



Productive  
Computing

# Address Book **Manipulator**



## Functions Guide

Revised June 26, 2013

# Table of Contents

<b>I. INTRODUCTION .....</b>	<b>4</b>
Description .....	4
Intended Audience .....	4
Successful Integration Practices .....	4
<b>II. FUNCTION DESCRIPTIONS .....</b>	<b>5</b>
1) Registration and Other Functions .....	5
PCAB_GetLastError( optType ) .....	5
PCAB_GetOperatingMode .....	5
PCAB_Register( ServerName ; ServerPort ; ServerPage ; LicenseID ) .....	6
PCAB_Version( Type ) .....	6
2) Record Specific Functions .....	7
PCAB_New( RecordType ; optName ) .....	7
PCAB_Open( UID ) .....	7
PCAB_Save( optType ) .....	7
PCAB_Delete( UID ) .....	8
PCAB_CompareProperty( Property; Value; optUID; optKey ) .....	8
PCAB_GetFirstParentID .....	8
PCAB_GetNextParentID .....	9
PCAB_GetImage .....	9
PCAB_SetImage( BinaryData ) .....	9
PCAB_Display( UID ) .....	10
3) Group Specific Functions .....	11
PCAB_AddPerson( UID ) .....	11
PCAB_RemovePerson( UID ) .....	11
PCAB_AddSubgroup( UID ) .....	11
PCAB_RemoveSubgroup( UID ) .....	12
PCAB_GetFirstMemberID .....	12
PCAB_GetNextMemberID .....	12
PCAB_GetFirstSubgroupID .....	13
PCAB_GetNextSubgroupID .....	13
PCAB_GetListOfValues( RecordType ; GroupOrMember ; FieldName ) .....	14
PCAB_ClearAll( GroupUID ; Whom ; ExcludeMe ) .....	14
4) Single Valued Property Functions (Group and Contact Records) .....	15
PCAB_SetValueForProperty( Property; Value ; optType ) .....	15
PCAB_DeleteValueForProperty( Property ; optType ) .....	15
PCAB_GetValueForProperty( Property ; optType ) .....	15
5) Multi-Valued Property Functions .....	16
PCAB_AddMV( Property; Value; Label ) .....	16
PCAB_AddAddress( Label; Street; City; State; Zip; Country; CountryCode ) .....	16
PCAB_GetLabelForUID( Property ; UID ) .....	17
PCAB_SetLabelForUID( Property; UID; Label ) .....	17
PCAB_GetValueForUID( Property; UID; optKey ) .....	17
PCAB_SetValueForUID( Property; UID; Value; optKey ) .....	18
PCAB_DeleteValueForUID( Property; UID; optKey ) .....	18
PCAB_OpenFirstMVProperty( Property ; optType) .....	19
PCAB_OpenNextMVProperty .....	19
6) Search Functions .....	20
PCAB_Search( Type; Property; Label; Key; Value; Comparison ; optExpandConstrain) .....	20
PCAB_OpenFirstRecord( optType ) .....	20
PCAB_OpenNextRecord( optType ) .....	21
7) Custom Field Related Functions .....	22
PCAB_AddCustomField( RecordType ; FieldName ) .....	22
PCAB_GetCustomField( RecordType ; FieldName ) .....	22
PCAB_SetCustomField( RecordType ; FieldName ; Value ) .....	22
<b>III. AVAILABLE PROPERTIES .....</b>	<b>23</b>
Single Valued Properties .....	23
Multi-Valued Properties .....	23
<b>IV. AVAILABLE KEY VALUES AND COMPARISON OPERATORS .....</b>	<b>24</b>
Available Key Values .....	24
Available Comparison Operators .....	25



## I. Introduction

### Description

The Address Book Manipulator plug-in offers functions that support a bi-directional data exchange between FileMaker® and Apple® Address Book or Contacts. With this plug-in FileMaker users are able to add, edit and delete records into the address book, pull records from the address book, and search for specific records in the 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) Download a demo: <http://www.productivecomputing.com/plugins/address-book-manipulator-id-39/>
- 4) Watch video tutorials: <http://www.productivecomputing.com/video>
- 5) Familiarize yourself with Apple’s Address Book/Contacts

## II. Function Descriptions

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

### 1) Registration and Other Functions

#### **PCAB\_GetLastError( optType )**

##### **Description:**

This function returns a description of the last error encountered. The parameter specified causes the function to return either a numeric or textual description.

##### **Parameters:**

optType = "Number" or "Text"

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

The numeric or textual description of the last error encountered.

#### **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

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

"DEMO" for demo mode, "LIVE" for live mode, "UNREGISTERED" for unregistered mode, "EXPIRED" for expired mode. Licensed plug-ins will be running in "LIVE" mode.

