



## Functions Guide

# Table of Contents

- I. Introduction.....3**
  - Description.....3
  - Intended Audience.....3
  - Successful Integration Practices.....3
- II. Function Descriptions.....4**
  - 1) Registration Specific Functions.....4
    - PCAB\_Register( ServerName ; ServerPort ; ServerPage ; licenseID ).....4
    - PCAB\_Version( Type ).....4
    - PCAB\_GetOperatingMode.....4
  - 2) Record Specific Functions.....5
    - PCAB\_Open( Address Book ID ).....5
    - PCAB\_Delete( Address Book ID ).....5
    - PCAB\_New( Type ).....5
    - PCAB\_Save.....6
    - PCAB\_GetFirstParentID.....6
    - PCAB\_GetNextParentID.....6
    - PCAB\_CompareProperty( Property; Value; optUID; optKey ).....7
  - 3) Group Specific Functions.....8
    - PCAB\_GetFirstMemberID.....8
    - PCAB\_GetNextMemberID.....8
    - PCAB\_AddPerson( Address Book ID ).....8
    - PCAB\_RemovePerson( Address Book ID ).....9
    - PCAB\_AddSubgroup( Address Book ID ).....9
    - PCAB\_RemoveSubgroup( Address Book ID ).....9
    - PCAB\_GetFirstSubgroupID.....10
    - PCAB\_GetNextSubgroupID.....10
  - 4) Single Valued Property Functions (Group and Contact Records).....11
    - PCAB\_SetValueForProperty( Property; Value ).....11
    - PCAB\_DeleteValueForProperty( Property ).....11
    - PCAB\_GetValueForProperty( Property ).....11
  - 5) Multi-Valued Property Functions (Contact Records Only).....12
    - PCAB\_AddMV( Property; Value; Label ).....12
    - PCAB\_AddAddress( Label; Street; City; State; Zip; Country; CountryCode ).....12
    - PCAB\_GetLabelForUID( Property; UID ).....13
    - PCAB\_SetLabelForUID( Property; UID; Label ).....13
    - PCAB\_GetValueForUID( Property; UID; optKey ).....13
    - PCAB\_SetValueForUID( Property; UID; Value; optKey ).....14
    - PCAB\_DeleteValueForUID( Property; UID; optKey ).....14
    - PCAB\_OpenFirstMVProperty( Property ).....14
    - PCAB\_OpenNextMVProperty.....15
  - 6) Search Functions.....16
    - PCAB\_Search( Type; Property; Label; Key; Value; Comparison ).....16
    - PCAB\_OpenFirstRecord.....16
    - PCAB\_OpenNextRecord.....16
- III. Available Fields, Key Values and Operators.....17**
  - Table 1: Fields for Single Valued Properties.....17
  - Table 2: Fields for Multi-Valued Properties.....17
  - Table 3: Available Key Values.....18
  - Table 4: Available Comparison Operators.....18
- IV. Contact Us.....19**

## I. Introduction

### Description

The Address Book Manipulator plug-in offers functions that support a bi-directional data exchange between FileMaker® and Apple® Address Book. With this plug-in FileMaker users are able to 'push' new records into Address Book, 'pull' records from Address Book, and modify existing records in Address Book. These operations are accomplished by using FileMaker function calls from within FileMaker calculations. These calculations are generally determined from within FileMaker 'SetField' or 'If' script steps.

### 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 Guides
- 3) Look at our FileMaker Demo and video tutorials  
Demo and video tutorials can be found here: <http://www.addressbookmanipulator.com>
- 4) Familiarize yourself with Apple's Address Book

## II. Function Descriptions

This section describes the functions that are available with the Address Book Manipulator plug-in.

### 1) Registration Specific Functions

#### PCAB\_Register( ServerName ; ServerPort ; ServerPage ; licenseID )

##### **Description:**

Registers the plug-in with Productive Computing's registration servers. Must call Register function once before using the plug-in.

##### **Parameters:**

ServerName = "licensing.productivecomputing.com"

ServerPort = "80"

ServerPage = "/PCIReg/pcireg.asp"

LicenseID = "your license ID" or "DEMO-AB2"

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

0 for success, anything other than 0 is an error

#### PCAB\_Version( Type )

##### **Description:**

Used to identify the product and product version installed on a machine.

