Address Book C Framework Reference

Apple Applications > Address Book

2006-05-23

Apple Inc.
© 2006 Apple Computer, 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, Carbon, Cocoa, iChat, Mac, Mac OS, Objective-C, 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

Framework	/System/Library/Frameworks/AddressBook.framework
Header file directories	/System/Library/Frameworks/AddressBook.framework/Headers
Declared in	ABActionsC.h ABAddressBookC.h ABGlobals.h ABGlobalsC.h ABPeoplePickerC.h ABTypedefs.h

The Address Book is a centralized database for contact and other personal information for people. Applications that support the Address Book framework share this contact information with other applications, include Apple's Mail and iChat. Both Carbon and Cocoa applications can access it.

Opaque Types

ABAddressBook C Reference

Derived From:	CFType
Framework:	AddressBook/ABAddressBookC.h
Declared in	ABAddressBookC.h ABGlobalsC.h
Companion guide	Address Book Programming Guide for Mac OS X

Overview

The ABAddressBook opaque type 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.

The ABAddressBook opaque type provides functions for accessing, adding, and removing group and person records including the "me" record corresponding to the logged-in user. For example, you use the `ABCopyArrayOfAllGroups` (page 11) function to get an array of all the group records in the database, or the `ABCopyArrayOfAllPeople` (page 11) function to get all the person records. You use the `ABGetMe` (page 14) function to get the person record corresponding to the logged-in user. You can also add and remove records using the `ABAddRecord` (page 10) and `ABRemoveRecord` (page 16) functions.

You can also search for records matching a particular query you specify by creating an ABSearchElement object. You use the `ABGroupCreateSearchElement` (page 27) or `ABPersonCreateSearchElement` (page 47) function to create an ABSearchElement object for the corresponding record. Then use the `ABCopyArrayOfMatchingValues` (page 12) ABAddressBook function, passing the ABSearchElement as the argument, to query the database. See ABSearchElement for more functions that create compound queries.

Your application uses a shared instance of ABAddressBook returned by the `ABGetSharedAddressBook` (page 15) function to interact with the database (multiple ABAddressBook instances are not supported). Changes you make to the record objects are stored in memory, and saved to disk when you invoke the `ABSave` (page 17) function.

The Address Book posts notifications if any application including yours makes changes to the database. Typically, you observe these notifications to update any dependent view or model objects in your application. Use CFNotificationCenter to register for the ABAddressBook notifications: `kABDatabaseChangedNotification` (page 21) and `kABDatabaseChangedExternallyNotification` (page 21). These notifications are not sent until `ABGetSharedAddressBook` (page 15) is called.

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

Functions

ABAddPropertiesAndTypes

Adds the given properties to all the records of the specified type in the Address Book database, and returns the number of properties successfully added.

```
CFIndex ABAddPropertiesAndTypes (
   ABAddressBookRef addressBook,
   CFStringRef recordType,
   CFDictionaryRef propertiesAnTypes
);
```

Parameters

addressBook

The address book for the logged-in user.

recordType

The record type you wish to add properties to: `kABGroupRecordType` or `kABPersonRecordType`.

propertiesAndTypes

A CFDictionary object containing the properties to add. 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 18).

Return Value

The number of properties successfully added.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABAddRecord

Adds a record of the specified type to the Address Book database.

```
bool ABAddRecord (
   ABAddressBookRef addressBook,
   ABRecordRef record
);
```

Parameters

addressBook

The address book for the logged-in user.

record

The record to add to the Address Book database. If this parameter is `NULL`, the function raises an exception.

Return Value

`true` if the record was added successfully, `false` otherwise.

Availability

Available in Mac OS X v10.2 and later.

Related Sample Code

AddressBookCarbon

Declared In

`ABAddressBookC.h`

ABCopyArrayOfAllGroups

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

```
CFArrayRef ABCopyArrayOfAllGroups (
   ABAddressBookRef addressBook
);
```

Parameters

addressBook

The address book for the logged-in user.

