



Productive
Computing Creating Efficiency Through Automation



Functions Guide

Revised January 29, 2021

Table of Contents

I. INTRODUCTION	3
II. FUNCTION DESCRIPTIONS	4
A. Session Functions.....	4
1) PCQB_BeginSession(CompanyFile ; ShareMode ; optHideFMWindow).....	4
2) PCQB_EndSession	6
3) PCQB_Browse	6
4) PCQB_GetCurrentFileName.....	6
5) PCQB_GetCurrentFileVersion	6
6) PCQB_GetCurrentQBVersion	7
B. Request Functions	8
1) PCQB_RqNew(MessageType ; optAttributes).....	8
2) PCQB_RqAddFieldWithValue(QBFieldName ; Value ; optAttributes)	8
3) PCQB_RqAddRelatedRecord(Type ; optValue ; optAttributes)	9
4) PCQB_RqCloseRelatedRecord	9
5) PCQB_RqExecute (optHideFMWindow ; optFilter ; optPath).....	10
6) PCQB_RqUseXML(XMLText ; optIsPath).....	10
7) PCQB_RqAddRequestToBatch	11
C. Response Functions.....	12
1) PCQB_RsOpenFirstRecord	12
2) PCQB_RsOpenNextRecord	12
3) PCQB_RsGetFirstFieldValue(QBFieldName).....	12
4) PCQB_RsGetAttribute(AttributeName)	13
5) PCQB_RsGetResponseAttribute(AttributeName).....	14
6) PCQB_RsOpenFirstRelatedRecord(Type)	14
7) PCQB_RsOpenNextRelatedRecord	15
8) PCQB_RsCloseRelatedRecord.....	15
9) PCQB_RsUseXML(XML ; optIsPath ; optFilter)	15
10) PCQB_RsOpenFirstBatchRecord.....	16
11) PCQB_RsOpenNextBatchRecord	16
D. Other Functions	17
1) PCQB_SGetXML(Type ; optPath)	17
2) PCQB_SGetStatus	17
3) PCQB_SFormatString(String ; qbType)	18
4) PCQB_ValidateRequests(bParam).....	18
5) PCQB_SClearBatchQueue(Type).....	19
6) PCQB_SToggleApplyFilters(Boolean).....	19
E. Registration Functions.....	20
1) PCQB_GetOperatingMode.....	20
2) PCQB_Register(ServerName ; ServerPort ; ServerPage ; LicenseID).....	20
3) PCQB_Version(Type)	21
F. Z Functions (FileBooks Link Migration Functions).....	22
1) PCQB_ZCloseConnection	22
2) PCQB_ZDeleteFile(FilePath).....	22
3) PCQB_ZExecute(XPath ; qbXML ; optFilePath).....	22
4) PCQB_ZModify(XPath ; Object ; SearchCriteria ; Modifications ; optFilePath)	23
5) PCQB_ZParseXML(XML ; IsPath ; Xpath ; optFilePath)	23
6) PCQB_ZQuery(XPath ; Object ; optSearchCriteria ; optFilePath).....	24
7) PCQB_ZValidateXML(XML ; IsPath; optWellFormedOnly)	24
III. CONTACT US.....	25

I. Introduction

Description:

The FM Books Connector plug-in is a powerful tool used to move data between FileMaker® Pro and Intuit® QuickBooks applications. In this Functions Guide we will describe all available FileMaker functions, parameters, return values and error handling.

Intended Audience:

Intermediate to advanced developers or persons with knowledge of FileMaker Pro or FileMaker Pro Advanced, especially in the areas of scripting, calculations and relationships as proper use of the plug-in requires that FileMaker integration scripts be created in your FileMaker solution.

