

MBS Python 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 Python Plugin

0.2 Content

- 1 List of all topics 3
- 2 List of all classes 5
- 3 All items in this plugin 7

Chapter 1

List of Topics

| | |
|---|----|
| • 3 Python | 7 |
| – 3.2.1 class PythonMBS | 8 |
| * 3.2.3 AddCustomFunction(FunctionName as String) | 8 |
| * 3.2.4 AddCustomFunction(FunctionName as String, theDelegate as CustomFunctionCalledMBS) | 9 |
| * 3.2.5 AddSysPath(Path as String) | 9 |
| * 3.2.6 Clear | 9 |
| * 3.2.7 Constructor | 10 |
| * 3.2.8 DeleteValue(Key as String) | 10 |
| * 3.2.9 Evaluate(Expression as String) as Variant | 10 |
| * 3.2.10 HasValue(Key as String) as Boolean | 11 |
| * 3.2.11 Keys as Variant() | 11 |
| * 3.2.12 Load(File as FolderItem) as Boolean | 12 |
| * 3.2.13 Load(Path as String) as Boolean | 12 |
| * 3.2.14 Run(Script as String) as Variant | 12 |
| * 3.2.15 RunFile(File as FolderItem) as Variant | 13 |
| * 3.2.17 ExecPrefix as String | 13 |
| * 3.2.18 Handle as Integer | 14 |
| * 3.2.19 LibraryBuildInfo as String | 14 |
| * 3.2.20 LibraryCompiler as String | 14 |
| * 3.2.21 LibraryCopyright as String | 15 |
| * 3.2.22 LibraryError as String | 15 |
| * 3.2.23 LibraryPlatform as String | 15 |
| * 3.2.24 LibraryVersion as String | 15 |
| * 3.2.25 Loaded as Boolean | 16 |
| * 3.2.26 MonotonicTime as Int64 | 16 |
| * 3.2.27 Path as String | 16 |

| | |
|---|----|
| * 3.2.28 Prefix as String | 17 |
| * 3.2.29 ProgramFullPath as String | 17 |
| * 3.2.30 ProgramName as String | 17 |
| * 3.2.31 PythonHome as String | 17 |
| * 3.2.32 RecursionLimit as Integer | 18 |
| * 3.2.33 Time as Int64 | 18 |
| * 3.2.34 ValueCount as Integer | 18 |
| * 3.2.35 Value(Key as String) as Variant | 19 |
| * 3.2.37 Flush | 19 |
| * 3.2.38 FunctionCall(FunctionName as String, Args() as Variant) as Variant | 19 |
| * 3.2.39 Input(Args() as Variant) as String | 20 |
| * 3.2.40 Print(text as String) | 20 |
| * 3.2.42 CustomFunctionCalledMBS(Args() as Variant) as Variant | 21 |

Chapter 2

List of all classes

- PythonExceptionMBS 7
- PythonMBS 8

Chapter 3

Python

3.1 class PythonExceptionMBS

3.1.1 class PythonExceptionMBS

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: An exception to report problems in Python code.

Notes: See message property for details.

Subclass of the RuntimeException class.

3.2 class PythonMBS

3.2.1 class PythonMBS

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: The class to use Python code in Xojo applications.

Example:

```
Var p As New PythonMBS
```

```
Var r As Variant = p.Evaluate("1+2")
```

```
Print "Result: "+r
```

```
Break
```

Notes: MBS Plugin does some exception handling for you as well as type conversion.

We convert variants to/from python as much as possible, including arrays and dictionaries.

Blog Entries

- [AI Development with the MBS Xojo Plugins](#)
- [MBS Xojo Plugins, version 25.5pr2](#)
- [News from the MBS Xojo Plugins in Version 25.4](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 25.4](#)
- [News from the MBS Xojo Plugins Version 25.2](#)
- [News from the MBS Xojo Plugins in Version 25.1](#)
- [News from the MBS Xojo Plugins Version 24.4](#)
- [News from the MBS Xojo Plugins Version 24.3](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 24.3](#)
- [Using Python in Xojo](#)

Xojo Developer Magazine

- [23.6, page 10: News](#)

3.2.2 Methods

3.2.3 AddCustomFunction(FunctionName as String)

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Adds a custom function to Python.

