



Productive
Computing



Developer's Guide

Revised August 5, 2013

Table of Contents

I. INTRODUCTION.....	3
II. INTEGRATION STEPS	4
1) Installing the Plug-in.....	4
2) Registering the Plug-in.....	4
3) Talking to Outlook	6
A. Get Folders from Outlook	6
B. Communication with IMAP Email Accounts	6
C. Communication with POP Email Accounts	6
D. Communication with Office 365 Accounts	7
F. Create a New Outlook Folder	7
G. Manually Adding Folders in Outlook	8
H. Create/Open a Record	8
I. Manipulate the Fields	9
J. Save Record or Send Email.....	9
4) Moving or Deleting a Record	10
A. Move a Record:.....	10
B. Delete a Record:	10
5) Error Handling	11
III. TIPS.....	12
IV. CONTACT US	13

I. Introduction

Description:

The Outlook Manipulator for Mac plug-in is a powerful tool used to exchange data between FileMaker® and Microsoft® Outlook for Mac OSX (32-bit). With this plug-in FileMaker users are able to bidirectionally exchange data between FileMaker and Outlook. Data exchange operations are accomplished using FileMaker function calls from within FileMaker calculations. These calculations are generally determined from within FileMaker "SetField" or "If" script steps. This document describes the basic integration steps, features and error handling. Please see the accompanying Functions Guide for a list of available plug-in functions and Outlook fields.

Product Version History:

http://www.productivecomputing.com/outlook-integration-mac/version_history

Intended Audience:

FileMaker developers or persons who have knowledge of FileMaker 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) Watch our video tutorials: <http://productivecomputing.com/video/>
- 4) Review our FileMaker Pro demo file
- 5) Familiarize yourself with Microsoft Outlook

II. Integration Steps

Accessing and using the plug-in involve the following steps.

1) Installing the Plug-in

The first step is to install the plug-in into FileMaker Pro.

FileMaker 12:

- 1) Open the FileMaker demo file available in the plug-in bundle (www.productivecomputing.com).
- 2) Select the "Install" button.

For FileMaker 11 or earlier, follow the steps below to manually install the plug-in into the FileMaker Extensions folder.

- 1) Quit FileMaker Pro completely.
- 2) Locate the plug-in in your download which will be located in a folder called "Plug-in." On Mac the plug-in will have a ".fmplugin" extension.
- 3) Copy the actual plug-in and paste it to the Extensions folder which is inside the FileMaker program folder.
 - On Mac this is normally located here: Volume/Applications/FileMaker X/Extensions (Volume is the name of the mounted volume)
- 4) Start FileMaker Pro. Confirm that the plug-in has been successfully installed by navigating to "Preferences" in FileMaker, then select the "Plug-ins" tab. There you should see the plug-in listed with a corresponding check box. This indicates that you have successfully installed the plug-in.

2) Registering the Plug-in

The next step is to register the plug-in which enables all plug-in functions.

- 1) Confirm that you have access to the internet and open our FileMaker demo file, which can be found in the "FileMaker Demo File" folder in your original download.
- 2) If you are registering the plug-in in Demo mode, then simply click the "Register" button and do not change any of the fields. Your plug-in should now be running in "DEMO" mode. The mode is always noted on the Setup tab of the FileMaker demo.
- 3) If you are registering a licensed copy, then simply enter your license number in the "LicenseID" field and select the "Register" button. Ensure you have removed the Demo License ID and enter your registration information exactly as it appears in your confirmation email. Your plug-in should now be running in "LIVE" mode. The mode is always noted on the Setup tab of the FileMaker demo, or by calling the PCEM_GetOperatingMode function.

Congratulations! You have now successfully installed and registered the plug-in!

Why do I need to Register?

In an effort to reduce software piracy, Productive Computing, Inc. has implemented a registration process for all plug-ins. The registration process sends information over the internet to a server managed by Productive Computing, Inc. The server uses this information to confirm that there is a valid license available and identifies the machine. If there is a license available, then the plug-in receives an acknowledgment from the server and installs a certificate on the machine. This certificate never expires. If the certificate is ever moved, modified or deleted, then the client will be required to register again. On Mac this certificate is in the form of a ".plist" file.

