



# Functions guide

*Revised November 18, 2020*

## Table of Contents

I. Introduction .....	3
II. Functions Descriptions .....	4
1. General Functions .....	4
PCCM_Version( Type ) .....	4
PCCM_GetOperatingMode .....	4
PCCM_Register( ServerName ; ServerPort ; ServerPage ; LicenseID ) .....	5
PCCM_GetLastError( Type ) .....	6
2. Session Functions .....	7
PCCM_UseCredentials( Service ; Username ; Password ; optParams ) .....	7
3. Folder Functions .....	8
PCCM_ListAllFolders .....	8
PCCM_FetchFolderContents( FolderID ; optPrefix ; optDelimiter ; optMaxKeys ; optContinuationToken ) .....	9
PCCM_GetPropertyForFolder( PropertyName ; ContentIndex ) .....	10
PCCM_CreateFolder( FolderName ) .....	11
PCCM_DeleteFolder( FolderID ) .....	12
4. Object Functions .....	13
PCCM_GetObject( FolderID ; ObjectID ; FilePath ) .....	13
PCCM_PostObject( FolderID ; ObjectID ; FilePath ; optDeleteAfterPost ) .....	14
PCCM_CopyObject( SrcFolderID ; SrcObjectID ; DestFolderID ; DestObjectID ) .....	15
PCCM_DeleteObject( FolderID ; ObjectID ) .....	16
PCCM_GetPresignedURL( FolderID ; ObjectID ; optExpireMinutes ) .....	17
III. List of Available Fields .....	18
Folder Fields .....	18
Object Fields .....	19
IV. Contact Us .....	20

## I. Introduction

### **Description:**

The Cloud Manipulator plug-in connects your FileMaker solution with Amazon Web Services (AWS) S3 storage. This plug-in provides FileMaker users with the ability to perform high-level management of AWS S3 buckets, as well as upload and download files with the AWS S3 service, using a convenient, easy-to-understand set of functions. 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 AWS fields. Please see the accompanying "Developer's Guide" for a list of 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 tutorial videos
- 4) Review our FileMaker demo file
- 5) Familiarize yourself with Google Suite

## II. Functions Descriptions

### 1. General Functions

#### **PCCM\_Version( Type )**

**Purpose:**

This function returns either the version string or the version number of the plug-in, depending on the Type parameter value.

**Parameters:**

Name	Purpose	Values	Default
Type	Type of version to return	“Long” or “Short”	“Long”

**Return Values:**

The version of the plug-in.

#### **PCCM\_GetOperatingMode**

**Purpose:**

This function returns the current operating status of the plug-in. The operating status is dependent on whether the plug-in has been registered with PCCM\_Register and what kind of license was used in registration.

**Parameters:**

None.

**Return Value:**

“UNREGISTERED” if not registered, “DEMO” if registered in demo mode, “LIVE” if registered in live mode, or “EXPIRED” if the demo expiration time has lapsed.

## **PCCM\_Register( ServerName ; ServerPort ; ServerPage ; LicenseID )**

### **Purpose:**

This function registers the product with the Productive Computing, Inc. registration server. See the Installation and Registration sections in the Developer's Guide for more information about the registration process.

### **Parameters:**

<b>Name</b>	<b>Purpose</b>	<b>Values</b>	<b>Default</b>
ServerName	Name of the registration server		"licensing.productivecomputing.com"
ServerPort	Port of the registration server		"80"
ServerPage	Registration page		"/PCIReg/pcireg.php"
LicenseID	Purchased or demo license ID		"DEMO-CMX"

### **Returns:**

0 for success, otherwise "-10" and the reason of the error

## PCCM\_GetLastError( Type )

### Purpose:

This function returns the last error encountered by the plug-in during an operation. The Type determines whether to return the error string or the error number.

### Parameters:

Name	Purpose	Values	Default
Type	Type of error to return	"Text" or "Number"	"Text"

### Returns:

Last error string or number encountered.

## 2. Session Functions

### PCCM\_UseCredentials( Service ; Username ; Password ; optParams )

**Purpose:**

This function accepts a set of credentials for a specific service, connecting to that service and storing valid authentication information for any subsequent requests. Additional parameters can also be provided in a semicolon-separated key=value pair list, if applicable.

