



Productive  
Computing



# Functions Guide

Revised June 26, 2013

# Table of Contents

<b>I. INTRODUCTION .....</b>	<b>3</b>
<b>II. FUNCTION DESCRIPTIONS .....</b>	<b>4</b>
1) Registration and Other Functions.....	4
PCEM_Version( Type ) .....	4
PCEM_Register( ServerName ; ServerPort ; ServerPage ; LicenseID ) .....	4
PCEM_GetOperatingMode.....	5
PCEM_GetLastError( Type ) .....	5
PCEM_GetGlobalDelimiter .....	5
PCEM_SetGlobalDelimiter( Delimiter ) .....	6
PCEM_OutlookVersion .....	6
2) Record Specific Functions .....	7
PCEM_GetFirstRecord .....	7
PCEM_GetNextRecord.....	7
PCEM_OpenRecord( OutlookID ) .....	7
PCEM_GetRecordCount .....	8
PCEM_MoveRecord( FolderPath ) .....	8
PCEM_NewRecord( OptType ) .....	8
PCEM_DeleteCurrentRecord( Permanent ) .....	9
PCEM_Display( ShowInForeground ) .....	9
PCEM_SaveRecordAsMsg( StrPath ) .....	9
PCEM_GetFieldData( FieldName ) .....	10
PCEM_SendMail( To ; Cc ; Bcc ; Subject ; Body ; Attachments ; OptUseHtml ) .....	11
3) Attachment Specific Functions .....	11
PCEM_GetAttachmentCount .....	12
PCEM_GetAttachmentName( Index ) .....	12
PCEM_GetAttachmentSize( Index ) .....	13
PCEM_GetAttachmentType( Index ) .....	13
PCEM_SaveAttachment( FilePath ; Index ) .....	14
PCEM_SaveContactImage( Path ) .....	14
4) Folder Specific Functions .....	15
PCEM_GetDefaultRootFolder.....	15
PCEM_GetFirstFolder.....	15
PCEM_GetNextFolder .....	15
PCEM_OpenFolder( FolderPath ) .....	16
PCEM_GetFolderCount .....	16
PCEM_GetFolderType .....	16
PCEM_GetDefaultFolderName( ItemType ) .....	17
PCEM_NewFolder( Name ; OptType ) .....	17
5) Root and Shared Folder Functions .....	18
PCEM_GetFirstRootFolder .....	18
PCEM_GetNextRootFolder.....	18
PCEM_OpenRootFolder( RootFolderName ) .....	18
<b>III. AVAILABLE OUTLOOK FIELDS .....</b>	<b>19</b>
Available Contact Fields .....	19
Available Mail Fields .....	21
Available Events Fields .....	22
Available Task Fields .....	23
Available Note Fields .....	23
<b>IV. CONTACT US .....</b>	<b>24</b>

## 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. These 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 all available plug-in functions and Outlook fields. Please see the accompanying Developer’s Guide for a list of the integration steps, features, and error handling.

### **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 demo file
- 5) Familiarize yourself with Microsoft Outlook

## II. Function Descriptions

This section describes the functions that are available with the Outlook Manipulator for Mac plug-in.

### 1) Registration and Other Functions

#### PCEM\_Version( Type )

##### **Description:**

This function returns the current plug-in version number in either a long or short format.

##### **Parameters:**

Type = 'Short' or 'Long'

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

The long includes the plug-in name and version number. For example: "Outlook Manipulator for Mac 1.0.0.0"

The default version string returned is "Short."

##### **Return Values:**

A version string otherwise an "!!ERROR!!"

#### PCEM\_Register( ServerName ; ServerPort ; ServerPage ; LicenseID )

##### **Description:**

Registers the plug-in with Productive Computing's registration servers, provided the proper parameters and a valid license ID.

