
ABPerson Class Reference

[Cocoa](#) > [Apple Applications](#)



2007-07-07



Apple Inc.
© 2007 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, Cocoa, Mac, Mac OS, and QuickTime are trademarks of Apple Inc., registered in the United States and other countries.

Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS

PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

ABPerson Class Reference 5

Overview	5
Tasks	6
Managing Properties	6
Managing Groups	7
Managing Images	7
Searching	7
Importing and Exporting VCard Formatted Files	7
Getting Identity Properties	7
Class Methods	7
addPropertiesAndTypes:	7
cancelLoadingImageDataForTag:	8
properties	8
removeProperties:	8
searchElementForProperty:label:key:value:comparison:	9
typeOfProperty:	10
Instance Methods	10
beginLoadingImageDataForClient:	10
identity	10
identityUniqueld	11
imageData	11
initWithVCardRepresentation:	11
parentGroups	12
setImageData:	12
vCardRepresentation	12
Constants	13

Document Revision History 21

Index 23

ABPerson Class Reference

Inherits from	ABRecord : NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/AddressBook.framework
Availability	Available in Mac OS X v10.2 and later.
Declared in	ABImageLoading.h ABPerson.h
Companion guides	Address Book Programming Guide for Mac OS X Identity Services Programming Guide
Related sample code	ABPresence AddressBookCocoa Birthdays

Overview

The ABPerson class encapsulates all information about a person in the Address Book database—an instance of ABPerson corresponds to a single person record in the database. The ABPerson class defines properties such as the person’s name, company, address, email addresses, and phone numbers.

Some of these properties have multiple values that are accessed via standard and user-defined labels. For example, a person may have a home, work, mobile, and fax phone numbers. Therefore, the phone attribute is defined as an ABMultiValue object containing NSString objects for each number. For example, you might get a person’s primary phone number as follows:

```
ABMultiValue *phones = [aPerson valueForKey:kABPhoneProperty];
id value = [phones valueAtIndex:[phones indexOfIdentifier:
    [phones primaryIdentifier]]];
```

See ABRecord for common methods to get and set properties. See the “Constants” (page 13) section for a list of all the properties, labels, and keys used to access fields in a person record. See ABMultiValue for more details on multi-value lists and how primary values work.

You can add your own properties to person records too using the [addPropertiesAndTypes:](#) (page 7) method—that is, attach additional program-defined data to each person record. Because the Address Book database is stored as a property list, these program-defined properties can be ignored by other applications. Note that the AddressBook database is accessed by multiple application and is not encrypted so your application should not store any sensitive information in the database like credit card numbers.

A person may also have an associated picture or image. The image is not actually stored in the Address Book database (a property list)—it's stored in a separate image file. You can set a person's image using the [setImageData:](#) (page 12) method, or get an image using the [imageData](#) (page 11) method. Use the `UIImage initWithData:` method to convert an `NSData` object, returned by the `imageData` method, to an `UIImage` object.

Image files may be local or remote. Local images are any images in `.../Library/Images/People` or images the user has set using the Address Book application. Remote images are images stored on the network. These images take time to download, so `ABPerson` provides an asynchronous API for fetching remote images.

Use the [beginLoadingImageDataForClient:](#) (page 10) method if an image file is not local and you want to perform an asynchronous fetch. You pass a client object that implements the `ABImageClient` protocol as an argument to this method. The [beginLoadingImageDataForClient:](#) (page 10) method will return an image tracking number. A `consumeImageData:forTag:` message is sent to your client object when the fetch is done. Implement this method to handle the new fetched image. Use the [cancelLoadingImageDataForTag:](#) (page 8) class method if for some reason you want to cancel an asynchronous fetch.

Person records may belong to multiple groups. Use the [parentGroups](#) (page 12) method to get the groups a person belongs to. See `ABGroup` for more information about groups.

You can also search for records matching a particular “query” you specify by creating an `ABSearchElement` object. Use the [searchElementForProperty:label:key:value:comparison:](#) (page 9) method to create an `ABSearchElement` object containing your query. Then use the `ABAddressBook recordsMatchingSearchElement:` method, passing the `ABSearchElement` as the argument, to query the database. See `ABSearchElement` for more methods that create compound queries.