Successful Integration Practices:

- 1) Familiarize yourself with basic accounting practices and QuickBooks
- 2) Read the Developer's Guide:
https://www.productivecomputing.com/documents/FM_Books_Connector/Developers_Guide_FM_Books_Connector.pdf
- 3) Read the Functions Guide:
https://www.productivecomputing.com/documents/FM_Books_Connector/Functions_Guide_FM_Books_Connector.pdf
- 4) Review articles in our help center:
<https://help.productivecomputing.com/help/fm-books-connector>
- 5) Download the demo:
<https://www.productivecomputing.com/products/filemaker-quickbooks-integration/>
- 6) Watch our video tutorials on YouTube for installing the plug-in and using the demo file:
<https://www.youtube.com/watch?v=zSEh8MerdSw&list=PLF9936A5DB331BCF2>
- 7) Purchase our training course for more detailed instructions:
<https://www.productivecomputinguniversity.com/courses/fm-books-connector-for-quickbooks-desktop>
- 8) Use the OSR (Intuit's Onscreen Reference Manual):
OSR: <https://static.developer.intuit.com/qbSDK-current/common/newosr/index.html>

II. Function Descriptions

This section of the document describes the available FileMaker plug-in functions for FM Books Connector.

A. Session Functions

1) PCQB BeginSession(CompanyFile ; ShareMode ; optHideFMWindow)

Description:

This function begins a session working with the specified QuickBooks company file in the specified mode.

Parameters:

CompanyFile = Full path to the desired QuickBooks company file.

If QuickBooks is already launched with an open company file, this value can be an empty string and FM Books Connector will attach to the currently open company file. If no file is open and no company file is specified, an error will occur. If the company file opened is different than the specified file path, than an error will occur. (If this call succeeds with an empty string for this parameter, or an error is returned, you can obtain the name of the currently open company file by calling the PCQB_GetCurrentFileName function.)

Share Mode = Identifies if QuickBooks will be available to other users while the plug-in has it opened. See Table 1 below for the effects of the QuickBooks mode.

Following are acceptable values:

"Single" which allows the FM Books Connector exclusive access to the QuickBooks file or shares access with the current user.

"Multi" which allows the FM Books Connector to share access to the specified company file with other applications, as well as with a user who is interacting directly with the company file through QuickBooks.

If left blank, then a default "DoNotCare" is used which allows either single-user or multi-user mode. If the specified company file is already open, QuickBooks allows the authentication process to proceed. If no company file is open, QuickBooks opens the company file in multi-user mode. If a different company file is already open, an error is returned.

optHideFMWindow = Optional - If this parameter = "Y" then the FileMaker Pro window is hidden from the user until the function returns. A dialog is displayed in its place. This should be used when execution takes longer than two (2) seconds.

Return Values:

0 for success or !!ERROR!!

Table 1 : Effects of mode on QuickBooks operation

Who started QuickBooks	Selected Mode	Who else may obtain access
FM Books Connector	"Single"	No one else
FM Books Connector	"Multi"	QuickBooks user on same machine = no access All other integrated applications = access QuickBooks users on other machines = access
QuickBooks User	"Single"	QuickBooks on same machine = access Only one integrated application = access
QuickBooks User	"Multi"	QuickBooks users = access Integrated applications = access

2) PCQB EndSession

Description:

This function ends the current session with the QuickBooks file. After the session is ended, file access is returned to the default for the file.

Parameters: None.

Return Values:

0 for success or !!ERROR!!

3) PCQB Browse

Description:

This function displays file browser dialog for locating a QuickBooks company file. The full path to the select file is returned on 'OK'.

Parameters: None.

Return Values:

A full path to the selected file is returned.

4) PCQB GetCurrentFileName

Description:

This function returns the name of the file that is currently attached to the session. There must be a session established in order to return the file name.

Parameters: None.

Return Values:

The name of the QuickBooks company file is returned.

5) PCQB GetCurrentFileVersion

Description:

Use this function to determine which SDK version is supported by the currently opened file. This function is generally used for development and troubleshooting purposes.

Parameters: None.

Return Values:

The SDK Version in string format or !!ERROR!!

6) PCQB GetCurrentQBVersion

Description:

This function returns detailed information about the user's QuickBooks installation and is primarily meant for debugging purposes. In order to use this function you must have a session opened with QuickBooks.

Parameters: None.

