Debugging problems on FileMaker Cloud for AWS

Since our plugin supports AWS and we built a Linux version of it, we had questions on how to troubleshoot problems. First, the plugin can be installed twice on a FileMaker Server, once for server scripting (PSoS and Scheduled Scripts) and for Web Direct. Both are separate folders, but usually you'll just use the Install Plugin Script Step to install them automatically.

Sometimes you may want to debug a problem related to MBS FileMaker Plugin calls on a FileMaker Cloud server on AWS. So here a few notes:

Trace

First you may want to try our Trace function. This function logs calls to the MBS Plugin. You may see them on MacOS in Console.app, on Windows in DebugView application and on Linux in the log file.

When the plugin starts on Linux, it may output an info line like this:

```
MBS Plugin version 9.3.0.04 (Linux 64-bit) loaded for ServerScripting, SDK version 18.
```

You see the plugin version loaded, the platform, which part of FileMaker loaded it and the SDK version of FileMaker. Platform is ServerScripting for the scripting engine on a FileMaker Server. If the plugin is also installed for Web Direct, you may see an entry there with Web as platform.

Next you may see calls in the trace log like this one. We show the current number of the cal and the parameters:

```
MBS Plugin call #6 with 1 parameters.
Parameter 0: "ProgressDialog.Reset"
Result #6: "[MBS] This function is only available on macOS and Windows: ProgressDialog.Reset"
```

When the call finishes, the result is print. Calls taking longer may also print how long they run in seconds.

On Linux, those log entries are redirected into files to help debugging. So MBS Plugin redirect stdout into /FileMakerData/Logs/ServerScriptingPluginsStdOut.log and stderr with the error messages /FileMakerData/Logs/ServerScriptingPluginsStdErr.log. You find the trace messages in ServerScriptingPluginsStdErr.log file. For WebDirect the file names are WebPluginsStdErr.log and WebPluginsStdOut.log.

For the log files, it can be very useful to just login via ssh and use tail -f command to show new messages as they arrive in the terminal window:

```
tail -f /FileMakerData/Logs/ServerScriptingPluginsStdErr.log
```

CURL

Next if you do CURL transfer, be sure to always check the debug log messages. You query the log with MBS("CURL.GetDebugAsText"; $curl) function after you performed the transfer. In case you look for a problem which causes a crash in the curl library, you can ask the plugin to log to a file. Use the function CURL.CreateDebugOutputFile like this: MBS("CURL.CreateDebugOutputFile"; $curl; "/tmp/curl-upload.log"). This will write a log file live, so when a crash happens, you can inspect last line.

If needed, you can include data in the log with CURL.SetDebugWithData. If you use CURL.SetDebugWithTime, you also see the time stamps. With CURL.SetDebugWithProgress we can log the progress happening, so you see how far the transfer got. The whole lines

```
# create a new debug log file for live log including times and progress details, even if app crashes while the transfer:
Set Variable [ $r ; Value: MBS("CURL.SetDebugWithData"; $curl; 0) ]
Set Variable [ $r ; Value: MBS("CURL.SetDebugWithTime"; $curl; 1) ]
Set Variable [ $r ; Value: MBS("CURL.SetDebugWithProgress"; $curl; 1) ]
Set Variable [ $r ; Value: MBS("CURL.CreateDebugOutputFile"; $curl; "/tmp/curl-upload.log" ) ]
```

