

MBS Xojo Addressbook Kit

Version 1.1, © 2019 by Christian Schmitz

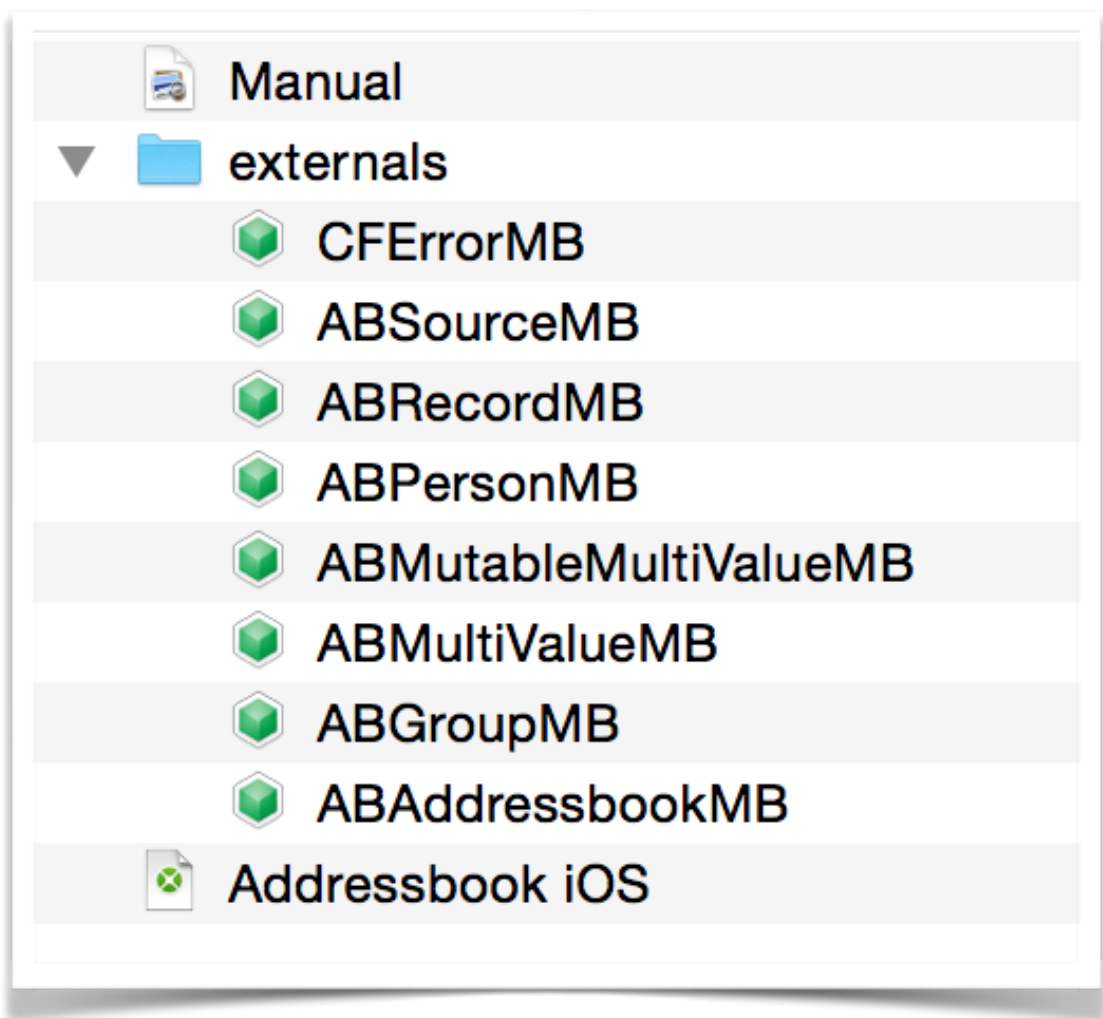
About the MBS Xojo Addressbook Kit	2
<i>Addressbook</i>	3
Interfaces	4
<i>ABAddressbookMB</i>	4
<i>ABGroupMB</i>	8
<i>ABMultiValueMB</i>	9
<i>ABMutableMultiValueMB</i>	10
<i>ABPersonMB</i>	11
<i>ABRecordMB</i>	14
<i>ABSourceMB</i>	16
<i>CFErrorMB</i>	17
Version History	18
Installation	19
Requirements	20
License	21
Contact	22

About the MBS Xojo Addressbook Kit

The MBS Xojo Addressbook Kit provides you with a few useful classes provide address book access to your Xojo iOS application.