Return Value

An array of ABGroup objects representing all the groups in the Address Book database. If the database doesn't contain any groups, the function returns an empty array. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ABAddressBookC.h`

ABCopyArrayOfAllPeople

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

```
CFArrayRef ABCopyArrayOfAllPeople (
   ABAddressBookRef addressBook
);
```

Parameters

addressBook

The address book for the logged-in user.

Return Value

An array of ABPerson objects representing all the people in the Address Book database. If the database does not contain any people, the function returns an empty array. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ABAddressBookC.h`

ABCopyArrayOfMatchingRecords

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

```
CFArrayRef ABCopyArrayOfMatchingRecords (
   ABAddressBookRef addressBook,
   ABSearchElementRef search
);
```

Parameters

addressBook

The address book for the logged-in user.

search

The search element that specifies the query. If *search* is NULL, this function raises an exception. Create an ABSearchElement object using the record specific functions: ABGroupCreateSearchElement (page 27) or ABPersonCreateSearchElement (page 47). See ABSearchElement for more functions that create compound queries.

Return Value

A new array containing ABRecord objects representing all the records that match *search*. If no records match *search*, this function returns an empty array. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.2 and later.

Related Sample Code

AddressBookCarbon

Declared In

ABAddressBookC.h

ABCopyArrayOfPropertiesForRecordType

Returns an array containing the names of all the properties for the specified record type.

```
CFArrayRef ABCopyArrayOfPropertiesForRecordType (
   ABAddressBookRef addressBook,
   CFStringRef recordType
);
```

Parameters

addressBook

The address book for the logged-in user.

recordType

Specifies the type of record: kABGroupRecordType or kABPersonRecordType.

Return Value

An new array containing the names (CFString objects) of all the properties in *recordType*. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

ABCopyDefaultCountryCode

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

```
CFStringRef ABCopyDefaultCountryCode (
   ABAddressBookRef addressBook
);
```

Parameters

addressBook

The address book for the logged-in user.

Return Value
A string with the default country code. You are responsible for releasing this object.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABAddressBookC.h`

ABCopyRecordForUniqueId

Returns the record that matches the given unique ID.

```
ABRecordRef ABCopyRecordForUniqueId (
   ABAddressBookRef addressBook,
   CFStringRef uniqueId
);
```

Parameters

addressBook

The address book for the logged-in user.

uniqueId

A unique ID for the record. If this is `NULL`, this function raises an exception.

Return Value
The record that matches the given unique ID. If no record matches *uniqueId*, the function returns `NULL`. You are responsible for releasing this object.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

ABCopyRecordTypeFromUniqueId

Returns the type name of the record that matches a given unique ID.

```
CFStringRef ABCopyRecordTypeFromUniqueId (
   ABAddressBookRef addressBook,
   CFStringRef uniqueId
);
```

Parameters

addressBook

The address book for the logged-in user.

uniqueId

A unique ID for the record. If this is NULL, this function raises an exception.

Return Value

A string with the name of the type for the record that matches the given unique ID. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABAddressBookC.h

ABCreateFormattedAddressFromDictionary

Returns a string containing the formatted address.

```
CFStringRef ABCreateFormattedAddressFromDictionary (
   ABAddressBookRef addressBook,
   CFDictionaryRef address
);
```

Parameters

addressBook

The address book for the logged-in user.

Return Value

Returns a string containing the formatted address. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABAddressBookC.h

ABGetMe

Returns the ABPerson object for the logged-in user.

```
ABPersonRef ABGetMe (
   ABAddressBookRef addressBook
);
```

Parameters

addressBook

The address book for the logged-in user.

Return Value
The ABPerson record that represents the logged-in user, or `NULL` if the user never specified such a record. You are responsible for retaining and releasing this object as needed.

Availability
Available in Mac OS X v10.2 and later.

Related Sample Code
MyFirstJNIProject

Declared In
`ABAddressBookC.h`

ABGetSharedAddressBook

Returns the unique shared ABAddressBook object.

```
ABAddressBookRef ABGetSharedAddressBook (
   void
);
```

Return Value
The address book for the logged-in user. You are responsible for retaining and releasing this object as needed.