Crash

Well, if a crash happens in C++ code, you may get a crash report on MacOS, e.g. in ~/Library/Logs/DiagnosticReports for FileMaker Pro and in /Library/Logs/DiagnosticReports for a server. On Linux, it may be more difficult to get a crash report. So we recently added a function Plugin.InstallSignalHandlers to our plugin. It lets the plugin put in a signal handler to catch such a crash and print the backtrace before aborting the process. So it still crashes, but we print the backtrace.

```
Signal SIGSEGV
/opt/FileMaker/FileMaker Server/Database Server/Extensions/MBS.fmx(+0x540f40)
[0x7f924095bf40]
/lib64/libc.so.6(+0x36280) [0x7f926db20280]
/opt/FileMaker/FileMaker Server/Database Server/Extensions/MBS.fmx(+0x35cd61)
[0x7f9240777d61]
/opt/FileMaker/FileMaker Server/Database Server/Extensions/MBS.fmx(+0x35f041)
[0x7f924077a041]
/opt/FileMaker/FileMaker Server/Database Server/Extensions/MBS.fmx(+0x35f51a)
[0x7f924077a51a]
/opt/FileMaker/lib/
libDBEngine.so(_ZN5Draco8DBPlugIn14SafePlugInCallEPFssRKNS_15DBExpressionEnv
ERKNS_10DBDataVectERNS_6DBDataEEsS3_S6_S8_+0x11) [0x7f9272790fc1]
/opt/FileMaker/lib/
libDBEngine.so(_ZN5Draco12DBCalcSolver12SolvePluginsERNS_7DBErrorE+0x846)
[0x7f9272649cd6]
/opt/FileMaker/lib/
libDBEngine.so(_ZN5Draco12DBCalcSolver11SolvePolishERNS_9DBDataArgEt+0xe9e)
[0x7f927264317e]
/opt/FileMaker/lib/
libDBEngine.so(_ZNK5Draco13DBCalcFormula5SolveERNS_6DBDataEPKNS_8DBCursor
ES5_PKS1_PKNS_11DBFieldSpecEPKNS_7DBFieldEt+0x156) [0x7f92725b3376]
/opt/FileMaker/lib/
libFMEngine.so(_ZN5Draco13FMWindowModel16SolveCalculationERNS_6DBDataEPNS_
13DBCalcFormulaEsbPKNS_8DBCursorE+0xc2) [0x7f9273646fd2]
/opt/FileMaker/lib/
libFMEngine.so(_ZN5Draco17ScriptRuntimeBase16SolveCalculationERNS_6DBDataEss
bPKNS_8DBCursorEPNS_13FMWindowModelEb+0x112) [0x7f927366b012]
/opt/FileMaker/lib/
libFMEngine.so(_ZN5Draco17ScriptRuntimeBase13DoSetVariableEPNS_9unistringE+0x
18f) [0x7f927366f1cf]
/opt/FileMaker/lib/libFMEngine.so(_ZN5Draco17ScriptRuntimeBase7ExecuteEv+0x1f3)
[0x7f927366c583]
/opt/FileMaker/lib/
libFMEngine.so(_ZN5Draco17ScriptRuntimeBase10DoNextStepEv+0x192)
[0x7f927366c2b2]
/opt/FileMaker/lib/
libFMEngine.so(_ZN5Draco17ScriptRuntimeBase9DoRunLoopEv+0x1a6)
[0x7f927366c016]
/opt/FileMaker/lib/libFMEngine.so(_ZN5Draco15SMScriptRuntime6OnIdleEv+0x6c)
[0x7f927365fdac]
/opt/FileMaker/lib/libFMEngine.so(_ZN5Draco9FMSession6OnIdleEb+0x46)
[0x7f92736305b6]
/opt/FileMaker/lib/libFMEngine.so(_ZN5Draco9SMSession4IdleEv+0x5e)
[0x7f9273661d3e]
```

```
/opt/FileMaker/lib/
libFMEngine.so(_ZN5Draco15SMScriptRuntime14CompleteScriptEPNS_9unistringE+0x
6c) [0x7f927365f4cc]
/opt/FileMaker/lib/
libFMEngine.so(_ZN5Draco15SMScriptRuntime13ExecuteScriptERNS_10SMFileUserERK
NS_9unistringES5_PS3_+0x208) [0x7f927365f9b8]
/opt/FileMaker/lib/
libFMEngine.so(_ZN5Draco9SMSession17RunScriptAndCloseERKNS_13UniversalPathE
RNS_20DBAuthenticationInfoERKNS_9unistringES8_PS6_+0x286) [0x7f9273662096]
/opt/FileMaker/FileMaker Server/Database Server/bin/fmsased() [0x472c13]
/opt/FileMaker/FileMaker Server/Database Server/bin/fmsased() [0x4725df]
/opt/FileMaker/lib/libSupport.so(_ZN5Draco6Thread9RunThreadEv+0x2d)
[0x7f9271eb3c6d]
/opt/FileMaker/lib/libSupport.so(_ZN5Draco6Thread14ThreadUnixProcEPv+0x1bd)
[0x7f9271eb2b1d]
/lib64/libpthread.so.0(+0x7dd5) [0x7f926fa10dd5]
/lib64/libc.so.6(clone+0x6d) [0x7f926dbe7ead]
```

This stack trace shows the crash is in MBS Plugin (we caused it for this test). But for the future we hope to enable showing function names in the stack trace. Not yet sure what's missing there.

We can still detect the function names, if you provide us the crash report and the matching plugin file.