Your application can also import and export persons in the vCard file format using the [initWithVCardRepresentation:](#) (page 11) and [vCardRepresentation](#) (page 12) methods.

The `ABPerson` class is “toll-free bridged” with its procedural C, opaque type, counterpart. This means that the `ABPersonRef` type is interchangeable in function or method calls with instances of the `ABPerson` class.

Tasks

Managing Properties

+ [addPropertiesAndTypes:](#) (page 7)

+ [removeProperties:](#) (page 8)

+ [properties](#) (page 8)

+ [typeOfProperty:](#) (page 10)

Managing Groups

- [parentGroups](#) (page 12)

Managing Images

- + [cancelLoadingImageDataForTag:](#) (page 8)
- [beginLoadingImageDataForClient:](#) (page 10)
- [imageData](#) (page 11)
- [setImageData:](#) (page 12)

Searching

- + [searchElementForProperty:label:key:value:comparison:](#) (page 9)

Importing and Exporting VCard Formatted Files

- [initWithVCardRepresentation:](#) (page 11)
- [vCardRepresentation](#) (page 12)

Getting Identity Properties

- [identityUniqueId](#) (page 11)
Returns the unique identifier for a person.
- [identity](#) (page 10)
Returns the CSIdentityRef object associated with a person.

Class Methods

addPropertiesAndTypes:

- + (NSInteger)addPropertiesAndTypes:(NSDictionary *)*properties*

Discussion

Adds the given properties to all the records of this type in the Address Book database, and returns the number of properties successfully added. In each dictionary entry, the key is a string with the property's name, and the value is a constant with the property's type. The property's name must be unique. You may want to use Java-style package names for your properties, for example, "org.dogclub.dogname" or "com.mycompany.customerID". The property type must be one of the constants described in "Constants" in ABRecord. Returns -1 if an error occurs.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABPerson.h

cancelLoadingImageDataForTag:

```
+ (void)cancelLoadingImageDataForTag:(NSInteger)tag
```

Discussion

Cancels an asynchronous fetch of the images for a given tag. The *tag* argument should have been returned from a previous call to the [beginLoadingImageDataForClient:](#) (page 10) method.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [beginLoadingImageDataForClient:](#) (page 10)
- `consumeImageData:forTag:` (ABImageClient)

Declared In

ABImageLoading.h

properties

```
+ (NSArray *)properties
```

Discussion

Returns an array of the names of all the properties for the ABPerson record in the Address Book database.

Availability

Available in Mac OS X v10.2 and later.

See Also

- + [typeOfProperty:](#) (page 10)

Declared In

ABPerson.h

removeProperties:

```
+ (NSInteger)removeProperties:(NSArray *)properties
```


Discussion

Removes the given properties from all the records of this type in the Address Book database, and returns the number of properties successfully removed. Returns -1 if an error occurs.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABPerson.h

searchElementForProperty:label:key:value:comparison:

```
+ (ABSearchElement *)searchElementForProperty:(NSString *)property label:(NSString *)label key:(NSString *)key value:(id)value comparison:(ABSearchComparison)comparison
```

Discussion

Returns a search element object that specifies a query for records of this type.

- *property* is the name of the property to search on, such as `kABAddressProperty` or `kABLastNameProperty`. It cannot be `nil`. For a full list of the properties, see “Constants” (page 13) and “Constants” in `ABRecord`.
- *label* is the label name for a multi-value list, such as `kABAddressHomeLabel`, `kABPhoneWorkLabel`, or a user-specified label, such as “Summer Home”. If the specified property does not have multiple values, pass `nil`. If the specified property does have multiple values, pass `nil` to search all the values. For a full list of label names, see “Constants” (page 13).
- *key* is the key name for a dictionary, such as `kABAddressCityKey` or `kABAddressStreetKey`. If the specified property is not a dictionary, pass `nil`. If the specified property is a dictionary, pass `nil` to search all keys. For a full list of key names, see “Constants” (page 13)
- *value* is what you’re searching for. If `nil`, then the only supported value for *comparison* is `kABEqual` or `kABNotEqual`.
- *comparison* specifies the type of comparison to perform and is an `ABSearchComparison`, such as `kABEqual` or `kABPrefixMatchCaseInsensitive`.