Return Values:

Returns the string "Unknown" if the plug-in is not synchronized to any QuickBooks installation. Otherwise detailed version information about the QuickBooks installation is returned.

B. Request Functions

1) PCQB RqNew(MessageType ; optAttributes)

Description:

This function creates a new request of the desired type.

Parameters:

MessageType = See the 'OSR' for valid Message Types and meanings.

opt Attributes = Allows the user to set the attributes of particular nodes in underlying qbXML. Available Attributes for any node are listed in the OSR next to the node in parenthesis.

Return Values:

0 for success, otherwise !!ERROR!!

2) PCQB RqAddFieldWithValue(QBFieldName ; Value ; optAttributes)

Description:

This function is used to add a field and value to the current request. The function will automatically format the Value parameter for use with QuickBooks if the field identified by the QBFieldName function is of one of the following types:

BOOLTYPE, DATETYPE, DATETIMETYPE, QUANTTYPE, PERCENTTYPE, PRICETYP, AMTTYPE, INTTYPE
Other types are formatted as entered.

Parameters:

QBFieldName = The name of the QuickBooks field to be added to the request. See the 'OSR' for valid field names.

Value = The value to be inserted for the specified field.

opt Attributes = Allows the user to set the attributes of particular nodes in underlying qbXML. Available Attributes for any node are listed in the OSR next to the node in parenthesis.

Example for an Invoice: PCQB_RqAddFieldWithValue ("RefNumber" ; TheirInvoicenumber)

Return Values

0 for success, otherwise !!ERROR!!

3) PCQB_RqAddRelatedRecord(Type ; optValue ; optAttributes)

Description:

This function adds related records to the request. The related record becomes active until the PCQB_RqCloseRelatedRecord function is called. Calls to PCQB_RqAddFieldWithValue and PCQB_RqAddRelatedRecord will add fields and records to the currently active record until PCQB_RqCloseRelatedRecord is called.

****The RqAddRelatedRecord now behaves identical to the RqAddFieldWithValue with the exception that it will create new nodes and apply the optValue to the new node instead of overwriting any current nodes. This is very useful when dealing with 1-n nodes that contain values. See the FullName node in the GeneralDetailReportQuery object in the OSR**.**

Parameters:

- Type = The type of related record to be added.
- optValue = The value to be inserted for the specified field.
- opt Attributes = Allows the user to set the attributes of particular nodes in underlying qbXML.
Available Attributes for any node are listed in the OSR next to the node in parenthesis.

Returns:

0 for success, otherwise !!ERROR!!

4) PCQB_RqCloseRelatedRecord

Description:

This function closes the related record and sets the active record to its parent.

Parameters: None.

Returns:

0 for success, otherwise !!ERROR!!

5) PCQB RqExecute (optHideFMWindow ; optFilter ; optPath)

Description:

This function posts the request to QuickBooks. If the request is processed by QuickBooks a set of records will be created in a response and held in memory for parsing. If the request is not valid or some other error has occurred then an error is returned.

Parameters:

- optHideFMWindow = Optional - If this parameter = "Y" then the FileMaker Pro window is hidden from the user until the function returns. A dialog is displayed in its place. This should be used when execution takes longer than two (2) seconds.
- optFilter = Optional XPath string that will filter the response from QuickBooks. If left empty or omitted, then the response is not filtered.
- optPath = Optional absolute path where the response will be saved as a file. If empty or omitted, then the response is not saved to a file.

Returns:

0 for success, otherwise !!ERROR!!

Notes:

On receiving a status code 530 response, function result will be "!!WARNING!!". Response will still be processed as though a "0" result is received.

6) PCQB RqUseXML(XMLText ; optIsPath)

Description:

The FM Books Connector uses XML to communicate with QuickBooks. The several functions offered by the plug-in are used to create the qbXML requests. In some scenarios the end user might want to use pre-defined qbXML requests. This function allows the user to pass a pre-defined qbXML request to the plug-in.

Parameters:

- XMLText = The entire XML text that will be used as the request or a path to an XML file.
- optIsPath = Optional Boolean value indicating if XMLText is a path to a file or if XMLText contains actual XML data. Default value is false.