Example:

```
Var p As New MyPythonMBS
```

```
p.AddCustomFunction "Test"
p.AddCustomFunction "UserName"
p.AddCustomFunction "Sum"
```

```
Var r1 As Variant = p.Evaluate("Test('test')")
Var r2 As Variant = p.Evaluate("UserName()")
Var r3 As Variant = p.Evaluate("Sum(2,3,6)")
```

Break

Notes: You can define multiple functions all with different names.

Function names are case sensitive.

See also:

- 3.2.4 AddCustomFunction(FunctionName as String, theDelegate as CustomFunctionCalledMBS) 9

3.2.4 AddCustomFunction(FunctionName as String, theDelegate as CustomFunctionCalledMBS)

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Adds a custom function to call the delegate instead of the event.

See also:

- 3.2.3 AddCustomFunction(FunctionName as String) 8

3.2.5 AddSysPath(Path as String)

Plugin Version: 25.1, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Adds a file path to the system path list.

Notes: This allows you to add the path to the current virtual environment.

3.2.6 Clear

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Clears the variables in the Python environment.

Example:

```
Var p As New PythonMBS
```

```
p.Value("Hello") = 123.0  
p.Clear
```

```
Var check As Boolean = p.HasValue("Hello")
```

```
Break
```

3.2.7 Constructor

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: The constructor.

3.2.8 DeleteValue(Key as String)

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Deletes a variable in the Python environment.

Example:

```
Var p As New PythonMBS
```

```
p.Value("Hello") = 123.0  
p.DeleteValue "Hello"
```

```
Var check As Boolean = p.HasValue("Hello")
```

```
Break
```

3.2.9 Evaluate(Expression as String) as Variant

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Evaluates an expression in Python.

Example:

```
Var p As New PythonMBS
```

```
Var r As Variant = p.Evaluate("1+2")  
Print "Result: "+r
```

Break

Notes: Pass an expression to evaluate with Python.
Returns the result of the expression.

May raise a PythonExceptionMBS exception.

3.2.10 HasValue(Key as String) as Boolean

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Checks whether a variable is set for the Python environment.

Example:

```
Var p As New PythonMBS
```

```
p.Value("Hello") = 123.0
Var hasIt As Boolean = p.HasValue("Hello")
Var hasNothing As Boolean = p.HasValue("other")
```

Break

Notes: Returns true if variable is set.

3.2.11 Keys as Variant()

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Queries the list of names for values.

Example:

```
Var p As New MyPythonMBS
```

```
p.AddCustomFunction "Test"
p.Value("Other") = 123
```

```
For Each key As Variant In p.keys
Print "key: "+key
Next
```

Break

Notes: This includes the registered custom functions.

3.2.12 Load(File as FolderItem) as Boolean

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Loads the Python library.

Notes: You may need to check what version you have as the paths usually contain version numbers.

See also:

- 3.2.13 Load(Path as String) as Boolean

12

3.2.13 Load(Path as String) as Boolean

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Loads the Python library.

Example:

```
const path = "/opt/homebrew/Cellar/python@3.12/3.12.3/Frameworks/Python.framework"
```

```
Var loaded As Boolean = PythonMBS.Load(path)
```

```
Print "Error: "+PythonMBS.LibraryError
```

```
Print "Version: "+PythonMBS.LibraryVersion
```

Notes: You may need to check what version you have as the paths usually contain version numbers.

e.g.

MacOS: /opt/homebrew/Cellar/python@3.12/3.12.3/Frameworks/Python.framework

Windows: C:\Users\cs\AppData\Local\Programs\Python\Python312\python312.dll

Linux: libpython3.11.so

See also:

- 3.2.12 Load(File as FolderItem) as Boolean

12

3.2.14 Run(Script as String) as Variant

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Runs a Python script.

Example:

```
Var p As New MyPythonMBS
```

```
// calls print event
Call p.run("print('Hello')")
```

Notes: Pass the script text to run.

Returns the result of the script.

3.2.15 RunFile(File as FolderItem) as Variant

Plugin Version: 24.4, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Runs a Python script file.

