



Productive  
Computing

Creating Efficiency Through Automation



# Functions Guide

January 26, 2021

# Table of Contents

|                                                                                                                      |           |
|----------------------------------------------------------------------------------------------------------------------|-----------|
| <b>I. Introduction .....</b>                                                                                         | <b>3</b>  |
| <b>II. Function Descriptions.....</b>                                                                                | <b>4</b>  |
| A. Session Functions.....                                                                                            | 4         |
| 1) PCQO_BeginSession .....                                                                                           | 4         |
| 2) PCQO_Authorize( SessionString ) .....                                                                             | 5         |
| 3) PCQO_EndSession .....                                                                                             | 5         |
| 4) PCQO_SSaveSessionInfo.....                                                                                        | 6         |
| 5) PCQO_SLoadSessionInfo( SessionInfoString ) .....                                                                  | 7         |
| 6) PCQO_RefreshSession .....                                                                                         | 8         |
| 7) PCQO_SGetMinorVersion .....                                                                                       | 8         |
| 8) PCQO_SSetMinorVersion( MinorVersion ) .....                                                                       | 9         |
| B. Request Functions.....                                                                                            | 10        |
| 1) PCQO_RqNew( Operation ; EntityType ) .....                                                                        | 10        |
| 2) PCQO_RqAddFieldWithValue( FieldName ; Value ) .....                                                               | 11        |
| 3) PCQO_RsGetFirstFieldValue( FieldName ) .....                                                                      | 11        |
| 4) PCQO_RqAddRelatedRecord( Type ) .....                                                                             | 12        |
| 5) PCQO_RqCloseRelatedRecord .....                                                                                   | 12        |
| 6) PCQO_RqExecute .....                                                                                              | 13        |
| 7) PCQO_RqUseJSON( JSON ) .....                                                                                      | 13        |
| 8) PCQO_RqUseXML( XML ) .....                                                                                        | 14        |
| 9) PCQO_RqAddRequestToBatch .....                                                                                    | 14        |
| C. Response Functions.....                                                                                           | 15        |
| 1) PCQO_RsOpenFirstRecord.....                                                                                       | 15        |
| 2) PCQO_RsOpenNextRecord .....                                                                                       | 15        |
| 3) PCQO_RsGetFirstFieldValue( FieldName ) .....                                                                      | 16        |
| 4) PCQO_RsOpenFirstRelatedRecord( Type ) .....                                                                       | 16        |
| 5) PCQO_RsOpenNextRelatedRecord .....                                                                                | 17        |
| 6) PCQO_RsCloseRelatedRecord .....                                                                                   | 17        |
| 7) PCQO_RsUseJSON ( JSON ) .....                                                                                     | 17        |
| 8) PCQO_RsUseXML ( XML ) .....                                                                                       | 18        |
| 9) PCQO_RsOpenFirstBatchRecord .....                                                                                 | 18        |
| 10) PCQO_RsOpenNextBatchRecord .....                                                                                 | 19        |
| D. Other Functions .....                                                                                             | 20        |
| 1) PCQO_SGetStatus .....                                                                                             | 20        |
| 2) PCQO_SGetJSON( Type ) .....                                                                                       | 20        |
| 3) PCQO_SCountEntities( EntityType ) .....                                                                           | 21        |
| 4) PCQO_SClearBatchQueues( Type ) .....                                                                              | 21        |
| 5) PCQO_SUploadAttachableFile( ForEntityType ; EntityID ; FilePathOrContainer ; IsContainer ;<br>optFileName ) ..... | 22        |
| 6) PCQO_SDownloadAttachableFile( AttachableID ; optFilePath ; bSaveToContainer ) .....                               | 23        |
| 7) PCQO_SSendInvoice( InvoiceID ; ToEmail ) .....                                                                    | 24        |
| E. Registration Functions .....                                                                                      | 25        |
| 1) PCQO_GetOperatingMode.....                                                                                        | 25        |
| 2) PCQO_Register( ServerName ; ServerPort ; ServerPage ; LicenseID ) .....                                           | 25        |
| 3) PCQO_Version( Type ) .....                                                                                        | 26        |
| F. Additional Fields .....                                                                                           | 27        |
| G. Supported Attachable File Types .....                                                                             | 28        |
| <b>III. Contact Us .....</b>                                                                                         | <b>29</b> |

