



Functions Guide

Revised May 4, 2011

950 Boardwalk, Suite 205, San Marcos, CA 92078 • (760) 510-1200 • www.productivecomputing.com

© Copyright 2011 Productive Computing, Inc.

TABLE OF CONTENTS

I. Introduction	3
II. Function Descriptions	4
1) Error, Registration and Authentication Related Functions.....	4
PCXP_Register(ServerName ; ServerPort ; ServerPage ; LicenseID)	4
PCXP_GetOperatingMode.....	4
PCXP_Version(type)	5
PCXP_GetLastError(type)	5
PCXP_Authenticate(UserName ; Password)	6
2) Contact Specific Functions	7
PCXP_GetAllContacts.....	7
PCXP_GetFirstContact	7
PCXP_GetNextContact	8
PCXP_GetContactField(FieldName)	8
PCXP_GetUpdates(SinceDate ; ListTypeOrID).....	9
PCXP_NewContact(Email ; ListID)	9
PCXP_OpenContact(IDorEmail)	10
PCXP_OptOutContact	10
PCXP_SaveContact.....	11
PCXP_SetContactField(FieldName ; Value)	11
PCXP_RemoveFromList(ListID).....	12
3) List Specific Functions	13
PCXP_DeleteList(ListID)	13
PCXP_GetAllLists.....	13
PCXP_GetFirstList.....	14
PCXP_GetContactsFromList.....	14
PCXP_GetNextList.....	14
PCXP_GetListField(FieldName)	15
PCXP_NewList(Name)	15
PCXP_SaveList.....	16
PCXP_SetListField(FieldName ; Value)	16
III. Available Constant Contact Fields	17
Available Contact Fields.....	17
Available List Fields.....	18
IV. Contact Us	19

I. Introduction

Description:

The eMail Marketing Connector is a tool used to bidirectionally exchange data between FileMaker® and Constant Contact®. Now you can import, add, update and remove contacts or lists between FileMaker and Constant Contact allowing you to effectively manage your eMail Marketing. These operations are accomplished using FileMaker function calls from within FileMaker calculations. These calculations are generally determined from within FileMaker "SetField," "SetVariable" or "If" script steps. For a list of the basic integration steps, please see the accompanying Developers Guide document.

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) Reverse engineer our FileMaker demo and review video tutorials

Demo and video tutorials can be found here: <http://www.productivecomputing.com/plugins/email-marketing-connector-id-134/>

- 4) Familiarize yourself with Constant Contact: <http://www.constantcontact.com>

Error Handling:

Any of the plug-in functions may encounter an error during processing. When an error occurs during processing, immediately call the PCXP_GetLastError function in order to obtain a full description of the error or error number. This function returns the error message or error number associated with the last error in order to troubleshoot script or logic failures. Please see PCXP_GetLastError function description in the Functions Guide and the "Handling Errors" section in the Developer's Guide for further clarification on how to properly trap for errors.

II. Function Descriptions

This section describes the functions that are available with the eMail Marketing Connector plug-in.

1) Error, Registration and Authentication Related Functions

PCXP_Register(ServerName ; ServerPort ; ServerPage ; LicenseID)

Purpose:

This function is used to register the plug-in with Productive Computing. It must be called successfully before the plug-in can be used.

Dependencies:

None

Parameters:

Parameter Name	Purpose	Values	Default Value
ServerName	The url of the Productive Computing registration server.	"licensing.productivecomputing.com"	
ServerPort	The port that the registration server is listening on.	"80"	
ServerPage	The path of the registration script.	"/PCIReg/pcireg.asp"	
LicenseID	Either the demo code, or the license ID provided by Productive Computing upon purchase of the plug-in.	"your license ID" or "DEMO-XP1"	

Return Values:

0 on success, otherwise an !!ERROR!!

Notes & Examples:

Example: PCXP_Register("licensing.productivecomputing.com" ; "80" ; "/PCIReg/pcireg.asp" ; "DEMO-XP1")

PCXP_GetOperatingMode

Purpose:

This function will return the current operating mode of the plug-in.