Notes: Returns the result of the script.

3.2.16 Properties

3.2.17 ExecPrefix as String

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Return the exec-prefix for installed platform-dependent files.

Notes: This is derived through a number of complicated rules from the program name set with `Py_SetProgramName()` and some environment variables; for example, if the program name is `'/usr/local/bin/python'`, the exec-prefix is `'/usr/local'`.

The value is available to Python code as `sys.exec_prefix`. It is only useful on Unix.

Background: The exec-prefix differs from the prefix when platform dependent files (such as executables and shared libraries) are installed in a different directory tree. In a typical installation, platform dependent files may be installed in the `/usr/local/plat` subtree while platform independent may be installed in `/usr/local`.

Generally speaking, a platform is a combination of hardware and software families, e.g. Sparc machines running the Solaris 2.x operating system are considered the same platform, but Intel machines running Solaris 2.x are another platform, and Intel machines running Linux are yet another platform. Different major revisions of the same operating system generally also form different platforms. Non-Unix operating systems are a different story; the installation strategies on those systems are so different that the prefix and exec-prefix are meaningless, and set to the empty string. Note that compiled Python bytecode files are platform independent (but not independent from the Python version by which they were compiled!).

System administrators will know how to configure the mount or automount programs to share /usr/local between platforms while having /usr/local/plat be a different filesystem for each platform.
(Read only property)

3.2.18 Handle as Integer

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: The internal object reference.

Notes: (Read and Write property)

3.2.19 LibraryBuildInfo as String

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Return information about the sequence number and build date and time of the current Python interpreter instance.

Example:

MessageBox PythonMBS.LibraryBuildInfo

Notes: The value is available to Python code as part of the variable `sys.version`.

e.g. returns "main, Dec 3 2024, 17:59:52"
(Read only property)

3.2.20 LibraryCompiler as String

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Return an indication of the compiler used to build the current Python version, in square brackets.

Example:

MessageBox PythonMBS.LibraryCompiler

Notes: The value is available to Python code as part of the variable `sys.version`.

e.g. returns [Clang 16.0.0 (clang-1600.0.26.4)]
(Read only property)

3.2.21 LibraryCopyright as String

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Return the official copyright string for the current Python version.

Example:

```
MessageBox PythonMBS.LibraryCopyright
```

Notes: The value is available to Python code as `sys.copyright`.
(Read only property)

3.2.22 LibraryError as String

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: The error message from Load methods in case of the error.

Notes: (Read only property)

3.2.23 LibraryPlatform as String

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Return the platform identifier for the current platform.

Example:

```
MessageBox PythonMBS.LibraryPlatform
```

Notes: On Unix, this is formed from the „official“ name of the operating system, converted to lower case, followed by the major revision number; e.g., for Solaris 2.x, which is also known as SunOS 5.x, the value is 'sunos5'. On macOS, it is 'darwin'. On Windows, it is 'win'.

The value is available to Python code as `sys.platform`.

Value is e.g. "darwin" on macOS.

(Read only property)

3.2.24 LibraryVersion as String

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Queries the version text of the Python library.

Notes: Returns e.g. "3.12.3 (main, Apr 9 2024, 08:09:14) [Clang 15.0.0 (clang-1500.3.9.4)] "

(Read only property)

3.2.25 Loaded as Boolean

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Queries whether Python libraries have been loaded.

Notes: Returns true if loaded or false if not.

(Read only property)

3.2.26 MonotonicTime as Int64

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Read the monotonic clock.

Example:

MessageBox PythonMBS.MonotonicTime.ToString

Notes: e.g. 1221942957636375

(Read only property)

3.2.27 Path as String

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Return the default module search path.

Notes: This is computed from the program name (set by assigning ProgramName) and some environment variables. The returned string consists of a series of directory names separated by a platform dependent delimiter character. The delimiter character is ':' on Unix and macOS, ';' on Windows.

