, from which the method ID is obtained. The method ID must be obtained from the real class of the object or from one of its superclasses.

Programmers place all arguments to the method in an args variant array that immediately follows the methodID argument.

This is generic version, where our plugin translates between native Xojo data types and Java data types. We support conversion of boolean, byte (integer), char (integer), short (integer), int (integer), long (int64), double, float (single) and Java objects. Objects can be JavaObjectMBS or subclasses including JavaStringMBS and the JavaArrayMBS subclasses. For your convenience you can pass in string and we convert to JavaStringMBS for you.

### 3.15.19 CallNonvirtualObjectMethod(TheClass as JavaClassMBS, MethodID as JavaMethodMBS, args as memoryblock) as JavaObjectMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a non virtual method with an object value.
**Notes:** This call invokes an instance (non-static) method on a Java object, according to the specified class and method ID. The methodID argument must be obtained by calling GetMethodID on the class TheClass.

The CallNonvirtualMethod family of routines and the CallMethod family of routines are different. CallMethod routines invoke the method based on the class of the object, while CallNonvirtualMethod routines invoke the method based on the class, designated by the TheClass parameter, from which the method ID is obtained. The method ID must be obtained from the real class of the object or from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the

methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.20 CallNonvirtualShortMethod(TheClass as JavaClassMBS, MethodID as JavaMethodMBS, args as memoryblock) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a non virtual method with a short value.
**Notes:** This call invokes an instance (non-static) method on a Java object, according to the specified class and method ID. The methodID argument must be obtained by calling GetMethodID on the class TheClass.

The CallNonvirtualMethod family of routines and the CallMethod family of routines are different. CallMethod routines invoke the method based on the class of the object, while CallNonvirtualMethod routines invoke the method based on the class, designated by the TheClass parameter, from which the method ID is obtained. The method ID must be obtained from the real class of the object or from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.21 CallNonvirtualVoidMethod(TheClass as JavaClassMBS, MethodID as JavaMethodMBS, args as memoryblock)

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a non virtual method with no return value.
**Notes:** This call invokes an instance (non-static) method on a Java object, according to the specified class and method ID. The methodID argument must be obtained by calling GetMethodID on the class TheClass.

The CallNonvirtualMethod family of routines and the CallMethod family of routines are different. CallMethod routines invoke the method based on the class of the object, while CallNonvirtualMethod routines invoke the method based on the class, designated by the TheClass parameter, from which the method ID is obtained. The method ID must be obtained from the real class of the object or from one of its superclasses.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.22 CallObjectMethod(MethodID as JavaMethodMBS, args as memoryblock) as JavaObjectMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a virtual method with an object value.
**Notes:** This family of operations invokes an instance (non-static) method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling GetMethodID.

When this function is used to call private methods and constructors, the method ID must be derived from the real class of obj, not from one of its super classes.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.23 CallShortMethod(MethodID as JavaMethodMBS, args as memoryblock) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a virtual method with a short value.
**Notes:** This family of operations invokes an instance (non-static) method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling GetMethodID.

When this function is used to call private methods and constructors, the method ID must be derived from the real class of obj, not from one of its super classes.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.24 CallVoidMethod(MethodID as JavaMethodMBS, args as memoryblock)

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Calls a virtual method with no return value.
**Notes:** This family of operations invokes an instance (non-static) method on a Java object, according to the specified method ID. The methodID argument must be obtained by calling GetMethodID.

When this function is used to call private methods and constructors, the method ID must be derived from the real class of obj, not from one of its super classes.

Programmers place all arguments to the method in an args memoryblock that immediately follows the methodID argument.
In the memoryblock you need to use 8 bytes per argument and align them correctly. (alignment depends on platform)

### 3.15.25 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 3.15.26 GetDirectBufferAddress(directbuffer as JavaObjectMBS) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns the address of the memory from a directbuffer object.
**Notes:** Returns 0 on any error.

### 3.15.27 GetDirectBufferCapacity(directbuffer as JavaObjectMBS) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns the size of the memory from a directbuffer object.
**Notes:** Returns 0 on any error.

### 3.15.28 IsInstanceOf(TheClass as JavaClassMBS) as boolean

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Tests whether an object is an instance of a class.
**Notes:** Returns true if obj can be cast to TheClass; otherwise, returns false. A nil object can be cast to any class.

### 3.15.29 IsSameObject(obj as JavaObjectMBS) as boolean

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Tests whether two references refer to the same Java object.
**Notes:** Returns true if ref1 and ref2 refer to the same Java object, or are both nil; otherwise, returns false.

False on any error.

### 3.15.30 ObjectClass as JavaClassMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns the class of an object.
**Notes:** Returns nil on any error.

### 3.15.31 Properties

### 3.15.32 ClassName as String

Plugin Version: 16.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Queries class name of object.
**Notes:** (Read only property)

### 3.15.33 Database as Variant

Plugin Version: 12.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The reference to the database object.
**Notes:** The variant is a JavaDatabaseMBS.
Do not assign new values, please.
This is set for all database classes, so the database isn't release from memory before you finished using this object.
(Read and Write property)

### 3.15.34 Handle as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The handle to the jobject.
**Notes:** (Read and Write property)

### 3.15.35 Lasterror as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

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

### 3.15.36 Tag as Variant

Plugin Version: 12.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** This is a property you can use for whatever you like.
**Notes:** The property value is stored as long as the RB object lives. The Java object may live longer.
(Read and Write property)

### 3.15.37 VM as JavaVMMBS

Plugin Version: 12.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The reference to the virtual machine.
**Notes:** Please do not assign new value unless you know what you do.
(Read and Write property)

### 3.15.38 BooleanField(TheField as JavaFieldMBS) as boolean

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a boolean field in this class.
**Notes:** Do not use for static fields!
(Read and Write computed property)

### 3.15.39 ByteField(TheField as JavaFieldMBS) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a byte field in this class.
**Notes:** Do not use for static fields!
(Read and Write computed property)

### 3.15.40 CharField(TheField as JavaFieldMBS) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a char field in this class.
**Notes:** Do not use for static fields!
(Read and Write computed property)

### 3.15.41 DoubleField(TheField as JavaFieldMBS) as Double

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a double field in this class.
**Notes:** Do not use for static fields!
(Read and Write computed property)

### 3.15.42 Field(TheField as JavaFieldMBS) as Variant

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a field in this class.
**Notes:** This is generic version, where our plugin translates between native Xojo data types and Java data types. We support conversion of boolean, byte (integer), char (integer), short (integer), int (integer), long (int64), double, float (single) and Java objects. Objects can be JavaObjectMBS or subclasses including JavaStringMBS and the JavaArrayMBS subclasses. For your convenience you can pass in string and we convert to JavaStringMBS for you.

Do not use for static fields!
(Read and Write computed property)

### 3.15.43 FloatField(TheField as JavaFieldMBS) as single

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a single field in this class.
**Notes:** Do not use for static fields!
(Read and Write computed property)

### 3.15.44 IntField(TheField as JavaFieldMBS) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for an integer field in this class.
**Notes:** Do not use for static fields!
(Read and Write computed property)

### 3.15.45 LongField(TheField as JavaFieldMBS) as Int64

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a long field in this class.
**Notes:** Do not use for static fields!
(Read and Write computed property)

### 3.15.46 ObjectField(TheField as JavaFieldMBS) as JavaObjectMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for an object field in this class.
**Notes:** Do not use for static fields!
(Read and Write computed property)

### 3.15.47 ShortField(TheField as JavaFieldMBS) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value for a short field in this class.
**Notes:** Do not use for static fields!
(Read and Write computed property)

# 3.16 class JavaShortArrayMBS

## 3.16.1 class JavaShortArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for the java short array object.
**Notes:** Subclass of the JavaArrayMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

## 3.16.2 Methods

## 3.16.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

## 3.16.4 Elements as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** All the elements in this array as one memoryblock.
**Notes:** Use memoryblock.short(index*2) to access.

## 3.16.5 Values as Int16()

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns a Xojo array with values.

## 3.16.6 Properties

## 3.16.7 Region(start as Integer, len as Integer) as memoryblock

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set a region of the array.
**Notes:** Start is the starting index (0 based) and len the number of items to copy.
Throws ArrayIndexOutOfBoundsException if (start + len - 1) does not specify a valid index in the array.

(Read and Write computed property)

### 3.16.8 Value(index as Integer) as Int16

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get or set the value with given index.
**Notes:** May raise OutOfBoundsException if index is out of range.
(Read and Write computed property)

# 3.17 class JavaStringMBS

## 3.17.1 class JavaStringMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for the java string class.
**Notes:** Subclass of the JavaObjectMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
**Blog Entries**

- News from the MBS Xojo Plugins Version 25.3
- Query Java classes in Xojo
- Trying Java 19 in Xojo
- MBS Xojo Plugins, version 19.4pr6
- MBS Xojo Plugins, version 19.2pr5

## 3.17.2 Methods

## 3.17.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

## 3.17.4 CopyString as string

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Deprecated:** This item is deprecated and should no longer be used. You can use StringValue instead.
**Function:** Copies the content of the string into a Xojo string.
**Notes:** Returns "" on any error. The string returned is marked as being Unicode (16bit).
Deprecated in favor of StringValue property and auto conversion with Operator_Convert.
See also:

- 3.17.5 CopyString(start as Integer, len as Integer) as string 93

## 3.17.5 CopyString(start as Integer, len as Integer) as string

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Copies the content of the string into a Xojo string.
**Example:**

```
Var s as string
Var js as JavaStringMBS // your java string

s=js.CopyString(0,1) // copies first character
s=js.CopyString(3,6) // copies six charcters from starting at the forth
```

**Notes:** Returns "" on any error. The string returned is marked as being Unicode (16bit).
For the first character to be the start use start=0.
For the first character to be the end use len=1 and start=0.
May crash on bad values for start and len.
See also:

- 3.17.4 CopyString as string 93

### 3.17.6 CopyStringUTF as string

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Deprecated:** This item is deprecated and should no longer be used. You can use StringValue instead.
**Function:** Copies the content of the string into a Xojo string.
**Notes:** Returns "" on any error. The string returned is marked as being UTF8.
Deprecated in favor of StringValue property and auto conversion with Operator_Convert.
See also:

- 3.17.7 CopyStringUTF(start as Integer, len as Integer) as string 94

### 3.17.7 CopyStringUTF(start as Integer, len as Integer) as string

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Copies the content of the string into a Xojo string.
**Notes:** Returns "" on any error. The string returned is marked as being UTF8.
Start is 0 based.
May crash on bad values for start and len.
See also:

- 3.17.6 CopyStringUTF as string 94

### 3.17.8 Operator_Convert as string

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Converts to a string automatically.
**Notes:** Do not call it, just assign JavaStringMBS to a string an Xojo calls this internally.

### 3.17.9 UTFLength as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Deprecated:** This item is deprecated and should no longer be used. You can use StringValue instead.
**Function:** The length of the string in bytes encoded as UTF8.
**Notes:** Returns 0 on any error.

### 3.17.10 Properties

### 3.17.11 Length as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The length of this string in unicode characters.
**Notes:** Returns 0 on any error.
(Read only property)

### 3.17.12 StringValue as String

Plugin Version: 19.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The string value as property.
**Notes:** Uses UTF-8 internally.
(Read only property)

## 3.18 class JavaThrowableMBS

### 3.18.1 class JavaThrowableMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The wrapper class for the java throwable object.
**Notes:** Subclass of the JavaObjectMBS class.
This is a subclass of an abstract class. You can't create an instance, but you can get one from various plugin functions.

# 3.19 class JavaVMMBS

## 3.19.1 class JavaVMMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The class for a java virtual machine.
**Example:**

```
Var vm as JavaVMMBS // global

const JNI_VERSION_1_1 = &h00010001
const JNI_VERSION_1_2 = &h00010002
const JNI_VERSION_1_4 = &h00010004

if TargetLinux then
// change path for your linux PC!
JavaVMMBS.SetLibraryPath("/home/cs/jre1.6.0_05/lib/i386/client/libjvm.so")
end if

Var options(-1) as string
Var f as FolderItem=GetFolderItem("test.jar")

vm=new JavaVMMBS(JNI_VERSION_1_4, options, f, false)

if vm.Handle = 0 then
MsgBox "Can't create Java VM"
else
MsgBox "Java Initialized."
end if
```

**Notes:** Add Linux support plugin version 8.7.

Releasing the java vm (by releasing all java objects), and reinitializing can fail.

Please make sure this Java VM object stays alive until you are done with all your java stuff. So all the java objects go away and this vm object is destroyed on the end. Because if some java code is still running like an background java thread, quitting the VM can lead into crashes.

While the plugin supports to have several instances, it seems like JNI does not support that.

On Windows, we look into Local Machine\SOFTWARE\JavaSoft\Java Runtime Environment in registry to find the newest entry and the path to the jvm.dll file. Please note that on Windows you can install 32-bit and 64-bit version of Java and you need the matching bit number to FileMaker application.

If you get error 126 on Windows, maybe MSVCR100.DLL is missing. So install Visual Studio 2010 runtime libraries. You can download them from Microsoft website. Do not download them from other websites spreading malware!

**Blog Entries**

- News from the MBS Xojo Plugins Version 25.3
- Query Java classes in Xojo
- MonkeyBread Software Releases the MBS Xojo Plugins in version 25.3
- MBS Xojo Plugins, version 25.3pr1
- MBS Xojo Plugins, version 23.2pr1
- Trying Java 19 in Xojo
- News from the MBS Xojo Plugins Version 20.3
- MonkeyBread Software Releases the MBS Xojo Plugins in version 20.3
- Java with MBS Plugin
- Encrypted Access database in Xojo

**Xojo Developer Magazine**

- 23.5, page 9: News

### 3.19.2 Methods

### 3.19.3 Constructor(path as folderitem)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Initializes the Java virtual machine.
**Notes:** This is a convenience function which initializes the java with version=JNI_VERSION_1_4 and no options except the specified path.
The path can be a folderitem pointing to a jar file or a folder with class files.

This method raises UnsupportedOperationException with error about missing CreateJavaVM function if there is no Java found. Either because it is not installed or the bit number (32 vs. 64) does not match.

Since version 16.5 the plugin will no longer raise exception if an existing JavaVM was found. In that case we use that JavaVM and return normally. Lasterror will be set to -5 which indicates this. In that case your options and paths are not passed to VM.
See also:

### 3.19.4 Constructor(path as string)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Initializes the Java virtual machine.
**Notes:** This is a convenience function which initializes the java engine with version=JNI_VERSION_1_4 and no options except the specified path.
The path can be a path pointing to a jar file or a folder with class files. If you use more than one path, you need to separate them with ";". Seems like on Mac OS X and Linux the separator is ":".

This method raises UnsupportedOperationException with error about missing CreateJavaVM function if there is no Java found. Either because it is not installed or the bit number (32 vs. 64) does not match.

Since version 16.5 the plugin will no longer raise exception if an existing JavaVM was found. In that case we use that JavaVM and return normally. Lasterror will be set to -5 which indicates this. In that case your options and paths are not passed to VM.
See also:



### 3.19.5 Constructor(version as Integer, options() as string, ignoreUnrecognizedOptions as boolean)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Initializes the java virtual machine.
**Notes:** Only one VM can run at a time with this plugin.
You can specify whatever command line options you need in the options array.
Version must be one of the JNI_VERSION_1_x constants.

A note for Mac OS X and Java versions:

To specify the current preferred JDK in a family of JVM's, say the 1.5.x family, applications should set the environment variable JAVA_JVM_VERSION to 1.5, and then pass JNI_VERSION_1_4 into JNI_CreateJavaVM as the vm_args.version. To get a specific Java 1.5 JVM, say Java 1.5.0, set the environment variable JAVA_JVM_VERSION to 1.5.0. For Java 1.6 it will be the same in that applications will need to set the environment variable JAVA_JVM_VERSION to 1.6 to specify the current preferred 1.6 Java VM, and to get a specific Java 1.6 JVM, say Java 1.6.1, set the environment variable JAVA_JVM_VERSION to 1.6.1.

To make this sample bring up the current preferred 1.5 JVM, set the environment variable JAVA_JVM_VERSION to 1.5 before calling JNI_CreateJavaVM as shown below. Applications must currently check for availability of JDK 1.5 before requesting it. If your application requires JDK 1.5 and it is not found, it is your responsibility to report an error to the user. To verify if a JVM is installed, check to see if the symlink, or directory exists for the JVM in /System/Library/Frameworks/JavaVM.framework/Versions/ before setting the environment variable JAVA_JVM_VERSION.

If the environment variable JAVA_JVM_VERSION is not set, and JNI_VERSION_1_4 is passed into JNI_CreateJavaVM as the vm_args.version, JNI_CreateJavaVM will return the current preferred JDK. Java 1.4.2 is the preferred JDK as of the release of this sample and the release of Mac OS X 10.4.

Useful option strings:

| | |
|---|---|
| "-verbose:jni" | show debug output on the console |
| "-Xms256M" | initial memory |
| "-Xmx512M" | maximum memory |

This method raises UnsupportedOperationException with error about missing CreateJavaVM function if there is no Java found. Either because it is not installed or the bit number (32 vs. 64) does not match.

Since version 16.5 the plugin will no longer raise exception if an existing JavaVM was found. In that case we use that JavaVM and return normally. Lasterror will be set to -5 which indicates this. In that case your options and paths are not passed to VM.

If you get error 126 on Windows, maybe MSVCR100.DLL is missing. So install Visual Studio 2010 runtime libraries. You can download them from Microsoft website. Do not download them from other websites spreading malware!
See also:


- 3.19.3 Constructor(path as folderitem) 98
- 3.19.4 Constructor(path as string) 99
- 3.19.6 Constructor(version as Integer, options() as string, path as folderitem, ignoreUnrecognizedOp-

### 3.19.6 Constructor(version as Integer, options() as string, path as folderitem, ignoreUnrecognizedOptions as boolean)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Initializes the Java virtual machine.
**Example:**

```
Var JarPath as folderitem = specialfolder.desktop.child("test.jar")
Var options() as string

options.append "-Xms256M"
options.append "-Xmx512M"

Var v as new JavaVMMBS( JavaVMMBS.JNI_VERSION_1_4, options, JarPath, false)
```

**Notes:** This is a convenience function which initializes the java engine with adding the given path to the options.

Only one VM can run at a time with this plugin.
You can specify whatever command line options you need in the options array.
Version must be one of the JNI_VERSION_1_x constants.
The path can be a folderitem pointing to a jar file or a folder with class files.

Useful option strings:

| | |
|---|---|
| "-verbose:jni" | show debug output on the console |
| "-Xms256M" | initial memory |
| "-Xmx512M" | maximum memory |

This method raises UnsupportedOperationException with error about missing CreateJavaVM function if there is no Java found. Either because it is not installed or the bit number (32 vs. 64) does not match.

Since version 16.5 the plugin will no longer raise exception if an existing JavaVM was found. In that case we use that JavaVM and return normally. Lasterror will be set to -5 which indicates this. In that case your options and paths are not passed to VM.
See also:

- 3.19.3 Constructor(path as folderitem) 98

### 3.19.7 Constructor(version as Integer, options() as string, path as string, ignoreUnrecognizedOptions as boolean)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Initializes the java virtual machine.
**Notes:** This is a convenience function which initializes the java engine with adding the given path to the options.

Only one VM can run at a time with this plugin.
You can specify whatever command line options you need in the options array.
Version must be one of the JNI_VERSION_1_x constants.
The path can be a path pointing to a jar file or a folder with class files. If you use more than one path, you need to separate them with ";". Seems like on Mac OS X and Linux the separator is ":".

Useful option strings:

| | |
|---|---|
| "-verbose:jni" | show debug output on the console |
| "-Xms256M" | initial memory |
| "-Xmx512M" | maximum memory |

This method raises UnsupportedOperationException with error about missing CreateJavaVM function if there is no Java found. Either because it is not installed or the bit number (32 vs. 64) does not match.

Since version 16.5 the plugin will no longer raise exception if an existing JavaVM was found. In that case we use that JavaVM and return normally. Lasterror will be set to -5 which indicates this. In that case your options and paths are not passed to VM.
See also:

### 3.19.8 DefineClass(name as string, Data as MemoryBlock) as JavaClassMBS

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Loads a class from a buffer of raw class data.
**Notes:** Returns nil in case of error or valid class object on success.
You may want to load a class file and pass it here as String or MemoryBlock.
To read jar file, you can use our archive classes to expand zip archives.
See also:

- 3.19.9 DefineClass(name as string, Data as String) as JavaClassMBS 103

### 3.19.9 DefineClass(name as string, Data as String) as JavaClassMBS

Plugin Version: 20.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Loads a class from a buffer of raw class data.
**Notes:** Returns nil in case of error or valid class object on success.
You may want to load a class file and pass it here as String or MemoryBlock.
To read jar file, you can use our archive classes to expand zip archives.
See also:

- 3.19.8 DefineClass(name as string, Data as MemoryBlock) as JavaClassMBS 103

### 3.19.10 FindClass(name as string) as JavaClassMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Searches a class based on the name.
**Example:**

```
Var e as JavaVMMBS // global
Var c as JavaClassMBS
c=e.FindClass("java/lang/String")
```

**Notes:** This function loads a locally defined class. It searches the directories and zip files specified by the CLASSPATH environment variable for the class with the specified name.

name: a fully qualified class name (that is, a package name, delimited by "/", followed by the class name). If the name begins with " [ " (the array signature character), it returns an array class.

Returns nil on any error.

If your class is not found, it may be possible that it can't be loaded as the jar archive has dependencies to other jar archives.

### 3.19.11 FreeCurrentThread

Plugin Version: 8.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Releases the thread in Java.
**Example:**

```
Var w as JavaVMMBS // your vm object
w.FreeCurrentThread
```

**Notes:** The plugin is written to detect if you use it in a thread. But when the thread ends you need to deregister it with the Java runtime.

### 3.19.12 FromReflectedField(field as JavaObjectMBS) as JavaFieldMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Searchs the field which matches the reflected field object.

### 3.19.13 FromReflectedMethod(method as JavaObjectMBS) as JavaMethodMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Searchs the method which matches the reflected method object.

### 3.19.14 IsAssignableFrom(TheSubClass as JavaClassMBS, TheSuperClass as JavaClassMBS) as boolean

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** True if the class is assignable from the other class.
**Notes:** That means that the sub class is somewhere down the class tree from the super class.

### 3.19.15 LoadedClasses as JavaClassMBS()

Plugin Version: 25.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Queries the list of classes.
**Example:**

```
// show all class names

var vm as JavaVMMBS // your VM
Var arg As New MemoryBlock(8)
Var LoadedClassNames() As String
Var LoadedClasses() As JavaClassMBS = vm.LoadedClasses
For Each c As JavaClassMBS In LoadedClasses
Var jc As JavaClassMBS = c.ObjectClass
Var jm As JavaMethodMBS = jc.GetMethod("getName", "()Ljava/lang/String;")
Var jo As Variant = c.CallObjectMethod(jm, arg)
Var js As JavaStringMBS = jo
Var name As String = js.StringValue
LoadedClassNames.Add name
Next
MessageBox String.FromArray(LoadedClassNames, EndOfLine)
```

**Notes:** Does not include items from super class.

### 3.19.16 MonitorEnter(obj as JavaObjectMBS) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Enters the monitor associated with the underlying Java object referred to by obj.
**Notes:** Returns zero on success; otherwise, returns a negative value on failure.

Each Java objects has a monitor associated with it. If the current thread already owns the monitor associated with ref, it increments a counter in the monitor indicating the number of times this thread has entered the monitor. If the monitor associated with ref is not owned by any thread, the current thread becomes the owner of the monitor, setting the entry count of this monitor to 1. If another thread already owns the monitor associated with ref, the current thread waits until the monitor is released, then tries again to gain ownership.

### 3.19.17 MonitorExit(obj as JavaObjectMBS) as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Releases the monitor.
**Notes:** The current thread must be the owner of the monitor associated with the underlying Java object referred to by ref. The thread decrements the counter indicating the number of times it has entered this monitor. If as a result the value of the counter becomes zero, the current thread releases the montior.

Returns zero on success; otherwise, returns a negative value on failure.