The Kit contains:

- Wrapper for iOS Addressbook framework
 - CFErrorMB class
 - ABSourceMB class
 - ABRecordMB class
 - ABPersonMB class
 - ABMutableMultiValueMB class
 - ABMultiValueMB class
 - ABGroupMB class
 - ABAddressbookMB class
- Sample application



Addressbook

The Addressbook Kit contains a complete wrapper for Addressbook framework:

Features

- Classes for Source, Record, Person, Group, MultiValue and Addressbook itself.
- Convenience properties on Record, Group and Person for properties
- Example project showing how to add contact
- Example project showing how to browse contacts

Wrapper Features

- For new Xojo framework
- Using exception handling to track error
- Using enums for saver constant passing
- Test code included
- All classes with MB postfix to avoid name conflicts.
- All module definitions are protected to avoid conflicts
- Inline documentation
- Full Source code, no encryption
- Works for 32bit and 64bit targets.

Possible future features

- Add views provided by AddressbookUI framework

Interfaces

ABAddressbookMB

Class ABAddressbookMB

ComputedProperty Shared AuthorizationStatus **As Integer**

Sub Get()

Returns the authorization status of your app for accessing address book data.

To check the app's access to contact data. Based upon the access, the app could display or hide its UI elements that would access any AddressBook API.

kABAuthorizationStatusNotDetermined

The user has not yet made a choice regarding whether this app can access the data class.

kABAuthorizationStatusRestricted

This application is not authorized to access the data class. The user cannot change this application's status, possibly due to active restrictions such as parental controls being in place.

kABAuthorizationStatusDenied

The user explicitly denied access to the data class for this application.

kABAuthorizationStatusAuthorized

This application is authorized to access the data class.

ComputedProperty DefaultSource **As** ABSourceMB

Sub Get()

Returns the default source.

ComputedProperty GroupCount **As Integer**

Sub Get()

Returns the number of groups in an address book.

ComputedProperty HasUnsavedChanges **As Boolean**

Sub Get()

Indicates whether an address book has changes that have not been saved to the Address Book

database.

Returns true when addressBook contains unsaved changes, false otherwise.

ComputedProperty PersonCount **As Integer**

Sub Get()

Returns the number of person records in an address book.

ComputedProperty Shared RequestingAccessAddressbook **As** ABAddressbookMB

Sub Set()

Sub Get()

Const kABAuthorizationStatusAuthorized = 3

Const kABAuthorizationStatusDenied = 2

Const kABAuthorizationStatusNotDetermined = 0

Const kABAuthorizationStatusRestricted = 1

Const kABOperationNotPermittedByStoreError = 0

Const kABOperationNotPermittedByUserError = 1

Function AddRecord(record **as** ABRecordMB, **byref** error **as** CLErrorMB) **As Boolean**

Adds a record to an address book.

Returns true when successful, false otherwise.

Function AllGroups() **As** ABGroupMB()

Returns an array with all the groups in an address book.

Function AllGroups(source **as** ABSourceMB) **As** ABGroupMB()

Returns an array of all groups from a particular source.

Function AllPeople() **As** ABPersonMB()

Returns all the person records in an address book.

Function AllPeople(source **as** ABSourceMB) **As** ABPersonMB()

Returns an array of all person records from a particular source.

Function AllPeople(source **as** ABSourceMB, Ordering **as integer**) **As** ABPersonMB()

Returns an array of all person records in the address book, sorted with the specified order.

Function AllPeople(Name **as** Text) **As** ABPersonMB()

Performs a prefix search on the composite names of people in an address book and returns an array of persons that match the search criteria.

Function AllSources() **As** ABSourceMB()

Returns an array of all sources in the address book.

Shared Function AutoToCFType(a **as** auto) **As** Integer

will need CFRelease

Shared Function CFDataToMemoryBlock(Ref **as** integer) **As** MemoryBlock

Shared Function CFTYPEToAuto(Ref **as** integer) **As** auto

will need CFRelease

Sub Constructor(**byref** error **as** CFErrorMB)

Call ABAddressBookCreate to create an instance of AddressBook. The ABAddressBookRef will initially not have access to contact data. The app must then call ABAddressBookRequestAccessWithCompletion to request this access.

If access to contact data is already restricted or denied, this will fail returning a zero ABAddressBookRef with error kABOperationNotPermittedByUserError.

Sub Constructor(H **as** Integer, Retain **as** Boolean)

Create from Handle

Sub Destructor()

Function GroupWithRecordID(GroupID **as** integer) **As** ABGroupMB

Returns the group with a given record ID.