##### **Parameters:**

ServerName = "licensing.productivecomputing.com"

ServerPort = "80"

ServerPage = "/PCIReg/pcireg.php"

LicenseID = "your license ID" or "DEMO-OU1"

##### **Return Values:**

0 for success or !!ERROR!! or error text. Anything other than 0 is an error.

##### **Notes & Examples:**

PCEM\_Register( "licensing.productivecomputing.com" ; "80" ; "/PCIReg/pcireg.php" ; "DEMO-OU1" )

## **PCEM\_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

### **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.

## **PCEM\_GetLastError( Type )**

### **Description:**

If another function returns the string "!!ERROR!!", this function can be called immediately after to retrieve the details of the error.

### **Dependencies:**

A previous function call.

### **Parameters:**

Type = Values are either Text or Number. The default value is text.

### **Return Values:**

A detailed error string or error number, otherwise an "!!ERROR!!"

## **PCEM\_GetGlobalDelimiter**

### **Description:**

This plug-in uses a string of characters to delimit things like folder names or other repeating values. The current delimiter string can be retrieved with this function.

### **Dependencies:**

None

### **Parameters:**

None

### **Return Values:**

The current delimiter string, otherwise an "!!ERROR!!"

### **PCEM\_SetGlobalDelimiter( Delimiter )**

**Description:**

This function is used to set the delimiter string described in the GetGlobalDelimiter function.

**Parameters:**

Delimiter = The string that will be used by the plug-in to separate any repeating values.

**Return Values:**

"0" on success, otherwise an "!!ERROR!!"

### **PCEM\_OutlookVersion**

**Description:**

Returns the version number of the user's Outlook application.

**Dependencies:**

None

**Parameters:**

None

**Return Values:**

A version string on success, otherwise an "!!ERROR!!"

## 2) Record Specific Functions

### PCEM\_GetFirstRecord

**Description:**

This function opens the first record in the currently selected folder.

**Dependencies:**

Must have a folder currently open.

**Parameters:**

None

**Return Values:**

The ID of the record, otherwise an "!!ERROR!!"

### PCEM\_GetNextRecord

**Description:**

This function opens the next record in the currently selected folder.

**Dependencies:**

Must have made a previous call to PCEM\_GetFirstRecord.

**Parameters:**

None

**Return Values:**

The record ID on success, "End" if there are no more records, otherwise an "!!ERROR!!"

### PCEM\_OpenRecord( OutlookID )

**Description:**

This function will search across the entire outlook application for the provided record ID.

**Dependencies:**

None

**Parameters:**

None

**Return Values:**

"0" on success, otherwise an "!!ERROR!!"

## **PCEM\_GetRecordCount**

### **Description:**

This function will return the number of records in the currently selected folder.

### **Dependencies:**

Must have a folder currently open.

### **Parameters:**

None

### **Return Values:**

A number on success, otherwise an "!!ERROR!!"

## **PCEM\_MoveRecord( FolderPath )**

### **Description:**

This function will move the currently selected record to the provided folder path.

### **Dependencies:**

Must have a record currently open.

### **Parameters:**

FolderPath = The destination for the current record.

Values = A folder path delimited with the current global delimiter.

### **Return Values:**

"0" on success, otherwise an "!!ERROR!!"

## **PCEM\_NewRecord( OptType )**

### **Description:**

This function creates a new record in Outlook and either uses the context of the current folder to determine its type, or uses the provided type and places the new record in the type's corresponding default folder.

### **Dependencies:**

None

### **Parameters:**

OptType = The optional type of the newly created record

Values = "Contacts", "Note", "Task", or "Event"

### **Return Values:**

The new record's ID, otherwise an "!!ERROR!!"

## PCEM\_DeleteCurrentRecord( Permanent )

### **Description:**

This function will delete the currently selected record.

### **Parameters:**

Permanent = A "True" value permanently deletes the item. A "False" value moves the item to the deleted items folder in Outlook.

Default Value = "False"

### **Return Values:**

0 for success or !!ERROR!!

## PCEM\_Display( ShowInForeground )

### **Description:**

This function will open the currently selected record in Outlook.