### 3.19.18 NewBooleanArray(ref as JavaObjectMBS) as JavaBooleanArrayMBS

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array object based on the given java object.
**Notes:** This function is a convenience function to convert a java object array reference to a java array object in Xojo. It can crash if the java object used is not the array of the requested type.
See also:

- 3.19.19 NewBooleanArray(size as Integer) as JavaBooleanArrayMBS 106
- 3.19.20 NewBooleanArray(values() as Boolean) as JavaBooleanArrayMBS 106

### 3.19.19 NewBooleanArray(size as Integer) as JavaBooleanArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array for booleans with the given size.
**Notes:** Returns nil on any error.
See also:

- 3.19.18 NewBooleanArray(ref as JavaObjectMBS) as JavaBooleanArrayMBS 106
- 3.19.20 NewBooleanArray(values() as Boolean) as JavaBooleanArrayMBS 106

### 3.19.20 NewBooleanArray(values() as Boolean) as JavaBooleanArrayMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates new boolean array with given values.
See also:

- 3.19.18 NewBooleanArray(ref as JavaObjectMBS) as JavaBooleanArrayMBS 106
- 3.19.19 NewBooleanArray(size as Integer) as JavaBooleanArrayMBS 106

### 3.19.21 NewByteArray(ref as JavaObjectMBS) as JavaByteArrayMBS

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array object based on the given java object.
**Notes:** This function is a convenience function to convert a java object array reference to a java array object in Xojo. It can crash if the java object used is not the array of the requested type.
See also:

### 3.19.22 NewByteArray(size as Integer) as JavaByteArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array for bytes with the given size.
**Notes:** Returns nil on any error.
See also:



### 3.19.23 NewByteArray(values() as UInt8) as JavaBooleanArrayMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates new byte array with given values.
See also:



### 3.19.24 NewCharArray(ref as JavaObjectMBS) as JavaCharArrayMBS

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array object based on the given java object.
**Notes:** This function is a convenience function to convert a java object array reference to a java array object in Xojo. It can crash if the java object used is not the array of the requested type.
See also:



### 3.19.25 NewCharArray(size as Integer) as JavaCharArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array for chars (16bit) with the given size.
**Notes:** Returns nil on any error.
See also:

### 3.19.26 NewCharArray(values() as UInt16) as JavaCharArrayMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates new char array with given values.
See also:



### 3.19.27 NewDirectByteBuffer(address as Integer, size as Integer) as JavaObjectMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new DirectBuffer object based on the values in address and size.
See also:



### 3.19.28 NewDirectByteBuffer(mem as memoryblock) as JavaObjectMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new DirectBuffer object based on a memoryblock.
**Notes:** Keep the memoryblock until this object is destroyed.
See also:



### 3.19.29 NewDoubleArray(ref as JavaObjectMBS) as JavaDoubleArrayMBS

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array object based on the given java object.
**Notes:** This function is a convenience function to convert a java object array reference to a java array object in Xojo. It can crash if the java object used is not the array of the requested type.
See also:

### 3.19.30 NewDoubleArray(size as Integer) as JavaDoubleArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array for doubles with the given size.
**Notes:** Returns nil on any error.
See also:

- 3.19.29 NewDoubleArray(ref as JavaObjectMBS) as JavaDoubleArrayMBS 108
- 3.19.31 NewDoubleArray(values() as Double) as JavaDoubleArrayMBS 109

### 3.19.31 NewDoubleArray(values() as Double) as JavaDoubleArrayMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates new double array with given values.
See also:

- 3.19.29 NewDoubleArray(ref as JavaObjectMBS) as JavaDoubleArrayMBS 108
- 3.19.30 NewDoubleArray(size as Integer) as JavaDoubleArrayMBS 109

### 3.19.32 NewFloatArray(ref as JavaObjectMBS) as JavaFloatArrayMBS

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array object based on the given java object.
**Notes:** This function is a convenience function to convert a java object array reference to a java array object in Xojo. It can crash if the java object used is not the array of the requested type.
See also:

- 3.19.33 NewFloatArray(size as Integer) as JavaFloatArrayMBS 109
- 3.19.34 NewFloatArray(values() as Single) as JavaFloatArrayMBS 110

### 3.19.33 NewFloatArray(size as Integer) as JavaFloatArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array for singles with the given size.
**Notes:** Returns nil on any error.
See also:

- 3.19.32 NewFloatArray(ref as JavaObjectMBS) as JavaFloatArrayMBS 109
- 3.19.34 NewFloatArray(values() as Single) as JavaFloatArrayMBS 110

### 3.19.34 NewFloatArray(values() as Single) as JavaFloatArrayMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates new float array with given values.
See also:

- 3.19.32 NewFloatArray(ref as JavaObjectMBS) as JavaFloatArrayMBS 109
- 3.19.33 NewFloatArray(size as Integer) as JavaFloatArrayMBS 109

### 3.19.35 NewIntArray(ref as JavaObjectMBS) as JavaIntArrayMBS

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array object based on the given java object.
**Notes:** This function is a convenience function to convert a java object array reference to a java array object in Xojo. It can crash if the java object used is not the array of the requested type.
See also:

- 3.19.36 NewIntArray(size as Integer) as JavaIntArrayMBS 110
- 3.19.37 NewIntArray(values() as Int32) as JavaIntArrayMBS 110

### 3.19.36 NewIntArray(size as Integer) as JavaIntArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array for integers with the given size.
**Notes:** Returns nil on any error.
See also:

- 3.19.35 NewIntArray(ref as JavaObjectMBS) as JavaIntArrayMBS 110
- 3.19.37 NewIntArray(values() as Int32) as JavaIntArrayMBS 110

### 3.19.37 NewIntArray(values() as Int32) as JavaIntArrayMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates new Int32 array with given values.
See also:

- 3.19.35 NewIntArray(ref as JavaObjectMBS) as JavaIntArrayMBS 110
- 3.19.36 NewIntArray(size as Integer) as JavaIntArrayMBS 110

### 3.19.38 NewLongArray(ref as JavaObjectMBS) as JavaLongArrayMBS

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array object based on the given java object.
**Notes:** This function is a convenience function to convert a java object array reference to a java array object in Xojo. It can crash if the java object used is not the array of the requested type.
See also:

- 3.19.39 NewLongArray(size as Integer) as JavaLongArrayMBS 111
- 3.19.40 NewLongArray(values() as Int64) as JavaLongArrayMBS 111

### 3.19.39 NewLongArray(size as Integer) as JavaLongArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array for 64 bit integers with the given size.
**Notes:** Returns nil on any error.
See also:

- 3.19.38 NewLongArray(ref as JavaObjectMBS) as JavaLongArrayMBS 111
- 3.19.40 NewLongArray(values() as Int64) as JavaLongArrayMBS 111

### 3.19.40 NewLongArray(values() as Int64) as JavaLongArrayMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates new Int64 array with given values.
See also:

- 3.19.38 NewLongArray(ref as JavaObjectMBS) as JavaLongArrayMBS 111
- 3.19.39 NewLongArray(size as Integer) as JavaLongArrayMBS 111

### 3.19.41 NewObjectArray(ref as JavaObjectMBS) as JavaObjectArrayMBS

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array object based on the given java object.
**Notes:** This function is a convenience function to convert a java object array reference to a java array object in Xojo. It can crash if the java object used is not the array of the requested type.
See also:

- 3.19.42 NewObjectArray(size as Integer, TheClass as JavaClassMBS, InitialValue as JavaObjectMBS = nil) as JavaObjectArrayMBS 112
- 3.19.43 NewObjectArray(values() as JavaObjectMBS) as JavaObjectArrayMBS 112

### 3.19.42 NewObjectArray(size as Integer, TheClass as JavaClassMBS, InitialValue as JavaObjectMBS = nil) as JavaObjectArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Constructs a new array holding objects in class elementClass.
**Notes:** All elements are initially set to initialElement.
Returns nil on any error.
See also:

- 3.19.41 NewObjectArray(ref as JavaObjectMBS) as JavaObjectArrayMBS 111
- 3.19.43 NewObjectArray(values() as JavaObjectMBS) as JavaObjectArrayMBS 112

### 3.19.43 NewObjectArray(values() as JavaObjectMBS) as JavaObjectArrayMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new Java object array with given size.
See also:

- 3.19.41 NewObjectArray(ref as JavaObjectMBS) as JavaObjectArrayMBS 111
- 3.19.42 NewObjectArray(size as Integer, TheClass as JavaClassMBS, InitialValue as JavaObjectMBS = nil) as JavaObjectArrayMBS 112

### 3.19.44 NewShortArray(ref as JavaObjectMBS) as JavaShortArrayMBS

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array object based on the given java object.
**Notes:** This function is a convenience function to convert a java object array reference to a java array object in Xojo. It can crash if the java object used is not the array of the requested type.
See also:

- 3.19.45 NewShortArray(size as Integer) as JavaShortArrayMBS 112
- 3.19.46 NewShortArray(values() as Int16) as JavaShortArrayMBS 113

### 3.19.45 NewShortArray(size as Integer) as JavaShortArrayMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new array for shorts with the given size.
**Notes:** Returns nil on any error.
See also:

### 3.19.46 NewShortArray(values() as Int16) as JavaShortArrayMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates new short array with given values.
See also:



### 3.19.47 NewStringArray(size as integer, InitialValue as JavaStringMBS = nil) as JavaObjectArrayMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new Java string array with given size.
See also:



### 3.19.48 NewStringArray(values() as String) as JavaObjectArrayMBS

Plugin Version: 19.2, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates new string array with given values.
See also:



### 3.19.49 NewStringUnicode(s as string) as JavaStringMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new string.
**Notes:** Preferres an Unicode encoded string.

### 3.19.50 NewStringUTF8(s as string) as JavaStringMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new string.
**Example:**

```
// init Java
Var options() as string
Var javaVm as new JavaVMMBS(JavaVMMBS.JNI_VERSION_1_4, options, true)

// Get system class
Var system As JavaClassMBS = javaVm.FindClass("java/lang/System")

If system <>Nil Then

// query method
Var transfomerGetPropertyId as JavaMethodMBS = system.GetStaticMethod("getProperty", "(Ljava/lang/String;)Ljava/lan
If transfomerGetPropertyId <>Nil Then

// make parameters
Var keyString As JavaStringMBS = javaVm.NewStringUTF8("os.version")
Var m as New MemoryBlock(8)
m.Int64Value(0)=keyString.Handle

// run it
Var r as JavaObjectMBS = system.CallStaticObjectMethod(transfomerGetPropertyId, m)

if r<>Nil then
// show result
Var s as JavaStringMBS = JavaStringMBS(r)
MsgBox s.CopyStringUTF
end if

End If
End If
```

**Notes:** Preferres an UTF8 encoded string.

### 3.19.51 Runtime as JavaRuntimeMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns the runtime object for the current VM so you can query the memory statistics.

### 3.19.52 SetLibraryPath(path as folderitem)

Plugin Version: 8.7, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Defines the path where to find the java library.
**Notes:** Must be called before you use the constructor.
If this is empty on Windows, we will look into registry.
If this is empty on MacOS, we ask the Java.framework for the functions.
See also:

- 3.19.53 SetLibraryPath(path as string) 115

### 3.19.53 SetLibraryPath(path as string)

Plugin Version: 8.7, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Defines the path where to find the java library.
**Example:**

```
if TargetLinux then
// change path for your linux PC!
JavaVMMBS.SetLibraryPath("/home/cs/jre1.6.0_05/lib/i386/client/libjvm.so")
end if
```

**Notes:** Must be called before you use the constructor.
If this is empty on Windows, we will look into registry.
If this is empty on MacOS, we ask the Java.framework for the functions.
See also:

- 3.19.52 SetLibraryPath(path as folderitem) 115

### 3.19.54 ToReflectedField(TheClass as JavaClassMBS, fieldID as JavaFieldMBS, isStatic as boolean) as JavaObjectMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new reflected field object for the given field.

### 3.19.55 ToReflectedMethod(TheClass as JavaClassMBS, methodID as JavaMethodMBS, isStatic as boolean) as JavaObjectMBS

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new reflected method object for the given method.

### 3.19.56 Version as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns the version of the java engine.
**Notes:** Currently the plugin always uses version 1.4.

Returns the major version number in the higher 16 bits and the minor version number in the lower 16 bits.

In JDK1.1, GetVersion() returns 0x00010001.

### 3.19.57 Properties

### 3.19.58 Handle as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The handle of the JavaVM.
**Notes:** (Read only property)

### 3.19.59 Lasterror as Integer

Plugin Version: 4.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

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

### 3.19.60 Constants

Constants

| Constant | Value | Description |
|---|---|---|
| JNI_VERSION_1_1 | &h00010001 | One of the constants to specify the JNI version in the constructor. |
| JNI_VERSION_1_2 | &h00010002 | One of the constants to specify the JNI version in the constructor. |
| JNI_VERSION_1_4 | &h00010004 | One of the constants to specify the JNI version in the constructor. |
| JNI_VERSION_1_6 | &h00010006 | One of the constants to specify the JNI version in the constructor. |

# Chapter 4

# Java Database

## 4.1 class JavaBlobMBS

### 4.1.1 class JavaBlobMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The java class for a binary large object.
**Notes:** Subclass of the JavaObjectMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

### 4.1.2 Methods

### 4.1.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 4.1.4 getBytes(Position as Int64, Length as Integer) as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves all or part of the BLOB value that this Blob object represents, as an array of bytes.
**Notes:** This byte array contains up to length consecutive bytes starting at position pos.

pos: the ordinal position of the first byte in the BLOB value to be extracted; the first byte is at position 1

length: the number of consecutive bytes to be copied

Returns a byte array (as string) containing up to length consecutive bytes from the BLOB value designated by this Blob object, starting with the byte at position pos

### 4.1.5 length as int64

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns the number of bytes in the BLOB value designated by this Blob object.

### 4.1.6 position(SearchString as JavaBlobMBS, Position as Int64) as Int64

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the byte position in the BLOB value designated by this Blob object at which pattern begins. The search begins at position start.
**Notes:** pattern: the Blob object designating the BLOB value for which to search
start: the position in the BLOB value at which to begin searching; the first position is 1

Returns the position at which the pattern begins, else -1
See also:

- 4.1.7 position(SearchString as String, Position as Int64) as Int64 118

### 4.1.7 position(SearchString as String, Position as Int64) as Int64

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the byte position at which the specified byte array pattern begins within the BLOB value that this Blob object represents.
**Notes:** The search for pattern begins at position start.

pattern: the byte array for which to search
start: the position at which to begin searching; the first position is 1

Returns the position at which the pattern appears, else -1
See also:

- 4.1.6 position(SearchString as JavaBlobMBS, Position as Int64) as Int64 118

### 4.1.8 setBytes(Position as Int64, Value as String) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Writes the given array of bytes to the BLOB value that this Blob object represents, starting at position pos, and returns the number of bytes written.
**Notes:** pos: the position in the BLOB object at which to start writing
bytes: the array of bytes to be written to the BLOB value that this Blob object represents

Returns the number of bytes written
See also:

- 4.1.9 setBytes(Position as Int64, Value as String, Offset as Integer, Length as Integer) as Integer 119

### 4.1.9 setBytes(Position as Int64, Value as String, Offset as Integer, Length as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Writes all or part of the given byte array to the BLOB value that this Blob object represents and returns the number of bytes written.
**Notes:** Writing starts at position pos in the BLOB value; len bytes from the given byte array are written.

pos: the position in the BLOB object at which to start writing
bytes: the array of bytes to be written to this BLOB object
offset: the offset into the array bytes at which to start reading the bytes to be set
len: the number of bytes to be written to the BLOB value from the array of bytes bytes

Returns the number of bytes written
See also:

- 4.1.8 setBytes(Position as Int64, Value as String) as Integer 119

### 4.1.10 truncate(len as int64)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Truncates the BLOB value that this Blob object represents to be len bytes in length.
**Notes:** len: the length, in bytes, to which the BLOB value that this Blob object represents should be truncated

## 4.2 class JavaCallableStatementMBS

### 4.2.1 class JavaCallableStatementMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The Xojo class to handle a CallableStatement in Java.
**Notes:** The interface used to execute SQL stored procedures. The JDBC API provides a stored procedure SQL escape syntax that allows stored procedures to be called in a standard way for all RDBMSs. This escape syntax has one form that includes a result parameter and one that does not. If used, the result parameter must be registered as an OUT parameter. The other parameters can be used for input, output or both. Parameters are referred to sequentially, by number, with the first parameter being 1.

{ ?= call <procedure-name>[ <arg1>,<arg2>, ... ] }
{ call <procedure-name>[ <arg1>,<arg2>, ... ] }

IN parameter values are set using the set methods inherited from PreparedStatement. The type of all OUT parameters must be registered prior to executing the stored procedure; their values are retrieved after execution via the get methods provided here.

A CallableStatement can return one ResultSet object or multiple ResultSet objects. Multiple ResultSet objects are handled using operations inherited from Statement.

For maximum portability, a call's ResultSet objects and update counts should be processed prior to getting the values of output parameters.
Subclass of the JavaStatementMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

### 4.2.2 Methods

### 4.2.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 4.2.4 getBlob(parameterIndex as Integer) as JavaBlobMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated JDBC BLOB parameter as a Blob object in the Java programming language.

**Notes:** Parameters:
i - the first parameter is 1, the second is 2, and so on
Returns:
the parameter value as a Blob object in the Java programming language. If the value was SQL NULL, the value null is returned.
See also:


- 4.2.5 getBlob(parameterName as string) as JavaBlobMBS 121


### 4.2.5 getBlob(parameterName as string) as JavaBlobMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of a JDBC BLOB parameter as a Blob object in the Java programming language.
**Notes:** Parameters:
parameterName - the name of the parameter
Returns:
the parameter value as a Blob object in the Java programming language. If the value was SQL NULL, the value null is returned.
See also:


- 4.2.4 getBlob(parameterIndex as Integer) as JavaBlobMBS 120


### 4.2.6 getBoolean(parameterIndex as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated JDBC BIT parameter as a boolean in the Java programming language.
**Notes:** Parameters:
parameterIndex - the first parameter is 1, the second is 2, and so on
Returns:
the parameter value. If the value is SQL NULL, the result is false.
See also:


- 4.2.7 getBoolean(parameterName as string) as boolean 121


### 4.2.7 getBoolean(parameterName as string) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of a JDBC BIT parameter as a boolean in the Java programming language.
**Notes:** Parameters:
parameterName - the name of the parameter

Returns:
the parameter value. If the value is SQL NULL, the result is false.
See also:

- 4.2.6 getBoolean(parameterIndex as Integer) as boolean 121

### 4.2.8 getByte(parameterIndex as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated JDBC TINYINT parameter as a byte in the Java programming language.
**Notes:** Parameters:
parameterIndex - the first parameter is 1, the second is 2, and so on
Returns:
the parameter value. If the value is SQL NULL, the result is 0.
See also:

- 4.2.9 getByte(parameterName as string) as Integer 122

### 4.2.9 getByte(parameterName as string) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of a JDBC TINYINT parameter as a byte in the Java programming language.
**Notes:** Parameters:
parameterName - the name of the parameter
Returns:
the parameter value. If the value is SQL NULL, the result is 0.
See also:

- 4.2.8 getByte(parameterIndex as Integer) as Integer 122

### 4.2.10 getClob(parameterIndex as Integer) as JavaClobMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated JDBC CLOB parameter as a Clob object in the Java programming language.
**Notes:** Parameters:
i - the first parameter is 1, the second is 2, and so on
Returns:
the parameter value as a Clob object in the Java programming language. If the value was SQL NULL, the value null is returned.
See also:

### 4.2.11 getClob(parameterName as string) as JavaClobMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of a JDBC CLOB parameter as a Clob object in the Java programming language.
**Notes:** Parameters:
parameterName - the name of the parameter
Returns:
the parameter value as a Clob object in the Java programming language. If the value was SQL NULL, the value null is returned.
See also:


- 4.2.10 getClob(parameterIndex as Integer) as JavaClobMBS 122


### 4.2.12 getDouble(parameterIndex as Integer) as Double

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated JDBC DOUBLE parameter as a double in the Java programming language.
**Notes:** Parameters:
parameterIndex - the first parameter is 1, the second is 2, and so on
Returns:
the parameter value. If the value is SQL NULL, the result is 0.
See also:


- 4.2.13 getDouble(parameterName as string) as Double 123


### 4.2.13 getDouble(parameterName as string) as Double

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of a JDBC DOUBLE parameter as a double in the Java programming language.
**Notes:** Parameters:
parameterName - the name of the parameter
Returns:
the parameter value. If the value is SQL NULL, the result is 0.
See also:


- 4.2.12 getDouble(parameterIndex as Integer) as Double 123

### 4.2.14 getFloat(parameterIndex as Integer) as single

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated JDBC FLOAT parameter as a float in the Java programming language.
**Notes:** Parameters:
parameterIndex - the first parameter is 1, the second is 2, and so on
Returns:
the parameter value. If the value is SQL NULL, the result is 0.
See also:

- 4.2.15 getFloat(parameterName as string) as single 124

### 4.2.15 getFloat(parameterName as string) as single

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of a JDBC FLOAT parameter as a float in the Java programming language.
**Notes:** Parameters:
parameterName - the name of the parameter
Returns:
the parameter value. If the value is SQL NULL, the result is 0.
See also:

- 4.2.14 getFloat(parameterIndex as Integer) as single 124

### 4.2.16 getInt(parameterIndex as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated JDBC INTEGER parameter as an int in the Java programming language.
**Notes:** Parameters:
parameterIndex - the first parameter is 1, the second is 2, and so on
Returns:
the parameter value. If the value is SQL NULL, the result is 0.
See also:

- 4.2.17 getInt(parameterName as string) as Integer 124

### 4.2.17 getInt(parameterName as string) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of a JDBC INTEGER parameter as an int in the Java programming language.
**Notes:** Parameters:
parameterName - the name of the parameter
Returns:
the parameter value. If the value is SQL NULL, the result is 0.
See also:

- 4.2.16 getInt(parameterIndex as Integer) as Integer 124

### 4.2.18 getLong(parameterIndex as Integer) as Int64

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated JDBC BIGINT parameter as a long in the Java programming language.
**Notes:** Parameters:
parameterIndex - the first parameter is 1, the second is 2, and so on
Returns:
the parameter value. If the value is SQL NULL, the result is 0.
See also:

- 4.2.19 getLong(parameterName as string) as Int64 125

### 4.2.19 getLong(parameterName as string) as Int64

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of a JDBC BIGINT parameter as a long in the Java programming language.
**Notes:** Parameters:
parameterName - the name of the parameter
Returns:
the parameter value. If the value is SQL NULL, the result is 0.
See also:

- 4.2.18 getLong(parameterIndex as Integer) as Int64 125

### 4.2.20 getShort(parameterIndex as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated JDBC SMALLINT parameter as a short in the Java programming language.
**Notes:** Parameters:
parameterIndex - the first parameter is 1, the second is 2, and so on
Returns:

the parameter value. If the value is SQL NULL, the result is 0.
See also:

- 4.2.21 getShort(parameterName as string) as Integer 126

### 4.2.21 getShort(parameterName as string) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of a JDBC SMALLINT parameter as a short in the Java programming language.
**Notes:** Parameters:
parameterName - the name of the parameter
Returns:
the parameter value. If the value is SQL NULL, the result is 0.
See also:

- 4.2.20 getShort(parameterIndex as Integer) as Integer 125

### 4.2.22 getString(parameterIndex as Integer) as String

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated JDBC CHAR, VARCHAR, or LONGVARCHAR parameter as a String in the Java programming language.
**Notes:** For the fixed-length type JDBC CHAR, the String object returned has exactly the same value the JDBC CHAR value had in the database, including any padding added by the database.

Parameters:
parameterIndex - the first parameter is 1, the second is 2, and so on
Returns:
the parameter value. If the value is SQL NULL, the result is null.
See also:

- 4.2.23 getString(parameterName as string) as String 126

### 4.2.23 getString(parameterName as string) as String

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of a JDBC CHAR, VARCHAR, or LONGVARCHAR parameter as a String in the Java programming language.
**Notes:** For the fixed-length type JDBC CHAR, the String object returned has exactly the same value the JDBC CHAR value had in the database, including any padding added by the database.

Parameters:
parameterName - the name of the parameter
Returns:
the parameter value. If the value is SQL NULL, the result is null.
See also:

- 4.2.22 getString(parameterIndex as Integer) as String 126

### 4.2.24 registerOutParameter(parameterIndex as Integer, sqlType as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Registers the OUT parameter in ordinal position parameterIndex to the JDBC type sqlType.
**Notes:** All OUT parameters must be registered before a stored procedure is executed.
The JDBC type specified by sqlType for an OUT parameter determines the Java type that must be used in the get method to read the value of that parameter.

If the JDBC type expected to be returned to this output parameter is specific to this particular database, sqlType should be java.sql.Types.OTHER. The method getObject(int) retrieves the value.

parameterIndex - the first parameter is 1, the second is 2, and so on
sqlType - the JDBC type code defined by java.sql.Types. If the parameter is of JDBC type NUMERIC or DECIMAL, the version of registerOutParameter that accepts a scale value should be used.
See also:

- 4.2.25 registerOutParameter(parameterIndex as Integer, sqlType as Integer, scale as Integer) 127
- 4.2.26 registerOutParameter(parameterIndex as Integer, sqlType as Integer, typeName as string) 128
- 4.2.27 registerOutParameter(parameterName as string, sqlType as Integer) 129
- 4.2.28 registerOutParameter(parameterName as string, sqlType as Integer, scale as Integer) 129
- 4.2.29 registerOutParameter(parameterName as string, sqlType as Integer, typeName as string) 130

### 4.2.25 registerOutParameter(parameterIndex as Integer, sqlType as Integer, scale as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Registers the parameter in ordinal position parameterIndex to be of JDBC type sqlType.
**Notes:** This method must be called before a stored procedure is executed.
The JDBC type specified by sqlType for an OUT parameter determines the Java type that must be used in the get method to read the value of that parameter.

This version of registerOutParameter should be used when the parameter is of JDBC type NUMERIC or DECIMAL.

parameterIndex - the first parameter is 1, the second is 2, and so on
sqlType - the SQL type code defined by java.sql.Types.
scale - the desired number of digits to the right of the decimal point. It must be greater than or equal to zero.
See also:


- 4.2.24 registerOutParameter(parameterIndex as Integer, sqlType as Integer) 127
- 4.2.26 registerOutParameter(parameterIndex as Integer, sqlType as Integer, typeName as string) 128
- 4.2.27 registerOutParameter(parameterName as string, sqlType as Integer) 129
- 4.2.28 registerOutParameter(parameterName as string, sqlType as Integer, scale as Integer) 129
- 4.2.29 registerOutParameter(parameterName as string, sqlType as Integer, typeName as string) 130


### 4.2.26 registerOutParameter(parameterIndex as Integer, sqlType as Integer, typeName as string)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Registers the designated output parameter.
**Notes:** This version of the method registerOutParameter should be used for a user-defined or REF output parameter. Examples of user-defined types include: STRUCT, DISTINCT, JAVA_OBJECT, and named array types. Before executing a stored procedure call, you must explicitly call registerOutParameter to register the type from java.sql.Types for each OUT parameter. For a user-defined parameter, the fully-qualified SQL type name of the parameter should also be given, while a REF parameter requires that the fully-qualified type name of the referenced type be given. A JDBC driver that does not need the type code and type name information may ignore it. To be portable, however, applications should always provide these values for user-defined and REF parameters. Although it is intended for user-defined and REF parameters, this method may be used to register a parameter of any JDBC type. If the parameter does not have a user-defined or REF type, the typeName parameter is ignored.
Note: When reading the value of an out parameter, you must use the getter method whose Java type corresponds to the parameter's registered SQL type.

Parameters:
paramIndex - the first parameter is 1, the second is 2,...
sqlType - a value from Types
typeName - the fully-qualified name of an SQL structured type
See also:


- 4.2.24 registerOutParameter(parameterIndex as Integer, sqlType as Integer) 127
- 4.2.25 registerOutParameter(parameterIndex as Integer, sqlType as Integer, scale as Integer) 127

### 4.2.27 registerOutParameter(parameterName as string, sqlType as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Registers the OUT parameter named parameterName to the JDBC type sqlType.
**Notes:** Registers the OUT parameter named parameterName to the JDBC type sqlType. All OUT parameters must be registered before a stored procedure is executed.
The JDBC type specified by sqlType for an OUT parameter determines the Java type that must be used in the get method to read the value of that parameter.

If the JDBC type expected to be returned to this output parameter is specific to this particular database, sqlType should be java.sql.Types.OTHER. The method getObject(int) retrieves the value.

Parameters:
parameterName - the name of the parameter
sqlType - the JDBC type code defined by java.sql.Types. If the parameter is of JDBC type NUMERIC or DECIMAL, the version of registerOutParameter that accepts a scale value should be used.
See also:



### 4.2.28 registerOutParameter(parameterName as string, sqlType as Integer, scale as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Registers the parameter named parameterName to be of JDBC type sqlType.
**Notes:** This method must be called before a stored procedure is executed.
The JDBC type specified by sqlType for an OUT parameter determines the Java type that must be used in the get method to read the value of that parameter.

This version of registerOutParameter should be used when the parameter is of JDBC type NUMERIC or DECIMAL.

Parameters:
parameterName - the name of the parameter
sqlType - SQL type code defined by java.sql.Types.
scale - the desired number of digits to the right of the decimal point. It must be greater than or equal to zero.
See also:


- 4.2.24 registerOutParameter(parameterIndex as Integer, sqlType as Integer) 127
- 4.2.25 registerOutParameter(parameterIndex as Integer, sqlType as Integer, scale as Integer) 127
- 4.2.26 registerOutParameter(parameterIndex as Integer, sqlType as Integer, typeName as string) 128
- 4.2.27 registerOutParameter(parameterName as string, sqlType as Integer) 129
- 4.2.29 registerOutParameter(parameterName as string, sqlType as Integer, typeName as string) 130


### 4.2.29 registerOutParameter(parameterName as string, sqlType as Integer, typeName as string)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Registers the designated output parameter.
**Notes:** This version of the method registerOutParameter should be used for a user-named or REF output parameter. Examples of user-named types include: STRUCT, DISTINCT, JAVA_OBJECT, and named array types. Before executing a stored procedure call, you must explicitly call registerOutParameter to register the type from java.sql.Types for each OUT parameter. For a user-named parameter the fully-qualified SQL type name of the parameter should also be given, while a REF parameter requires that the fully-qualified type name of the referenced type be given. A JDBC driver that does not need the type code and type name information may ignore it. To be portable, however, applications should always provide these values for user-named and REF parameters. Although it is intended for user-named and REF parameters, this method may be used to register a parameter of any JDBC type. If the parameter does not have a user-named or REF type, the typeName parameter is ignored.
Note: When reading the value of an out parameter, you must use the getXXX method whose Java type XXX corresponds to the parameter's registered SQL type.

