Productive Computing

Exchange Manipulator SE

Migration Guide

Outlook Manipulator to Exchange Manipulator SE

Revised April 21, 2022

This migration guide is intended to provide guidance on migrating your scripts from using the Outlook Manipulator plug-in integrated to communicate with the Microsoft® Outlook application to using the Exchange Manipulator SE plug-in features. This migration guide only applies if your solution is designed to communicate with Exchange accounts; if usage of IMAP/POP3, Gmail, or other non-Exchange mailbox accounts are required, the original Outlook Manipulator plug-in integration will suffice.

The Exchange Manipulator SE plug-in also allows for server-side integration. For further information on server-side integration, please refer to the Developer's Guide.

Authentication

Unlike the Outlook Manipulator, the Exchange Manipulator SE plug-in provides two different forms of authentication, depending on what kind of Exchange server the plug-in will communicate with:

> 1) Exchange On-Premise, which is a locally-hosted Exchange server, usually in-house or in-network
> 2) Exchange Online, which is an Exchange server hosted online through the Office 365 or Outlook 365 mailbox service

For communicating with Exchange On-Premise servers, the solution should use the "PCEX_Authenticate" function, providing the mailbox address, user name, password, and optionally the domain. The method of communication with On-Premise servers is known as "basic authentication", as it requires a basic username and password to connect to the mailbox.

For communicating with Exchange Online servers, the solution should use the "PCEX_BeginSession" and "PCEX_Authorize" functions. Exchange Online servers require "OAuth" authentication methods in order rto establish a connection, which explains why authentication takes place in a two-step process.
For more information on authentication, as well as how to configure the plug-in for communicating with Exchange On-Premise and Exchange Online servers, please refer to the Developer's Guide.

Authenticated User Permissions

When the plug-in has successfully authenticated with Exchange, all communication will be performed as though the user had performed the actions. This also means that anything the user has permission to perform, the plug-in will also have permission to perform. If the solution attempts to communicate using another user's mailbox while authenticated (this is similar to calling PCEM_OpenRootFolder on another Outlook account's mailbox), the plug-in will only be able to do what that other mailbox allows the authenticated user to do.

For more information on this, please see the example present in the Developer's Guide and in the entry for PCEX_OpenMailbox in the Functions Guide.

Functional Workflows

The Exchange Manipulator SE plug-in functions very similarly to the Outlook Manipulator plug-in withregards to its logical order of operations and handling of records and folders. To pull an email into FileMaker, for example, the plug-in would perform the following logical pseudoscript:

PCEX_Authenticate
PCEX_OpenFolder(Some Mail Folder)
PCEX_OpenRecord(Record's Unique Identifier)
PCEX_GetFieldData(Some Field)
Repeat until done

The plug-in can also navigate records using PCEX_GetFirstRecord and PCEX_GetNextRecord, similar to the Outlook Manipulator's PCEM_GetFirstRecord and PCEM_GetNextRecord functions.

However, due to the inherent differences in the underlying structures of the Exchange Manipulator SE and Outlook Manipulator plug-ins, the plug-in functions will not map when replacing the Outlook Manipulator plug-in with the Exchange Manipulator SE plug-in. We advise keeping the original Outlook Manipulator plug-in installed while performing the migration so that your scripting will retain the original plug-in function calls, and reference those when replacing the functions with those from the Exchange Manipulator SE plug-in.

Functions

The following table displays a mapping of functions from the Outlook Manipulator plug-in to the Exchange Manipulator SE plug-in. Functions in **blue** are only found in the Outlook Manipulator plug-in, and functions in **green** are only found in the Exchange Manipulator plug-in:

