

Our JavaScript engine for FileMaker

For next MBS FileMaker Plugin we add a JavaScript engine to be used within your FileMaker solution without a web viewer control. You can execute JavaScript and offer it to your users as a scripting language within your solution. If you need, you can run JavaScript snippets at runtime to do calculations and even integrate your own functions to call back to FileMaker scripts.

Evaluate JavaScript

You can use JS functions in next plugin version and instantiate a JavaScript environment. Then you can evaluate some JavaScript code and get back a result. For your convenience we convert JavaScript data types to JSON.

```
Set Variable [ $js ; Value: MBS( "JS.New" ) ]  
Set Variable [ $r ; Value: MBS( "JS.Evaluate"; $js; "4+5" ) ]  
Show Custom Dialog [ "Result" ; $r ]  
Set Variable [ $r ; Value: MBS( "JS.Free"; $JS ) ]
```

Alternative you can use [JS.EvaluateToString](#) function which converts the result to text. If something goes wrong you may get an error message, so it may be good to check for an error with our [IsError](#) function.

Own functions

Using [JS.AddFunction](#) you can add a custom function defined as JavaScript. We compile the JavaScript and put it in the global name space. If you like to call it directly without evaluate, you can use [JS.CallFunction](#) function.

```
Set Variable [ $r ; Value: MBS( "JS.AddFunction"; $js; "adder";  
"function (x,y) { return x+y; }" ) ]  
Show Custom Dialog [ "Result" ; MBS( "JS.CallFunction"; $JS; "adder";  
3; 4 ) ]
```

Global properties

You can use [JS.GetGlobalProperty](#) and [JS.SetGlobalProperty](#) to get and set values in variables linked to the global namespace in JavaScript. The data is passed/received in JSON format.

```
# try number  
Set Variable [ $r ; Value: MBS( "JS.SetGlobalProperty"; $JS; "v1";  
123 ) ]  
Set Variable [ $v1 ; Value: MBS( "JS.GetGlobalProperty"; $JS; "v1" ) ]
```

```
# try some JSON array
Set Variable [ $r ; Value: MBS( "JS.SetGlobalProperty"; $JS; "v2";
"[1,2,3]" ) ]
Set Variable [ $v2 ; Value: MBS( "JS.GetGlobalProperty"; $JS; "v2") ]
```

```
# try some text
Set Variable [ $r ; Value: MBS( "JS.SetGlobalProperty"; $JS; "v3";
 "\"test\"" ) ]
Set Variable [ $v3 ; Value: MBS( "JS.GetGlobalProperty"; $JS; "v3") ]
```

As long as you keep the engine object alive, the values are stored and you can use them in your JavaScript code.

Callbacks to FileMaker

With [JS.AddFileMakerEvaluateFunction](#) function you can add an evaluate function in JavaScript with your own name. In JavaScript you can then call this function with a parameter, which the plugin passes to FileMaker to evaluate. The result is converted to text and returned to JavaScript. By default the engine starts without such a function as it provides access to all your global variables and with Let() statement users can define variables. You may want to define your own function name, which other people may not know. You can even wrap calls to your evaluate function with a custom JavaScript function and filter the queries.

```
Set Variable [ $r ; Value: MBS( "JS.AddFileMakerEvaluateFunction"; $js;
"FMEval" ) ]
Show Custom Dialog [ "Result" ; MBS( "JS.Evaluate"; $JS; " \"got
AccountName: \" + FMEval(\"Get(AccountName)\")" ) ]
```

To execute SQL you can define a function with [JS.AddFileMakerSQLFunction](#). You pass in JavaScript the file name, the SQL expression and optional all the parameters. Again define your own name, possibly wrap this into a function as people may run DELETE command to delete data:

```
Set Variable [ $r ; Value: MBS( "JS.AddFileMakerSQLFunction"; $js;
"FMSQL" ) ]
Show Custom Dialog [ "Result" ; MBS( "JS.Evaluate"; $JS; " \"got SQL
result: \" + FMSQL(\"JavaScript.fmp12\", \"SELECT COUNT(*) FROM
JavaScript\")" ) ]
```

With [JS.AddFileMakerRunScriptFunction](#) you can run a script. You pass file name, script name and parameter. The script launch is queued and will run later when the JavaScript is finished and any script using it. Please be aware people may

trigger any script defined in any open file, even if it is hidden in the menu.
Triggering scripts does not work on server due to a limitation in the plugin SDK.

```
Set Variable [ $r ; Value: MBS( "JS.AddFileMakerRunScriptFunction";  
$js; "FMRun") ]  
Show Custom Dialog [ "Result" ; MBS( "JS.Evaluate"; $JS; " \start  
script: \" + FMRun(\"JavaScript.fmp12\", \"testRun\", \"Hello  
World\")" ) ]
```

We hope you enjoy the new JavaScript functions. Let us know if you have any questions!