# I. Introduction

**Description:** The FM Books Connector plug-in is a powerful tool used to move data between FileMaker® Pro and Intuit® QuickBooks Online. 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) Read the Developer's Guide
- 2) Read the Functions Guide
- 3) Familiarize yourself with basic accounting practices and QuickBooks
- 4) Download a demo: <http://www.productivecomputing.com/fm-books-connector-online/>
- 5) Watch video tutorials: <http://www.productivecomputing.com/video>
- 6) Use the Intuit's Accounting API Reference guide: <https://developer.intuit.com/docs/api/accounting>
- 7) Use the Intuit's Migration guide  
[https://developer.intuit.com/docs/0025\\_quickbooksapi/0058\\_faq/qbo\\_xml\\_sdk\\_\(soap\)\\_to\\_v3\\_rest\\_migration](https://developer.intuit.com/docs/0025_quickbooksapi/0058_faq/qbo_xml_sdk_(soap)_to_v3_rest_migration)

## II. Function Descriptions

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

### A. Session Functions

#### 1) PCQO\_BeginSession

**Description:**

This function begins a session for working with a QuickBooks Online company.

**Dependencies:**

None

**Parameters:**

None

**Return Values:**

0 for success, or "!!ERROR!!"

**Notes & Examples:**

This function will set the plug-in up for authentication with a QuickBooks Online company. Upon a successful call, it will bring up the default internet browser navigated to Intuit's authentication portal.

Starting in version 2.0.0.0, the Mac version of the FM Books Connector Online plug-in utilizes an in-app authentication window instead of the default internet browser. This in-app window combines the functionality of PCQO\_BeginSession and PCQO\_Authorize in order to completely authorize the session, from start to finish.

The Server version of the plug-in cannot use the PCQO\_BeginSession function. This function will return an error if called from the server Scripting Engine or Web Publishing Engine. To acquire a session, please use a client build of the FM Books Connector Online and pass the session string from PCQO\_SSaveSessionInfo to the server plugin's PCQO\_SLoadSessionInfo call. For more information, please reference the "Server Session Management" section in the Developer's Guide.

## 2) PCQO\_Authorize( SessionString )

### Description:

This function authorizes the plug-in with a provided session string generated by Intuit when authenticating to a QuickBooks Online company.

### Dependencies:

PCQO\_BeginSession must have been called, and the user authorized the FM Books Connector Online to communicate with their company.

### Parameters:

| Parameter Name | Purpose                                                                        | Values | Default Value |
|----------------|--------------------------------------------------------------------------------|--------|---------------|
| SessionString  | An encoded session string acquired from a successful call to PCQO_BeginSession |        |               |

### Return Values:

0 for success, or "!!ERROR!!"

### Notes and Examples:

This function is only necessary when using the Windows version of the plug-in. The Mac version requires only PCQO\_BeginSession to authorize a connection. The Server version cannot begin, authorize, or end its own session.

## 3) PCQO\_EndSession

### Description:

This function ends the currently begun and/or authorized session with QuickBooks Online.

### Dependencies:

None

### Parameters:

None

### Return Values:

0 for success, or "!!ERROR!!"

### Notes & Examples:

If no session is currently active, will return 0.

The Server version of the plug-in cannot begin, authorize, or end its own session.

## 4) PCQO\_SSaveSessionInfo

### **Purpose:**

Saves the currently active and authenticated session with QuickBooks Online to an encoded session information string. This session information can be shared with another instance of the FM Books Connector plug-in through the FileMaker solution, or shared between FileMaker sessions.

### **Dependencies:**

An active and authenticated session with QuickBooks Online.

### **Parameters:**

None

### **Return Values:**

An encoded session string, or "!!ERROR!!"

### **Notes & Examples:**

Sample script:

```
# Assume that the session has already begun and the user has been authenticated...
#
# Get the active session information
Set Variable [ $sessionInfo, Value: PCQO_SSaveSessionInfo ]
If ( $sessionInfo = "!!ERROR!!" )
# Handle error...
Else
# Set the session field with the active QBO session information
Set Field [ Preferences::QBOSessionInfo, Value: $sessionInfo ]
End If
#
```

When using the server-side plug-in, session information must be generated by a client-side version of the FM Books Connector Online plug-in and passed through FileMaker to the server plug-in, via a field or script parameter. For more information and for examples, see the "Server Session Management" section in the Developer's Guide.