## **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.php"  
LicenseID = "your license ID" or "DEMO-AB3"

### **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: "3.0.0.2". The long version string includes the plug-in name and version number. For example: "PC Address Book Manipulator 3.0.0.2"

### **Return Values:**

Product version and/or name, or !!ERROR!!

## 2) Record Specific Functions

### **PCAB\_New( RecordType ; optName )**

#### **Description:**

The function creates a new record of the desired type and adds it to the Address Book. Any new records will not be committed until the PCAB\_Save function is called.

#### **Parameters:**

RecordType =	"Person" or "Group" depending on the type of record to create. "Contact" is also an acceptable value for backwards compatibility with the previous plug-in version.
optName =	When a new group record is being created this value will contain the name of the new group record. This field is NOT optional for the group record type.

#### **Return Values:**

The Address Book unique identifier (AddressBook UID) for the new record or !!ERROR!!

### **PCAB\_Open( UID )**

#### **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:**

UID =	the Address Book unique identifier for the desired record.
-------	--

#### **Return Values:**

0 for success or !!ERROR!!

### **PCAB\_Save( optType )**

#### **Description:**

This function commits any changes to the currently open record to the Address Book. Since a group and person record can both be open at the same time, they will both be saved.

#### **Parameters:**

optType = "group" operates on the currently opened group record. "person" operates on the currently opened person record. If empty, the operation is performed on the last record opened/created.

#### **Return Values:**

0 for success or !!ERROR!!

## **PCAB\_Delete( UID )**

### **Description:**

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

### **Parameters:**

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

### **Return Values:**

0 for success or !!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"

### **Return Values:**

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

### **Return Values:**

An Address Book UID or !!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

### **Return Values:**

An Address Book UID, END or !!ERROR!!

## **PCAB\_GetImage**

### **Description:**

This function retrieves the binary image data for the currently opened person record. The returned value must be stored in a FileMaker Container field.

### **Parameters:**

None

### **Return Values:**

The image data or !!ERROR!! or blank if no image is present

## **PCAB\_SetImage( BinaryData )**

### **Description:**

This function sets the image for the currently opened person record.

### **Parameters:**

BinaryData =        The image. Pass the contents of a FileMaker container field.

### **Return Values:**

0 for success or !!ERROR!!

## **PCAB\_Display( UID )**

### **Description:**

This function displays the provided Contact UID in Address Book.

### **Parameters:**

UID =                      the UID of a Contact record in Address Book.

### **Note:**

This function will display the Contact record only if the Address Book application is running.

### **Return Values:**

0 for success or !!ERROR!!

### 3) Group Specific Functions

#### **PCAB\_AddPerson( UID )**

##### **Description:**

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

##### **Parameters:**

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

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

0 for success or !!ERROR!!

#### **PCAB\_RemovePerson( UID )**

##### **Description:**

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

##### **Parameters:**

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

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

0 for success or !!ERROR!!

#### **PCAB\_AddSubgroup( UID )**

##### **Description:**

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

##### **Parameters:**

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

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

0 for success or !!ERROR!!

### **PCAB\_RemoveSubgroup( UID )**

#### **Description:**

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

#### **Parameters:**

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

#### **Return Values:**

0 for success or !!ERROR!!

### **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

#### **Return Values:**

An Address Book ID, END or !!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

#### **Return Values:**

An Address Book ID, END or !!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

### **Return Values:**

An AddressBook UID, END or !!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

### **Return Values:**

An AddressBook UID, END or !!ERROR!!

## **PCAB\_GetListOfValues( RecordType ; GroupOrMember ; FieldName )**

### **Description:**

This function is used to retrieve a value list of group members, or groups to which a record belongs. The value list contains either the names or uids of the groups or members for the currently opened person or group record.

### **Parameters:**

- RecordType = "group" or "person". Indicates which of the currently opened records to retrieve the value list from. If "person" is specified, then the "GroupOrMember" parameter must be "groups."
- GroupOrMember = "groups" or "members". Indicates the contents of the value list. "groups" will return the list of parent groups for the currently active record. "members" will return a list of persons that are members of the currently opened group record. When specifying "members" the RecordType parameter must be "group."
- FieldName = "uid" or "name". When retrieving members of groups and specifying "name," the full name including the name prefix, middle name, and name suffix will be included in the list. Otherwise the group names or record UIDs will be returned in the list.

### **Return Values:**

The desired value list or !!ERROR!!

## **PCAB\_ClearAll( GroupUID ; Whom ; ExcludeMe )**

### **Description:**

This function is used to delete person or group records from the AddressBook according to which group they belong. It may also be used to delete all records from the AddressBook.