Discussion
Every application shares the address book for the logged-in user and this function returns it. If you call this function more than once or try to create a new address book, you get a pointer to the same shared address book.

Availability
Available in Mac OS X v10.2 and later.

Related Sample Code
AddressBookCarbon

MyFirstJNIProject

Declared In
`ABAddressBookC.h`

ABHasUnsavedChanges

Returns whether if there are unsaved changes in the address book.

```
bool ABHasUnsavedChanges (
   ABAddressBookRef addressBook
);
```

Parameters

addressBook

The address book for the logged-in user.

Return Value
`true` if there are unsaved changes, `false` otherwise.

Discussion

The unsaved changes flag is set automatically whenever changes are made to the address book.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABRemoveProperties

Removes the given properties from all the records of this type in the Address Book database, and returns the number of properties successfully removed.

```
CFIndex ABRemoveProperties (
   ABAddressBookRef addressBook,
   CFStringRef recordType,
   CFArrayRef properties
);
```

Parameters

addressBook

The address book for the logged-in user.

recordType

The name of record to remove the properties from: `kABGroupRecordType` or `kABPersonRecordType`.

properties

An array of properties (CFString objects) to remove.

Return Value

The number of properties successfully removed.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABRemoveRecord

Removes the specified record from the Address Book database.

```
bool ABRemoveRecord (
   ABAddressBookRef addressBook,
   ABRecordRef record
);
```

Parameters

addressBook

The address book for the logged-in user.

record

The ABRecord object to be removed. If `NULL`, this function raises an exception.

Return Value

`true` if the record was removed successfully, `false` otherwise.

Availability

Available in Mac OS X 10.2 and later.

Declared In

`ABAddressBookC.h`

ABSave

Saves all the changes made since the last save.

```
bool ABSave (
   ABAddressBookRef addressBook
);
```

Parameters

addressBook

The address book for the logged-in user.

Return Value

`true` if this function is successful or if there were no changes, `false` otherwise.

Availability

Available in Mac OS X v10.2 and later.

Related Sample Code

AddressBookCarbon

Declared In

`ABAddressBookC.h`

ABSetMe

Sets the record that represents the logged-in user.

```
void ABSetMe (
   ABAddressBookRef addressBook,
   ABPersonRef moi
);
```

Parameters

addressBook

The address book for the logged-in user.

person

The ABPerson object that represents the logged-in user. Pass `NULL` if you don't want a record to represent the logged-in user.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABTypeOfProperty

Returns the type of a given property for a given record.

```
ABPropertyType ABTypeOfProperty (
   ABAddressBookRef addressBook,
   CFStringRef recordType,
   CFStringRef property
);
```

Parameters

addressBook

The address book for the logged-in user.

recordType

The record type that contains *property*: kABGroupRecordType or kABPersonRecordType.

property

The property whose type you wish to obtain.

Return Value

The type of *property* as defined in Property Types (page 18). If *property* does not exist in *recordType*, this function returns kABErrorInProperty.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABAddressBookC.h

Data Types

ABAddressBookRef

A reference to an ABAddressBook object.

```
typedef struct __ABAddressBookRef *ABAddressBookRef;
```

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABAddressBookC.h

Constants

Property Types

These are the possible types of ABRecord properties.

```
typedef enum _ABPropertyType {
    kABErrorInProperty = 0x0,
    kABStringProperty = 0x1,
    kABIntegerProperty = 0x2,
    kABRealProperty = 0x3,
    kABDateProperty = 0x4,
    kABArrayProperty = 0x5,
    kABDictionaryProperty = 0x6,
    kABDataProperty = 0x7,
    kABMultiStringProperty = 0x100 | kABStringProperty,
    kABMultiIntegerProperty = 0x100 | kABIntegerProperty,
    kABMultiRealProperty = 0x100 | kABRealProperty,
    kABMultiDateProperty = 0x100 | kABDateProperty,
    kABMultiArrayProperty = 0x100 | kABArrayProperty,
    kABMultiDictionaryProperty = 0x100 | kABDictionaryProperty,
    kABMultiDataProperty = 0x100 | kABDataProperty
} ABPropertyType;
```

