



Productive
Computing

Creating Efficiency Through Automation



Migration Guide

FM Books Connector Online to FM Books Connector Online Server-Side

February 21, 2017

950 Boardwalk, Suite 205, San Marcos, CA 92078 • (760) 510-1200 • www.productivecomputing.com

© Copyright 2017 Productive Computing, Inc.

As of version 2.0.1.0, the FM Books Connector Online Edition plug-in is able to run on FileMaker Server within the FileMaker Server Scripting Engine or Web Publishing Engine. In order for a solution to utilize the plug-in in the Scripting Engine or Web Publishing Engine, there may be changes required in the scripting and integration in the handling of the plug-in. The following is a useful document that should help a developer prepare their solution for posting financial information to QuickBooks Online from their FileMaker Solution using server-side processes.

Plug-in Installation

As mentioned in the Developer's Guide, the server-side plug-in must be installed on the server in order to be utilized. If the desire is to use the plug-in for Perform Script on Server or in server-side scheduled scripts, the plug-in must be installed in the scripting engine's Extensions folder, located within the "Database Server" folder in FileMaker Server's install directory. For WebDirect, this Extensions folder would be in the "cwpc" folder (Custom Web Publishing Core) in the Web Publishing Engine. See the paths below for the plug-in installation locations on both Windows and Mac servers (assuming a default installation path):

Windows:

Database Server:

C:\Program Files\FileMaker\FileMaker Server\Database Server\Extensions

Web Direct & Custom Web Publishing:

C:\Program Files\FileMaker\FileMaker Server\Web Publishing\publishingengine\cwpc\Plugins

Mac:

Database Server:

/Library/FileMaker Server/Database Server/Extensions

Web Direct & Custom Web Publishing:

/Library/FileMaker Server/Web Publishing/publishing-engine/cwpc/Plugins

An important note: In order to allow FileMaker Server to install and use a server-side plug-in from a hosted database, both options in the Database Server > Server Plug-ins section in the Admin Console must be checked, as well as both options in the Web Publishing > General Settings section. To make installation easy, you can use the server-side demo file to install the plug-in automatically, whether on Web Direct or via Perform Script on Server.

Authentication to QuickBooks Online

The FM Books Connector Online server-side plug-in cannot initiate or authenticate a new session with Intuit QuickBooks Online on its own. This is in part due to the restriction for FileMaker server-side plug-ins that prevents the use of external dialog windows or pop-ups. In order to allow the FM Books Connector Online server-side plug-in to communicate with QuickBooks Online while respecting the FileMaker server-side plug-in restrictions, the information of a session must be passed to the server-side plug-in during any scripted process.

A client-side version of FM Books Connector Online must proceed with beginning and authenticating a session with QuickBooks Online. Once this is successful, the solution must call “PCQO_SSaveSessionInfo” and store the resulting encrypted session information string within a field in the hosted FileMaker solution. When the server-side plug-in is then called to perform a task (such as a user accessing the solution via FileMaker Go and seeking to push a set of invoices to QuickBooks Online), the scripted process must load that stored session information string by calling “PCQO_SLoadSessionInfo”. If the result of the call is 0, the connection has been established and the server-side plug-in is able to communicate freely with QuickBooks for the duration of the scripted process.

Conversion of Scripts

Depending on the intended deployment of the FM Books Connector Online server-side plug-in, scripts utilizing the plug-in may need to be changed.

Perform Script on Server

When a client calls Perform Script on Server, the action sends the desired script to allow the FileMaker Server Scripting Engine to perform the heavy lifting of calculation and data manipulation, and return the result back to the caller. For handling plug-in usage, the following is a handy checklist of what to ensure the script that is performed on the server should do:

- 1) Ensure the script space ends up in the right table occurrence context, with the appropriate record(s) in the found set and accessed
- 2) Perform a call to PCQO_Register with valid registration information to put the plug-in into the “Live” operating mode
- 3) Perform a call to PCQO_SLoadSessionInfo with an open and active session created by an earlier call from the client plug-in, as mentioned in “Authentication to QuickBooks Online”

See below for an example of a usage of Perform Script on Server, using scripts from the PCQO_Demo_Server-side demo file:

Script 1: Pull All Customer Names [Server]

```
# Ensure the solution is hosted
If [IsEmpty( Get(HostIPAddress) )]
    Show Custom Dialog ["File Not Hosted"; "This solution must be hosted to use this
feature."]
    Halt Script
End If

If [$$IsWebDirect]
    # $$IsWebDirect is a true or false flag that is set to true if the file is
    # being accessed via Web Direct. This would be set in the file's opening script.

    # Perform the script for Web Direct. Web Direct uses "client"-style scripting
    # just as well as Perform Script On Server.
    Perform Script ["SVR – Pull All Customers Names"]
Else
    # Perform the script on the server
    Perform Script on Server [Wait for completion; "SVR – Pull All Customers Names"]
End If
```

```

# Get the result and check for error
If [Get(ScriptResult) <> 0]
    Show Custom Dialog ["Pull Failed"; "Failed to pull customer names: " &
Get(ScriptResult)]
End If