Shared Function LocalizedLabel(Label **as** Text) **As** Text

Some labels are special keys representing generic labels. Use this function to obtain a localized string for a label to display to a user.

Shared Function LocalizedPropertyName(propertyID **as** integer) **As** Text

Returns the localized name of a person property

Shared Function MemoryBlockToCFData(Data **as** MemoryBlock) **As** Integer

Function PersonWithRecordID(PersonID **as** integer) **As** ABPersonMB

Returns the person record with a given record ID.

Function RemoveRecord(record **as** ABRecordMB, **byref** error **as** CFErrorMB) **As** Boolean

Removes a record from an address book.

Returns true when successful, false otherwise.

Sub RequestAccess()

Requests access to address book data from the user.

Use this function to request access to address book data. This call will not block while the user is being asked for access, allowing your app to continue running. Until access has been granted, any address book references your app has will not contain any data, and any attempt to modify data will fail with an error type of kABOperationNotPermittedByUserError. The user is only asked for permission the first time you request access. Later calls use the permission granted by the user.

later the AccessCompletion event is raised to inform you about status

Sub Revert()

Revert discards all cached properties and any unsaved changes so that records and properties retrieved subsequently will reflect any changes made by external ABAddressBook instances. If records were deleted externally, the corresponding ABRecord instances will return nil for all properties. If necessary you can identify these deleted records by checking whether the class accessor (eg. PersonWithRecordID()) returns the record.

Function Save(**byref** error **as** CFErrorMB) **As** Boolean

Saves any unsaved changes to the Address Book database.

Function SourceForRecordID(RecordID **as** Integer) **As** ABSourceMB

Returns the source record with the given record ID.

Shared Sub Test()

Note "About"

The ABAddressBook opaque type (whose instances are known as address books) provides a programming interface to the Address Book—a centralized database used by multiple applications to store personal information about people. The Address Book database also supports the notion of a “group” containing one or more persons. People may belong to multiple groups, and groups may also belong to other groups.

The ABAddressBook opaque type provides functions for creating references to

the Address Book database, saving changes, discarding changes, and registering for changes made externally (by other threads or processes) to the database.

Property Shared ABAddressBookErrorDomain **As Text**
// Error domain under which Address Book errors are grouped.

Property Shared kABGroupNameProperty **As Integer**
// Street address - kABMultiDictionaryPropertyType

Property Shared kABHomeLabel **As Text**

Property Shared kABOtherLabel **As Text**

Property Shared kABPersonAddressCityKey **As Text**

Property Shared kABPersonAddressCountryCodeKey **As Text**

Property Shared kABPersonAddressCountryKey **As Text**

Property Shared kABPersonAddressProperty **As Integer**
// Street address - kABMultiDictionaryPropertyType

Property Shared kABPersonAddressStateKey **As Text**

Property Shared kABPersonAddressStreetKey **As Text**

Property Shared kABPersonAddressZIPKey **As Text**

Property Shared kABPersonAlternateBirthdayCalendarIdentifierKey **As Text**

Property Shared kABPersonAlternateBirthdayDayKey **As Text**

Property Shared kABPersonAlternateBirthdayEraKey **As Text**

Property Shared kABPersonAlternateBirthdayIsLeapMonthKey **As Text**

Property Shared kABPersonAlternateBirthdayMonthKey **As Text**

Property Shared kABPersonAlternateBirthdayProperty **As Integer**
// Alternate birthday
// kABDictionaryPropertyType

Property Shared kABPersonAlternateBirthdayYearKey **As Text**

Property Shared kABPersonAnniversaryLabel **As Text**

Property Shared kABPersonAssistantLabel **As Text**

Property Shared kABPersonBirthdayProperty **As Integer**
// Birthday associated with this person - kABDateTimePropertyType

Property Shared kABPersonBrotherLabel **As Text**

Property Shared kABPersonChildLabel **As Text**

Property Shared kABPersonCreationDateProperty **As Integer**
// Creation Date (when first saved)

Property Shared kABPersonDateProperty **As Integer**
// Dates associated with this person - kABMultiDatePropertyType

Property Shared kABPersonDepartmentProperty **As Integer**
// Department name - kABStringPropertyType

Property Shared kABPersonEmailProperty **As Integer**
// Email(s) - kABMultiStringPropertyType

Property Shared kABPersonFatherLabel **As Text**

Property Shared kABPersonFirstNamePhoneticProperty **As Integer**
// First name Phonetic - kABStringPropertyType