Constants

`kABErrorInProperty`

Returned by some functions when an invalid property is used.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABStringProperty`

Indicates a CFString object.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABIntegerProperty`

Indicates a CFNumber object representing an integer.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABRealProperty`

Indicates a CFNumber object representing a real number.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABDateProperty`

Indicates a CFDate object.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABArrayProperty`

Indicates a CFArray object.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABDictionaryProperty`

Indicates a CFDictionary object.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABDataProperty`
Indicates a CFData object.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABMultiStringProperty`
Indicates an ABMultiValue containing NSString objects.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABMultiIntegerProperty`
Indicates an ABMultiValue containing NSNumber objects representing integers.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABMultiRealProperty`
Indicates an ABMultiValue containing NSNumber objects representing real numbers.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABMultiDateProperty`
Indicates an ABMultiValue containing NSDate objects.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABMultiArrayProperty`
Indicates an ABMultiValue containing NSArray objects.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABMultiDictionaryProperty`
Indicates an ABMultiValue containing NSDictionary objects.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

`kABMultiDataProperty`
Indicates an ABMultiValue containing NSData objects.

Available in Mac OS X v10.2 and later.

Declared in `ABTypedefs.h`.

Database Notifications

These are notifications published when something changes in the Address Book database. These notifications are not sent until `ABGetSharedAddressBook` (page 15) has been called.

```
CFStringRef kABDatabaseChangedNotification;
CFStringRef kABDatabaseChangedExternallyNotification;
```

Constants

`kABDatabaseChangedNotification`

This process has changed the AddressBook database.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABDatabaseChangedExternallyNotification`

Another process has changed the AddressBook database. The following keys are included in the user-info dictionary of the notification: `kABInsertedRecords`, `kABUpdatedRecords`, and `kABDeletedRecords`. If the values for all the keys are `nil`, everything has changed, such as when the Address Book database is restored from a backup copy.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

ABGroup Reference for C

Derived From:	ABRecord : CFType
Framework:	AddressBook/ABAddressBookC.h
Declared in	ABAddressBookC.h ABGlobalsC.h
Companion guide	Address Book Programming Guide for Mac OS X

Overview

The ABGroup opaque type 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 as long as the relationship does not cause 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 allow you to create email distribution lists.

Use the `ABGroupCopyArrayOfAllMembers` (page 25) function to get all the members of a group, use the `ABGroupAddMember` (page 24) function to add people to a group, and the `ABGroupRemoveMember` (page 28) function to remove people from a group. Use the `ABGroupAddGroup` (page 24) function to create a subgroup.

Use the ABAddressBook `ABAddPropertiesAndTypes` (page 10) function to add additional program-defined properties to group records. Because the Address Book database is stored as a property list, these program-defined properties may be ignored by other applications. Note that the Address Book database is accessed by multiple application and is not encrypted so your application should not store any sensitive information in the database.

You can also search for records matching a particular "query" you specify by creating an ABSearchElement object. Use the `ABGroupCreateSearchElement` (page 27) function to create an ABSearchElement object containing your query. Then use the ABAddressBook `ABCopyArrayOfMatchingRecords` (page 12) function, passing the ABSearchElement as the argument, to query the database. See ABSearchElement for functions that create compound queries.

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

Functions

ABGroupAddGroup

Adds a subgroup to another group.

```
bool ABGroupAddGroup (
   ABGroupRef group,
   ABGroupRef groupToAdd
);
```

Parameters

group

The group you wish to add a subgroup to. If `NULL`, this function raises an exception.

groupToAdd

The subgroup you wish to add to *group*.

Return Value

Returns `true` if successful. If the *group* argument is already part of the receiver, this function does nothing and returns `false`. If adding the group would create a recursion, this function also does nothing and returns `false`. 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 function won’t allow.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABGroupAddMember

Adds a person to a group.

```
bool ABGroupAddMember (
   ABGroupRef group,
   ABPersonRef personToAdd
);
```

Parameters

group

The group you wish to add *person* to.

person

The person to add to *group*. If *person* is `NULL`, this function raises an exception.