Parameters:
parameterName - the name of the parameter
sqlType - a value from Types
typeName - the fully-qualified name of an SQL structured type
See also:


- 4.2.24 registerOutParameter(parameterIndex as Integer, sqlType as Integer) 127
- 4.2.25 registerOutParameter(parameterIndex as Integer, sqlType as Integer, scale as Integer) 127

### 4.2.30 setBoolean(parameterName as string, x as boolean)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java boolean value.
**Notes:** The driver converts this to an SQL BIT value when it sends it to the database.
Parameters:
parameterName - the name of the parameter
x - the parameter value

### 4.2.31 setByte(parameterName as string, x as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java byte value.
**Notes:** The driver converts this to an SQL TINYINT value when it sends it to the database.
Parameters:
parameterName - the name of the parameter
x - the parameter value

### 4.2.32 setDouble(parameterName as string, x as Double)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java double value.
**Notes:** The driver converts this to an SQL DOUBLE value when it sends it to the database.
Parameters:
parameterName - the name of the parameter
x - the parameter value

### 4.2.33 setFloat(parameterName as string, x as single)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java float value.
**Notes:** The driver converts this to an SQL FLOAT value when it sends it to the database.
Parameters:

parameterName - the name of the parameter
x - the parameter value

### 4.2.34 setInt(parameterName as string, x as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java int value.
**Notes:** The driver converts this to an SQL INTEGER value when it sends it to the database.
Parameters:
parameterName - the name of the parameter
x - the parameter value

### 4.2.35 setLong(parameterName as string, x as int64)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java long value.
**Notes:** The driver converts this to an SQL BIGINT value when it sends it to the database.
Parameters:
parameterName - the name of the parameter
x - the parameter value

### 4.2.36 setNull(parameterName as string, sqlType as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to SQL NULL.
**Notes:** Note: You must specify the parameter's SQL type.

Parameters:
parameterName - the name of the parameter
sqlType - the SQL type code defined in java.sql.Types
See also:

- 4.2.37 setNull(parameterName as string, sqlType as Integer, typeName as string) 132

### 4.2.37 setNull(parameterName as string, sqlType as Integer, typeName as string)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to SQL NULL.
**Notes:** This version of the method setNull should be used for user-defined types and REF type parameters. Examples of user-defined types include: STRUCT, DISTINCT, JAVA_OBJECT, and named array types. Note: To be portable, applications must give the SQL type code and the fully-qualified SQL type name when specifying a NULL user-defined or REF parameter. In the case of a user-defined type the name is the type name of the parameter itself. For a REF parameter, the name is the type name of the referenced type. If a JDBC driver does not need the type code or type name information, it may ignore it. Although it is intended for user-defined and Ref parameters, this method may be used to set a null parameter of any JDBC type. If the parameter does not have a user-defined or REF type, the given typeName is ignored.

Parameters:
parameterName - the name of the parameter
sqlType - a value from java.sql.Types
typeName - the fully-qualified name of an SQL user-defined type; ignored if the parameter is not a user-defined type or SQL REF value
See also:

- 4.2.36 setNull(parameterName as string, sqlType as Integer) 132

### 4.2.38 setShort(parameterName as string, x as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java short value.
**Notes:** The driver converts this to an SQL SMALLINT value when it sends it to the database.
Parameters:
parameterName - the name of the parameter
x - the parameter value

### 4.2.39 setString(parameterName as string, x as string)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java String value.
**Notes:** The driver converts this to an SQL VARCHAR or LONGVARCHAR value (depending on the argument's size relative to the driver's limits on VARCHAR values) when it sends it to the database.
Parameters:
parameterName - the name of the parameter
x - the parameter value

### 4.2.40 wasNull as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether the last OUT parameter read had the value of SQL NULL.
**Notes:** Note that this method should be called only after calling a getter method; otherwise, there is no value to use in determining whether it is null or not.

Returns:
true if the last parameter read was SQL NULL; false otherwise

# 4.3 class JavaClobMBS

## 4.3.1 class JavaClobMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The java class for large character objects.
**Notes:** Subclass of the JavaObjectMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

## 4.3.2 Methods

## 4.3.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

## 4.3.4 getSubString(Position as int64, Length as Integer) as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a copy of the specified substring in the CLOB value designated by this Clob object.
**Notes:** The substring begins at position pos and has up to length consecutive characters.

Parameters:
pos: the first character of the substring to be extracted. The first character is at position 1.
length: the number of consecutive characters to be copied

Returns a String that is the specified substring in the CLOB value designated by this Clob object

## 4.3.5 length as int64

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the number of characters in the CLOB value designated by this Clob object.
**Notes:** Returns length of the CLOB in characters.

### 4.3.6 position(SearchString as JavaClobMBS, Start as Int64) as Int64

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the character position at which the specified Clob object searchstr appears in this Clob object.
**Notes:** The search begins at position start.

SearchString: the Clob object for which to search
start: the position at which to begin searching; the first position is 1

Returns the position at which the Clob object appears or -1 if it is not present; the first position is 1
See also:

- 4.3.7 position(SearchString as String, Start as Int64) as Int64 136

### 4.3.7 position(SearchString as String, Start as Int64) as Int64

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the character position at which the specified substring searchstr appears in the SQL CLOB value represented by this Clob object.
**Notes:** The search begins at position start.

searchstr: the substring for which to search
start: the position at which to begin searching; the first position is 1

Returns the position at which the substring appears or -1 if it is not present; the first position is 1
See also:

- 4.3.6 position(SearchString as JavaClobMBS, Start as Int64) as Int64 136

### 4.3.8 setString(Position as Int64, Value as String) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Writes the given Java String to the CLOB value that this Clob object designates at the position pos.
**Notes:** Position: the position at which to start writing to the CLOB value that this Clob object represents
Value: the string to be written to the CLOB value that this Clob designates

Returns the number of characters written
See also:

### 4.3.9 setString(Position as Int64, Value as String, Offset as Integer, Length as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Writes Length characters of Value, starting at character offset, to the CLOB value that this Clob represents.
**Notes:** Position: the position at which to start writing to this CLOB object
Value: the string to be written to the CLOB value that this Clob object represents
Offset: the offset into str to start reading the characters to be written
Length: the number of characters to be written

Returns the number of characters written
See also:



### 4.3.10 truncate(len as int64)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Truncates the CLOB value that this Clob designates to have a length of len characters.
**Notes:** len: the length, in bytes, to which the CLOB value should be truncated

## 4.4 class JavaConnectionMBS

### 4.4.1 class JavaConnectionMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The class for a java connection.
**Notes:** Subclass of the JavaObjectMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
**Blog Entries**

- Encrypted Access database in Xojo
- MBS Real Studio Plugins, version 11.3pr11

### 4.4.2 Methods

### 4.4.3 clearWarnings

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Clears all warnings reported for this Connection object.
**Notes:** After a call to this method, the method getWarnings returns null until a new warning is reported for this Connection object.

### 4.4.4 close

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Releases this Connection object's database and JDBC resources immediately instead of waiting for them to be automatically released.

### 4.4.5 CLOSE_CURSORS_AT_COMMIT as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that ResultSet objects should be closed when the method Connection.commit is called.

### 4.4.6 commit

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Makes all changes made since the previous commit/rollback permanent and releases any database locks currently held by the Connection.
**Notes:** See the java documentation for details on java.sql.Connection.Commit.

### 4.4.7 CONCUR_READ_ONLY as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating the concurrency mode for a ResultSet object that may NOT be updated.

### 4.4.8 CONCUR_UPDATABLE as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating the concurrency mode for a ResultSet object that may be updated.

### 4.4.9 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 4.4.10 createBlob as JavaBlobMBS

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Constructs an object that implements the Blob interface.
**Notes:** The object returned initially contains no data. The setBinaryStream and setBytes methods of the Blob interface may be used to add data to the Blob.

### 4.4.11 createClob as JavaClobMBS

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Constructs an object that implements the Clob interface.
**Notes:** The object returned initially contains no data. The setAsciiStream, setCharacterStream and setString methods of the Clob interface may be used to add data to the Clob.

### 4.4.12 createStatement as JavaStatementMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a Statement object for sending SQL statements to the database.
**Notes:** SQL statements without parameters are normally executed using Statement objects. If the same SQL statement is executed many times, it may be more efficient to use a PreparedStatement object.
Result sets created using the returned Statement object will by default be type TYPE_FORWARD_ONLY and have a concurrency level of CONCUR_READ_ONLY.

Returns:
a new default Statement object
See also:

- 4.4.13 createStatement(resultSetType as Integer, resultSetConcurrency as Integer) as JavaStatementMBS 140
- 4.4.14 createStatement(resultSetType as Integer, resultSetConcurrency as Integer, resultSetHoldability as Integer) as JavaStatementMBS 140

### 4.4.13 createStatement(resultSetType as Integer, resultSetConcurrency as Integer) as JavaStatementMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a Statement object that will generate ResultSet objects with the given type and concurrency.
**Notes:** This method is the same as the createStatement method above, but it allows the default result set type and concurrency to be overridden.
Parameters:
resultSetType - a result set type; one of ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE
resultSetConcurrency - a concurrency type; one of ResultSet.CONCUR_READ_ONLY or ResultSet.CONCUR_UPDATABLE
Returns:
a new Statement object that will generate ResultSet objects with the given type and concurrency
See also:

- 4.4.12 createStatement as JavaStatementMBS 140
- 4.4.14 createStatement(resultSetType as Integer, resultSetConcurrency as Integer, resultSetHoldability as Integer) as JavaStatementMBS 140

### 4.4.14 createStatement(resultSetType as Integer, resultSetConcurrency as Integer, resultSetHoldability as Integer) as JavaStatementMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a Statement object that will generate ResultSet objects with the given type, concurrency, and holdability.
**Notes:** This method is the same as the createStatement method above, but it allows the default result set type, concurrency, and holdability to be overridden.
Parameters:
resultSetType - one of the following ResultSet constants: ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE
resultSetConcurrency - one of the following ResultSet constants: ResultSet.CONCUR_READ_ONLY or ResultSet.CONCUR_UPDATABLE
resultSetHoldability - one of the following ResultSet constants: ResultSet.HOLD_CURSORS_OVER_COMMIT or ResultSet.CLOSE_CURSORS_AT_COMMIT
Returns:
a new Statement object that will generate ResultSet objects with the given type, concurrency, and holdability
See also:


- 4.4.12 createStatement as JavaStatementMBS 140
- 4.4.13 createStatement(resultSetType as Integer, resultSetConcurrency as Integer) as JavaStatementMBS 140


### 4.4.15 FETCH_FORWARD as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that the rows in a result set will be processed in a forward direction; first-to-last.
**Notes:** This constant is used by the method setFetchDirection as a hint to the driver, which the driver may ignore.

### 4.4.16 FETCH_REVERSE as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that the rows in a result set will be processed in a reverse direction; last-to-first.
**Notes:** This constant is used by the method setFetchDirection as a hint to the driver, which the driver may ignore.

### 4.4.17 FETCH_UNKNOWN as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that the order in which rows in a result set will be processed is unknown.
**Notes:** This constant is used by the method setFetchDirection as a hint to the driver, which the driver may

ignore.

### 4.4.18 getMetaData as JavaDatabaseMetaDataMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a DatabaseMetaData object that contains metadata about the database to which this Connection object represents a connection.
**Notes:** The metadata includes information about the database's tables, its supported SQL grammar, its stored procedures, the capabilities of this connection, and so on.
Returns:
a DatabaseMetaData object for this Connection object

### 4.4.19 HOLD_CURSORS_OVER_COMMIT as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that ResultSet objects should not be closed when the method Connection.commit is called.

### 4.4.20 isClosed as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this Connection object has been closed.
**Notes:** A connection is closed if the method close has been called on it or if certain fatal errors have occurred. This method is guaranteed to return true only when it is called after the method Connection.close has been called.
This method generally cannot be called to determine whether a connection to a database is valid or invalid. A typical client can determine that a connection is invalid by catching any exceptions that might be thrown when an operation is attempted.

Returns:
true if this Connection object is closed; false if it is still open

### 4.4.21 nativeSQL(sql as string) as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Converts the given SQL statement into the system's native SQL grammar.
**Notes:** A driver may convert the JDBC SQL grammar into its system's native SQL grammar prior to

sending it. This method returns the native form of the statement that the driver would have sent.
Parameters:
sql - an SQL statement that may contain one or more '?' parameter placeholders
Returns:
the native form of this statement

### 4.4.22 prepareCall(sql as string) as JavaCallableStatementMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a CallableStatement object for calling database stored procedures.
**Notes:** The CallableStatement object provides methods for setting up its IN and OUT parameters, and methods for executing the call to a stored procedure.
Note: This method is optimized for handling stored procedure call statements. Some drivers may send the call statement to the database when the method prepareCall is done; others may wait until the CallableStatement object is executed. This has no direct effect on users; however, it does affect which method throws certain SQLExceptions.

Result sets created using the returned CallableStatement object will by default be type TYPE_FORWARD_ONLY and have a concurrency level of CONCUR_READ_ONLY.

Parameters:
sql - an SQL statement that may contain one or more '?' parameter placeholders. Typically this statement is a JDBC function call escape string.
Returns:
a new default CallableStatement object containing the pre-compiled SQL statement
See also:


- 4.4.23 prepareCall(sql as string, resultSetType as Integer, resultSetConcurrency as Integer) as JavaCallableStatementMBS 143
- 4.4.24 prepareCall(sql as string, resultSetType as Integer, resultSetConcurrency as Integer, resultSetHoldability as Integer) as JavaCallableStatementMBS 144


### 4.4.23 prepareCall(sql as string, resultSetType as Integer, resultSetConcurrency as Integer) as JavaCallableStatementMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a CallableStatement object that will generate ResultSet objects with the given type and concurrency.
**Notes:** This method is the same as the prepareCall method above, but it allows the default result set type and concurrency to be overridden.
Parameters:

sql - a String object that is the SQL statement to be sent to the database; may contain on or more ? parameters
resultSetType - a result set type; one of ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE
resultSetConcurrency - a concurrency type; one of ResultSet.CONCUR_READ_ONLY or ResultSet.CONCUR_UPDATABLE
Returns:
a new CallableStatement object containing the pre-compiled SQL statement that will produce ResultSet objects with the given type and concurrency
See also:


- 4.4.22 prepareCall(sql as string) as JavaCallableStatementMBS 143
- 4.4.24 prepareCall(sql as string, resultSetType as Integer, resultSetConcurrency as Integer, resultSetHoldability as Integer) as JavaCallableStatementMBS 144


### 4.4.24 prepareCall(sql as string, resultSetType as Integer, resultSetConcurrency as Integer, resultSetHoldability as Integer) as JavaCallableStatementMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a CallableStatement object that will generate ResultSet objects with the given type and concurrency.
**Notes:** This method is the same as the prepareCall method above, but it allows the default result set type, result set concurrency type and holdability to be overridden.
Parameters:
sql - a String object that is the SQL statement to be sent to the database; may contain on or more ? parameters
resultSetType - one of the following ResultSet constants: ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE
resultSetConcurrency - one of the following ResultSet constants: ResultSet.CONCUR_READ_ONLY or ResultSet.CONCUR_UPDATABLE
resultSetHoldability - one of the following ResultSet constants: ResultSet.HOLD_CURSORS_OVER_COMMIT or ResultSet.CLOSE_CURSORS_AT_COMMIT
Returns:
a new CallableStatement object, containing the pre-compiled SQL statement, that will generate ResultSet objects with the given type, concurrency, and holdability
See also:


- 4.4.22 prepareCall(sql as string) as JavaCallableStatementMBS 143
- 4.4.23 prepareCall(sql as string, resultSetType as Integer, resultSetConcurrency as Integer) as JavaCallableStatementMBS 143

### 4.4.25 prepareStatement(sql as string) as JavaPreparedStatementMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a PreparedStatement object for sending parameterized SQL statements to the database.
**Notes:** A SQL statement with or without IN parameters can be pre-compiled and stored in a PreparedStatement object. This object can then be used to efficiently execute this statement multiple times.

Note: This method is optimized for handling parametric SQL statements that benefit from precompilation. If the driver supports precompilation, the method prepareStatement will send the statement to the database for precompilation. Some drivers may not support precompilation. In this case, the statement may not be sent to the database until the PreparedStatement object is executed. This has no direct effect on users; however, it does affect which methods throw certain SQLException objects.

Result sets created using the returned PreparedStatement object will by default be type TYPE_FORWARD_ONLY and have a concurrency level of CONCUR_READ_ONLY.

Parameters:
sql - an SQL statement that may contain one or more '?' IN parameter placeholders
See also:


- 4.4.26 prepareStatement(sql as string, autoGeneratedKeys as Integer) as JavaPreparedStatementMBS 145
- 4.4.27 prepareStatement(sql as string, resultSetType as Integer, resultSetConcurrency as Integer) as JavaPreparedStatementMBS 146
- 4.4.28 prepareStatement(sql as string, resultSetType as Integer, resultSetConcurrency as Integer, resultSetHoldability as Integer) as JavaPreparedStatementMBS 147


### 4.4.26 prepareStatement(sql as string, autoGeneratedKeys as Integer) as JavaPreparedStatementMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a default PreparedStatement object that has the capability to retrieve auto-generated keys.
**Notes:** The given constant tells the driver whether it should make auto-generated keys available for retrieval. This parameter is ignored if the SQL statement is not an INSERT statement.
Note: This method is optimized for handling parametric SQL statements that benefit from precompilation. If the driver supports precompilation, the method prepareStatement will send the statement to the database for precompilation. Some drivers may not support precompilation. In this case, the statement may not be sent to the database until the PreparedStatement object is executed. This has no direct effect on users; however, it does affect which methods throw certain SQLExceptions.

Result sets created using the returned PreparedStatement object will by default be type TYPE_FORWARD_ONLY and have a concurrency level of CONCUR_READ_ONLY.

Parameters:
sql - an SQL statement that may contain one or more '?' IN parameter placeholders
autoGeneratedKeys - a flag indicating whether auto-generated keys should be returned; one of Statement.RETURN_GENERATED_KEYS or Statement.NO_GENERATED_KEYS
Returns:
a new PreparedStatement object, containing the pre-compiled SQL statement, that will have the capability of returning auto-generated keys
See also:


- 4.4.25 prepareStatement(sql as string) as JavaPreparedStatementMBS 145
- 4.4.27 prepareStatement(sql as string, resultSetType as Integer, resultSetConcurrency as Integer) as JavaPreparedStatementMBS 146
- 4.4.28 prepareStatement(sql as string, resultSetType as Integer, resultSetConcurrency as Integer, resultSetHoldability as Integer) as JavaPreparedStatementMBS 147


### 4.4.27 prepareStatement(sql as string, resultSetType as Integer, resultSetConcurrency as Integer) as JavaPreparedStatementMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a PreparedStatement object that will generate ResultSet objects with the given type and concurrency.
**Notes:** This method is the same as the prepareStatement method above, but it allows the default result set type and concurrency to be overridden.
Parameters:
sql - a String object that is the SQL statement to be sent to the database; may contain one or more ? IN parameters
resultSetType - a result set type; one of ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE
resultSetConcurrency - a concurrency type; one of ResultSet.CONCUR_READ_ONLY or ResultSet.CONCUR_UPDATABLE
Returns:
a new PreparedStatement object containing the pre-compiled SQL statement that will produce ResultSet objects with the given type and concurrency
See also:


- 4.4.25 prepareStatement(sql as string) as JavaPreparedStatementMBS 145
- 4.4.26 prepareStatement(sql as string, autoGeneratedKeys as Integer) as JavaPreparedStatementMBS 145
- 4.4.28 prepareStatement(sql as string, resultSetType as Integer, resultSetConcurrency as Integer, resultSetHoldability as Integer) as JavaPreparedStatementMBS 147

### 4.4.28 prepareStatement(sql as string, resultSetType as Integer, resultSetConcurrency as Integer, resultSetHoldability as Integer) as JavaPreparedStatementMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a PreparedStatement object that will generate ResultSet objects with the given type, concurrency, and holdability.
**Notes:** This method is the same as the prepareStatement method above, but it allows the default result set type, concurrency, and holdability to be overridden.

Parameters:
sql - a String object that is the SQL statement to be sent to the database; may contain one or more ? IN parameters
resultSetType - one of the following ResultSet constants: ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE
resultSetConcurrency - one of the following ResultSet constants: ResultSet.CONCUR_READ_ONLY or ResultSet.CONCUR_UPDATABLE
resultSetHoldability - one of the following ResultSet constants: ResultSet.HOLD_CURSORS_OVER_COMMIT or ResultSet.CLOSE_CURSORS_AT_COMMIT
Returns:
a new PreparedStatement object, containing the pre-compiled SQL statement, that will generate ResultSet objects with the given type, concurrency, and holdability
See also:

- 4.4.25 prepareStatement(sql as string) as JavaPreparedStatementMBS 145
- 4.4.26 prepareStatement(sql as string, autoGeneratedKeys as Integer) as JavaPreparedStatementMBS 145
- 4.4.27 prepareStatement(sql as string, resultSetType as Integer, resultSetConcurrency as Integer) as JavaPreparedStatementMBS 146

### 4.4.29 releaseSavepoint(safepoint as JavaSavepointMBS)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Removes the given Savepoint object from the current transaction.
**Notes:** Any reference to the savepoint after it have been removed will cause an SQLException to be thrown.

savepoint: the Savepoint object to be removed

### 4.4.30 rollback

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Drops all changes made since the previous commit/rollback and releases any database locks currently held by this Connection.
**Notes:** This method should be used only when auto- commit has been disabled.

See the java documentation for details on java.sql.Connection.Rollback.
See also:

- 4.4.31 rollback(safepoint as JavaSavepointMBS) 148

### 4.4.31 rollback(safepoint as JavaSavepointMBS)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Undoes all changes made after the given Savepoint object was set.
**Notes:** This method should be used only when auto-commit has been disabled.

savepoint: the Savepoint object to roll back to
See also:

- 4.4.30 rollback 147

### 4.4.32 setSavepoint as JavaSavepointMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates an unnamed savepoint in the current transaction and returns the new Savepoint object that represents it.
See also:

- 4.4.33 setSavepoint(name as string) as JavaSavepointMBS 148

### 4.4.33 setSavepoint(name as string) as JavaSavepointMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a savepoint with the given name in the current transaction and returns the new Savepoint object that represents it.
**Notes:** name: a String containing the name of the savepoint
returns the new Savepoint object
See also:

- 4.4.32 setSavepoint as JavaSavepointMBS 148

### 4.4.34 TRANSACTION_NONE as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** A constant indicating that transactions are not supported.

### 4.4.35 TRANSACTION_READ_COMMITTED as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** A constant indicating that dirty reads are prevented; non-repeatable reads and phantom reads can occur.
**Notes:** This level only prohibits a transaction from reading a row with uncommitted changes in it.

### 4.4.36 TRANSACTION_READ_UNCOMMITTED as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** A constant indicating that dirty reads, non-repeatable reads and phantom reads can occur.
**Notes:** This level allows a row changed by one transaction to be read by another transaction before any changes in that row have been committed (a "dirty read"). If any of the changes are rolled back, the second transaction will have retrieved an invalid row.

### 4.4.37 TRANSACTION_REPEATABLE_READ as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** A constant indicating that dirty reads and non-repeatable reads are prevented; phantom reads can occur.
**Notes:** This level prohibits a transaction from reading a row with uncommitted changes in it, and it also prohibits the situation where one transaction reads a row, a second transaction alters the row, and the first transaction rereads the row, getting different values the second time (a "non-repeatable read").

### 4.4.38 TRANSACTION_SERIALIZABLE as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** A constant indicating that dirty reads, non-repeatable reads and phantom reads are prevented.
**Notes:** This level includes the prohibitions in TRANSACTION_REPEATABLE_READ and further prohibits the situation where one transaction reads all rows that satisfy a WHERE condition, a second transaction inserts a row that satisfies that WHERE condition, and the first transaction rereads for the same

condition, retrieving the additional ”phantom” row in the second read.

### 4.4.39 typeARRAY as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type ARRAY.

### 4.4.40 typeBIGINT as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type BIGINT.
**Example:**

```
Var d as JavaConnectionMBS
MsgBox str(d.typeBIGINT)
```

### 4.4.41 typeBINARY as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type BINARY.

### 4.4.42 typeBIT as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type BIT.

### 4.4.43 typeBLOB as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type BLOB.

### 4.4.44 typeCHAR as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type CHAR.

### 4.4.45 typeCLOB as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type CLOB.

### 4.4.46 typeDATE as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type DATE.

### 4.4.47 typeDECIMAL as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type DECIMAL.

### 4.4.48 typeDISTINCT as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type DISTINCT.

### 4.4.49 typeDOUBLE as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type DOUBLE.

### 4.4.50 typeFLOAT as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type FLOAT.

### 4.4.51 typeINTEGER as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type INTEGER.

### 4.4.52 typeJAVA_OBJECT as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type JAVA_OBJECT.

### 4.4.53 typeLONGVARBINARY as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type LONGVARBINARY.

### 4.4.54 typeLONGVARCHAR as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type LONGVARCHAR.

### 4.4.55 typeNULL as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type NULL.

### 4.4.56 typeNUMERIC as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type NUMERIC.

### 4.4.57 typeOTHER as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language that indicates that the SQL type is database-specific and gets mapped to a Java object that can be accessed via the methods getObject and setObject.

### 4.4.58 typeREAL as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type REAL.

### 4.4.59 typeREF as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type REF.

### 4.4.60 typeSMALLINT as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type SMALLINT.

### 4.4.61 typeSTRUCT as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type STRUCT.

### 4.4.62 typeTIME as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type TIME.

### 4.4.63 typeTIMESTAMP as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type TIMESTAMP.

### 4.4.64 typeTINYINT as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type TINYINT.

### 4.4.65 typeVARBINARY as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type VARBINARY.

### 4.4.66 typeVARCHAR as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant in the Java programming language, sometimes referred to as a type code, that identifies the generic SQL type VARCHAR.

### 4.4.67 TYPE_FORWARD_ONLY as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating the type for a ResultSet object whose cursor may move only forward.

### 4.4.68 TYPE_SCROLL_INSENSITIVE as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating the type for a ResultSet object that is scrollable but generally not sensitive to changes made by others.

### 4.4.69 TYPE_SCROLL_SENSITIVE as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating the type for a ResultSet object that is scrollable and generally sensitive to changes made by others.