Availability

Available in Mac OS X v10.2 and later.

See Also

```
+ searchElementForProperty:label:key:value:comparison: (ABGroup)
+ searchElementForConjunction:children: (ABSearchElement)
- recordsMatchingSearchElement: (ABAddressBook)
```

Related Sample Code

AddressBookCocoa

Birthdays

Declared In

ABPerson.h

typeofProperty:

+ (ABPropertyType)typeofProperty:(NSString *)*property*

Discussion

Returns the type for a given property. If the property does not exist, this method returns `kABErrorInProperty`.

Availability

Available in Mac OS X v10.2 and later.

See Also

+ [properties](#) (page 8)

Declared In

ABPerson.h

Instance Methods

beginLoadingImageDataForClient:

- (NSInteger)beginLoadingImageDataForClient:(id <ABImageClient>)*client*

Discussion

Starts an asynchronous fetch for image data in all locations, and returns a non-zero tag for tracking. The *client* object should conform to the `ABImageClient` protocol. A `consumeImageData:forTag: message` is sent to *client* when the fetch is done. Use the [cancelLoadingImageDataForTag:](#) (page 8) method if you need to cancel an asynchronous fetch.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [imageData](#) (page 11)

Declared In

ABImageLoading.h

identity

Returns the `CSIdentityRef` object associated with a person.

- (CSIdentityRef)*identity*

Return Value

A `CSIdentityRef` object associated with the receiver or `NULL` if it doesn't exist.

See Also

- [identityUniqueId](#) (page 11)

- [beginSheetModalForWindow:modalDelegate:didEndSelector:contextInfo:](#)

- [runModalIdentityPicker](#)

identityUniqueId

Returns the unique identifier for a person.

- (NSString*)identityUniqueId

Return Value

A unique identifier for the receiver or `nil` if it doesn't exist.

See Also

- [identity](#) (page 10)
- `beginSheetModalForWindow:modalDelegate:didEndSelector:contextInfo:`
- `runModalIdentityPicker`

imageData

- (NSData *)imageData

Discussion

Returns data that contains a picture of this person. This method searches only the local file system and operates synchronously. To perform an asynchronous search or to search over a network, use [beginLoadingImageDataForClient:](#) (page 10).

The returned data is in a QuickTime-compatible format. To create an image from it, use the `NSImage` method `initWithData:`.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [beginLoadingImageDataForClient:](#) (page 10)
- [setImageData:](#) (page 12)

Declared In

ABImageLoading.h

initWithVCardRepresentation:

- (id)initWithVCardRepresentation:(NSData *)vCardData

Discussion

Returns an `ABPerson` instance initialized with the given data. If `vCardData` is `nil` or is not a valid vCard format, this method returns `nil`.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [vCardRepresentation](#) (page 12)

Declared In

ABPerson.h

parentGroups

- (NSArray *)parentGroups

Discussion

Returns an array of the ABGroups this person belongs to. If the person doesn't belong to any groups, this method returns an empty array.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABPerson.h

setImageData:

- (BOOL)setImageData:(NSData *)data

Discussion

Sets the image for this person to the given data. The *data* argument must be in a QuickTime-compatible format. Pass *nil* to specify that there is no image for this person.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [imageData](#) (page 11)
- [beginLoadingImageDataForClient:](#) (page 10)

Declared In

ABImageLoading.h

vCardRepresentation

- (NSData *)vCardRepresentation

Discussion

Returns the vCard representation of the person as a data object in vCard format.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [initWithVCardRepresentation:](#) (page 11)

Declared In

ABPerson.h

Constants

These are the properties that an ABPerson record contains by default. ABRecord describes additional properties that all records contain, in “Constants”. Note that some of these properties are not displayed in the Address Book application. Developers can add their own properties with the ABRecord method [addPropertiesAndTypes:](#) (page 7).