Dependencies:

None

Parameters:

None

Return Values:

"LIVE", "DEMO", "EXPIRED" or "UNREGISTERED" otherwise an "!!ERROR!!"

PCXP_Version(type)

Purpose:

Used to identify the plug-in version and/or plug-in name installed on a machine.

Dependencies:

None

Parameters:

Parameter Name	Purpose	Values	Default Value
type	selects the long or short format for the returned version string.	"long" or "short"	short

Return Values:

A version string, otherwise an !!ERROR!!

PCXP_GetLastError(type)

Purpose:

This function is used to retrieve the last error logged by the plug-in.

Dependencies:

None

Parameters:

Parameter Name	Purpose	Values	Default Value
type	To select between the error number or error string.	"text" or "number"	text

Return Values:

An error text or number, otherwise "!!ERROR!!"

PCXP Authenticate(UserName ; Password)

Purpose:

This function is used to set the username and password for the user's Constant Contact account.

Dependencies:

None

Parameters:

Parameter Name	Purpose	Values	Default Value
UserName	The username of the user's Constant Contact account.		
Password	The password of the user's Constant Contact account.		

Return Values:

0 on success, otherwise an !!ERROR!!

2) Contact Specific Functions

PCXP_GetAllContacts

Purpose:

This function retrieves a list of all the user's contacts from the Constant Contact server.

Dependencies:

PCXP_Authenticate

Parameters:

None

Return Values:

A time stamp on success, otherwise an !!ERROR!!

PCXP_GetFirstContact

Purpose:

This function retrieves the first contact ID from a list of contacts.

Dependencies:

A previous call to PCXP_GetAllContacts, PCXP_GetUpdates, or PCXP_GetContactsFromList

Parameters:

None

Return Values:

A contact ID on success, otherwise an !!ERROR!!

PCXP_GetNextContact

Purpose:

This function retrieves the next contact ID from a list of contacts.

Dependencies:

A previous call to PCXP_GetAllContacts, PCXP_GetUpdates, or PCXP_GetContactsFromList

Parameters:

None

Return Values:

A contact ID, END or !ERROR!!

Notes & Examples:

A contact ID will be returned upon success. When the end of the contacts list is reached, then "END" will be returned.

PCXP_GetContactField(FieldName)

Purpose:

This function retrieves field values from a selected contact for the associated key.

Dependencies:

A previous call to PCXP_GetContact

Parameters:

Parameter Name	Purpose	Values	Default Value
FieldName	Identifies what data to retrieve	see "Available Field Names" section in this document	

Return Values:

The field value on success, otherwise an !!ERROR!!

Notes & Examples:

When calling GetAllContacts and then GetFirstContact or GetNextContact, the GetContactfield function can retrieve only the ID and email of the contact. If more detailed information is desired, then OpenContact must be called.

PCXP_GetUpdates(SinceDate ; ListTypeOrID)

Purpose:

This function retrieves all of the contacts that have been updated since a previous date.

Dependencies:

A previous call to PCXP_GetAllContacts, PCXP_GetUpdates or PCXP_GetContactsFromList and the associated time stamp.

Parameters:

Parameter Name	Purpose	Values	Default Value
SinceDate	All contacts that have been updated since this time stamp will be returned.		
ListTypeOrID	Identifies which list to retrieve the updates from.	"Active", "DoNotMail", "Removed" or a list ID number	

Return Values:

A time stamp from the list of updates, otherwise an !!ERROR!!

Notes & Examples:

PCXP_GetUpdates("2011-04-20T18:43:37.390Z" ; "1")

PCXP_NewContact(Email ; ListID)

Purpose:

This function creates a new contact with the provided email and assigns it to the ListID.

Dependencies:

A previous call to PCXP_Authenticate

Parameters:

Parameter Name	Purpose	Values	Default Value
Email	A unique email for a new contact.		
ListID	The ID of an existing list.		

Return Values:

0 on success, otherwise an !!ERROR!!

Notes & Examples:

A new or existing contact can be added to multiple lists with subsequent calls to PCXP_SetContactField ("ContactList"; ListID)

PCXP_OpenContact(IDorEmail)

Purpose:

This function will retrieve/open the specified contact allowing you to subsequently call PCXP_GetContactField to retrieve the contact details.