### **Parameters:**

- UID = The UID for the group from which to delete select records for deletion. If left empty it is assumed that all records from the AddressBook are desired to be deleted.
- Whom = empty or "groups" or "members". Indicates which record types to remove. "groups" will cause only subgroups (or groups if the UID is omitted) to be deleted. "members" will cause only person records to be deleted. Leaving the parameter empty will cause all records to be deleted.
- ExcludeMe = True or False. Indicates that the record marked as "me" in the AddressBook will or will not be deleted. True indicates that the record will not be deleted. False indicates that it will be deleted.

### **Return Values:**

0 for success or !!ERROR!!

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

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

###### **Description:**

This function sets a property value stored in the Address Book to that of "Value" for an opened record. Note that "Value" must contain a value. To empty or remove a property value use the PCAB\_DeleteValueForProperty function.

###### **Parameters:**

Property = the Address Book property that accepts the value  
Value = the value that to be stored in the property. The value must NOT be empty.  
optType = "group" operates on the currently opened group record. "person" operates on the currently opened person record. If empty, the operation is performed on the last record opened/created.

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

0 for success or !!ERROR!!

##### **PCAB\_DeleteValueForProperty( Property ; optType )**

###### **Description:**

This function deletes a property value stored in the Address Book for an opened record.

###### **Parameters:**

Property = the Address Book property that holds the value  
optType = "group" operates on the currently opened group record. "person" operates on the currently opened person record. If empty, the operation is performed on the last record opened/created.

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

0 for success or !!ERROR!!

##### **PCAB\_GetValueForProperty( Property ; optType )**

###### **Description:**

This function retrieves a property value stored in the Address Book for a currently opened record.

###### **Parameters:**

Property = the Address Book property that contains the value  
optType = "group" operates on the currently opened group record. "person" operates on the currently opened person record. If empty, the operation is performed on the last record opened/created.

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

The value stored in the property or !!ERROR!!

## 5) Multi-Valued Property Functions

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

#### **Description:**

This function adds a new value to the multi-valued property identified by the "property" parameter for the currently opened person record.

#### **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 )

#### **Return Values:**

Either the unique identifier for the property entry or !!ERROR!!

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

#### **Description:**

This function adds a new address to the currently opened person record.

#### **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

#### **Return Values:**

Either the unique identifier for the address entry or !!ERROR!!



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

#### **Description:**

This function is used to retrieve the Label for a particular property value for the currently opened person record.

#### **Parameters:**

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

#### **Return Values:**

Either the unique identifier for the property entry or !!ERROR!!

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

#### **Description:**

This function is used to change the label of a particular property entry for the currently opened person record.

#### **Parameters:**

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

#### **Return Values:**

0 for success or !!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 for the currently opened person record.

#### **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."

#### **Return Values:**

Either the value stored in the property entry or !!ERROR!!

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

#### **Description:**

This function is used to change the value stored in a particular property entry for the currently opened person record. Note that the Value cannot contain an empty value. To remove a value for a particular property call the PCAB\_DeleteValueForUID function.

#### **Parameters:**

Property = the Address Book property type for the property

UID = the unique identifier for the property entry

Value= the value for the property. Must not be empty.

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."

#### **Return Values:**

0 for success or '!!ERROR!!'

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

#### **Description:**

This function is used to delete the value stored in a particular property entry for the currently opened person record.

#### **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."

#### **Return Values:**

0 for success or !!ERROR!!

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

### **Description:**

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

### **Parameters:**

Property =            the Address Book property to be opened

optType =            "group" operates on the currently opened group record. "person" operates on the currently opened person record. If empty, the operation is performed on the last record opened/created.

### **Return Values:**

Either the UID for the property entry or !!ERROR!!

## **PCAB\_OpenNextMVProperty**

### **Description:**

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

### **Parameters:**

None

### **Return Values:**

Either the UID for the property entry or END or !!ERROR!!

## 6) Search Functions

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

#### **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. The new optExpandConstrain parameter allows for the function to expand or constrain the last search results.

#### **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. Available Key Values found later in this guide.

Value = This is the value to compare to the property values

Comparison = This is the type of comparison to make between the "Property" and the "Value" parameters. Available Comparison Operators found later in this guide.

optExpandConstrain = An empty value indicates that a new search should be performed. "expand" basically 'ORs' the new search with the last search. The found set will be expanded to include the new search parameters. "constrain" basically 'ANDs' the new search with the last search. The last found set will be constrained to the new search parameters.

#### **Return Values:**

The number of records in the found set or !!ERROR!!

### **PCAB\_OpenFirstRecord( optType )**

#### **Description:**

This function opens the first record in the found set of person or group records depending on the optType parameter. If the optType parameter is empty, then the first record from the last call to PCAB\_Search is opened.

#### **Parameters:**

optType = "group" operates on the currently opened group record. "person" operates on the currently opened person record. It empty, the operation is performed on the last record opened/created.

#### **Return Values:**

An AddressBook UID or !!ERROR!!