Returns:

0 for success, otherwise !!ERROR!!

7) PCQB RqAddRequestToBatch

Description:

This function adds the current request in memory to the batch queue. Once added, the request will be "closed" to editing, meaning that it cannot be defined any further without first clearing the batch queue and starting again. Requests are added in queue order: the first request in is the first request to be executed, and after execution, the first request matches with the first response in the response batch queue.

Parameters:

None

Returns:

0 for success, otherwise !!ERROR!!

Notes:

The batch only holds up to 10 records. After that point, PCQB_RqExecute should be called and the results handled, before the next batch is submitted. This is to avoid buffer overflow and help manage resource consumption by the plug-in.

C. Response Functions

1) PCQB RsOpenFirstRecord

Description:

Since a response may include several records it is necessary to be able to iterate through these records. This function will access the first record in the response. Use PCQB_RsOpenNextRecord to iterate through the rest of the records in the response.

Parameters: None.

Return Values:

0 for success, 'End' if there are no records in the response, or !!ERROR!!

2) PCQB RsOpenNextRecord

Description:

This function opens the sibling record of the current record, or the next record in the response. The record becomes the active record for reading.

Parameters: None.

Returns:

0 for success, 'End' if there are no more records in the response, or !!ERROR!!

3) PCQB RsGetFirstFieldValue(QBFieldName)

Description:

Returns the value stored in the first field with the name specified by Name

Parameters:

QBFieldName = The name of the field from which to retrieve the value.
 See the 'OSR' for valid field names.

Returns:

The value stored in the field or !!ERROR!!

4) PCQB_RsGetAttribute(AttributeName)

Description:

This function returns the attribute from a response XML element (such as a field or related record). Attributes are specified as key=value properties within the brackets of an XML element.

Parameters:

AttributeName = Name of the XML attribute to retrieve

Returns:

Value of the attribute, or "!!ERROR!!" for error

Notes:

Attributes are key-value pairs, adding context or details about a given element. Not every XML element has an attribute. Below is an example of an attribute, using a Report-type response:

```
<ColDesc colID="1" datatype="dtSTRTYPE">
    <ColTitle titleRow="1" value="Account"/>
    <ColTitle titleRow="2" value="Name"/>
</ColDesc>
```

To access the inner attributes of an element, the plug-in would need to first get into the context of the element in question, and then extract the attribute. This would be performed as follows, using the above XML:

```
PCQB_RsOpenFirstRelatedRecord( "ColDesc" )
PCQB_RsGetAttribute( "colID" )
PCQB_RsGetAttribute( "datatype" )
PCQB_RsCloseRelatedRecord
```

It is important to note that even if a field is not a related record, it can still be entered into via the RsOpenFirstRelatedRecord function, and doing so is needed when trying to access the inner attribute data.

5) PCQB RsGetResponseAttribute(AttributeName)

Description:

This function returns the value of the attribute from a QXML response, specifically focusing on the details of the primary response element. This function is useful in extracting information about the status, any error messages, as well as for use in pagination.

Parameters:

AttributeName = Name of the XML response attribute to retrieve

Returns:

Value of the attribute, or "!!ERROR!!" for error

Notes:

Below is an example of a response XML document with response attribute information specified:

```
<?xml version="1.0"?>
<QXML>
<QXMLMsgsRs>
<AgingReportQueryRs statusCode="0" statusSeverity="Info" statusMessage="Status OK">
<ReportRet>
<ReportTitle>A/R Aging Summary</ReportTitle>
<ReportSubtitle>As of December 15, 2019</ReportSubtitle>
<ReportBasis>Accrual</ReportBasis>
<NumRows>52</NumRows>
```

...

The attributes "statusCode", "statusSeverity" and "statusMessage" are response attributes. Other response attribute values commonly found in QXML responses are:

- iterator
- iteratorID
- iteratorRemainingCount
- requestID
- retCount
- statusCode
- statusSeverity
- statusMessage

6) PCQB RsOpenFirstRelatedRecord(Type)

Description:

This function sets the active record to the first record of the specified type.