Dependencies:

A previous call to PCXP_Authenticate

Parameters:

Parameter Name	Purpose	Values	Default Value
IDorEmail	Can be either a contact's email address or the contact's ID number.		

Return Values:

A contact ID or email depending on the parameter passed, otherwise an !!ERROR!!

PCXP_OptOutContact

Purpose:

This function will set a contact's status so that they are opted out from any mailing lists.

Dependencies:

A previous call to PCXP_OpenContact

Parameters:

None

Return Values:

0 on success, otherwise an !!ERROR!!

PCXP_SaveContact

Purpose:

This function will either add a new contact or save changes to a existing contact, depending on previous function calls.

Dependencies:

A previous call to PCXP_NewContact or PCXP_OpenContact.

Parameters:

None

Return Values:

A contact ID on success, otherwise an !!ERROR!!

PCXP_SetContactField(FieldName : Value)

Purpose:

This function can set data for a contact.

Dependencies:

A previous call to PCXP_OpenContact or PCXP_NewContact

Parameters:

Parameter Name	Purpose	Values	Default Value
FieldName	The field to be added or modified.	see "Available Field Names" section in this document	
Value	The value to be assigned to the provided FieldName.		

Return Values:

0 on success, otherwise an !!ERROR!!

Notes & Examples:

Must call PCXP_SaveContact after any number of calls to this function, if the changes are to be made permanent.

PCXP_RemoveFromList(ListID)

Purpose:

This function will remove the currently selected contact from the provided list.

Dependencies:

A previous call to PCXP_OpenContact

Parameters:

Parameter Name	Purpose	Values	Default Value
ListID	The ID of the list that the contact is to be removed from.		

Return Values:

0 on success, otherwise an !!ERROR!!

Notes & Examples:

This function must be followed by a call to PCXP_SaveContact, in order for the change to be made permanent.

3) List Specific Functions

PCXP_DeleteList(ListID)

Purpose:

This function will permanently delete the currently selected list. It will not delete any of the contacts associated with that list.

Dependencies:

A previous call to PCXP_GetFirstList or PCXP_GetNextList

Parameters:

Parameter Name	Purpose	Values	Default Value
ListID	The ID of the list to be deleted. This operation will fail on any of the three built in lists "active", "donotmail" and "removed". It will half fail when trying to delete the default list, meaning the list won't be removed, but all the contacts will be flushed from that list.		

Return Values:

0 on success, otherwise an !!ERROR!!

PCXP_GetAllLists

Purpose:

This function retrieves a list of existing lists for the user's Constant Contact account.

Dependencies:

A previous call PCXP_Authenticate

Parameters:

None

Return Values:

A time stamp on success, otherwise an !!ERROR!!

PCXP_GetFirstList

Purpose:

This function retrieves the ID of the first list after a call to PCXP_GetAllLists

Dependencies:

A previous call PCXP_GetAllLists

Parameters:

None

Return Values:

A list ID on success, otherwise an !!ERROR!!

PCXP_GetContactsFromList

Purpose:

This function retrieves all of the contacts for the currently selected list.

Dependencies:

A previous call to PCXP_GetFirstList or PCXP_GetNextList

Parameters:

None

Return Values:

A time stamp on success, otherwise an !!ERROR!!

PCXP_GetNextList

Purpose:

This function retrieves the next list.

Dependencies:

A previous call to PCXP_GetAllLists and PCXP_GetFirstList

Parameters:

None

Return Values:

A list ID on success, otherwise an !!ERROR!!

PCXP_GetListField(FieldName)

Purpose:

This function returns data from the current list for the associated FieldName.

Dependencies:

A previous call to PCXP_NewList OR to PCXP_GetAllLists, PCXP_GetFirstList, PCXP_GetNextList

Parameters:

Parameter Name	Purpose	Values	Default Value
FieldName	Identifies what data to retrieve	see "Available Field Names" section in this document	

Return Values:

The field value on success, otherwise an !!ERROR!!

PCXP_NewList(Name)

Purpose:

This function creates a new list with the provided name.