##### **Parameters:**

type = "Short" or "Long" . The default version string returned is "short". The short version string includes only the version number of the plug-in. For example: "2.0.3.2". The long includes the plug-in name and version number. For example: "Address Book Manipulator 2.0.3.2"

##### **Returns:**

Product name and/or version, otherwise an error.

#### PCAB\_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

##### **Returns:**

"DEMO" for demo mode, "LIVE" for live mode, or "UNREGISTERED" for unregistered mode

## 2) Record Specific Functions

### PCAB\_Open( Address Book ID )

**Description:**

The PCAB\_Open function opens an Address Book record for reading and modifying. The desired record is indicated by the Address Book ID passed as a parameter. Use this function to open an Address Book record directly.

**Parameters:**

Address Book ID = the Address Book unique identifier for the desired record.

**Returns:**

0 for success otherwise an error

### PCAB\_Delete( Address Book ID )

**Description:**

This function deletes an Address Book record from the Address Book. This function should be used with care as the delete action cannot be reversed. The record indicated by the parameter is deleted.

**Parameters:**

Address Book ID = the Address Book unique identifier for the desired record.

**Returns:**

0 for success otherwise an error

### PCAB\_New( Type )

**Description:**

The function creates a new record of the desired type and adds it to the Address Book. When this function returns the new record is available for modification. Any modifications will not be committed until the PCAB\_Save function is called.

**Parameters:**

Type = the type of record to create. This is either 'Contact' or 'Group'

**Returns:**

The Address Book unique identifier ( Address Book ID) for the new record, or an error.

## **PCAB\_Save**

### **Description:**

This function commits to the Address Book any changes to the currently open record.

### **Parameters:**

None

### **Returns:**

0 for success otherwise an error

## **PCAB\_GetFirstParentID**

### **Description:**

This function and the PCAB\_GetNextParentID functions are used to iterate through the ABGroup record IDs of which the currently open record is a member. Call this function to retrieve the Address Book ID of the first group to which this record belongs.

### **Parameters:**

None

### **Returns:**

An Address Book ID or an error

## **PCAB\_GetNextParentID**

### **Description:**

This function and the PCAB\_GetFirstParentID functions are used to iterate through the ABGroup record IDs of which the currently open record is a member. Call this function to retrieve the Address Book ID of the next group to which this record belongs.

### **Parameters:**

None

### **Returns:**

An Address Book ID, 'END', or an error

## **PCAB CompareProperty( Property; Value; optUID; optKey )**

### **Description:**

This function compares 'Value' to a value stored in Address Book. The value stored in Address Book to be compared is identified by 'Property', 'optUID' and, 'optKey'.

### **Parameters:**

Property =	the Address Book property containing the value to be compared
Value =	the value that is to be compared to the Address Book property value
optUID =	optional - the universal identifier for the Address Book property
optKey =	optional - in the case of an Address Book address this is the key to identify the portion of the address. Acceptable values are 'Street', 'City', 'State', 'Zip', 'Country', and 'Country Code'

### **Returns:**

0 if the values match, -1 if they do not match

### 3) Group Specific Functions

#### PCAB\_GetFirstMemberID

**Description:**

This and the PCAB\_GetNextMemberID functions are used to iterate through the ABPerson record IDs which are members of the currently open Group record. Call this function to retrieve the Address Book ID of the first ABPerson record which is a member of the currently active Group record.

**Parameters:**

None

**Returns:**

An Address Book ID, 'END', or an error

#### PCAB\_GetNextMemberID

**Description:**

This and the PCAB\_GetFirstMemberID functions are used to iterate through the ABPerson record IDs which are members of the currently open Group record. Call this function to retrieve the Address Book ID of the next ABPerson record which is a member of the currently active Group record. Requires a call to PCAB\_GetFirstMemberID first.

**Parameters:**

None

**Returns:**

An Address Book ID, 'END', or an error

#### PCAB\_AddPerson( Address Book ID )

**Description:**

This function adds an ABPerson record to the currently open group record.