## 5) PCQO\_SLoadSessionInfo( SessionInfoString )

### Purpose:

This function will take a session information string generated by PCQO\_SSaveSessionInfo and load it into the plug-in. This will make the loaded session information into the plug-in's active session, allowing it to communicate using a previously-begun and authorized session with QuickBooks Online. Loading a new session information string will overwrite the existing session. This is useful when sharing the session with multiple users of the FM Books Connector Online plug-in, or when managing multiple connections to multiple companies with one plug-in.

### Dependencies:

None

### Parameters:

| Parameter Name    | Purpose                                                                                        | Values | Default Value |
|-------------------|------------------------------------------------------------------------------------------------|--------|---------------|
| SessionInfoString | An encoded string that contains the session information, as generated by PCQO_SSaveSessionInfo |        |               |

### Return Values:

0 for success, or "!!ERROR!!"

### Notes & Examples:

Sample script:

```
# Assume that the fields and tables referenced below are the same as the previous sample script
#
# Load the session information from the Preferences table into the plug-in
Set Variable [ $result, Value: PCQO_SLoadSessionInfo( Preferences::QBOSessionInfo ) ]
If ( $result = "!!ERROR!!" )
# Handle error...
Else
# Report success, set connection state flags, etc.
End If
#
```

Note that if the PCQO\_EndSession function has been called by any plug-in using the same session as the one loaded by this function, it will invalidate ALL instances of that session, and a fresh call to PCQO\_BeginSession will need to be made.

## 6) PCQO\_RefreshSession

### **Purpose:**

This function will attempt to refresh the current session, thus obtaining a new session token and extending the life of the connection by six months. It will only be successful if called within 30 days of the expiration of the current token.

### **Dependencies:**

Must have a session currently active.

### **Return Values:**

0 for success, or "!!ERROR!!"

### **Notes & Examples:**

The lifetime of an authenticated session with Intuit QuickBooks Online is six months, or 180 days. During this time period, information can be transferred freely between QuickBooks Online and FileMaker via the FM Books Connector Online plug-in. However, after this period, a new session will be required in order to continue communication.

Instead of requiring a complete session reconnection (i.e. beginning a new session, authorizing the connection, and restoring the session information), an authenticated session can be refreshed using the PCQO\_RefreshSession call. This call is only successful and valid if within 30 days of the expiration date for the current session. If the call is successful, the current token is disabled and invalidated, and a new token is loaded into memory. After calling PCQO\_RefreshSession successfully, make sure to store the new session information back into FileMaker by calling PCQO\_SSaveSessionInfo.

This function can also be called by the server version of the plug-in.

## 7) PCQO\_SGetMinorVersion

### **Description:**

This function returns the current minor version that the plug-in is using. The minor version is a special version value that QuickBooks Online uses to determine what kind of business logic to apply. Older minor versions may use different logic or features than newer ones, and may also include bug fixes that are specifically for the Intuit QuickBooks Online service as opposed to the FM Books Connector Online plug-in. By default, the plug-in is built to always use the latest minor version from QuickBooks Online as of the plug-in release date.

### **Dependencies:**

None.

### **Parameters:**

None.

### **Return Values:**

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



## 8) PCQO\_SSetMinorVersion( MinorVersion )

### Description:

This function sets the current minor version value used by the FM Books Connector plug-in to the provided numerical value. The minor version determines what business logic and enhancements from Quickbooks Online will be used when processing requests and responses.

### Dependencies:

None.

### Parameters:

| Parameter Name | Purpose                                           | Values                                 |
|----------------|---------------------------------------------------|----------------------------------------|
| MinorVersion   | A numerical value representing the minor version. | Any valid QBO transaction entity type. |

### Return Values:

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

## B. Request Functions

### 1) PCQO\_RqNew( Operation ; EntityType )

**Description:**

Begins a new request. The request will be performing an operation based off of the Operation parameter, and affect the type of QuickBooks Online entities designated by EntityType.