Parameters:

Type = The type of related record to open.
See the 'OSR' for valid Types aka Element Names.

Returns:

0 for success or !!ERROR!!

7) PCQB RsOpenNextRelatedRecord

Description:

This function opens the sibling record of the current record, or the next related record. The record becomes the active record for reading.

Parameters: None.

Returns:

0 for success, 'End' if there are no more records in the response, or !!ERROR!!

8) PCQB RsCloseRelatedRecord

Description:

This function closes the related records and returns to the parent record. The parent becomes the active record and is available for parsing using the several Response functions.

Parameters: none

Returns:

0 on success or !!ERROR!!.

9) PCQB RsUseXML(XML ; optIsPath ; optFilter)

Description:

The FM Books Connector uses XML to communicate with QuickBooks. Several functions offered by the plug-in are used to create the qbXML requests. In some scenarios the user might want to use pre-defined qbXML requests. This function allows the user to pass a pre-defined qbXML response to the plug-in.

For example: This function is used to load a valid response from QuickBooks into the parser. Once the response is loaded in the parser, it is available to the other PCQB_RsXXXX functions.

Parameters:

XML =	The entire XML text that will be used as the request or path to XML file.
optIsPath =	Optional Boolean value indicating if XML Text is a path to a file or if XML Text contains actual XML data. Default value is false.
optFilter =	Optional XPath string that will filter the response from QuickBooks. If left empty or omitted the response is not filtered.

Returns:

0 on success or !!ERROR!!

10) PCQB RsOpenFirstBatchRecord

Description:

This function opens the first response record after executing a batch of requests. If the plug-in's last call to PCQB_RqExecute processed a batch request queue, the function will load the first response into memory for access with the other Response functions.

Parameters:

None

Returns:

0 on success or !!ERROR!!

11) PCQB RsOpenNextBatchRecord

Description:

This function opens the next response record in the response queue, loading it into memory for access with the other Response functions.

Parameters:

None

Returns:

0 on success or !!ERROR!!

D. Other Functions

These functions are primarily for debugging purposes.

1) PCQB_SGetXML(Type ; optPath)

Description:

This function returns the request or response XML text in its entirety.

Parameters:

Type = Either "Request" to get the current request text, "Response" to get the response text, "BatchRequests" to get the list of requests for the batch operation, or "BatchResponses" to get the list of responses for the batch operation.

optPath = Optional absolute path where the XML will be saved as a file. If omitted then the XML is not saved to a file.

Returns:

The XML text or !!ERROR!!

2) PCQB_SGetStatus

Description:

This function is used to retrieve a verbose explanation of errors. Immediately after an error is returned please call PCQB_SGetStatus to obtain a full description of the error. This function returns the message associated with the last error. The status is used to identify errors in the request or the processing of requests. The text returned by this function helps the user to troubleshoot script or logic failures.

Parameters: None.

Returns:

This function returns a description of the last error to occur.

Notes:

PCQB_SGetStatus will return the proper error result.

Code: 530

Severity: Warning

Message: <message from QB detailing what caused the 530 response>

3) PCQB SFormatString(String ; qbType)

Description:

This function returns the String parameter properly formatted for use with the desired qbType.

This function is used to format a string for use with a specific QuickBooks field type. The function returns the exact same value that will be used with the PCQB_RqAddFieldWithValue function.

Parameters:

String = The value to be formatted.
 "Response" to get the response text.

qbType = The qbXML /QuickBooks type for which the string will be formatted.
 Acceptable values are: BOOLTYPE, DATETYPE, DATETIMETYPE, QUANTYPE,
 PERCENTTYPE, PRICETYPE, AMTTYPE, INTTYPE, TIMEINTERVALTYPE,
 STRTYPE, TEXT, TIME, DATE, INTERVAL.