Property Shared kABPersonFirstNameProperty **As Integer**
// First name - kABStringPropertyType

Property Shared kABPersonFriendLabel **As Text**

Property Shared kABPersonHomePageLabel **As Text**

Property Shared kABPersonInstantMessageProperty **As Integer**
// Instant Messaging - kABMultiDictionaryPropertyType

Property Shared kABPersonInstantMessageServiceAIM **As Text**

Property Shared kABPersonInstantMessageServiceFacebook **As Text**

Property Shared kABPersonInstantMessageServiceGaduGadu **As Text**

Property Shared kABPersonInstantMessageServiceGoogleTalk **As Text**

Property Shared kABPersonInstantMessageServiceICQ **As Text**

Property Shared kABPersonInstantMessageServiceJabber **As Text**

Property Shared kABPersonInstantMessageServiceKey **As Text**

Property Shared kABPersonInstantMessageServiceMSN **As Text**

Property Shared kABPersonInstantMessageServiceQQ **As Text**

Property Shared kABPersonInstantMessageServiceSkype **As Text**

Property Shared kABPersonInstantMessageServiceYahoo **As Text**

Property Shared kABPersonInstantMessageUsernameKey **As Text**

Property Shared kABPersonJobTitleProperty **As Integer**
// Job Title - kABStringPropertyType

Property Shared kABPersonKindOrganization **As Integer**

Property Shared kABPersonKindPerson **As Integer**

Property Shared kABPersonKindProperty **As Integer**

```

// Person/Organization - kABIntegerPropertyType
Property Shared kABPersonLastNamePhoneticProperty As Integer
// Last name Phonetic - kABStringPropertyType
Property Shared kABPersonLastNameProperty As Integer
// Last name - kABStringPropertyType
Property Shared kABPersonManagerLabel As Text
Property Shared kABPersonMiddleNamePhoneticProperty As Integer
// Middle name Phonetic - kABStringPropertyType
Property Shared kABPersonMiddleNameProperty As Integer
// Middle name - kABStringPropertyType
Property Shared kABPersonModificationDateProperty As Integer
// Last saved date
Property Shared kABPersonMotherLabel As Text
Property Shared kABPersonNicknameProperty As Integer
// Nickname - kABStringPropertyType
Property Shared kABPersonNoteProperty As Integer
// Note - kABStringPropertyType
Property Shared kABPersonOrganizationProperty As Integer
// Company name - kABStringPropertyType
Property Shared kABPersonParentLabel As Text
Property Shared kABPersonPartnerLabel As Text
Property Shared kABPersonPhoneHomeFAXLabel As Text
Property Shared kABPersonPhonePhoneLabel As Text
Property Shared kABPersonPhoneMainLabel As Text
Property Shared kABPersonPhoneMobileLabel As Text
Property Shared kABPersonPhoneOtherFAXLabel As Text
Property Shared kABPersonPhonePagerLabel As Text
Property Shared kABPersonPhoneProperty As Integer
// Generic phone number - kABMultiStringPropertyType
Property Shared kABPersonPhoneWorkFAXLabel As Text
Property Shared kABPersonPrefixProperty As Integer
// Prefix ("Sir" "Duke" "General") - kABStringPropertyType
Property Shared kABPersonRelatedNamesProperty As Integer
// Names - kABMultiStringPropertyType
Property Shared kABPersonSisterLabel As Text
Property Shared kABPersonSocialProfileProperty As Integer
// kABMultiDictionaryPropertyType // the following properties are optional
Property Shared kABPersonSocialProfileServiceFacebook As Text
Property Shared kABPersonSocialProfileServiceFlickr As Text
Property Shared kABPersonSocialProfileServiceGameCenter As Text
Property Shared kABPersonSocialProfileServiceKey As Text
Property Shared kABPersonSocialProfileServiceLinkedIn As Text
Property Shared kABPersonSocialProfileServiceMyspace As Text
Property Shared kABPersonSocialProfileServiceSinaWeibo As Text
Property Shared kABPersonSocialProfileServiceTwitter As Text
Property Shared kABPersonSocialProfileURLKey As Text
Property Shared kABPersonSocialProfileUserIdentifierKey As Text
Property Shared kABPersonSocialProfileUsernameKey As Text
Property Shared kABPersonSpouseLabel As Text
Property Shared kABPersonSuffixProperty As Integer
// Suffix ("Jr." "Sr." "III") - kABStringPropertyType
Property Shared kABPersonURLProperty As Integer
// URL - kABMultiStringPropertyType
Property Shared kABSourceNameProperty As Integer
Property Shared kABSourceTypeProperty As Integer
Property Shared kABWorkLabel As Text

```

End Class

ABGroupMB

Class ABGroupMB **Inherits** ABRecordMB

ComputedProperty Name **As** Text

Sub Set()

convenience property for name

Sub Get()

convenience property for name

ComputedProperty Source **As** ABSourceMB

Sub Get()

Returns the source that the group is from.

