
Address Book Objective-C Framework Reference

Data Management: Contact Data



2007-07-08



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, iChat, Mac, Mac OS, Objective-C, QuickTime, and Xcode 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

Introduction **Introduction** 7

Part I **Classes** 9

Chapter 1 **ABAddressBook Class Reference** 11

Overview 11
Tasks 11
Class Methods 13
Instance Methods 14
Constants 21
Notifications 21

Chapter 2 **ABGroup Class Reference** 23

Overview 23
Tasks 23
Class Methods 24
Instance Methods 27

Chapter 3 **ABMultiValue Class Reference** 33

Overview 33
Tasks 33
Instance Methods 34

Chapter 4 **ABMutableMultiValue Class Reference** 39

Overview 39
Tasks 39
Instance Methods 40

Chapter 5 **ABPeoplePickerView Class Reference** 45

Overview 45
Tasks 45
Properties 47
Instance Methods 50
Constants 57
Notifications 58

Chapter 6 **ABPerson Class Reference 59**

Overview 59
Tasks 59
Class Methods 60
Instance Methods 63
Constants 66

Chapter 7 **ABRecord Class Reference 69**

Overview 69
Tasks 69
Instance Methods 70

Chapter 8 **ABSearchElement Class Reference 75**

Overview 75
Tasks 75
Class Methods 76
Instance Methods 76
Constants 77

Part II **Protocols 83**

Chapter 9 **ABActionDelegate Protocol Reference 85**

Overview 85
Tasks 85
Instance Methods 86

Chapter 10 **ABImageClient Protocol Reference 89**

Overview 89
Tasks 89
Instance Methods 89

Part III **Functions 91**

Chapter 11 **AddressBook Functions Reference 93**

Overview 93
Functions 93

Part IV Data Types 95

Chapter 12 Address Book Data Types Reference 97

Overview 97
Data Types 97

Part V Constants 101

Chapter 13 Address Book Objective-C Constants Reference 103

Overview 103
Constants 103

Document Revision History 117

Index 119

Introduction

Framework	/System/Library/Frameworks/AddressBook.framework
Header file directories	/System/Library/Frameworks/AddressBook.framework/Headers
Declared in	ABActions.h ABAddressBook.h ABGlobals.h ABGroup.h ABImageLoading.h ABMultiValue.h ABPeoplePickerView.h ABPerson.h ABRecord.h ABSearchElement.h ABTypeDefs.h

The Address Book is a centralized database for contact and other personal information for people. Users need to enter personal information about themselves and their friends only once, instead of entering it repeatedly whenever the information is used. Applications that support the Address Book framework share this contact information with other applications, include Apple's Mail and iChat.

Classes

ABAddressBook Class Reference

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

Overview

The `ABAddressBook` class provides a programming interface to the Address Book—a centralized database used by multiple applications to store contact and other 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 with some restrictions (for example, no circular references are allowed).

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

Tasks

Creating and Initializing an Address Book

- + [sharedAddressBook](#) (page 14)
Returns the unique shared instance of `ABAddressBook`.
- + [addressBook](#) (page 13)
Returns a new instance of `ABAddressBook`.

Retrieving Groups and People

- [groups](#) (page 16)
Returns an array of all the groups in the Address Book database.
- [people](#) (page 17)
Returns an array of all the people in the Address Book database.

Setting and Retrieving the Logged-in User's Record

- [me](#) (page 17)
Returns the `ABPerson` record that represents the logged-in user.
- [setMe:](#) (page 20)
Sets the record that represents the logged-in user.

Retrieving a Specific Record

- [recordForUniqueId:](#) (page 18)
Returns the person or group record that matches the given unique ID.

Retrieving the Class of a Record

- [recordClassFromUniqueId:](#) (page 18)
Returns the class name of the record that matches the given unique ID.

Retrieving a Formatted Address

- [formattedAddressFromDictionary:](#) (page 15)
Returns an attributed string containing the formatted address.

Retrieving Default Values

- [defaultCountryCode](#) (page 15)
Returns the default country code for records with unspecified country codes.
- [defaultNameOrdering](#) (page 15)
Returns the default name ordering defined by the user in the Address Book application's preferences.

Adding and Removing Records

- [addRecord:](#) (page 14)
Adds an `ABPerson` or `ABGroup` record to the Address Book database.
- [removeRecord:](#) (page 19)
Removes an `ABPerson` or `ABGroup` record from the Address Book database.

Searching

- [recordsMatchingSearchElement:](#) (page 18)
Returns an array of records that match the given search element, or returns an empty array if no records match the search element.

Saving and Detecting Changes

- [hasUnsavedChanges](#) (page 16)
Indicates whether an address book has changes that have not been saved to the Address Book database.
- [save](#) (page 19)
Saves all the changes made since the last save.
- [saveAndReturnError:](#) (page 20)
Saves all the changes made since the last save.

Class Methods

addressBook

Returns a new instance of `ABAddressBook`.

```
+ (ABAddressBook *)addressBook
```

Discussion

If you're just making one-off lookups and edits, the [sharedAddressBook](#) (page 14) method is probably more appropriate. If your code is executing a tight loop, the `addressBook` method can yield significant performance improvements when used with the `ABPerson initWithAddressBook:` (page 70) method. For example, you can replace code such as the following:

```
for (id item in someDataStructure) {
    ABPerson* person = [[ABPerson alloc] init];
    // Populate the person from the item
}
[[ABAddressBook sharedAddressBook] save];
```

With code like the following:

```
ABAddressBook* tempBook = [ABAddressBook addressBook];
for (id item in someDataStructure) {
    ABPerson* person = [[ABPerson alloc] initWithAddressBook:tempBook];
    // Populate the person from the item
}
[tempBook save];
```

Availability

Available in Mac OS X v10.5 and later.

See Also

+ `sharedAddressBook`

Declared In

ABAddressBook.h

sharedAddressBook

Returns the unique shared instance of `ABAddressBook`.

```
+ (ABAddressBook *)sharedAddressBook
```

Discussion

This method returns the address book for the logged-in user that is shared by every application. If you call this method more than once or try to create a new address book, you will get a pointer to the same shared address book.

If you're just making one-off lookups and edits, this is the appropriate method to use. If your code is executing a tight loop, using the `addressBook` (page 13) method with the `ABPerson initWithAddressBook:` (page 70) method can yield significant performance improvements.

Availability

Available in Mac OS X v10.2 and later.

See Also

+ `addressBook`

Related Sample Code

ABPresence

AddressBookCocoa

Birthdays

Declared In

ABAddressBook.h

Instance Methods

addRecord:

Adds an `ABPerson` or `ABGroup` record to the Address Book database.

```
- (BOOL)addRecord:(ABRecord *)record
```

Parameters

record

The record to add.

Return Value

YES if the record was added successfully; otherwise NO.

Discussion

If the *record* argument is *nil*, this method raises an exception. Your changes are not committed until you call the *save* (page 19) method.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [removeRecord:](#) (page 19)
- [initWithVCardRepresentation:](#) (page 64) (ABPerson)

Declared In

ABAddressBook.h

defaultCountryCode

Returns the default country code for records with unspecified country codes.

```
- (NSString *)defaultCountryCode
```

Return Value

The default country code.

Discussion

This value returned is set by the user in the Address Book application's general preference. The supported country codes are listed in "Country Codes".

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABAddressBook.h

defaultNameOrdering

Returns the default name ordering defined by the user in the Address Book application's preferences.

```
- (NSInteger)defaultNameOrdering
```

Discussion

The possible values are *kABFirstNameFirst* and *kABLastNameFirst*.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABAddressBook.h

formattedAddressFromDictionary:

Returns an attributed string containing the formatted address.

- (NSAttributedString *)formattedAddressFromDictionary:(NSDictionary *)address

Parameters

address

The dictionary containing a street address.

Return Value

An attributed string containing the formatted address.

Discussion

The string's attributes match address dictionary keys, such as `kABAddressStreetKey`. Each attribute value contains the localized description of the key. (For example, the value of a Canadian `kABAddressZIPKey` field would be "Postal Code"; while the value of a French one would be "Code Postal".)

To get the dictionary containing a street address for a person record, use [valueForProperty:](#) (page 73) with the property `kABAddressProperty`, and then getting one of the values from the multivalue that is returned.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ABAddressBook.h`

groups

Returns an array of all the groups in the Address Book database.

- (NSArray *)groups

Discussion

If the database doesn't contain any groups, this method returns an empty array.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [people](#) (page 17)

Declared In

`ABAddressBook.h`

hasUnsavedChanges

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

- (BOOL)hasUnsavedChanges

Return Value

YES if there are unsaved changes; otherwise, NO.

Discussion

The unsaved changes flag is set automatically whenever changes are made.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [saveAndReturnError:](#) (page 20)

Declared In

ABAddressBook.h

me

Returns the ABPerson record that represents the logged-in user.

- (ABPerson *)me

Discussion

If the user never specified such a record, this method returns `nil`.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [setMe:](#) (page 20)

Related Sample Code

Birthdays

Declared In

ABAddressBook.h

people

Returns an array of all the people in the Address Book database.

- (NSArray *)people

Discussion

If the database doesn't contain any people, this method returns an empty array.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [groups](#) (page 16)

Related Sample Code

ABPresence

Declared In

ABAddressBook.h

recordClassFromUniqueId:

Returns the class name of the record that matches the given unique ID.

```
- (NSString *)recordClassFromUniqueId:(NSString *)uniqueId
```

Parameters

uniqueId

The unique ID of the record.

Return Value

The name of the class of the record, for example `ABPerson`.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ABAddressBook.h`

recordForUniqueId:

Returns the person or group record that matches the given unique ID.

```
- (ABRecord *)recordForUniqueId:(NSString *)uniqueId
```

Parameters

uniqueId

The unique ID of the record. This value must not be `nil`; otherwise, an exception is raised.

Discussion

If no record has the given ID, this method returns `nil`.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [uniqueId](#) (page 72) (`ABRecord`)

Declared In

`ABAddressBook.h`

recordsMatchingSearchElement:

Returns an array of records that match the given search element, or returns an empty array if no records match the search element.

```
- (NSArray *)recordsMatchingSearchElement:(ABSearchElement *)search
```

Parameters

search

The search element to perform the search against.

Return Value

An array of records that match the given search element, or an empty array if no records match the search element.

Discussion

If *search* is *nil*, this method raises an exception.

Availability

Available in Mac OS X v10.2 and later.

See Also

- + [searchElementForProperty:label:key:value:comparison:](#) (page 62) (ABPerson)
- + [searchElementForProperty:label:key:value:comparison:](#) (page 26) (ABGroup)
- + [searchElementForConjunction:children:](#) (page 76) (ABSearchElement)

Related Sample Code

AddressBookCocoa

Birthdays

Declared In

ABAddressBook.h

removeRecord:

Removes an ABPerson or ABGroup record from the Address Book database.

```
- (BOOL)removeRecord:(ABRecord *)record
```

Parameters

record

The record to be removed.

Return Value

YES if the record was removed successfully; otherwise, NO.

Discussion

If *record* is *nil*, this method raises an exception. Your changes are not committed until you call the [save](#) (page 19) method.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [addRecord:](#) (page 14)

Declared In

ABAddressBook.h

save

Saves all the changes made since the last save.

```
- (BOOL)save
```

Return Value

YES if successful or there were no changes; otherwise, NO.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [hasUnsavedChanges](#) (page 16)
- [saveAndReturnError:](#) (page 20)

Declared In

ABAddressBook.h

saveAndReturnError:

Saves all the changes made since the last save.

```
- (BOOL)saveAndReturnError:(NSError **)error
```

Parameters

error

A pointer to an error object that is set to an `NSError` instance if an error occurs.

Return Value

YES if successful or there were no changes; otherwise, NO.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [save](#) (page 19)
- [hasUnsavedChanges](#) (page 16)

Declared In

ABAddressBook.h

setMe:

Sets the record that represents the logged-in user.

```
- (void)setMe:(ABPerson *)person
```

Parameters

person

The person to set as representing the logged-in user.

Discussion

If you don't want a record to represent the logged-in user, then pass `nil` as the *person* argument. Note that this will not delete the existing "me" card, if one is set.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [me](#) (page 17)

Declared In

ABAddressBook.h

Constants

Database change notification keys

Keys contained by the user-info dictionary of the notifications posted by the Address Book framework.

```
NSString * const kABInsertedRecords;
NSString * const kABUpdatedRecords;
NSString * const kABDeletedRecords;
```

Constants

kABInsertedRecords

Records that have been inserted.

Available in Mac OS X v10.3 and later.

Declared in ABGlobals.h.

kABUpdatedRecords

Records that have been updated.

Available in Mac OS X v10.3 and later.

Declared in ABGlobals.h.

kABDeletedRecords

Records that have been deleted.

Available in Mac OS X v10.3 and later.

Declared in ABGlobals.h.

Notifications

These notifications are sent when something changes in the Address Book database. These are not sent until the `sharedAddressBook` class method has been invoked.

kABDatabaseChangedNotification

Posted when this process has changed the Address Book database.