**Parameters:**

Address Book ID = the Address Book unique Identifier for the desired record.

**Returns:**

0 for success otherwise an error

### **PCAB\_RemovePerson( Address Book ID )**

**Description:**

This function removes an ABPerson record from the currently open group record.

**Parameters:**

Address Book ID = the Address Book unique Identifier for the desired record.

**Returns:**

0 for success otherwise an error

### **PCAB\_AddSubgroup( Address Book ID )**

**Description:**

This function adds an ABGroup record as a subgroup to the currently open group record.

**Parameters:**

Address Book ID = the Address Book unique Identifier for the desired record.

**Returns:**

0 for success otherwise an error

### **PCAB\_RemoveSubgroup( Address Book ID )**

**Description:**

This function removes an ABGroup record as a subgroup from the currently open group record.

**Parameters:**

Address Book ID = the Address Book unique Identifier for the desired record.

**Returns:**

0 for success otherwise an error

## **PCAB\_GetFirstSubgroupID**

### **Description:**

This and the PCAB\_GetNextSubgroupID function are used to iterate through the ABGroup record IDs which are subgroups of the currently open Group record. Call this function to retrieve the Address Book ID of the first ABGroup record which is a subgroup of the currently active Group record.

### **Parameters:**

None

### **Returns:**

An Address Book ID, 'END', or an error

## **PCAB\_GetNextSubgroupID**

### **Description:**

This and the PCAB\_GetFirstSubgroupID functions are used to iterate through the ABGroup record IDs which are subgroups of the currently open Group record. Call this function to retrieve the Address Book ID of the next ABGroup record which is a subgroup of the currently active Group record. Requires a call to PCAB\_GetFirstSubgroupID first.

### **Parameters:**

None

### **Returns:**

An Address Book ID, 'END', or an error

#### 4) Single Valued Property Functions (Group and Contact Records)

##### PCAB\_SetValueForProperty( Property; Value )

**Description:**

This function sets a property value stored in the Address Book to that of 'Value'

**Parameters:**

Property = the Address Book property that accepts the value

Value = the value that to be stored in the property

**Returns:**

0 for success otherwise an error

##### PCAB\_DeleteValueForProperty( Property )

**Description:**

This function deletes a property value stored in the Address Book.

**Parameters:**

Property = the Address Book property that holds the value

**Returns:**

0 for success otherwise an error

##### PCAB\_GetValueForProperty( Property )

**Description:**

This function retrieves a property value stored in the Address Book to that of 'Value'

**Parameters:**

Property = the Address Book property that contains the value

**Returns:**

The value stored in the property otherwise an error

## 5) Multi-Valued Property Functions (Contact Records Only)

### PCAB AddMV( Property; Value; Label )

#### **Description:**

This function adds a new value to the multi-valued property identified by the 'property' parameter.

#### **Parameters:**

Property = the Address Book property that accepts the value

Value = the value that to be stored in the property

Label = the label for the property ('Home' , 'Work', etc )

#### **Returns:**

Either the unique identifier for the property entry or an error is returned

### PCAB AddAddress( Label; Street; City; State; Zip; Country; CountryCode )

#### **Description:**

This function adds a new value to the multi-valued property identified by the 'property' parameter.

#### **Parameters:**

Label = the label for the address ('Home' , 'Work', etc )

Street = the street portion of the address

City = the city portion of the address

State = the state portion of the address

Zip = the zip portion of the address

Country = the country portion of the address

CountryCode = the country code portion of the address. This value affects how Address Book prints and displays the address

#### **Returns:**

Either the unique identifier for the address entry or an error is returned

### **PCAB\_GetLabelForUID( Property; UID )**

#### **Description:**

This function is used to retrieve the Label for a particular property value stored in the Address Book

#### **Parameters:**

Property = the Address Book property type for the property  
UID = the unique identifier for the property entry

#### **Returns:**

Either the unique identifier for the property entry or an error is returned

### **PCAB\_SetLabelForUID( Property; UID; Label )**

#### **Description:**

This function is used to change the label of a particular property entry.