**Dependencies:**

Plug-in must be authorized.

**Parameters:**

| Parameter Name | Purpose                                        | Values                                   | Default Value |
|----------------|------------------------------------------------|------------------------------------------|---------------|
| Operation      | The type of "CRUD" operation to perform.       | "Create", "Query", "Update", or "Delete" |               |
| EntityType     | The entity to perform the "CRUD" operation on. | Any valid QuickBooks Online entity.      |               |

**Return Values:**

0 for success, or "!!ERROR!!"

**Notes & Examples:**

Explanation of the "CRUD" Operations:

- Create: Creates a new entity.
- Query: Sets up a query for one or more entities.
- Update: Updates an existing entity.
- Delete: Deletes an existing entity.

**Example 1:**

In the FM Books Connector, when creating a new customer, the RqNew function would be structured as:

```
PCQB_RqNew( "CustomerAdd" )
```

In the FM Books Connector Online, the matching RqNew function call would be structured as:

```
PCQO_RqNew( "Create" ; "Customer" )
```

**Example 2:**

FM Books Connector:

```
PCQB_RqNew( "InvoiceMod" )
```

FM Books Connector Online:

```
PCQO_RqNew( "Update" ; "Invoice" )
```

## 2) PCQO\_RqAddFieldWithValue( FieldName ; Value )

### Description:

This function is used to add a field and a value to the current request.

### Dependencies:

Plug-in must be authorized.

### Parameters:

| Parameter Name | Purpose                                      | Values | Default Value |
|----------------|----------------------------------------------|--------|---------------|
| FieldName      | The name of the field in the request to set. |        |               |
| Value          | The value to set the field to.               |        |               |

### Return Values:

0 for success, or "!!ERROR!!"

## 3) PCQO\_RsGetFirstFieldValue( FieldName )

### Description:

Returns the value stored for the field with the name specified by \$field.

### Dependencies:

Plug-in must be authorized. Must have an active record in memory.

### Parameters:

| Parameter Name | Purpose                                      | Values                                              | Default Value |
|----------------|----------------------------------------------|-----------------------------------------------------|---------------|
| FieldName      | The name of the field to get the value from. | Any valid field name for the requested entity type. |               |

### Return Values:

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

### Notes & Examples:

As of version 1.0.0.0, all values are returned as string values.

#### 4) PCQO\_RqAddRelatedRecord( Type )

**Description:**

This function adds a related record to the current request. The related record becomes active until the PCQO\_RqCloseRelatedRecord function is called. Calls to PCQO\_RqAddRelatedRecord and PCQO\_RqAddFieldWithValue will add records and fields to the currently active record until PCQO\_RqCloseRelatedRecord is called.

**Dependencies:**

Plug-in must be authorized.

**Parameters:**

| Parameter Name | Purpose                            | Values                                                                 | Default Value |
|----------------|------------------------------------|------------------------------------------------------------------------|---------------|
| Type           | The type of related record to add. | Any valid related record type according to the primary request entity. |               |

**Return Values:**

0 for success, or "!!ERROR!!"

#### 5) PCQO\_RqCloseRelatedRecord

**Description:**

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

**Dependencies:**

Plug-in must be authorized. A related record must be currently active.

**Parameters:**

None

**Return Values:**

0 for success, or "!!ERROR!!"

## 6) PCQO\_RqExecute

### Description:

This function posts the request to QuickBooks Online. If the request is processed by QuickBooks, a set of records will be created in response and held in memory for parsing. If the request is not valid or some other error has occurred, then an error is returned. When at least one batch request has been added to the batch queue, this function will execute the request as a "Batch" request, instead of the standard request. Any request currently in memory that has not been added to the queue will be added.

### Dependencies:

Plug-in must be authorized. A request must be created and currently active.

**Parameters:** None

### Return Values:

0 for success, or "!!ERROR!!"

## 7) PCQO\_RqUseJSON( JSON )

### Description:

This function attempts to load a provided JSON-formatted string into the plug-in as the QBO Request. If PCQO\_RqExecute is then called, the loaded request will be submitted to QuickBooks Online. The Type parameter also accepts the values "BatchRequest" and "BatchResponse", which will return the JSON values of the Batch requests or responses, as appropriate.