### **Dependencies:**

Must have a record currently opened.

### **Parameters:**

ShowInForeground = Determines if the opened record will be the active window

Values = "true" or "false"

### **Return Values:**

0 for success or !!ERROR!!

## PCEM\_SaveRecordAsMsg( StrPath )

### **Description:**

This function saves the currently selected record to the provided path, as the ".EML" format.

### **Dependencies:**

Must have a record currently open.

### **Parameters:**

StrPath = The path on the hard drive to save the currently selected record

Values = A path similar to "~/Desktop/"

### **Return Values:**

"0" on success, otherwise an "!!ERROR!!"

### **PCEM\_SetFieldData( FieldName ; Data )**

#### **Description:**

This function allows the user to set fields in the currently selected record. The list of field types are documented in the fields list for each type of record.

#### **Dependencies:**

Must have a record currently open.

#### **Parameters:**

FieldName = The field for which the data is to be set. Values are any predefined field name

Data = The data to set for the specified field

#### **Return Values:**

"0" on success, otherwise an "!!ERROR!!"

### **PCEM\_GetFieldData( FieldName )**

#### **Description:**

This function allows you to retrieve field data from a particular record, for the corresponding field name.

#### **Dependencies:**

Must have a record currently open.

#### **Parameters:**

FieldName = The name of the field the user wishes to retrieve data for

Values = Any of the documented fields for the corresponding record type

#### **Return Values:**

The field data on success, otherwise an "!!ERROR!!"

## **PCEM SendMail( To ; Cc ; Bcc ; Subject ; Body ; Attachments ; OptUseHtml )**

### **Description:**

This function allows you to send an email through Outlook, while setting multiple attachments and email formatting if desired.

### **Dependencies: none**

### **Parameters:**

To	The recipients of the email. Values are valid email addresses.
Cc	The carbon copy recipients. Values are valid email addresses.
Bcc	The blind carbon recipients. Values are valid email addresses.
Subject	The subject of the message. Values are any string.
Body	The body of the message. Values are any string.
Attachments	Any attachments for the message. Values are valid file paths.
OptUseHTML	A flag to tell the plug-in if the email is HTML formatted. Values are "True" for HTML formatting, or "False" for plain text formatting.

### **Return Values:**

"0" on success, otherwise an "!!ERROR!!"

## **PCEM GetAttachmentCount**

### **Description:**

This function returns the number of files attached to the current email message.

### **Dependencies:**

Must have a mail record currently open.

### **Parameters:**

None

### **Return Values:**

The number of attachments on success, otherwise an "!!ERROR!!"

## **PCEM GetAttachmentName( Index )**

### **Description:**

This function returns the name of an attachment from a specified record at a particular index.

### **Dependencies:**

Must have a record currently open.

### **Parameters:**

Index = The index that the attachment is located at for a particular record

Values = Any value greater than 0 and less than or equal to the number of attachments

### **Return Values:**

The name of the attachment on success, otherwise an "!!ERROR!!"

### **PCEM GetAttachmentSize( Index )**

**Description:**

This function returns the size of a particular attachment in the number of bytes.

**Dependencies:**

Must have a mail record currently open and that record must have one or more attachments.

**Parameters:**

Index = The index that the attachment is located at for a particular record

Values = Any value greater than 0 and less than or equal to the number of attachments

**Return Values:**

A number representing the size of the attachment in bytes on success, otherwise an "!!ERROR!!"

### **PCEM GetAttachmentType( Index )**

**Description:**

This function returns the type of the attachment as specified in the email's header.

**Dependencies:**

Must have an email currently open with files attached.

**Parameters:**

Index = The index that the attachment is located at for a particular record

Values = Any value greater than 0 and less than or equal to the number of attachments

**Return Values:**

The type of the attachment on success, otherwise an "!!ERROR!!"

### **PCEM SaveAttachment( FilePath ; Index )**

#### **Description:**

This function allows the user to save an attachment to the hard drive.