Function AddMember(Person **as** ABPersonMB, **byref** error **as** CLErrorMB) **As Boolean**

Adds a person record to a group.

Before a person record can be added to a group, the record must be saved to the Address Book

database.

Function AllMembers() **As** ABPersonMB()

Returns the records in a group.

Function AllMembers(SortOrdering **as integer**) **As** ABPersonMB()

Returns the records in a group, using a sort ordering.

Sub Constructor()

creates a new group in the default source

Sub Constructor(source **as** ABSourceMB)

creates a new person in the default source

Sub Constructor(H **as Integer**, Retain **as Boolean**)

Create from Handle

Function RemoveMember(Member **as** ABRecordMB, **byref** error **as** CLErrorMB) **As Boolean**

Removes a record from a group.

End Class

ABMultiValueMB

Class ABMultiValueMB

ComputedProperty Count **As Integer**

Sub Get()

Returns the number of values in a multivalue property.

ComputedProperty PropertyType **As Integer**

Sub Get()

Returns the type of the values contained in a multivalue property.

The type of the values contained in multiValue or kABInvalidPropertyType when multiValue is empty or contains values of different types.

Const kABMultiValueInvalidIdentifier = -1

Function AllValues() **As** auto()

Returns an array with the values in a multivalue property.

Sub Constructor(H **as Integer**, Retain **as Boolean**)

Sub Destructor()

Cleanup

Function FirstIndexOfValue(value **as** auto) **As Integer**

Returns the first location of a value in a multivalue property.

Location of value within multiValue, -1 if value is not present in multiValue.

Function IdentifierAtIndex(Index **as integer**) **As Integer**

Returns the identifier of a value in a multivalue property.

Function IndexForIdentifier(Identifier **as integer**) **As Integer**

Returns the location (within a multivalue property) of a value with a given identifier.

Function LabelAtIndex(Index **as integer**) **As** Text

Returns the label for a value in a multivalue property.

Function MutableCopy() **As** ABMutableMultiValueMB

returns mutable copy

Function ValueAtIndex(Index **as integer**) **As** auto

Returns the value at a particular location within a multivalue property.

Note "About"

The ABMultiValue opaque type (whose objects are known as multivalues) implements a property that can have multiple values. All the contained values must be of the same type. Each value has a unique identifier and a nonunique label, which may be one of the provided labels or one defined by the user.

For example, if a multivalue property is used to store phone numbers, there may be multiple work phone numbers. These have the same label, but different unique identifiers.

Multivalue properties are immutable. To use mutable multivalue properties, see ABMutableMultiValue class.

Property Handle **As Integer**

End Class

ABMutableMultiValueMB

Class ABMutableMultiValueMB **Inherits** ABMultiValueMB

Function AddValueAndLabel(value **as** auto, label **as** text, **byref** identifier **as Integer**) **As boolean**

Adds a value and its corresponding label to a multivalue property.

This function performs no type checking. It allows the addition of values whose type does not match the type declared for multiValue.

Sub Constructor(Other **as** ABMultiValueMB)

creates a copy

Sub Constructor(Type **as Integer**)

Returns a new, empty, mutable multivalue property.

type: The type of the values that the new property will contain.

Sub Constructor(H **as Integer**, Retain **as Boolean**)

Create from Handle

Function InsertValueAndLabelAtIndex(value **as** auto, label **as** text, index **as Integer**, **byref** identifier **as Integer**) **As boolean**

Inserts a value and a label into a multivalue property.

Sub LabelAtIndex(Index **as integer**, **assigns** label **as** text)

for assignment: m.LabelAtIndex(index) = "new label"

Function RemoveValueAndLabelAtIndex(Index **as integer**) **As boolean**

Removes a value from a multivalue property.

Function ReplaceLabelAtIndex(Index **as integer**, label **as** text) **As boolean**

Replaces a label in a multivalue property with another label.

Function ReplaceValueAtIndex(value **as** auto, index **as Integer**) **As boolean**

Replaces a value in a multivalue property with another value.