### Dependencies:

None

### Parameters:

| Parameter Name | Purpose                                               | Values | Default Value |
|----------------|-------------------------------------------------------|--------|---------------|
| JSON           | JSON - formatted document to be loaded as the request |        |               |

### Return Values:

0 for success, or "!!ERROR!!"

## 8) PCQO\_RqUseXML( XML )

### Description:

This function attempts to convert a provided XML-formatted string to a JSON document, and load it as the Request. If PCQO\_RqExecute is then called, the loaded request will be submitted to QuickBooks Online.

### Dependencies:

None

### Parameters:

| Parameter Name | Purpose                                               | Values | Default Value |
|----------------|-------------------------------------------------------|--------|---------------|
| XML            | XML - formatted document to be loaded as the response |        |               |

### Return Values:

0 for success, or "!!ERROR!!"

## 9) PCQO\_RqAddRequestToBatch

### Description:

This function adds the current request object to the batch queue. The batch queue is an internal buffer that uses the Batch supporting entity for QuickBooks Online's Accounting API to execute a batch of requests. These requests can be a mix of Create, Query, Update or Delete requests. When executing a batch request, the internal batch response queue will be populated with the results, whether success or failure, for each batch in the order they were added; the first batch request will be the first batch response, and so on.

### Dependencies:

Plug-in must be authorized.

### Parameters:

None.

### Return Values:

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

### Notes & Examples:

The Batch requests follow the same logic as the Batch supporting entity in the QuickBooks Online Accounting API. You can find details about the appearance and functionality of a Batch object at the following link:

<https://developer.intuit.com/docs/api/accounting/batch>

## C. Response Functions

### 1) PCQO\_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 PCQO\_RsOpenNextRecord to iterate through the rest of the records in this response.

**Dependencies:**

Plug-in must be authorized. A call to PCQO\_RqExecute must have succeeded.

**Parameters:**

None

**Return Values:**

0 for success, "END" if there are no records in the response, or "!!ERROR!!"

### 2) PCQO\_RsOpenNextRecord

**Description:**

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

**Dependencies:**

Plug-in must be authorized. Must have already called PCQO\_RsOpenFirstRecord.

**Parameters:** none**Return Values:**

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

### 3) PCQO\_RsGetFirstFieldValue( FieldName )

**Description:**

Returns the value stored for the field with the name specified by \$field.

**Dependencies:**

Plug-in must be authorized. Must have an active record in memory.

**Parameters:**

| Parameter Name | Purpose                                      | Values                                              | Default Value |
|----------------|----------------------------------------------|-----------------------------------------------------|---------------|
| FieldName      | The name of the field to get the value from. | Any valid field name for the requested entity type. |               |

**Return Values:**

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

**Notes & Examples:**

As of version 1.0.0.0, all values are returned as string values.

### 4) PCQO\_RsOpenFirstRelatedRecord( Type )

**Description:**

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

**Dependencies:**

Plug-in must be authorized. Must have an active record in memory.

**Parameters:**

| Parameter Name | Purpose                             | Values                                                      | Default Value |
|----------------|-------------------------------------|-------------------------------------------------------------|---------------|
| Type           | The type of related record to open. | Any valid related entity type for the active parent record. |               |

**Return Values:**

0 for success or "!!ERROR!!"



## 5) PCQO\_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.

### Dependencies:

Plug-in must be authorized. Must have another related record as the active record.

**Parameters:** none

### Return Values:

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

## 6) PCQO\_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.

### Dependencies:

Plug-in must be authorized. Must have a related record to the parent active in memory.

### Parameters:

None

### Return Values:

0 for success, or "!!ERROR!!"

## 7) PCQO\_RsUseJSON ( JSON)

### Description:

This function attempts to load a provided JSON-formatted string into the plug-in as the Response. Any subsequent calls to Response functions (e.g. PCQO\_RsOpenFirstRelatedRecord, PCQO\_RsGetFirstFieldValue, etc.) will act on the loaded JSON string.

### Dependencies:

| Parameter Name | Purpose                                                | Values | Default Value |
|----------------|--------------------------------------------------------|--------|---------------|
| JSON           | JSON - formatted document to be loaded as the response |        |               |

None

### Parameters:

### Return Values:

0 for success, or "!!ERROR!!"