### 4.4.70 Properties

### 4.4.71 AutoCommit as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The current auto-commit state.
**Notes:** See the java documentation for details on java.sql.Connection.SetAutoCommit.
(Read and Write computed property)

### 4.4.72 Catalog as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets or gets the given catalog name in order to select a subspace of this Connection object's database in which to work.
**Notes:** If the driver does not support catalogs, it will silently ignore this request.
(Read and Write computed property)

### 4.4.73 Holdability as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Changes or retrieves the holdability of ResultSet objects created using this Connection object to the given holdability.
**Notes:** Parameters:
holdability - a ResultSet holdability constant; one of ResultSet.HOLD_CURSORS_OVER_COMMIT or ResultSet.CLOSE_CURSORS_AT_COMMIT
(Read and Write computed property)

### 4.4.74 ReadOnly as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Whether this connection in read-only mode as a hint to enable database optimizations.
**Notes:** See the java documentation for details on java.sql.Connection.SetReadOnly.
(Read and Write computed property)

### 4.4.75 TransactionIsolation as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Attempts to change the transaction isolation level for this Connection object to the one given.
**Notes:** The constants defined in the interface Connection are the possible transaction isolation levels.
Note: If this method is called during a transaction, the result is implementation-defined.

Parameters:
level - one of the following Connection constants: Connection.TRANSACTION_READ_UNCOMMITTED, Connection.TRANSACTION_READ_COMMITTED, Connection.TRANSACTION_REPEATABLE_READ, or Connection.TRANSACTION_SERIALIZABLE. (Note that Connection.TRANSACTION_NONE cannot be used because it specifies that transactions are not supported.)
(Read and Write computed property)

# 4.5 class JavaDatabaseMBS

## 4.5.1 class JavaDatabaseMBS

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The class to handle database access using JDBC drivers.
**Notes:** This class is not a subclass of RB's database class, so you can use it with Xojo Standard Edition.

Nearly all methods on this class can raise java exceptions which you can get using the error property. (and errorstring and errorcode)

Add Linux support plugin version 8.7.

Please make sure this Java VM object stays alive until you are done with all your java stuff. So all the java objects go away and this vm object is destroyed on the end. Because if some java code is still running like an background java thread, quitting the VM can lead into crashes.
Subclass of the JavaObjectMBS class.
**Blog Entries**

- Prefetching records from databases
- Encrypted Access database in Xojo
- MBS Real Studio Plugins, version 12.1pr6
- MonkeyBread Software Releases the MBS Plugins 8.3

## 4.5.2 Methods

## 4.5.3 connect(url as string) as JavaConnectionMBS

Plugin Version: 12.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Connects directly to the database calling the Drivers's connect method.
**Notes:** While getConnection calls DriverManager, this calls directly the driver.

Attempts to make a database connection to the given URL. The driver should return "null" if it realizes it is the wrong kind of driver to connect to the given URL. This will be common, as when the JDBC driver manager is asked to connect to a given URL it passes the URL to each loaded driver in turn.
The driver should throw an SQLException if it is the right driver to connect to the given URL but has trouble connecting to the database.

The java.util.Properties argument can be used to pass arbitrary string tag/value pairs as connection arguments. Normally at least "user" and "password" properties should be included in the Properties object. (the

plugin passes empty Properties object)

url - the URL of the database to which to connect

Returns a Connection object that represents a connection to the URL
Throws SQLException - if a database access error occurs

### 4.5.4 Constructor(vm as JavaVMMBS, driverclass as string)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Creates a new java database object.
**Example:**

```
Var vm as JavaVMMBS // your VM
Var db as JavaDatabaseMBS

db=new JavaDatabaseMBS(vm,"com.mysql.jdbc.Driver")
```

**Notes:** The driverclass is the name of the main class of the jdbc driver.

### 4.5.5 getConnection(url as string) as JavaConnectionMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Attempts to establish a connection to the given database URL.
**Example:**

```
Var d as JavaDatabaseMBS
Var c as JavaConnectionMBS

// get database

// connect to Oracle database using service name:
c=d.getConnection("jdbc:oracle:thin:@//192.168.10.20:1521/adbprod","user","pw")

// connect with SID:
c=d.getConnection("jdbc:oracle:thin:@192.168.10.20:1521:adbprod","user","pw")
```

**Notes:** The DriverManager attempts to select an appropriate driver from the set of registered JDBC drivers.

Parameters:

url - a database url of the form jdbc:subprotocol:subname
Returns:
a connection to the URL
See also:

- 4.5.6 getConnection(url as string, username as string, password as string) as JavaConnectionMBS 159

### 4.5.6 getConnection(url as string, username as string, password as string) as JavaConnectionMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Attempts to establish a connection to the given database URL.
**Example:**

```
Var d as JavaDatabaseMBS
Var c as JavaConnectionMBS

// get database

// connect to Oracle database using service name:
c=d.getConnection("jdbc:oracle:thin:@//192.168.10.20:1521/adbprod","user","pw")

// connect with SID:
c=d.getConnection("jdbc:oracle:thin:@192.168.10.20:1521:adbprod","user","pw")
```

**Notes:** The DriverManager attempts to select an appropriate driver from the set of registered JDBC drivers.

Parameters:
url - a database url of the form jdbc:subprotocol:subname
user - the database user on whose behalf the connection is being made
password - the user's password

Returns:
a connection to the URL
See also:

- 4.5.5 getConnection(url as string) as JavaConnectionMBS 158

### 4.5.7 IsDriverLoaded as Boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Whether a driver has been loaded.

### 4.5.8 println(message as string)

Plugin Version: 12.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Prints a message to the current JDBC log stream.

### 4.5.9 Properties

### 4.5.10 LoginTimeout as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The maximum time in seconds that a driver will wait while attempting to connect to a database.
**Notes:** The login time limit in seconds.
(Read and Write computed property)

# 4.6 class JavaDatabaseMetaDataMBS

## 4.6.1 class JavaDatabaseMetaDataMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Comprehensive information about the database as a whole.
**Notes:** This interface is implemented by driver vendors to let users know the capabilities of a Database Management System (DBMS) in combination with the driver based on JDBCTM technology ("JDBC driver") that is used with it. Different relational DBMSs often support different features, implement features in different ways, and use different data types. In addition, a driver may implement a feature on top of what the DBMS offers. Information returned by methods in this interface applies to the capabilities of a particular driver and a particular DBMS working together. Note that as used in this documentation, the term "database" is used generically to refer to both the driver and DBMS.

A user for this interface is commonly a tool that needs to discover how to deal with the underlying DBMS. This is especially true for applications that are intended to be used with more than one DBMS. For example, a tool might use the method getTypeInfo to find out what data types can be used in a CREATE TABLE statement. Or a user might call the method supportsCorrelatedSubqueries to see if it is possible to use a correlated subquery or supportsBatchUpdates to see if it is possible to use batch updates.

Some DatabaseMetaData methods return lists of information in the form of ResultSet objects. Regular ResultSet methods, such as getString and getInt, can be used to retrieve the data from these ResultSet objects. If a given form of metadata is not available, the ResultSet getter methods throw an SQLException.

Some DatabaseMetaData methods take arguments that are String patterns. These arguments all have names such as fooPattern. Within a pattern String, "%" means match any substring of 0 or more characters, and "_" means match any one character. Only metadata entries matching the search pattern are returned. If a search pattern argument is set to null, that argument's criterion will be dropped from the search.

A method that gets information about a feature that the driver does not support will throw an SQLException. In the case of methods that return a ResultSet object, either a ResultSet object (which may be empty) is returned or an SQLException is thrown.
Subclass of the JavaObjectMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

## 4.6.2 Methods

## 4.6.3 allProceduresAreCallable as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether the current user can call all the procedures returned by the method getPro-

cedures.
**Notes:** Returns true if so; false otherwise

### 4.6.4 allTablesAreSelectable as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether the current user can use all the tables returned by the method getTables in a SELECT statement.
**Notes:** Returns true if so; false otherwise

### 4.6.5 attributeNoNulls as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that NULL values might not be allowed.
**Notes:** A possible value for the column NULLABLE in the ResultSet object returned by the method getAttributes.

### 4.6.6 attributeNullable as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that NULL values are definitely allowed.
**Notes:** A possible value for the column NULLABLE in the ResultSet object returned by the method getAttributes.

### 4.6.7 attributeNullableUnknown as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that whether NULL values are allowed is not known.
**Notes:** A possible value for the column NULLABLE in the ResultSet object returned by the method getAttributes.

### 4.6.8 bestRowNotPseudo as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the best row identifier is NOT a pseudo column.
**Notes:** A possible value for the column PSEUDO_COLUMN in the ResultSet object returned by the method getBestRowIdentifier.

### 4.6.9 bestRowPseudo as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the best row identifier is a pseudo column.
**Notes:** A possible value for the column PSEUDO_COLUMN in the ResultSet object returned by the method getBestRowIdentifier.

### 4.6.10 bestRowSession as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the scope of the best row identifier is the remainder of the current session.
**Notes:** A possible value for the column SCOPE in the ResultSet object returned by the method getBestRowIdentifier.

### 4.6.11 bestRowTemporary as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the scope of the best row identifier is very temporary, lasting only while the row is being used.
**Notes:** A possible value for the column SCOPE in the ResultSet object returned by the method getBestRowIdentifier.

### 4.6.12 bestRowTransaction as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the scope of the best row identifier is the remainder of the current transaction.
**Notes:** A possible value for the column SCOPE in the ResultSet object returned by the method getBestRowIdentifier.

### 4.6.13 bestRowUnknown as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the best row identifier may or may not be a pseudo column.
**Notes:** A possible value for the column PSEUDO_COLUMN in the ResultSet object returned by the method getBestRowIdentifier.

### 4.6.14 columnNoNulls as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the column might not allow NULL values.
**Notes:** A possible value for the column NULLABLE in the ResultSet returned by the method getColumns.

### 4.6.15 columnNullable as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the column definitely allows NULL values.
**Notes:** A possible value for the column NULLABLE in the ResultSet returned by the method getColumns.

### 4.6.16 columnNullableUnknown as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the nullability of columns is unknown.
**Notes:** A possible value for the column NULLABLE in the ResultSet returned by the method getColumns.

### 4.6.17 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 4.6.18 dataDefinitionCausesTransactionCommit as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a data definition statement within a transaction forces the transaction to commit.

### 4.6.19 dataDefinitionIgnoredInTransactions as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database ignores a data definition statement within a transaction.

### 4.6.20 deletesAreDetected(type as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether or not a visible row delete can be detected by calling the method ResultSet.rowDeleted.
**Notes:** If the method deletesAreDetected returns false, it means that deleted rows are removed from the result set.

Parameters:
type - the ResultSet type; one of ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE

Returns:
true if deletes are detected by the given result set type; false otherwise

### 4.6.21 doesMaxRowSizeIncludeBlobs as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether the return value for the method getMaxRowSize includes the SQL data types LONGVARCHAR and LONGVARBINARY.
**Notes:** Returns true if so; false otherwise.

### 4.6.22 getAttributes(catalog as string, schemaPattern as string, typeNamePattern as string, attributeNamePattern as string) as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of the given attribute of the given type for a user-defined type (UDT) that is available in the given schema and catalog.

**Notes:** Descriptions are returned only for attributes of UDTs matching the catalog, schema, type, and attribute name criteria. They are ordered by TYPE_SCHEM, TYPE_NAME and ORDINAL_POSITION. This description does not contain inherited attributes.

The ResultSet object that is returned has the following columns:

TYPE_CAT String =>type catalog (may be null)
TYPE_SCHEM String =>type schema (may be null)
TYPE_NAME String =>type name
ATTR_NAME String =>attribute name
DATA_TYPE int =>attribute type SQL type from java.sql.Types
ATTR_TYPE_NAME String =>Data source dependent type name. For a UDT, the type name is fully qualified. For a REF, the type name is fully qualified and represents the target type of the reference type.
ATTR_SIZE int =>column size. For char or date types this is the maximum number of characters; for numeric or decimal types this is precision.
DECIMAL_DIGITS int =>the number of fractional digits
NUM_PREC_RADIX int =>Radix (typically either 10 or 2)
NULLABLE int =>whether NULL is allowed
attributeNoNulls - might not allow NULL values
attributeNullable - definitely allows NULL values
attributeNullableUnknown - nullability unknown
REMARKS String =>comment describing column (may be null)
ATTR_DEF String =>default value (may be null)
SQL_DATA_TYPE int =>unused
SQL_DATETIME_SUB int =>unused
CHAR_OCTET_LENGTH int =>for char types the maximum number of bytes in the column
ORDINAL_POSITION int =>index of column in table (starting at 1)
IS_NULLABLE String =>"false" means column definitely does not allow NULL values; "true" means the column might allow NULL values. An empty string means unknown.
SCOPE_CATALOG String =>catalog of table that is the scope of a reference attribute (null if DATA_TYPE isn't REF)
SCOPE_SCHEMA String =>schema of table that is the scope of a reference attribute (null if DATA_TYPE isn't REF)
SCOPE_TABLE String =>table name that is the scope of a reference attribute (null if the DATA_TYPE isn't REF)
SOURCE_DATA_TYPE short =>source type of a distinct type or user-generated Ref type,SQL type from java.sql.Types (null if DATA_TYPE isn't DISTINCT or user-generated REF)
Parameters:
catalog - a catalog name; must match the catalog name as it is stored in the database; "" retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
schemaPattern - a schema name pattern; must match the schema name as it is stored in the database; "" retrieves those without a schema; null means that the schema name should not be used to narrow the search
typeNamePattern - a type name pattern; must match the type name as it is stored in the database
attributeNamePattern - an attribute name pattern; must match the attribute name as it is declared in the database

Returns:
a ResultSet object in which each row is an attribute description

Throws:
SQLException - if a database access error occurs

### 4.6.23 getCatalogs as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the catalog names available in this database. The results are ordered by catalog name.
**Notes:** The catalog column is:

TABLE_CAT String =>catalog name

Returns:
a ResultSet object in which each row has a single String column that is a catalog name

Throws:
SQLException - if a database access error occurs

### 4.6.24 getCatalogSeparator as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the String that this database uses as the separator between a catalog and table name.

### 4.6.25 getCatalogTerm as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the database vendor's preferred term for "catalog".

### 4.6.26 getColumnPrivileges(catalog as string, schema as string, table as string, columnNamePattern as string) as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of the access rights for a table's columns.
**Notes:** Only privileges matching the column name criteria are returned. They are ordered by COLUMN_NAME and PRIVILEGE.

Each privilige description has the following columns:

TABLE_CAT String =>table catalog (may be null)
TABLE_SCHEM String =>table schema (may be null)
TABLE_NAME String =>table name
COLUMN_NAME String =>column name
GRANTOR =>grantor of access (may be null)
GRANTEE String =>grantee of access
PRIVILEGE String =>name of access (SELECT, INSERT, UPDATE, REFRENCES, ...)
IS_GRANTABLE String =>"true" if grantee is permitted to grant to others; "false" if not; null if unknown
Parameters:
catalog - a catalog name; must match the catalog name as it is stored in the database; "" retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
schema - a schema name; must match the schema name as it is stored in the database; "" retrieves those without a schema; null means that the schema name should not be used to narrow the search
table - a table name; must match the table name as it is stored in the database
columnNamePattern - a column name pattern; must match the column name as it is stored in the database

Returns:
ResultSet - each row is a column privilege description

Throws:
SQLException - if a database access error occurs

### 4.6.27 getColumns(catalog as string, schemaPattern as string, tableNamePattern as string, columnNamePattern as string) as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of table columns available in the specified catalog.
**Notes:** Only column descriptions matching the catalog, schema, table and column name criteria are returned. They are ordered by TABLE_SCHEM, TABLE_NAME, and ORDINAL_POSITION.

Each column description has the following columns:

TABLE_CAT String =>table catalog (may be null)
TABLE_SCHEM String =>table schema (may be null)
TABLE_NAME String =>table name

COLUMN_NAME String =>column name
DATA_TYPE int =>SQL type from java.sql.Types
TYPE_NAME String =>Data source dependent type name, for a UDT the type name is fully qualified
COLUMN_SIZE int =>column size. For char or date types this is the maximum number of characters, for numeric or decimal types this is precision.
BUFFER_LENGTH is not used.
DECIMAL_DIGITS int =>the number of fractional digits
NUM_PREC_RADIX int =>Radix (typically either 10 or 2)
NULLABLE int =>is NULL allowed.
columnNoNulls - might not allow NULL values
columnNullable - definitely allows NULL values
columnNullableUnknown - nullability unknown
REMARKS String =>comment describing column (may be null)
COLUMN_DEF String =>default value (may be null)
SQL_DATA_TYPE int =>unused
SQL_DATETIME_SUB int =>unused
CHAR_OCTET_LENGTH int =>for char types the maximum number of bytes in the column
ORDINAL_POSITION int =>index of column in table (starting at 1)
IS_NULLABLE String =>"false" means column definitely does not allow NULL values; "true" means the column might allow NULL values. An empty string means nobody knows.
SCOPE_CATLOG String =>catalog of table that is the scope of a reference attribute (null if DATA_TYPE isn't REF)
SCOPE_SCHEMA String =>schema of table that is the scope of a reference attribute (null if the DATA_TYPE isn't REF)
SCOPE_TABLE String =>table name that this the scope of a reference attribute (null if the DATA_TYPE isn't REF)
SOURCE_DATA_TYPE short =>source type of a distinct type or user-generated Ref type, SQL type from java.sql.Types (null if DATA_TYPE isn't DISTINCT or user-generated REF)
Parameters:
catalog - a catalog name; must match the catalog name as it is stored in the database; "" retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
schemaPattern - a schema name pattern; must match the schema name as it is stored in the database; "" retrieves those without a schema; null means that the schema name should not be used to narrow the search
tableNamePattern - a table name pattern; must match the table name as it is stored in the database
columnNamePattern - a column name pattern; must match the column name as it is stored in the database

Returns:
ResultSet - each row is a column description

Throws:
SQLException - if a database access error occurs

### 4.6.28 getConnection as JavaConnectionMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the connection that produced this metadata object.

### 4.6.29 getCrossReference(primaryCatalog as string, primarySchema as string, primaryTable as string, foreignCatalog as string, foreignSchema as string, foreignTable as string) as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of the foreign key columns in the given foreign key table that reference the primary key columns of the given primary key table (describe how one table imports another's key).
**Notes:** This should normally return a single foreign key/primary key pair because most tables import a foreign key from a table only once. They are ordered by FKTABLE_CAT, FKTABLE_SCHEM, FKTABLE_NAME, and KEY_SEQ.
Each foreign key column description has the following columns:

PKTABLE_CAT String =>primary key table catalog (may be null)
PKTABLE_SCHEM String =>primary key table schema (may be null)
PKTABLE_NAME String =>primary key table name
PKCOLUMN_NAME String =>primary key column name
FKTABLE_CAT String =>foreign key table catalog (may be null) being exported (may be null)
FKTABLE_SCHEM String =>foreign key table schema (may be null) being exported (may be null)
FKTABLE_NAME String =>foreign key table name being exported
FKCOLUMN_NAME String =>foreign key column name being exported
KEY_SEQ short =>sequence number within foreign key
UPDATE_RULE short =>What happens to foreign key when primary is updated:
importedNoAction - do not allow update of primary key if it has been imported
importedKeyCascade - change imported key to agree with primary key update
importedKeySetNull - change imported key to NULL if its primary key has been updated
importedKeySetDefault - change imported key to default values if its primary key has been updated
importedKeyRestrict - same as importedKeyNoAction (for ODBC 2.x compatibility)
DELETE_RULE short =>What happens to the foreign key when primary is deleted.
importedKeyNoAction - do not allow delete of primary key if it has been imported
importedKeyCascade - delete rows that import a deleted key
importedKeySetNull - change imported key to NULL if its primary key has been deleted
importedKeyRestrict - same as importedKeyNoAction (for ODBC 2.x compatibility)
importedKeySetDefault - change imported key to default if its primary key has been deleted
FK_NAME String =>foreign key name (may be null)
PK_NAME String =>primary key name (may be null)
DEFERRABILITY short =>can the evaluation of foreign key constraints be deferred until commit
importedKeyInitiallyDeferred - see SQL92 for definition
importedKeyInitiallyImmediate - see SQL92 for definition
importedKeyNotDeferrable - see SQL92 for definition
Parameters:
primaryCatalog - a catalog name; must match the catalog name as it is stored in the database; "" retrieves those without a catalog; null means drop catalog name from the selection criteria

primarySchema - a schema name; must match the schema name as it is stored in the database; "" retrieves those without a schema; null means drop schema name from the selection criteria
primaryTable - the name of the table that exports the key; must match the table name as it is stored in the database
foreignCatalog - a catalog name; must match the catalog name as it is stored in the database; "" retrieves those without a catalog; null means drop catalog name from the selection criteria
foreignSchema - a schema name; must match the schema name as it is stored in the database; "" retrieves those without a schema; null means drop schema name from the selection criteria
foreignTable - the name of the table that imports the key; must match the table name as it is stored in the database
Returns:
ResultSet - each row is a foreign key column description
Throws:
SQLException - if a database access error occurs

### 4.6.30 getDatabaseMajorVersion as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the major version number of the underlying database.

### 4.6.31 getDatabaseMinorVersion as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the minor version number of the underlying database.

### 4.6.32 getDatabaseProductName as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the name of this database product.

### 4.6.33 getDatabaseProductVersion as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the version number of this database product.

### 4.6.34 getDefaultTransactionIsolation as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves this database's default transaction isolation level.
**Notes:** The possible values are defined in java.sql.Connection.

### 4.6.35 getDriverMajorVersion as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves this JDBC driver's major version number.

### 4.6.36 getDriverMinorVersion as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves this JDBC driver's minor version number.

### 4.6.37 getDriverName as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the name of this JDBC driver.

### 4.6.38 getDriverVersion as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the version number of this JDBC driver as a String.

### 4.6.39 getExportedKeys(catalog as string, schema as string, table as string) as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of the foreign key columns that reference the given table's primary key columns (the foreign keys exported by a table).
**Notes:** They are ordered by FKTABLE_CAT, FKTABLE_SCHEM, FKTABLE_NAME, and KEY_SEQ.

Each foreign key column description has the following columns:

PKTABLE_CAT String =>primary key table catalog (may be null)
PKTABLE_SCHEM String =>primary key table schema (may be null)
PKTABLE_NAME String =>primary key table name
PKCOLUMN_NAME String =>primary key column name
FKTABLE_CAT String =>foreign key table catalog (may be null) being exported (may be null)
FKTABLE_SCHEM String =>foreign key table schema (may be null) being exported (may be null)
FKTABLE_NAME String =>foreign key table name being exported
FKCOLUMN_NAME String =>foreign key column name being exported
KEY_SEQ short =>sequence number within foreign key
UPDATE_RULE short =>What happens to foreign key when primary is updated:
importedNoAction - do not allow update of primary key if it has been imported
importedKeyCascade - change imported key to agree with primary key update
importedKeySetNull - change imported key to NULL if its primary key has been updated
importedKeySetDefault - change imported key to default values if its primary key has been updated
importedKeyRestrict - same as importedKeyNoAction (for ODBC 2.x compatibility)
DELETE_RULE short =>What happens to the foreign key when primary is deleted.
importedKeyNoAction - do not allow delete of primary key if it has been imported
importedKeyCascade - delete rows that import a deleted key
importedKeySetNull - change imported key to NULL if its primary key has been deleted
importedKeyRestrict - same as importedKeyNoAction (for ODBC 2.x compatibility)
importedKeySetDefault - change imported key to default if its primary key has been deleted
FK_NAME String =>foreign key name (may be null)
PK_NAME String =>primary key name (may be null)
DEFERRABILITY short =>can the evaluation of foreign key constraints be deferred until commit
importedKeyInitiallyDeferred - see SQL92 for definition
importedKeyInitiallyImmediate - see SQL92 for definition
importedKeyNotDeferrable - see SQL92 for definition

Parameters:
catalog - a catalog name; must match the catalog name as it is stored in this database; "" retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
schema - a schema name; must match the schema name as it is stored in the database; "" retrieves those without a schema; null means that the schema name should not be used to narrow the search
table - a table name; must match the table name as it is stored in this database

Returns:
a ResultSet object in which each row is a foreign key column description

Throws:
SQLException - if a database access error occurs

### 4.6.40 getExtraNameCharacters as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves all the "extra" characters that can be used in unquoted identifier names (those beyond a-z, A-Z, 0-9 and _).

### 4.6.41 getIdentifierQuoteString as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the string used to quote SQL identifiers. This method returns a space " " if identifier quoting is not supported.

### 4.6.42 getImportedKeys(catalog as string, schema as string, table as string) as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of the primary key columns that are referenced by a table's foreign key columns (the primary keys imported by a table).
**Notes:** They are ordered by PKTABLE_CAT, PKTABLE_SCHEM, PKTABLE_NAME, and KEY_SEQ. Each primary key column description has the following columns:

PKTABLE_CAT String =>primary key table catalog being imported (may be null)
PKTABLE_SCHEM String =>primary key table schema being imported (may be null)
PKTABLE_NAME String =>primary key table name being imported
PKCOLUMN_NAME String =>primary key column name being imported
FKTABLE_CAT String =>foreign key table catalog (may be null)
FKTABLE_SCHEM String =>foreign key table schema (may be null)
FKTABLE_NAME String =>foreign key table name
FKCOLUMN_NAME String =>foreign key column name
KEY_SEQ short =>sequence number within a foreign key
UPDATE_RULE short =>What happens to a foreign key when the primary key is updated:
importedNoAction - do not allow update of primary key if it has been imported
importedKeyCascade - change imported key to agree with primary key update
importedKeySetNull - change imported key to NULL if its primary key has been updated
importedKeySetDefault - change imported key to default values if its primary key has been updated
importedKeyRestrict - same as importedKeyNoAction (for ODBC 2.x compatibility)
DELETE_RULE short =>What happens to the foreign key when primary is deleted.
importedKeyNoAction - do not allow delete of primary key if it has been imported
importedKeyCascade - delete rows that import a deleted key
importedKeySetNull - change imported key to NULL if its primary key has been deleted
importedKeyRestrict - same as importedKeyNoAction (for ODBC 2.x compatibility)

importedKeySetDefault - change imported key to default if its primary key has been deleted
FK_NAME String =>foreign key name (may be null)
PK_NAME String =>primary key name (may be null)
DEFERRABILITY short =>can the evaluation of foreign key constraints be deferred until commit
importedKeyInitiallyDeferred - see SQL92 for definition
importedKeyInitiallyImmediate - see SQL92 for definition
importedKeyNotDeferrable - see SQL92 for definition
Parameters:
catalog - a catalog name; must match the catalog name as it is stored in the database; "" retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
schema - a schema name; must match the schema name as it is stored in the database; "" retrieves those without a schema; null means that the schema name should not be used to narrow the search
table - a table name; must match the table name as it is stored in the database

Returns:
ResultSet - each row is a primary key column description

Throws:
SQLException - if a database access error occurs

### 4.6.43 getJDBCMajorVersion as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the major JDBC version number for this driver.

### 4.6.44 getJDBCMinorVersion as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the minor JDBC version number for this driver.