Note "About"

The ABMutableMultiValue opaque type (whose objects are known as mutable multivalues) implements a mutable multivalue property—that is, a property whose value set can be modified. This opaque type extends the capabilities of ABMultiValue.

End Class

ABPersonMB

Class ABPersonMB **Inherits** ABRecordMB

ComputedProperty Addresses **As** ABMultiValueMB
Sub Set()
convenience property for Address
Sub Get()
convenience property for Address

ComputedProperty Birthday **As** date
Sub Set()
convenience property for Birthday
Sub Get()
convenience property for Birthday

ComputedProperty Dates **As** ABMultiValueMB
Sub Set()
convenience property for Dates
Sub Get()
convenience property for Dates

ComputedProperty Department **As** Text
Sub Set()
convenience property for Department
Sub Get()
convenience property for Department

ComputedProperty Emails **As** ABMultiValueMB
Sub Set()
convenience property for Email
Sub Get()
convenience property for Email

ComputedProperty FirstName **As** Text
Sub Set()
convenience property for FirstName
Sub Get()
convenience property for FirstName

ComputedProperty FirstNamePhonetic **As** Text
Sub Set()
convenience property for FirstNamePhonetic
Sub Get()
convenience property for FirstNamePhonetic

ComputedProperty HasImageData **As Boolean**
Sub Get()
Indicates whether a person has a picture.

ComputedProperty Image **As** UIImage
Sub Set()
convenience function to set
Sub Get()

ComputedProperty InstantMessages **As** ABMultiValueMB
Sub Set()
convenience property for InstantMessage
Sub Get()
convenience property for InstantMessage

ComputedProperty JobTitle **As** Text
Sub Set()
convenience property for JobTitle
Sub Get()
convenience property for JobTitle

ComputedProperty Kind **As Integer**
Sub Set()
convenience property for Kind setting
Sub Get()
convenience property for Kind

ComputedProperty LastName **As** Text
Sub Set()
convenience property for LastName
Sub Get()
convenience property for LastName

ComputedProperty LastNamePhonetic **As** Text

```

    Sub Set()
        convenience property for LastNamePhonetic
    Sub Get()
        convenience property for LastNamePhonetic
ComputedProperty MiddleName As Text
    Sub Set()
        convenience property for MiddleName
    Sub Get()
        convenience property for MiddleName
ComputedProperty MiddleNamePhonetic As Text
    Sub Set()
        convenience property for LastNamePhonetic
    Sub Get()
        convenience property for LastNamePhonetic
ComputedProperty Nickname As Text
    Sub Set()
        convenience property for Nickname
    Sub Get()
        convenience property for Nickname
ComputedProperty Note As Text
    Sub Set()
        convenience property for Note
    Sub Get()
        convenience property for Note
ComputedProperty Organization As Text
    Sub Set()
        convenience property for Organization
    Sub Get()
        convenience property for Organization
ComputedProperty Phones As ABMultiValueMB
    Sub Set()
        convenience property for phone
    Sub Get()
        convenience property for phone
ComputedProperty RelatedNames As ABMultiValueMB
    Sub Set()
        convenience property for RelatedNames
    Sub Get()
        convenience property for RelatedNames
ComputedProperty SocialProfiles As ABMultiValueMB
    Sub Set()
        convenience property for SocialProfile
    Sub Get()
        convenience property for SocialProfile
ComputedProperty Source As ABSourceMB
    Sub Get()
        Returns the source that the person record is from.
ComputedProperty Thumbnail As iOSImage
    Sub Get()
ComputedProperty URLs As ABMultiValueMB
    Sub Set()
        convenience property for URLs
    Sub Get()
        convenience property for URLs
ComputedProperty VCardRepresentation As MemoryBlock
    Sub Get()
        convenience function to get VCard for this person
ComputedProperty isOrganization As Boolean
    Sub Get()
        convenience property for Kind
ComputedProperty isPerson As Boolean
    Sub Get()
        convenience property for Kind
Const kABPersonCompositeNameFormatFirstNameFirst = 0
Const kABPersonCompositeNameFormatLastNameFirst = 1
Const kABPersonImageFormatOriginalSize = 2