The process ID of the sender and the effective user ID are always included in the user-info dictionary; you access them through the keys `kABSenderProcessID` and `kABUserID`, respectively. In addition, depending on the operation performed on the address book, one or more of the following keys may be included:

`kABInsertedRecords`, `kABUpdatedRecords`, and `kABDeletedRecords`. The values for each of the keys are the unique IDs of the records that were inserted, updated, or deleted, respectively. If the values for all the keys are `nil`, every record has changes. For example, this happens when the Address Book database is restored from a backup copy.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABGlobals.h

kABDatabaseChangedExternallyNotification

Posted when a process other than the current one has changed the Address Book database.

The process id of the sender and the effective user ID are always included in the user-info dictionary; you access them through the keys `kABSenderProcessID` and `kABUserID`, respectively. In addition, depending on the operation performed on the address book, one or more of the following keys may be included: `kABInsertedRecords`, `kABUpdatedRecords`, and `kABDeletedRecords`. The values for each of the keys are the unique IDs of the records that were inserted, updated, or deleted, respectively. If the values for all the keys are `nil`, every record has changes. For example, this happens when the Address Book database is restored from a backup copy.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABGlobals.h

ABGroup 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	ABGroup.h
Companion guide	Address Book Programming Guide for Mac OS X

Overview

The `ABGroup` class supports the concept of a “group” containing one or more persons. People may belong to multiple groups, and groups may also belong to other groups unless the relationship causes a circular reference. The only predefined property of a group is its name. However, similar to person records, you can add your own properties to group records. Groups not only help to organize person records, but also allow you to create email distribution lists.

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

Tasks

Managing Properties

- + [addPropertiesAndTypes:](#) (page 24)
Adds the given properties to all records of this type in the Address Book database.
- + [removeProperties:](#) (page 25)
Removes the given properties from all the records of this type in the Address Book database.
- + [properties](#) (page 25)
Returns an array of the names of all the properties for this record type in the Address Book database.
- + [typeOfProperty:](#) (page 26)
Returns the type for a given property.

Managing Persons

- [addMember:](#) (page 27)
Adds a person to a group.
- [removeMember:](#) (page 29)
Removes a person from a group.
- [members](#) (page 29)
Returns an array of persons in a group.

Managing Subgroups

- [addSubgroup:](#) (page 28)
Adds a subgroup to another group.
- [removeSubgroup:](#) (page 30)
Removes a subgroup from a group.
- [parentGroups](#) (page 29)
Returns an array containing a group's parents—that is, the groups that a group belongs to.
- [subgroups](#) (page 31)
Returns an array containing a group's subgroups.

Managing Distribution Lists

- [distributionIdentifierForProperty:person:](#) (page 28)
Returns the distribution identifier for the given property and person.
- [setDistributionIdentifier:forProperty:person:](#) (page 31)
Assigns a specific distribution identifier for a person's multivalue list property so that the group can be used as a distribution list.

Searching

- + [searchElementForProperty:label:key:value:comparison:](#) (page 26)
Returns a search element object that searches for records of this type.

Class Methods

addPropertiesAndTypes:

Adds the given properties to all records of this type in the Address Book database.

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

Parameters*properties*

A dictionary of properties to be added and their types.

Return Value

The number of properties successfully added.

Discussion

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.groupID`. The property type must be one of the constants described in “Property Types” (page 113).

Availability

Available in Mac OS X v10.2 and later.

See Also

+ [removeProperties:](#) (page 25)

Declared In

ABGroup.h

properties

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

+ (NSArray *)*properties*

Availability

Available in Mac OS X v10.2 and later.

See Also

+ [typeOfProperty:](#) (page 26)

Declared In

ABGroup.h

removeProperties:

Removes the given properties from all the records of this type in the Address Book database.

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

Parameters*properties*

An array of the properties to be removed.

Return Value

The number of properties successfully removed.

Availability

Available in Mac OS X v10.2 and later.

See Also

+ [addPropertiesAndTypes:](#) (page 24)

Declared In

ABGroup.h

searchElementForProperty:label:key:value:comparison:

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

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

Parameters

property

The name of the property to search on. It cannot be `nil`. For a full list of the properties, see “[Default Record Properties](#)” (page 108) and “[Default Group Properties](#)” (page 109).

label

The label name for a multivalue list. If *property* does not have multiple values, pass `nil`. If *property* does have multiple values, pass `nil` to search all the values. By default, ABGroup records don’t contain any multivalue list properties.

key

The key name for a dictionary. Pass `nil` if *property* is not a dictionary. If *property* is a dictionary, pass `nil` to search all keys. By default, ABGroup records don’t contain any properties that are dictionaries.

value

What you’re searching for. If `nil`, the only supported value for *comparison* is `kABEqual` or `kABNotEqual`.

comparison

The type of comparison to perform and is an [ABSearchComparison](#) (page 98), such as `kABEqual` or `kABPrefixMatchCaseInsensitive`.

Availability

Available in Mac OS X v10.2 and later.

See Also

+ [searchElementForProperty:label:key:value:comparison:](#) (page 62) (ABPerson)

+ [searchElementForConjunction:children:](#) (page 76) (ABSearchElement)

– [recordsMatchingSearchElement:](#) (page 18) (ABAddressBook)

Declared In

ABGroup.h

typeOfProperty:

Returns the type for a given property.

```
+ (ABPropertyType)typeOfProperty:(NSString *)property
```

Parameters

property

The property whose type will be returned.

Return Value

The property type of *property*.

Discussion

If the property does not exist, this method returns `kABErrorInProperty`.

Availability

Available in Mac OS X v10.2 and later.

See Also

+ [properties](#) (page 25)

Declared In

`ABGroup.h`

Instance Methods

addMember:

Adds a person to a group.

```
- (BOOL)addMember:(ABPerson *)person
```

Parameters

person

The person record to be added.

Return Value

YES if successful; otherwise, NO.

Discussion

If the *person* argument is already part of the group, this method does nothing and returns NO. If the *person* argument is `nil`, this method raises an exception.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [removeMember:](#) (page 29)

- [addSubgroup:](#) (page 28)

- [members](#) (page 29)

Declared In

`ABGroup.h`

addSubgroup:

Adds a subgroup to another group.

```
- (BOOL)addSubgroup:(ABGroup *)group
```

Parameters

group

The group to add as a subgroup.

Return Value

YES if successful; otherwise, NO.

Discussion

If the *group* argument is already part of the receiver, this method does nothing and returns NO. If adding the group would create a recursion, this method also does nothing and returns NO. (For example, if the group Animal Lovers is in Dog Lovers, and you add Dog Lovers to Animal Lovers, that would create a recursion, which this method won't allow.) If the *group* argument is *nil*, this method raises an exception.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [removeSubgroup:](#) (page 30)
- [addMember:](#) (page 27)
- [subgroups](#) (page 31)

Declared In

ABGroup.h

distributionIdentifierForProperty:person:

Returns the distribution identifier for the given property and person.

```
- (NSString *)distributionIdentifierForProperty:(NSString *)property person:(ABPerson *)person
```

Parameters

property

The property whose distribution identifier will be returned.

person

The person whose distribution identifier will be returned.

Return Value

The distribution identifier for the given property and person.

Discussion

If a distribution identifier is not set, this method returns the multivalue's primary identifier. If either the *property* or the *person* argument is *nil*, this method returns *nil*. This method also returns *nil* if *property* is not a multivalue list property.

Distribution identifiers let you use groups as distribution lists, by indicating which value in a multivalue property should be used when addressing the group. See "Using Address Book Groups as Distribution Lists" for more detailed discussion.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [setDistributionIdentifier:forProperty:person:](#) (page 31)
- [primaryIdentifier](#) (page 36) (ABMultiValue)

Declared In

ABGroup.h

members

Returns an array of persons in a group.

- (NSArray *)members

Discussion

If this group doesn't contain any people, this method returns an empty array.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [addMember:](#) (page 27)
- [removeMember:](#) (page 29)
- [subgroups](#) (page 31)

Declared In

ABGroup.h

parentGroups

Returns an array containing a group's parents—that is, the groups that a group belongs to.

- (NSArray *)parentGroups

Discussion

If this group doesn't belong to any groups, this method returns an empty array.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [subgroups](#) (page 31)

Declared In

ABGroup.h

removeMember:

Removes a person from a group.

- (BOOL)removeMember:(ABPerson *)*person*

Parameters

person

The person to be removed.

Return Value

YES if successful; otherwise, NO.

Discussion

If the *person* argument is not in the group, this method does nothing and returns NO. If *person* is *nil*, this method raises an exception.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [addMember:](#) (page 27)
- [members](#) (page 29)
- [removeSubgroup:](#) (page 30)

Declared In

ABGroup.h

removeSubgroup:

Removes a subgroup from a group.

- (BOOL)removeSubgroup:(ABGroup *)*group*

Parameters

group

The subgroup to be removed.

Return Value

YES if successful; otherwise, NO.

Discussion

If the *group* argument is not a subgroup, this method does nothing and returns NO. If *group* is *nil*, this method raises an exception.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [addSubgroup:](#) (page 28)
- [subgroups](#) (page 31)
- [removeMember:](#) (page 29)

Declared In

ABGroup.h

setDistributionIdentifier:forProperty:person:

Assigns a specific distribution identifier for a person's multivalue list property so that the group can be used as a distribution list.

```
- (BOOL)setDistributionIdentifier:(NSString *)identifier forProperty:(NSString *)property person:(ABPerson *)person
```

Parameters

identifier

The identifier to be set as the distribution identifier

property

The property whose distribution identifier will be set.

person

The person whose distribution identifier will be.

Return Value

YES if successful; otherwise, NO.

Discussion

The default distribution identifier is a multivalue list's primary identifier. If *person* is *nil*, this method raises an exception.

Distribution identifiers let you use groups as distribution lists, by indicating which value in a multivalue property should be used when addressing the group. See "Using Address Book Groups as Distribution Lists" for more detailed discussion.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [distributionIdentifierForProperty:person:](#) (page 28)
- [setPrimaryIdentifier:](#) (page 43) (ABMutableMultiValue)

Declared In

ABGroup.h

subgroups

Returns an array containing a group's subgroups.

```
- (NSArray *)subgroups
```

Discussion

If this group doesn't contain any groups, this method returns an empty array.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [addSubgroup:](#) (page 28)
- [removeSubgroup:](#) (page 30)
- [members](#) (page 29)

Declared In

ABGroup.h

ABMultiValue Class Reference

Inherits from	NSObject
Conforms to	NSCopying NSMutableCopying NSFastEnumeration NSObject (NSObject)
Framework	/System/Library/Frameworks/AddressBook.framework
Availability	Available in Mac OS X v10.2 and later.
Declared in	ABMultiValue.h
Companion guide	Address Book Programming Guide for Mac OS X

Overview

The `ABMultiValue` and `ABMutableMultiValue` classes are used to represent properties that might have multiple values. Each value in a multivalued list must be of the same type, and must have an associated predefined or user-defined label, and unique identifier. The labels, however, need not be unique. For example, you can have multiple Home phone numbers. Each multivalued object may have a primary identifier—used as a default value when a label is not provided. For example, a person record may have multiple addresses with the labels Home and Work, where Work is designated as the primary value. Instances of this class are immutable, see `ABMutableMultiValue` for methods that manipulate the content of a multivalued list.

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

Tasks

Accessing the Primary Identifier

- [primaryIdentifier](#) (page 36)
Returns the identifier for the primary value.

Accessing Identifiers

- `identifierAtIndex:` (page 35)
Returns the identifier for the given index.
- `indexForIdentifier:` (page 35)
Returns the index for the given identifier.

Accessing Entries

- `labelAtIndex:` (page 35)
Returns the label for the given index.
- `valueAtIndex:` (page 37)
Returns the value for the given index.
- `valueForIdentifier:` (page 37)
Returns the value for the given identifier.
- `labelForIdentifier:` (page 36)
Returns the label for the given identifier.

Querying the List

- `count` (page 34)
Returns the number of entries in a multivalue list.
- `propertyType` (page 37)
Returns the type for the values in a multivalue list.

Instance Methods

count

Returns the number of entries in a multivalue list.

- `(NSUInteger)count`

Availability

Available in Mac OS X v10.2 and later.

Related Sample Code

AddressBookCocoa

Declared In

ABMultiValue.h

identifierAtIndex:

Returns the identifier for the given index.

```
- (NSString *)identifierAtIndex:(NSUInteger) index
```

Parameters

index

The index of the identifier to be returned.

Discussion

If the *index* argument is out of bounds, this method raises an exception.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [indexOfIdentifier:](#) (page 35)

Declared In

ABMultiValue.h

indexOfIdentifier:

Returns the index for the given identifier.

```
- (NSUInteger)indexOfIdentifier:(NSString *) identifier
```

Parameters

identifier

The identifier whose index will be returned.

Discussion

If the identifier is not found, returns `NSNotFound`.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [identifierAtIndex:](#) (page 35)

Declared In

ABMultiValue.h

labelAtIndex:

Returns the label for the given index.

```
- (NSString *)labelAtIndex:(NSUInteger) index
```

Parameters

index

The index for the label to be returned.

Discussion

If the *index* argument is out of bounds, this method raises an exception.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [labelForIdentifier:](#) (page 36)

Related Sample Code

AddressBookCocoa

Declared In

ABMultiValue.h

labelForIdentifier:

Returns the label for the given identifier.

```
- (id)labelForIdentifier:(NSString *)identifier
```

Parameters

identifier

The identifier for the label to be returned.

