

How to use XL functions with MBS FileMaker Plugin

First when you plan to use the XL functions, be aware that you will need the license for our MBS plugin. In addition you need licenses for the [LibXL](#) library. Our plugin uses this library to actually perform the functions. [LibXL](#) is sold separately for \$199 per platform and you only buy one license for you as the developer, independent of the number of clients.

Initialize

Before using the XL functions, it is important to initialize the XL functions. Best practice is to check result of MBS(["XL.IsInitialized"](#)) function. If result is not 1, you need to initialize. For that you first need to locate the libxl files. Our examples include library files for Mac and Windows. We have libxl.dylib on Mac for 32 and 64 bit in one file. For Windows libxl.dll is the 32bit and libxl64.dll is the 64bit file. On Windows you can pass path to libxl.dll and if FileMaker is running as 64bit application, the plugin will add the 64 automatically and find the 64bit library. The example databases find the library in the same folder as the example database automatically. On a Server you normally copy the libxl files to the server and hard code the native path, e.g. `"/Library/FileMaker Server/libxl.dylib"`. So the initialization can look like this:

```
If [MBS("XL.IsInitialized") ≠ 1]
    Set Variable [$r; Value:MBS( "XL.Initialize"; "/Library/FileMaker
Server/libxl.dylib")]
End If
```

Create book

Next we have to create a new book. The plugin uses reference numbers to handle several books in memory at the same time. It is important you reference your book with a variable in the whole script and release it on the end. To create a workbook, please call [XL.NewBook](#) function. Here you need to decide if you plan to save later as XML based or binary excel file. We pass 0 for the normal xls file format:

```
Set Variable [$book; Value:MBS( "XL.NewBook"; 0 )]
```

After we got our book, we add a sheet labeled "Addresses". We get back the sheet reference number which is zero for first sheet:

```
Set Variable [$sheet; Value:MBS( "XL.Book.AddSheet"; $book;
"Addresses")]
```

Write data

Our main loop here loops over the records in our test table. Starting with first record we move one row at a time and one record a time. Inside the loop we set cell text for each cell in current row and fill them with values from the record. As you see we have to pass book reference number, sheet number, row number and column indexes. We define the columns ourself here.

Go to Record/Request/Page [First]

Loop

Set Variable [\$r; Value:MBS("[XL.Sheet.CellWriteText](#)"; \$book; \$sheet; \$row; 0; Export records::First Name)]

Set Variable [\$r; Value:MBS("[XL.Sheet.CellWriteText](#)"; \$book; \$sheet; \$row; 1; Export records::Last Name)]

Set Variable [\$r; Value:MBS("[XL.Sheet.CellWriteText](#)"; \$book; \$sheet; \$row; 2; Export records::Street)]

Set Variable [\$r; Value:MBS("[XL.Sheet.CellWriteText](#)"; \$book; \$sheet; \$row; 3; Export records::Zip)]

Set Variable [\$r; Value:MBS("[XL.Sheet.CellWriteText](#)"; \$book; \$sheet; \$row; 4; Export records::City)]

Set Variable [\$r; Value:MBS("[XL.Sheet.CellWriteText](#)"; \$book; \$sheet; \$row; 5; Export records::Country)]

Set Variable [\$r; Value:MBS("[XL.Sheet.CellWriteText](#)"; \$book; \$sheet; \$row; 6; Export records::Phone)]

Go to Record/Request/Page [Next; Exit after last]

Set Variable [\$row; Value:\$row + 1]

End Loop

Finish

On the end, we save to a container. For that we use the Set Field script step. the plugin function [XL.Book.Save](#) returns the container value with the Excel file inside. We pass a file name for the container to give it a name which FileMaker uses to export the field content later.

Set Field [Export records::File; MBS("[XL.Book.Save](#)"; \$book; "test.xls")]

Finally we have to release memory and close book. So the [XL.Book.Release](#) will clear memory:

Set Variable [\$r; Value:MBS("[XL.Book.Release](#)"; \$book)]

More?

This is of course a very easy example. It does not create fonts or formats. Maybe you want to check other example databases coming with plugin if you like to learn how to create formats, fonts and add images to Excel files.