## 8) PCQO\_RsUseXML ( XML )

### Description:

This function attempts to convert a provided XML-formatted string to a JSON document, and load it as the Response. Any subsequent calls to Response functions (e.g. PCQO\_RsOpenFirstRelatedRecord, PCQO\_RsGetFirstFieldValue, etc.) will act on the loaded JSON string.

### Dependencies:

None

### Parameters:

| Parameter Name | Purpose                                               | Values | Default Value |
|----------------|-------------------------------------------------------|--------|---------------|
| XML            | XML - formatted document to be loaded as the response |        |               |

### Return Values:

0 for success, or "!!ERROR!!"

## 9) PCQO\_RsOpenFirstBatchRecord

### Description:

This function opens the first batch response record retrieved by the PCQO\_RqExecute function after executing a batch request. The first response record always correlates to the first batch request record. After opening the first batch response record, the standard workflow of processing response results should be followed: open the first record, retrieve field data, open the next record, etc., until complete.

### Dependencies:

Plug-in must be authorized. Must have executed a Batch request.

### Parameters:

None.

### Return Values:

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

### Notes & Examples:

The Batch requests follow the same logic as the Batch supporting entity in the QuickBooks Online Accounting API. You can find details about the appearance and functionality of a Batch object at the following link: <https://developer.intuit.com/docs/api/accounting/batch>

For an example of how batch scripting looks, refer to the Developer's Guide.

## 10) PCQO\_RsOpenNextBatchRecord

### Description:

This function opens the next batch response record retrieved by the PCQO\_RqExecute function after executing a batch request. Each batch response record correlates to the same position of the batch request record (e.g. the 3rd batch response record is for the 3rd batch request record, and so on). After opening the first batch response record, the standard workflow of processing response results should be followed: open the first record, retrieve field data, open the next record, etc., until complete. At the end of the batch response queue, this function returns "END".

### Dependencies:

Plug-in must be authorized. Must have executed a Batch request. Must have previously called PCQO\_RsOpenFirstBatchRecord.

### Parameters:

None.

### Return Values:

0 for success, "END" if at the end of the batch response queue, otherwise "!!ERROR!!"

### Notes & Examples:

The Batch requests follow the same logic as the Batch supporting entity in the QuickBooks Online Accounting API. You can find details about the appearance and functionality of a Batch object at the following link: <https://developer.intuit.com/docs/api/accounting/batch>

For an example of how batch scripting looks, refer to the Developer's Guide.

## D. Other Functions

These functions are primarily for debugging purposes.

### 1) PCQO\_SGetStatus

**Description:**

This function is used to retrieve a verbose explanation of errors. Immediately after an error is returned, please call PCQO\_SGetStatus to obtain a full description of the error. The status is used to identify errors in the request or the processing of request. The text returned by this function helps the user to troubleshoot script or logic failures.

**Dependencies:**

None

**Parameters:**

None

**Return Values:**

A string description of the last error to occur.

### 2) PCQO\_SGetJSON( Type )

**Description:**

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

**Dependencies:**

Plug-in must be authorized and have prepared a request, or received a response.

**Parameters:**

| Parameter Name | Purpose                                  | Values                                                                                                            | Default Value |
|----------------|------------------------------------------|-------------------------------------------------------------------------------------------------------------------|---------------|
| Type           | The type of XML to pull from the plug-in | "Request" to get the XML representation of the request, "Response" to get the XML representation of the response. |               |

**Return Values:**

The XML text, or "!!ERROR!!"

**Notes & Examples:**

This function is intended to aid in determining issues with malformed or incomplete requests and responses. Though the plug-in can provide the XML of a request or response, it cannot accept XML for using as a request as of version 1.0.0.0.

### 3) PCQO\_SCountEntities( EntityType )

**Description:**

This function returns a count of all entities of a given EntityType.

**Dependencies:**

Plug-in must be authorized.

**Parameters:**

| Parameter Name | Purpose                                | Values                              | Default Value |
|----------------|----------------------------------------|-------------------------------------|---------------|
| EntityType     | The type of entity to get the count of | Any valid QuickBooks Online entity. |               |

**Return Values:**

Count of entities, or "!!ERROR!!"