#### **Dependencies:**

Must have a record currently open that has attachments.

#### **Parameters:**

FilePath = The path on the hard drive to save the attachment at. Values are any valid file path.

Index = The index that the attachment is located at for a particular record. Values are any value greater than 0 and less than or equal to the number of attachments.

#### **Return Values:**

"0" on success, otherwise an "!!ERROR!!"

### **PCEM SaveContactImage( Path )**

#### **Description:**

This function allows the user to save a contacts image to the hard drive. The opposite can be done with the "setfielddata" function.

#### **Dependencies:**

Must have a contact record currently open.

#### **Parameters:**

Path = The path on the hard drive to save the attachment at. Values are any valid file path.

#### **Return Values:**

"0" on success, otherwise an "!!ERROR!!"

## 4) Folder Specific Functions

### PCEM\_GetDefaultRootFolder

**Description:**

Obtains the Default account in Outlook.

**Parameters:**

None

**Return Values:**

The name of that particular Default account, otherwise an "!!ERROR!!"

### PCEM\_GetFirstFolder

**Description:**

Returns the full path of the first folder identified by Outlook in the current root folder.

**Parameters:**

None

**Return Values:**

A path to a folder or !!ERROR!!

### PCEM\_GetNextFolder

**Description:**

Returns the full path of the next folder identified by outlook in the current root folder. A call to GetFirstFolder must be made before calling this function for the first time.

**Parameters:**

None

**Return Values:**

A path to a folder, "End" if there are no more folders or !!ERROR!!

## **PCEM\_OpenFolder( FolderPath )**

### **Description:**

This function can open a folder, provided the user already knows the name, across all folders in the current open account.

### **Dependencies:**

Must have called PCEM\_GetFirstRootFolder, GetNextRootFolder, or OpenRootFolder, otherwise it will point to the default account.

### **Parameters:**

None

### **Return Values:**

"0" on success, otherwise an "!!ERROR!!"

## **PCEM\_GetFolderCount**

### **Description:**

This function returns the total number of folders in the opened root folder.

### **Dependencies:**

Must have made a previous call to PCEM\_OpenRootFolder.

### **Parameters:**

None

### **Return Values:**

A number representing the amount of folders in the root folder/account, otherwise an "!!ERROR!!"

## **PCEM\_GetFolderType**

### **Description:**

This function returns the type of the currently opened folder.

### **Dependencies:**

Must have made a previous call to PCEM\_GetFirstFolder, GetNextFolder, or OpenFolder.

### **Parameters:**

None

### **Return Values:**

"Mail", "Contacts", "Calendar", "Tasks", "Note", "Generic", otherwise an "!!ERROR!!"

## **PCEM\_GetDefaultFolderName( ItemType )**

### **Description:**

Provided the type of folder, this function returns the default folder for a particular account.

### **Dependencies:**

Must have made a previous call to PCEM\_GetFirstFolder, GetNextFolder, or OpenFolder.

### **Parameters:**

ItemType = Designates which type of default folder to return

Values = "Inbox", "Contacts", "Notes", "Calendar", or "Tasks"

Default Value = "Inbox"

### **Return Values:**

The name of that particular default folder, otherwise an "!!ERROR!!"

## **PCEM\_NewFolder( Name ; OptType )**

### **Description:**

This function creates a new folder and places it inside of whatever folder you are currently located or based on the provided folder path.

### **Dependencies:**

Must have an account currently open.

### **Parameters:**

Name                      The name of the newly created folder. Values are any string made of standard alpha characters.

OptType                    The optional type of the newly created record. Values are "Mail," "Contact," "Note," "Task," "Calendar"

### **Return Values:**

"0" on success, otherwise an "!!ERROR!!"

## 5) Root and Shared Folder Functions

### PCEM\_GetFirstRootFolder

**Description:**

This function retrieves the name of the first designated account of Outlook.