Discussion

If the identifier is not found, this method returns `nil`.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [labelAtIndex:](#) (page 35)

Declared In

ABMultiValue.h

primaryIdentifier

Returns the identifier for the primary value.

```
- (NSString *)primaryIdentifier
```

Availability

Available in Mac OS X v10.2 and later.

See Also

- [indexForIdentifier:](#) (page 35)

- [setPrimaryIdentifier:](#) (page 43) (ABMutableMultiValue)

Declared In

ABMultiValue.h

propertyType

Returns the type for the values in a multivalue list.

- (ABPropertyType)propertyType

Discussion

If the multivalue list is empty or its values are of different types, it returns `kABErrorInProperty`.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABMultiValue.h

valueAtIndex:

Returns the value for the given index.

- (id)valueAtIndex:(NSUInteger) *index*

Parameters

index

The index for the value to be returned.

Discussion

If the *index* argument is out of bounds, this method raises an exception.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [valueForIdentifier:](#) (page 37)

Related Sample Code

AddressBookCocoa

Declared In

ABMultiValue.h

valueForIdentifier:

Returns the value for the given identifier.

- (id)valueForIdentifier:(NSString *) *identifier*

Parameters

identifier

The identifier for the value to be returned.

Discussion

If the identifier is not found, returns `nil`.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [valueAtIndex:](#) (page 37)

Declared In

ABMultiValue.h

ABMutableMultiValue Class Reference

Inherits from	ABMultiValue : NSObject
Conforms to	NSCopying (ABMultiValue) NSMutableCopying (ABMultiValue) NSFastEnumeration (ABMultiValue) NSObject (NSObject)
Framework	/System/Library/Frameworks/AddressBook.framework
Availability	Available in Mac OS X v10.2 and later.
Declared in	ABMultiValue.h
Companion guide	Address Book Programming Guide for Mac OS X
Related sample code	AddressBookCocoa

Overview

The `ABMultiValue` and `ABMutableMultiValue` classes are used to represent properties that might have multiple values. Each value in a multivalued list must be of the same type, and must have an associated predefined or user-defined label, and unique identifier. The labels, however, need not be unique. For example, you can have multiple Home phone numbers. Each multivalued object may have a primary identifier—used as a default value when a label is not provided. For example, a person record may have multiple addresses with the labels Home and Work, where Work is designated as the primary value. Instances of `ABMutableMultiValue` are mutable, see `ABMultiValue` for additional methods that access the content of a multivalued list.

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

Tasks

Adding a Value

- `addValue:withLabel:` (page 40)
Adds a value and its label to a multivalued list.

- [insertValue:withLabel:atIndex:](#) (page 41)
Inserts a value and its label at the given index in a multivalue list.

Replacing Values and Labels

- [replaceLabelAtIndex:withLabel:](#) (page 42)
Replaces the label at the given index.
- [replaceValueAtIndex:withValue:](#) (page 42)
Replaces the value at the given index.

Removing Values

- [removeValueAndLabelAtIndex:](#) (page 42)
Removes the value and label at the given index.

Setting the Primary Identifier

- [setPrimaryIdentifier:](#) (page 43)
Sets the primary value to be the value for the given identifier.

Instance Methods

addValue:withLabel:

Adds a value and its label to a multivalue list.

```
(NSString *)addValue:(id)value withLabel:(NSString *)label
```

Parameters

value

The value to add.

label

The label to associate with the value.

Return Value

The new identifier if *value* is added successfully; otherwise, `nil`.

Discussion

The *value* argument must be of the correct type. For example, if the receiver is the value for a property of type `kABMultiStringProperty`, then *value* needs to be an `NSString` object. See “[Property Types](#)” (page 113) for a list of supported types in a multivalue list (see descriptions of the `kABMulti...` constants). If either the *value* or the *label* argument is `nil`, this method raises an exception.

This method performs no type checking and will let you add a value whose type does not match the types of the other values in the list. However, if you try to use a multivalue list whose values are not all of the same type, other methods, such as the `ABRecord setValue:forProperty:` (page 71) method, will return `NO` or `kABErrorInProperty` (page 114).

Availability

Available in Mac OS X v10.2 and later.

See Also

- [insertValue:withLabel:atIndex:](#) (page 41)

Related Sample Code

AddressBookCocoa

Declared In

ABMultiValue.h

insertValue:withLabel:atIndex:

Inserts a value and its label at the given index in a multivalue list.

```
- (NSString *)insertValue:(id)value withLabel:(NSString *)label  
    atIndex:(NSUInteger)index
```

Parameters

value

The value to add.

label

The label to associate with the value.

index

The index where the value will be inserted.

Return Value

The identifier of the inserted value and label if they are added successfully; otherwise, `nil`.

Discussion

If either the value or the label is `nil` or if the index is out of bounds, this method raises an exception

This method performs no type checking and will let you add a value whose type does not match the types of the other values in the list. However, if you try to use a multivalue list whose values are not all of the same type, other methods, such as the `ABRecord setValue:forProperty:` (page 71) method, will return `NO` or `kABErrorInProperty`.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [addValue:withLabel:](#) (page 40)

Declared In

ABMultiValue.h

removeValueAndLabelAtIndex:

Removes the value and label at the given index.

- (BOOL)removeValueAndLabelAtIndex:(NSUInteger) *index*

Parameters

index

The index of the value-label pair that will be removed.

Return Value

YES if successful; otherwise NO.

Discussion

If the index is out of bounds, this method raises an exception.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABMultiValue.h

replaceLabelAtIndex:withLabel:

Replaces the label at the given index.

- (BOOL)replaceLabelAtIndex:(NSUInteger) *index* withLabel:(NSString *) *label*

Parameters

index

The index of the label that will be replaced.

label

The new label.

Return Value

YES if successful; otherwise, NO.

Discussion

If the label is `nil` or if the index is out of bounds, this method raises an exception.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [replaceValueAtIndex:withValue:](#) (page 42)

Declared In

ABMultiValue.h

replaceValueAtIndex:withValue:

Replaces the value at the given index.

- (BOOL)replaceValueAtIndex:(NSUInteger) *index* withValue:(id) *value*

Parameters*index*

The index of the value that will be replaced.

value

The new value.

Return Value

YES if successful; otherwise NO.

Discussion

If the value is `nil` or if the index is out of bounds, this method raises an exception.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [replaceLabelAtIndex:withLabel:](#) (page 42)

Declared In

ABMultiValue.h

setPrimaryIdentifier:

Sets the primary value to be the value for the given identifier.

```
- (BOOL)setPrimaryIdentifier:(NSString *)identifier
```

Parameters*identifier*

The identifier whose value will be used as the primary value for a multivalue property.

Return Value

YES if successful; otherwise NO.

Discussion

If the identifier is `nil`, this method raises an exception. Use the [identifierAtIndex:](#) (page 35) method to get the identifier given the index.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [primaryIdentifier](#) (page 36) (ABMultiValue)

Declared In

ABMultiValue.h

ABPeoplePickerView Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	/System/Library/Frameworks/AddressBook.framework
Declared in	ABPeoplePickerView.h
Availability	Available in Mac OS X v10.3 and later
Companion guide	Address Book Programming Guide for Mac OS X
Related sample code	CocoaPeoplePicker

Overview

The `ABPeoplePickerView` class allows you to customize the behavior of people-picker views in an application's user interface.

Tasks

Working with Properties in the Record List

- [addProperty:](#) (page 50)
Adds a property to the group of properties whose values are shown in the record list.
- [columnTitleForProperty:](#) (page 51)
Returns the title of a custom property.
- [displayedProperty](#) (page 48) *property*
The property currently displayed in the record list.
- [properties](#) (page 53)
Returns an array of the properties whose values are shown in the record list.
- [removeProperty:](#) (page 54)
Removes a property from the group of properties whose values are shown in the record list.
- [setColumnTitle:forProperty:](#) (page 57)
Sets the title displayed in the people picker for a property.

Specifying Selection Behavior

`valueSelectionBehavior` (page 50) *property*

The current selection behavior.

Selecting Groups and Records

`allowsGroupSelection` (page 47) *property*

A Boolean value that specifies whether the user can select entire groups in the group column.

`allowsMultipleSelection` (page 48) *property*

A Boolean value that specifies whether multiple groups, records, or values of multivalued properties can be selected at a time.

- `deselectAll`: (page 52)

Deselects all selected groups, records, and values in multivalued properties.

- `deselectGroup`: (page 52)

Deselects a group selected in the group list.

- `deselectIdentifier:forPerson`: (page 52)

Deselects a value selected in a multivalued property.

- `deselectRecord`: (page 53)

Deselects a record selected in the record list.

`selectedGroups` (page 49) *property*

The groups selected in the group list. (read-only)

- `selectedIdentifiersForPerson`: (page 54)

Returns the identifiers of the selected values in a multivalued property.

`selectedRecords` (page 49) *property*

The selection in the records list. (read-only)

- `selectedValues` (page 55)

Returns an array of all the values selected in the displayed multivalued property.

- `selectGroup:byExtendingSelection`: (page 55)

Selects a group or a set of groups in the group list.

- `selectIdentifier:forPerson:byExtendingSelection`: (page 55)

Selects a value or a set of values in a multivalued property.

- `selectRecord:byExtendingSelection`: (page 56)

Selects a record or a set of records in the record list.

Specifying the Accessory View

`accessoryView` (page 47) *property*

The view that is placed to the left of the search field.

Managing Actions

- `clearSearchField:` (page 51)
Clears the search field and resets the list of displayed records.
- `editInAddressBook:` (page 53)
Launches Address Book to edit the item selected in the people picker.
- `groupDoubleAction` (page 49) *property*
The action to be invoked when a group is double-clicked.
- `nameDoubleAction` (page 49) *property*
The action to be invoked when a name is double-clicked.
- `selectInAddressBook:` (page 56)
Launches Address Book and selects the item selected in the people picker.
- `target` (page 50) *property*
The target for double-click actions.

Managing Persistent User Settings

- `autosaveName` (page 48) *property*
The name under which the column positions and the filter selection are saved.

Properties

For more about Objective-C properties, see “Properties” in *The Objective-C Programming Language*.

accessoryView

The view that is placed to the left of the search field.

```
@property (nonatomic, retain) NSView* accessoryView
```

Discussion

If `accessory` is `nil`, the accessory view is removed.

Availability

Available in Mac OS X v10.6 and later.

Declared In

ABPeoplePickerView.h

allowsGroupSelection

A Boolean value that specifies whether the user can select entire groups in the group column.

```
@property (nonatomic, assign) BOOL allowsGroupSelection
```

Discussion

If YES, the user can to select entire groups. If NO, the user is required to select at least one person in the group.

Availability

Available in Mac OS X v10.6 and later.

Declared In

ABPeoplePickerView.h

allowsMultipleSelection

A Boolean value that specifies whether multiple groups, records, or values of multivalued properties can be selected at a time.

```
@property (nonatomic, assign) BOOL allowsMultipleSelection
```

Availability

Available in Mac OS X v10.6 and later.

Declared In

ABPeoplePickerView.h

autosaveName

The name under which the column positions and the filter selection are saved.

```
@property (nonatomic, retain) NSString* autosaveName
```

Availability

Available in Mac OS X v10.6 and later.

Declared In

ABPeoplePickerView.h

displayedProperty

The property currently displayed in the record list.

```
@property (nonatomic, retain) NSString* displayedProperty
```

Availability

Available in Mac OS X v10.6 and later.

Declared In

ABPeoplePickerView.h

groupDoubleAction

The action to be invoked when a group is double-clicked.

```
@property (nonatomic, assign) SEL groupDoubleAction
```

Availability

Available in Mac OS X v10.6 and later.

Declared In

ABPeoplePickerView.h

nameDoubleAction

The action to be invoked when a name is double-clicked.

```
@property (nonatomic, assign) SEL nameDoubleAction;
```

Availability

Available in Mac OS X v10.6 and later.

Declared In

ABPeoplePickerView.h

selectedGroups

The groups selected in the group list. (read-only)

```
@property (readonly) NSArray* selectedGroups
```

Discussion

The selected groups are returned as an array of `ABGroup` objects.

Availability

Available in Mac OS X v10.6 and later.

Declared In

ABPeoplePickerView.h

selectedRecords

The selection in the records list. (read-only)

```
@property (readonly) NSArray* selectedRecords
```

Discussion

The selection is returned as an array of `ABGroup` or `ABPerson` objects.

Availability

Available in Mac OS X v10.6 and later.

See Also

- [selectRecord:byExtendingSelection:](#) (page 56)

- [deselectRecord:](#) (page 53)

Declared In

ABPeoplePickerView.h

target

The target for double-click actions.

```
@property (nonatomic, assign) id target
```

Discussion

The target is the object on which the action specified by `groupDoubleAction` and `nameDoubleAction` is invoked.

Availability

Available in Mac OS X v10.6 and later.

Declared In

ABPeoplePickerView.h

valueSelectionBehavior

The current selection behavior.

```
@property (nonatomic, assign) ABPeoplePickerSelectionBehavior valueSelectionBehavior
```

Discussion

The default behavior is [ABSingleValueSelection](#) (page 57).