Note: The `kABHomePageProperty` property has been deprecated in Mac OS X version 10.4. Instead, use the `kABURLsProperty` multi-value property.

Constant	Description
<code>kABFirstNameProperty</code>	First name (NSString)
<code>kABLastNameProperty</code>	Last name (NSString)
<code>kABFirstNamePhoneticProperty</code>	Phonetic representation of the first name (NSString)
<code>kABLastNamePhoneticProperty</code>	Phonetic representation of the last name (NSString)
<code>kABBirthdayProperty</code>	Birth date (NSDate)
<code>kABOrganizationProperty</code>	Company name (NSString)
<code>kABJobTitleProperty</code>	Job title (NSString)
<code>kABHomePageProperty</code>	Home web page (NSString)
<code>kABURLsProperty</code>	Web pages (ABMultiValue containing NSString)
<code>kABCalendarURIsProperty</code>	Calendar URIs (ABMultiValue containing NSString)
<code>kABEmailProperty</code>	Email addresses (ABMultiValue containing NSString)
<code>kABAddressProperty</code>	Street addresses (ABMultiValue containing NSDictionary)
<code>kABPhoneProperty</code>	Generic phone number (ABMultiValue containing NSString)
<code>kABAIMInstantProperty</code>	AOL instant messaging ID (ABMultiValue containing NSString)
<code>kABJabberInstantProperty</code>	Jabber instant messaging ID (ABMultiValue containing NSString)
<code>kABMSNInstantProperty</code>	MSN instant messaging ID (ABMultiValue containing NSString)
<code>kABYahooInstantProperty</code>	Yahoo instant messaging ID (ABMultiValue containing NSString)
<code>kABICQInstantProperty</code>	ICQ instant messaging ID (ABMultiValue containing NSString)
<code>kABNoteProperty</code>	Notes. (NSString)
<code>kABMiddleNameProperty</code>	Middle name. This property isn't displayed in the Address Book application. (NSString)

Constant	Description
kABMiddleNamePhoneticProperty	Phonetic representation of the middle name. This property isn't displayed in the Address Book application. (NSString)
kABTitleProperty	Title, such as "Mr.," "Mrs.," "General," "Cardinal," or "Lord." This property isn't displayed in the Address Book application. (NSString)
kABSuffixProperty	Suffix, such as "Sr.," "Jr.," "III.," or "Esq." This property isn't displayed in the Address Book application. (NSString)
kABNicknameProperty	Nickname. This property isn't displayed in the Address Book application. (NSString)
kABMaidenNameProperty	Maiden name. This property isn't displayed in the Address Book application. (NSString)
kABOtherDatesProperty	Dates associated with a person (ABMultiDateProperty containing dates).
kABRelatedNamesProperty	Names of people related to a person (ABMultiStringProperty containing names).
kABDepartmentProperty	Department name.
kABPersonFlags	Property that specifies the name ordering and configuration of a record in the Address Book application.

The `ABPersonFlags` property is used to access the following settings:

Constant	Description
kABShowAsPerson	Record is displayed as a person.
kABShowAsCompany	Record is displayed as a company.
kABShowAsMask	Used in conjunction with <code>kABShowAsPerson</code> and <code>kABShowAsCompany</code> to determine record configuration.
kABDefaultNameOrdering	Default name ordering (whether a person's first name or last name is displayed first) the Address Book application.
kABFirstNameFirst	First name is displayed first in Address Book.
kABLastNameFirst	Last name is displayed first. in Address Book.
kABNameOrderingMask	Used in conjunction with <code>kABDefaultNameOrdering</code> , <code>kABFirstNameFirst</code> , and <code>kABLastNameFirst</code> to determine name ordering.

These are the default labels contained in the Address Book database for specifying different values in a multi-value list. Users can also add their own labels.

