

PHP Functions in MBS Filemaker Plugin

With our 2.9 plugin function we added PHP functions. Those functions allow you to load the PHP library from file system and use it. This gives you tons of new functionality with our plugin for either reusing PHP code from your website or by solving new issues with PHP.



PHP Library

We include a copy of the php library for your convenience with the plugin download. In your startup script you can find the library and load it to have PHP ready when you need it. If you like, you can compile your own PHP library with a custom set of extensions. For that you find instructions in the ReadMe file included with PHP examples.

So once your solution located the php library and got the native path, please call the [PHP.Load](#) function. Next you call [PHP.Init](#) function to initialize the PHP engine. If needed you could also use [PHP.ShutDown](#) to shutdown, but most times you let it run until Filemaker quits.

Plugin Functions

First we have [PHP.Execute](#), a convenient function to pass some PHP code and get the result of the function. With result we mean the plugin captures all output to standard output in PHP. Normally you write something there with print or echo commands. So here you either get "[MBS] Failed. Bad Syntax?" or the output. As you see the PHP interpreter can fail to parse your script due to a syntax error. But if everything is correct, the script should run just fine.

We do recommend to use php exception handling and report errors in a way where your Filemaker solution can handle the error gracefully.

Instead of Execute you can also run [PHP.Run](#) and [PHP.GetResultString](#). This way you can check if PHP.Run returns "OK" and pick result later.

Variables

PHP has global variables and we have a couple of plugin functions related to it. With [PHP.ExistsVariable](#) you can check if a variable with a given name exists. With [PHP.SetVariable](#) you can set a variable with a text value. And [PHP.GetVariable](#) gives you a variable value as text. If the variable is not text, it will be converted. You can of course use JSON or Base64 functions here for handling binary or object data.

Text Encoding

Our plugin expects text from PHP to be ISO 8859-1 encoded and also all text passed in for source code or variables. Maybe we someday change that to UTF-8. To be save you can of course use base64 encoding for your data, especially for passing raw byte strings.

Callback

Our plugin defines a new PHP module called FilemakerModule. This module has 3 functions: FMScript, FMEvaluate and FMExecuteSQL.

You can call FMScript in your PHP script and pass up to three parameters. This function works like [FM.RunScript](#). First is the name of a database file, second the name of the script and third the optional text parameter. The plugin will call the script soon. Soon as Filemaker script calls from plugin are queued and executed later.

Example:

```
echo FMScript("Simple_php", "TestScript", "Hello World");
```

The FMEvaluate function takes a text parameter with a string to evaluate. This way you can query things from Filemaker like variables or call functions. Useful for functions like Get(UserName).

Example:

```
echo FMEvaluate("1+2");  
echo FMEvaluate("Get(UserName)");
```

The FMExecuteSQL function now runs SQL code just like [FM.ExecuteSQL](#). First parameter is the SQL command. Second parameter defines the ASCII code of the column delimiter and third parameter the row delimiter. Default for column delimiter is tab (9) and for rows it is return (13). The result is passed back to PHP.

Example:

```
echo FMExecuteSQL("select * from Contacts");
```

Extensions

To get the list of extensions, you can run this PHP code:

```
print_r(get_loaded_extensions());
```

This shows currently on Mac OS X:

Array

```
(  
[0] => Core  
[1] => date  
[2] => ereg  
[3] => libxml  
[4] => pcre  
[5] => sqlite3  
[6] => zlib  
[7] => bcmath  
[8] => bz2  
[9] => calendar  
[10] => ctype  
[11] => curl  
[12] => dom  
[13] => hash  
[14] => fileinfo  
[15] => filter  
[16] => ftp  
[17] => SPL  
[18] => iconv  
[19] => json  
[20] => session  
[21] => PDO  
[22] => pdo_sqlite  
[23] => standard  
[24] => posix  
[25] => Reflection  
[26] => Phar  
[27] => SimpleXML  
[28] => soap  
[29] => sockets  
[30] => exif  
[31] => tokenizer  
[32] => xml  
[33] => xmlreader  
[34] => xmlwriter  
[35] => zip  
[36] => FilemakerModule  
)
```

As you see we have all the famous names like sqlite (database), zlib and zip (zip compression), pcre (Regular expressions), ftp, curl and even soap.

If you have questions, please do not hesitate to ask us.

This article is for Filemaker, but we have the same functionality available also in the [MBS Real Studio PHP Plugin](#).