Availability

Available in Mac OS X v10.6 and later.

See Also

[Selection Behavior](#) (page 99)

Declared In

ABPeoplePickerView.h

Instance Methods

addProperty:

Adds a property to the group of properties whose values are shown in the record list.

```
- (void)addProperty:(NSString *)property
```

Parameters

property

The property to add.

Discussion

For additional information about properties see “[Constants](#)” (page 66).

Availability

Available in Mac OS X v10.3 and later.

See Also

- [removeProperty:](#) (page 54)
- [properties](#) (page 53)

Declared In

ABPeoplePickerView.h

clearSearchField:

Clears the search field and resets the list of displayed records.

```
- (void)clearSearchField:(id)sender
```

Parameters

sender

The object sending this message.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABPeoplePickerView.h

columnTitleForProperty:

Returns the title of a custom property.

```
- (NSString *)columnTitleForProperty:(NSString *)property
```

Parameters

property

The property whose title will be returned.

Return Value

The title of the custom property.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setColumnTitle:forProperty:](#) (page 57)

Declared In

ABPeoplePickerView.h

deselectAll:

Deselects all selected groups, records, and values in multivalue properties.

- (void)deselectAll:(id) *sender*

Parameters

sender

The object sending this message.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABPeoplePickerView.h

deselectGroup:

Deselects a group selected in the group list.

- (void)deselectGroup:(ABGroup *) *group*

Parameters

group

The group to deselect.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [@property selectedGroups](#) (page 49)
- [selectGroup:byExtendingSelection:](#) (page 55)

Declared In

ABPeoplePickerView.h

deselectIdentifier:forPerson:

Deselects a value selected in a multivalue property.

- (void)deselectIdentifier:(NSString *) *identifier* forPerson:(ABPerson *) *person*

Parameters

identifier

The identifier of the value that will be deselected.

person

The person whose value will be deselected.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [selectedIdentifiersForPerson:](#) (page 54)

- [selectIdentifier:forPerson:byExtendingSelection:](#) (page 55)

Declared In

ABPeoplePickerView.h

deselectRecord:

Deselects a record selected in the record list.

- (void)deselectRecord:(ABRecord *)*record*

Parameters

record

The record to deselect.

Availability

Available in Mac OS X v10.3 and later.

See Also

[@property selectedRecords](#) (page 49)
- [selectRecord:byExtendingSelection:](#) (page 56)

Declared In

ABPeoplePickerView.h

editInAddressBook:

Launches Address Book to edit the item selected in the people picker.

- (void)editInAddressBook:(id)*sender*

Parameters

sender

The object sending this message.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [selectInAddressBook:](#) (page 56)

Declared In

ABPeoplePickerView.h

properties

Returns an array of the properties whose values are shown in the record list.

- (NSArray *)*properties*

Discussion

For additional information about properties see “Using Property Lists” in *Address Book Programming Guide for Mac OS X*.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [addProperty:](#) (page 50)
- [removeProperty:](#) (page 54)

Declared In

ABPeoplePickerView.h

removeProperty:

Removes a property from the group of properties whose values are shown in the record list.

```
- (void)removeProperty:(NSString *)property
```

Parameters

property

The property to remove.

Discussion

For additional information about properties see “Using Property Lists” in *Address Book Programming Guide for Mac OS X*.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [addProperty:](#) (page 50)
- [properties](#) (page 53)

Declared In

ABPeoplePickerView.h

selectedIdentifiersForPerson:

Returns the identifiers of the selected values in a multivalue property.

```
- (NSArray *)selectedIdentifiersForPerson:(ABPerson *)person
```

Parameters

person

The person whose identifiers for selected values will be returned.

Discussion

Returns `nil` if the property displayed is a single-value property.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [selectIdentifier:forPerson:byExtendingSelection:](#) (page 55)
- [deselectIdentifier:forPerson:](#) (page 52)

Declared In

ABPeoplePickerView.h

selectedValues

Returns an array of all the values selected in the displayed multivalued property.

- (NSArray *)selectedValues

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABPeoplePickerView.h

selectGroup:byExtendingSelection:

Selects a group or a set of groups in the group list.

- (void)selectGroup:(ABGroup *)group byExtendingSelection:(BOOL)extend

Parameters

group

The group to be selected, or to be added to the current selection.

extend

YES to extend the current selection; otherwise, NO.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [@property selectedGroups](#) (page 49)
- [deselectGroup:](#) (page 52)

Declared In

ABPeoplePickerView.h

selectIdentifier:forPerson:byExtendingSelection:

Selects a value or a set of values in a multivalued property.

- (void)selectIdentifier:(NSString *)identifier forPerson:(ABPerson *)person
byExtendingSelection:(BOOL)extend

Parameters

identifier

The identifier to be selected, or to be added to the current selection.

person

The person that the value to be selected is associated with.

extend

YES to extend the current selection; otherwise, NO.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [selectedIdentifiersForPerson:](#) (page 54)
- [deselectIdentifier:forPerson:](#) (page 52)

Declared In

ABPeoplePickerView.h

selectInAddressBook:

Launches Address Book and selects the item selected in the people picker.

```
- (void)selectInAddressBook:(id)sender
```

Parameters

sender

The object sending this message.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [editInAddressBook:](#) (page 53)

Declared In

ABPeoplePickerView.h

selectRecord:byExtendingSelection:

Selects a record or a set of records in the record list.

```
- (void)selectRecord:(ABRecord *)record byExtendingSelection:(BOOL)extend
```

Parameters

record

The record to be selected, or to be added to the current selection.

extend

YES to extend the current selection; otherwise, NO.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [deselectRecord:](#) (page 53)
- [@property selectedRecords](#) (page 49)

Declared In

ABPeoplePickerView.h

setColumnTitle:forProperty:

Sets the title displayed in the people picker for a property.

```
- (void)setColumnTitle:(NSString *)title forProperty:(NSString *)property
```

Parameters*title*

The title to be set.

property

The property being titled.

Availability

Available in Mac OS X v10.3 and later.

See Also- [columnTitleForProperty:](#) (page 51)**Declared In**

ABPeoplePickerView.h

Constants

Selection Behavior

These constants are of the type [Selection Behavior](#) (page 99) and are used by [valueSelectionBehavior](#) (page 50).

```
typedef enum {
    ABNoValueSelection          = 0,
    ABSingleValueSelection      = 1,
    ABMultipleValueSelection    = 2
} ABPeoplePickerSelectionBehavior;
```

Constants

ABNoValueSelection

The user cannot select individual values.

Available in Mac OS X v10.3 and later.

Declared in ABPeoplePickerView.h.

ABSingleValueSelection

The user can select a single value.

Available in Mac OS X v10.3 and later.

Declared in ABPeoplePickerView.h.

ABMultipleValueSelection

The user can select multiple values.

Available in Mac OS X v10.3 and later.

Declared in ABPeoplePickerView.h.

Notifications

ABPeoplePickerGroupSelectionDidChangeNotification

Posted when the selection in the group list is changed.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABPeoplePickerView.h

ABPeoplePickerNameSelectionDidChangeNotification

Posted when the selection in the name list is changed.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABPeoplePickerView.h

ABPeoplePickerValueSelectionDidChangeNotification

Posted when the selection in a multivalue property is changed.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABPeoplePickerView.h

ABPeoplePickerDisplayedPropertyDidChangeNotification

Posted when the displayed property in the record list is changed.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABPeoplePickerView.h

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	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.

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 60)
Adds the given properties to all the records of this type in the Address Book database.
- + [removeProperties:](#) (page 62)
Removes the given properties from all the records of this type in the Address Book database.
- + [properties](#) (page 61)
Returns an array of the names of all the properties for the record in the Address Book database.
- + [typeOfProperty:](#) (page 63)
Returns the type of a given property.

Managing Groups

- [parentGroups](#) (page 65)
Returns an array of the address book groups that this person belongs to.

Managing Images

- + [cancelLoadingImageDataForTag:](#) (page 61)
Cancels an asynchronous fetch of the images for a given tag.
- [beginLoadingImageDataForClient:](#) (page 63)
Starts an asynchronous fetch for image data in all locations
- [imageData](#) (page 64)
Returns data that contains a picture of this person.
- [setImageData:](#) (page 65)
Sets the image for this person to the given data.

Searching

- + [searchElementForProperty:label:key:value:comparison:](#) (page 62)
Returns a search element object that specifies a query for records of this type.

Importing and Exporting VCard Formatted Files

- [initWithVCardRepresentation:](#) (page 64)
Returns an `ABPerson` instance initialized with the given data.
- [vCardRepresentation](#) (page 66)
Returns the vCard representation of the person record as a data object in vCard format.

Class Methods

addPropertiesAndTypes:

Adds the given properties to all the records of this type in the Address Book database.

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

Parameters

properties

A dictionary of properties to add, and their types.

Return Value

The number of properties successfully added, or -1 if an error occurs.

Discussion

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 ["Property Types"](#) (page 113).

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABPerson.h

cancelLoadingImageDataForTag:

Cancels an asynchronous fetch of the images for a given tag.

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

Parameters

tag

The tag of the asynchronous fetch to be canceled.

Discussion

The tag is returned from the previous call to the [beginLoadingImageDataForClient:](#) (page 63) method that started the asynchronous fetch.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [beginLoadingImageDataForClient:](#) (page 63)
- [consumeImageData:forTag:](#) (page 89) (ABImageClient)

Declared In

ABImageLoading.h

properties

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

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

Availability

Available in Mac OS X v10.2 and later.

See Also

- + [typeOfProperty:](#) (page 63)

Declared In

ABPerson.h

removeProperties:

Removes the given properties from all the records of this type in the Address Book database.

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

Parameters

properties

An array of properties to remove.

Return Value

The number of properties successfully removed, or -1 if an error occurs.

Discussion

Only custom properties can be removed.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABPerson.h

searchElementForProperty:label:key:value:comparison:

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

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

Parameters

property

The name of the property to search on, such as `kABAddressProperty` or `kABLastNameProperty`. This name cannot be `nil`. For a full list of the properties, see [“Default Record Properties”](#) (page 108) and [“Default Person Properties”](#) (page 109).

label

The label name for a multivalue 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 [“Default Multivalue List Labels”](#) (page 103) and [“Generic Multivalue List Labels”](#) (page 108).

key

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 [“Address Keys”](#) (page 115).

value

What you’re searching for. If `nil`, then the only supported value for *comparison* is `kABEqual` or `kABNotEqual`.

comparison

The type of comparison to perform, such as `kABEqual` or `kABPrefixMatchCaseInsensitive`.

Availability

Available in Mac OS X v10.2 and later.

See Also

- + [searchElementForProperty:label:key:value:comparison:](#) (page 26) (ABGroup)
- + [searchElementForConjunction:children:](#) (page 76) (ABSearchElement)
- [recordsMatchingSearchElement:](#) (page 18) (ABAddressBook)

Related Sample Code

AddressBookCocoa

Birthdays

Declared In

ABPerson.h

typeofProperty:

Returns the type of a given property.

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

Parameters*property*

The property whose type will be returned.

DiscussionIf the property does not exist, this method returns `kABErrorInProperty`.**Availability**

Available in Mac OS X v10.2 and later.

See Also

- + [properties](#) (page 61)

Declared In

ABPerson.h

Instance Methods

beginLoadingImageDataForClient:

Starts an asynchronous fetch for image data in all locations

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

Parameters*client*

The object to be notified when the image finishes loading.

Return ValueA nonzero tag for tracking. This tag is used by the [cancelLoadingImageDataForTag:](#) (page 61) method to cancel a fetch operation.

Discussion

The *client* object should conform to the `ABImageClient` protocol. A `consumeImageData:forTag:` (page 89) message is sent to *client* when the fetch is done. Use the `cancelLoadingImageDataForTag:` (page 61) method if you need to cancel an asynchronous fetch.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [imageData](#) (page 64)

Declared In

`ABImageLoading.h`

imageData

Returns data that contains a picture of this person.

```
- (NSData *)imageData
```

Return Value

Data containing a picture of this person

Discussion

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 63).

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 63)

- [setImageData:](#) (page 65)

Declared In

`ABImageLoading.h`

initWithVCardRepresentation:

Returns an `ABPerson` instance initialized with the given data.

```
- (id)initWithVCardRepresentation:(NSData *)vCardData
```

Parameters

vCardData

A data object containing a vCard representation of a person record.

Return Value

An `ABPerson` instance initialized with the given data.

Discussion

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 66)

Declared In

ABPerson.h

parentGroups

Returns an array of the address book groups that this person belongs to.

- (NSArray *)parentGroups

Discussion

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:

Sets the image for this person to the given data.

- (BOOL)setImageData:(NSData *)data

Parameters

data

The image to be set.

Discussion

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 64)

- [beginLoadingImageDataForClient:](#) (page 63)

Declared In

ABImageLoading.h

vCardRepresentation

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

- (NSData *)vCardRepresentation

Return Value

A data object containing the vCard representation of the person record.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [initWithVCardRepresentation:](#) (page 64)

Declared In

ABPerson.h

Constants

Person Flags

Settings that determine how person records are displayed.

```
#define kABShowAsMask                000007
#define kABShowAsPerson              000000
#define kABShowAsCompany              000001
#define kABShowAsResource             000002
#define kABShowAsRoom                 000003
#define kABNameOrderingMask          000070
#define kABDefaultNameOrdering       000000
#define kABFirstNameFirst             000040
#define kABLastNameFirst              000020
```

Constants

kABShowAsMask

Used in conjunction with kABShowAsPerson, kABShowAsCompany, kABShowAsResource, and kABShowAsRoom to determine record configuration.