The list `sys.path` is initialized with this value on interpreter startup; it can be (and usually is) modified later to change the search path for loading modules.

e.g.

```
/opt/homebrew/Cellar/python@3.12/3.12.3/Frameworks/Python.framework/Versions/3.12/lib/python312.zip:/opt/homebrew/Cellar/python@3.12/3.12.3/Frameworks/Python.framework/Versions/3.12/lib/python3.12:/opt/homebrew/Cellar/python@3.12/3.12.3/Frameworks/Python.framework/Versions/3.12/lib/python3.12/lib-dynload
```

(Read only property)

3.2.28 Prefix as String

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Return the prefix for installed platform-independent files.

Notes: This is derived through a number of complicated rules from the program name set with `Py_SetProgramName()` and some environment variables; for example, if the program name is `'/usr/local/bin/python'`, the prefix is `'/usr/local'`.

The value is available to Python code as `sys.prefix`. It is only useful on Unix.

(Read only property)

3.2.29 ProgramFullPath as String

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Return the full program name of the Python executable

Notes: This is computed as a side-effect of deriving the default module search path from the program name (set by assignment to `ProgramName`).

The value is available to Python code as `sys.executable`.

This is usually the path to your Xojo application.

(Read only property)

3.2.30 ProgramName as String

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: The program name.

Notes: Can be set before initializing Python, otherwise the default will be `"python"`.

(Read and Write property)

3.2.31 PythonHome as String

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: The default "home" path.

Notes: The value is set by a previous with setting `PythonHome` property, or the value is the value of the `PYTHONHOME` environment variable if it is set.

(Read and Write property)

3.2.32 RecursionLimit as Integer

Plugin Version: 25.4, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: The recursion limit.

Notes: Default is 1000.

(Read and Write property)

3.2.33 Time as Int64

Plugin Version: 25.2, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Read the ,Äúwall clock,Äò time.

Example:

```
MessageBox PythonMBS.Time.ToString
```

Notes: Return the time in seconds since the epoch as a floating-point number. The handling of leap seconds is platform dependent. On Windows and most Unix systems, the leap seconds are not counted towards the time in seconds since the epoch. This is commonly referred to as Unix time.

e.g. 1743330201673808000

(Read only property)

3.2.34 ValueCount as Integer

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Queries number of variables.

Example:

```
Var p As New PythonMBS
```

```
Print "ValueCount: " + p.ValueCount.ToString
```

```
p.Value("Hello") = 123.0
```

```
Print "ValueCount: " + p.ValueCount.ToString
```

```
p.DeleteValue "Hello"
```

```
Print "ValueCount: " + p.ValueCount.ToString
```

Notes: See Keys method to get list of names.

(Read only property)

3.2.35 Value(Key as String) as Variant

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: Get or set a local variable in Python.

Example:

```
Var p As New MyPythonMBS
```

```
p.Value("Hello") = 123.0
Var d As Double = p.Value("Hello")
```

```
break
```

Notes: Data types are converted as far as possible.
(Read and Write computed property)

3.2.36 Events

3.2.37 Flush

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: The flush event sent when print is done.

3.2.38 FunctionCall(FunctionName as String, Args() as Variant) as Variant

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: The event raised if a custom function is called.

Example:

```
Function FunctionCall(FunctionName as String, Args() as Variant) Handles FunctionCall as Variant
Print CurrentMethodName
```

```
Select Case FunctionName
Case "Test"
Return Args(0)
case "UserName"
Return SystemInformationMBS.UserName
Case "Sum"
```

```

Var d As Double = 0

For Each v As Double In args
d = d + v
Next

Return d
End Select

Exception r As RuntimeException
Return Introspection.GetType(r).name+": "+r.message

End Function

```

Notes: Please use try/catch to catch any exceptions in your functions. Check with the name which function is called. Function names are case sensitive.

3.2.39 Input(Args() as Variant) as String

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: The input event.

Notes: Called when the input function is called in Python.

Please return whatever text to pass back to the Python script.

3.2.40 Print(text as String)

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: The event called when print function is used in Python.

Example:

```

Sub Print(text as String) Handles Print
Print CurrentMethodName
Print "received: " + Text
End Sub

```

3.2.41 Delegates

3.2.42 CustomFunctionCalledMBS(Args() as Variant) as Variant

Plugin Version: 24.3, Platforms: macOS, Linux, Windows, iOS, Targets: Desktop, Console & Web.

Function: The delegate for custom functions.