Installation

Installation instructions for the Monkeybread Software FileMaker Plugin.

Requirements 2

Links 4

macOS 5

- Get files 5
- Install 6
- After installation 7

Windows 8

- Version notes 8
- Get files 8
- Install 9
- Alternate Install Location FileMaker 11
- After installation 12

Install via Script 13

Demo Limitations 15

Server 16

DynaPDF 21

LibXL 22

Examples 23

Preferences 24

Registration 25

- Trial time 25
- After you bought a serial number 25

Troubleshooting 26

- Plugin not loading 26
- Duplicate Installation 26
- Plugin deactivated 27
- Errors with Installation by Script 27

Contact 28

Requirements

We kindly ask you to take backup and testing seriously and avoid common pitfalls.

Backups

Before you install our plugin new or as an update, please make a backup of your databases. Best is to have an automated hourly backup to make sure you have a copy of all the databases, external files and the current software, so in case something breaks, you could go back.

You should have:

- Real Time backups.
  An application like Backblaze watches for new files and uploads them directly. This allows you to have it keep multiple versions and provide a history.

- Hourly backups
  e.g. Time Machine on a Mac or the built-in backup for FileMaker Server can do an hourly backups to a folder.

- Daily backups
  to another computer. In general you should have at least 3 backups in 2 different locations.

- Monthly backups
  e.g. rotating various backup disks between a safe in a bank and the office.

Folders with text files like source code or configuration files can also get tracked with git or svn. Keep versions and changes automatically by running a script automatically hourly/daily to commit any changes with a timestamp.

Testing Requirements

We kindly ask you to test that every script does what you intend to do. Don't just type a script and run it. Use the debugger to step through and see if all the functions return what you expect them to return. Use our Trace function to log calls and look for error codes.

Common problems

- A function may have worked with the old plugin and silently failed in some cases without reporting an error. The updated plugin now reports an error and that breaks the logic of your scripts.

- A function got a new parameter and your function calls contains a value for that parameter. In the previous version the extra parameter was ignored, but now it has an effect.

- You update to a new version of FileMaker. An old plugin doesn't load or is missing, so all calls to plugin functions return "?". There is the risk, that your scripts have a logical error, which interprets an error you get as something different and does actions based on this.

- After an update you may still have scripts with an old registration code. You may see dialogs telling you that the license is outdated even if you updated one script. But depending on which script runs first, the old license dialog may show.

- You updated the plugin, but forgot to update DynaPDF, LibXL or other library files. Then the new plugin misses functionality in the old libraries or the function crashes as data structures changed. We try to prevent this with various checks.

- You write SQL statements. A small typo in the statement may change or delete thousands of records. While developing you may not see the change or not trigger it, but in production you may damage your data.

In general when a client reports a problem, we ask for logs with our trace functions, the debug log from CURL if that is used or for a copy of the script to read.

Links

Monkeybread Software Website:

https://www.monkeybreadsoftware.com/

MBS FileMaker Plugin Website:

https://www.monkeybreadsoftware.com/filemaker/

Plugin Reference

https://www.mbsplugins.eu

Plugin Guides

https://www.monkeybreadsoftware.com/filemaker/guide/

FAQ

https://www.monkeybreadsoftware.com/filemaker/faq.shtml

Blog with tips and new versions and announcements:

https://www.mbsplugins.de/filemaker

Mailing list for discussion, new versions and announcements:

https://www.monkeybreadsoftware.com/filemaker/mailinglist.shtml

macOS

Get files

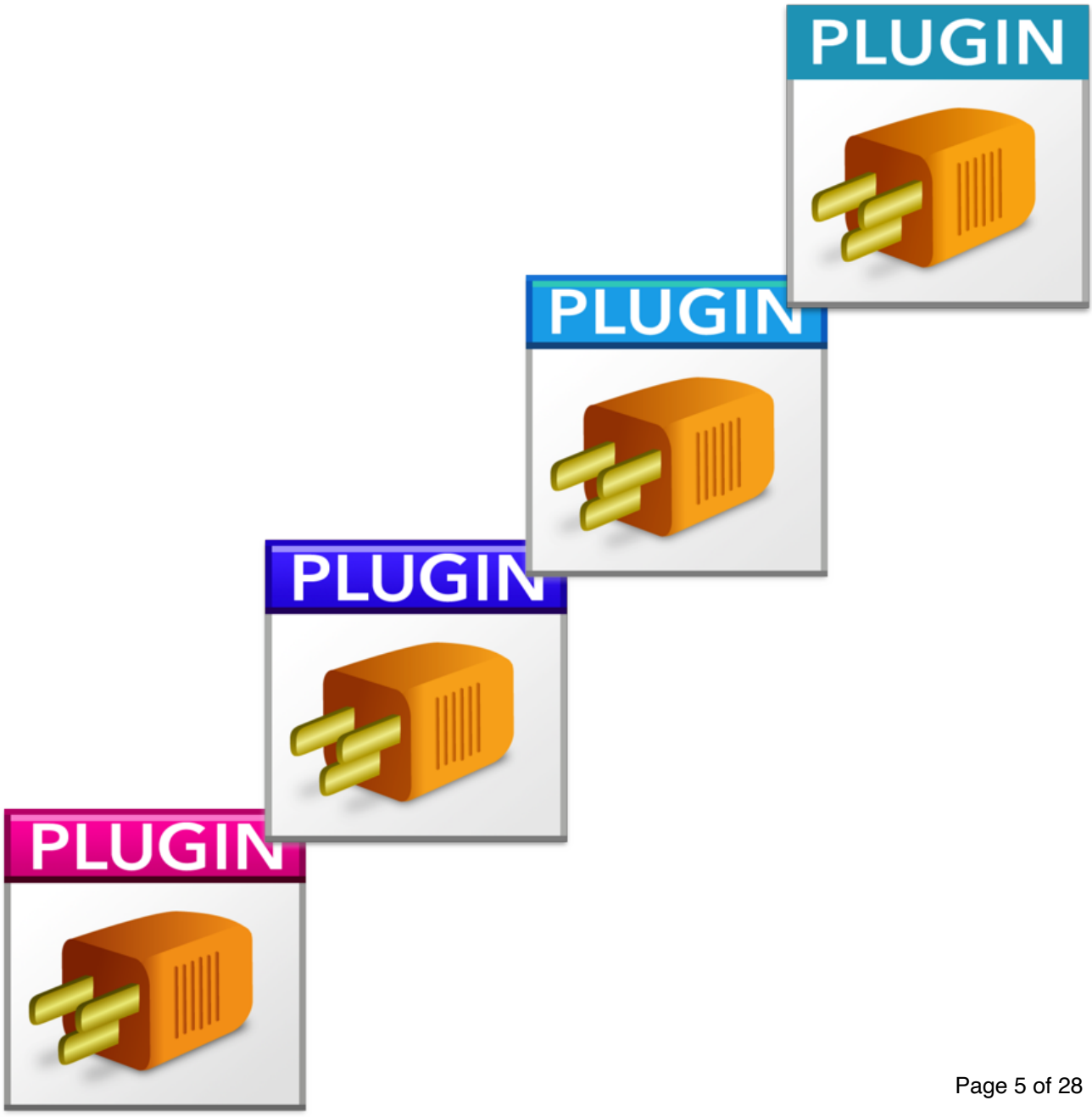
You first download the files needed. For FileMaker you need at least the „MBS.fmplugin“ file.

The file can be downloaded from https://www.monkeybreadsoftware.com/filemaker/ where you can also find a link to the download page.

There we have the current release version, older versions and newer prerelease versions of future plugins.

The help files on the website are often newer than the one coming with the plugin download and show already features for the next plugin version.