### 4.6.45 getMaxBinaryLiteralLength as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of hex characters this database allows in an inline binary literal.
**Notes:** Returns the maximum length (in hex characters) for a binary literal; a result of zero means that there is no limit or the limit is not known

### 4.6.46 getMaxCatalogNameLength as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of characters that this database allows in a catalog name.
**Notes:** Returns the maximum number of characters allowed in a catalog name; a result of zero means that there is no limit or the limit is not known

### 4.6.47 getMaxCharLiteralLength as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of characters this database allows for a character literal.
**Notes:** Returns the maximum number of characters allowed for a character literal; a result of zero means that there is no limit or the limit is not known

### 4.6.48 getMaxColumnNameLength as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of characters this database allows for a column name.
**Notes:** Returns the maximum number of characters allowed for a column name; a result of zero means that there is no limit or the limit is not known

### 4.6.49 getMaxColumnsInGroupBy as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of columns this database allows in a GROUP BY clause.
**Notes:** Returns the maximum number of columns allowed; a result of zero means that there is no limit or the limit is not known

### 4.6.50 getMaxColumnsInIndex as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of columns this database allows in an index.
**Notes:** Returns the maximum number of columns allowed; a result of zero means that there is no limit or the limit is not known

### 4.6.51 getMaxColumnsInOrderBy as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of columns this database allows in an ORDER BY clause.
**Notes:** Returns the maximum number of columns allowed; a result of zero means that there is no limit or the limit is not known

### 4.6.52 getMaxColumnsInSelect as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of columns this database allows in a SELECT list.
**Notes:** Returns the maximum number of columns allowed; a result of zero means that there is no limit or the limit is not known

### 4.6.53 getMaxColumnsInTable as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of columns this database allows in a table.
**Notes:** Returns the maximum number of columns allowed; a result of zero means that there is no limit or the limit is not known

### 4.6.54 getMaxConnections as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of concurrent connections to this database that are possible.
**Notes:** Returns the maximum number of active connections possible at one time; a result of zero means that there is no limit or the limit is not known

### 4.6.55 getMaxCursorNameLength as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of characters that this database allows in a cursor name.
**Notes:** Returns the maximum number of characters allowed in a cursor name; a result of zero means that there is no limit or the limit is not known

### 4.6.56 getMaxIndexLength as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of bytes this database allows for an index, including all of the parts of the index.
**Notes:** Returns the maximum number of bytes allowed; this limit includes the composite of all the constituent parts of the index; a result of zero means that there is no limit or the limit is not known

### 4.6.57 getMaxProcedureNameLength as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of characters that this database allows in a procedure name.
**Notes:** Returns the maximum number of characters allowed in a procedure name; a result of zero means that there is no limit or the limit is not known

### 4.6.58 getMaxRowSize as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of bytes this database allows in a single row.
**Notes:** Returns the maximum number of bytes allowed for a row; a result of zero means that there is no limit or the limit is not known

### 4.6.59 getMaxSchemaNameLength as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of characters that this database allows in a schema name.
**Notes:** Returns the maximum number of characters allowed in a schema name; a result of zero means that there is no limit or the limit is not known

### 4.6.60 getMaxStatementLength as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of characters this database allows in an SQL statement.
**Notes:** Returns the maximum number of characters allowed for an SQL statement; a result of zero means that there is no limit or the limit is not known

### 4.6.61 getMaxStatements as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of active statements to this database that can be open at the same time.
**Notes:** Returns the maximum number of statements that can be open at one time; a result of zero means that there is no limit or the limit is not known

### 4.6.62 getMaxTableNameLength as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of characters this database allows in a table name.
**Notes:** Returns the maximum number of characters allowed for a table name; a result of zero means that there is no limit or the limit is not known

### 4.6.63 getMaxTablesInSelect as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of tables this database allows in a SELECT statement.
**Notes:** Returns the maximum number of tables allowed in a SELECT statement; a result of zero means that there is no limit or the limit is not known

### 4.6.64 getMaxUserNameLength as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the maximum number of characters this database allows in a user name.
**Notes:** Returns the maximum number of characters allowed for a user name; a result of zero means that there is no limit or the limit is not known

### 4.6.65 getNumericFunctions as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a comma-separated list of math functions available with this database.
**Notes:** These are the Open /Open CLI math function names used in the JDBC function escape clause.

### 4.6.66 getPrimaryKeys(catalog as string, schema as string, table as string) as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of the given table's primary key columns. They are ordered by COLUMN_NAME.
**Notes:** Each primary key column description has the following columns:

TABLE_CAT String =>table catalog (may be null)
TABLE_SCHEM String =>table schema (may be null)
TABLE_NAME String =>table name
COLUMN_NAME String =>column name
KEY_SEQ short =>sequence number within primary key
PK_NAME String =>primary key name (may be null)
Parameters:
catalog - a catalog name; must match the catalog name as it is stored in the database; "" retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
schema - a schema name; must match the schema name as it is stored in the database; "" retrieves those without a schema; null means that the schema name should not be used to narrow the search
table - a table name; must match the table name as it is stored in the database

Returns:
ResultSet - each row is a primary key column description

### 4.6.67 getProcedureColumns(catalog as string, schemaPattern as string, procedureNamePattern as string, columnNamePattern as string) as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of the given catalog's stored procedure parameter and result columns.
**Notes:** Only descriptions matching the schema, procedure and parameter name criteria are returned. They are ordered by PROCEDURE_SCHEM and PROCEDURE_NAME. Within this, the return value, if any, is first. Next are the parameter descriptions in call order. The column descriptions follow in column number order.

Each row in the ResultSet is a parameter description or column description with the following fields:

PROCEDURE_CAT String =>procedure catalog (may be null)
PROCEDURE_SCHEM String =>procedure schema (may be null)
PROCEDURE_NAME String =>procedure name
COLUMN_NAME String =>column/parameter name
COLUMN_TYPE Short =>kind of column/parameter:

procedureColumnUnknown - nobody knows
procedureColumnIn - IN parameter
procedureColumnInOut - INOUT parameter
procedureColumnOut - OUT parameter
procedureColumnReturn - procedure return value
procedureColumnResult - result column in ResultSet
DATA_TYPE int =>SQL type from java.sql.Types
TYPE_NAME String =>SQL type name, for a UDT type the type name is fully qualified
PRECISION int =>precision
LENGTH int =>length in bytes of data
SCALE short =>scale
RADIX short =>radix
NULLABLE short =>can it contain NULL.
procedureNoNulls - does not allow NULL values
procedureNullable - allows NULL values
procedureNullableUnknown - nullability unknown
REMARKS String =>comment describing parameter/column
Note: Some databases may not return the column descriptions for a procedure. Additional columns beyond REMARKS can be defined by the database.

Parameters:
catalog - a catalog name; must match the catalog name as it is stored in the database; "" retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
schemaPattern - a schema name pattern; must match the schema name as it is stored in the database; "" retrieves those without a schema; null means that the schema name should not be used to narrow the search
procedureNamePattern - a procedure name pattern; must match the procedure name as it is stored in the database
columnNamePattern - a column name pattern; must match the column name as it is stored in the database
Returns:
ResultSet - each row describes a stored procedure parameter or column
Throws:
SQLException - if a database access error occurs

### 4.6.68 getProcedures(catalog as string, schemaPattern as string, procedureNamePattern as string) as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of the stored procedures available in the given catalog.
**Notes:** Only procedure descriptions matching the schema and procedure name criteria are returned. They are ordered by PROCEDURE_SCHEM and PROCEDURE_NAME.

Each procedure description has the the following columns:

PROCEDURE_CAT String =>procedure catalog (may be null)
PROCEDURE_SCHEM String =>procedure schema (may be null)
PROCEDURE_NAME String =>procedure name
reserved for future use
reserved for future use
reserved for future use
REMARKS String =>explanatory comment on the procedure
PROCEDURE_TYPE short =>kind of procedure:
procedureResultUnknown - May return a result
procedureNoResult - Does not return a result
procedureReturnsResult - Returns a result
Parameters:
catalog - a catalog name; must match the catalog name as it is stored in the database; "" retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
schemaPattern - a schema name pattern; must match the schema name as it is stored in the database; "" retrieves those without a schema; null means that the schema name should not be used to narrow the search
procedureNamePattern - a procedure name pattern; must match the procedure name as it is stored in the database

Returns:
ResultSet - each row is a procedure description

Throws:
SQLException - if a database access error occurs

### 4.6.69 getProcedureTerm as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the database vendor's preferred term for "procedure".

### 4.6.70 getResultSetHoldability as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the default holdability of this ResultSet object.
**Notes:** Returns:
the default holdability; either ResultSet.HOLD_CURSORS_OVER_COMMIT or ResultSet.CLOSE_CURSORS_AT_COMMIT

### 4.6.71 getSchemas as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the schema names available in this database.
**Notes:** The results are ordered by schema name.

The schema column is:

TABLE_SCHEM String =>schema name
TABLE_CATALOG String =>catalog name (may be null)
Returns:
a ResultSet object in which each row is a schema decription

Throws:
SQLException - if a database access error occurs

### 4.6.72 getSchemaTerm as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the database vendor's preferred term for "schema".

### 4.6.73 getSearchStringEscape as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the string that can be used to escape wildcard characters.
**Notes:** This is the string that can be used to escape '_' or '%' in the catalog search parameters that are a pattern (and therefore use one of the wildcard characters).
The '_' character represents any single character; the '%' character represents any sequence of zero or more characters.

### 4.6.74 getSQLKeywords as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a comma-separated list of all of this database's SQL keywords that are NOT also SQL92 keywords.

### 4.6.75 getSQLStateType as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates whether the SQLSTATE returned by SQLException.getSQLState is X/Open (now known as Open Group) SQL CLI or SQL99.
**Notes:** Returns:
the type of SQLSTATE; one of: sqlStateXOpen or sqlStateSQL99

### 4.6.76 getStringFunctions as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a comma-separated list of string functions available with this database.
**Notes:** These are the Open Group CLI string function names used in the JDBC function escape clause.

### 4.6.77 getSuperTables(catalog as string, schemaPattern as string, tableNamePattern as string) as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of the table hierarchies defined in a particular schema in this database.
**Notes:** Only supertable information for tables matching the catalog, schema and table name are returned. The table name parameter may be a fully- qualified name, in which case, the catalog and schemaPattern parameters are ignored. If a table does not have a super table, it is not listed here. Supertables have to be defined in the same catalog and schema as the sub tables. Therefore, the type description does not need to include this information for the supertable.

Each type description has the following columns:

TABLE_CAT String =>the type's catalog (may be null)
TABLE_SCHEM String =>type's schema (may be null)
TABLE_NAME String =>type name
SUPERTABLE_NAME String =>the direct super type's name
Note: If the driver does not support type hierarchies, an empty result set is returned.

Parameters:
catalog - a catalog name; "" retrieves those without a catalog; null means drop catalog name from the selection criteria
schemaPattern - a schema name pattern; "" retrieves those without a schema
tableNamePattern - a table name pattern; may be a fully-qualified name

Returns:
a ResultSet object in which each row is a type description

Throws:
SQLException - if a database access error occurs

### 4.6.78 getSuperTypes(catalog as string, schemaPattern as string, typeNamePattern as string) as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of the user-defined type (UDT) hierarchies defined in a particular schema in this database.
**Notes:** Only the immediate super type/ sub type relationship is modeled.
Only supertype information for UDTs matching the catalog, schema, and type name is returned. The type name parameter may be a fully-qualified name. When the UDT name supplied is a fully-qualified name, the catalog and schemaPattern parameters are ignored.

If a UDT does not have a direct super type, it is not listed here. A row of the ResultSet object returned by this method describes the designated UDT and a direct supertype. A row has the following columns:

TYPE_CAT String =>the UDT's catalog (may be null)
TYPE_SCHEM String =>UDT's schema (may be null)
TYPE_NAME String =>type name of the UDT
SUPERTYPE_CAT String =>the direct super type's catalog (may be null)
SUPERTYPE_SCHEM String =>the direct super type's schema (may be null)
SUPERTYPE_NAME String =>the direct super type's name
Note: If the driver does not support type hierarchies, an empty result set is returned.

Parameters:
catalog - a catalog name; "" retrieves those without a catalog; null means drop catalog name from the selection criteria
schemaPattern - a schema name pattern; "" retrieves those without a schema
typeNamePattern - a UDT name pattern; may be a fully-qualified name

Returns:
a ResultSet object in which a row gives information about the designated UDT

Throws:
SQLException - if a database access error occurs

### 4.6.79 getSystemFunctions as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a comma-separated list of system functions available with this database.
**Notes:** These are the Open Group CLI system function names used in the JDBC function escape clause.

### 4.6.80 getTablePrivileges(catalog as string, schemaPattern as string, tableNamePattern as string) as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of the access rights for each table available in a catalog.
**Notes:** Note that a table privilege applies to one or more columns in the table. It would be wrong to assume that this privilege applies to all columns (this may be true for some systems but is not true for all.)
Only privileges matching the schema and table name criteria are returned. They are ordered by TABLE_SCHEM, TABLE_NAME, and PRIVILEGE.

Each privilige description has the following columns:

TABLE_CAT String =>table catalog (may be null)
TABLE_SCHEM String =>table schema (may be null)
TABLE_NAME String =>table name
GRANTOR =>grantor of access (may be null)
GRANTEE String =>grantee of access
PRIVILEGE String =>name of access (SELECT, INSERT, UPDATE, REFRENCES, ...)
IS_GRANTABLE String =>”true” if grantee is permitted to grant to others; ”false” if not; null if unknown
Parameters:
catalog - a catalog name; must match the catalog name as it is stored in the database; ”” retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
schemaPattern - a schema name pattern; must match the schema name as it is stored in the database; ”” retrieves those without a schema; null means that the schema name should not be used to narrow the search
tableNamePattern - a table name pattern; must match the table name as it is stored in the database

Returns:
ResultSet - each row is a table privilege description

Throws:
SQLException - if a database access error occurs

### 4.6.81 getTables(catalog as string, schemaPattern as string, tableNamePattern as string) as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of the tables available in the given catalog.
**Notes:** Only table descriptions matching the catalog, schema, table name and type criteria are returned. They are ordered by TABLE_TYPE, TABLE_SCHEM and TABLE_NAME.
Each table description has the following columns:

TABLE_CAT String =>table catalog (may be null)
TABLE_SCHEM String =>table schema (may be null)
TABLE_NAME String =>table name
TABLE_TYPE String =>table type. Typical types are "TABLE", "VIEW", "SYSTEM TABLE", "GLOBAL TEMPORARY", "LOCAL TEMPORARY", "ALIAS", "SYNONYM".
REMARKS String =>explanatory comment on the table
TYPE_CAT String =>the types catalog (may be null)
TYPE_SCHEM String =>the types schema (may be null)
TYPE_NAME String =>type name (may be null)
SELF_REFERENCING_COL_NAME String =>name of the designated "identifier" column of a typed table (may be null)
REF_GENERATION String =>specifies how values in SELF_REFERENCING_COL_NAME are created. Values are "SYSTEM", "USER", "DERIVED". (may be null)
Note: Some databases may not return information for all tables.

Parameters:
catalog - a catalog name; must match the catalog name as it is stored in the database; "" retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
schemaPattern - a schema name pattern; must match the schema name as it is stored in the database; "" retrieves those without a schema; null means that the schema name should not be used to narrow the search
tableNamePattern - a table name pattern; must match the table name as it is stored in the database
types - a list of table types to include (optionally)

Returns:
ResultSet - each row is a table description

Throws:
SQLException - if a database access error occurs
See also:

### 4.6.82 getTables(catalog as string, schemaPattern as string, tableNamePattern as string, types() as string) as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of the tables available in the given catalog.
**Notes:** Only table descriptions matching the catalog, schema, table name and type criteria are returned. They are ordered by TABLE_TYPE, TABLE_SCHEM and TABLE_NAME.
Each table description has the following columns:

TABLE_CAT String =>table catalog (may be null)
TABLE_SCHEM String =>table schema (may be null)
TABLE_NAME String =>table name
TABLE_TYPE String =>table type. Typical types are "TABLE", "VIEW", "SYSTEM TABLE", "GLOBAL TEMPORARY", "LOCAL TEMPORARY", "ALIAS", "SYNONYM".
REMARKS String =>explanatory comment on the table
TYPE_CAT String =>the types catalog (may be null)
TYPE_SCHEM String =>the types schema (may be null)
TYPE_NAME String =>type name (may be null)
SELF_REFERENCING_COL_NAME String =>name of the designated "identifier" column of a typed table (may be null)
REF_GENERATION String =>specifies how values in SELF_REFERENCING_COL_NAME are created. Values are "SYSTEM", "USER", "DERIVED". (may be null)
Note: Some databases may not return information for all tables.

Parameters:
catalog - a catalog name; must match the catalog name as it is stored in the database; "" retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
schemaPattern - a schema name pattern; must match the schema name as it is stored in the database; "" retrieves those without a schema; null means that the schema name should not be used to narrow the search
tableNamePattern - a table name pattern; must match the table name as it is stored in the database
types - a list of table types to include (optionally)

Returns:
ResultSet - each row is a table description

Throws:
SQLException - if a database access error occurs
See also:

- 4.6.81 getTables(catalog as string, schemaPattern as string, tableNamePattern as string) as JavaResultSetMBS

### 4.6.83 getTableTypes as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the table types available in this database.
**Notes:** The table type is:

TABLE_TYPE String =>table type. Typical types are "TABLE", "VIEW", "SYSTEM TABLE", "GLOBAL TEMPORARY", "LOCAL TEMPORARY", "ALIAS", "SYNONYM".

Returns:
a ResultSet object in which each row has a single String column that is a table type

Throws:
SQLException - if a database access error occurs

### 4.6.84 getTimeDateFunctions as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a comma-separated list of the time and date functions available with this database.

### 4.6.85 getTypeInfos as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of all the standard SQL types supported by this database.
**Notes:** They are ordered by DATA_TYPE and then by how closely the data type maps to the corresponding JDBC SQL type.

Each type description has the following columns:

TYPE_NAME String =>Type name
DATA_TYPE int =>SQL data type from java.sql.Types
PRECISION int =>maximum precision
LITERAL_PREFIX String =>prefix used to quote a literal (may be null)
LITERAL_SUFFIX String =>suffix used to quote a literal (may be null)
CREATE_PARAMS String =>parameters used in creating the type (may be null)
NULLABLE short =>can you use NULL for this type.
typeNoNulls - does not allow NULL values
typeNullable - allows NULL values
typeNullableUnknown - nullability unknown

CASE_SENSITIVE boolean=>is it case sensitive.
SEARCHABLE short =>can you use "WHERE" based on this type:
typePredNone - No support
typePredChar - Only supported with WHERE .. LIKE
typePredBasic - Supported except for WHERE .. LIKE
typeSearchable - Supported for all WHERE ..
UNSIGNED_ATTRIBUTE boolean =>is it unsigned.
FIXED_PREC_SCALE boolean =>can it be a money value.
AUTO_INCREMENT boolean =>can it be used for an auto-increment value.
LOCAL_TYPE_NAME String =>localized version of type name (may be null)
MINIMUM_SCALE short =>minimum scale supported
MAXIMUM_SCALE short =>maximum scale supported
SQL_DATA_TYPE int =>unused
SQL_DATETIME_SUB int =>unused
NUM_PREC_RADIX int =>usually 2 or 10

Returns:
a ResultSet object in which each row is an SQL type description

Throws:
SQLException - if a database access error occurs

This method is named getTypeInfo in Java and getTypeInfos in this plugin because Xojo has a global method and sees a conflict.

### 4.6.86 getURL as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the URL for this DBMS.
**Notes:** Returns the URL for this DBMS or "" if it cannot be generated.

### 4.6.87 getUserName as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the user name as known to this database.

### 4.6.88 getVersionColumns(catalog as string, schema as string, table as string) as JavaResultSetMBS

Plugin Version: 9.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a description of a table's columns that are automatically updated when any value in a row is updated. They are unordered.
**Notes:** Each column description has the following columns:

SCOPE short =>is not used
COLUMN_NAME String =>column name
DATA_TYPE int =>SQL data type from java.sql.Types
TYPE_NAME String =>Data source-dependent type name
COLUMN_SIZE int =>precision
BUFFER_LENGTH int =>length of column value in bytes
DECIMAL_DIGITS short =>scale
PSEUDO_COLUMN short =>whether this is pseudo column like an Oracle ROWID
versionColumnUnknown - may or may not be pseudo column
versionColumnNotPseudo - is NOT a pseudo column
versionColumnPseudo - is a pseudo column
Parameters:
catalog - a catalog name; must match the catalog name as it is stored in the database; "" retrieves those without a catalog; null means that the catalog name should not be used to narrow the search
schema - a schema name; must match the schema name as it is stored in the database; "" retrieves those without a schema; null means that the schema name should not be used to narrow the search
table - a table name; must match the table name as it is stored in the database

Returns:
a ResultSet object in which each row is a column description

### 4.6.89 importedKeyCascade as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** For the column UPDATE_RULE, indicates that when the primary key is updated, the foreign key (imported key) is changed to agree with it.
**Notes:** For the column DELETE_RULE, it indicates that when the primary key is deleted, rows that imported that key are deleted.
A possible value for the columns UPDATE_RULE and DELETE_RULE in the ResultSet objects returned by the methods getImportedKeys, getExportedKeys, and getCrossReference.

### 4.6.90 importedKeyInitiallyDeferred as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates deferrability. See SQL-92 for a definition.
**Notes:** A possible value for the column DEFERRABILITY in the ResultSet objects returned by the methods getImportedKeys, getExportedKeys, and getCrossReference.

### 4.6.91 importedKeyInitiallyImmediate as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates deferrability. See SQL-92 for a definition.
**Notes:** A possible value for the column DEFERRABILITY in the ResultSet objects returned by the methods getImportedKeys, getExportedKeys, and getCrossReference.

### 4.6.92 importedKeyNoAction as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** For the columns UPDATE_RULE and DELETE_RULE, indicates that if the primary key has been imported, it cannot be updated or deleted.
**Notes:** A possible value for the columns UPDATE_RULE and DELETE_RULE in the ResultSet objects returned by the methods getImportedKeys, getExportedKeys, and getCrossReference.

### 4.6.93 importedKeyNotDeferrable as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates deferrability. See SQL-92 for a definition.
**Notes:** A possible value for the column DEFERRABILITY in the ResultSet objects returned by the methods getImportedKeys, getExportedKeys, and getCrossReference.

### 4.6.94 importedKeyRestrict as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** For the column UPDATE_RULE, indicates that a primary key may not be updated if it has been imported by another table as a foreign key.
**Notes:** For the column DELETE_RULE, indicates that a primary key may not be deleted if it has been imported by another table as a foreign key.
A possible value for the columns UPDATE_RULE and DELETE_RULE in the ResultSet objects returned

by the methods getImportedKeys, getExportedKeys, and getCrossReference.

### 4.6.95 importedKeySetDefault as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** For the columns UPDATE_RULE and DELETE_RULE, indicates that if the primary key is updated or deleted, the foreign key (imported key) is set to the default value.
**Notes:** A possible value for the columns UPDATE_RULE and DELETE_RULE in the ResultSet objects returned by the methods getImportedKeys, getExportedKeys, and getCrossReference.

### 4.6.96 importedKeySetNull as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** For the columns UPDATE_RULE and DELETE_RULE, indicates that when the primary key is updated or deleted, the foreign key (imported key) is changed to NULL.
**Notes:** A possible value for the columns UPDATE_RULE and DELETE_RULE in the ResultSet objects returned by the methods getImportedKeys, getExportedKeys, and getCrossReference.

### 4.6.97 insertsAreDetected(type as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether or not a visible row insert can be detected by calling the method ResultSet.rowInserted.
**Notes:** Parameters:
type - the ResultSet type; one of ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE

Returns:
true if changes are detected by the specified result set type; false otherwise

### 4.6.98 isCatalogAtStart as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a catalog appears at the start of a fully qualified table name. If not, the catalog appears at the end.

### 4.6.99 isReadOnly as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database is in read-only mode.

### 4.6.100 locatorsUpdateCopy as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates whether updates made to a LOB are made on a copy or directly to the LOB.
**Notes:** Returns:
true if updates are made to a copy of the LOB; false if updates are made directly to the LOB

### 4.6.101 nullPlusNonNullIsNull as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports concatenations between NULL and non-NULL values being NULL.

### 4.6.102 nullsAreSortedAtEnd as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether NULL values are sorted at the end regardless of sort order.

### 4.6.103 nullsAreSortedAtStart as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether NULL values are sorted at the start regardless of sort order.

### 4.6.104 nullsAreSortedHigh as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether NULL values are sorted high.
**Notes:** Sorted high means that NULL values sort higher than any other value in a domain. In an ascending order, if this method returns true, NULL values will appear at the end. By contrast, the method nullsAreSortedAtEnd indicates whether NULL values are sorted at the end regardless of sort order.

### 4.6.105 nullsAreSortedLow as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether NULL values are sorted low.
**Notes:** Sorted low means that NULL values sort lower than any other value in a domain. In an ascending order, if this method returns true, NULL values will appear at the beginning. By contrast, the method nullsAreSortedAtStart indicates whether NULL values are sorted at the beginning regardless of sort order.

### 4.6.106 othersDeletesAreVisible(type as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether deletes made by others are visible.
**Notes:** Parameters:
type - the ResultSet type; one of ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE

Returns:
true if deletes made by others are visible for the given result set type; false otherwise

### 4.6.107 othersInsertsAreVisible(type as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether inserts made by others are visible.
**Notes:** Parameters:
type - the ResultSet type; one of ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE

Returns:
true if inserts made by others are visible for the given result set type; false otherwise

### 4.6.108 othersUpdatesAreVisible(type as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether updates made by others are visible.
**Notes:** Parameters:
type - the ResultSet type; one of ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE

Returns:
true if updates made by others are visible for the given result set type; false otherwise

### 4.6.109 ownDeletesAreVisible(type as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a result set's own deletes are visible.
**Notes:** type - the ResultSet type; one of ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE

Returns true if deletes are visible for the given result set type; false otherwise

### 4.6.110 ownInsertsAreVisible(type as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a result set's own inserts are visible.
**Notes:** type - the ResultSet type; one of ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE

Returns true if inserts are visible for the given result set type; false otherwise

### 4.6.111 ownUpdatesAreVisible(type as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether for the given type of ResultSet object, the result set's own updates are visible.
**Notes:** type - the ResultSet type; one of ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE

### 4.6.112 procedureColumnIn as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the column stores IN parameters.
**Notes:** A possible value for the column COLUMN_TYPE in the ResultSet returned by the method getProcedureColumns.

### 4.6.113 procedureColumnInOut as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the column stores INOUT parameters.
**Notes:** A possible value for the column COLUMN_TYPE in the ResultSet returned by the method getProcedureColumns.

### 4.6.114 procedureColumnOut as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the column stores OUT parameters.
**Notes:** A possible value for the column COLUMN_TYPE in the ResultSet returned by the method getProcedureColumns.

### 4.6.115 procedureColumnResult as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the column stores results.
**Notes:** A possible value for the column COLUMN_TYPE in the ResultSet returned by the method getProcedureColumns.