Outlook Manipulator	Exchange Manipulator SE
PCEM_GetOperatingMode	PCEX_GetOperatingMode
PCEM_Register(ServerName ; ServerPort ; ServerPage ; LicenseID)	PCEX_Register(ServerName ; ServerPort ; ServerPage ; LicenseID)
PCEM_Version(Type)	PCEX_Version(Type)
PCEM_Authenticate(Username ; Password)	PCEX_Authenticate(Email ; optUsername ; optPassword ; optDomain)
PCEM_AddOneOffMember(Name ; Address ; Type)	
PCEM_CreateCustomField(Name ; Type)	
PCEM_DeleteCurentRecord(bPermanent)	PCEX_DeleteCurrentRecord(bPermanent)
PCEM_DeleteRecord(OutlookID ; Permanent)	PCEX_DeleteRecord(EntryID ; bPermanent)
PCEM_Display(ShowInForeground)	
PCEM_FilterByLastModified(Timestamp)	PCEX_FilterByLastModified(Timestamp)
PCEM_GetCustomFieldData(FieldName)	PCEX_GetCustomFieldData(FieldName ; FieldType)
PCEM_GetFieldData(FieldName)	PCEX_GetFieldData(FieldName)
PCEM_MoveRecord(FolderPath)	PCEX_MoveRecord(FolderPath)
PCEM_NewRecord(optModule)	PCEX_NewRecord(optModule)
PCEM_OpenRecord(OutlookID)	PCEX_OpenRecord(EntryID)
PCEM_OpenSavedItem(strPath)	
PCEM_SaveRecord(optionalParam)	PCEX_SaveRecord(optParam)
PCEM_SaveRecordAsMsg(strPath)	PCEX_SaveRecordAsEML(strPath)
PCEM_SendMail(To ; Cc ; Bcc ; Subject ; Body ; Attachments ; optUseHTML)	PCEX_SendMail(To ; Cc ; Bcc ; Subject ; Body ; Attachments ; optUseHTML)
PCEM_SetFieldData(FieldName ; Data)	PCEX_SetFieldData(FieldName ; FieldData)
PCEM_VerifyFieldData(FieldName ; VerifyData)	PCEX_VerifyFieldData(FieldName ; VerifyData)
PCEM_SetCustomFieldData(FieldName ; Data ; optType)	PCEX_SetCustomFieldData(FieldName ; FieldData ; FieldType)
PCEM_AddAttachment(FilePath ; Name)	PCEX_AddAttachment(FilePath ; Name)
PCEM_GetAttachmentCount	PCEX_GetAttachmentCount
PCEM_GetAttachmentName(Index ; optTruncated)	PCEX_GetAttachmentName(Index)
PCEX_GetAttachmentSize(Index)	PCEX_GetAttachmentSize(Index)
PCEX_SaveAttachment(FilePath ; Index)	PCEX_SaveAttachment(FilePath ; Index)
PCEM_DeleteAllRecords(ExcludePrivate ; ExcludeDistList)	PCEX_DeleteAllRecords(ExcludePrivate ; ExcludeDistList)
PCEM_EmptyAllDeletedItems	PCEX_EmptyAllDeletedItems
PCEM_GetCurrentFolderName	PCEX_GetCurrentFolderName

PCEM_GetFirstFolder	PCEX_GetFirstFolder
PCEM_GetNextFolder	PCEX_GetNextFolder
PCEM_GetFirstRecord	PCEX_GetFirstRecord
PCEM_GetNextRecord	PCEX_GetNextRecord
PCEM_GetDefaultFolderName(ItemType)	PCEX_GetDefaultFolderName(ItemType)
PCEM_GetFolderID	PCEX_GetFolderID
PCEM_GetFolderType	PCEX_GetFolderType
PCEM_GetRecordCount	PCEX_GetRecordCount
PCEM_NewFolder(Name ; Type)	PCEX_NewFolder(Name ; Type)
PCEM_OpenFolder(FolderPath)	PCEX_OpenFolder(FolderPath)
PCEM_GetFirstRootFolder	
PCEM_GetNextRootFolder	
PCEM_OpenRootFolder(RootFolderName)	PCEX_OpenMailbox(MailboxName)
PCEM_OpenSharedFolder(ContactName ; Folder)	PCEX_OpenSharedFolder(ContactName ; Folder)
PCEM_RPClearRecurrencePattern	
PCEM_RPCloseOccurrenceItem(optSaveChanges)	
PCEM_RPCloseRecurrencePattern(optSaveChanges)	
PCEM_RPDeleteOccurrence(Date)	
PCEM_RPGetExceptionCount	
PCEM_RPOpenOccurrence(Date)	
PCEM_RPOpenRecurrencePattern	
PCEM_RPOpenFirstException	
PCEM_OpenNextException	
PCEM_RPOpenExceptionAt(Index)	
	PCEX_GetContactImage
	PCEX_SetContactImage(Image)
	PCEX_GetAttachmentCID(Index)

Notable Differences:

1) The Exchange Manipulator SE plug-in does not have the ability to access and navigate a list of available "root folders" from the Exchange environment, as those root folders are a construct of the Outlook application. As a result, the GetFirstRootFolder and GetNextRootFolder functions are not available in the Exchange Manipulator SE plug-in. Other root folders or "mailboxes" can still be accessed, but must be specified explicitly by name when calling PCEX_OpenMailbox.

2) The ability to manipulate recurrence patterns is not currently implemented in the Exchange Manipulator SE plug-in.

Scripting

The way that data is accessed, manipulated, and transferred between FileMaker and an Exchange account in the Exchange Manipulator SE plug-in is primarily similar to the methodology of the Outlook Manipulator plug-in.

The following is a comparison of scripts showing the similarity of the Exchange Manipulator SE plug-in and the Outlook Manipulator plug-in.