**Parameters:**

None

**Return Values:**

The name of the first account on success, otherwise an "!!ERROR!!"

### PCEM\_GetNextRootFolder

**Description:**

Returns the Name of the next Mailbox/folder encountered by Outlook. Requires a call to GetFirstRootFolder before the first call to GetNextRootFolder.

**Dependencies:**

A call to PCEM\_GetFirstRootFolder.

**Parameters:**

None

**Return Values:**

The next account name on success, "End" if there are no more folders, otherwise an "!!ERROR!!"

### PCEM\_OpenRootFolder( RootFolderName )

**Description:**

This function opens an Outlook account based off its name, and also creates an index map of all the folders in that function. This allows the use of the iterative folder functions.

**Parameters:**

RootFolderName = The name of the account or root folder (Mailbox) to open. Values are any account name

Example: ( "Public Folders" , "Mailbox – User" or "[support@productivecomputing.com](mailto:support@productivecomputing.com)" )

**Return Values:**

"0" on success, otherwise an "!!ERROR!!"

### III. Available Outlook Fields

Outlook has various records or item types such as Contact Items, Task Items, Mail Item etc. Each item contains various fields. These fields have values which can either be read or set. In order to read or "get" an Outlook field value you must call the PCEM\_GetFieldData function and specify the field you would like to get. In order to write or "set" an Outlook field value you must call the PCEM\_SetFieldData function and specify the field you would like to set.

All available fields are listed below and ARE CASE SENSITIVE. The fields are separated by item type and indicate if you can "get" or "set" the fields. If there is a missing field that you would like to access, then please email a request to [support@productivecomputing.com](mailto:support@productivecomputing.com)

<b>Available Contact Fields</b>
---------------------------------

Contact Items represent a contact in a Contacts folder.

Method	Field Name	Acceptable Values/Notes
Get	Address Book	
Get/Set	Age	
Get/Set	Anniversary	
Get/Set	Assistant's Phone	
Get/Set	Astrology Sign	
Get/Set	Birthday	
Get/Set	Blood Type	
Get/Set	Body	
Get/Set	Body HTML	
Get/Set	Business Address City	
Get/Set	Business Address Country	
Get/Set	Business Address Postal Code	
Get/Set	Business Address State	
Get/Set	Business Address Street	
Get/Set	Business Fax	
Get/Set	Business Phone	
Get/Set	Business Phone 2	
Get/Set	Business Web Page	
Get/Set	Categories	
Get/Set	Children	
Get/Set	Children Furigana	
Get/Set	Company	
Get/Set	Company Furigana	
Get/Set	Complete	Not Flagged, Not Complete, Completed
Get/Set	Custom Date 1	
Get/Set	Custom Date 2	
Get/Set	Custom Field 1	
Get/Set	Custom Field 2	
Get/Set	Custom Field 3	
Get/Set	Custom Field 4	
Get/Set	Custom Field 5	
Get/Set	Custom Field 6	

Method	Field Name	Acceptable Values/Notes
Get/Set	Custom Field 7	
Get/Set	Custom Field 8	
Get/Set	Custom Phone 1	
Get/Set	Custom Phone 2	
Get/Set	Custom Phone 3	
Get/Set	Custom Phone 4	
Get	Date Completed	
Get/Set	Department	
Get/Set	Display Name	
Get/Set	Display Type	Person, DistributionList, PrivateDistributionList, MeetingRoom, Equipment, Unknown
Get/Set	Dot Net IMS	
Get/Set	Due Date	
Get	Exchange ID	
Get/Set	First Name	
Get/Set	First Name Furigana	
Get/Set	Hobbies	
Get/Set	Home Address City	
Get/Set	Home Address Country	
Get/Set	Home Address Postal Code	
Get/Set	Home Address State	
Get/Set	Home Address Street	
Get/Set	Home Emails	
Get/Set	Home Fax	
Get/Set	Home Phone	
Get/Set	Home Phone 2	
Set	Image	
Get/Set	Is Japanese Contact	
Get/Set	Job Title	
Get/Set	Last Name	
Get/Set	Last Name Furigana	
Get/Set	Middle Name	
Get/Set	Mobile Phone	
Get	Modify	
Get/Set	Nickname	
Get/Set	Object Description	
Get/Set	Other Emails	
Get/Set	Other IMS	
Get/Set	Pager	
Get/Set	Primary Phone	
Get/Set	Reminder Time	
Get/Set	SIP IMS	
Get/Set	Spouse	
Get/Set	Spouse Furigana	
Get/Set	Start Date	
Get/Set	Suffix	
Get/Set	Title	

