

# MBS FileMaker Plugin debugging help

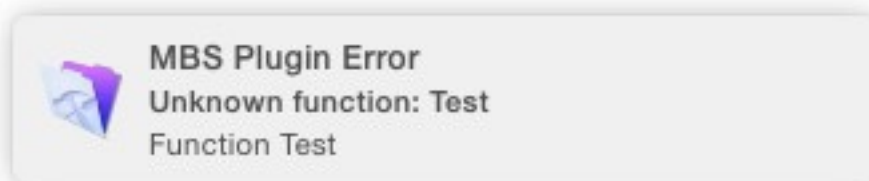
You may know already that MBS Plugin enhances the script editor on Mac with line numbers, script step colors, calculation colors and if enabled variable declaration check.

## Typo suggestions

Now with version 5.0 we have some more enhancements. If you have a typo in your function name, you get a message. So the call `MBS("ersio")` returns now "[MBS] Unknown function: ersio. Did you mean Version?". Our plugin simply scans internal function list to find a function name which matches best to the given function name. You can of course use similar best match search for your own solutions with the [List.BestMatch](#) and [QuickList.BestMatch](#) functions.

## Notifications

Next you can get notifications for script errors. Run [Trace.EnableErrorNotifications](#) function once to enable this feature on your Mac. Whenever a script error happens, you'll see this notification showing the error message and the function name:



Be sure to disable it when going into production so your users don't see the notifications. You should check in your scripts for errors and handle them gracefully.

## Trace

Please also use [Trace](#) functions. The [Trace](#) function will make sure that for every function call to the plugin, you get a log entry with parameters and result. For version 5.0 we added logging of script triggers. Trace can either log to a file or to the system debug log. So pass a native path, e.g. on Mac "/tmp/test.txt". Or simply use Console.app on Mac or install DebugView app on Windows. DebugView on Windows can be run as administrator and enabled to capture global Win32 in order to see log messages from FileMaker Server. Below a sample script to decide how to log for various cases:

```
If [Get ( SystemPlatform ) = 3]
    #iOS, no plugin
    Exit Script []
End If
If [MBS("IsServer")]
    If [MBS( "SystemInfo.isMacOSX" )]
        If [Get ( SystemPlatform ) = 4]
            #Server on Mac for WebDirect
            Set Variable [$r; Value:MBS("Trace"; "/tmp/TraceFMServerWebDirect.txt")]
        Else
            #Server on Mac
            Set Variable [$r; Value:MBS("Trace"; "/tmp/TraceFMServer.txt")]
        End If
    Else
        #Server on Windows
        Set Variable [$r; Value:MBS("Trace")]
        #read via DebugView app running as admin with Global Win32 capture
        enabled
    End If
Else
    If [MBS( "SystemInfo.isMacOSX" )]
        #Client on Mac
        Set Variable [$r; Value:MBS("Trace")]
        #read via Console.app
    Else
        #Client on Windows
        Set Variable [$r; Value:MBS("Trace")]
        #read via DebugView app
    End If
End If
```

As you see we use two log files for Mac server, depending if the plugin is loaded for web direct or normal server.

For FileMaker Server on Mac OS X you can see trace messages in /Library/FileMaker Server/Logs/stderr file.

## Registration

A common problem we see with people is wrong registration. When you work with clients and servers, it can get confusing. While the plugin allows you to use a runtime or server license for one seat to develop, please don't use a server or runtime license regularly for clients. A registration script can go like this:

```
If [Get ( SystemPlatform ) = 3]
    # iOS, no plugin to register
    Exit Script []
End If
If [MBS("isRegistered")]
    # already registered
Else
    # register
    If [MBS("IsServer")]
        Set Variable [$r; Value:Evaluate("MBS(\"Register\"; " &
MBSPlugin::LicenseKeyServer & ")")]
    Else If [MBS("isRuntime")]
        Set Variable [$r; Value:Evaluate("MBS(\"Register\"; " &
MBSPlugin::LicenseKeyRuntime & ")")]
    Else
        Set Variable [$r; Value:Evaluate("MBS(\"Register\"; " &
MBSPlugin::LicenseKeyClient & ")")]
    End If
End If
```

So first we exit early for iOS as there is no plugin for iOS. Then we check if plugin is registered already and do nothing in that case. Else we check the platform and use the appropriated registration key. In our case with Evaluate as the registration key is in a text field.

Using isRegistered function is better than a global variable and certainly it is better to call such a script from various places where you need the plugin to make sure you are registered. With FileMaker Server the script engine may crash and restart. In that case the plugin is loaded again and registration needs to be applied again.

Or you use our [StoreRegistration](#) function and save the registration to preferences file (Mac) or registry (Windows).