Constant	Description
kABEmailWorkLabel	Home email
kABEmailHomeLabel	Work email
kABAddressHomeLabel	Home address
kABAddressWorkLabel	Work address
kABPhoneWorkLabel	Work phone number
kABPhoneHomeLabel	Home phone number
kABPhoneMobileLabel	Cell phone number
kABPhoneMainLabel	Main phone number
kABPhoneHomeFAXLabel	Home FAX number
kABPhoneWorkFAXLabel	Work FAX number
kABPhonePagerLabel	Pager number
kABHomePageLabel	Web page URL
kABAIMWorkLabel	Work AOL instant messaging ID
kABAIMHomeLabel	Home AOL instant messaging ID
kABJabberWorkLabel	Work Jabber instant messaging ID
kABJabberHomeLabel	Home Jabber instant messaging ID
kABMSNWorkLabel	Work MSN instant messaging ID
kABMSNHomeLabel	Home MSN instant messaging ID
kABYahooWorkLabel	Work Yahoo instant messaging ID
kABYahooHomeLabel	Home Yahoo instant messaging ID
kABICQWorkLabel	Work ICQ instant messaging ID
kABICQHomeLabel	Home ICQ instant messaging ID
kABAnniversaryLabel	Anniversary date
kABMotherLabel	Mother
kABFatherLabel	Father
kABParentLabel	Parent
kABSisterLabel	Sister
kABBrotherLabel	Brother

Constant	Description
kABChildLabel	Child
kABFriendLabel	Friend
kABSpouseLabel	Spouse
kABPartnerLabel	Partner
kABAssistantLabel	Assistant
kABManagerLabel	Manager

Use these labels in searches to match all work, home, or other labels. For example, `kABWorkLabel` can be used in place of `kABEmailWorkLabel` or `kABAddressWorkLabel`.

Constant	Description
kABWorkLabel	All Work labels match this label
kABHomeLabel	All Home labels match this label
kABOtherLabel	All labels other than Home or Work labels match this label.

These are the keys contained in each address dictionary. Neither developers nor users can add more keys.

Constant	Description
kABAddressStreetKey	Street
kABAddressCityKey	City
kABAddressStateKey	State
kABAddressZIPKey	Zip
kABAddressCountryKey	Country
kABAddressCountryCodeKey	Country Code

The `kABAddressCountryCodeKey` must be one of the following ISO country codes.

Country Codes	Description
ae	United Arab Emirates
ar	Argentina
at	Austria
au	Australia

Country Codes	Description
ba	Bosnia and Herzegovina
be	Belgium
bh	Bahrain
br	Brazil
ca	Canada
ch	Switzerland
cn	China
cz	Czech Republic
de	Germany
dk	Denmark
eg	Egypt
es	Spain
fi	Finland
fr	France
gl	Greenland
gr	Greece
hk	Hong Kong
hr	Croatia
hu	Hungary
ie	Ireland
il	Israel
id	Indonesia
in	India
is	Iceland
it	Italy
jp	Japan
jo	Jordan

Country Codes	Description
kr	South Korea
kw	Kuwait
lb	Lebanon
lu	Luxembourg
mk	Macedonia
mx	Mexico
nl	Netherlands
no	Norway
nz	New Zealand
om	Oman
pl	Poland
pt	Portugal
qa	Qatar
ro	Romania
ru	Russian Federation
sa	Saudi Arabia
se	Sweden
sg	Singapore
si	Slovenia
sk	Slovakia
sy	Syrian Arab Republic
tr	Turkey
tw	Taiwan
ua	Ukraine
gb	United Kingdom
us	United States
ye	Yemen

Country Codes	Description
yu	Serbia and Montenegro
za	South Africa

Document Revision History

This table describes the changes to *ABPerson Class Reference*.

Date	Notes
2007-07-07	Updated for Mac OS X v10.5.
2006-05-23	Corrected typos.
2006-03-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

`addPropertiesAndTypes`: class method 7

B

`beginLoadingImageDataForClient`: instance method 10

C

`cancelLoadingImageDataForTag`: class method 8

I

`identity` instance method 10
`identityUniqueId` instance method 11
`imageData` instance method 11
`initWithVCardRepresentation`: instance method 11