### 4.6.116 procedureColumnReturn as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the column stores return values.
**Notes:** A possible value for the column COLUMN_TYPE in the ResultSet returned by the method getProcedureColumns.

### 4.6.117 procedureColumnUnknown as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that type of the column is unknown.
**Notes:** A possible value for the column COLUMN_TYPE in the ResultSet returned by the method getProcedureColumns.

### 4.6.118 procedureNoNulls as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that NULL values are not allowed.
**Notes:** A possible value for the column NULLABLE in the ResultSet object returned by the method getProcedureColumns.

### 4.6.119 procedureNoResult as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the procedure does not return a result.
**Notes:** A possible value for column PROCEDURE_TYPE in the ResultSet object returned by the method getProcedures.

### 4.6.120 procedureNullable as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that NULL values are allowed.
**Notes:** A possible value for the column NULLABLE in the ResultSet object returned by the method getProcedureColumns.

### 4.6.121 procedureNullableUnknown as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that whether NULL values are allowed is unknown.
**Notes:** A possible value for the column NULLABLE in the ResultSet object returned by the method getProcedureColumns.

### 4.6.122 procedureResultUnknown as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that it is not known whether the procedure returns a result.
**Notes:** A possible value for column PROCEDURE_TYPE in the ResultSet object returned by the method getProcedures.

### 4.6.123 procedureReturnsResult as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the procedure returns a result.
**Notes:** A possible value for column PROCEDURE_TYPE in the ResultSet object returned by the method getProcedures.

### 4.6.124 sqlStateSQL99 as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the value is an SQL99 SQLSTATE value.
**Notes:** A possible return value for the method SQLException.getSQLState.

### 4.6.125 sqlStateXOpen as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the value is an X/Open (now know as Open Group) SQL CLI SQLSTATE value.
**Notes:** A possible return value for the method SQLException.getSQLState.

### 4.6.126 storesLowerCaseIdentifiers as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database treats mixed case unquoted SQL identifiers as case insensitive and stores them in lower case.

### 4.6.127 storesLowerCaseQuotedIdentifiers as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database treats mixed case quoted SQL identifiers as case insensitive and stores them in lower case.

### 4.6.128 storesMixedCaseIdentifiers as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database treats mixed case unquoted SQL identifiers as case insensitive and stores them in mixed case.

### 4.6.129 storesMixedCaseQuotedIdentifiers as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database treats mixed case quoted SQL identifiers as case sensitive and as a result stores them in mixed case.

### 4.6.130 storesUpperCaseIdentifiers as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database treats mixed case unquoted SQL identifiers as case insensitive and stores them in upper case.

### 4.6.131 storesUpperCaseQuotedIdentifiers as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database treats mixed case quoted SQL identifiers as case insensitive and stores them in upper case.

### 4.6.132 supportsAlterTableWithAddColumn as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports ALTER TABLE with add column.

### 4.6.133 supportsAlterTableWithDropColumn as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports ALTER TABLE with drop column.

### 4.6.134 supportsANSI92EntryLevelSQL as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports the ANSI92 entry level SQL grammar.

### 4.6.135 supportsANSI92FullSQL as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports the ANSI92 full SQL grammar supported.

### 4.6.136 supportsANSI92IntermediateSQL as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports the ANSI92 intermediate SQL grammar supported.

### 4.6.137 supportsBatchUpdates as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports batch updates.
**Notes:** Returns:
true if this database supports batch upcates; false otherwise

### 4.6.138 supportsCatalogsInDataManipulation as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a catalog name can be used in a data manipulation statement.

### 4.6.139 supportsCatalogsInIndexDefinitions as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a catalog name can be used in an index definition statement.

### 4.6.140 supportsCatalogsInPrivilegeDefinitions as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a catalog name can be used in an index definition statement.

### 4.6.141 supportsCatalogsInProcedureCalls as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a catalog name can be used in a procedure call statement.

### 4.6.142 supportsCatalogsInTableDefinitions as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a catalog name can be used in a table definition statement.

### 4.6.143 supportsColumnAliasing as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports column aliasing.
**Notes:** If so, the SQL AS clause can be used to provide names for computed columns or to provide alias names for columns as required.

### 4.6.144 supportsConvert as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports the CONVERT function between SQL types.
See also:

- 4.6.145 supportsConvert(fromType as Integer, toType as Integer) as boolean 203

### 4.6.145 supportsConvert(fromType as Integer, toType as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports the CONVERT for two given SQL types.
**Notes:** Parameters:
fromType - the type to convert from; one of the type codes from the class JavaDatabaseMBS
toType - the type to convert to; one of the type codes from the class JavaDatabaseMBS

Returns:
true if so; false otherwise
See also:

- 4.6.144 supportsConvert as boolean 203

### 4.6.146 supportsCoreSQLGrammar as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports the ODBC Core SQL grammar.

### 4.6.147 supportsCorrelatedSubqueries as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports correlated subqueries.

### 4.6.148 supportsDataDefinitionAndDataManipulationTransactions as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports both data definition and data manipulation statements within a transaction.

### 4.6.149 supportsDataManipulationTransactionsOnly as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports only data manipulation statements within a transaction.

### 4.6.150 supportsDifferentTableCorrelationNames as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether, when table correlation names are supported, they are restricted to being different from the names of the tables.

### 4.6.151 supportsExpressionsInOrderBy as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports expressions in ORDER BY lists.

### 4.6.152 supportsExtendedSQLGrammar as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports the ODBC Extended SQL grammar.

### 4.6.153 supportsFullOuterJoins as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports full nested outer joins.

### 4.6.154 supportsGetGeneratedKeys as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether auto-generated keys can be retrieved after a statement has been executed.
**Notes:** Returns:
true if auto-generated keys can be retrieved after a statement has executed; false otherwise

### 4.6.155 supportsGroupBy as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports some form of GROUP BY clause.

### 4.6.156 supportsGroupByBeyondSelect as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports using columns not included in the SELECT statement in a GROUP BY clause provided that all of the columns in the SELECT statement are included in the GROUP BY clause.

### 4.6.157 supportsGroupByUnrelated as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports using a column that is not in the SELECT statement in a GROUP BY clause.

### 4.6.158 supportsIntegrityEnhancementFacility as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports the SQL Integrity Enhancement Facility.

### 4.6.159 supportsLikeEscapeClause as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports specifying a LIKE escape clause.

### 4.6.160 supportsLimitedOuterJoins as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database provides limited support for outer joins.
**Notes:** (This will be true if the method supportsFullOuterJoins returns true).

### 4.6.161 supportsMinimumSQLGrammar as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports the ODBC Minimum SQL grammar.

### 4.6.162 supportsMixedCaseIdentifiers as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database treats mixed case unquoted SQL identifiers as case sensitive and as a result stores them in mixed case.

### 4.6.163 supportsMixedCaseQuotedIdentifiers as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database treats mixed case quoted SQL identifiers as case sensitive and as a result stores them in mixed case.

### 4.6.164 supportsMultipleOpenResults as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether it is possible to have multiple ResultSet objects returned from a CallableStatement object simultaneously.
**Notes:** Returns:
true if a CallableStatement object can return multiple ResultSet objects simultaneously; false otherwise

### 4.6.165 supportsMultipleResultSets as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports getting multiple ResultSet objects from a single call to the method execute.

### 4.6.166 supportsMultipleTransactions as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database allows having multiple transactions open at once (on different connections).

### 4.6.167 supportsNamedParameters as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports named parameters to callable statements.
**Notes:** Returns:
true if named parameters are supported; false otherwise

### 4.6.168 supportsNonNullableColumns as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether columns in this database may be defined as non-nullable.

### 4.6.169 supportsOpenCursorsAcrossCommit as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports keeping cursors open across commits.

### 4.6.170 supportsOpenCursorsAcrossRollback as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports keeping cursors open across rollbacks.

### 4.6.171 supportsOpenStatementsAcrossCommit as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports keeping statements open across commits.

### 4.6.172 supportsOpenStatementsAcrossRollback as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports keeping statements open across rollbacks.

### 4.6.173 supportsOrderByUnrelated as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports using a column that is not in the SELECT statement in an ORDER BY clause.

### 4.6.174 supportsOuterJoins as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports some form of outer join.

### 4.6.175 supportsPositionedDelete as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a catalog name can be used in an index definition statement.

### 4.6.176 supportsPositionedUpdate as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports positioned UPDATE statements.

### 4.6.177 supportsResultSetConcurrency(type as Integer, concurrency as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports the given concurrency type in combination with the given result set type.

### 4.6.178 supportsResultSetHoldability(holdability as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports the given result set holdability.
**Notes:** Parameters:
holdability - one of the following constants: ResultSet.HOLD_CURSORS_OVER_COMMIT or ResultSet.CLOSE_CURSORS_AT_COMMIT
Returns:
true if so; false otherwise

### 4.6.179 supportsResultSetType(type as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports the given result set type.

### 4.6.180 supportsSavepoints as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports savepoints.
**Notes:** Returns:
true if savepoints are supported; false otherwise

### 4.6.181 supportsSchemasInDataManipulation as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a schema name can be used in a data manipulation statement.

### 4.6.182 supportsSchemasInIndexDefinitions as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a schema name can be used in an index definition statement.

### 4.6.183 supportsSchemasInPrivilegeDefinitions as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a schema name can be used in a privilege definition statement.

### 4.6.184 supportsSchemasInProcedureCalls as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a schema name can be used in a procedure call statement.

### 4.6.185 supportsSchemasInTableDefinitions as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a schema name can be used in a table definition statement.

### 4.6.186 supportsSelectForUpdate as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports SELECT FOR UPDATE statements.

### 4.6.187 supportsStatementPooling as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports statement pooling.
**Notes:** Returns:
true if so; false otherwise

### 4.6.188 supportsStoredProcedures as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports stored procedure calls that use the stored procedure escape syntax.

### 4.6.189 supportsSubqueriesInComparisons as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports subqueries in comparison expressions.

### 4.6.190 supportsSubqueriesInExists as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports subqueries in EXISTS expressions.

### 4.6.191 supportsSubqueriesInIns as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports subqueries in IN statements.

### 4.6.192 supportsSubqueriesInQuantifieds as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports subqueries in quantified expressions.

### 4.6.193 supportsTableCorrelationNames as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports table correlation names.

### 4.6.194 supportsTransactionIsolationLevel(level as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports the given transaction isolation level.
**Notes:** Parameters:
level - one of the transaction isolation levels defined in javaConnectionMBS

Returns:
true if so; false otherwise

### 4.6.195 supportsTransactions as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports transactions.
**Notes:** If not, invoking the method commit is a noop, and the isolation level is TRANSACTION_NONE.

### 4.6.196 supportsUnion as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports SQL UNION.

### 4.6.197 supportsUnionAll as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database supports SQL UNION ALL.

### 4.6.198 tableIndexClustered as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that this table index is a clustered index.
**Notes:** A possible value for column TYPE in the ResultSet object returned by the method getIndexInfo.

### 4.6.199 tableIndexHashed as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that this table index is a hashed index.
**Notes:** A possible value for column TYPE in the ResultSet object returned by the method getIndexInfo.

### 4.6.200 tableIndexOther as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that this table index is not a clustered index, a hashed index, or table statistics; it is something other than these.
**Notes:** A possible value for column TYPE in the ResultSet object returned by the method getIndexInfo.

### 4.6.201 tableIndexStatistic as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that this column contains table statistics that are returned in conjunction with a table's index descriptions.
**Notes:** A possible value for column TYPE in the ResultSet object returned by the method getIndexInfo.

### 4.6.202 typeNoNulls as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that a NULL value is NOT allowed for this data type.
**Notes:** A possible value for column NULLABLE in the ResultSet object returned by the method getTypeInfo.

### 4.6.203 typeNullable as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that a NULL value is allowed for this data type.
**Notes:** A possible value for column NULLABLE in the ResultSet object returned by the method getTypeInfo.

### 4.6.204 typeNullableUnknown as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that it is not known whether a NULL value is allowed for this data type.
**Notes:** A possible value for column NULLABLE in the ResultSet object returned by the method getTypeInfo.

### 4.6.205 typePredBasic as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that one can base all WHERE search clauses except WHERE . . . LIKE on this data type.
**Notes:** A possible value for column SEARCHABLE in the ResultSet object returned by the method getTypeInfo.

### 4.6.206 typePredChar as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that the only WHERE search clause that can be based on this type is WHERE . . . LIKE.
**Notes:** A possible value for column SEARCHABLE in the ResultSet object returned by the method getTypeInfo.

### 4.6.207 typePredNone as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that WHERE search clauses are not supported for this type.
**Notes:** A possible value for column SEARCHABLE in the ResultSet object returned by the method getTypeInfo.

### 4.6.208 typeSearchable as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that all WHERE search clauses can be based on this type.
**Notes:** A possible value for column SEARCHABLE in the ResultSet object returned by the method getTypeInfo.

### 4.6.209 updatesAreDetected(type as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether or not a visible row update can be detected by calling the method ResultSet.rowUpdated.
**Notes:** Parameters:
type - the ResultSet type; one of ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE

Returns:
true if changes are detected by the result set type; false otherwise

### 4.6.210 usesLocalFilePerTable as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database uses a file for each table.

### 4.6.211 usesLocalFiles as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether this database stores tables in a local file.

### 4.6.212 versionColumnNotPseudo as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that this version column is NOT a pseudo column.
**Notes:** A possible value for the column PSEUDO_COLUMN in the ResultSet object returned by the method getVersionColumns.

### 4.6.213 versionColumnPseudo as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that this version column is a pseudo column.
**Notes:** A possible value for the column PSEUDO_COLUMN in the ResultSet object returned by the method getVersionColumns.

### 4.6.214 versionColumnUnknown as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates that this version column may or may not be a pseudo column.
**Notes:** A possible value for the column PSEUDO_COLUMN in the ResultSet object returned by the method getVersionColumns.

# 4.7 class JavaExceptionMBS

## 4.7.1 class JavaExceptionMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The exception class used to report java exception.
**Notes:** For SQLExceptions the ErrorNumber property will be filled.
For all java exceptions the message property is filled.
Subclass of the RuntimeException class.
**Blog Entries**

- Encrypted Access database in Xojo

## 4.7.2 Methods

## 4.7.3 RaiseJavaException(message as string)

Plugin Version: 9.0, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** A method to test exception handling.
**Example:**

JavaExceptionMBS.RaiseJavaException "Just a test!"

**Notes:** This method raises a new JavaExceptionMBS with the given message.

## 4.8 class JavaInputStreamMBS

### 4.8.1 class JavaInputStreamMBS

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The plugin class for an inputstream.
**Example:**

```
// your result set
Var r as JavaResultSetMBS

// get binary data for BLOB column nameed 'data'
Var myInputStream as JavaInputStreamMBS = r.getBinaryStream( "data" )
if myInputStream <>nil Then

// read byte for byte in a loop
// better use other read() method with buffer
Var data as string

Do
Var c as Integer = myInputStream.read
If c = -1 then exit

data = data + chrb(c)
Loop

Var myPicture as Picture = Picture.FromData(data)

// work with picture here
End If
```

**Notes:** This abstract class in java is the superclass of all classes representing an input stream of bytes.
Subclass of the JavaObjectMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
**Blog Entries**

- MonkeyBread Software Releases the MBS Real Studio plug-ins in version 13.1
- MBS Real Studio Plugins, version 13.1pr7

### 4.8.2 Methods

### 4.8.3 available as Integer

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns an estimate of the number of bytes that can be read (or skipped over) from this input stream without blocking by the next invocation of a method for this input stream.
**Notes:** The next invocation might be the same thread or another thread. A single read or skip of this many bytes will not block, but may read or skip fewer bytes.
Note that while some implementations of InputStream will return the total number of bytes in the stream, many will not. It is never correct to use the return value of this method to allocate a buffer intended to hold all data in this stream.

A subclass' implementation of this method may choose to throw an IOException if this input stream has been closed by invoking the close() method.

The available method for class InputStream always returns 0.

This method should be overridden by subclasses.

Returns an estimate of the number of bytes that can be read (or skipped over) from this input stream without blocking or 0 when it reaches the end of the input stream.

Throws:
IOException - if an I/O error occurs.

### 4.8.4 close

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Closes this input stream and releases any system resources associated with the stream.

### 4.8.5 Constructor

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 4.8.6 mark(readlimit as Integer)

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Marks the current position in this input stream.
**Notes:** A subsequent call to the reset method repositions this stream at the last marked position so that subsequent reads re-read the same bytes.

The readlimit arguments tells this input stream to allow that many bytes to be read before the mark position gets invalidated.

The general contract of mark is that, if the method markSupported returns true, the stream somehow remembers all the bytes read after the call to mark and stands ready to supply those same bytes again if and whenever the method reset is called. However, the stream is not required to remember any data at all if more than readlimit bytes are read from the stream before reset is called.

Marking a closed stream should not have any effect on the stream.

The mark method of InputStream does nothing.

Parameters:
readlimit - the maximum limit of bytes that can be read before the mark position becomes invalid.

### 4.8.7 markSupported as boolean

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Tests if this input stream supports the mark and reset methods.
**Notes:** Whether or not mark and reset are supported is an invariant property of a particular input stream instance. The markSupported method of InputStream returns false.

Returns true if this stream instance supports the mark and reset methods; false otherwise.

### 4.8.8 read as Integer

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Reads the next byte of data from the input stream.
**Notes:** The value byte is returned as an int in the range 0 to 255. If no byte is available because the end of the stream has been reached, the value -1 is returned. This method blocks until input data is available, the end of the stream is detected, or an exception is thrown.
A subclass must provide an implementation of this method.

Returns the next byte of data, or -1 if the end of the stream is reached.

Throws:
IOException - if an I/O error occurs.
See also:

### 4.8.9 read(bytes as JavaByteArrayMBS) as Integer

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Reads some number of bytes from the input stream and stores them into the buffer array bytes.
**Notes:** The number of bytes actually read is returned as an integer. This method blocks until input data is available, end of file is detected, or an exception is thrown.
If the length of bytes is zero, then no bytes are read and 0 is returned; otherwise, there is an attempt to read at least one byte. If no byte is available because the stream is at the end of the file, the value -1 is returned; otherwise, at least one byte is read and stored into bytes.

The first byte read is stored into element bytes [ 0 ] , the next one into bytes [ 1 ] , and so on. The number of bytes read is, at most, equal to the length of bytes. Let k be the number of bytes actually read; these bytes will be stored in elements bytes [ 0 ] through bytes [ k-1 ] , leaving elements bytes [ k ] through bytes [ bytes.length-1 ] unaffected.

The read(bytes) method for class InputStream has the same effect as:

read(bytes, 0, bytes.length)

Parameters:
bytes: the buffer into which the data is read.

Returns the total number of bytes read into the buffer, or -1 is there is no more data because the end of the stream has been reached.

Throws:
IOException - If the first byte cannot be read for any reason other than the end of the file, if the input stream has been closed, or if some other I/O error occurs.
NullPointerException - if bytes is null.
See also:



### 4.8.10 read(bytes as JavaByteArrayMBS, Offset as Integer, Length as Integer) as Integer

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Reads up to Length bytes of data from the input stream into an array of bytes.
**Notes:** An attempt is made to read as many as Length bytes, but a smaller number may be read. The number of bytes actually read is returned as an integer.
This method blocks until input data is available, end of file is detected, or an exception is thrown.

If Length is zero, then no bytes are read and 0 is returned; otherwise, there is an attempt to read at least one byte. If no byte is available because the stream is at end of file, the value -1 is returned; otherwise, at least one byte is read and stored into bytes.

The first byte read is stored into element bytes [ Offset ] , the next one into bytes [ Offset+1 ] , and so on. The number of bytes read is, at most, equal to Length. Let k be the number of bytes actually read; these bytes will be stored in elements bytes [ Offset ] through bytes [ Offset+k-1 ] , leaving elements bytes [ Offset+k ] through bytes [ Offset+Length-1 ] unaffected.

In every case, elements bytes [ 0 ] through bytes [ Offset ] and elements bytes [ Offset+Length ] through bytes [ bytes.length-1 ] are unaffected.

The read(bytes, Offset, Length) method for class InputStream simply calls the method read() repeatedly. If the first such call results in an IOException, that exception is returned from the call to the read(bytes, Offset, Length) method. If any subsequent call to read() results in a IOException, the exception is caught and treated as if it were end of file; the bytes read up to that point are stored into bytes and the number of bytes read before the exception occurred is returned. The default implementation of this method blocks until the requested amount of input data len has been read, end of file is detected, or an exception is thrown. Subclasses are encouraged to provide a more efficient implementation of this method.

Parameters:
bytes: the buffer into which the data is read.
Offset: the start offset in array bytes at which the data is written.
Length: the maximum number of bytes to read.

Returns the total number of bytes read into the buffer, or -1 if there is no more data because the end of the stream has been reached.

Throws:
IOException - If the first byte cannot be read for any reason other than end of file, or if the input stream has been closed, or if some other I/O error occurs.
NullPointerException - If bytes is null.
IndexOutOfBoundsException - If Offset is negative, Length is negative, or Length is greater than bytes.length - Offset
See also:


- 4.8.8 read as Integer 220
- 4.8.9 read(bytes as JavaByteArrayMBS) as Integer 221

### 4.8.11 reset

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Repositions this stream to the position at the time the mark method was last called on this input stream.
**Notes:** The general contract of reset is:

- If the method markSupported returns true, then:
- If the method mark has not been called since the stream was created, or the number of bytes read from the stream since mark was last called is larger than the argument to mark at that last call, then an IOException might be thrown.
- If such an IOException is not thrown, then the stream is reset to a state such that all the bytes read since the most recent call to mark (or since the start of the file, if mark has not been called) will be resupplied to subsequent callers of the read method, followed by any bytes that otherwise would have been the next input data as of the time of the call to reset.
- If the method markSupported returns false, then:
- The call to reset may throw an IOException.
- If an IOException is not thrown, then the stream is reset to a fixed state that depends on the particular type of the input stream and how it was created. The bytes that will be supplied to subsequent callers of the read method depend on the particular type of the input stream.

The method reset for class InputStream does nothing except throw an IOException.

Throws:
IOException - if this stream has not been marked or if the mark has been invalidated.

### 4.8.12 skip(count as Int64) as Int64

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Skips over and discards count bytes of data from this input stream.
**Notes:** The skip method may, for a variety of reasons, end up skipping over some smaller number of bytes, possibly 0. This may result from any of a number of conditions; reaching end of file before count bytes have been skipped is only one possibility. The actual number of bytes skipped is returned. If count is negative, no bytes are skipped.
The skip method of this class creates a byte array and then repeatedly reads into it until count bytes have been read or the end of the stream has been reached. Subclasses are encouraged to provide a more efficient implementation of this method. For instance, the implementation may depend on the ability to seek.

Parameters:
count: the number of bytes to be skipped.

Returns the actual number of bytes skipped.

Throws:
IOException - if the stream does not support seek, or if some other I/O error occurs.

# 4.9 class JavaParameterMetaDataMBS

## 4.9.1 class JavaParameterMetaDataMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** An object that can be used to get information about the types and properties of the parameters in a PreparedStatement object.
**Notes:** Subclass of the JavaObjectMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

## 4.9.2 Methods

## 4.9.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

## 4.9.4 getParameterClassName(param as Integer) as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the fully-qualified name of the Java class whose instances should be passed to the method PreparedStatement.setObject.
**Notes:** Parameters:
param - the first parameter is 1, the second is 2, ...
Returns:
the fully-qualified name of the class in the Java programming language that would be used by the method PreparedStatement.setObject to set the value in the specified parameter. This is the class name used for custom mapping.

## 4.9.5 getParameterCount as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the number of parameters in the PreparedStatement object for which this ParameterMetaData object contains information.
**Notes:** Returns:
the number of parameters

### 4.9.6 getParameterMode(param as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the designated parameter's mode.
**Notes:** Parameters:
param - the first parameter is 1, the second is 2, ...
Returns:
mode of the parameter; one of ParameterMetaData.parameterModeIn, ParameterMetaData.parameterModeOut, or ParameterMetaData.parameterModeInOut ParameterMetaData.parameterModeUnknown.

### 4.9.7 getParameterType(param as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the designated parameter's SQL type.
**Notes:** Parameters:
param - the first parameter is 1, the second is 2, ...
Returns:
SQL type from java.sql.Types

### 4.9.8 getParameterTypeName(param as Integer) as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the designated parameter's database-specific type name.
**Notes:** Parameters:
param - the first parameter is 1, the second is 2, ...
Returns:
type the name used by the database. If the parameter type is a user-defined type, then a fully-qualified type name is returned.

### 4.9.9 getPrecision(param as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the designated parameter's number of decimal digits.
**Notes:** Parameters:
param - the first parameter is 1, the second is 2, ...
Returns:
precision

### 4.9.10 getScale(param as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the designated parameter's number of digits to right of the decimal point.
**Notes:** Parameters:
param - the first parameter is 1, the second is 2, ...
Returns:
scale

### 4.9.11 isNullable(param as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether null values are allowed in the designated parameter.
**Notes:** Parameters:
param - the first parameter is 1, the second is 2, ...
Returns:
the nullability status of the given parameter; one of ParameterMetaData.parameterNoNulls, ParameterMetaData.parameterNullable, or ParameterMetaData.parameterNullableUnknown

### 4.9.12 isSigned(param as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether values for the designated parameter can be signed numbers.
**Notes:** Parameters:
param - the first parameter is 1, the second is 2, ...
Returns:
true if so; false otherwise

### 4.9.13 parameterModeIn as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that the parameter's mode is IN.

### 4.9.14 parameterModeInOut as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that the parameter's mode is INOUT.

### 4.9.15 parameterModeOut as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that the parameter's mode is OUT.

### 4.9.16 parameterModeUnknown as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that the mode of the parameter is unknown.

### 4.9.17 parameterNoNulls as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that a parameter will not allow NULL values.

### 4.9.18 parameterNullable as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that a parameter will allow NULL values.

### 4.9.19 parameterNullableUnknown as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that the nullability of a parameter is unknown.

# 4.10 class JavaPreparedStatementMBS

## 4.10.1 class JavaPreparedStatementMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** An object that represents a precompiled SQL statement.
**Notes:** A SQL statement is precompiled and stored in a PreparedStatement object. This object can then be used to efficiently execute this statement multiple times.

Note: The setter methods (setShort, setString, and so on) for setting IN parameter values must specify types that are compatible with the defined SQL type of the input parameter. For instance, if the IN parameter has SQL type INTEGER, then the method setInt should be used.

If arbitrary parameter type conversions are required, the method setObject should be used with a target SQL type.

In the following example of setting a parameter, con represents an active connection:

PreparedStatement pstmt = con.prepareStatement("UPDATE EMPLOYEES SET SALARY = ? WHERE ID = ?");
pstmt.setBigDecimal(1, 153833.00)
pstmt.setInt(2, 110592)
Subclass of the JavaStatementMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
**Blog Entries**

- MBS Xojo Plugins, version 24.4pr3
- MBS Real Studio Plugins, version 11.3pr1

## 4.10.2 Methods

## 4.10.3 addBatch

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Adds a set of parameters to this PreparedStatement object's batch of commands.