```

Const kABPersonImageFormatThumbnail = 0
Const kABPersonSortByFirstName = 0
Const kABPersonSortByLastName = 1
Const kCFCompareEqualTo = 0
Const kCFCompareGreaterThan = 0
Const kCFCompareLessThan = -1
Function AllLinkedPeople() **As** ABPersonMB()
Returns an array of all the linked people, including the person passed in.
If the person is not linked, returns an array with the person passed in.
Shared Function CFArrayToPeople(ArrayRef **as** integer) **As** ABPersonMB()
Function ComparePeopleByName(Other **as** ABPersonMB, SortOrdering **as** Integer) **As** Integer
Indicates how two person records get sorted.
Sub Constructor()
Creates a new person record in the default source.
To add the newly created person record to the Address Book database, use
ABAddressBookMB.AddRecord.
Sub Constructor(source **as** ABSourceMB)
Creates a new person record in a particular source.
Sub Constructor(H **as** Integer, Retain **as** Boolean)
Create from Handle
Shared Function CreatePeopleInSourceWithVCardRepresentation(data **as** MemoryBlock) **As** ABPersonMB()
Creates an array of people from a vcard representation. Source is optional. The people
will be created in the default source.
Function ImageData() **As** MemoryBlock
Returns the picture for a person record.
Function ImageData(Format **as** Integer) **As** MemoryBlock
Returns the picture for a person record in the given format.
Format: kABPersonImageFormatThumbnail or kABPersonImageFormatOriginalSize
Function RemoveImageData(**byref** error **as** CFErrorMB) **As** Boolean
Removes a person's picture.
Function SetImageData(data **as** memoryblock, **byref** error **as** CFErrorMB) **As** Boolean
Sets a person's picture.
Shared Function SortOrdering() **As** integer
Returns the user's sort-ordering preference for lists of persons.
Shared Function TypeOfProperty(propertyID **as** integer) **As** integer
Returns the type of a person property.
Shared Function VCardRepresentationWithPeople(people() **as** ABPersonMB) **As** MemoryBlock
Creates a vCard representation of the people passed in.
Note "About"
Person records are represented by instances of the ABRecordRef opaque type,
whose record type is kABPersonType. Person records store information about
a person, such as first name, last name, address, email, and phone number.

Person records don't necessarily have to be stored in the Address Book
database. You can use person records as a way to group contact information
in memory and present it to the user through, for example, a person view
controller (ABPersonViewController).

End Class

ABRecordMB

Class ABRecordMB

ComputedProperty CompositeName **As** Text

Sub Get()

Returns an appropriate, human-friendly name for the record.

ComputedProperty CompositeNameDelimiter **As** Text

Sub Get()

The delimiter that should be used to separate name components.

If the record has no first, middle and last name, or if the record is 0, then a default delimiter is returned.

ComputedProperty CreationDate **As** date

Sub Set()

convenience property for CreationDate

Sub Get()

convenience property for CreationDate

ComputedProperty ID **As** Integer

Sub Get()

Returns the unique ID of a record.

ComputedProperty IsGroup **As** Boolean

Sub Get()

ComputedProperty IsPerson **As** Boolean

Sub Get()

ComputedProperty ModificationDate **As** date

Sub Set()

convenience property for ModificationDate

Sub Get()

convenience property for ModificationDate

ComputedProperty Type **As** Integer

Sub Get()

Returns the type of a record.

can be kABGroupType, kABPersonType or kABSourceType

ComputedProperty isSource **As** Boolean

Sub Get()

Const kABDateTimePropertyType = 4

Const kABDictionaryPropertyType = 5

Const kABGroupType = 1

Const kABIntegerPropertyType = 2

Const kABInvalidPropertyType = 0

Const kABMultiDateTimePropertyType = 260

Const kABMultiDictionaryPropertyType = 261

Const kABMultiIntegerPropertyType = 258

Const kABMultiRealPropertyType = 259

Const kABMultiStringPropertyType = 257

Const kABMultiValueMask = 256

Const kABPersonType = 0

Const kABPropertyInvalidID = -1

Const kABRealPropertyType = 3

Const kABRecordInvalidID = -1

Const kABSourceType = 2

Const kABStringPropertyType = 1

Function CompositeNameFormat() **As** integer

Returns the person-name display format to use for the given record.

The appropriate "First, Last" or "Last, First" order for the contact depending on the detected locale of the contact's name.

If the record has no first, middle and last name, or if the record is nil, then the Contacts Display Order preference is returned.

Sub Destructor()

Function RemoveValue(ABPropertyID **as** Integer, **byref** error **as** CFErrorMB) **As** Boolean

Removes the value of a record property.

Function SetValue(ABPropertyID **as** Integer, value **as** auto, **byref** error **as** CFErrorMB) **As** Boolean

Sets the value of a record property.

Function Value(propertyID **as** integer) **As** auto

Returns the value of a record property.