How do I hard code the registration process?

You can hard code the registration process inside a simple "Plug-in Checker" script. The "Plug-in Checker" script should be called at the beginning of any script using a plug-in function and uses the PCEM_Register, PCEM_GetOperatingMode and PCEM_Version functions. This eliminates the need to manually register each machine and ensures that the plug-in is installed and properly registered. Below are the basic steps to create a "Plug-in Checker" script.

```
If [ PCEM_Version( "short" ) = "" or PCEM_Version( "short" ) = "?" ]  
Show Custom Dialog [ Title: "Warning"; Message: "Plug-in not installed."; Buttons: "OK" ]  
If [ PCEM_GetOperatingMode ≠ "LIVE" ]  
Set Field [Main::gRegResult; PCEM_Register( "licensing.productivecomputing.com" ; "80" ; "/PCIReg/pcireg.php" ;  
"your license ID" )  
If [ Main::gRegResult ≠ 0 ]  
Show Custom Dialog [ Title: "Registration Error"; Message: "Plug-in Registration Failed"; Buttons: "OK" ]
```

3) Talking to Outlook

Productive Computing, Inc. will only support those that upgrade to the latest patch of Outlook for Mac 2011. Please ensure that you are running the latest version of Microsoft Outlook on Macintosh.

Talking to Outlook typically requires that you follow these steps.

A. Get Folders from Outlook

First the plug-in must specify what "root" folder or mailbox will be used. This is accomplished by selecting the "Get Folders from Outlook" button.

Specify the root folder to work with by calling `PCEM_OpenRootFolder(RootFolderName)` and passing the name of the desired root folder to be opened. Once the Outlook root folder has been selected then any subfolder of the root folder can be accessed such as Mail folders, Calendar folders, Contact Folders, etc.

Please note that since Outlook refers to user mailboxes and modules such as Mail, Contacts, Calendar, Tasks, and Notes modules as folders, then so does our documentation and plug-in.

B. Communication with IMAP Email Accounts

This plug-in is able to retrieve email from an IMAP server when Outlook for Mac is synced to that account. Accounts like G-mail, Hotmail, and Yahoo offer IMAP connectivity and can be synced to Outlook on a Mac. Due to the nature of network requests and handling multiple processes, performance across all use of the plug-in can be affected by connecting to an external account. Plug-in performance can be improved when connected to an IMAP account by turning off the IDLE service. This service is a means of instant synchronization between Outlook and the IMAP server. Because it requires constant attention from Outlook, communication between Outlook and the plug-in can suffer when this feature is not turned off. Please note that when IDLE is turned off, it becomes necessary for the user to press Outlook's "Send/Receive" button to retrieve and send the latest messages. To turn IDLE off in Outlook 2011, go to:

Outlook Preferences > Accounts > Select the IMAP Email Account > Advanced >
Uncheck the box 'Use IMAP IDLE (if the server supports it)'

C. Communication with POP Email Accounts

When Outlook is connected to a POP account, the Outlook Manipulator for Mac plug-in will fail to communicate with Outlook and may cause FileMaker to crash. Although Outlook is capable of connecting to a POP email account, doing so causes the communication channel between FileMaker and Outlook to slow to a crawl. Therefore, we discourage the use of POP with Outlook while employing this plug-in.

D. Communication with Office 365 Accounts

Ensure that the Office 365 account set up in Outlook is listed as the "Default Account." This will need to be done in order to successfully send mail using the Outlook Manipulator for Mac plug-in. To set your Default email account navigate to:

Outlook Preferences > Accounts > Select the appropriate email account > Click the lower gear icon > Set as Default

The words "Default Account" will be above the Default email address.

E. Select an Outlook Folder

Now that we have set the default account in Outlook and specified a "root" folder, then the next step is to navigate to the desired folder. In order to access any records in Outlook the user must first open the desired folder that contains the desired records. This is accomplished by calling `PCEM_OpenFolder(FolderPath)` and specifying the desired folder by passing the FolderPath parameter to the OpenFolder function. When the FolderPath is prefixed with a "|" then the FolderPath is relative to the current root folder. In other words the '|' indicates a subfolder.

