Read And Write Excel Files With FileMaker

Do you already know our LibXL functions for FileMaker, with which it is possible to read, create and edit Excel documents although you don't have Excel installed? This part of the plugin requires an additional LibXL license and you need to attach an additional library to your database. I will show you in this blog article how to make the preparations to use the LibXL functions in your solution.

The LibXL License

You can order LibXL licenses on our website. There are different licenses, depending on the platform on which you want to use LibXL.

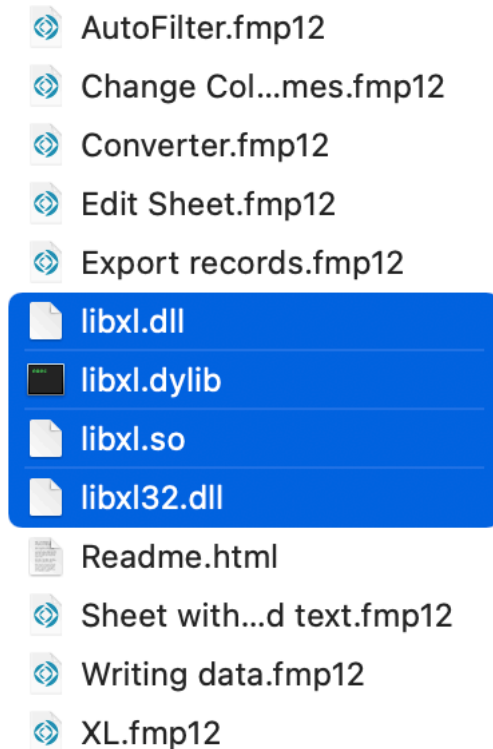| Library for Excel functions	Price Euro	Price USD		
LibXL macOS	199 €	about $239	Order	Update
LibXL Windows	199 €	about $239	Order	Update
LibXL Linux	199 €	about $239	Order	Update
LibXL iOS	199 €	about $239	Order	Update
LibXL Enterprise	1099 €	about $1399	Order	Update
LibXL Enterprise + Source	2499 €	about $2999	Order	Update

The licenses for Mac OS and Windows are clear so far. The license for Linux is for using FileMaker on the cloud for AWS. The iOS version you can use in combination with an iOS SDK app. You can NOT use the features directly in FileMaker Go, as we cannot provide plugins for this platform. If you want to use LibXL or other plugin functionality on FileMaker Go, you need to run the appropriate script on the server via Perform Script on Server. If you choose the Enterprise license, you will receive a license code that you can use for multiple developers on multiple servers and on all platforms mentioned above. You also get priority technical support from LibXL. But before you need to choose any license, you can test LibXL for free. In the test mode is a note in the first row that says it's a test version. This first line cannot be modified or deleted.

Created by LibXL trial version 4.0.3. Please buy the LibXL full version for removing this message.						
	Hello World					
	1000					
	39567					

Including The Library

The current LibXL library is included in the [MBS FileMaker Plugin](#) download. It is located in the folder Exempels in the subfolder XL. Here we have 4 different library files.

- AutoFilter.fmp12
- Change Col...mes.fmp12
- Converter.fmp12
- Edit Sheet.fmp12
- Export records.fmp12
- **libxl.dll**
- **libxl.dylib**
- **libxl.so**
- **libxl32.dll**
- Readme.html
- Sheet with...d text.fmp12
- Writing data.fmp12
- XL.fmp12

The libxl.dylib file is needed if we want to use the LibXL functions on the Mac. The libxl.so is for the use in the AWS Cloud on Linux. The two files libxl.dll and libxl32.dll are for usage on Windows. It depends if your system is a 64 bit system, then you use libxl.dll, or a 32 bit system, then you need the libxl32.dll file. You can copy these files and place them in the desired location. In your database LibXL has to be initialized before you use the first libXL function. To do this, call the [XL.Initialize](#) function in your script. You give it the path to the corresponding library and if available the information about your license.

```
Set Variable [ $r ; Value: MBS( "XL.Initialize"; $path;
$LicenseeName; $LicenseKey) ]
```

If you want to prepare for all operating systems, then just have a look at our InitXL Script, which is included in all XL sample files.

I also highly recommend to check in every script where you use [XL](#) functions with the [XL.IsInitialized](#) function if LibXL was initialized and if not to initialize it. This can look like this:

```
If [ MBS("XL.IsInitialized") ≠ 1 ]
    Perform Script [ Specified: From list ; "InitXL" ;
Parameter:        ]
End If
```

It costs you only one additional query, but saves you a lot of trouble.

Red Text and blue in FileMaker				
Hello World				

Now you can start. Feel free to try out our examples. Have fun with LibXL.