## 4.10.4 clearParameters

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Clears the current parameter values immediately.
**Notes:** In general, parameter values remain in force for repeated use of a statement. Setting a parameter value automatically clears its previous value. However, in some cases it is useful to immediately release the resources used by the current parameter values; this can be done by calling the method clearParameters.

### 4.10.5 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 4.10.6 execute as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Executes the SQL statement in this PreparedStatement object, which may be any kind of SQL statement.
**Notes:** Some prepared statements return multiple results; the execute method handles these complex statements as well as the simpler form of statements handled by the methods executeQuery and executeUpdate. The execute method returns a boolean to indicate the form of the first result. You must call either the method getResultSet or getUpdateCount to retrieve the result; you must call getMoreResults to move to any subsequent result(s).

Returns:
true if the first result is a ResultSet object; false if the first result is an update count or there is no result

### 4.10.7 executeQuery as JavaResultSetMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Executes the SQL query in this PreparedStatement object and returns the ResultSet object generated by the query.
**Notes:** Returns:
a ResultSet object that contains the data produced by the query; never null

### 4.10.8 executeUpdate as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Executes the SQL statement in this PreparedStatement object, which must be an SQL INSERT, UPDATE or DELETE statement; or an SQL statement that returns nothing, such as a DDL statement.

**Notes:** Returns:
either (1) the row count for INSERT, UPDATE, or DELETE statements or (2) 0 for SQL statements that return nothing.

### 4.10.9 getMetaData as JavaResultSetMetaDataMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves a ResultSetMetaData object that contains information about the columns of the ResultSet object that will be returned when this PreparedStatement object is executed.
**Notes:** Because a PreparedStatement object is precompiled, it is possible to know about the ResultSet object that it will return without having to execute it. Consequently, it is possible to invoke the method getMetaData on a PreparedStatement object rather than waiting to execute it and then invoking the ResultSet.getMetaData method on the ResultSet object that is returned.

NOTE: Using this method may be expensive for some drivers due to the lack of underlying DBMS support.

Returns:
the description of a ResultSet object's columns or null if the driver cannot return a ResultSetMetaData object

### 4.10.10 getParameterMetaData as JavaParameterMetaDataMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the number, types and properties of this PreparedStatement object's parameters.
**Notes:** Returns:
a ParameterMetaData object that contains information about the number, types and properties of this PreparedStatement object's parameters

### 4.10.11 setBlob(parameterIndex as Integer, value as JavaBlobMBS)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Blob object.
**Notes:** The driver converts this to an SQL BLOB value when it sends it to the database.

Parameters:
parameterIndex - the first parameter is 1, the second is 2, ...
value - a Blob object that maps an SQL BLOB value

### 4.10.12 setBoolean(parameterIndex as Integer, value as boolean)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java boolean value.
**Notes:** The driver converts this to an SQL BIT value when it sends it to the database.

Parameters:
parameterIndex - the first parameter is 1, the second is 2, ...
value - the parameter value

### 4.10.13 setByte(parameterIndex as Integer, value as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java byte value.
**Notes:** The driver converts this to an SQL TINYINT value when it sends it to the database.

Parameters:
parameterIndex - the first parameter is 1, the second is 2, ...
value - the parameter value

### 4.10.14 setBytes(parameterIndex as Integer, Value as String)

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java array of bytes.
**Notes:** The driver converts this to an SQL VARBINARY or LONGVARBINARY (depending on the argument's size relative to the driver's limits on VARBINARY values) when it sends it to the database.

parameterIndex: the first parameter is 1, the second is 2, ...
Value: the parameter value as a string.

### 4.10.15 setClob(parameterIndex as Integer, value as JavaClobMBS)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Clob object.
**Notes:** The driver converts this to an SQL CLOB value when it sends it to the database.

Parameters:

i - the first parameter is 1, the second is 2, ...
value - a Clob object that maps an SQL CLOB value

### 4.10.16 setDate(parameterIndex as integer, value as JavaObjectMBS)

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Date object.
**Notes:** You need to create a java.sql.Date object yourself to use this.

### 4.10.17 setDouble(parameterIndex as Integer, value as Double)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java double value.
**Notes:** The driver converts this to an SQL DOUBLE value when it sends it to the database.

Parameters:
parameterIndex - the first parameter is 1, the second is 2, ...
value - the parameter value

### 4.10.18 setFloat(parameterIndex as Integer, value as single)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java float value.
**Notes:** The driver converts this to an SQL FLOAT value when it sends it to the database.

Parameters:
parameterIndex - the first parameter is 1, the second is 2, ...
value - the parameter value

### 4.10.19 setInt(parameterIndex as Integer, value as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java int value.
**Notes:** The driver converts this to an SQL INTEGER value when it sends it to the database.

Parameters:
parameterIndex - the first parameter is 1, the second is 2, ...
value - the parameter value

### 4.10.20 setLong(parameterIndex as Integer, value as Int64)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java long value.
**Notes:** The driver converts this to an SQL BIGINT value when it sends it to the database.

Parameters:
parameterIndex - the first parameter is 1, the second is 2, ...
value - the parameter value

### 4.10.21 setNull(parameterIndex as Integer, sqlType as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to SQL NULL.
**Notes:** You must specify the parameter's SQL type.

Parameters:
parameterIndex - the first parameter is 1, the second is 2, ...
sqlType - the SQL type code defined in java.sql.Types

### 4.10.22 setShort(parameterIndex as Integer, value as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java short value.
**Notes:** The driver converts this to an SQL SMALLINT value when it sends it to the database.

Parameters:
parameterIndex - the first parameter is 1, the second is 2, ...
value - the parameter value

### 4.10.23 setString(parameterIndex as Integer, value as string)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Java String value.
**Notes:** The driver converts this to an SQL VARCHAR or LONGVARCHAR value (depending on the argument's size relative to the driver's limits on VARCHAR values) when it sends it to the database.

Parameters:
parameterIndex - the first parameter is 1, the second is 2, ...
value - the parameter value

### 4.10.24 setTime(parameterIndex as integer, value as JavaObjectMBS)

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given time object.
**Notes:** You need to create a java.sql.Time object yourself to use this.

### 4.10.25 setTimestamp(parameterIndex as integer, value as JavaObjectMBS)

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the designated parameter to the given Timestamp object.
**Notes:** You need to create a java.sql.Timestamp object yourself to use this.

## 4.11 class JavaResultSetMBS

### 4.11.1 class JavaResultSetMBS

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The class for a recordset in Java.
**Notes:** Nearly all methods on this class can raise java exceptions which you can get using the error property. (and errorstring and errorcode)
Subclass of the JavaObjectMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
**Blog Entries**

- MBS Xojo Plugins, version 24.4pr3
- Prefetching records from databases
- Encrypted Access database in Xojo
- MBS Real Studio Plugins, version 13.1pr7
- MBS Real Studio Plugins, version 11.3pr1

### 4.11.2 Methods

### 4.11.3 absolute(row as Integer) as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Moves the cursor to the given row number in this ResultSet object.
**Example:**

```
Var db as JavaConnectionMBS // your database
Var r as JavaResultSetMBS
Var s as JavaStatementMBS

// check second row
s=db.createStatement
r=s.executeQuery("SELECT * from myTable")

if r<>Nil then
if r.absolute(2) then
MsgBox str(R.getInt("test_id"))+" "+r.getString("test_val")
end if
end if
```

**Notes:** See the java documentation for details on java.sql.ResultSet.absolute.

### 4.11.4 afterLast

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Moves the cursor to the end of this ResultSet object, just after the last row.
**Notes:** See the java documentation for details on java.sql.ResultSet.afterLast.

### 4.11.5 beforeFirst

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Moves the cursor to the front of this ResultSet object, just before the first row.
**Notes:** See the java documentation for details on java.sql.ResultSet.beforeFirst.

### 4.11.6 cancelRowUpdates

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Cancels the updates made to the current row in this ResultSet object.
**Notes:** This method may be called after calling an updater method(s) and before calling the method updateRow to roll back the updates made to a row. If no updates have been made or updateRow has already been called, this method has no effect.

See the java documentation for details on java.sql.ResultSet.cancelRowUpdates.

### 4.11.7 clearWarnings

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Clears all warnings reported on this ResultSet object.
**Notes:** See the java documentation for details on java.sql.ResultSet.clearWarnings.

### 4.11.8 CLOSE_CURSORS_AT_COMMIT as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that ResultSet objects should be closed when the method Connection.commit is called.

### 4.11.9 CONCUR_READ_ONLY as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating the concurrency mode for a ResultSet object that may NOT be updated.

### 4.11.10 CONCUR_UPDATABLE as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating the concurrency mode for a ResultSet object that may be updated.

### 4.11.11 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 4.11.12 deleteRow

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Deletes the current row from this ResultSet object and from the underlying database.
**Notes:** See the java documentation for details on java.sql.ResultSet.deleteRow.

### 4.11.13 FETCH_FORWARD as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that the rows in a result set will be processed in a forward direction; first-to-last.
**Notes:** This constant is used by the method setFetchDirection as a hint to the driver, which the driver may ignore.

### 4.11.14 FETCH_REVERSE as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that the rows in a result set will be processed in a reverse direction; last-to-first.

**Notes:** This constant is used by the method setFetchDirection as a hint to the driver, which the driver may ignore.

### 4.11.15 FETCH_UNKNOWN as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that the order in which rows in a result set will be processed is unknown.
**Notes:** This constant is used by the method setFetchDirection as a hint to the driver, which the driver may ignore.

### 4.11.16 findColumn(column as string) as Integer

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Maps the given ResultSet column name to its ResultSet column index.
**Example:**

```
Var r as JavaResultSetMBS // your result set
MsgBox str(r.findColumn("test_id"))+" "+str(r.findColumn("test_val"))
```

**Notes:** Returns 0 on any error.
See the java documentation for details on java.sql.ResultSet.findColumn.

### 4.11.17 first as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Moves the cursor to the first row in this ResultSet object.
**Notes:** See the java documentation for details on java.sql.ResultSet.first.

### 4.11.18 getAsciiStream(column as Integer) as JavaInputStreamMBS

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as a stream of ASCII characters.
**Notes:** Retrieves the value of the designated column in the current row of this ResultSet object as a stream of ASCII characters. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large LONGVARCHAR values. The JDBC driver will do any necessary conversion from the database format into ASCII.

Note: All the data in the returned stream must be read prior to getting the value of any other column. The next call to a getter method implicitly closes the stream. Also, a stream may return 0 when the method InputStream.available is called whether there is data available or not.

Parameters:
column: the first column is 1, the second is 2, ...

Returns a Java input stream that delivers the database column value as a stream of one-byte ASCII characters; if the value is SQL NULL, the value returned is null

Throws:
SQLException - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set.
See also:

- 4.11.19 getAsciiStream(column as string) as JavaInputStreamMBS 240

### 4.11.19 getAsciiStream(column as string) as JavaInputStreamMBS

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as an Array object in the Java programming language.
**Notes:** Retrieves the value of the designated column in the current row of this ResultSet object as a stream of ASCII characters. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large LONGVARCHAR values. The JDBC driver will do any necessary conversion from the database format into ASCII.
Note: All the data in the returned stream must be read prior to getting the value of any other column. The next call to a getter method implicitly closes the stream. Also, a stream may return 0 when the method available is called whether there is data available or not.

Parameters:
column: the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column

Returns a Java input stream that delivers the database column value as a stream of one-byte ASCII characters. If the value is SQL NULL, the value returned is null.

Throws:
SQLException - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set
See also:

- 4.11.18 getAsciiStream(column as Integer) as JavaInputStreamMBS 239

### 4.11.20 getBinaryStream(column as Integer) as JavaInputStreamMBS

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as a stream of uninterpreted bytes.
**Notes:** Retrieves the value of the designated column in the current row of this ResultSet object as a stream of uninterpreted bytes. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large LONGVARBINARY values.
Note: All the data in the returned stream must be read prior to getting the value of any other column. The next call to a getter method implicitly closes the stream. Also, a stream may return 0 when the method InputStream.available is called whether there is data available or not.

Parameters:
column: the first column is 1, the second is 2, ...

Returns a Java input stream that delivers the database column value as a stream of uninterpreted bytes; if the value is SQL NULL, the value returned is null

Throws:
SQLException - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set
See also:

- 4.11.21 getBinaryStream(column as string) as JavaInputStreamMBS 241

### 4.11.21 getBinaryStream(column as string) as JavaInputStreamMBS

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as a stream of uninterpreted bytes.
**Notes:** Retrieves the value of the designated column in the current row of this ResultSet object as a stream of uninterpreted bytes. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large LONGVARBINARY values.
Note: All the data in the returned stream must be read prior to getting the value of any other column. The next call to a getter method implicitly closes the stream. Also, a stream may return 0 when the method available is called whether there is data available or not.

Parameters:
column: the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column

Returns a Java input stream that delivers the database column value as a stream of uninterpreted bytes; if the value is SQL NULL, the result is null.

Throws SQLException - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set
See also:

- 4.11.20 getBinaryStream(column as Integer) as JavaInputStreamMBS 241

### 4.11.22 getBlob(column as Integer) as JavaBlobMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as a Blob object in the Java programming language.
**Notes:** Column: the first column is 1, the second is 2, ...

Returns a Blob object representing the SQL BLOB value in the specified column
See also:

- 4.11.23 getBlob(column as string) as JavaBlobMBS 242

### 4.11.23 getBlob(column as string) as JavaBlobMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as a Blob object in the Java programming language.
**Notes:** column: the name of the column from which to retrieve the value
Returns a Blob object representing the SQL BLOB value in the specified column
See also:

- 4.11.22 getBlob(column as Integer) as JavaBlobMBS 242

### 4.11.24 getBoolean(column as Integer) as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gets the value of the designated column in the current row of this ResultSet object as a boolean.
**Notes:** See the java documentation for details on java.sql.ResultSet.getBoolean.
See also:

- 4.11.25 getBoolean(column as string) as boolean 243

### 4.11.25 getBoolean(column as string) as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gets the value of the designated column in the current row of this ResultSet object as a boolean.
**Notes:** See the java documentation for details on java.sql.ResultSet.getBoolean.
See also:

- 4.11.24 getBoolean(column as Integer) as boolean 242

### 4.11.26 getByte(column as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as a byte in the Java programming language.
**Notes:** Parameters:
columnIndex - the first column is 1, the second is 2, ...
Returns:
the column value; if the value is SQL NULL, the value returned is 0
See also:

- 4.11.27 getByte(column as string) as Integer 243

### 4.11.27 getByte(column as string) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as a byte in the Java programming language.
**Notes:** Parameters:
columnName - the SQL name of the column
Returns:
the column value; if the value is SQL NULL, the value returned is 0
See also:

- 4.11.26 getByte(column as Integer) as Integer 243

### 4.11.28 getBytes(column as Integer) as string

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as a byte array in the Java programming language.
**Notes:** column: the first column is 1, the second is 2, ...

the plugin gives you the bytes as a string with no encoding.
See also:

- 4.11.29 getBytes(column as string) as string 244

### 4.11.29 getBytes(column as string) as string

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as a byte array in the Java programming language.
**Notes:** the plugin gives you the bytes as a string with no encoding.
See also:

- 4.11.28 getBytes(column as Integer) as string 243

### 4.11.30 getClob(column as Integer) as JavaClobMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as a Clob object in the Java programming language.
**Notes:** Column: the first column is 1, the second is 2, ...
Returns a Clob object representing the SQL CLOB value in the specified column
See also:

- 4.11.31 getClob(column as string) as JavaClobMBS 244

### 4.11.31 getClob(column as string) as JavaClobMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as a Clob object in the Java programming language.
**Notes:** column: the name of the column from which to retrieve the value
Returns a Clob object representing the SQL CLOB value in the specified column
See also:

- 4.11.30 getClob(column as Integer) as JavaClobMBS 244

### 4.11.32 getConcurrency as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the concurrency mode of this ResultSet object.
**Notes:** The concurrency used is determined by the Statement object that created the result set.
Returns:
the concurrency type, either ResultSet.CONCUR_READ_ONLY or ResultSet.CONCUR_UPDATABLE

### 4.11.33 getCursorName as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the name of the SQL cursor used by this ResultSet object.
**Notes:** In SQL, a result table is retrieved through a cursor that is named. The current row of a result set can be updated or deleted using a positioned update/delete statement that references the cursor name. To insure that the cursor has the proper isolation level to support update, the cursor's SELECT statement should be of the form SELECT FOR UPDATE. If FOR UPDATE is omitted, the positioned updates may fail.

The JDBC API supports this SQL feature by providing the name of the SQL cursor used by a ResultSet object. The current row of a ResultSet object is also the current row of this SQL cursor.

Note: If positioned update is not supported, a SQLException is thrown.

Returns:
the SQL name for this ResultSet object's cursor

### 4.11.34 getDate(column as integer) as JavaObjectMBS

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Queries date value.
See also:

- 4.11.35 getDate(column as string) as JavaObjectMBS 245

### 4.11.35 getDate(column as string) as JavaObjectMBS

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Queries date value.
See also:

- 4.11.34 getDate(column as integer) as JavaObjectMBS 245

### 4.11.36 getDouble(column as Integer) as Double

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gets the value of the designated column in the current row of this ResultSet object as a double.
**Notes:** See the java documentation for details on java.sql.ResultSet.getDouble.
See also:

- 4.11.37 getDouble(column as string) as Double 246

### 4.11.37 getDouble(column as string) as Double

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gets the value of the designated column in the current row of this ResultSet object as a double.
**Notes:** See the java documentation for details on java.sql.ResultSet.getDouble.
See also:

- 4.11.36 getDouble(column as Integer) as Double 246

### 4.11.38 getFloat(column as Integer) as single

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as a float in the Java programming language.
**Notes:** Parameters:
columnIndex - the first column is 1, the second is 2, ...
Returns:
the column value; if the value is SQL NULL, the value returned is 0
See also:

- 4.11.39 getFloat(column as string) as single 246

### 4.11.39 getFloat(column as string) as single

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as a float in the Java programming language.
**Notes:** Parameters:
columnName - the SQL name of the column
Returns:
the column value; if the value is SQL NULL, the value returned is 0
See also:

- 4.11.38 getFloat(column as Integer) as single 246

### 4.11.40 getInt(column as Integer) as Integer

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gets the value of the designated column in the current row of this ResultSet object as an int.
**Notes:** Parameters:
columnIndex - the first column is 1, the second is 2, ...
Returns:
the column value; if the value is SQL NULL, the value returned is 0

See the java documentation for details on java.sql.ResultSet.getInt.
See also:

- 4.11.41 getInt(column as string) as Integer 247

### 4.11.41 getInt(column as string) as Integer

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gets the value of the designated column in the current row of this ResultSet object as an int.
**Example:**

```
Var r as JavaResultSetMBS // your result set
MsgBox str(R.getInt("test_id"))+" "+r.getString("test_val")
```

**Notes:** See the java documentation for details on java.sql.ResultSet.getInt.
See also:

- 4.11.40 getInt(column as Integer) as Integer 247

### 4.11.42 getLong(column as Integer) as int64

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gets the value of the designated column in the current row of this ResultSet object as an int64.
**Notes:** See the java documentation for details on java.sql.ResultSet.getLong.
See also:

- 4.11.43 getLong(column as string) as int64 247

### 4.11.43 getLong(column as string) as int64

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gets the value of the designated column in the current row of this ResultSet object as an int64.
**Notes:** See the java documentation for details on java.sql.ResultSet.getLong.
See also:

- 4.11.42 getLong(column as Integer) as int64 247

### 4.11.44 getMetaData as JavaResultSetMetaDataMBS

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the number, types and properties of this ResultSet object's columns.
**Notes:** Returns:
the description of this ResultSet object's columns

### 4.11.45 getRow as Integer

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the current row number.
**Notes:** See the java documentation for details on java.sql.ResultSet.getRow.

### 4.11.46 getShort(column as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as a short in the Java programming language.
**Notes:** Parameters:
columnIndex - the first column is 1, the second is 2, ...
Returns:
the column value; if the value is SQL NULL, the value returned is 0.
See also:

- 4.11.47 getShort(column as string) as Integer 248

### 4.11.47 getShort(column as string) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the value of the designated column in the current row of this ResultSet object as a short in the Java programming language.
**Notes:** Parameters:
columnName - the SQL name of the column

Returns:
the column value; if the value is SQL NULL, the value returned is 0
See also:

- 4.11.46 getShort(column as Integer) as Integer 248

### 4.11.48 getString(column as Integer) as string

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gets the value of the designated column in the current row of this ResultSet object as a String.
**Notes:** See the java documentation for details on java.sql.ResultSet.getString.
See also:

- 4.11.49 getString(column as string) as string 249

### 4.11.49 getString(column as string) as string

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gets the value of the designated column in the current row of this ResultSet object as a String.
**Example:**

```
Var r as JavaResultSetMBS // your result set
MsgBox str(R.getInt("test_id"))+" "+r.getString("test_val")
```

**Notes:** See the java documentation for details on java.sql.ResultSet.getString.
See also:

- 4.11.48 getString(column as Integer) as string 249

### 4.11.50 getTime(column as integer) as JavaObjectMBS

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Queries time value.
See also:

- 4.11.51 getTime(column as string) as JavaObjectMBS 249

### 4.11.51 getTime(column as string) as JavaObjectMBS

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Queries time value.
See also:

- 4.11.50 getTime(column as integer) as JavaObjectMBS 249

### 4.11.52 getTimestamp(column as integer) as JavaObjectMBS

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Queries timestamp value.
See also:

- 4.11.53 getTimestamp(column as string) as JavaObjectMBS 250

### 4.11.53 getTimestamp(column as string) as JavaObjectMBS

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Queries timestamp value.
See also:

- 4.11.52 getTimestamp(column as integer) as JavaObjectMBS 250

### 4.11.54 getType as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the type of this ResultSet object.
**Notes:** The type is determined by the Statement object that created the result set.
Returns:
ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_SENSITIVE

### 4.11.55 getUnicodeStream(column as Integer) as JavaInputStreamMBS

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Deprecated. use getCharacterStream in place of getUnicodeStream
**Notes:** Retrieves the value of the designated column in the current row of this ResultSet object as as a stream of two-byte 3 characters. The first byte is the high byte; the second byte is the low byte. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large LONGVARCHARvalues. The JDBC driver will do any necessary conversion from the database format into Unicode.
Note: All the data in the returned stream must be read prior to getting the value of any other column. The

next call to a getter method implicitly closes the stream. Also, a stream may return 0 when the method InputStream.available is called, whether there is data available or not.

Parameters:
column: the first column is 1, the second is 2, ...

Returns a Java input stream that delivers the database column value as a stream of two-byte Unicode characters; if the value is SQL NULL, the value returned is null

Throws:
SQLException - if the columnIndex is not valid; if a database access error occurs or this method is called on a closed result set
SQLFeatureNotSupportedException - if the JDBC driver does not support this method
See also:


- 4.11.56 getUnicodeStream(column as string) as JavaInputStreamMBS 251


### 4.11.56 getUnicodeStream(column as string) as JavaInputStreamMBS

Plugin Version: 13.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Deprecated. use getCharacterStream instead
**Notes:** Retrieves the value of the designated column in the current row of this ResultSet object as a stream of two-byte Unicode characters. The first byte is the high byte; the second byte is the low byte. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large LONGVARCHAR values. The JDBC technology-enabled driver will do any necessary conversion from the database format into Unicode.
Note: All the data in the returned stream must be read prior to getting the value of any other column. The next call to a getter method implicitly closes the stream. Also, a stream may return 0 when the method InputStream.available is called, whether there is data available or not.

Parameters:
column: the label for the column specified with the SQL AS clause. If the SQL AS clause was not specified, then the label is the name of the column

Returns a Java input stream that delivers the database column value as a stream of two-byte Unicode characters. If the value is SQL NULL, the value returned is null.

Throws:
SQLException - if the columnLabel is not valid; if a database access error occurs or this method is called on a closed result set
SQLFeatureNotSupportedException - if the JDBC driver does not support this method
See also:

### 4.11.57 HOLD_CURSORS_OVER_COMMIT as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that ResultSet objects should not be closed when the method Connection.commit is called.

### 4.11.58 insertRow

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Inserts the contents of the insert row into this ResultSet objaect and into the database.
**Notes:** See the java documentation for details on java.sql.ResultSet.insertRow.

### 4.11.59 isAfterLast as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates whether the cursor is after the last row in this ResultSet object.
**Notes:** See the java documentation for details on java.sql.ResultSet.isAfterLast.

### 4.11.60 isBeforeFirst as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether the cursor is before the first row in this ResultSet object.
**Notes:** Returns:
true if the cursor is before the first row; false if the cursor is at any other position or the result set contains no rows

See the java documentation for details on java.sql.ResultSet.isBeforeFirst.

### 4.11.61 isFirst as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates whether the cursor is on the first row of this ResultSet object.
**Notes:** See the java documentation for details on java.sql.ResultSet.isFirst.

### 4.11.62 isLast as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates whether the cursor is on the last row of this ResultSet object.
**Notes:** See the java documentation for details on java.sql.ResultSet.isLast.

### 4.11.63 last as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Moves the cursor to the last row in this ResultSet object.
**Notes:** See the java documentation for details on java.sql.ResultSet.last.

### 4.11.64 moveToCurrentRow

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Moves the cursor to the remembered cursor position, usually the current row.
**Notes:** This method has no effect if the cursor is not on the insert row.

See the java documentation for details on java.sql.ResultSet.moveToCurrentRow.

### 4.11.65 moveToInsertRow

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Moves the cursor to the insert row.
**Notes:** The current cursor position is remembered while the cursor is positioned on the insert row. The insert row is a special row associated with an updatable result set. It is essentially a buffer where a new row may be constructed by calling the updater methods prior to inserting the row into the result set. Only the updater, getter, and insertRow methods may be called when the cursor is on the insert row. All of the columns in a result set must be given a value each time this method is called before calling insertRow. An updater method must be called before a getter method can be called on a column value.

See the java documentation for details on java.sql.ResultSet.moveToInsertRow.

### 4.11.66 NextRecord as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Moves the cursor down one row from its current position.
**Notes:** See the java documentation for details on java.sql.ResultSet.next.

### 4.11.67 previousRecord as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Moves the cursor to the previous row in this ResultSet object.
**Notes:** See the java documentation for details on java.sql.ResultSet.previous.

### 4.11.68 refreshRow

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Refreshes the current row with its most recent value in the database.
**Notes:** This method cannot be called when the cursor is on the insert row.
The refreshRow method provides a way for an application to explicitly tell the JDBC driver to refetch a row(s) from the database. An application may want to call refreshRow when caching or prefetching is being done by the JDBC driver to fetch the latest value of a row from the database. The JDBC driver may actually refresh multiple rows at once if the fetch size is greater than one.

All values are refetched subject to the transaction isolation level and cursor sensitivity. If refreshRow is called after calling an updater method, but before calling the method updateRow, then the updates made to the row are lost. Calling the method refreshRow frequently will likely slow performance.

### 4.11.69 relative(row as Integer) as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Moves the cursor a relative number of rows, either positive or negative.
**Notes:** See the java documentation for details on java.sql.ResultSet.relative.