#### **Parameters:**

Property = the Address Book property type for the property  
UID = the unique identifier for the property entry

#### **Returns:**

0 for success otherwise an error

### **PCAB\_GetValueForUID( Property; UID; optKey )**

#### **Description:**

This function is used to retrieve the value for a particular property value stored in the Address Book

#### **Parameters:**

Property = the Address Book property type for the property  
UID = the unique identifier for the property entry  
optKey = optional - in the case that the 'Property' is a address this is the key to identify the portion of the address. Acceptable values are 'Street', 'City', 'State', 'Zip', 'Country', and 'Country Code'.

#### **Returns:**

Either the value stored in the property entry or an error is returned

### **PCAB\_SetValueForUID( Property; UID; Value; optKey )**

#### **Description:**

This function is used to change the value stored in a particular property entry.

#### **Parameters:**

Property = the Address Book property type for the property  
UID = the unique identifier for the property entry  
optKey = optional - in the case that the 'Property' is a address this is the key to identify the portion of the address. Acceptable values are 'Street', 'City', 'State', 'Zip', 'Country', and 'Country Code'.

#### **Returns:**

0 for success otherwise an error

### **PCAB\_DeleteValueForUID( Property; UID; optKey )**

#### **Description:**

This function is used to delete the value stored in a particular property entry.

#### **Parameters:**

Property = the Address Book property type for the property  
UID = the unique identifier for the property entry  
optKey = optional - in the case that the 'Property' is a address this is the key to identify the portion of the address. Acceptable values are 'Street', 'City', 'State', 'Zip', 'Country', and 'Country Code'.

#### **Returns:**

0 for success otherwise an error

### **PCAB\_OpenFirstMVProperty( Property )**

#### **Description:**

This and the PCAB\_OpenNextMVProperty is used to iterate through the values stored in a multi-valued property. Use this function to open and access the first value stored in the property

#### **Parameters:**

Property = the Address Book property to be opened

#### **Returns:**

Either the UID for the property entry or an error is returned

## **PCAB\_OpenNextMVProperty**

### **Description:**

This and the PCAB\_OpenFirstMVProperty is used to iterate through the values stored in a multi-valued property. Use this function to open and access the next value stored in the property

### **Parameters:**

None

### **Returns:**

Either the UID for the property entry or END or an error

## 6) Search Functions

### PCAB\_Search( Type; Property; Label; Key; Value; Comparison )

#### **Description:**

The Search function locates Address Books that match the criteria parameters. The resulting found record set is retained in memory for access by the OpenFirstRecord and OpenNextRecord functions. For example, use this function to search for records modified after a specific time interval or search for contacts by group name, zip code or other specified criteria. This function proves quite useful when creating a bi-directional exchange as you can specify your specific search criteria.

#### **Parameters:**

Type =	This parameter indicates the type of records to search. This parameter is either 'Contact' or 'Group'
Property =	Limit searches to the properties of this property type
Label =	Limit searches to properties with this label only
Key =	Limit searches to properties with this key only. See Table 3 below.
Value =	This is the value to compare to the property values
Comparison =	This is the type of comparison to make between the property value and the 'Value' parameter. See table 4 below.

#### **Returns:**

The number of records in the found set or an error code

### PCAB\_OpenFirstRecord

#### **Description:**

This function opens the first record in the found set from the previous call to PCAB\_Search.

#### **Parameters:**

None

#### **Returns:**

The Address Book ID of the first record in the found set is returned or an error

### PCAB\_OpenNextRecord

#### **Description:**

This function opens the next record in the found set from the previous call to PCAB\_Search and PCAB\_OpenFirstRecord

#### **Parameters:**

None

#### **Returns:**

The Address Book ID of the next record in the found set, or 'END', or an error.