Return Value

`true` if successful, false otherwise. For example, if *person* is already in *group*, this function does nothing but returns *false*.

Availability

Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

ABGroupCopyArrayOfAllMembers

Returns an array of persons in a group.

```
CFArrayRef ABGroupCopyArrayOfAllMembers (
   ABGroupRef group
);
```

Parameters

group

The ABGroup object whose members you wish to obtain.

Return Value
An array of ABPerson objects representing the people in *group*. If this group doesn't contain any people, this function returns an empty array. You are responsible for releasing this object.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

ABGroupCopyArrayOfAllSubgroups

Returns an array containing a group's subgroups.

```
CFArrayRef ABGroupCopyArrayOfAllSubgroups (
   ABGroupRef group
);
```

Parameters

group

The ABGroup object whose subgroups you wish to obtain.

Return Value
An array of ABGroup objects representing the subgroups of *group*. If *group* doesn't contain any groups, this function returns an empty array. You are responsible for releasing this object.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

ABGroupCopyDistributionIdentifier

Returns the distribution identifier for the given property and person.

```
CFStringRef ABGroupCopyDistributionIdentifier (
   ABGroupRef group,
   ABPersonRef person,
   CFStringRef property
);
```

Parameters

group

The group object that *person* belongs to.

person

A person object whose distribution identifier you want to obtain.

property

The name of a person's multi-value list property whose distribution identifier you want to obtain.

Return Value

The distribution identifier for *person* and *property* if it was set, otherwise returns the property's primary identifier. If either *person* or *property* are NULL, this function returns NULL. Also, returns NULL if *property* is not a multi-value list property. You are responsible for releasing this object.

Discussion

Use the ABGroupSetDistributionIdentifier (page 29) function to set the distribution identifier for a person's multi-value list property.

Version Notes

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABAddressBookC.h

ABGroupCopyParentGroups

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

```
CFArrayRef ABGroupCopyParentGroups (
   ABGroupRef group
);
```

Parameters

group

The group whose parent groups you wish to obtain.

Return Value

An array containing ABGroup objects representing the parents of *group*. If *group* doesn't belong to any groups, this function returns an empty array. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABAddressBookC.h

ABGroupCreate

Returns a new ABGroup object.

```
ABGroupRef ABGroupCreate (
   void
);
```

Return Value
A newly created ABGroup object. You are responsible for releasing this object.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

ABGroupCreateSearchElement

Creates an ABSearchElement object that specifies a query for ABGroup records.

```
ABSearchElementRef ABGroupCreateSearchElement (
   CFStringRef property,
   CFStringRef label,
   CFStringRef key,
   CFTypeRef value,
   ABSearchComparison comparison
);
```

Parameters

property

The name of the property to search on. It cannot be `NULL`. For a full list of the properties, see `Group Properties` (page 30) and `Common Properties` (page 83).

label

The label name for a multi-value list. If *property* does not have multiple values, pass `NULL`. If *property* does have multiple values, pass `NULL` to search all the values. By default, ABGroup records don't contain any multi-value list properties.

key

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

value

The value you are searching for. It cannot be `NULL`

comparison

Specifies the type of comparison to perform, such as `kABEqual` or `kABPrefixMatchCaseInsensitive`. For a full list, see `ABSearchComparison` (page 87).

Return Value
A search element object that specifies a query according to the above parameters. You are responsible for releasing this object.

Discussion
Use the ABAddressBook `ABCopyArrayOfMatchingRecords` (page 12) function to actually perform the query. Also, see ABSearchElement for more functions that create compound queries.

Availability

Available in Mac OS X 10.2 and later.

Declared In

`ABAddressBookC.h`

ABGroupRemoveGroup

Removes a subgroup from a group.

```
bool ABGroupRemoveGroup (
   ABGroupRef group,
   ABGroupRef groupToRemove
);
```

Parameters

group

If `NULL`, this function raises an exception.

groupToRemove

The subgroup to be removed from *group*.