K

`kABAddressCityKey` constant 16
`kABAddressCountryCodeKey` constant 16
`kABAddressCountryKey` constant 16
`kABAddressHomeLabel` constant 15
`kABAddressProperty` constant 13
`kABAddressStateKey` constant 16
`kABAddressStreetKey` constant 16
`kABAddressWorkLabel` constant 15
`kABAddressZIPKey` constant 16
`kABAIMHomeLabel` constant 15
`kABAIMInstantProperty` constant 13
`kABAIMWorkLabel` constant 15
`kABAnniversaryLabel` constant 15

`kABAssistantLabel` constant 16
`kABBirthdayProperty` constant 13
`kABBrotherLabel` constant 15
`kABCalendarURIsProperty` constant 13
`kABChildLabel` constant 16
`kABDefaultNameOrdering` constant 14
`kABDepartmentProperty` constant 14
`kABEmailHomeLabel` constant 15
`kABEmailProperty` constant 13
`kABEmailWorkLabel` constant 15
`kABFatherLabel` constant 15
`kABFirstNameFirst` constant 14
`kABFirstNamePhoneticProperty` constant 13
`kABFirstNameProperty` constant 13
`kABFriendLabel` constant 16
`kABHomeLabel` constant 16
`kABHomePageLabel` constant 15
`kABHomePageProperty` constant 13
`kABICQHomeLabel` constant 15
`kABICQInstantProperty` constant 13
`kABICQWorkLabel` constant 15
`kABJabberHomeLabel` constant 15
`kABJabberInstantProperty` constant 13
`kABJabberWorkLabel` constant 15
`kABJobTitleProperty` constant 13
`kABLastNameFirst` constant 14
`kABLastNamePhoneticProperty` constant 13
`kABLastNameProperty` constant 13
`kABMaidenNameProperty` constant 14
`kABManagerLabel` constant 16
`kABMiddleNamePhoneticProperty` constant 14
`kABMiddleNameProperty` constant 13
`kABMotherLabel` constant 15
`kABMSNHomeLabel` constant 15
`kABMSNInstantProperty` constant 13
`kABMSNWorkLabel` constant 15
`kABNameOrderingMask` constant 14
`kABNicknameProperty` constant 14
`kABNoteProperty` constant 13
`kABOrganizationProperty` constant 13
`kABOtherDatesProperty` constant 14
`kABOtherLabel` constant 16

[kABParentLabel](#) **constant** [15](#)
[kABPartnerLabel](#) **constant** [16](#)
[kABPersonFlags](#) **constant** [14](#)
[kABPhoneHomeFAXLabel](#) **constant** [15](#)
[kABPhoneHomeLabel](#) **constant** [15](#)
[kABPhoneMainLabel](#) **constant** [15](#)
[kABPhoneMobileLabel](#) **constant** [15](#)
[kABPhonePagerLabel](#) **constant** [15](#)
[kABPhoneProperty](#) **constant** [13](#)
[kABPhoneWorkFAXLabel](#) **constant** [15](#)
[kABPhoneWorkLabel](#) **constant** [15](#)
[kABRelatedNamesProperty](#) **constant** [14](#)
[kABShowAsCompany](#) **constant** [14](#)
[kABShowAsMask](#) **constant** [14](#)
[kABShowAsPerson](#) **constant** [14](#)
[kABSisterLabel](#) **constant** [15](#)
[kABSpouseLabel](#) **constant** [16](#)
[kABSuffixProperty](#) **constant** [14](#)
[kABTitleProperty](#) **constant** [14](#)
[kABURLsProperty](#) **constant** [13](#)
[kABWorkLabel](#) **constant** [16](#)
[kABYahooHomeLabel](#) **constant** [15](#)
[kABYahooInstantProperty](#) **constant** [13](#)
[kABYahooWorkLabel](#) **constant** [15](#)

P

[parentGroups](#) **instance method** [12](#)
[properties](#) **class method** [8](#)

R

[removeProperties:](#) **class method** [8](#)

S

[searchElementForProperty:label:key:value:](#)
[comparison:](#) **class method** [9](#)
[setImageData:](#) **instance method** [12](#)

T

[typeOfProperty:](#) **class method** [10](#)

V

[vCardRepresentation](#) **instance method** [12](#)