**Notes & Examples:**

This will return ALL entities within the QuickBooks Online company file for the matching entity type. If you wish to limit the entities to a certain criteria, create a custom request using the standard PCQO\_RqNew and PCQO\_RqAddFieldWithValue functions for a "Query" operation.

This function has the same result as the following function call combination:

```
PCQO_RqNew( "Query" ; $entityType )
PCQO_RqAddFieldWithValue( "QueryString" ; "SELECT * FROM "" & $entityType& "" )
PCQO_RqExecute
```

### 4) PCQO\_SClearBatchQueues( Type )

**Description:**

This function clears the batch queues specified by "Type". This is important to call after completing any batch handling workflow, as it ensures that the plug-in will not retain batch requests from a previous batch when working with a new workflow.

**Dependencies:**

Plug-in must be authorized.

**Parameters:**

| Parameter Name | Purpose                       | Values                             | Default Value |
|----------------|-------------------------------|------------------------------------|---------------|
| Type           | Type of batch queue to clear. | "Requests", "Responses", or "Both" |               |

**Return Values:**

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

**Notes & Examples:**

This function should always be called at the end of any batch scripting process. The parameter value of "Requests" will only clear the batch request queue. The parameter value of "Responses" will only clear the batch response queue. The parameter value of "Both" will clear both queues of any batch records.

For an example of how batch scripting looks, refer to the Developer's Guide.

### 5) PCQO\_SUploadAttachableFile( ForEntityType ; EntityID ; FilePathOrContainer ; IsContainer ; optFileName )

**Description:**

This function uploads a file from the provided file path or container field to QuickBooks Online, linking it as an Attachable record for the specified entity. Only certain file types are supported; for a list, please refer to the "Attachable File Types" table at the end of this document.

**Dependencies:**

Plug-in must be authorized.

**Parameters:**

| Parameter Name      | Purpose                                                                                                         | Values                                 |
|---------------------|-----------------------------------------------------------------------------------------------------------------|----------------------------------------|
| ForEntityType       | Type of entity to link the attachment to (e.g. "Invoice", "Bill", etc.)                                         | Any valid QBO transaction entity type. |
| EntityID            | QuickBooks Online ID of the entity                                                                              |                                        |
| FilePathOrContainer | A system-formatted file path to the file, or a container field holding embedded, uncompressed binary file data. |                                        |
| IsContainer         | Boolean flag for whether FilePathOrContainer is a container field reference.                                    | True or False                          |
| optFileName         | File name to assign to the file. Required if using a container field.                                           |                                        |

**Return Values:**

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

**Notes & Examples:**

When uploading files from a container field, the file must be stored as an embedded container object that is uncompressed.

The file name and type must match a valid file type from the "Attachable File Types" table below.

## 6) PCQO\_SDownloadAttachableFile( AttachableID ; optFilePath ; bSaveToContainer )

### Description:

This function downloads an Attachable file record from QuickBooks Online, saving it either to a file path or to a container field. If bSaveToContainer is "True", the results of the function will be binary container data on a successful download; otherwise, on success, the attachable file will be located at the path specified by optFilePath.

### Dependencies:

Plug-in must be authorized.

### Parameters:

| Parameter Name   | Purpose                                                                                        | Values        |
|------------------|------------------------------------------------------------------------------------------------|---------------|
| AttachableID     | QuickBooks Online ID of the attachable file record.                                            |               |
| optFilePath      | System-formatted file path for the attachable record. Required if bSaveToContainer is "False". |               |
| bSavetoContainer | Boolean flag indicating whether or not to save the attachable file to a container field.       | True or False |

### Return Values:

0 or binary container data for success, otherwise "!!ERROR!!"

### Notes & Examples:

Container data returned by the plug-in is uncompressed binary file data.

The ID of the attachable record can be retrieved by a standard "Query" request for the "Attachable" entity type (see "Entity Query Support" for examples on creating a Query request, in the Developer's Guide).