Return Value

`true` if successful. If the *group* parameter is not a subgroup, this function does nothing and returns `false`.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABGroupRemoveMember

Removes a person from a group.

```
bool ABGroupRemoveMember (
   ABGroupRef group,
   ABPersonRef personToRemove
);
```

Parameters

group

The group that you wish to remove *person* from.

person

The member that you wish to remove from *group*.

Return Value

`true` if successful. If the *person* parameter is not in *group*, this function does nothing and returns `false`.

Availability

Available in Mac OS X 10.2 and later.

Declared In

`ABAddressBookC.h`

ABGroupSetDistributionIdentifier

Assigning a specific distribution identifier for a person's multi-value list property so that the group can be used as a distribution list (mailing list, in the case of an email property).

```
bool ABGroupSetDistributionIdentifier (
   ABGroupRef group,
   ABPersonRef person,
   CFStringRef property,
   CFStringRef identifier
);
```

Parameters

group

The group that *person* belongs to.

person

The person whose distribution identifier for *property* you wish to change. If NULL, this function raises an exception.

property

The multi-value list property whose distribution identifier you wish to change.

identifier

The new distribution identifier, a label used by a multi-value list such as kABAddressHomeLabel for a kABAddressProperty. Pass NULL to reset the distribution identifier to its default, a multi-value list's primary identifier.

Return Value

true if successful, false otherwise.

Discussion

The default distribution identifier is a multi-value list's primary identifier. Use this function if you need to change the distribution identifier for a particular person. For example, if the default identifier is a person's home email but you want to use John's work email, invoke this function passing kABEmailWorkLabel as the *identifier* parameter, kABEmailProperty as the *property* parameter, and John's person object as the *person* parameter.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABAddressBookC.h

Data Types

ABGroupRef

A reference to an ABGroup object.

```
typedef struct __ABGroup *ABGroupRef;
```

Availability

Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

Constants

Group Properties

Properties specific to ABGroup objects.

```
CFStringRef kABGroupNameProperty;
```

Constants
`kABGroupNameProperty`

Name of the group.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

Record Type

Used to designate record types.

```
CFStringRef kABGroupRecordType;
```

Constants
`kABGroupRecordType`

Indicates record of an ABGroup object.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobalsC.h`.

ABMultiValue Reference for C

Derived From:	CFType
Framework:	AddressBook/ABAddressBookC.h
Declared in	ABAddressBookC.h
Companion guide	Address Book Programming Guide for Mac OS X

Overview

The ABMultiValue and ABMutableMultiValue opaque types are used to represent properties that might have multiple values. Each value in a multi-value list must be the same type, and has an associated pre-defined or user-defined label, and unique identifier. The labels, however, need not be unique. For example, you can have multiple “Home” phone numbers. Each multi-value object may have a primary identifier—used to lookup 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 functions that manipulate the content of a multi-value list.

You can access values using a numeric index (similar to an array). Use the `ABMultiValueCopyIdentifierAtIndex` (page 31) function to get an identifier, the `ABMultiValueCopyLabelAtIndex` (page 32) function to get a label, and the `ABMultiValueCopyValueAtIndex` (page 33) function to get a value. However, a numeric index is temporary since a multi-value list may change. Each value or entry in a multi-value list has a unique identifier which can be used to save a reference to a specific value—the identifier is guaranteed never to change.

Use the `ABMultiValueCopyPrimaryIdentifier` (page 33) function to get the primary identifier (the identifier associated with the primary value).

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

Functions

ABMultiValueCopyIdentifierAtIndex

Returns the identifier at the given index.

```
CFStringRef ABMultiValueCopyIdentifierAtIndex (
   ABMultiValueRef multiValue,
   CFIndex index
);
```

Parameters

multiValue
: The multi-value list that you wish to access.

index
: The index of the identifier you wish to obtain. If this parameter is out of bounds, this function raises an exception.

Return Value

The identifier at *index* in *multiValue*. You are responsible for releasing this object.