You can manually copy the plugin file MBS.fmplugin (may show with folder icon) into the Extensions folder for FileMaker or use the Install Plugin Script Step.



Install

To install, you drop the plugin file into the Extensions folder.

On the right side you see a picture from the ~/Library/Application Support/FileMaker/ folder on one of our Macs.

You can install different plugins for different versions in the various Extensions folders.

We recommend to install our plugin into

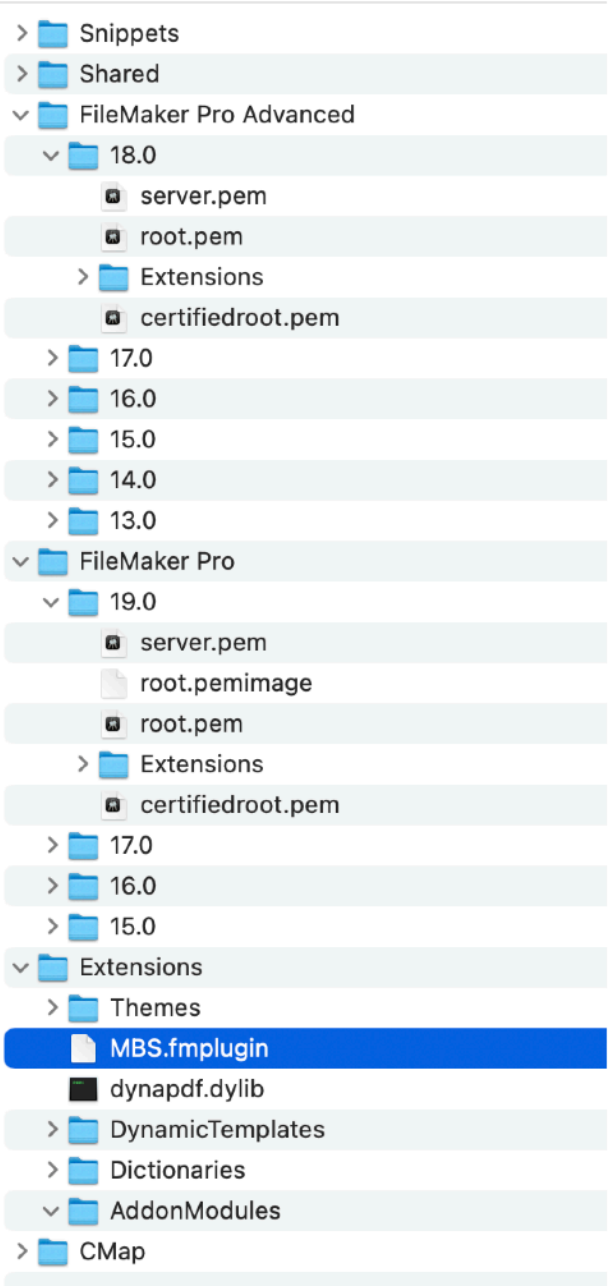
~/Library/Application Support/FileMaker/Extensions

Since that is a shared folder for **all** versions.

The Install Plugin Script Step installs the plugin in the version specific folder.

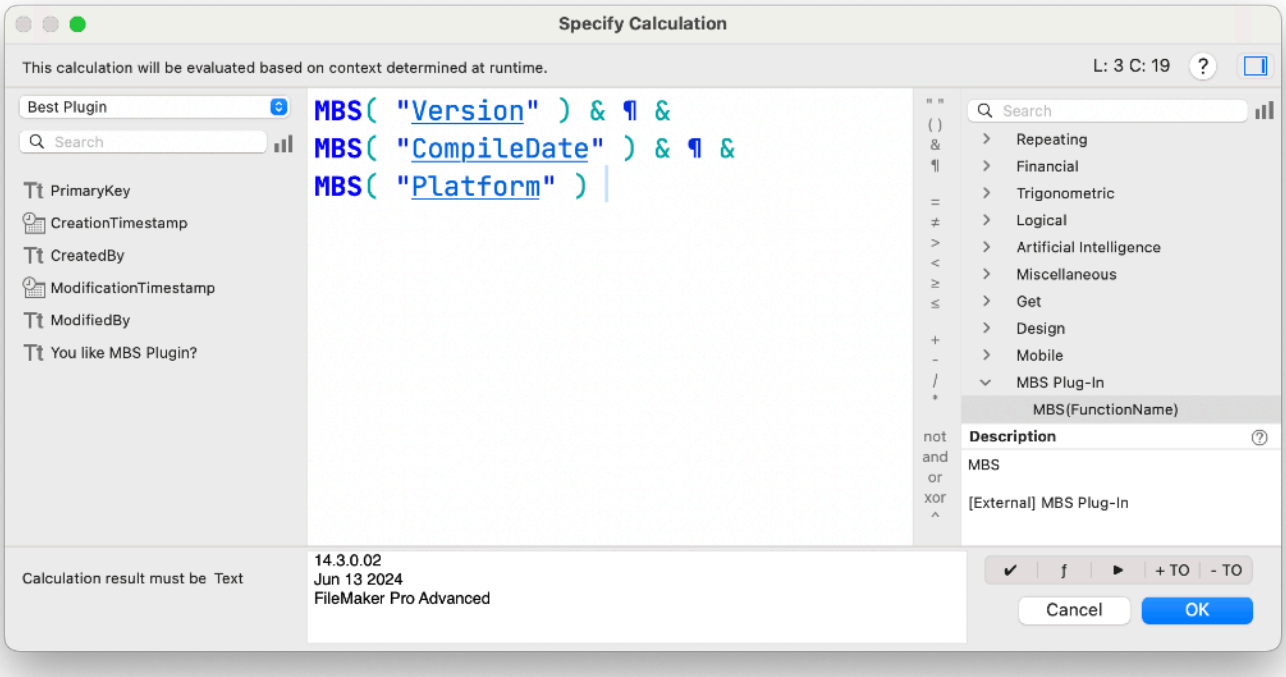
If you have plugins in both folders, you may get an error about two plugins installed.

Use the button in preferences dialog in FileMaker to easily find the right folder for your version.



After installation

To test whether the plugins have been installed, you can launch FileMaker, go to the calculation editor and see the MBS function on the external functions:



As you see above, the MBS function is listed for external functions. Also you can run the MBS("Version") command to query the version of the plugin you have. And finally the MBS function should be colored blue.

Windows

Version notes

The MBS Plugins should run fine on Windows 7 and newer with FileMaker 7.0 and newer.

Get files

You first download the files needed. For FileMaker on Windows, please take MBS.fmx64 plugin file. Older 32-bit versions of FileMaker may use the MBS.fmx file. You can put both in the folder and FileMaker picks the right one.