Available in Mac OS X v10.3 and later.

Declared in ABGlobals.h.

kABShowAsPerson

Record is displayed as a person.

Available in Mac OS X v10.3 and later.

Declared in ABGlobals.h.

kABShowAsCompany

Record is displayed as a company.

Available in Mac OS X v10.3 and later.

Declared in ABGlobals.h.

`kABShowAsResource`

Record is displayed as a resource.

Available in Mac OS X v10.6 and later.

Declared in `ABGlobals.h`.

`kABShowAsRoom`

Record is displayed as a room.

Available in Mac OS X v10.6 and later.

Declared in `ABGlobals.h`.

`kABNameOrderingMask`

Used in conjunction with `kABDefaultNameOrdering`, `kABFirstNameFirst`, and `kABLastNameFirst` to determine name ordering.

Available in Mac OS X v10.3 and later.

Declared in `ABGlobals.h`.

`kABDefaultNameOrdering`

Default name ordering (whether a person's first name or last name is displayed first) the Address Book application.

Available in Mac OS X v10.3 and later.

Declared in `ABGlobals.h`.

`kABFirstNameFirst`

First name is displayed first in Address Book.

Available in Mac OS X v10.3 and later.

Declared in `ABGlobals.h`.

`kABLastNameFirst`

Last name is displayed first in Address Book.

Available in Mac OS X v10.3 and later.

Declared in `ABGlobals.h`.

ABRecord Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/AddressBook.framework
Availability	Available in Mac OS X v10.2 and later.
Declared in	ABRecord.h
Companion guide	Address Book Programming Guide for Mac OS X
Related sample code	AddressBookCocoa CocoaPeoplePicker

Overview

`ABRecord` is an abstract superclass providing a common interface to, and defining common properties for, all Address Book records. A property is a field in the database record, such as the first or last name of a person record. `ABRecord` defines the types of properties supported, and basic methods for getting, setting, and removing property values.

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

Tasks

Initializing Records

- `initWithAddressBook:` (page 70)
Initializes a record using the given address book.
- `init` (page 70)
Initializes a record using the shared address book.

Retrieving and Setting Values

- `removeValueForProperty:` (page 71)
Removes the value for a given property.
- `setValue:forProperty:` (page 71)
Sets the value of a given property for a record.
- `valueForProperty:` (page 73)
Returns the value of a given property for a record.

Retrieving a Specific Record

- `isReadOnly` (page 71)
Returns whether a record is read-only.

Determining Properties

- `uniqueId` (page 72)
Returns the unique ID for a record.

Instance Methods

init

Initializes a record using the shared address book.

```
- (id) init
```

Discussion

On Mac OS X 10.5 and later, this is equivalent to `initWithAddressBook:[ABAddressBook addressBook]`.

Availability

Available in Mac OS X v10.5 and later.

Declared In

ABRecord.h

initWithAddressBook:

Initializes a record using the given address book.

```
- (id) initWithAddressBook:(ABAddressBook *) addressBook
```

Parameters

addressBook

The address book with which to initialize the record.

Discussion

The record is added to *addressBook* but is not visible to other address books until *addressBook* is saved. This method is the designated initializer for this class.

Availability

Available in Mac OS X v10.5 and later.

Declared In

ABRecord.h

isReadOnly

Returns whether a record is read-only.

```
- (BOOL) isReadOnly
```

Return Value

YES if the record is read-only; otherwise, NO.

Availability

Available in Mac OS X v10.4 and later.

Declared In

ABRecord.h

removeValueForProperty:

Removes the value for a given property.

```
- (BOOL) removeValueForProperty:(NSString *) property
```

Parameters

property

The property whose value will be removed.

Return Value

YES if the value is removed successfully; otherwise, NO.

Discussion

When you next call [valueForProperty:](#) (page 73) on that property, it returns `nil`.

If *property* is `nil`, this method raises an exception.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABRecord.h

setValue:forProperty:

Sets the value of a given property for a record.

```
- (BOOL) setValue:( id) value forProperty:(NSString *) property
```

Parameters*value*

The value to set for property.

property

The property whose value will be set.

Return Value

YES if the value was set successfully; otherwise, NO.

Discussion

The type of the value must match the property's type (see [“Property Types”](#) (page 113) for a list of possible property types). If *property* is *nil* or if *value* is not of the correct type, this method raises an exception. If *property* is a multivalued list property, this method checks to see if the values in the multivalued list are the same type. If the multivalued list contains mixed types, the value will not be set successfully.

Availability

Available in Mac OS X v10.2 and later.

Related Sample Code

AddressBookCocoa

Declared In

ABRecord.h

uniqueId

Returns the unique ID for a record.

```
- (NSString *) uniqueId
```

Return Value

The unique ID.

Discussion

This method is equivalent to invoking [valueForProperty:](#) (page 73), passing `kABUIDProperty` as the argument.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [recordForUniqueId:](#) (page 18)(ABAddressBook)

Related Sample Code

Birthdays

Declared In

ABRecord.h

valueForProperty:

Returns the value of a given property for a record.

```
- (id) valueForProperty:(NSString *) property
```

Parameters

property

The property whose value will be returned.

Return Value

The value of the given property.

Discussion

The type of the value depends on the property type (see [“Property Types”](#) (page 113) for a list of possible property types). Note that the returned value is always of an immutable type (for example, an `NSString` type, not an `NSMutableString` type, is returned).

If *property* is `nil`, this method raises an exception.

Availability

Available in Mac OS X v10.2 and later.

Related Sample Code

`AddressBookCocoa`

`Birthdays`

Declared In

`ABRecord.h`

ABSearchElement Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/AddressBook.framework
Availability	Available in Mac OS X v10.2 and later.
Declared in	ABSearchElement.h
Companion guide	Address Book Programming Guide for Mac OS X
Related sample code	AddressBookCocoa Birthdays

Overview

The `ABSearchElement` class is used to specify a search query for records in the Address Book database.

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

Tasks

Searching

+ `searchElementForConjunction:children:` (page 76)

Returns a compound search element, created by combining the search elements in an array with the given conjunction.

Matching

- `matchesRecord:` (page 76)

Tests whether or not a record matches a search element.

Class Methods

searchElementForConjunction:children:

Returns a compound search element, created by combining the search elements in an array with the given conjunction.

```
+ (ABSearchElement *)searchElementForConjunction:(ABSearchConjunction)conjunction
  children:(NSArray *)children
```

Parameters

conjunction

The logical operator with which to combine the search elements.

children

An array of search elements to be combined.

Return Value

A compound search element, created by combining the given search elements with the given conjunction

Discussion

The objects in the *children* array must be `ABSearchElement` objects. The conjunction can be `kABSearchAnd` or `kABSearchOr`. If *children* is `nil` or empty, this method raises an exception.

Availability

Available in Mac OS X v10.2 and later.

See Also

- + [searchElementForProperty:label:key:value:comparison:](#) (page 62) (`ABPerson`)
- + [searchElementForProperty:label:key:value:comparison:](#) (page 26) (`ABGroup`)
- [recordsMatchingSearchElement:](#) (page 18) (`ABAddressBook`)

Declared In

`ABSearchElement.h`

Instance Methods

matchesRecord:

Tests whether or not a record matches a search element.

```
- (BOOL)matchesRecord:(ABRecord *)record
```

Parameters

record

The record to be tested against the search object.

Return Value

YES if the *record* argument satisfies the conditions in the search element; otherwise, NO.

Discussion

If *record* is *nil*, this method raises an exception.

Availability

Available in Mac OS X v10.2 and later.

See Also

- + [searchElementForProperty:label:key:value:comparison:](#) (page 62) (ABPerson)
- + [searchElementForProperty:label:key:value:comparison:](#) (page 26) (ABGroup)
- + [searchElementForConjunction:children:](#) (page 76)

Declared In

ABSearchElement.h

Constants

Search Conjunctions

Logical operators used to combine search elements.

```
enum _ABSearchConjunction {
    kABSearchAnd,
    kABSearchOr
};
typedef CFIndex ABSearchConjunction;
```

Constants

kABSearchAnd

Join the search elements together with the AND operand.

Available in Mac OS X v10.2 and later.

Declared in ABTypedefs.h.

kABSearchOr

Join the search elements together with the OR operand.

Available in Mac OS X v10.2 and later.

Declared in ABTypedefs.h.

Discussion

These constants are used by the [searchElementForConjunction:children:](#) (page 76) method.

Search Comparisons

Comparisons used to construct search elements.

```
enum _ABSearchComparison {
    kABEqual,
    kABNotEqual,
    kABLessThan,
    kABLessThanOrEqual,
    kABGreaterThan,
    kABGreaterThanOrEqual,
    kABEqualCaseInsensitive,
    kABContainsSubString,
    kABContainsSubStringCaseInsensitive,
    kABPrefixMatch,
    kABPrefixMatchCaseInsensitive,
    kABBitsInBitFieldMatch,
    kABDoesNotContainSubString,
    kABDoesNotContainSubStringCaseInsensitive,
    kABNotEqualCaseInsensitive,
    kABSuffixMatch,
    kABSuffixMatchCaseInsensitive,
    kABWithinIntervalAroundToday,
    kABWithinIntervalAroundTodayYearless,
    kABNotWithinIntervalAroundToday,
    kABNotWithinIntervalAroundTodayYearless,
    kABWithinIntervalFromToday,
    kABWithinIntervalFromTodayYearless,
    kABNotWithinIntervalFromToday,
    kABNotWithinIntervalFromTodayYearless
};
typedef CFIndex ABSearchComparison;
```

Constants

kABEqual

Search for elements that are equal to the value.

Available in Mac OS X v10.2 and later.

Declared in ABTypedefs.h.

kABNotEqual

Search for elements that are not equal to the value.

Available in Mac OS X v10.2 and later.

Declared in ABTypedefs.h.

kABLessThan

Search for elements that are less than the value.

Available in Mac OS X v10.2 and later.

Declared in ABTypedefs.h.

kABLessThanOrEqual

Search for elements that are less than or equal to the value.

Available in Mac OS X v10.2 and later.

Declared in ABTypedefs.h.

kABGreaterThan

Search for elements that are greater than the value.

Available in Mac OS X v10.2 and later.

Declared in ABTypedefs.h.

- `kABGreaterThanOrEqual`
Search for elements that are greater than or equal to the value.
Available in Mac OS X v10.2 and later.
Declared in `ABTypedefs.h`.
- `kABEqualCaseInsensitive`
Search for elements that are equal to the value, ignoring case.
Available in Mac OS X v10.2 and later.
Declared in `ABTypedefs.h`.
- `kABContainsSubString`
Search for elements that contain the value.
Available in Mac OS X v10.2 and later.
Declared in `ABTypedefs.h`.
- `kABContainsSubStringCaseInsensitive`
Search for elements that contain the value, ignoring case.
Available in Mac OS X v10.2 and later.
Declared in `ABTypedefs.h`.
- `kABPrefixMatch`
Search for elements that begin with the value.
Available in Mac OS X v10.2 and later.
Declared in `ABTypedefs.h`.
- `kABPrefixMatchCaseInsensitive`
Search for elements that begin with the value, ignoring case.
Available in Mac OS X v10.2 and later.
Declared in `ABTypedefs.h`.
- `kABBitsInBitFieldMatch`
Search for elements that match the bits in `ABPersonFlags`.
Available in Mac OS X v10.3 and later.
Declared in `ABTypedefs.h`.
- `kABDoesNotContainSubString`
Search for elements that do not contain the value.
Available in Mac OS X v10.4 and later.
Declared in `ABTypedefs.h`.
- `kABDoesNotContainSubStringCaseInsensitive`
Search for elements that do not contain the value, ignoring case.
Available in Mac OS X v10.4 and later.
Declared in `ABTypedefs.h`.
- `kABNotEqualCaseInsensitive`
Search for elements that are not equal to the value, ignoring case.
Available in Mac OS X v10.4 and later.
Declared in `ABTypedefs.h`.

`kABSuffixMatch`

Search for elements that end with the value.

Available in Mac OS X v10.4 and later.

Declared in `ABTypedefs.h`.

`kABSuffixMatchCaseInsensitive`

Search for elements that end with the value, ignoring case.

Available in Mac OS X v10.4 and later.

Declared in `ABTypedefs.h`.

`kABWithinIntervalAroundToday`

Search for elements that are within a time interval (in seconds) forward or backward from today.

Available in Mac OS X v10.4 and later.

Declared in `ABTypedefs.h`.

`kABWithinIntervalAroundTodayYearless`

Search for elements that are within a time interval (in seconds) forward or backward from this day in any year.

Available in Mac OS X v10.4 and later.

Declared in `ABTypedefs.h`.

`kABNotWithinIntervalAroundToday`

Search for elements that are *not* within a time interval (in seconds) forward or backward from today.

Available in Mac OS X v10.4 and later.