For example the following call opens the folder named "NewItems" located in the "Inbox" Folder:

- `PCEM_OpenFolder("Inbox|NewItems")`

Another example shows the following call opens the folder named "Holiday" located in the "Calendar" Folder:

- `PCEM_OpenFolder("Calendar|Holiday")`

It is up to the developer to "know" their location within the Outlook Tree and to know what types of records are contained in the currently opened folder. Outlook has various items types which are: Mail Items, Contact Items, Calendar Items, Note Items, and Task Items. When importing records it is important to pay special attention to the Active Folder. This plug-in will only import records contained in the Active Folder (specified by each tab's drop-down list).

Note that a developer can set the delimiter for the plugin by using the function

`PCEM_SetGlobalDelimiter(Delimiter)`. The default delimiter for the plugin is "|". Changes to the delimiter will stay until the user closes FileMaker completely.

F. Create a New Outlook Folder

Folders can be created by calling the `PCEM_NewFolder(Name ; OptType)` function. Subfolders cannot be created if the parent folders haven't already been added. For example `PCEM_NewFolder("Inbox|SubFolder1|SubFolder2 ; Mail")` will result in "!!ERROR!!" if either Inbox or SubFolder1 do not exist in Outlook. When all parent folders exist in Outlook, this function will result in "0" and the folder will appear immediately in Outlook.

G. Manually Adding Folders in Outlook

The Outlook Manipulator plug-in can communicate with folders created in a mail account as defined in Outlook's Preferences. It does not have the ability to communicate with local folders or folders "on my computer."

H. Create/Open a Record

After navigating to the desired folder you can now create new records to be placed in the folder or access the records within that folder.

Create a New Record:

There is one method for creating a new record which is by calling the `PCEM_NewRecord(OptType)` function. Anytime the `PCEM_NewRecord(OptType)` function is called the plug-in creates a new blank record immediately in Outlook. The record exists in Outlook and data is immediately entered into that record as the `PCEM_SetFieldData` script steps are called.

Open an Existing Record:

There are two methods for accessing existing Outlook records which are directly or indirectly.

Directly:

If the user knows the OutlookID of the desired record, then they can open that record directly by calling the `PCEM_OpenRecord(OutlookID)` function and passing the OutlookID to the function.

Note that this function closes the currently opened record and any changes made to that record have been properly saved.

Indirectly:

The functions `PCEM_GetFirstRecord` and `PCEM_GetNextRecord` offer the user the ability to iterate through the records contained in the currently opened folder. For example:

```
/* Opens the first Record in the Current Folder */
SetField( someField ; PCEM_GetFirstRecord )
Loop
Exit Loop If[ someField = 'End' or someField < 0]
/* Access - Modify the fields */
...
/* Opens the next record */
New Record/Request
SetField( someField ; PCEM_GetNextRecord )
End Loop
...
```

Note that these functions make either the First or Next record the currently opened record available. They also close the currently opened record with all the information saved properly.

I. Manipulate the Fields

Once a record is created or opened then you can access the fields contained within that record. You can either set or retrieve the contents of the available Outlook fields for that record using PCEM_SetFieldData or PCEM_GetFieldData. A list of all available Outlook fields can be found in the "Functions Guide."

The PCEM_SetFieldData(FieldName ; Data) function is used to populate fields in the new or currently opened Outlook record. See the "Functions Guide" for list of all available FieldNames.

The PCEM_GetFieldData(FieldName) function is used to extract data from the currently opened Outlook record. See the "Functions Guide" for list of all available FieldNames.

J. Save Record or Send Email

After creating or modifying records, use the PCEM_SaveRecordAsMsg(StrPath) function to save the record as a ".EML" formatted file to your desktop.

You can send email using the "short method" in a single script step using PCEM_SendMail(To ; Cc ; Bcc ; Subject ; Body ; Attachments ; OptUseHtml). All parameters except Subject, Body and OptUseHTML can contain multiple comma separated values. Please reference PCEM_SendMail(To ; Cc ; Bcc ; Subject ; Body ; Attachments ; OptUseHtml) function in the "Functions Guide" for further details.

Emails can only be sent from the Default Account set in Outlook's account preferences

In the examples below an email will be sent with three attachments using the various methods. Please refer to the Functions Guide and our FileMaker demo file for further details and live examples.