## **PCAB\_OpenNextRecord( optType )**

### **Description:**

This function opens the next record in the found set of person or group records depending on the optType parameter. If the optType parameter is empty, then the next record from the last call to PCAB\_Search is opened.

### **Parameters:**

optType =                "group" operates on the currently opened group record. "person" operates on the currently opened person record. It empty, the operation is performed on the last record opened/created.

### **Return Values:**

An AddressBook UID or !!ERROR!!

## 7) Custom Field Related Functions

### **PCAB\_AddCustomField( RecordType ; FieldName )**

#### **Description:**

Adds a custom field to the Address Book records. The field is added to group or person records according to the RecordType parameter and is user defined. Note that custom fields may not be removed from Address Book records, are not visible in the Address Book user interface and are not synced with any other application (MobileMe, Mail, etc. ). This is a limitation imposed by Apple rather than the plug-in. Please take caution as this is a one way street. Any custom field created in Address Book cannot be deleted. In addition any custom field created with this function will be only of single valued string type.

#### **Parameters:**

RecordType = "group" or "person"

FieldName = The name for the new field.

#### **Return Values:**

0 for success or !!ERROR!!

### **PCAB\_GetCustomField( RecordType ; FieldName )**

#### **Description:**

This function retrieves the value stored in a custom field.

#### **Parameters:**

RecordType = "group" or "person"

FieldName = The name of the custom from which to retrieve the value.

#### **Return Values:**

The value in the field or !!ERROR!!

### **PCAB\_SetCustomField( RecordType ; FieldName ; Value )**

#### **Description:**

This function sets the value of a custom field in the currently opened group or person record.

#### **Parameters:**

RecordType = "group" or "person"

FieldName = The name of the custom field for which to set the value.

Value = The value to which the custom field will be set.

#### **Return Values:**

0 for success or !!ERROR!!

### III. Available Properties

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

#### Single Valued Properties

A single valued property can only have one value in Address Book. Use PCAB\_SetValueForProperty function to set these properties and PCAB\_GetValueForProperty function to get these 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
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
Person Flags	"ShowAsCompany" or "ShowAsPerson"
Phonetic First	Phonetic representation of the first name
Phonetic Last	Phonetic representation of the last name
Phonetic Middle	Phonetic representation of the middle name

#### Multi-Valued Properties

Multi valued properties can have more than one value. Use PCAB\_AddMV function to set these properties and PCAB\_GetLabelForUID and PCAB\_GetValueForUID to get the property label and property.

Value	Description
Address	Physical address
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

## IV. Available Key Values and Comparison Operators

### Available Key Values

Use PCAB\_AddAddress function to set an address with the values specified below. Use PCAB\_GetLabelForUID( "Address" ; "SomeUID" ) and PCAB\_GetValueForUID( "Address" ; "SomeUID" ; "SomeValueBelow" ) to get the address label and address. These key values are also used in the "Key" parameter of the PCAB\_Search function.

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.

ISO codes: [http://developer.apple.com/mac/library/documentation/UserExperience/Reference/AddressBook/Classes/ABPerson\\_Class/Reference/CountryCodes.html#//apple\\_ref/doc/uid/TP40003945-RH2](http://developer.apple.com/mac/library/documentation/UserExperience/Reference/AddressBook/Classes/ABPerson_Class/Reference/CountryCodes.html#//apple_ref/doc/uid/TP40003945-RH2)



## Available Comparison Operators

These comparison operators are used to indicate the type of comparison to make in the "Comparison" parameter of the PCAB\_Search function.

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.
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.
SuffixMatchCaseInsensitive	Search for elements that end with the value, ignoring case.
BitsInBitFieldMatch	Search for elements that match the bits in ABPersonFlags.
DoesNotContainSubString	Search for elements that do not contain the value.
DoesNotContainSubStringCaseInsensitive	Search for elements that do not contain the value, ignoring case.
WithinIntervalAroundToday	Search for elements that are within a time interval (in seconds) forward or backward from today.
WithinIntervalAroundTodayYearless	Search for elements that are within a time interval (in seconds) forward or backward from this day in any year.
NotWithinIntervalAroundToday	Search for elements that are not within a time interval (in seconds) forward or backward from today.
NotWithinIntervalAroundTodayYearless	Search for elements that are not within a time interval (in seconds) forward or backward from this day in any year.
WithinIntervalFromToday	Search for elements that are within a time interval (in seconds) forward from today.
WithinIntervalFromTodayYearless	Search for elements that are within a time interval (in seconds) forward from this day in any year.
NotWithinIntervalFromToday	Search for elements that are not within a time interval (in seconds) forward from today.
NotWithinIntervalFromTodayYearless	Search for elements that are not within a time interval (in seconds) forward from this day in any year.

## V. 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!