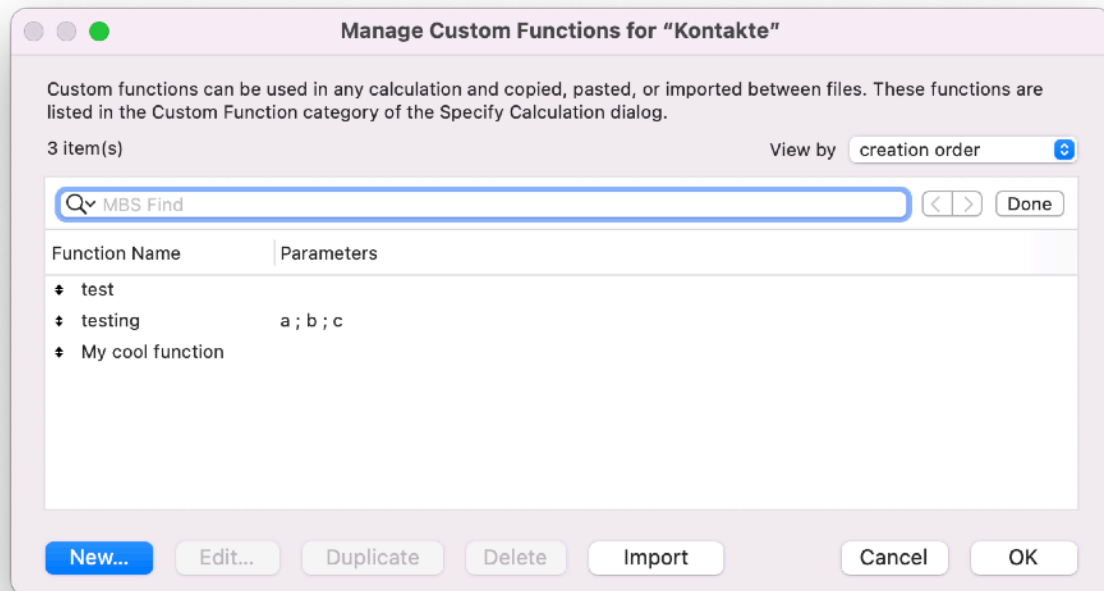


Query custom functions automatically in FileMaker

We have clients with lots of custom functions in various FileMaker databases. How would you automate this to track custom functions used in various files?



For [MBS FileMaker Plugin](#) 12.1 we got two new functions for this: [FM.CustomFunctionsList](#) and [FM.QueryCustomFunctionsList](#). Those functions do a bit of GUI scripting on macOS to get the list. The user needs of course permissions to open the dialog and the menu entry into the menubar for the plugin to click.

The first example script will first trigger the seconds script to run in a second. Make sure the fmplugin extended privilege is checked, so the plugin is allowed to run a script. Once that is set, we run the [FM.QueryCustomFunctionsList](#) script to start the plugin function to do its job. The script now has to end, so FileMaker allows the custom functions dialog to show. Here is the complete first script:

```
# schedule script to read results in a second
Set Variable [ $r ; Value: MBS( "FM.RunScriptLater"; 1; Get(FileName);
"show result") ]
# now start query
Set Variable [ $r ; Value: MBS("FM.QueryCustomFunctionsList") ]
```

If everything works right, the dialog shows for a fraction of a second. The plugin copies the list of the table and stores it internally in a global

variable. Then our second script runs, which can use [FM.CustomFunctionsList](#) function to query the list:

```
Set Variable [ $list ; Value: MBS("FM.CustomFunctionsList") ]  
Show Custom Dialog [ "results" ; $list ) ]
```

If the list is empty, the function failed to run properly. For example if there is a custom menu and no menu entry to edit the custom functions.

Please try it. It may help you to automatically query the list of custom functions and then write that information into records. A way to monitor which file uses what custom functions. You can use Evaluate() to run those custom functions dynamically. That is of course only useful if you have some global variable or parameter to switch your custom functions to return some metadata like their version.

Have fun and thanks to Nils Waldherr for the idea!