Property Handle **As** Integer

End Class

ABSourceMB

```
Class ABSourceMB Inherits ABRecordMB
  ComputedProperty Name As Text
    Sub Set()
      convenience property for Source name
    Sub Get()
      convenience property for the name
  ComputedProperty Type As Integer
    Sub Set()
      convenience property for Source Type
    Sub Get()
      convenience property for the Type
  Const kABSourceTypeCardDAV = 4
  Const kABSourceTypeCardDAVSearch = &h01000004
  Const kABSourceTypeExchange = 1
  Const kABSourceTypeExchangeGAL = &h01000001
  Const kABSourceTypeLDAP = &h01000003
  Const kABSourceTypeLocal = 0
  Const kABSourceTypeMobileMe = 2
  Const kABSourceTypeSearchableMask = &h01000000
  Sub Constructor(H as Integer, Retain as Boolean)
  Function CreatePeopleInSourceWithVCardRepresentation(data as MemoryBlock) As ABPersonMB()
    Creates an array of people from a vcard representation. Source is optional. The people
    will be created in the source given as the first argument, or the default source if 0.
  Note "About"
    Sources of contact information are represented by instances of the
    ABRecordRef opaque type, whose record type is kABSourceType. Examples of
    sources include the local address book database, CardDAV servers, and
    social networking sites.

    Not all sources support groups. Each record in the address book database
    can belong to only one source. Folders from Exchange accounts are
    represented in the address book database as sources.
```

End Class

CFErrorMB

```
Class CFErrorMB
  ComputedProperty Code As Integer
    Sub Get()
  ComputedProperty Description As Text
    Sub Get()
  ComputedProperty Domain As Text
    Sub Get()
  ComputedProperty FailureReason As Text
    Sub Get()
  ComputedProperty RecoverySuggestion As Text
    Sub Get()
  Sub Constructor(H As Integer, Retain As Boolean)
  Sub Destructor()
  Note "About"
    Wrapper for CFErrorRef to provide error details
End Class
```

Version History

Tip: If you want to update your existing code with new release, you'd best compare projects with Arbed (<http://www.tempel.org/Arbed>) and copy modifications to your project.

1.1, 31st July 2019

- Updated for Xojo 2019r1

1.0, first release, March 2015

Installation

To get your projects working with this Addressbook Kit, you need to follow a few steps.

Drop the folder „externals“ into your project and access all the common crypto or zip modules and classes. Or copy from existing example projects what you need.

Requirements

You need Xojo 2015r1 or newer.

We did not test with older versions, but you can if you need.

The encryption parts for desktop can use MBS Plugins to perform similar operations on Mac OS X, Windows and Linux.

The SQL connection used in the DatabaseConnector uses MBS Xojo SQL Plugin to connect to database.

License

Summary:

- You may use Addressbook Kit only with one licensed Xojo installation.
- You agree not to share the Addressbook Kit or use someone else's Addressbook Kit copy.

Christian Schmitz Software GmbH, of Nickenich Germany is the owner, developer and sole copyright holder of this product, which is licensed -not sold- to you on a non-exclusive basis.

You agree not to share your MBS Xojo Addressbook Kit with anyone.

You may transfer your license to another person only after receiving written authorization from Christian Schmitz Software GmbH and only if the recipient agrees to be bound by the terms of this agreement.

Christian Schmitz Software GmbH reserves the right to cancel the license key(s) of any user who Christian Schmitz Software GmbH determines is in violation of this agreement.

THE WARRANTIES IN THIS AGREEMENT REPLACE ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE IS PROVIDED "AS IS" AND Christian Schmitz Software GmbH DISCLAIMS AND EXCLUDES ALL OTHER WARRANTIES. IN NO EVENT WILL Christian Schmitz Software GmbH BE LIABLE FOR ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, EVEN IF WE HAVE KNOWLEDGE OF THE POTENTIAL LOSS OR DAMAGE.

If you are located in Germany this agreement is subject to the laws of Germany. If you are located outside Germany local law may apply. Some states do not allow the exclusion of warranties, so the above exclusion may not apply to you.

Christian Schmitz Software GmbH does not charge royalties or deployment fees for Xojo applications.

Access to updates is included for one year. After that time you can order an update or keep using the old version you have.

Contact

Christian Schmitz Software GmbH
Eckertshohl 22
56645 Nickenich
Germany

Email: support@monkeybreadsoftware.de

Phone: +49 26 32 95 89 55 (Office) or +49 17 58 36 37 10 (Mobile)