**Parameters:**

Name	Purpose	Values	Default
Service	Identifier for the storage service to communicate with	“S3”	
Username	Username or API Key for the service		
Password	Password or API Secret for the service		
optParams	Additional parameters, if requested	Semicolon-separated key=value pairs	

**Returns:**

0 for success, otherwise “!!ERROR!!”

**Notes:**

Currently, only the “S3” service is acceptable. This may change in the future as more services are implemented.

### 3. Folder Functions

#### **PCCM\_ListAllFolders**

**Purpose:**

This function queries the service selected by PCCM\_UseCredentials and returns all available folders that are accessible by the authenticated user account. The name “Folder” is another term for “Bucket” or “Directory”. Folders are returned as a return-separated list.

**Parameters:**

None.

**Returns:**

A return-separated list of folders or “!!ERROR!!” for an error.

## **PCCM\_FetchFolderContents( FolderID ; optPrefix ; optDelimiter ; optMaxKeys ; optContinuationToken )**

### **Purpose:**

This function queries the service for the contents and details of a provided folder. The result set can be customized by providing a prefix (the preceding part of the folder's objects' names), a delimiter (the character that separates each individual object), and the maximum number of keys or objects to return.

### **Parameters:**

<b>Name</b>	<b>Purpose</b>	<b>Values</b>	<b>Default</b>
FolderID	Identifier of the folder		
optPrefix	Prefix portion of object names to filter for		
optDelimiter	Separation character to split up the result set		Carriage return
optMaxKeys	Maximum number of objects to retrieve	Integer value	1000
optContinuationToken	A unique token code referencing the next set of contents to fetch		

### **Returns:**

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

### **Notes:**

If specified, the optContinuationToken parameter will tell the plug-in to fetch the next set of records belonging to the bucket. This is used in a process called "pagination", for when the plug-in needs to retrieve more objects than the optMaxKeys parameter specifies (e.g. the plug-in fetches up to 50 objects, and there are 100 objects belonging to the bucket). For more information on how to use pagination, please refer to the Developer's Guide.

## PCCM\_GetPropertyForFolder( PropertyName ; ContentIndex )

### Purpose:

This function returns data about a folder or object retrieved from the service. For a list of applicable property names, please see the Available Fields section below.

### Parameters:

Name	Purpose	Values	Default
PropertyName	Name of the property to retrieve		
ContentIndex	Optional. Index of the object in the folder's list	Integer value	

### Returns:

The property value or "!!ERROR!!" for error.

### Notes:

When requesting the properties of a folder, the ContentIndex field can be ignored or left blank.

When requesting the properties of an object, the ContentIndex specified the 0-based index of the object in the object list obtained by the PCCM\_FetchFolderContents function. The first object in the list is always index 0.

## PCCM\_CreateFolder( FolderName )

### Purpose:

This function creates a folder under the authenticated account. If provided (and if applicable), the Region parameter determines which region to create the folder under.

### Parameters:

Name	Purpose	Values	Default
FolderName	Name of the folder to create		

### Returns:

0 for success, otherwise “!!ERROR!!”

### Notes:

The user account that is authenticated must have “create” permissions in order to successfully create a folder. Permissions can be managed within the account’s security settings on the service’s website.

The folder will be created in the same region that the plug-in has authenticated to. The folder name must be globally unique; in other words, the name used for the new bucket must be unique across the entirety of Amazon AWS S3, not just for the user’s region.

## PCCM\_DeleteFolder( FolderID )

### Purpose:

This function deletes the specified folder from the authenticated user's account. Files belonging to the folder will also be deleted.

### Parameters:

Name	Purpose	Values	Default
FolderID	Identifier of the folder to delete		

### Returns:

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

### Notes:

The user account that is authenticated must have "delete" permissions in order to successfully delete a folder. Permissions can be managed within the account's security settings on the service's website.

## 4. Object Functions

### PCCM\_GetObject( FolderID ; ObjectID ; FilePath )

#### Purpose:

This function will retrieve an object from the service as specified by its ID and its folder ID, and save the contents to the provided FilePath. The file path must be a valid file that adheres to the operating system file format (such as “C:\Folder\File.ext” for Windows, or “~/Folder/File.ext” for Mac).

#### Parameters:

Name	Purpose	Values	Default
FolderID	Identifier of the folder containing the object		
ObjectID	Identifier of the object to download		
FilePath	Path to save the object to	Must be a valid file path according to the operating system	

#### Returns:

0 for success, otherwise “!!ERROR!!”