****The values "TEXT", "TIME", "DATE" and "INTERVAL" convert the first parameter FROM QuickBooks format to FileMaker format instead of converting from FileMaker to QuickBooks format.****

Returns:

The string properly formatted for the desired qbType or !!ERROR!!. An empty string is returned for empty values passed to the function as the first parameter.

4) PCQB ValidateRequests(bParam)

Description:

This function toggles the validation of XML by the plug-in on PCQB_RqExecute.

Parameters:

bParam = Boolean value to toggle validation.

Returns:

0 for success, otherwise !!ERROR!!.

5) PCQB_SClearBatchQueue(Type)

Description:

This function clears the specified batch queue, emptying it of requests or responses. We advise calling this function at the end of any batch process, as it will ensure that there will be no record creep between batch operations.

Parameters:

Type = Either "Requests", "Responses", or "Both". If left blank, default value will be "Both".

Returns:

0 for success, otherwise !!ERROR!!.

6) PCQB_SToggleApplyFilters(Boolean)

Description:

This function will instruct the FM Books Connector plug-in to either apply (if Boolean is "True") or not apply (if Boolean is "False") the XML special character filter whenever a field value is added to the QBXML request. By default, any value added to a QBXML request in memory will have any characters that are considered "special characters" by XML standards escaped (as though this function was called with a Boolean value of "True"). Disabling the special character filter can be a useful way to troubleshoot instances where users are encountering "double-escaped" characters, such as "&" or """ in QuickBooks.

Parameters:

Boolean = Boolean value. Either "True" to apply the filter or "False" to not apply the filter.

Returns:

0 for success, otherwise "!!ERROR!!"

Notes:

This function must be called before any call to PCQB_RqAddFieldWithValue in order for the filter settings to be applied. It is advised that if this function is necessary, it should be called upon opening the solution.

Example: If we have a field "Name" with a value of "John & James Shop" and the special character filter is being applied, the corresponding XML would read: <Name>John & James Shop</Name>. If special character filtering is NOT being applied, the corresponding XML would read: <Name>John & James Shop</Name>.

E. Registration Functions

The plug-in offers the following functions for registration and plug-in checker purposes.

1) PCQB GetOperatingMode

Description:

Used to identify the operating mode of the plug-in. Operating mode indicates if the plug-in has been properly registered.

Parameters: None.

Returns:

"DEMO" for demo mode, "LIVE" for live mode, or "UNREGISTERED" for unregistered mode, or "EXPIRED" for expired mode after 2 hours or 30 days timeout.

2) PCQB Register(ServerName ; ServerPort ; ServerPage ; LicenseID)

Description:

The plug-in must be registered with Productive Computing, Inc. before use. This function registers the plug-in. The values for each of the parameters of this function are supplied by Productive Computing, Inc.

Parameters:

ServerName	=	"licensing.productivecomputing.com"
ServerPort	=	"80"
ServerPage	=	"/PCIReg/pcireg.php"
LicenseID	=	"your license ID or "DEMO-FMBC"

Returns:

0 for success, otherwise "-10" and an error message.

3) PCQB Version(Type)

Description:

Used to identify the product and product version installed on a machine.

Parameter:

type = 'Short' or 'Long'.

The short version string includes only the version number of the plug-in. For example: "4.0.0.2".

The long includes the plug-in name and version number. For example: "FM Books Connector 4.0.0.2"

The default version string returned is short.

Returns:

The version string and/or name of the plug-in.

F. Z Functions (FileBooks Link Migration Functions)

The following 'Z' functions were created for the sole purpose of migrating from FileBooks Link to FM Books Connector. These functions should only be used by developers who are following the procedures from the accompanying Migration Guide.

1) PCQB ZCloseConnection

Description:

Ends any session with QuickBooks and closes the connection to the QuickBooks application.

Parameters: None.

Returns:

0 on success otherwise !!ERROR!!

2) PCQB ZDeleteFile(FilePath)

Description:

The function deletes a file on the local system.

Parameters:

FilePath = The path to the file to be deleted.

Returns:

0 for success, otherwise !!ERROR!!

3) PCQB ZExecute(XPath ; qbXML ; optFilePath)

Description:

The function posts the qbXML to QuickBooks as a request.

