

Custom Functions in JavaScript reloaded

We had a previous blog post about [Custom Functions in JavaScript](#) using the built-in JavaScript engine within our MBS Plugin. Now we got a second way to do the same using the JavaScript engine coming with WebKit on macOS and iOS: [JavaScriptMac](#)

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 using either JS or JavaScript functions:

The screenshot shows a window titled "JavaScriptMac Custom Functions". The interface includes a toolbar with icons for navigation and actions, and a search bar. Below the toolbar, there's a blue header section with the title "JavaScript Custom Functions via MBS Plugin" and a brief description: "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. This is the version of the file for using WebKit JavaScript engine on macOS or iOS."

The main form contains the following fields:

- Name:** Concat
- Declaration:** Concat(Value1, Value2)
- Description:** A quick test function to concat two texts
- Parameters:** A table with three columns: Name, Description, and Value.

Value1	The first parameter	Hello
Value2	The second parameter	World
- Body:** A text area containing the code: `return Value1 + " " + Value2;`

At the bottom, there are two buttons: "Register Function" and "Register and Try". Below these buttons is a "Sample Call" field containing the text: `MBS("JavaScript.CF"; "Concat"; Value1; Value2)`

This example database included with 12.2pr5 shows how to do it using JavaScriptMac functions. It sets up the whole thing in the start script. With [JavaScript.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 [JavaScript.RegisterFunction](#) function to define our custom functions in JavaScript. If you like, you can use [JavaScript.RegisterEvaluateFunction](#), [JavaScript.RegisterRunScriptFunction](#) and [JavaScript.RegisterSQLQueryFunction](#) to allow JavaScript to evaluate FileMaker expressions, e.g. to call Get functions, trigger scripts or run a SQL command. You can also use [JavaScript.SetProperty](#) 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 [JavaScript.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. [JavaScript.CF](#) is a shortcut to [JavaScript.CallFunction](#) with passing "js" as the identifier of the environment. But you can of course also call [JavaScript.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 [JavaScript.CF](#) function with [MBS FileMaker Plugin](#) in version 12.2p5 or later.

And for cross-platform development, check the [JS](#) functions.