Declared in `ABTypedefs.h`.

`kABNotWithinIntervalAroundTodayYearless`

Search for elements that are *not* within a time interval (in seconds) forward or backward from this day in any year.

Available in Mac OS X v10.4 and later.

Declared in `ABTypedefs.h`.

`kABWithinIntervalFromToday`

Search for elements that are within a time interval (in seconds) forward from today.

Available in Mac OS X v10.4 and later.

Declared in `ABTypedefs.h`.

`kABWithinIntervalFromTodayYearless`

Search for elements that are within a time interval (in seconds) forward from this day in any year.

Available in Mac OS X v10.4 and later.

Declared in `ABTypedefs.h`.

`kABNotWithinIntervalFromToday`

Search for elements that are *not* within a time interval (in seconds) forward from today.

Available in Mac OS X v10.4 and later.

Declared in `ABTypedefs.h`.

`kABNotWithinIntervalFromTodayYearless`

Search for elements that are *not* within a time interval (in seconds) forward from this day in any year.

Available in Mac OS X v10.4 and later.

Declared in `ABTypedefs.h`.

Discussion

These constants are used by the `ABPerson` method

`searchElementForProperty:label:key:value:comparison:` (page 62) and the `ABGroup` method `searchElementForProperty:label:key:value:comparison:` (page 26)

Protocols

ABActionDelegate Protocol Reference

(informal protocol)

Framework	/System/Library/Frameworks/AddressBook.framework
Declared in	ABActions.h
Availability	Available in Mac OS X v10.3 and later.
Companion guide	Address Book Programming Guide for Mac OS X

Overview

The `ABActionDelegate` informal protocol allows you to populate the rollover menus of Address Book with custom items. You do this by implementing an Address Book action plug-in. The plug-in's `NSBundle` must implement `actionProperty:`, `titleForPerson:identifier:` and `performActionForPerson:identifier:`.

Each action plug-in can implement only one action. Actions can only apply to items with labels. An action can display a simple window inside the Address Book application; if your action actions needs to do anything else, it should launch your own application to perform the action.

Use Xcode to create Address Book action plug-ins. Place action plug-ins in `~/Library/Address Book Plug-Ins` or `/Library/Address Book Plug-Ins`, depending on the scope you want for the action.

Tasks

Performing Actions

- [performActionForPerson:identifier:](#) (page 86)
Sent to the delegate to perform the action.

Querying

- [actionProperty](#) (page 86)
Sent to the delegate to request the property the action applies to.
- [shouldEnableActionForPerson:identifier:](#) (page 86)
Sent to the delegate to determine whether the action should be enabled.
- [titleForPerson:identifier:](#) (page 87)
Sent to the delegate to request the title of the menu item for the action.

Instance Methods

actionProperty

Sent to the delegate to request the property the action applies to.

```
- (NSString *)actionProperty
```

Return Value

The property that the action applies to.

Discussion

See “Constants” (page 66) for the properties for person records, and “Constants” in *ABGroup Class Reference* for the properties for group records.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABActions.h

performActionForPerson:identifier:

Sent to the delegate to perform the action.

```
- (void)performActionForPerson:(ABPerson *)person identifier:(NSString *)identifier
```

Parameters

person

The person on which the action will be taken.

identifier

The unique identifier of the selected value.

Discussion

If the property returned by [actionProperty](#) (page 86) is a multivalue property, *identifier* contains the unique identifier of the value selected. The person being displayed in the Address Book application’s card view when the rollover menu is accessed is passed as *person*.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABActions.h

shouldEnableActionForPerson:identifier:

Sent to the delegate to determine whether the action should be enabled.

```
- (BOOL)shouldEnableActionForPerson:(ABPerson *)person identifier:(NSString *)identifier
```

Parameters*person*

The person on which the action will be taken.

identifier

The unique identifier of the selected value.

Return Value

YES if the action is applicable; otherwise, NO.

Discussion

If the property returned by [actionProperty](#) (page 86) is a multivalued property, *identifier* contains the unique identifier of the value selected.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABActions.h

titleForPerson:identifier:

Sent to the delegate to request the title of the menu item for the action.

```
- (NSString *)titleForPerson:(ABPerson *)person identifier:(NSString *)identifier
```

Parameters*person*

The person on which the action will be taken.

identifier

The unique identifier of the value for which the menu item will be displayed.

Return Value

The title of the menu item for the action.

Discussion

If the property returned by [actionProperty](#) (page 86) is a multivalued property, *identifier* contains the unique identifier of the value selected.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABActions.h

ABImageClient Protocol Reference

Adopted by	ABPerson
Conforms to	NSObject
Framework	/System/Library/Frameworks/AddressBook.framework
Availability	Available in Mac OS X v10.2 and later.
Declared in	ABImageLoading.h
Companion guide	Address Book Programming Guide for Mac OS X

Overview

The `ABImageClient` protocol is used to handle images loaded from an asynchronous fetch for `ABPerson` objects.

Tasks

Loading an Image

- `consumeImageData:forTag:` (page 89) *required method*
Gets the image data for the given tag that was initiated by an asynchronous fetch. (required)

Instance Methods

consumeImageData:forTag:

Gets the image data for the given tag that was initiated by an asynchronous fetch. (required)

```
- (void)consumeImageData:(NSData *)data forTag:(NSInteger)tag
```

Parameters*data*

A pointer to a data object that will be set to an `NSImage/QuickTime` compatible format, or `nil` if no image could be found.

You can use this image data with the `initWithData:` method of the `NSImage` class.

tag

The tag returned from a previous call to the `ABPerson beginLoadingImageDataForClient:` (page 63) method.

Discussion

In the case of a multithreaded application, this method is always called on the main thread.

Availability

Available in Mac OS X v10.2 and later.

See Also

+ `cancelLoadingImageDataForTag:` (page 61) (`ABPerson`)

- `imageData` (page 64) (`ABPerson`)

Declared In

`ABImageLoading.h`

Functions

AddressBook Functions Reference

Framework: AddressBook/AddressBook.h

Overview

This chapter describes the functions and function-like macros found in AddressBook.

Functions

ABLocalizedPropertyOrLabel

Returns the localized version of a built in property, label, or key.

```
NSString * ABLocalizedPropertyOrLabel (  
    NSString *propertyOrLabel  
);
```

Discussion

The *propertyOrLabel* argument is the property, label, or key you wish to localize. Returns *propertyOrLabel* if a localized string can not be found

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABGlobals.h

Data Types

Address Book Data Types Reference

Framework: AddressBook/AddressBook.h

Overview

This chapter describes the data types and constants found in the Address Book framework.

Data Types

ABPropertyType

Defines the possible types of ABRecord properties.

```
enum _ABPropertyType {
    kABErrorInProperty          = 0x0,
    kABStringProperty           = 0x1,
    kABIntegerProperty          = 0x2,
    kABRealProperty             = 0x3,
    kABDateProperty             = 0x4,
    kABArrayProperty            = 0x5,
    kABDictionaryProperty        = 0x6,
    kABDataProperty             = 0x7,
    kABMultiStringProperty      = kABMultiValueMask | kABStringProperty,
    kABMultiIntegerProperty     = kABMultiValueMask | kABIntegerProperty,
    kABMultiRealProperty        = kABMultiValueMask | kABRealProperty,
    kABMultiDateProperty        = kABMultiValueMask | kABDateProperty,
    kABMultiArrayProperty       = kABMultiValueMask | kABArrayProperty,
    kABMultiDictionaryProperty  = kABMultiValueMask | kABDictionaryProperty,
    kABMultiDataProperty        = kABMultiValueMask | kABDataProperty
};
typedef CFIndex ABPropertyType;
```

Discussion

These constants are described in “Constants” in “ABRecord”.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABTypedefs.h

ABSearchComparison

Defines constants used to construct search elements.

```
enum _ABSearchComparison {
    kABEqual,
    kABNotEqual,
    kABLessThan,
    kABLessThanOrEqual,
    kABGreaterThan,
    kABGreaterThanOrEqual,
    kABEqualCaseInsensitive,
    kABContainsSubString,
    kABContainsSubStringCaseInsensitive,
    kABPrefixMatch,
    kABPrefixMatchCaseInsensitive,
    kABBitsInBitFieldMatch,
    kABDoesNotContainSubString,
    kABDoesNotContainSubStringCaseInsensitive,
    kABNotEqualCaseInsensitive,
    kABSuffixMatch,
    kABSuffixMatchCaseInsensitive,
    kABWithinIntervalAroundToday,
    kABWithinIntervalAroundTodayYearless,
    kABNotWithinIntervalAroundToday,
    kABNotWithinIntervalAroundTodayYearless,
    kABWithinIntervalFromToday,
    kABWithinIntervalFromTodayYearless,
    kABNotWithinIntervalFromToday,
    kABNotWithinIntervalFromTodayYearless
};
typedef CFIndex ABSearchComparison;
```

Discussion

These constants are described in “[Constants](#)” (page 77) in “[ABSearchElement](#)”.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABTypedefs.h

ABSearchConjunction

Defines constants used to combine search elements.

```
enum _ABSearchConjunction {
    kABSearchAnd,
    kABSearchOr
};
typedef CFIndex ABSearchConjunction;
```

Discussion

These constants are described in “[Constants](#)” (page 77) in “[ABSearchElement](#)”.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABTypedefs.h

ABPeoplePickerSelectionBehavior

```
typedef enum {  
    ABNoValueSelection = 0,  
    ABSingleValueSelection = 1,  
    ABMultipleValueSelection = 2  
} ABPeoplePickerSelectionBehavior;
```

Discussion

These constants are described in “[Constants](#)” (page 57) in “[ABPeoplePickerView](#)”.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABPeoplePickerView.h

Constants

Address Book Objective-C Constants Reference

Framework:	AddressBook/AddressBook.h
Declared in	AddressBook/ABTypedefs.h AddressBook/ABGlobals.h

Overview

Constants

Default Multivalue List Labels

The default labels contained in the Address Book database for specifying different values in a multivalue list. Users can also add their own labels.

```

NSString * const kABHomePageLabel;
NSString * const kABEmailWorkLabel;
NSString * const kABEmailHomeLabel;
NSString * const kABEmailMobileMeLabel;
NSString * const kABAddressHomeLabel;
NSString * const kABAddressWorkLabel;
NSString * const kABAnniversaryLabel;
NSString * const kABFatherLabel;
NSString * const kABMotherLabel;
NSString * const kABParentLabel;
NSString * const kABBrotherLabel;
NSString * const kABSisterLabel;
NSString * const kABChildLabel;
NSString * const kABFriendLabel;
NSString * const kABSpouseLabel;
NSString * const kABPartnerLabel;
NSString * const kABAssistantLabel;
NSString * const kABManagerLabel;
NSString * const kABPhoneWorkLabel;
NSString * const kABPhoneHomeLabel;
NSString * const kABPhoneiPhoneLabel;
NSString * const kABPhoneMobileLabel;
NSString * const kABPhoneMainLabel;
NSString * const kABPhoneHomeFAXLabel;
NSString * const kABPhoneWorkFAXLabel;
NSString * const kABPhonePagerLabel;
NSString * const kABAIMWorkLabel;
NSString * const kABAIMHomeLabel;
NSString * const kABAIMMobileMeLabel;
NSString * const kABJabberWorkLabel;
NSString * const kABJabberHomeLabel;
NSString * const kABMSNWorkLabel;
NSString * const kABMSNHomeLabel;
NSString * const kABYahooWorkLabel;
NSString * const kABYahooHomeLabel;
NSString * const kABICQWorkLabel;
NSString * const kABICQHomeLabel;

```

Constants

kABHomePageLabel

Webpage URL.

Available in Mac OS X v10.4 and later.

Declared in ABGlobals.h.

kABEmailWorkLabel

Work email.

Available in Mac OS X v10.2 and later.

Declared in ABGlobals.h.

kABEmailHomeLabel

Home email.

Available in Mac OS X v10.2 and later.

Declared in ABGlobals.h.

- `kABEmailMobileMeLabel`
Mobile Me email.
Available in Mac OS X v10.6 and later.
Declared in `ABGlobals.h`.
- `kABAddressHomeLabel`
Home address.
Available in Mac OS X v10.2 and later.
Declared in `ABGlobals.h`.
- `kABAddressWorkLabel`
Work address.
Available in Mac OS X v10.2 and later.
Declared in `ABGlobals.h`.
- `kABAnniversaryLabel`
Anniversary date.
Available in Mac OS X v10.3 and later.
Declared in `ABGlobals.h`.
- `kABFatherLabel`
Father.
Available in Mac OS X v10.3 and later.
Declared in `ABGlobals.h`.
- `kABMotherLabel`
Mother.
Available in Mac OS X v10.3 and later.
Declared in `ABGlobals.h`.
- `kABParentLabel`
Parent.
Available in Mac OS X v10.3 and later.
Declared in `ABGlobals.h`.
- `kABBrotherLabel`
Brother.
Available in Mac OS X v10.3 and later.
Declared in `ABGlobals.h`.
- `kABSisterLabel`
Sister.
Available in Mac OS X v10.3 and later.
Declared in `ABGlobals.h`.
- `kABChildLabel`
Child.
Available in Mac OS X v10.3 and later.
Declared in `ABGlobals.h`.