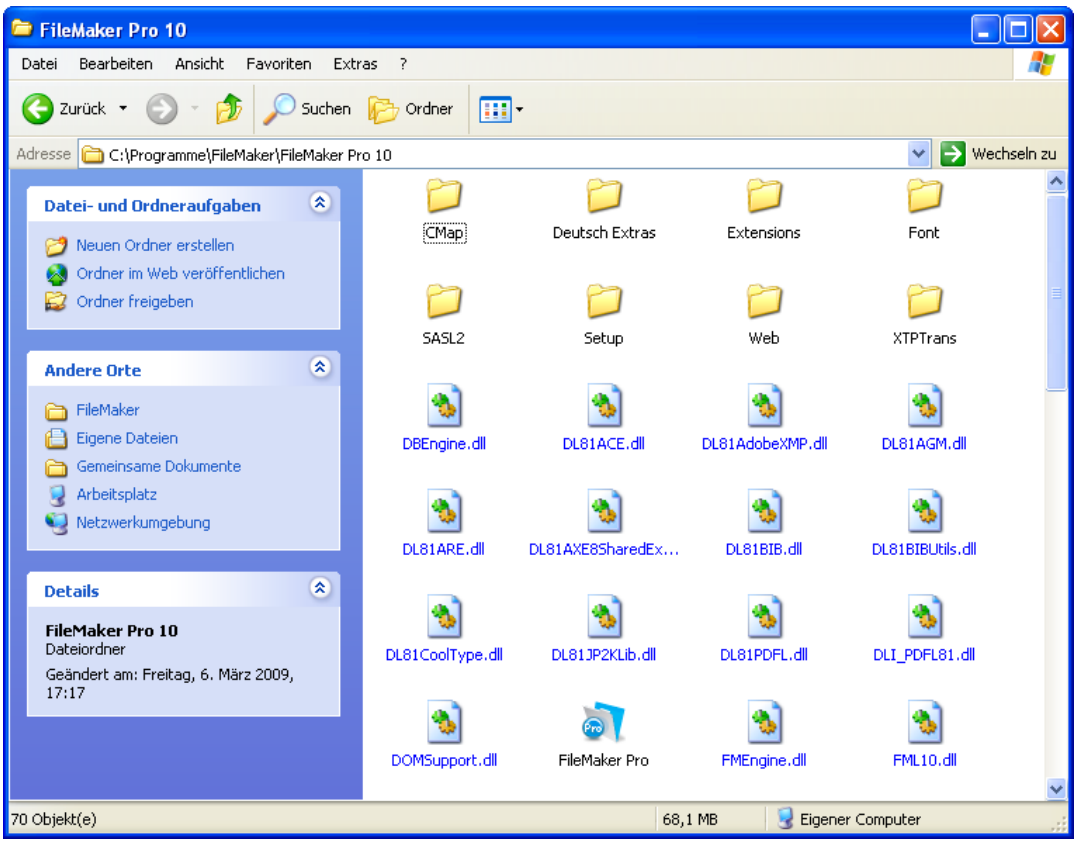
The file can be downloaded from https://www.monkeybreadsoftware.de/filemaker/ where you can also find a link to the download page.

There we have the current release version, older versions and newer prerelease versions of future plugins.

The help files on the website are often newer than the one coming with the plugin download and show already features for the next plugin version.

There is a separate download for older Windows versions like Windows 7, 8 and Server 2012.

First locate the plugin files and locate your FileMaker folder which may look like this:

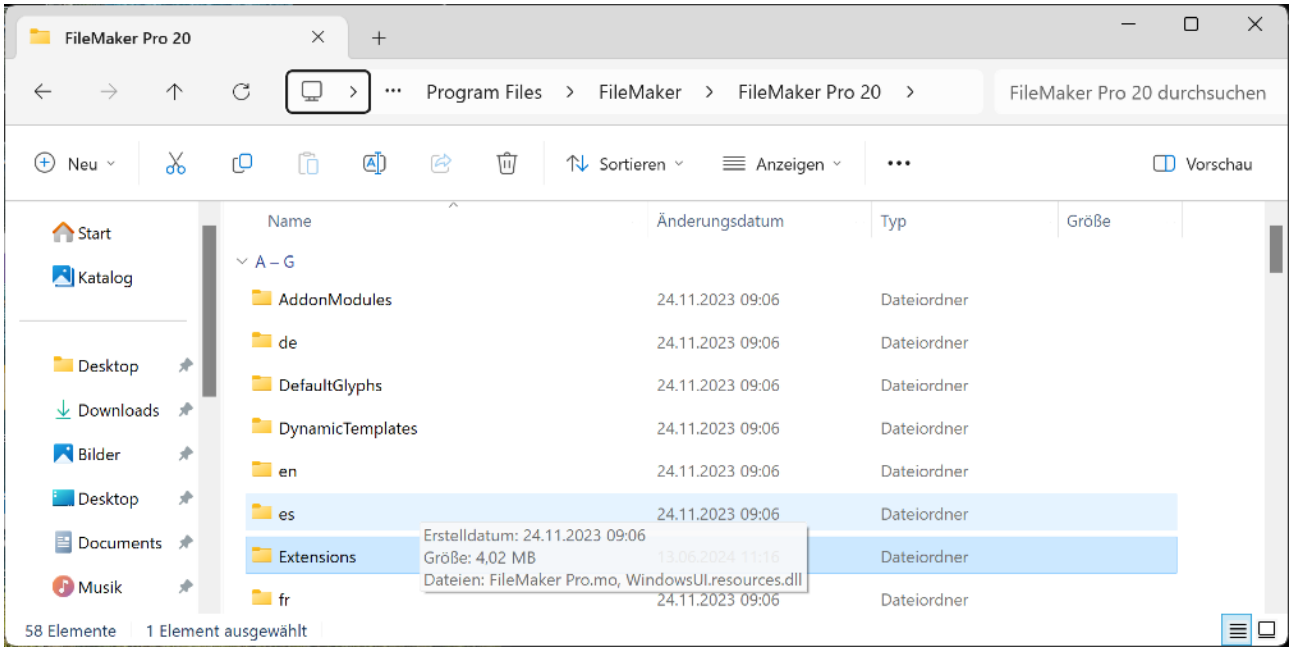


Install

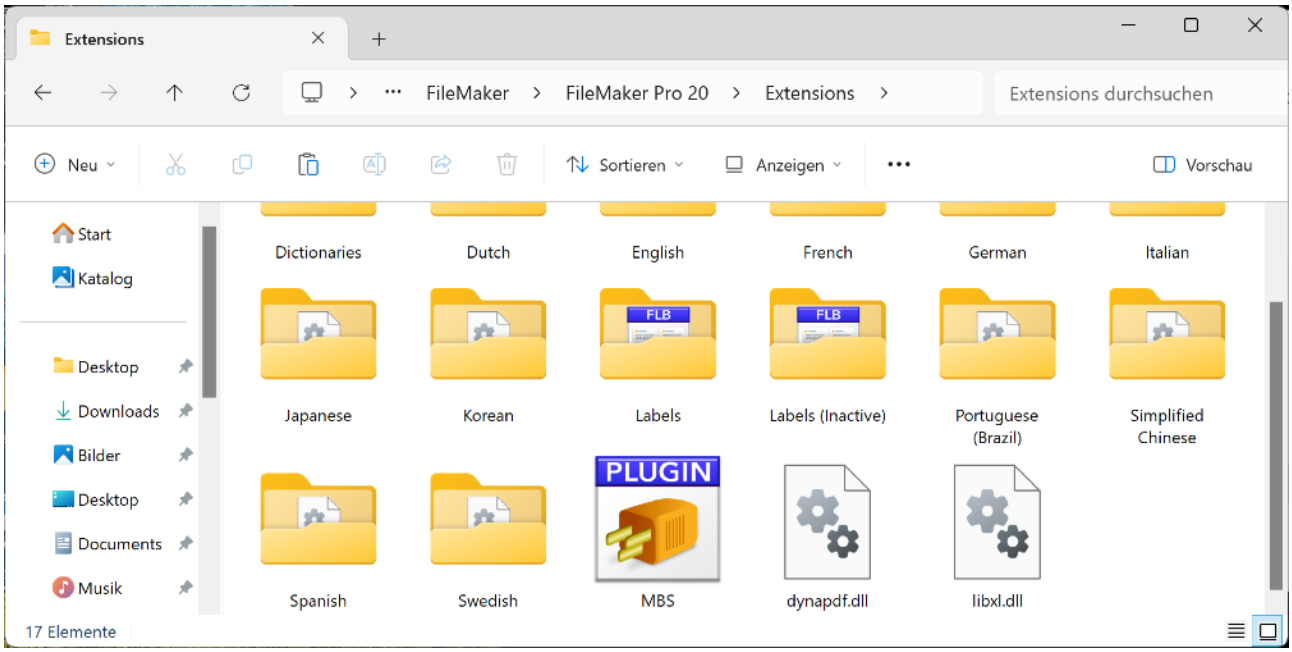
You can install the plugin:

- In the FileMaker application folder
- In the user's app data folder.