Parameters:

XPath = Contains the XPath query to be applied to the desired XML.
qbXML = Valid qbXML formatted text.
optFilePath = Optional parameter indicating a file to save the results of the XPath query. If left empty (or if the final semicolon and parameter are omitted entirely) the results will not be saved.

Returns:

On success the result of the XPath query applied to the response or the error string !!ERROR!!

4) PCQB ZModify(XPath ; Object ; SearchCriteria ; Modifications ; optFilePath)

Description:

The function modifies an existing QuickBooks record based on the parameters passed to the function. A query is performed to locate the record, then the modifications are applied to the found record.

Parameters:

XPath =	Contains the XPath query to be applied to the desired XML.
SearchCriteria =	Contains XML fragments that will be applied to the search. For example, when searching for a customer whose name is "Sample Customer", the text required in the search criteria parameter would simply be, "<FullName>Sample Customer</FullName>".
Object =	Any object in QuickBooks that responds to the query request, such as Host, Company, Customer, Invoice, and many more. See the appendix for available objects.
Modifications =	Contains XML fragments that will be applied to the record found by the Object query. For example, when modifying the phone number the value for the parameter would be "<Phone>somePhoneNumber</Phone>".
optFilePath =	Optional parameter indicating a file to save the results of the XPath query. If left empty the results will not be saved.

Returns:

The result of the XPath query or the error string !!ERROR!!

5) PCQB ZParseXML(XML ; IsPath ; Xpath ; optFilePath)

Description:

The function returns the result of an XPath query applied to an XML document.

Parameters:

XML =	Contains the text of the XML document or a path to an XML document stored on the local hard disk. This is the XML that will be queried with the XPath.
IsPath =	A Boolean operator indicating whether or not the XML argument is a path to a file. A 'True' value indicates the XML argument is a path to a file.
XPath =	Contains the XPath query to be applied to the desired XML.
optFilePath =	Optional parameter indicating a file to save the results of the XPath query. If left empty the results will not be saved.

Returns:

The result of the XPath query or the error string !!ERROR!!

6) PCQB ZQuery(XPath ; Object ; optSearchCriteria ; optFilePath)

Description:

The function queries QuickBooks for existing QuickBooks records based on the parameters passed to the function.

Parameters:

XPath =	Contains the XPath query to be applied to the desired XML.
optSearchCriteria =	Contains XML fragments that will be applied to the search. For example, when searching for a customer whose name is "Sample Customer", the text required in the search criteria parameter would simply be, "<FullName>Sample Customer</FullName>". This parameter is optional and if omitted all records matching the desired object are queried.
Object =	Any object in QuickBooks that responds to the query request, such as Host, Company, Customer, Invoice, and many more. See the appendix for available objects.
optFilePath =	Optional parameter indicating a file to save the results of the XPath query. If left empty (or if the final semicolon and parameter are omitted entirely) the results will not be saved.

Returns:

On success the result of the XPath query applied to the response or the error string !!ERROR!!

7) PCQB ZValidateXML(XML ; IsPath; optWellFormedOnly)

Description:

This function validates the desired XML against the qbXML schema for the currently opened QuickBooks file.

Parameters:

XML =	A string containing the desired qbXML or a path to the document containing the desired qbXML
IsPath =	A Boolean operator indicating whether or not the XML argument a path to a file. A 'True' value indicates the XML argument is a path to a file.
optWellFormedOnly =	Default value is False. Set it to True if you want to only test if the XML passed to the function is well formed, leave it empty or set to False if you want full validation against the QBXML templates.

Returns:

0 if the desired XML is a valid qbXML document for use with the currently opened file or !!ERROR!!

III. Contact Us

Successful integration of a FileMaker plug-in requires the creation of integration scripts within your FileMaker solution. A working knowledge of FileMaker Pro, especially in the areas of scripting and calculations is necessary. If you need additional support for scripting, customization or setup (excluding registration) after reviewing the videos, documentation, FileMaker demo and sample scripts, then 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!