Example 1: Sending an email using the "short method" by hardcoding parameter values:

- PCEM_SendMail("joe@somecompany.com" ; "" ; "" ; "New Release" ; "A new product was just released allowing for please contact us" ; "MacBookProHD/Users/Pricing.pdf|MacBookProHD/Users/Overview.pdf|MacBookProHD/Users/UserName/Desktop/SpecSheet.pdf" ; "N")

Example 2: Sending an email using the "short method" by using field names:

- PCEM_SendMail(Main::gEmail To ; Main::gEmail Cc ; Main::gEmail Bcc ; Main::gEmail Subject ; FullPath1 & "|" & FullPath2 & "|" & FullPath3 ; Main::gUseHTML)

The FullPath1, FullPath2 and FullPath3 represent the field that stores the full file path.

4) Moving or Deleting a Record

A. Move a Record:

After a record has been opened, call the `PCEM_MoveRecord(FolderPath)` to move the record from the current folder to another folder. After calling the `MoveRecord` function, the folder identified by the `FolderPath` becomes the currently opened folder. If you are going to move multiple items, then a call to `OpenRootFolder (RootFolderName)` and/or `OpenFolder (FolderPath)` must be called to return to the previously opened folder.

The following caveats must be recognized:

- **Exchange Accounts:** When you move a record to another folder such as a Public Folder the folder path should be formatted with the root folder first and "|" to indicate any subfolders. For example: `Inbox|folderpath` or `Public Folders|All Public Folders|PCI`
- **IMAP Accounts:** Email messages in IMAP accounts change their unique IDs when being moved to various folders. Due to this, there is a plug-in limitation which allows only one single push of an email message to a different folder.

B. Delete a Record:

In order to delete a record you can use the `PCEM_DeleteCurrentRecord(Permanent)` function. The `DeleteCurrentRecord` function will delete the currently opened record and maintain the index of items. This command should be used with extreme caution as it will remove ALL items permanently and will not save a copy of the items in the "Deleted Items" folder in Outlook. Please reference the "Functions Guide" for additional information about these functions.

5) Error Handling

When something unexpected happens, a plug-in function will return a result of !!ERROR!!. This makes it simple to check for errors. If a plug-in function returns !!ERROR!!, then immediately after call PCEM_GetLastError function for a detailed description of what the exact error was.

We find that most developers run into issues due to a lack of error trapping. Please ensure that you properly trap for errors in your solutions. Here are a few samples of how you can check for errors.

```
Set Variable [ $result = MyPluginFunction( "a" ; "b" ; "c" ) ]
```

```
If [ $result = !!ERROR!! ]
```

```
Show Custom Dialog [ "An error occurred: " & PCEM_GetLastError ]
```

```
End If
```

The PCEM_GetLastError(format) function gives you the option to display the error description or error number. Displaying the error number is more user friendly in international environments, where an English error description may not be desired. If the format parameter is set to "Number" such as PCEM_GetLastError("Number"), then an error number will be returned. If format parameter is empty such as PCEM_GetLastError or PCEM_GetLastError(""), then an English error description will be returned.

Please find a list of return codes and descriptions below for your reference.

Code	Description
0	SUCCESS
-1	Plug-in not registered or session expired
-3	Invalid number of parameters
-10	Registration failed
-1004	No account found
-1005	Invalid record type
-1006	Index out of bounds
-1015	Folder not found
-1018	Root folder not selected
-1023	Item operation called without opening/creating a record
-1028	Unable to find records
-1036	Item type does not match destination folder

III. Tips

- This “Developer’s Guide” is designed to give you a basic understanding of how Outlook and FileMaker “talk” to each other. There are various other functions available by the plug-in depending on your needs. Please reference the “Functions Guide” for a detailed list of all available functions.
- The Outlook Manipulator for Mac plug-in is intended to run on Macintosh OSX Version 10.7 or later. Please see Platform Requirements on <http://www.outlookmanipulatorformac.com> for the latest system requirements.
- Outlook IDs are unique identifiers created by Outlook. IDs are unique to specific folders (Mail, Contacts, Calendar, Tasks, and Notes) and prefixed with type. For example: Mail-117 or Task-213

IV. 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!