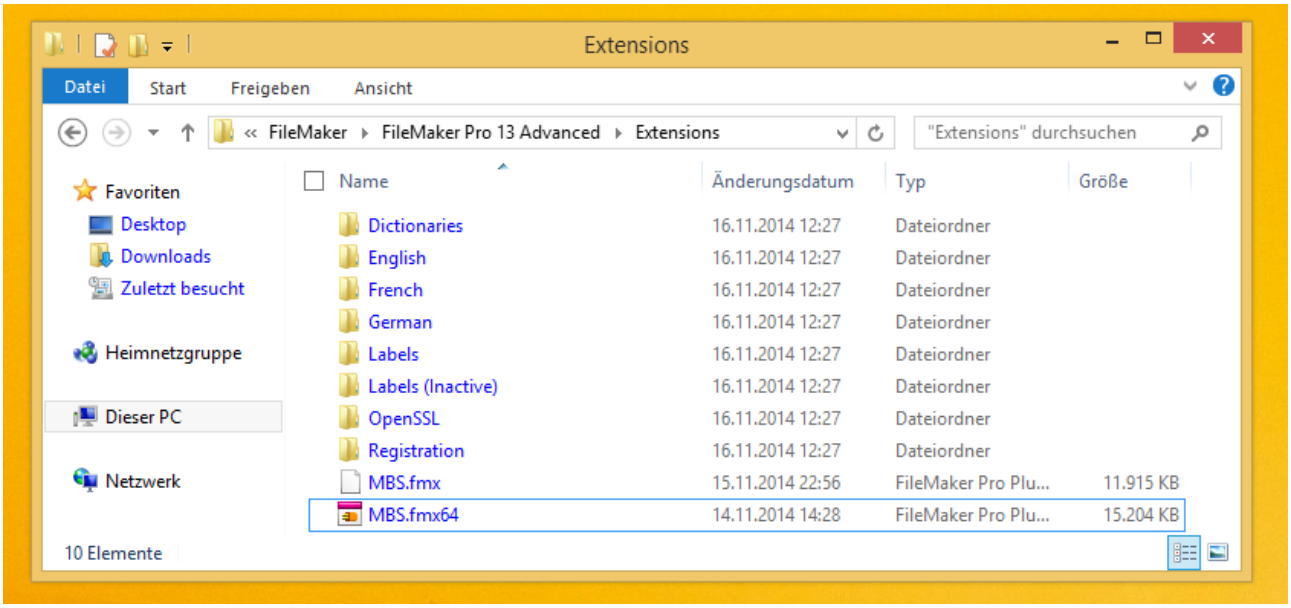
First locate the FileMaker directory on your computer. Normally you find it on your drive C: in Programs.



Once you have the FileMaker folder, you can copy the plugin file into the extensions folder:



The FileMaker extension folder now looks like the one above and you can launch FileMaker. Windows 8.1 with FileMaker 13 looks like this:



As you see we installed both 32-bit and 64-bit here.

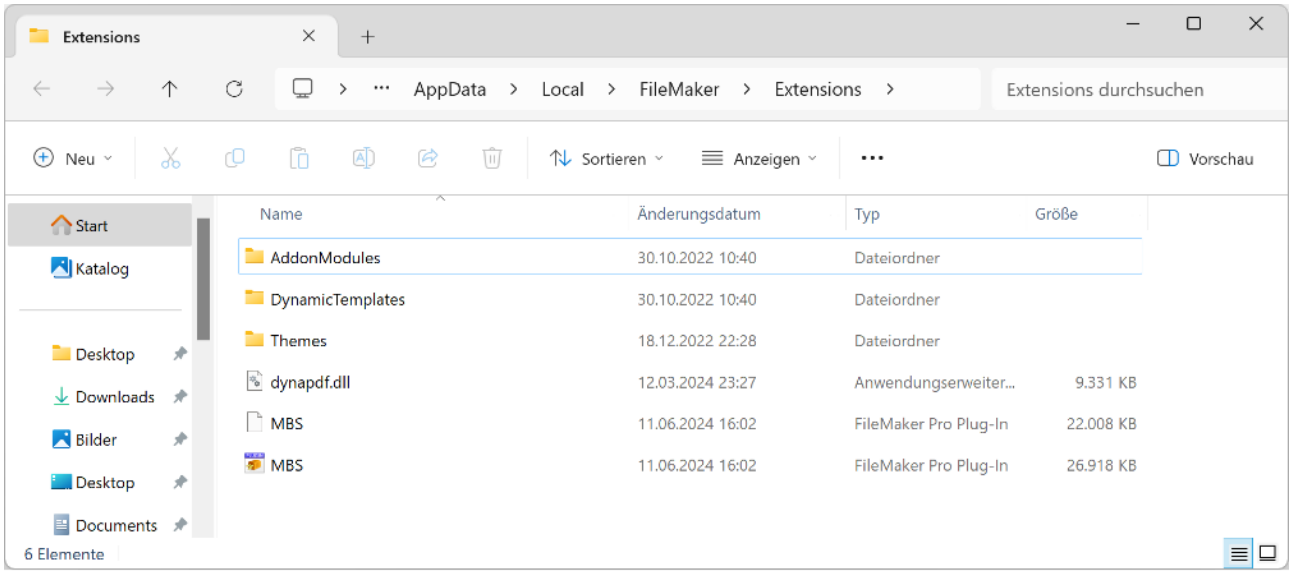
Optionally you can copy the dynapdf and libxl

Alternate Install Location FileMaker

FileMaker has three install locations. The first is the normal one detailed above:

e.g.

C:\Program Files\FileMaker\FileMaker Pro 20\Extensions



The other ones are in the user's AppData folder:

C:\Users\yourname\AppData\Local\FileMaker\Extensions

for all versions or

C:\Users\yourname\AppData\Local\FileMaker\FileMaker Pro\21.0\Extensions\

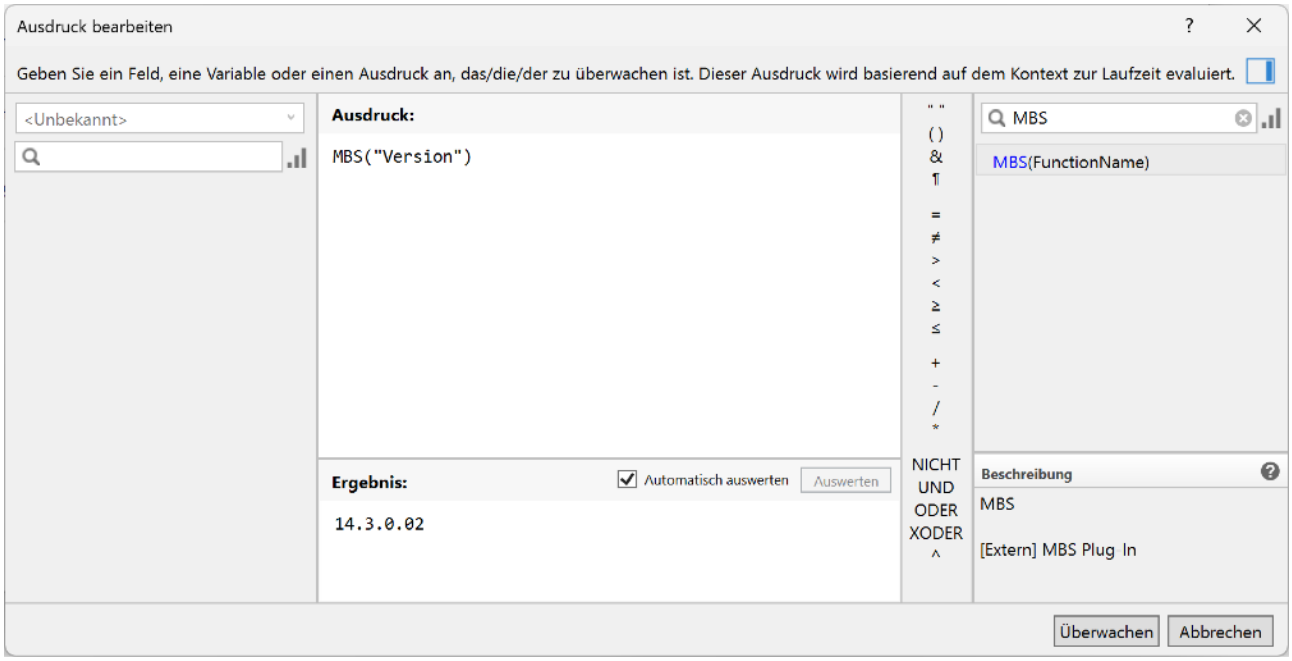
for a specific version.