- `kABFriendLabel`
Friend.
Available in Mac OS X v10.3 and later.
Declared in `ABGlobals.h`.
- `kABSpouseLabel`
Spouse.
Available in Mac OS X v10.3 and later.
Declared in `ABGlobals.h`.
- `kABPartnerLabel`
Partner.
Available in Mac OS X v10.3 and later.
Declared in `ABGlobals.h`.
- `kABAssistantLabel`
Assistant.
Available in Mac OS X v10.3 and later.
Declared in `ABGlobals.h`.
- `kABManagerLabel`
Manager.
Available in Mac OS X v10.3 and later.
Declared in `ABGlobals.h`.
- `kABPhoneWorkLabel`
Work phone number.
Available in Mac OS X v10.2 and later.
Declared in `ABGlobals.h`.
- `kABPhoneHomeLabel`
Home phone number.
Available in Mac OS X v10.2 and later.
Declared in `ABGlobals.h`.
- `kABPhoneiPhoneLabel`
iPhone phone number.
Available in Mac OS X v10.6 and later.
Declared in `ABGlobals.h`.
- `kABPhoneMobileLabel`
Cell phone number.
Available in Mac OS X v10.2 and later.
Declared in `ABGlobals.h`.
- `kABPhoneMainLabel`
Main phone number.
Available in Mac OS X v10.2 and later.
Declared in `ABGlobals.h`.

- `kABPhoneHomeFAXLabel`
Home fax number.
Available in Mac OS X v10.2 and later.
Declared in `ABGlobal.s.h`.
- `kABPhoneWorkFAXLabel`
Work fax number.
Available in Mac OS X v10.2 and later.
Declared in `ABGlobal.s.h`.
- `kABPhonePagerLabel`
Pager number.
Available in Mac OS X v10.2 and later.
Declared in `ABGlobal.s.h`.
- `kABAIMWorkLabel`
Work AOL instant messaging ID.
Available in Mac OS X v10.2 and later.
Declared in `ABGlobal.s.h`.
- `kABAIMHomeLabel`
Home AOL instant messaging ID.
Available in Mac OS X v10.2 and later.
Declared in `ABGlobal.s.h`.
- `kABAIMMobileMeLabel`
Mobile Me AOL instant messaging ID.
Available in Mac OS X v10.6 and later.
Declared in `ABGlobal.s.h`.
- `kABJabberWorkLabel`
Work Jabber instant messaging ID.
Available in Mac OS X v10.2 and later.
Declared in `ABGlobal.s.h`.
- `kABJabberHomeLabel`
Home Jabber instant messaging ID.
Available in Mac OS X v10.2 and later.
Declared in `ABGlobal.s.h`.
- `kABMSNWorkLabel`
Work MSN instant messaging ID.
Available in Mac OS X v10.2 and later.
Declared in `ABGlobal.s.h`.
- `kABMSNHomeLabel`
Home MSN instant messaging ID.
Available in Mac OS X v10.2 and later.
Declared in `ABGlobal.s.h`.

`kABYahooWorkLabel`

Work Yahoo instant messaging ID.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABYahooHomeLabel`

Home Yahoo instant messaging ID.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABICQWorkLabel`

Work ICQ instant messaging ID.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABICQHomeLabel`

Home ICQ instant messaging ID.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

Generic Multivalue List Labels

The generic labels contained in the Address Book database for specifying different values in a multivalue list. These labels are especially useful if there is no default label for a property. Users can also add their own labels.

```
NSString * const kABWorkLabel;
NSString * const kABHomeLabel;
NSString * const kABOtherLabel;
```

Constants

`kABWorkLabel`

All Work labels match this label.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABHomeLabel`

All Home labels match this label.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABOtherLabel`

All labels other than Home or Work labels match this label.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

Default Record Properties

Properties common to all types of records.

```
NSString * const kABUIDProperty;
NSString * const kABCreationDateProperty;
NSString * const kABModificationDateProperty;
```

Constants

`kABUIDProperty`

The unique ID for this record. It's guaranteed never to change, no matter how much the record changes. If you need to store a reference to a record, use this value. Type: [kABStringProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABCreationDateProperty`

The date when the record was first saved. Type: [kABDateProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABModificationDateProperty`

The date when the record was last saved. Type: [kABDateProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

Default Group Properties

The properties that a group record contains by default. Developers can add their own properties with the `ABGroup` method [addPropertiesAndTypes:](#) (page 24)

```
NSString * const kABGroupNameProperty;
```

Constants

`kABGroupNameProperty`

Name of the group

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

Default Person Properties

The properties that a person record contains by default. Developers can add their own properties with the `ABPerson` method [addPropertiesAndTypes:](#) (page 60).

```

NSString * const kABFirstNameProperty;
NSString * const kABLastNameProperty;
NSString * const kABFirstNamePhoneticProperty;
NSString * const kABLastNamePhoneticProperty;
NSString * const kABNicknameProperty;
NSString * const kABMaidenNameProperty;
NSString * const kABBirthdayProperty;
NSString * const kABOrganizationProperty;
NSString * const kABJobTitleProperty;
NSString * const kABHomePageProperty;
NSString * const kABURLsProperty;
NSString * const kABCalendarURIsProperty;
NSString * const kABEmailProperty;
NSString * const kABAddressProperty;
NSString * const kABOtherDatesProperty;
NSString * const kABRelatedNamesProperty;
NSString * const kABDepartmentProperty;
NSString * const kABPersonFlags;
NSString * const kABPhoneProperty;
NSString * const kABAIMInstantProperty;
NSString * const kABJabberInstantProperty;
NSString * const kABMSNInstantProperty;
NSString * const kABYahooInstantProperty;
NSString * const kABICQInstantProperty;
NSString * const kABNoteProperty;
NSString * const kABMiddleNameProperty;
NSString * const kABMiddleNamePhoneticProperty;
NSString * const kABTitleProperty;
NSString * const kABSuffixProperty;
NSString * const kABMiddleNameProperty;
NSString * const kABMiddleNamePhoneticProperty;
NSString * const kABTitleProperty;
NSString * const kABSuffixProperty;

```

Constants

kABFirstNameProperty

First name. Type: [kABStringProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in ABGlobals.h.

kABLastNameProperty

Last name. Type: [kABStringProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in ABGlobals.h.

kABFirstNamePhoneticProperty

Phonetic representation of the first name. Type: [kABStringProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in ABGlobals.h.

kABLastNamePhoneticProperty

Phonetic representation of the last name. Type: [kABStringProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in ABGlobals.h.

kABNicknameProperty

Nickname. Type: [kABStringProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

kABMaidenNameProperty

Maiden name. Type: [kABStringProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

kABBirthdayProperty

Birth date. Type: [kABDateProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

kABOrganizationProperty

Company name. Type: [kABStringProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

kABJobTitleProperty

Job title. Type: [kABStringProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

kABHomePageProperty

Home web page. Type: [kABStringProperty](#) (page 114). (**Deprecated.** Use [kABURLsProperty](#) instead.)

The `kABHomePageProperty` property has been deprecated in Mac OS X version 10.4. Instead, use the `kABURLsProperty` multivalue property with the label `kABHomePageLabel`.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

kABURLsProperty

Web pages. Type: [kABMultiStringProperty](#) (page 115).

Available in Mac OS X v10.4 and later.

Declared in `ABGlobals.h`.

kABCalendarURIsProperty

Calendar URIs. Type: [kABMultiStringProperty](#) (page 115).

Available in Mac OS X v10.5 and later.

Declared in `ABGlobals.h`.

kABEmailProperty

Email addresses. Type: [kABMultiStringProperty](#) (page 115).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABAddressProperty`

Street addresses. Type: [kABMultiDictionaryProperty](#) (page 115).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABOtherDatesProperty`

Dates associated with a person. Type: [kABMultiDateProperty](#) (page 115).

Available in Mac OS X v10.3 and later.

Declared in `ABGlobals.h`.

`kABRelatedNamesProperty`

Names of people related to a person. Type: [kABMultiStringProperty](#) (page 115).

Available in Mac OS X v10.3 and later.

Declared in `ABGlobals.h`.

`kABDepartmentProperty`

Department name. Type: [kABStringProperty](#) (page 114).

Available in Mac OS X v10.3 and later.

Declared in `ABGlobals.h`.

`kABPersonFlags`

Property that specifies the name ordering and configuration of a record in the Address Book application. See [“Person Flags”](#) (page 66).

Available in Mac OS X v10.3 and later.

Declared in `ABGlobals.h`.

`kABPhoneProperty`

Generic phone number. Type: [kABMultiStringProperty](#) (page 115).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABAIMInstantProperty`

AOL instant messaging ID. Type: [kABMultiStringProperty](#) (page 115).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABJabberInstantProperty`

Jabber instant messaging ID. Type: [kABMultiStringProperty](#) (page 115).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABMSNInstantProperty`

MSN instant messaging ID. Type: [kABMultiStringProperty](#) (page 115).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABYahooInstantProperty`

Yahoo instant messaging ID. Type: [kABMultiStringProperty](#) (page 115).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABICQInstantProperty`

ICQ instant messaging ID. Type: [kABMultiStringProperty](#) (page 115).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABNoteProperty`

Notes. Type: [kABStringProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABMiddleNameProperty`

Middle name. Type: [kABStringProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABMiddleNamePhoneticProperty`

Phonetic representation of the middle name. Type: [kABStringProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABTitleProperty`

Title, such as “Mr.,” “Mrs.,” “General,” “Cardinal,” or “Lord.” Type: [kABStringProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABSuffixProperty`

Suffix, such as “Sr.,” “Jr.,” “III.,” or “Esq.” Type: [kABStringProperty](#) (page 114).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

Multivalue Property

A multivalue property type.

```
#define kABMultiValueMask 0x100
```

Constants

`kABMultiValueMask`

Used by [ABPropertyType](#) (page 97) to specify a multivalue property.

Available in Mac OS X v10.2 and later.

Declared in `ABTypeDefs.h`.

Property Types

The possible [ABPropertyType](#) (page 97) types for `ABRecord` properties:

```

enum _ABPropertyType {
kABErrorInProperty          = 0x0,
kABStringProperty           = 0x1,
kABIntegerProperty          = 0x2,
kABRealProperty             = 0x3,
kABDateProperty             = 0x4,
kABArrayProperty            = 0x5,
kABDictionaryProperty       = 0x6,
kABDataProperty             = 0x7,
kABMultiStringProperty      = kABMultiValueMask | kABStringProperty,
kABMultiIntegerProperty     = kABMultiValueMask | kABIntegerProperty,
kABMultiRealProperty        = kABMultiValueMask | kABRealProperty,
kABMultiDateProperty        = kABMultiValueMask | kABDateProperty,
kABMultiArrayProperty       = kABMultiValueMask | kABArrayProperty,
kABMultiDictionaryProperty  = kABMultiValueMask | kABDictionaryProperty,
kABMultiDataProperty        = kABMultiValueMask | kABDataProperty
};

```

Constants

kABErrorInProperty

An invalid property was used.

Available in Mac OS X v10.2 and later.

Declared in ABTypedefs.h.

kABStringProperty

This property is an NSString object.

Available in Mac OS X v10.2 and later.

Declared in ABTypedefs.h.

kABIntegerProperty

This property is an NSNumber object representing an integer.

Available in Mac OS X v10.2 and later.

Declared in ABTypedefs.h.

kABRealProperty

This property is an NSNumber object representing a real number.

Available in Mac OS X v10.2 and later.

Declared in ABTypedefs.h.

kABDateProperty

This property is an NSDate object.

Available in Mac OS X v10.2 and later.

Declared in ABTypedefs.h.

kABArrayProperty

This property is an NSArray object.

Available in Mac OS X v10.2 and later.

Declared in ABTypedefs.h.

kABDictionaryProperty

This property is an NSDictionary object.

Available in Mac OS X v10.2 and later.

Declared in ABTypedefs.h.

`kABDataProperty`

This property is an `NSData` object.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABMultiStringProperty`

This property is an `ABMultiValue` object containing `NSString` objects.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABMultiIntegerProperty`

This property is an `ABMultiValue` object containing `NSNumber` objects representing integers.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABMultiRealProperty`

This property is an `ABMultiValue` object containing `NSNumber` objects representing real numbers.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABMultiDateProperty`

This property is an `ABMultiValue` object containing `NSDate` objects.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABMultiArrayProperty`

This property is an `ABMultiValue` object containing `NSArray` objects.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABMultiDictionaryProperty`

This property is an `ABMultiValue` object containing `NSDictionary` objects.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABMultiDataProperty`

This property is an `ABMultiValue` object containing `NSData` objects.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

Address Keys

The keys used to specify the different fields in a `KABAddressProperty`. Neither developers nor users can add more keys.

```
NSString * const kABAddressStreetKey;  
NSString * const kABAddressCityKey;  
NSString * const kABAddressStateKey;  
NSString * const kABAddressZIPKey;  
NSString * const kABAddressCountryKey;  
NSString * const kABAddressCountryCodeKey;
```

Constants

kABAddressStreetKey

Street.

Available in Mac OS X v10.2 and later.