Method	Field Name	Acceptable Values/Notes
Get	Vcard Data	
Get/Set	Web Page	
Get/Set	Work Emails	

## Available Mail Fields

Mail items represent a mail message in an Inbox folder.

Method	Field Name	Acceptable Values/Notes
Get	Account	
Get	Bcc	
Get	Bcc Names	
Get/Set	Body	
Get/Set	Body HTML	
Get/Set	Categories	
Get	Cc Emails	
Get	Cc Names	
Get/Set	Complete	Not Flagged, Not Complete, Completed
Get	Date Completed	
Get/Set	Due Date	
Get	Edited	
Get	Exchange ID	
Get	Folder	
Get	Forwarded	
Get	Has HTML	
Get	Header	
Get	Is Meeting	
Get	Is Read	
Get	Modify	
Get	Priority	High, Normal, Low
Get	Received	
Get	Redirected	
Get/Set	Reminder Time	
Get	Replied To	
Get	Replied To All	
Get	Sender Email	
Get	Sender Name	
Get	Sent	
Get	S/MIME Encrypted	
Get	S/MIME Signed	
Get	Source	
Get/Set	Start Date	
Get/Set	Subject	
Get	To Emails	
Get	To Names	
Get	Type	Meeting, Message

## Available Events Fields

Event Items represent calendar events such as meetings, one-time appointments, or recurring events.

Method	Field Name	Acceptable Values/Notes
Get	Account	
Get/Set	All Day Event	
Get/Set	Body	
Get/Set	Body HTML	
Get/Set	Busy Status	Busy, Free, Tentative, OutOfOffice, Unknown
Get	Calendar	
Get/Set	Categories	
Get/Set	End	
Get	Exchange ID	
Get	Has HTML	
Get	iCalendar Data	
Get/Set	Is Occurrence	
Get/Set	Is Private	
Get/Set	Is Recurring	
Get/Set	Location	
Get/Set	Master	
Get	Modify	
Get/Set	Organizer	
Get	Recurrence	
Get	Recurrence ID	
Get/Set	Remind Beforehand	
Get/Set	Reminder	
Get/Set	Request Responses	
Get/Set	Required Attendees	
Get/Set	Start	
Get/Set	Subject	
Get	Timezone Name	
Get	Timezone Offset	

## Available Task Fields

Represents a task (an assigned, delegated, or self-imposed task to be performed within a specified time frame) in a Tasks folder.

Method	Field Name	Acceptable Values/Notes
Get/Set	Body	
Get/Set	Body HTML	
Get/Set	Categories	
Get/Set	Complete	Not Flagged, Not Complete, Completed
Get	Date Completed	
Get/Set	Due Date	
Get	Exchange ID	
Get	Folder	
Get	iCalendar Data	
Get/Set	Importance	Low, Normal, High, Unknown
Get	Modify	
Get/Set	Reminder Time	
Get/Set	Start Date	
Get/Set	Subject	

## Available Note Fields

Note Items represents a note in a Notes folder.

Method	Field Name	Acceptable Values/Notes
Get/Set	Body	
Get/Set	Body HTML	
Get/Set	Categories	
Get	Created	
Get	Exchange ID	
Get	Folder	
Get	iCalendar Data	
Get	Modify	
Get/Set	Name	

## 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](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!