Dependencies:

A previous call PCXP_Authenticate

Parameters:

Parameter Name	Purpose	Values	Default Value
Name	The desired name for the new list to be created.		

Return Values:

0 on success, otherwise an !!ERROR!!

Notes & Examples:

Must make a subsequent call to PCXP_SaveList in order for the new list to be created on the Constant Contact server.

PCXP_SaveList

Purpose:

This function saves a newly created list or any changes that have been made to an existing list.

Dependencies:

A previous call to PCXP_NewList, PCXP_GetFirstList, or PCXP_GetNextList

Parameters:

None

Return Values:

A list ID on success, otherwise an !!ERROR!!

PCXP_SetListField(FieldName : Value)

Purpose:

This function sets a value in a list for the corresponding key.

Dependencies:

A previous call to PCXP_NewList, PCXP_GetFirstList or PCXP_GetNextList.

Parameters:

Parameter Name	Purpose	Values	Default Value
FieldName	The field name for the value to be set.	see "Available Field Names" section in this document	
Value	The value to be assigned to the associated key.		

Return Values:

0 on success, otherwise an !!ERROR!!

III. Available Constant Contact Fields

Constant Contact has various available fields when pushing and pulling contacts or lists. In order to set fields, you will call the PCXP_SetContactField or PCXP_SetListField functions. In order to get Constant Contact fields you must call the PCXP_GetContactField or PCXP_GetListField functions.

All available fields are listed below. If there is a missing field that you would like to access, then please email a request to support@productivecomputing.com

Available Contact Fields			
Module	Method	Field Name	Acceptable Values/Notes
Contact	Get/Set	Addr1	address line 1
Contact	Get/Set	Addr2	address line 2
Contact	Get/Set	Addr3	address line 3
Contact	Get/Set	City	
Contact	Get/Set	CompanyName	
Contact	Get	Confirmed	
Contact	Get/Set	ContactList	Any number (1 - n) corresponding to an existing contact list
Contact	Get/Set	CountryCode	text
Contact	Get/Set	CountryName	text
Contact	Get/Set	CustomField1	
Contact	Get/Set	CustomField2	
Contact	Get/Set	CustomField3	
Contact	Get/Set	CustomField4	
Contact	Get/Set	CustomField5	
Contact	Get/Set	CustomField6	
Contact	Get/Set	CustomField7	
Contact	Get/Set	CustomField8	
Contact	Get/Set	CustomField9	
Contact	Get/Set	CustomField10	
Contact	Get/Set	CustomField11	
Contact	Get/Set	CustomField12	
Contact	Get/Set	CustomField13	
Contact	Get/Set	CustomField14	
Contact	Get/Set	CustomField15	
Contact	Get/Set	EmailAddress	valid email address
Contact	Get/Set	EmailType	text
Contact	Get/Set	FirstName	
Contact	Get/Set	HomePhone	
Contact	Get	InsertTime	timestamp from Constant Contact
Contact	Get/Set	JobTitle	
Contact	Get/Set	LastName	
Contact	Get	LastUpdateTime	timestamp from Constant Contact

Module	Method	Field Name	Acceptable Values/Notes
Contact	Get/Set	MiddleName	
Contact	Get/Set	Name	First and Last Name
Contact	Get/Set	Note	
Contact	Get/Set	PostalCode	also known as zip code
Contact	Get/Set	StateCode	text
Contact	Get/Set	StateName	text
Contact	Get/Set	Status	Active, Unconfirmed, Removed, or Do Not Mail
Contact	Get/Set	SubPostalCode	text
Contact	Get/Set	WorkPhone	

Available List Fields

Module	Method	Field Name	Acceptable Values/Notes
List	Get	ContactCount	
List	Get	Default	
List	Get/Set	Name	The name of the list. String of up to 255 characters.
List	Get/Set	OptInDefault	The values are "true" or "false." If the list is set to "true," then contacts who sign up or change their settings may add themselves to this list.
List	Get	ShortName	System-calculated display name for the list for use where space is limited.
List	Get	SortOrder	An integer representing the relative position of the list. For example, a list with the sort order of 25 will be shown after a list with the sort order of 15.

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!