The file path must be formatted as per the system specifics. Example valid file paths:

Windows: C:\Users\Test Testerson\Desktop\Invoice\_111.pdf

Mac: Volumes/MacOS/Users/testtesterson/Desktop/Invoice\_111.pdf

## 7) PCQO\_SSendInvoice( InvoiceID ; ToEmail )

### Description:

This function will send an invoice specified by the InvoiceID parameter to an individual's email address specified by the ToEmail parameter. The individual will receive an email from QuickBooks Online containing a PDF document of the invoice using the default Invoice template as per the company settings.

### Dependencies:

Plug-in must be authorized.

### Parameters:

| Parameter Name | Purpose                                                          |
|----------------|------------------------------------------------------------------|
| InvoiceID      | Transaction ID of the invoice to send                            |
| ToEmail        | Email address of the recipient to which the Invoice will be sent |

### Return Values:

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

### Notes:

This is considered the "Short" method. This function performs the actions of making a new "Send Invoice" request, populating the invoice ID and email address, and executing the request with QuickBooks Online. For more details, please review the Developer's Guide.



## E. Registration Functions

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

### 1) PCQO\_GetOperatingMode

**Description:**

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

**Dependencies:**

None

**Parameters:**

None

**Return Values:**

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

### 2) PCQO\_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.

**Dependencies:**

None

**Parameters:**

| Parameter Name | Purpose                                                                                        | Values                                     | Default Values |
|----------------|------------------------------------------------------------------------------------------------|--------------------------------------------|----------------|
| ServerName     | Name of the Productive Computing licensing server                                              | "licensing.productivecomputing.com"        |                |
| ServerPort     | System-formatted file path for the attachable record. Required if bSaveToContainer is "False". | "80"                                       |                |
| ServerPage     | PCI Registration page                                                                          | "/PCIReg/pcireg.php"                       |                |
| LicenseID      | License ID to register against                                                                 | Your purchased license ID, or "DEMO-FMBO1" |                |

**Return Values:**

0 for success, or an error string.

### 3) PCQO\_Version( Type )

**Description:**

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

**Dependencies:**

None

**Parameters:**

| Parameter Name | Purpose                               | Values          | Default Values |
|----------------|---------------------------------------|-----------------|----------------|
| Type           | Type of version string to be returned | "Long", "Short" | "Short"        |

**Return Values:**

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

**Notes & Examples:**

The short version string includes only the version number of the plug-in. For example: "1.0.0.0".

The long version string includes the plug-in name and version number. For example: "FM Books Connector Online 1.0.0.0".

The default version string returned is short.

## F. Additional Fields

The following fields are not defined in the ESR but are acceptable fields for the PCQO\_RqAddFieldWithValue function for query operations.

| Module | Method | Field Name    | Acceptable Values/Notes                                                                                                                                                                 |
|--------|--------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Query  | Set    | MaxResults    | Any integer value greater than 0. Default value is the number of the corresponding entity in QuickBooks Online, or 500 entities, whichever is less.                                     |
| Query  | Set    | StartPosition | Any integer value greater than 0. Default value is 1.                                                                                                                                   |
| Query  | Set    | QueryString   | Any SQL Select statement that follows Intuit's SQL Syntax rules.<br><br>The SQL Select syntax can be found at the following link:<br><a href="#">Intuit SQL Select Statement Syntax</a> |
| Query  | Get    | CountResult   | Returns the number of entities counted by a "count(*)" query operation                                                                                                                  |
| Update | Set    | SparseUpdate  | Determines whether the update will be a "Complete" or "Sparse" update. True or False. Default value is true.                                                                            |

## G. Supported Attachable File Types

The following table is a list of acceptable file extensions when uploading Attachable file documents to QuickBooks Online. Files with extensions not on this list will not be uploaded to QuickBooks Online, and will report an error.

| File Extension |
|----------------|
| .ai            |
| .csv           |
| .doc           |
| .docx          |
| .eps           |
| .gif           |
| .jpg           |
| .jpeg          |
| .ods           |
| .pdf           |
| .png           |
| .rtf           |
| .tif           |
| .txt           |
| .xls           |
| .xlsx          |
| .xml           |

### 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](mailto:support@productivecomputing.com)

Forum: [www.productivecomputing.com/forum](http://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](http://www.productivecomputing.com/rfq). We are ready to assist and look forward to hearing from you!