This last install location is used by FileMaker's Install Plugin File script step.

Use the button in preferences dialog in FileMaker to easily find the right folder for your version.

After installation

To test whether the plugins have been installed, you can launch FileMaker, go to the calculation editor and see the MBS function on the external functions:



So in data viewer in the Monitor section you can add a calculation and in this dialog, you see the result if the use the „Evaluate Now“ button.

Install via Script

You can use a script to install the plugin if needed. See the example database called „Install Plugin Update if needed.fmp12“ in the „Utility functions“ folder inside „Examples“ folder. Here is an excerpt which first finds which container to use and installs plugin via „Install Plug-In File“ script step:

```
# Check Architecture
# will be ? for older 32-bit FileMaker Pro
Set Variable [$Arch; Value:Get ( ApplicationArchitecture )]
# Check Platform
Set Variable [$Plat; Value:Get ( SystemPlatform )]

If [$Plat = 1]

    # Mac
    Install Plug-In File [Install Plugin Update if needed::Plugin File Mac]

Else If [$Plat = 3]

    # iOS
    Exit Script []

Else If [$Plat = -2]

    # Win
    If [$Arch = "x86_64"]
        # Win 64-bit
        Install Plug-In File [Install Plugin Update if needed::Plugin File Win 64bit]
    Else
        # Win 32-bit
        Install Plug-In File [Install Plugin Update if needed::Plugin File Win 32bit]
    End If

Else If [$Plat = 5 OR $Plat = 8]

    # Linux
    If [$Arch = "x86_64"]
        Install Plug-In File [Install Plugin Update if needed::Plugin File Linux Intel]
    Else
        Install Plug-In File [Install Plugin Update if needed::Plugin File Linux ARM]
    End If

Else

    # something new or old?
    Exit Script []

End If
```

```
Set Variable [$LastError; Value:Get(LastError)]
If [$LastError = 0]
	If [MBS("VersionAutoUpdate") = Install Plugin Update if needed::Version Number]
		Show Custom Dialog ["Success"; "Installed MBS Plugin: " & MBS("Version")]
	Else
		Show Custom Dialog ["Success"; "Installed MBS Plugin: " & Install Plugin Update if needed::Version Display & ¶ &  "Please restart …"]
	End If
Else
	Show Custom Dialog ["Failed"; "Failed to install plugin due to error " & $LastError]
End If
```

Demo Limitations

Without a license you can test the plugin.

The plugin may show a warning dialog if not licensed every few minutes.

On the server, the plugin can decide to stop working after 2 hours of usage without license. In that case you can restart server to continue testing.

Purchasing a license and using the license key will remove those limitations.

Except for seat and server licenses where plugin can check and fall back to demo mode if license is used on too many seats or servers.

Server

To install MBS Plugin on a Server version of FileMaker is easy. The preferred way nowadays is to run a script which installs the file using the Install Plugin script step.

Or you just drop the plugin in the right folder:

macOS

/Library/FileMaker Server/Database Server/Extensions

Windows

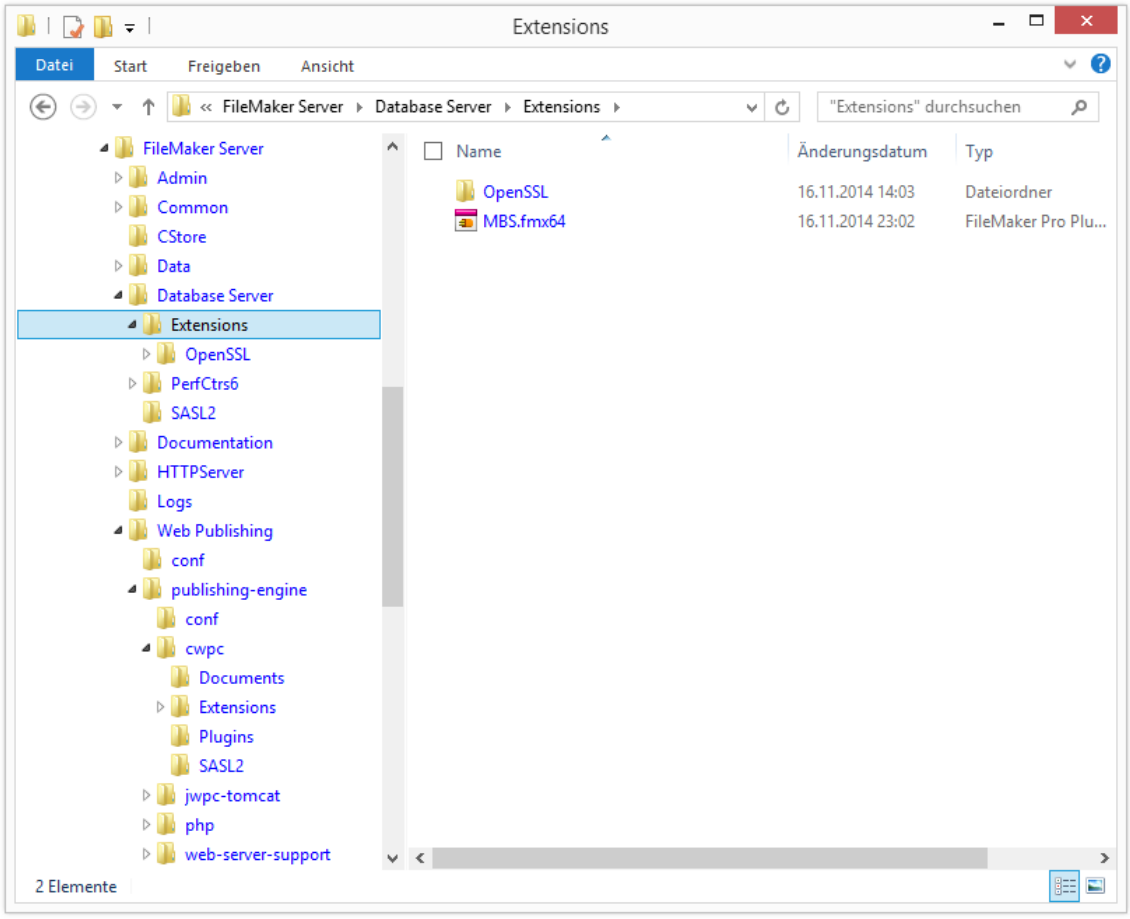
C:\Program Files\FileMaker\FileMaker Server\Database Server\Extensions

FileMaker Server on Linux

/opt/FileMaker/FileMaker Server/Database Server/Extensions

Please activate plugin after installation in the admin console. You may need to go to the Terminal and run "fmsadmin restart fmse" to restart script engine.