Declared in ABGlobals.h.

kABAddressCityKey

City.

Available in Mac OS X v10.2 and later.

Declared in ABGlobals.h.

kABAddressStateKey

State.

Available in Mac OS X v10.2 and later.

Declared in ABGlobals.h.

kABAddressZIPKey

Zip.

Available in Mac OS X v10.2 and later.

Declared in ABGlobals.h.

kABAddressCountryKey

Country, identified by ISO country code. For a list of these codes, see Country Codes.

Available in Mac OS X v10.2 and later.

Declared in ABGlobals.h.

kABAddressCountryCodeKey

Country Code.

Available in Mac OS X v10.2 and later.

Declared in ABGlobals.h.

Document Revision History

This table describes the changes to *Address Book Objective-C Framework Reference*.

Date	Notes
2007-07-08	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a collection of separate documents.

REVISION HISTORY

Document Revision History

Index

A

ABLocalizedPropertyOrLabel **function** 93
ABMultipleValueSelection **constant** 58
ABNoValueSelection **constant** 57
ABPeoplePickerDisplayedPropertyDidChange-
Notification **notification** 58
ABPeoplePickerGroupSelectionDidChangeNotification
notification 58
ABPeoplePickerNameSelectionDidChangeNotification
notification 58
ABPeoplePickerSelectionBehavior **data type** 99
ABPeoplePickerValueSelectionDidChangeNotification
notification 58
ABPropertyType **data type** 97
ABSearchComparison **data type** 98
ABSearchConjunction **data type** 98
ABSinglValueSelection **constant** 57
accessoryView **instance property** 47
actionProperty <NSObject> **instance method** 86
addMember: **instance method** 27
addPropertiesAndTypes: **class method** 24, 60
addProperty: **instance method** 50
addRecord: **instance method** 14
Address Keys 115
addressBook **class method** 13
addSubgroup: **instance method** 28
addValue:withLabel: **instance method** 40
allowsGroupSelection **instance property** 47
allowsMultipleSelection **instance property** 48
autosaveName **instance property** 48

B

beginLoadingImageDataForClient: **instance method**
63

C

cancelLoadingImageDataForTag: **class method** 61
clearSearchField: **instance method** 51
columnTitleForProperty: **instance method** 51
consumeImageData:forTag: **protocol instance method**
89
count **instance method** 34

D

Database change notification keys 21
Default Group Properties 109
Default Multivalue List Labels 103
Default Person Properties 109
Default Record Properties 108
defaultCountryCode **instance method** 15
defaultNameOrdering **instance method** 15
deselectAll: **instance method** 52
deselectGroup: **instance method** 52
deselectIdentifier:forPerson: **instance method**
52
deselectRecord: **instance method** 53
displayedProperty **instance property** 48
distributionIdentifierForProperty:person:
instance method 28

E

editInAddressBook: **instance method** 53

F

formattedAddressFromDictionary: **instance method**
15

G

Generic Multivalue List Labels [108](#)
 groupDoubleAction [instance property](#) [49](#)
 groups [instance method](#) [16](#)

H

hasUnsavedChanges [instance method](#) [16](#)

I

identifierAtIndex: [instance method](#) [35](#)
 imageData [instance method](#) [64](#)
 indexOfIdentifier: [instance method](#) [35](#)
 init [instance method](#) [70](#)
 initWithAddressBook: [instance method](#) [70](#)
 initWithVCardRepresentation: [instance method](#) [64](#)
 insertValue:withLabel:atIndex: [instance method](#)
[41](#)
 isReadOnly [instance method](#) [71](#)

K

kABAddressCityKey [constant](#) [116](#)
 kABAddressCountryCodeKey [constant](#) [116](#)
 kABAddressCountryKey [constant](#) [116](#)
 kABAddressHomeLabel [constant](#) [105](#)
 kABAddressProperty [constant](#) [112](#)
 kABAddressStateKey [constant](#) [116](#)
 kABAddressStreetKey [constant](#) [116](#)
 kABAddressWorkLabel [constant](#) [105](#)
 kABAddressZIPKey [constant](#) [116](#)
 kABAIMHomeLabel [constant](#) [107](#)
 kABAIMInstantProperty [constant](#) [112](#)
 kABAIMMobileMeLabel [constant](#) [107](#)
 kABAIMWorkLabel [constant](#) [107](#)
 kABAnniversaryLabel [constant](#) [105](#)
 kABArrayProperty [constant](#) [114](#)
 kABAssistantLabel [constant](#) [106](#)
 kABBirthdayProperty [constant](#) [111](#)
 kABBitsInBitFieldMatch [constant](#) [79](#)
 kABBrotherLabel [constant](#) [105](#)
 kABCalendarURIsProperty [constant](#) [111](#)
 kABChildLabel [constant](#) [105](#)
 kABContainsSubString [constant](#) [79](#)
 kABContainsSubStringCaseInsensitive [constant](#)
[79](#)

kABCreationDateProperty [constant](#) [109](#)
 kABDatabaseChangedExternallyNotification
[notification](#) [22](#)
 kABDatabaseChangedNotification [notification](#) [21](#)
 kABDataProperty [constant](#) [115](#)
 kABDateProperty [constant](#) [114](#)
 kABDefaultNameOrdering [constant](#) [67](#)
 kABDeletedRecords [constant](#) [21](#)
 kABDepartmentProperty [constant](#) [112](#)
 kABDictionaryProperty [constant](#) [114](#)
 kABDoesNotContainSubString [constant](#) [79](#)
 kABDoesNotContainSubStringCaseInsensitive
[constant](#) [79](#)
 kABEmailHomeLabel [constant](#) [104](#)
 kABEmailMobileMeLabel [constant](#) [105](#)
 kABEmailProperty [constant](#) [111](#)
 kABEmailWorkLabel [constant](#) [104](#)
 kABEqual [constant](#) [78](#)
 kABEqualCaseInsensitive [constant](#) [79](#)
 kABErrorInProperty [constant](#) [114](#)
 kABFatherLabel [constant](#) [105](#)
 kABFirstNameFirst [constant](#) [67](#)
 kABFirstNamePhoneticProperty [constant](#) [110](#)
 kABFirstNameProperty [constant](#) [110](#)
 kABFriendLabel [constant](#) [106](#)
 kABGreaterThan [constant](#) [78](#)
 kABGreaterThanOrEqual [constant](#) [79](#)
 kABGroupNameProperty [constant](#) [109](#)
 kABHomeLabel [constant](#) [108](#)
 kABHomePageLabel [constant](#) [104](#)
 kABHomePageProperty [constant](#) [111](#)
 kABICQHomeLabel [constant](#) [108](#)
 kABICQInstantProperty [constant](#) [113](#)
 kABICQWorkLabel [constant](#) [108](#)
 kABInsertedRecords [constant](#) [21](#)
 kABIntegerProperty [constant](#) [114](#)
 kABJabberHomeLabel [constant](#) [107](#)
 kABJabberInstantProperty [constant](#) [112](#)
 kABJabberWorkLabel [constant](#) [107](#)
 kABJobTitleProperty [constant](#) [111](#)
 kABLastNameFirst [constant](#) [67](#)
 kABLastNamePhoneticProperty [constant](#) [110](#)
 kABLastNameProperty [constant](#) [110](#)
 kABLessThan [constant](#) [78](#)
 kABLessThanOrEqual [constant](#) [78](#)
 kABMaidenNameProperty [constant](#) [111](#)
 kABManagerLabel [constant](#) [106](#)
 kABMiddleNamePhoneticProperty [constant](#) [113](#)
 kABMiddleNameProperty [constant](#) [113](#)
 kABModificationDateProperty [constant](#) [109](#)
 kABMotherLabel [constant](#) [105](#)
 kABMSNHomeLabel [constant](#) [107](#)
 kABMSNInstantProperty [constant](#) [112](#)

kABMSNWorkLabel **constant** 107
 kABMultiArrayProperty **constant** 115
 kABMultiDataProperty **constant** 115
 kABMultiDateProperty **constant** 115
 kABMultiDictionaryProperty **constant** 115
 kABMultiIntegerProperty **constant** 115
 kABMultiRealProperty **constant** 115
 kABMultiStringProperty **constant** 115
 kABMultiValueMask **constant** 113
 kABNameOrderingMask **constant** 67
 kABNicknameProperty **constant** 111
 kABNoteProperty **constant** 113
 kABNotEqual **constant** 78
 kABNotEqualCaseInsensitive **constant** 79
 kABNotWithinIntervalAroundToday **constant** 80
 kABNotWithinIntervalAroundTodayYearless **constant** 80
 kABNotWithinIntervalFromToday **constant** 80
 kABNotWithinIntervalFromTodayYearless **constant** 80
 kABOrganizationProperty **constant** 111
 kABOtherDatesProperty **constant** 112
 kABOtherLabel **constant** 108
 kABParentLabel **constant** 105
 kABPartnerLabel **constant** 106
 kABPersonFlags **constant** 112
 kABPhoneHomeFAXLabel **constant** 107
 kABPhoneHomeLabel **constant** 106
 kABPhoneiPhoneLabel **constant** 106
 kABPhoneMainLabel **constant** 106
 kABPhoneMobileLabel **constant** 106
 kABPhonePagerLabel **constant** 107
 kABPhoneProperty **constant** 112
 kABPhoneWorkFAXLabel **constant** 107
 kABPhoneWorkLabel **constant** 106
 kABPrefixMatch **constant** 79
 kABPrefixMatchCaseInsensitive **constant** 79
 kABRealProperty **constant** 114
 kABRelatedNamesProperty **constant** 112
 kABSearchAnd **constant** 77
 kABSearchOr **constant** 77
 kABShowAsCompany **constant** 66
 kABShowAsMask **constant** 66
 kABShowAsPerson **constant** 66
 kABShowAsResource **constant** 67
 kABShowAsRoom **constant** 67
 kABSisterLabel **constant** 105
 kABSpouseLabel **constant** 106
 kABStringProperty **constant** 114
 kABSuffixMatch **constant** 80
 kABSuffixMatchCaseInsensitive **constant** 80
 kABSuffixProperty **constant** 113
 kABTitleProperty **constant** 113

kABUIDProperty **constant** 109
 kABUpdatedRecords **constant** 21
 kABURLsProperty **constant** 111
 kABWithinIntervalAroundToday **constant** 80
 kABWithinIntervalAroundTodayYearless **constant** 80
 kABWithinIntervalFromToday **constant** 80
 kABWithinIntervalFromTodayYearless **constant** 80
 kABWorkLabel **constant** 108
 kABYahooHomeLabel **constant** 108
 kABYahooInstantProperty **constant** 112
 kABYahooWorkLabel **constant** 108

L

labelAtIndex: **instance method** 35
 labelForIdentifier: **instance method** 36

M

matchesRecord: **instance method** 76
 me **instance method** 17
 members **instance method** 29
 Multivalue Property 113

N

nameDoubleAction **instance property** 49

P

parentGroups **instance method** 29, 65
 people **instance method** 17
 performActionForPerson:identifier: <NSObject>
 instance method 86
 Person Flags 66
 primaryIdentifier **instance method** 36
 properties **class method** 25, 61
 properties **instance method** 53
 Property Types 113
 propertyType **instance method** 37

R

recordClassFromUniqueId: **instance method** 18

recordForUniqueId: **instance method** 18
 recordsMatchingSearchElement: **instance method** 18
 removeMember: **instance method** 29
 removeProperties: **class method** 25, 62
 removeProperty: **instance method** 54
 removeRecord: **instance method** 19
 removeSubgroup: **instance method** 30
 removeValueAndLabelAtIndex: **instance method** 42
 removeValueForProperty: **instance method** 71
 replaceLabelAtIndex:withLabel: **instance method** 42
 replaceValueAtIndex:withValue: **instance method** 42

S

save **instance method** 19
 saveAndReturnError: **instance method** 20
Search Comparisons 77
Search Conjunctions 77
 searchElementForConjunction:children: **class method** 76
 searchElementForProperty:label:key:value: comparison: **class method** 26, 62
 selectedGroups **instance property** 49
 selectedIdentifiersForPerson: **instance method** 54
 selectedRecords **instance property** 49
 selectedValues **instance method** 55
 selectGroup:byExtendingSelection: **instance method** 55
 selectIdentifier:forPerson:byExtendingSelection: **instance method** 55
 selectInAddressBook: **instance method** 56
Selection Behavior 57
 selectRecord:byExtendingSelection: **instance method** 56
 setColumnName:forProperty: **instance method** 57
 setDistributionIdentifier:forProperty:person: **instance method** 31
 setImageData: **instance method** 65
 setMe: **instance method** 20
 setPrimaryIdentifier: **instance method** 43
 setValue:forProperty: **instance method** 71
 sharedAddressBook **class method** 14
 shouldEnableActionForPerson:identifier: <NSObject> **instance method** 86
 subgroups **instance method** 31

T

target **instance property** 50
 titleForPerson:identifier: <NSObject> **instance method** 87
 typeOfProperty: **class method** 26, 63

U

uniqueId **instance method** 72

V

valueAtIndex: **instance method** 37
 valueForIdentifier: **instance method** 37
 valueForProperty: **instance method** 73
 valueSelectionBehavior **instance property** 50
 vCardRepresentation **instance method** 66