Below is the script for adding a contact named John Smith to a Contacts subfolder called "Summer Friends" using the Outlook Manipulator:

```
# Open the root folder
If [PCEM_OpenRootFolder ( Main::gFolder Root ) <> 0]
    Show Custom Dialog ["Warning"; PCEM_GetLastError]
    Exit Script [Text Result:]
End If

# Open the folder (sets the pointer in Outlook to the specified folder)
Set Variable [$FolderID; Value: PCEM_OpenFolder( "/Contacts/Summer Friends" )]
If [$FolderID = "!!ERROR!!" or $FolderID = "?"]
    Show Custom Dialog ["Warning"; PCEM_GetLastError]
    Exit Script [Text Result:]
End If

# New Record
If [PCEM_NewRecord( "Contacts" ) <> 0]
    Show Custom Dialog ["Warning"; PCEM_GetLastError]
    Exit Script [Text Result:]
End If

# Set fields; more fields are available, but we're only using a small set.
Set Variable [$result; Value: PCEM_SetFieldData ( "First Name" ; "John" )]
Set Variable [$result; Value: PCEM_SetFieldData ( "Last Name" ; "Smith" )]
Set Variable [$result; Value: PCEM_SetFieldData ( "E-mail" ; "John.Smith@TestCompany.com"
)]
Set Variable [$result; Value: PCEM_SetFieldData ( "Home Address City" ; "Testville" )]
Set Variable [$result; Value: PCEM_SetFieldData ( "Home Address Country" ; "USA" )]
Set Variable [$result; Value: PCEM_SetFieldData ( "Home Address Postal Code" ; "55555" )]
Set Variable [$result; Value: PCEM_SetFieldData ( "Home Address State" ; "Testsota" )]
Set Variable [$result; Value: PCEM_SetFieldData ( "Home Address Street" ; "123 Any Lane" )]

# Save the record
Set Variable [$OutlookID; Value: PCEM_SaveRecord]
If [$OutlookID = "!!ERROR!!" or $OutlookID = "?"]
    Show Custom Dialog ["Warning"; PCEM_GetLastError]
    Exit Script [Text Result:]
End If

# Store the Outlook ID for future use
Set Field [Contacts::OutlookID; $OutlookID]
```

The following script demonstrates the same process of adding a contact to the "Summer Friends" Contact folder using the Exchange Manipulator SE plug-in:

```
# Open the root mailbox
If [PCEX_OpenMailbox ( Main::gFolder Mailbox ) <> 0]
    Show Custom Dialog ["Warning"; PCEX_GetLastError]
    Exit Script [Text Result:]
End If

# Open the folder
Set Variable [$FolderID; Value: PCEX_OpenFolder( "/Contacts/Summer Friends" )]
If [$FolderID = "!!ERROR!!" or $FolderID = "?"]
    Show Custom Dialog ["Warning"; PCEX_GetLastError]
    Exit Script [Text Result:]
End If

# New Record
If [PCEX_NewRecord( "Contacts" ) <> 0]
    Show Custom Dialog ["Warning"; PCEX_GetLastError]
    Exit Script [Text Result:]
End If

# Set fields; more fields are available, but we're only using a small set.
Set Variable [$result; Value: PCEX_SetFieldData ( "First Name" ; "John" )]
Set Variable [$result; Value: PCEX_SetFieldData ( "Last Name" ; "Smith" )]
Set Variable [$result; Value: PCEX_SetFieldData ( "E-mail" ; "John.Smith@TestCompany.com" )]
Set Variable [$result; Value: PCEX_SetFieldData ( "Home Address City" ; "Testville" )]
Set Variable [$result; Value: PCEX_SetFieldData ( "Home Address Country" ; "USA" )]
Set Variable [$result; Value: PCEX_SetFieldData ( "Home Address Postal Code" ; "55555" )]
Set Variable [$result; Value: PCEX_SetFieldData ( "Home Address State" ; "Testsota" )]
Set Variable [$result; Value: PCEX_SetFieldData ( "Home Address Street" ; "123 Any Lane" )]

# Save the record
Set Variable [$EntryID; Value: PCEX_SaveRecord]
If [$EntryID = "!!ERROR!!" or $EntryID = "?"]
    Show Custom Dialog ["Warning"; PCEX_GetLastError]
    Exit Script [Text Result:]
End If

# Store the Entry ID for future use
Set Field [Contacts::EntryID; $EntryID]
```

As the above scripts show, the functionality of the two plug-ins is quite similar. When converting the scripts and calculation function calls from using Outlook Manipulator to using Exchange Manipulator SE, as mentioned earlier in this document, we highly recommend keeping the original Outlook Manipulator plug-in installed and loaded by FileMaker, so that the new functions can be matched quickly and easily.

If you are planning on using the server-side implementation of the Exchange Manipulator SE plug-in, please refer to the Developer's Guide for tips and tricks on using the Exchange Manipulator SE plug-in in a server environment.

Additional Support

If you require additional support in your migration from Outlook Manipulator to Exchange Manipulator SE for integration with Microsoft Exchange Web Services or would like to start a new integration between FileMaker and Microsoft Exchange Web Services, 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!