The Linux version may require fmsadmin as group and fmserver as user, but this seems to be optional as long as the plugin files can be read by FileMaker Server.



For **Web Direct** please put an extra copy in a different folder:

macOS

/Library/FileMaker Server/Web Publishing/publishing-engine/cwpc/Plugins/

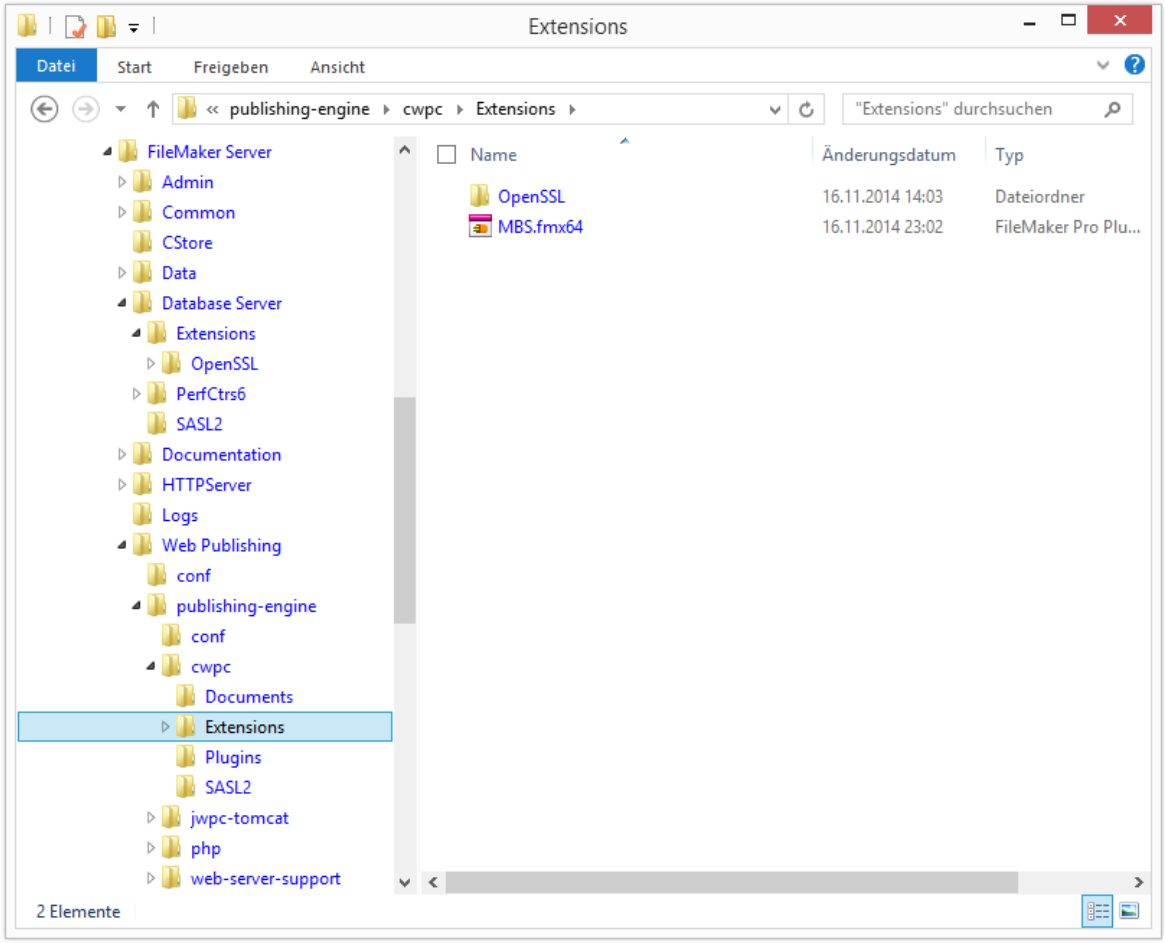
Windows

C:\Program Files\FileMaker\FileMaker Server\Web Publishing\publishing-engine\cwpc\Plugins

FileMaker Server on Linux

/opt/FileMaker/FileMaker Server/Web Publishing/publishing-engine/cwpc/Plugins

If the plugin is not loaded and active, you may need to restart web direct process. For this go to the Terminal and run "fmsadmin restart wpe" to restart the web process.



For **Web Publishing** there is a different set of folders:

macOS

/Library/FileMaker Server/Web Publishing/publishing-engine/wpc/Plugins

Windows

C:\Program Files\FileMaker\FileMaker Server\Web Publishing\publishing-engine\wpc\Plug-ins

You may need to create a folder named Plugins there if it is missing.

Please note that web publishing on FileMaker 12 requires the 64-bit plugin.

FileMaker 13 Server and newer always need a 64-bit plugin.

For **Data API** please put an extra copy in a different folder:

macOS

/Library/FileMaker Server/Web Publishing/publishing-engine/wip/Plugins/

Windows

C:\Program Files\FileMaker\FileMaker Server\Web Publishing\publishing-engine\wip\Plug-ins

FileMaker Server on Linux

/opt/FileMaker/FileMaker Server/Web Publishing/publishing-engine/wip/Plugins

If your admin console has no toggle for Data API, you may need to get a newer version of FileMaker Server to use plugins there. Since version 19.4 there is a toggle for plugins in Data API.

After installation

It may be required that you restart the server, so the WebDirect or Custom Web Publishing engine sees the plugin.

File permissions must be set correct, so FileMaker Server running under a different user (FMServer by default) can read it. Best to set file permissions to have the file read- and executable by all users.

A common way to test if plugin works well, is to have global field in a table where you write by script some details like MBS("Version") and MBS("Platform"). So you can run script with Set Field script steps with those formulas and later see on the values which plugin version is installed and running.

If Platform returns Developer or Pro, you run the script locally. If it returns Web, Server, ServerScripting or InstantWebPublishing, the script is running on the server. Which value is returned is defined by which part of FileMaker server you use.

On macOS you find a log file named stderr in /Library/FileMaker Server/Logs with various messages written out by FileMaker Server. Those include messages about loaded plugins with version:

2015-03-08 17:46:05.204 fmscwpc[251:2408] MBS Plugin version 5.0.0.12 (64 bit) loaded.

2015-03-08 19:28:20.779 fmsased[233:2342] MBS Plugin version 5.0.0.12 (64 bit) loaded.

As you see here both the server side scripting process (fmsased) and the WebDirect process (fmscwpc) loaded the 64-bit version of our plugin.

If the plugin doesn't show up in the server, please check the file permissions. The server user account fmserver may need to have permissions to read and execute the plugin file.

Log files

MBS Plugin will redirect stdout and stderr to text files in Logs folder. This got improved in v11.2 for our plugin to be consistent for all platforms.

StdErrServerScripting.log
StdOutServerScripting.log
StdErrWeb.log
StdOutWeb.log
StdErrDataAPI.log
StdOutDataAPI.log

Web is for Web Direct, ServerScripting for scheduled & PSoS scripts and Data API for alls scripts running in Data API.

Trace messages and errors go to stderr. We normally don't output data to stdout, but some library may do so.

If you login via ssh, this command can show you live messages. This shows messages from our Trace function:

tail -f /opt/FileMaker/FileMaker Server/Logs/ServerScriptingPluginsStdErr.log

FileMaker Cloud 2.0

Please note that FileMaker Cloud 2.0 does not support plugins on the server.