```

Script 2: SVR – Pull All Customers Names

```

# Ensure the plug-in is registered. In this demo, we use subscripts for better readability.
Perform Script ["SVR – Register Plug-in"]
If [Get(ScriptResult) <> 0]
    # There was an error with registration. Return the result back to the client.
    Exit Script [Text Result:Get(ScriptResult)]
End If

# Ensure the plug-in has an authorized connection.
Perform Script ["SVR – Load Session"]
If [Get(ScriptResult) <> 0]
    # There was an error with loading the session. Return the result back to the client.
    Exit Script [Text Result:Get(ScriptResult)]
End If

# Clear any existing customer records
Go to Layout ["Customer List" (Customers)]
Show All Records
Delete All Records [With dialog:Off]

# Create the request (we want all customers in the QuickBooks Online company)
Set Variable [$$Result; Value: PCQO_RqNew( "Query" ; "Customer" )]

# Execute the request
Set Variable [$$Result; Value: PCQO_RqExecute]
If [$$Result <> 0]
    # There was an error during execution. Return the result back to the client.
    Exit Script [Text Result:PCQO_SGetStatus]
End If

# Process the response and create records in FileMaker
Set Variable [$$Result; Value: PCQO_RsOpenFirstRecord]
Set Variable [ $Count; Value: 0 ]
Loop
    Exit Loop If [ $$Result = "!!ERROR!!" or $$Result = "End" or $$Result = "?" or $Count
= 100 // Only pull the first 100 customers]
    New Record/Request
    Set Variable [ $Count; Value: $Count + 1 ]
    Set Field [Customers::QB_List_ID; PCQO_RsGetFirstFieldValue( "Id" )]
    Set Field [Customers::Customer_Name;
PCQO_RsGetFirstFieldValue( "FullyQualifiedName" )]
    Set Field [Customers::Total_Balance; PCQO_RsGetFirstFieldValue( "BalanceWithJobs" )]
    Set Variable [ $$Result; Value: PCQO_RsOpenNextRecord]
End Loop

# At the end of the script, since we would have exited if there was an issue, return 0.
Exit Script [Text Result:0]

```

Script 3: SVR – Register Plug-in

```
# Register the plug-in using the license information stored in the solution
Set Variable [$result; Value: PCQO_Register( Main::RegistrationServer;
Main::RegistrationPort; Main::RegistrationPage; Main::RegistrationLicenseID )]
If [$result <> 0]
    Exit Script [Text Result:$result]
Else
    Exit Script [Text Result:0]
End If
```

Script 4: SVR – Load Session

```
# Check if the session information is stored
If [IsEmpty( Main::Session Info String )]
    Exit Script [Text Result:"No session information is saved."]
End If

# Load the QBO session using the session information stored in the solution
Set Variable [$result; Value: PCQO_SLoadSessionInfo( Main::Session Info String )]
If [$result <> 0]
    Exit Script [Text Result:PCQO_SGetStatus]
Else
    Exit Script [Text Result:0]
End If
```

Script #1 is considered the “Driver” script. This kind of script is usually tied to a button control or some other interactive object, and its purpose is to ensure that the server-side script has what it needs from the client, and then calls the “Worker” script using Perform Script on Server. In a WebDirect environment, the “Driver” script can simply call the “Worker” script directly, or it can call Perform Script on Server if the developer decides to make the server scripting engine’s plug-in perform the work.

Script #2 is considered the “Worker” script. This script is the entire scope of the script session, and is the “scripted process” mentioned in the earlier sections. At the beginning of the Worker script, a call to register the server-side plug-in (Script #3) and load a stored QuickBooks Online session (Script #4) must be made successfully before any further progress can be made. The actual work scripting is identical between the client and server styles, with the notable exception being that there can be no dialog windows used when running in the server session.

Once the Worker script has completed all of its script steps, control will return back to the Driver script, and the server-side plug-in will release the registration and session information that it has and return to the original “neutral” state it was in before the call to the Worker script. You can call multiple Worker scripts *sequentially* and each call will register the plug-in, load the session information, and perform its task before returning to the Driver script.

Server-Side Scheduled Scripts

Scheduled scripts would follow similar adjustments to the order of scripting as scripts called by Perform Script on Server. The “driver” script, in this case, could be largely ignored, but the script space still needs to be in the correct context, PCQO_Register must still be called, and PCQO_SLoadSessionInfo must load an active QuickBooks Online connection before any work can commence.

WebDirect Script

At a glance, scripting in WebDirect and scripting in FileMaker Pro are identical when it comes to plug-in calls and how they are structured. The only critical difference between WebDirect scripting and standard FileMaker Pro scripting is the requirement of loading the session information for the QuickBooks Online connection. All other features, including registration and transaction handling, are the same.

Converting the scripts of a solution to work in a server-side aspect should not be a significant undertaking, but as with any scripting customization, the developer should review the requirements of the system in order to achieve the best results.

Additional Support

If you require additional support in your migration from FM Books Connector to FM Books Connector Online for integration with QuickBooks Online, or would like to start a new integration between FileMaker and Intuit® QuickBooks Online, please contact us via the avenues listed below.

Phone: 760-510-1200

Email: support@productivecomputing.com

Forum: www.productivecomputing.com/forum

Please note assisting you with implementing this plug-in (excluding registration) is billable at our standard hourly rate. We bill on a time and materials basis billing only for the time in minutes it takes to assist you. We will be happy to create your integration scripts for you and can provide you with a free estimate if you fill out a Request For Quote (RFQ) at www.productivecomputing.com/rfq. We are ready to assist and look forward to hearing from you!