#### Notes:

The user account that is authenticated must have “read” permissions in order to download an object from the service. Permissions can be managed within the account’s security settings on the service’s website.

## **PCCM\_PostObject( FolderID ; ObjectID ; FilePath ; optDeleteAfterPost )**

### **Purpose:**

This function will upload a file to the service, saving it to the specified folder. The file path must be a valid file that adheres to the operating system file format (such as "C:\Folder\File.ext" for Windows, or "~/Folder/File.ext" for Mac), and the file to be uploaded must exist at that path. If specified, optDeleteAfterPost will tell the plug-in to delete the file from the local machine system after uploading.

### **Parameters:**

<b>Name</b>	<b>Purpose</b>	<b>Values</b>	<b>Default</b>
FolderID	Identifier of the folder containing the object		
ObjectID	Identifier to apply to the object	The identifier must be unique	
FilePath	Path to save the object to	Must be a valid file path according to the operating system	
optDeleteAfterPost	Optional. Flag for whether to clear the local file after upload	True or False	False

### **Returns:**

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

### **Notes:**

The user account that is authenticated must have "write" permissions in order to download an object from the service. Permissions can be managed within the account's security settings on the service's website.

## PCCM\_CopyObject( SrcFolderID ; SrcObjectID ; DestFolderID ; DestObjectID )

### Purpose:

This function will copy the source object located at the source folder and save it to the destination folder with the destination object's name/identifier. This is useful if the user wishes to "backup" a file on the service that is already uploaded.

### Parameters:

Name	Purpose	Values	Default
SrcFolderID	Identifier of the folder containing the object to copy		
SrcObjectID	Identifier of the object to be copied		
DestFolderID	Identifier of the folder the object will be copied to		
DestObjectID	Identifier to apply to the newly-copied object	The identifier must be unique	

### Returns:

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

### Notes:

The user account that is authenticated must have "write" permissions in order to copy an object from the service. Permissions can be managed within the account's security settings on the service's website.

## PCCM\_DeleteObject( FolderID ; ObjectID )

### Purpose:

This function will delete the specified object, clearing it completely from the service folder. This will completely delete the file; there is no way to recover the file after it is deleted.

### Parameters:

Name	Purpose	Values	Default
FolderID	Identifier of the folder containing the object		
ObjectID	Identifier of the object to be deleted		

### Returns:

0 for success, otherwise “!!ERROR!!”

### Notes:

The user account that is authenticated must have “delete” permissions in order to copy an object from the service. Permissions can be managed within the account’s security settings on the service’s website.

## PCCM\_GetPresignedURL( FolderID ; ObjectID ; optExpireMinutes )

### Purpose:

This function will retrieve the presigned URL for the provided object within the specified folder. This presigned URL is generated automatically by Amazon AWS and allows any user that uses the URL to access the object regardless of their user permissions. The optExpireMinutes parameter will allow the developer to specify how long the presigned URL will remain “active”, in minutes; by default, this is 7 days (10080 minutes).

### Parameters:

Name	Purpose	Values	Default
FolderID	Identifier of the folder containing the object		
ObjectID	Identifier of the object		
optExpireMinutes	Number of minutes to keep the URL alive. Optional.	Between 0 and 10800	10800

### Returns:

The presigned URL of the object, or “!!ERROR!!”

### Notes:

If an expire time of greater than 10800 minutes is provided to the function, the time will be set to 10800 minutes.

### III. List of Available Fields

#### Folder Fields

Name	Description
Keys	List of object names that belong to the folder
Count	Count of objects in the folder
IsTruncated	Boolean value indicating whether there are more objects to fetch (true) or not (false).
ContinuationToken	A unique token string that references the next set of records to retrieve from AWS S3. This value is only populated if IsTruncated is true.

## Object Fields

Name	Description
Key	Unique identifier of the object
Etag	Edit tag of the object. Read-only
OwnerName	Name of the account that owns the object
OwnerID	Unique identifier of the owner
Size	Size of the file in bytes
LastModified	Timestamp of when the object was last modified. This is returned as an ISO-8601 format date text.
Name	Name of the object. Also known as "DisplayName"
StorageClass	The class of storage assigned to the object

## 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: <https://fmforums.com/forum/297-filemaker-utility-plug-ins/>

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!