If you need a FileMaker Server with plugin support, there are plenty of hosting companies providing this.

DynaPDF

If you plan to use the DynaPDF functions, you need both to install an extra library and you need an extra license key. Without a license you are in trial mode and DynaPDF puts a watermark on all PDF documents.

For Windows you need dynapdf.dll (or dynapdf32.dll for 32-bit). For MacOS you need the dynapdf.dylib file. You can install it where you want and pass the native path to the DynaPDF.Initialize function.

In newer plugins you can pass the file name to this function („dynapdf.dll“ or „dynapdf.dylib“) and the plugin tries to find it in the extensions folder. Pass "" if the plugin should look for the usual file name.

To purchase a license, please visit the Monkeybread Software plugin website and order it form us. The license keys sold by dynaforms directly do not work in the plugin.

Once you got the license key from us, you pass it as an extra parameter to the DynaPDF.Initialize function.

For iOS you can download DynaPDF.framework from us. See website in Libs folder.

LibXL

If you plan to use the XL functions, you need both to install an extra library and you need an extra license key. Without a license you are in trial mode and libXL puts a watermark on all XL documents and reads only the first 300 cells of a document.

For Windows you need libxl.dll and libxl64.dll. For MacOS you need the libxl.dylib file. You can install it where you want and pass the native path to the XL.Initialize function.

In newer plugins you can pass the file name to this function („libxl.dll“ or „libxl.dylib“) and the plugin tries to find it in the extensions folder. Pass "" if the plugin should look for the usual file name.

To purchase a license, please visit the libXL order website. You may need the libXL Mac and libXL Windows licenses for cross platform projects. Licenses for MacOS, Windows, Linux, iOS and Enterprises are available.

Examples

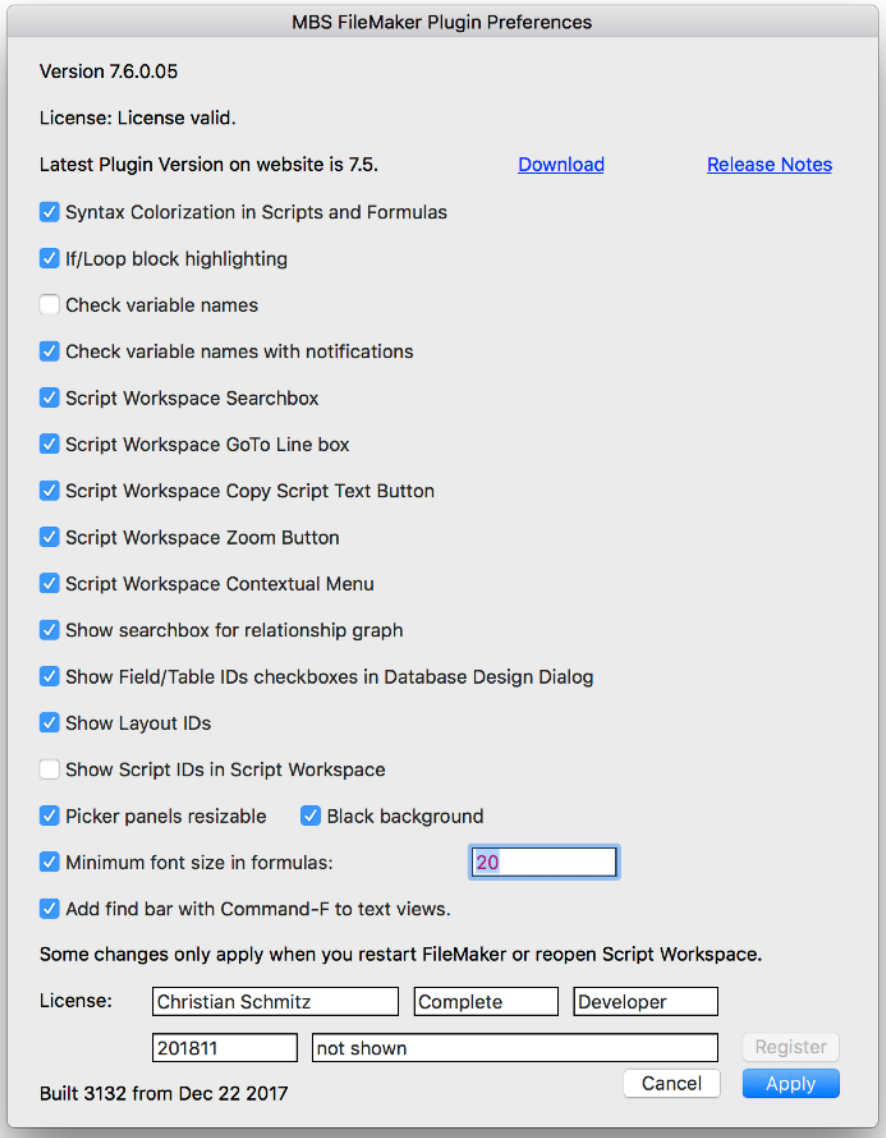
The MBS Plugin comes with over 500 example databases. Please take the time to check them. Examples are provided in fmp12 format, so you can try them with older and newer FileMaker versions. You can find examples for FileMaker 11 or older on the website in a separate download.

The examples for DynaPDF, XL and PHP5 look for a library file located next to database file. If you move those examples to a FileMaker server, you will have to change that script steps to find the library files on the server.

With thousands of functions in our plugin the examples are often the only way to learn how to use our functions together.

Preferences

The MBS FileMaker Plugin for MacOS has a preferences dialog to configure the various options. You reach it by double clicking the plugin in the preferences dialog or clicking the configure button.



Please be aware that not all features are available for all FileMaker versions.

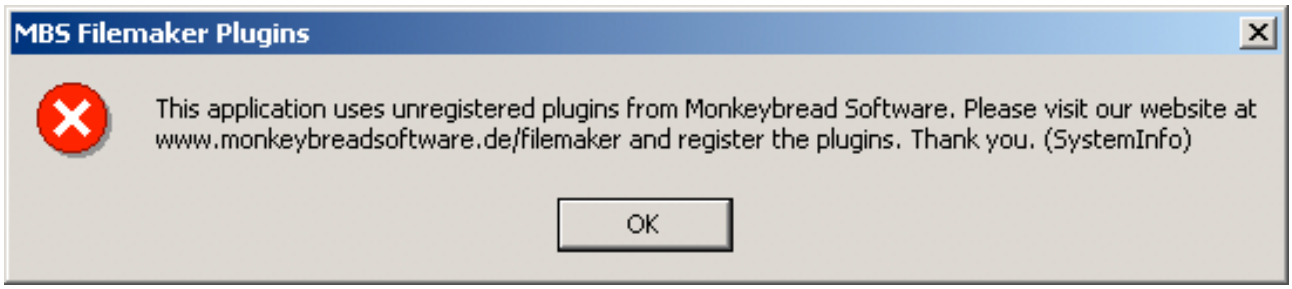
You can configure all options by plugin functions. See SyntaxColoring

Registration

Trial time

You can evaluate the plugin before you order a license.

While you are still evaluating the plugin you will certainly see a dialog like this:



To test without those dialogs, you can request a trial license key.

After you bought a serial number

When you register and you provided all data needed like your full name, your postal address and if needed your VAT ID, you get an email with your serial number.

Now you need to add this serial number to your scripts.

for example you can do it like this:

```
MBS( "Register" ; Licensee Name; Component; License Type; Expire Month; Serial number )
```

You can use this function anywhere that you can create a FileMaker calculation, but most often people place it in the startup script of a solution. That way the plug-in will be registered when the solution starts up. You only need to call this function one time for each FileMaker session.