### III. Available Fields, Key Values and Operators

\*\* Fields ARE case sensitive.\*\*

**Table 1: Fields for Single Valued Properties**

Value	Description
<b>All Records</b>	
Modified Date	The date the record was last modified
Creation Date	The date the record was created
<b>Group Records</b>	
Name	The Name of the Group
<b>Person Records</b>	
Birthday	Birth date
Company Name	Company name
Department	Department name.
First Name	First name
Home Page	Home web page (10.2 and 10.3 only)
Job Title	Job title
Last Name	Last name
Maiden Name	Maiden name
Middle Name	Middle name
Name Prefix	Title
Name Suffix	Suffix
Nickname	Nickname
Note	Notes
Phonetic First	Phonetic representation of the first name
Phonetic Last	Phonetic representation of the last name
Phonetic Middle	Phonetic representation of the middle name
Person Flags	'ShowAsCompany or 'ShowAsPerson'

**Table 2: Fields for Multi-Valued Properties**

Value	Description
Address	Street addresses
Dates	Dates associated with a person
E-Mail	Email addresses
IM AOL	AOL instant messaging ID
IM ICQ	ICQ instant messaging ID
IM Jabber	Jabber instant messaging ID
IM MSN	MSN instant messaging ID
IM Yahoo	Yahoo instant messaging ID
Web Pages	Web pages
Phone	Generic phone number
Related Names	Names of people related to a person

**Table 3: Available Key Values**

Value	Description
Street	The street portion of the address
City	The city portion of the address
State	The state/province portion of the address
Zip	The Postal Code portion of the address
Country	The country portion of an address
Country Code	A two-letter ISO code for the country

**Table 4: Available Comparison Operators**

Value	Meaning
Equal	Search for elements that are equal to the value.
NotEqual	Search for elements that are not equal to the value.
NotEqualCaseInsensitive	Search for elements that are not equal to the value, ignoring case. (Available in Mac OS X v10.4 and later.)
LessThan	Search for elements that are less than the value.
LessThanOrEqual	Search for elements that are less than or equal to the value.
GreaterThan	Search for elements that are greater than the value.
GreaterThanOrEqual	Search for elements that are greater than or equal to the value.
EqualCaseInsensitive	Search for elements that are equal to the value, ignoring case.
ContainsSubString	Search for elements that contain the value.
ContainsSubStringCaseInsensitive	Search for elements that contain the value, ignoring case.
PrefixMatch	Search for elements that begin with the value.
PrefixMatchCaseInsensitive	Search for elements that begin with the value, ignoring case.
SuffixMatch	Search for elements that end with the value. (Available in Mac OS X v10.4 and later.)
SuffixMatchCaseInsensitive	Search for elements that end with the value, ignoring case. (Available in Mac OS X v10.4 and later.)
BitsInBitFieldMatch	Search for elements that match the bits in ABPersonFlags. (Available in Mac OS X v10.3 and later.)
DoesNotContainSubString	Search for elements that do not contain the value. (Available in Mac OS X v10.4 and later.)
DoesNotContainSubStringCaseInsensitive	Search for elements that do not contain the value, ignoring case. (Available in Mac OS X v10.4 and later.)
WithinIntervalAroundToday	Search for elements that are within a time interval (in seconds) forward or backward from today. (Available in Mac OS X v10.4 and later.)
WithinIntervalAroundTodayYearless	Search for elements that are within a time interval (in seconds) forward or backward from this day in any year. (Available in Mac OS X v10.4 and later.)
NotWithinIntervalAroundToday	Search for elements that are not within a time interval (in seconds) forward or backward from today. (Available in Mac OS X v10.4 and later.)
NotWithinIntervalAroundTodayYearless	Search for elements that are not within a time interval (in seconds) forward or backward from this day in any year. (Available in Mac OS X v10.4 and later.)
WithinIntervalFromToday	Search for elements that are within a time interval (in seconds) forward from today. (Available in Mac OS X v10.4 and later.)
WithinIntervalFromTodayYearless	Search for elements that are within a time interval (in seconds) forward from this day in any year. (Available in Mac OS X v10.4 and later.)
NotWithinIntervalFromToday	Search for elements that are not within a time interval (in seconds) forward from today. (Available in Mac OS X v10.4 and later.)
NotWithinIntervalFromTodayYearless	Search for elements that are not within a time interval (in seconds) forward from this day in any year. (Available in Mac OS X v10.4 and later.)

## 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!