

# Query MBS Version from plugin in container

If you use automatic plugin installation scripts, you may have the MBS Plugin in a container field in FileMaker. But how to automatically know the version number?

## Windows

We got two short scripts to query the version from a container:

```
# get temp path
Set Variable [ $path ; Value: MBS( "Path.AddPathComponent";
MBS( "Folders.UserTemporary" ); "MBS.dll" ) ]
# write DLL file to disk
Set Variable [ $r ; Value: MBS( "Container.WriteFile"; Install Plugin Update if
needed::Plugin File Win 64bit; $path ) ]
# Query file information
Set Variable [ $version ; Value: MBS( "Files.FileInfo"; $Path; "version") ]
# Delete file
Set Variable [ $version ; Value: MBS( "Files.Delete"; $Path ) ]
```

As you see this script gets a temp path, writes the plugin there with DLL file extension and queries file information. Since MBS Plugin has all metadata in the right headers for DLL handler, the file information can be read by OS functions.

Since this uses OS functions, it needs to run on Windows. Technically we could also do something cross platform, but that would need to read file in memory, scan for the version entry and return it.

## macOS

For macOS, you can use our [Archive](#) functions to get the info.plist file and then read it via [XML](#) functions:

```
# extract info file with metadata
Set Variable [ $InfoFile ; Value: MBS( "Archive.ExtractFile"; Install Plugin Update if
needed::Plugin File Mac; "MBS.fplugin/Contents/Info.plist" ) ]
# get it as text
Set Variable [ $InfoText ; Value: MBS( "Container.GetText"; $InfoFile; "FILE") ]
# query version entry in XML
Set Variable [ $version ; Value: MBS( "XML.GetPathValue"; $InfoText;
"plist.dict.string[6]") ]
Show Custom Dialog [ "Version of MBS Plugin" ; $version ]
```

This doesn't need a temporary file and may actually be workable on other platforms.

## Linux

On Linux the libraries don't contain version information as far as we see. But our plugin contains a version string like this "MBS FileMaker Plugin, version 11.2.0.03, built Apr 16 2021". So we can look for the prefix, take the version behind and cut at the first comma. Here is a sample script:

```
Set Variable [ $InfoText ; Value: MBS( "Container.GetText"; Install Plugin Update if needed::Plugin File Linux; "FILE") ]
# text like this: MBS FileMaker Plugin, version 11.2.0.03, built Apr 16 2021
Set Variable [ $pos ; Value: Position($InfoText; "MBS FileMaker Plugin, version"; 1; 1) ]
Set Variable [ $version ; Value: Middle ( $InfoText ; $pos + 30 ; 12 ) ]
# if we have comma, remove it
Set Variable [ $pos ; Value: Position($version; ","; 1; 1) ]
If [ $pos > 0 ]
    Set Variable [ $version ; Value: Left ( $version ; $pos-1 ) ]
End If
Show Custom Dialog [ "MBS Version" ; $version ]
```

## Windows, second try

Since we got it for Linux with finding version directly in the text, we can use our recent [RegEx.Extract](#) function to get the value directly from the file:

```
# extract plugin file
Set Variable [ $InfoText ; Value: MBS( "Container.GetText"; Install Plugin Update if needed::Plugin File Win 64bit; "FILE"; "Windows") ]
# get version as text
Set Variable [ $version ; Value: MBS( "RegEx.Extract"; $InfoText; "MBS FileMaker Plugin (\d+\.\d+\.\d+\.\d+);" ; "\1"; "greedy, caseless" ) ]
Show Custom Dialog [ "MBS Version" ; $version ]
```

The regular expression looks for the right prefix and then looks for four groups of digits separated by dots. The greedy option is needed to get the last digit on the end.

## Linux, second try

The RegEx functions also work on Linux, so we can simplify the script to this:

```
# extract plugin file
Set Variable [ $InfoText ; Value: MBS( "Container.GetText"; Install Plugin Update if needed::Plugin File Linux; "FILE") ]
# get version as text
Set Variable [ $version ; Value: MBS( "RegEx.Extract"; $InfoText; "MBS FileMaker Plugin, version (\d+\.\d+\.\d+\.\d+);" ; "\1"; "greedy, caseless" ) ]
Show Custom Dialog [ "MBS Version" ; $version ]
```

## macOS, second try

The RegEx functions also work on Linux, so we can simplify the script to this:

```
# extract info file with metadata
```

```
Set Variable [ $InfoFile ; Value: MBS( "Archive.ExtractFile"; Install Plugin Update if  
needed::Plugin File Mac; "MBS.fmplugin/Contents/Info.plist" ) ]
```

```
# get version as text
```

```
Set Variable [ $version ; Value: MBS( "RegEx.Extract"; $infoText;  
"<key>CFBundleVersion</key>\s*<string>(.)</string>"; "\1"; "greedy, caseless" ) ]  
Show Custom Dialog [ "MBS Version" ; $version ]
```

This now also finds the right version number and show us the version. May also work for other plugins with an info.plist.

Please try and do not hesitate to contact us with your questions.