Please make the startup script unavailable to people to look inside and copy the serial number from a solution.

Here an example script to registered if needed for the right platform:

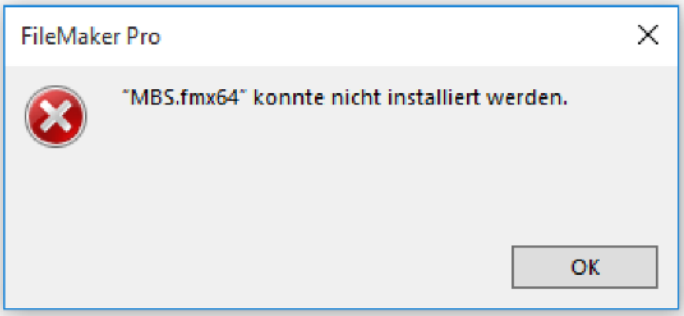
```
If [MBS("IsRegistered") ≠1]
    If [MBS("IsRuntime")]
        Set Variable [$r; Value:MBS("Register"; "test"; "Runtime"; 123; 123)]
    Else If [MBS("IsServer")]
        Set Variable [$r; Value:MBS("Register"; "test"; "Server"; 123; 123)]
    Else
        Set Variable [$r; Value:MBS("Register"; "test"; "5 seats"; 123; 123)]
    End If
End If
```

Troubleshooting

Plugin not loading

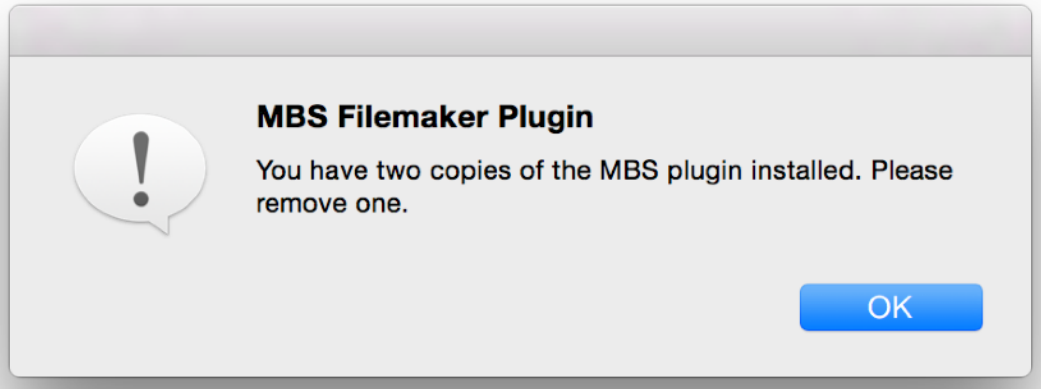
Please check if you see an error message in DebugView app (Windows) or Console app (Mac). For FM Server check the logs for errors about loading plugin.

As we ship plugins in 32-bit or 64-bit version for Windows, you can install both everywhere. FileMaker loads the one it needs and ignores the other variant. But if you only install 32-bit plugin for 64-bit FileMaker, it will simply be ignored. Or you get an error like this:



Duplicate Installation

Installing two plugin versions at the same time may lead to strange problems. To avoid this we try to detect that and report an error message to user:



On Mac having two plugins can lead into one plugin calling functions and those could be located in the other version. As parameters change, this can lead to crashes. So please only have one copy of MBS Plugin in various folders.

Plugin deactivated

If you deactivate any plugin on Mac, this can result in crashes in FileMaker. Especially in low level cocoa runtime methods.

e.g. in method_setImplementation or in +[<class name> initialize].

The cocoa class accessed is in the deactivate plugin. FileMaker loaded the plugin, the class was registered, the plugin unloaded. The class registration entry in the global data structures now points to memory which has been freed when the plugin was unloaded.

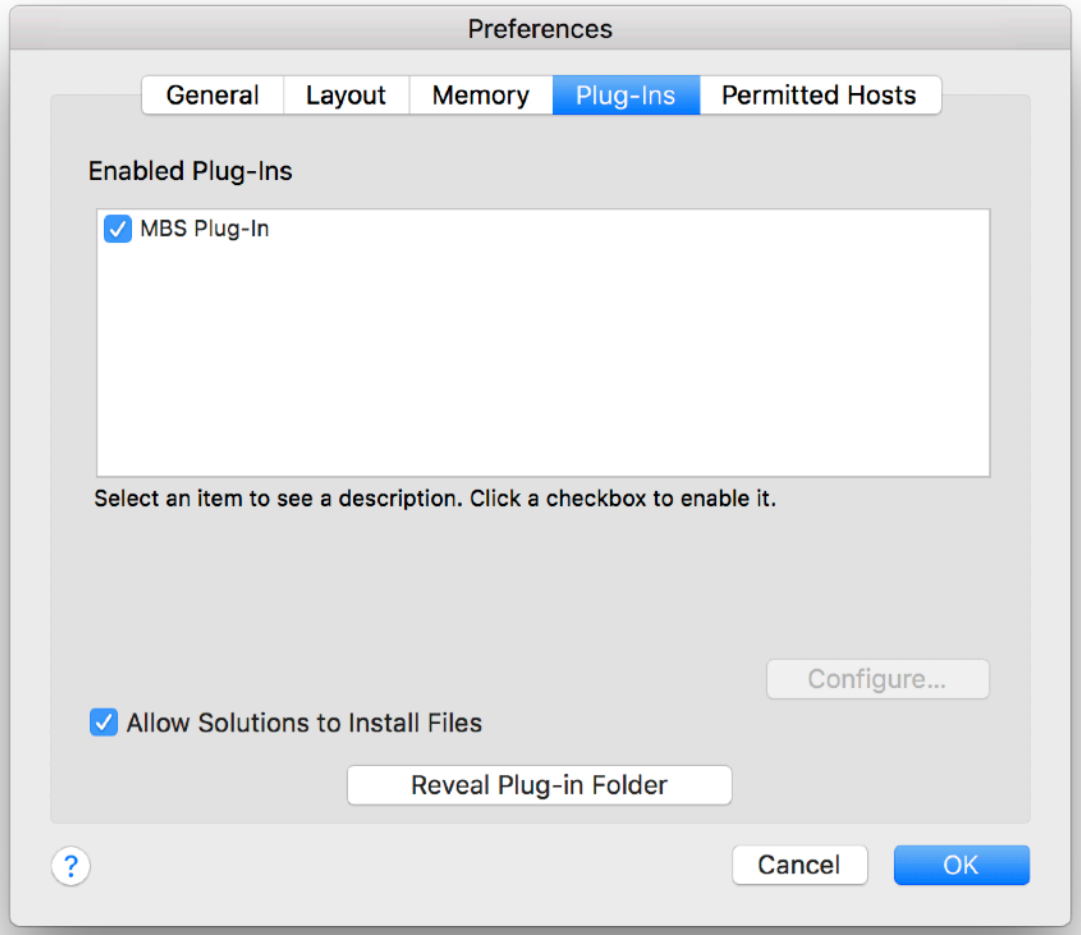
So please never deactivate plugins on Mac. Remove them from the extension folder if necessary.

Errors with Installation by Script

You can install the plugin via the Install Plugin Script Step.

But this can fail with various reasons:

- The Script Step may be disabled by permissions, e.g. in FileMaker Pro Preferences option for "Allow Solutions to Install Files" or for server in the admin console.
- The plugin file already exists and can't be overwritten due to file permissions.
- You tried to install the wrong plugin file, e.g. 32 vs 64 bit mismatch between plugin and FileMaker application.



Contact

The author Christian Schmitz can be contacted at support@monkeybreadsoftware.de.

All comments are welcome to improve this guide.