

Perform Script on Server with Audit for FileMaker Go

You may have enjoyed our [Audit](#) functions in [MBS FileMaker Plugin](#), but wondered how to use them with FileMaker Go, where our plugin doesn't run. Well, you could always build an iOS app with the FileMaker iOS SDK and include the [MBS FileMaker Plugin](#) to use [Audit](#) functions on device. But with FileMaker Go, you can only use [MBS FileMaker Plugin](#) indirectly, e.g. via Perform Script on Server, so lets show it to you.

For our Audit example we have our call to [Audit.Changed](#) function to do the audit bookkeeping with the plugin. We pass the timestamp field, the table name and any custom values you like, e.g. \$\$InsideScript, which is a variable set in a script. You can also use an If conditions to disable audit if needed:

```
If ($$AuditDisabled; 0;  
MBS("Audit.Changed"; AuditTimeStamp; "Audit"; "InsideScript|" & $  
$InsideScript))
```

For FileMaker Go the calculation above will produce an error. Let's write a script, which runs on record commit as layout script trigger:

Trigger Audit on Server in file Audit

```
Set Error Capture [ On ]  
#  
# check for missing plugin, e.g. on FileMaker Go  
If [ GetAsText(MBS("Version")) = "?" ]  
    Perform Script on Server [ Specified: From list ; "Audit on  
server" ; Parameter: Get(LayoutName) & ¶ & Get(RecordID) & ¶ &  
Get(Username) ; Wait for completion: Off ]  
End If
```

As you see we turn on error capture to avoid an error for calling plugin function if it's missing. When our version function returns a question mark, the plugin is not there. In that case we perform a script on the server and pass the layout name, the record ID and the user name. The script on the server will run as a new session, so we need to restore context and go to the right layout, the right record and perform Audit there. Some values may not be the same when PSoS runs, so you can e.g. pass an extra Username value here.

Audit on server in file Audit

```
# give FileMaker Go some time to save
Pause/Resume Script [ Duration (seconds): ,5 ]
#
# get parameter with layout and record id
Set Variable [ $Param ; Value: Get(ScriptParameter) ]
Set Variable [ $LayoutName ; Value: GetValue($Param; 1) ]
Set Variable [ $RecordID ; Value: GetValue($Param; 2) ]
Set Variable [ $UserName ; Value: GetValue($Param; 3) ]
#
# go to record
Go to Layout [ $LayoutName ; Animation: None ]
Go to Record/Request/Page [ With dialog: Off ; $RecordID ]
#
# log changes
Set Variable [ $r ; Value: MBS("Audit.Changed";
Audit::AuditTimeStamp; "Audit"; "ServerUserName|" & $UserName) ]
Exit Script [ Text Result: $r ]
```

In this script we start with a script pause as our script otherwise may run before the record is saved. The on commit record script trigger runs before the commit is done and we miss a trigger for after the record saved.

Next we split parameter to layout name, record ID and user name. We go to the layout and the record via ID to restore context. Then we can run [Audit.Changed](#) on the record as usual and maybe also populate an extra field in the AuditLog. For our example we populate ServerUserName with the user name from the device.

If you customize this, please consider using [Audit.Changed2](#) where you pass the list of fields to match, so we can skip gathering the field list.

Please do not hesitate to contact us with questions.

See also older blog post: [Audit with MBS FileMaker Plugin](#)