### 4.11.70 rowDeleted as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether a row has been deleted.
**Notes:** A deleted row may leave a visible ”hole” in a result set. This method can be used to detect holes in a result set. The value returned depends on whether or not this ResultSet object can detect deletions.
Returns:
true if a row was deleted and deletions are detected; false otherwise

### 4.11.71 rowInserted as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether the current row has had an insertion.
**Notes:** The value returned depends on whether or not this ResultSet object can detect visible inserts.
Returns:
true if a row has had an insertion and insertions are detected; false otherwise

### 4.11.72 rowUpdated as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves whether the current row has been updated.
**Notes:** The value returned depends on whether or not the result set can detect updates.
Returns:
true if both (1) the row has been visibly updated by the owner or another and (2) updates are detected

### 4.11.73 TYPE_FORWARD_ONLY as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating the type for a ResultSet object whose cursor may move only forward.

### 4.11.74 TYPE_SCROLL_INSENSITIVE as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating the type for a ResultSet object that is scrollable but generally not sensitive to changes made by others.

### 4.11.75 TYPE_SCROLL_SENSITIVE as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating the type for a ResultSet object that is scrollable and generally sensitive to changes made by others.

### 4.11.76 updateBlob(column as Integer, value as JavaBlobMBS)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a java.sql.Blob value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

column: the first column is 1, the second is 2, ...
value: the new column value
See also:

- 4.11.77 updateBlob(column as string, value as JavaBlobMBS) 256

### 4.11.77 updateBlob(column as string, value as JavaBlobMBS)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a java.sql.Blob value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

column: the name of the column
value: the new column value
See also:

- 4.11.76 updateBlob(column as Integer, value as JavaBlobMBS) 256

### 4.11.78 updateBoolean(column as Integer, value as boolean)

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a boolean value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are

called to update the database.
Parameters:
columnIndex - the first column is 1, the second is 2, ...
x - the new column value
See also:

- 4.11.79 updateBoolean(column as string, value as boolean) 257

### 4.11.79 updateBoolean(column as string, value as boolean)

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a boolean value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

Parameters:
columnName - the name of the column
x - the new column value
See also:

- 4.11.78 updateBoolean(column as Integer, value as boolean) 256

### 4.11.80 updateByte(column as Integer, value as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a byte value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.
Parameters:
columnIndex - the first column is 1, the second is 2, ...
x - the new column value
See also:

- 4.11.81 updateByte(column as string, value as Integer) 257

### 4.11.81 updateByte(column as string, value as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a byte value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The

updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

Parameters:
columnName - the name of the column
x - the new column value
See also:

- 4.11.80 updateByte(column as Integer, value as Integer) 257

### 4.11.82 updateBytes(column as Integer, Value as String)

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a byte array value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.
column: column index starting at 1.
Value: The new byte array as a string.
See also:

- 4.11.83 updateBytes(column as string, Value as String) 258

### 4.11.83 updateBytes(column as string, Value as String)

Plugin Version: 11.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a byte array value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.
column: The column name.
Value: The new byte array as a string.
See also:

- 4.11.82 updateBytes(column as Integer, Value as String) 258

### 4.11.84 updateClob(column as Integer, value as JavaClobMBS)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a java.sql.Clob value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are

called to update the database.

column: the first column is 1, the second is 2, ...
value: the new column value
See also:

- 4.11.85 updateClob(column as string, value as JavaClobMBS) 259

### 4.11.85 updateClob(column as string, value as JavaClobMBS)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a java.sql.Clob value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

column: the name of the column
value: the new column value
See also:

- 4.11.84 updateClob(column as Integer, value as JavaClobMBS) 258

### 4.11.86 updateDate(column as integer, value as JavaObjectMBS)

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates date value.
**Notes:** You need to create a java.sql.Date object yourself to use this.
See also:

- 4.11.87 updateDate(column as string, value as JavaObjectMBS) 259

### 4.11.87 updateDate(column as string, value as JavaObjectMBS)

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates date value.
**Notes:** You need to create a java.sql.Date object yourself to use this.
See also:

- 4.11.86 updateDate(column as integer, value as JavaObjectMBS) 259

### 4.11.88 updateDouble(column as Integer, value as Double)

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a double value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.
Parameters:
columnIndex - the first column is 1, the second is 2, ...
x - the new column value
See also:

- 4.11.89 updateDouble(column as string, value as Double) 260

### 4.11.89 updateDouble(column as string, value as Double)

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a double value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

Parameters:
columnName - the name of the column
x - the new column value
See also:

- 4.11.88 updateDouble(column as Integer, value as Double) 260

### 4.11.90 updateFloat(column as Integer, value as single)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a float value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

Parameters:
columnIndex - the first column is 1, the second is 2, ...
x - the new column value
See also:

- 4.11.91 updateFloat(column as string, value as single) 261

### 4.11.91 updateFloat(column as string, value as single)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a float value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

Parameters:
columnName - the name of the column
x - the new column value
See also:

- 4.11.90 updateFloat(column as Integer, value as single) 260

### 4.11.92 updateInt(column as Integer, value as Integer)

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with an int value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.
Parameters:
columnIndex - the first column is 1, the second is 2, ...
x - the new column value
See also:

- 4.11.93 updateInt(column as string, value as Integer) 261

### 4.11.93 updateInt(column as string, value as Integer)

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with an int value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

Parameters:
columnName - the name of the column
x - the new column value
See also:

- 4.11.92 updateInt(column as Integer, value as Integer) 261

### 4.11.94 updateLong(column as Integer, value as int64)

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a long value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.
Parameters:
columnIndex - the first column is 1, the second is 2, ...
x - the new column value
See also:

- 4.11.95 updateLong(column as string, value as int64) 262

### 4.11.95 updateLong(column as string, value as int64)

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a long value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

Parameters:
columnName - the name of the column
x - the new column value
See also:

- 4.11.94 updateLong(column as Integer, value as int64) 262

### 4.11.96 updateNull(column as Integer)

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gives a nullable column a null value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.
Parameters:
columnIndex - the first column is 1, the second is 2, ...
See also:

- 4.11.97 updateNull(column as string) 263

### 4.11.97 updateNull(column as string)

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a null value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

Parameters:
columnName - the name of the column
See also:

- 4.11.96 updateNull(column as Integer) 262

### 4.11.98 updateRow

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the underlying database with the new contents of the current row of this ResultSet object.
**Example:**

```
Var r as JavaResultSetMBS
Var db as JavaConnectionMBS // your database
Var s as JavaStatementMBS

s=db.createStatement
r=s.executeQuery("SELECT test_id from myTable")

if r<>Nil then
while r.NextRecord
r.updateInt("test_id",10+r.getInt("test_id"))
r.updateRow
wend
end if
```

**Notes:** This method cannot be called when the cursor is on the insert row.

See the java documentation for details on java.sql.ResultSet.updateRow.

### 4.11.99 updateShort(column as Integer, value as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a short value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.
Parameters:
columnIndex - the first column is 1, the second is 2, ...
x - the new column value
See also:

- 4.11.100 updateShort(column as string, value as Integer) 264

### 4.11.100 updateShort(column as string, value as Integer)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a short value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

Parameters:
columnName - the name of the column
x - the new column value
See also:

- 4.11.99 updateShort(column as Integer, value as Integer) 263

### 4.11.101 updateString(column as Integer, value as string)

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a String value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.

Parameters:
columnIndex - the first column is 1, the second is 2, ...
x - the new column value
See also:

- 4.11.102 updateString(column as string, value as string) 265

### 4.11.102 updateString(column as string, value as string)

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates the designated column with a String value.
**Notes:** The updater methods are used to update column values in the current row or the insert row. The updater methods do not update the underlying database; instead the updateRow or insertRow methods are called to update the database.
Parameters:
columnName - the name of the column
x - the new column value
See also:

- 4.11.101 updateString(column as Integer, value as string) 264

### 4.11.103 updateTime(column as integer, value as JavaObjectMBS)

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates time value.
**Notes:** You need to create a java.sql.Time object yourself to use this.
See also:

- 4.11.104 updateTime(column as string, value as JavaObjectMBS) 266

### 4.11.104 updateTime(column as string, value as JavaObjectMBS)

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates time value.
**Notes:** You need to create a java.sql.Time object yourself to use this.
See also:

- 4.11.103 updateTime(column as integer, value as JavaObjectMBS) 266

### 4.11.105 updateTimestamp(column as integer, value as JavaObjectMBS)

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates timestamp value.
**Notes:** You need to create a java.sql.Timestamp object yourself to use this.
See also:

- 4.11.106 updateTimestamp(column as string, value as JavaObjectMBS) 266

### 4.11.106 updateTimestamp(column as string, value as JavaObjectMBS)

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Updates timestamp value.
**Notes:** You need to create a java.sql.Timestamp object yourself to use this.
See also:

- 4.11.105 updateTimestamp(column as integer, value as JavaObjectMBS) 266

### 4.11.107 wasNull as boolean

Plugin Version: 8.3, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Reports whether the last column read had a value of SQL NULL.
**Notes:** See the java documentation for details on java.sql.ResultSet.wasNull.

### 4.11.108 Properties

### 4.11.109 FetchDirection as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gives a hint as to the direction in which the rows in this ResultSet object will be processed.
**Notes:** The initial value is determined by the Statement object that produced this ResultSet object. The fetch direction may be changed at any time.

Parameters:
direction - an int specifying the suggested fetch direction; one of ResultSet.FETCH_FORWARD, ResultSet.FETCH_REVERSE, or ResultSet.FETCH_UNKNOWN
(Read and Write computed property)

### 4.11.110 FetchSize as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gives the JDBC driver a hint as to the number of rows that should be fetched from the database when more rows are needed for this ResultSet object.
**Notes:** If the fetch size specified is zero, the JDBC driver ignores the value and is free to make its own best guess as to what the fetch size should be. The default value is set by the Statement object that created the result set. The fetch size may be changed at any time.

Parameters:
rows - the number of rows to fetch
(Read and Write computed property)

## 4.12 class JavaResultSetMetaDataMBS

### 4.12.1 class JavaResultSetMetaDataMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** An object that can be used to get information about the types and properties of the columns in a ResultSet object.
**Notes:** Subclass of the JavaObjectMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

### 4.12.2 Methods

### 4.12.3 columnNoNulls as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that a column does not allow NULL values.

### 4.12.4 columnNullable as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that a column allows NULL values.

### 4.12.5 columnNullableUnknown as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that the nullability of a column's values is unknown.

### 4.12.6 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 4.12.7 getCatalogName(Column as Integer) as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gets the designated column's table's catalog name.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
the name of the catalog for the table in which the given column appears or "" if not applicable

### 4.12.8 getColumnClassName(Column as Integer) as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns the fully-qualified name of the Java class whose instances are manufactured if the method ResultSet.getObject is called to retrieve a value from the column.
**Notes:** ResultSet.getObject may return a subclass of the class returned by this method.

Parameters:
column - the first column is 1, the second is 2, ...
Returns:
the fully-qualified name of the class in the Java programming language that would be used by the method ResultSet.getObject to retrieve the value in the specified column. This is the class name used for custom mapping.

### 4.12.9 getColumnCount as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns the number of columns in this ResultSet object.

### 4.12.10 getColumnDisplaySize(Column as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates the designated column's normal maximum width in characters.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
the normal maximum number of characters allowed as the width of the designated column

### 4.12.11 getColumnLabel(Column as Integer) as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gets the designated column's suggested title for use in printouts and displays.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
the suggested column title

### 4.12.12 getColumnName(Column as Integer) as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get the designated column's name.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
column name

### 4.12.13 getColumnType(Column as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the designated column's SQL type.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
SQL type from java.sql.Types

### 4.12.14 getColumnTypeName(Column as Integer) as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the designated column's database-specific type name.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
type name used by the database. If the column type is a user-defined type, then a fully-qualified type name is returned.

### 4.12.15 getPrecision(Column as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get the designated column's number of decimal digits.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
precision

### 4.12.16 getScale(Column as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gets the designated column's number of digits to right of the decimal point.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
scale

### 4.12.17 getSchemaName(Column as Integer) as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Get the designated column's table's schema.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
schema name or "" if not applicable

### 4.12.18 getTableName(Column as Integer) as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gets the designated column's table name.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
table name or "" if not applicable

### 4.12.19 isAutoIncrement(Column as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates whether the designated column is automatically numbered, thus read-only.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
true if so; false otherwise

### 4.12.20 isCaseSensitive(Column as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates whether a column's case matters.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
true if so; false otherwise

### 4.12.21 isCurrency(Column as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates whether the designated column is a cash value
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
true if so; false otherwise

### 4.12.22 isDefinitelyWritable(Column as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates whether a write on the designated column will definitely succeed.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
true if so; false otherwise

### 4.12.23 isNullable(Column as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates the nullability of values in the designated column.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
the nullability status of the given column; one of columnNoNulls, columnNullable or columnNullableUnknown

### 4.12.24 isReadOnly(Column as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates whether the designated column is definitely not writable.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
true if so; false otherwise

### 4.12.25 isSearchable(Column as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates whether the designated column can be used in a where clause.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
true if so; false otherwise

### 4.12.26 isSigned(Column as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates whether values in the designated column are signed numbers.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
true if so; false otherwise

### 4.12.27 isWritable(Column as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Indicates whether it is possible for a write on the designated column to succeed.
**Notes:** Parameters:
column - the first column is 1, the second is 2, ...
Returns:
true if so; false otherwise

# 4.13 class JavaRuntimeMBS

## 4.13.1 class JavaRuntimeMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The java class for runtime information.
**Notes:** Subclass of the JavaObjectMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

## 4.13.2 Methods

## 4.13.3 availableProcessors as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns the number of processors available to the Java virtual machine.
**Notes:** This value may change during a particular invocation of the virtual machine. Applications that are sensitive to the number of available processors should therefore occasionally poll this property and adjust their resource usage appropriately.

## 4.13.4 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

## 4.13.5 freeMemory as Int64

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns the amount of free memory in the Java Virtual Machine.
**Notes:** Calling the gc method may result in increasing the value returned by freeMemory.

## 4.13.6 gc

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Invokes the garbage collection.
**Notes:** Calls "System.gc" in Java.

### 4.13.7 maxMemory as Int64

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns the maximum amount of memory that the Java virtual machine will attempt to use.
**Notes:** If there is no inherent limit then the value Long.MAX_VALUE will be returned.

### 4.13.8 totalMemory as Int64

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Returns the total amount of memory in the Java virtual machine.
**Notes:** The value returned by this method may vary over time, depending on the host environment.
Note that the amount of memory required to hold an object of any given type may be implementation-dependent.

# 4.14 class JavaSavepointMBS

## 4.14.1 class JavaSavepointMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The representation of a savepoint, which is a point within the current transaction that can be referenced from the Connection.rollback method. When a transaction is rolled back to a savepoint all changes made after that savepoint are undone.
**Notes:** Savepoints can be either named or unnamed. Unnamed savepoints are identified by an ID generated by the underlying data source.
Subclass of the JavaObjectMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

## 4.14.2 Methods

## 4.14.3 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

## 4.14.4 getSavepointId as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the generated ID for the savepoint that this Savepoint object represents.

## 4.14.5 getSavepointName as string

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the name of the savepoint that this Savepoint object represents.

## 4.15 class JavaStatementMBS

### 4.15.1 class JavaStatementMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The object used for executing a static SQL statement and returning the results it produces.
**Notes:** By default, only one ResultSet object per Statement object can be open at the same time. Therefore, if the reading of one ResultSet object is interleaved with the reading of another, each must have been generated by different Statement objects. All execution methods in the Statement interface implicitly close a statment's current ResultSet object if an open one exists.
Subclass of the JavaObjectMBS class.
This is an abstract class. You can't create an instance, but you can get one from various plugin functions.
**Blog Entries**

- Prefetching records from databases
- MBS Xojo / Real Studio Plugins, version 15.1pr1

### 4.15.2 Methods

### 4.15.3 addBatch(sql as string)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Adds the given SQL command to the current list of commmands for this Statement object.
**Notes:** The commands in this list can be executed as a batch by calling the method executeBatch.

NOTE: This method is optional.

Parameters:
sql - typically this is a static SQL INSERT or UPDATE statement

### 4.15.4 cancel

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Cancels this Statement object if both the DBMS and driver support aborting an SQL statement.
**Notes:** This method can be used by one thread to cancel a statement that is being executed by another thread.

### 4.15.5 clearBatch

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Empties this Statement object's current list of SQL commands.
**Notes:** NOTE: This method is optional.

### 4.15.6 clearWarnings

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Clears all the warnings reported on this Statement object.
**Notes:** After a call to this method, the method getWarnings will return null until a new warning is reported for this Statement object.

### 4.15.7 close

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Releases this Statement object's database and JDBC resources immediately instead of waiting for this to happen when it is automatically closed.
**Notes:** It is generally good practice to release resources as soon as you are finished with them to avoid tying up database resources.
Calling the method close on a Statement object that is already closed has no effect.

Note: A Statement object is automatically closed when it is garbage collected. When a Statement object is closed, its current ResultSet object, if one exists, is also closed.

### 4.15.8 CLOSE_ALL_RESULTS as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that all ResultSet objects that have previously been kept open should be closed when calling getMoreResults.

### 4.15.9 CLOSE_CURRENT_RESULT as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that the current ResultSet object should be closed when calling getMoreResults.

### 4.15.10 Constructor

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The private constructor.

### 4.15.11 execute(sql as string) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Executes the given SQL statement, which may return multiple results.
**Notes:** In some (uncommon) situations, a single SQL statement may return multiple result sets and/or update counts. Normally you can ignore this unless you are (1) executing a stored procedure that you know may return multiple results or (2) you are dynamically executing an unknown SQL string.
The execute method executes an SQL statement and indicates the form of the first result. You must then use the methods getResultSet or getUpdateCount to retrieve the result, and getMoreResults to move to any subsequent result(s).

Parameters:
sql - any SQL statement

Returns:
true if the first result is a ResultSet object; false if it is an update count or there are no results
See also:

- 4.15.12 execute(sql as string, autoGeneratedKeys as Integer) as boolean 280

### 4.15.12 execute(sql as string, autoGeneratedKeys as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Executes the given SQL statement, which may return multiple results, and signals the driver that any auto-generated keys should be made available for retrieval.
**Notes:** The driver will ignore this signal if the SQL statement is not an INSERT statement.
In some (uncommon) situations, a single SQL statement may return multiple result sets and/or update counts. Normally you can ignore this unless you are (1) executing a stored procedure that you know may return multiple results or (2) you are dynamically executing an unknown SQL string.

The execute method executes an SQL statement and indicates the form of the first result. You must then use the methods getResultSet or getUpdateCount to retrieve the result, and getMoreResults to move to any

subsequent result(s).

Parameters:
sql - any SQL statement
autoGeneratedKeys - a constant indicating whether auto-generated keys should be made available for retrieval using the method getGeneratedKeys; one of the following constants: Statement.RETURN_GENERATED_KEYS or Statement.NO_GENERATED_KEYS
Returns:
true if the first result is a ResultSet object; false if it is an update count or there are no results
See also:

- 4.15.11 execute(sql as string) as boolean 280

### 4.15.13 executeBatch as Integer()

Plugin Version: 15.1, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Submits a batch of commands to the database for execution and if all commands execute successfully, returns an array of update counts.

### 4.15.14 executeQuery(sql as string) as JavaResultSetMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Executes the given SQL statement, which returns a single ResultSet object.
**Notes:** Parameters:
sql - an SQL statement to be sent to the database, typically a static SQL SELECT statement
Returns:
a ResultSet object that contains the data produced by the given query; never null

### 4.15.15 executeUpdate(Sql as string) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Executes the given SQL statement, which may be an INSERT, UPDATE, or DELETE statement or an SQL statement that returns nothing, such as an SQL DDL statement.
**Notes:** Parameters:
sql - an SQL INSERT, UPDATE or DELETE statement or an SQL statement that returns nothing
Returns:
either the row count for INSERT, UPDATE or DELETE statements, or 0 for SQL statements that return nothing
See also:

- 4.15.16 executeUpdate(Sql as string, autoGeneratedKeys as Integer) as Integer 282

### 4.15.16 executeUpdate(Sql as string, autoGeneratedKeys as Integer) as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Executes the given SQL statement and signals the driver with the given flag about whether the auto-generated keys produced by this Statement object should be made available for retrieval.
**Notes:** Parameters:
sql - must be an SQL INSERT, UPDATE or DELETE statement or an SQL statement that returns nothing
autoGeneratedKeys - a flag indicating whether auto-generated keys should be made available for retrieval; one of the following constants: Statement.RETURN_GENERATED_KEYS Statement.NO_GENERATED_KEYS
Returns:
either the row count for INSERT, UPDATE or DELETE statements, or 0 for SQL statements that return nothing
See also:

- 4.15.15 executeUpdate(Sql as string) as Integer 281

### 4.15.17 EXECUTE_FAILED as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that an error occured while executing a batch statement.

### 4.15.18 getGeneratedKeys as JavaResultSetMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves any auto-generated keys created as a result of executing this Statement object.
**Notes:** If this Statement object did not generate any keys, an empty ResultSet object is returned.

Returns:
a ResultSet object containing the auto-generated key(s) generated by the execution of this Statement object

### 4.15.19 getMoreResults as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Moves to this Statement object's next result, returns true if it is a ResultSet object, and implicitly closes any current ResultSet object(s) obtained with the method getResultSet.
**Notes:** There are no more results when the following is true:

```
// stmt is a Statement object
((stmt.getMoreResults() == false) && (stmt.getUpdateCount() == -1))
```

Returns:
true if the next result is a ResultSet object; false if it is an update count or there are no more results
See also:

- 4.15.20 getMoreResults(current as Integer) as boolean 283

### 4.15.20 getMoreResults(current as Integer) as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Moves to this Statement object's next result, deals with any current ResultSet object(s) according to the instructions specified by the given flag, and returns true if the next result is a ResultSet object.
**Notes:** There are no more results when the following is true:

```
// stmt is a Statement object
((stmt.getMoreResults() == false) && (stmt.getUpdateCount() == -1))
```

Parameters:
current - one of the following Statement constants indicating what should happen to current ResultSet objects obtained using the method getResultSet: Statement.CLOSE_CURRENT_RESULT, Statement.KEEP_CURRENT_RESULT, or Statement.CLOSE_ALL_RESULTS
Returns:
true if the next result is a ResultSet object; false if it is an update count or there are no more results
See also:

- 4.15.19 getMoreResults as boolean 282

### 4.15.21 getResultSet as JavaResultSetMBS

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the current result as a ResultSet object.
**Notes:** This method should be called only once per result.

Returns:
the current result as a ResultSet object or null if the result is an update count or there are no more results

### 4.15.22 getResultSetConcurrency as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the result set concurrency for ResultSet objects generated by this Statement object.
**Notes:** Returns:
either ResultSet.CONCUR_READ_ONLY or ResultSet.CONCUR_UPDATABLE

### 4.15.23 getResultSetHoldability as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the result set holdability for ResultSet objects generated by this Statement object.
**Notes:** Returns:
either ResultSet.HOLD_CURSORS_OVER_COMMIT or ResultSet.CLOSE_CURSORS_AT_COMMIT

### 4.15.24 getResultSetType as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the result set type for ResultSet objects generated by this Statement object.
**Notes:** Returns:
one of ResultSet.TYPE_FORWARD_ONLY, ResultSet.TYPE_SCROLL_INSENSITIVE, or ResultSet.TYPE_SCROLL_S
SITIVE

### 4.15.25 getUpdateCount as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Retrieves the current result as an update count; if the result is a ResultSet object or there are no more results, -1 is returned.
**Notes:** This method should be called only once per result.

Returns:
the current result as an update count; -1 if the current result is a ResultSet object or there are no more results

### 4.15.26 KEEP_CURRENT_RESULT as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that the current ResultSet object should not be closed when calling getMoreResults.

### 4.15.27 NO_GENERATED_KEYS as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that generated keys should not be made available for retrieval.

### 4.15.28 RETURN_GENERATED_KEYS as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that generated keys should be made available for retrieval.

### 4.15.29 setCursorName(name as string)

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Sets the SQL cursor name to the given String, which will be used by subsequent Statement object execute methods.
**Notes:** This name can then be used in SQL positioned update or delete statements to identify the current row in the ResultSet object generated by this statement. If the database does not support positioned update/delete, this method is a noop. To insure that a cursor has the proper isolation level to support updates, the cursor's SELECT statement should have the form SELECT FOR UPDATE. If FOR UPDATE is not present, positioned updates may fail.
Note: By definition, the execution of positioned updates and deletes must be done by a different Statement object than the one that generated the ResultSet object being used for positioning. Also, cursor names must be unique within a connection.

Parameters:
name - the new cursor name, which must be unique within a connection

### 4.15.30 SUCCESS_NO_INFO as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The constant indicating that a batch statement executed successfully but that no count of the number of rows it affected is available.

### 4.15.31 Properties

### 4.15.32 EscapeProcessing as boolean

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Whether escape processing is on or off.
**Notes:** If escape scanning is on (the default), the driver will do escape substitution before sending the SQL statement to the database. Note: Since prepared statements have usually been parsed prior to making this call, disabling escape processing for PreparedStatements objects will have no effect.
Parameters:
enable - true to enable escape processing; false to disable it
(Read and Write computed property)

### 4.15.33 FetchDirection as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The fetch direction.
**Notes:** Gives the driver a hint as to the direction in which rows will be processed in ResultSet objects created using this Statement object. The default value is ResultSet.FETCH_FORWARD.
Note that this method sets the default fetch direction for result sets generated by this Statement object. Each result set has its own methods for getting and setting its own fetch direction.

Parameters:
direction - the initial direction for processing rows
(Read and Write computed property)

### 4.15.34 FetchSize as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** Gives the JDBC driver a hint as to the number of rows that should be fetched from the database when more rows are needed.
**Notes:** The number of rows specified affects only result sets created using this statement. If the value specified is zero, then the hint is ignored. The default value is zero.
Parameters:
rows - the number of rows to fetch
(Read and Write computed property)

### 4.15.35 MaxFieldSize as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The maximum number of bytes that can be returned for character and binary column values in a ResultSet object produced by this Statement object.
**Notes:** This limit applies only to BINARY, VARBINARY, LONGVARBINARY, CHAR, VARCHAR, and LONGVARCHAR columns. If the limit is exceeded, the excess data is silently discarded.
(Read and Write computed property)

### 4.15.36 MaxRows as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The maximum number of rows that a ResultSet object produced by this Statement object can contain.
**Notes:** If this limit is exceeded, the excess rows are silently dropped.
(Read and Write computed property)

### 4.15.37 QueryTimeout as Integer

Plugin Version: 8.5, Platforms: macOS, Linux, Windows, Targets: Desktop, Console & Web.

**Function:** The number of seconds the driver will wait for a Statement object to execute. If the limit is exceeded, a SQLException is thrown.
**Notes:** Returns:
the current query timeout limit in seconds; zero means there is no limit
(Read and Write computed property)

# Chapter 5

# List of Questions in the FAQ

# Chapter 6

# The FAQ

### 6.0.1 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