

Custom Functions in JavaScript

We saw a presentation at dotfmp conference from Philipp Puls about him storing his custom functions in a table inside the database. With help of one custom function in FileMaker itself, he can run them all. This dispatcher function would use [execute SQL](#) functions in [MBS FileMaker Plugin](#) to query the expression by name, then add the parameter inside and evaluate it. Quite convenient to have all custom functions stored in one database. Based on that idea we developed two new ways at MBS to do custom functions in JavaScript and later without.

Custom Functions in JavaScript

Since [MBS FileMaker Plugin](#) comes with a built-in JavaScript engine, which works without web viewer and even on a server side script, we can use it to define custom functions, which we can call anywhere in FileMaker:

JavaScript Custom Functions

You can initialize a JavaScript environment in your startup file and load custom functions into it and later call them at any time from any file.

Name Concat

Declaration Concat(Value1, Value2)

Description A quick test function to concat two texts

Parameters

Value1	The first parameter	Hello
Value2	The second parameter	World

Body return Value1 + " " + Value2;

[Register Function](#) [Register and Try](#)

SampleCall MBS("JS.CF"; "Concat"; Value1, Value2)

This example database included with 11.3pr2 shows how to do it. It sets up the whole thing in the start script. With [JS.New](#) we make a new JavaScript environment with the special name "js". Instead of getting a handle number from the plugin, we define the identifier ourselves. This way we can refer to this JavaScript environment everywhere. It's global for all files and all solutions opened in FileMaker and can be initialized in FileMaker Pro, Server (Scripting + Web Direct) as well as FileMaker iOS SDK based applications.

Next we fill the JavaScript environment with [JS.AddFunction](#) function to define our custom functions in JavaScript. If you like, you can use [JS.AddFileMakerEvaluateFunction](#), [JS.AddFileMakerRunScriptFunction](#) and [JS.AddFileMakerSQLFunction](#) to allow JavaScript to evaluate FileMaker expressions, e.g. to call Get functions, trigger scripts or run a SQL command. You can also use [JS.SetGlobalPropertyValue](#) or [JS.SetGlobalProperty](#) to store global data for JavaScript to use later, like big tables or JSON objects. Since functions for JavaScript can have variable number of parameters, you can also pass as many as needed via our plugin.

Now when you like to use the custom functions in JavaScript, you can just call [JS.CF](#) function in our plugin, pass the function name and the parameters. The plugin converts them to JavaScript (as text or numbers), performs the function and returns the result as value. [JS.CF](#) is a shortcut to [JS.CallFunctionValues](#) with passing "js" as the identifier of the environment. But you can of course also call [JS.CallFunctionValues](#) or [JS.CallFunction](#) as needed. The last one passes values as JSON, so texts go in quotes.

If you have some JavaScript code to do some math, text manipulation, checksum calculation, JSON processing or otherwise be useful, you can now use it like a custom function.

Please try the [JS.CF](#) function with [MBS FileMaker Plugin](#) in version 11.3pr2 or later.

Part two will follow soon to explain how to do this without JavaScript and replace the normal custom functions you know.