Discussion

Each value in a multi-value list must be the same type, and has an associated pre-defined or user-defined label, and unique identifier. Use the ABMultiValueCopyLabelAtIndex (page 32) function to get a label, and the ABMultiValueCopyValueAtIndex (page 33)function to get a value.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABAddressBookC.h

ABMultiValueCopyLabelAtIndex

Returns the label for the given index.

```
CFStringRef ABMultiValueCopyLabelAtIndex (
   ABMultiValueRef multiValue,
   CFIndex index
);
```

Parameters

multiValue
: The multi-value list that you wish to access.

index
: The index of the identifier you wish to obtain. If this parameter is out of bounds, this function raises an exception.

Return Value

The label at *index* in *multiValue*. You are responsible for releasing this object.

Discussion

Each value in a multi-value list must be the same type, and has an associated pre-defined or user-defined label, and unique identifier. Use the ABMultiValueCopyIdentifierAtIndex (page 31) function to get a identifier, and the ABMultiValueCopyValueAtIndex (page 33)function to get a value.

Availability

Available in Mac OS X v10.2 and later.

Related Sample Code

AddressBookCarbon

Declared In

`ABAddressBookC.h`

ABMultiValueCopyPrimaryIdentifier

Returns the identifier for the primary value.

```
CFStringRef ABMultiValueCopyPrimaryIdentifier (
   ABMultiValueRef multiValue
);
```

Parameters

multiValue

The multi-value list that you wish to access.

Return Value

The unique identifier for the primary value. You are responsible for releasing this object.

Discussion

Use the `ABMultiValueCopyIdentifierAtIndex` (page 31) function to get index for the returned identifier, and the `ABMultiValueCopyValueAtIndex` (page 33)function to get its value.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABMultiValueCopyValueAtIndex

Returns the value for the given index.

```
CFTypeRef ABMultiValueCopyValueAtIndex (
   ABMultiValueRef multiValue,
   CFIndex index
);
```

Parameters

multiValue

The multi-value list that you wish to access.

index

The index of the identifier you wish to obtain. If this parameter is out of bounds, this function raises an exception.

Return Value

The value at *index* in *multiValue*. You are responsible for releasing this object.

Discussion

Each value in a multi-value list must be the same type, and has an associated pre-defined or user-defined label, and unique identifier. Use the `ABMultiValueCopyIdentifierAtIndex` (page 31) function to get a identifier, and the `ABMultiValueCopyLabelAtIndex` (page 32)function to get a label.

Availability

Available in Mac OS X v10.2 and later.

Related Sample Code
AddressBookCarbon

Declared In
`ABAddressBookC.h`

ABMultiValueCount

Returns the number of entries in a multi-value list.

```
CFIndex ABMultiValueCount (
   ABMultiValueRef multiValue
);
```

Parameters

multiValue

The multi-value list that you wish to access.

Return Value
The number of entries in *multiValue*.

Availability
Available in Mac OS X v10.2 and later.

Related Sample Code
AddressBookCarbon

Declared In
`ABAddressBookC.h`

ABMultiValueCreate

Returns a new ABMultiValue object.

```
ABMultiValueRef ABMultiValueCreate (
   void
);
```

Return Value
A new ABMultiValue object. You are responsible for releasing this object.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

ABMultiValueCreateCopy

Returns a copy of a multi-value object.

```
ABMultiValueRef ABMultiValueCreateCopy (
   ABMultiValueRef multiValue
);
```

Parameters

multiValue

The multi-value object you wish to copy. You are responsible for releasing this object.

Return Value

A copy of *multiValue*.

Availability

Available in Mac OS X 10.2 and later.

Declared In

`ABAddressBookC.h`

ABMultiValueCreateMutableCopy

Returns a mutable copy of a multi-value object.

```
ABMutableMultiValueRef ABMultiValueCreateMutableCopy (
   ABMultiValueRef multiValue
);
```

Parameters

multiValue

The multi-value object you wish to copy.

Return Value

A mutable copy of *multiValue*. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABMultiValueIndexForIdentifier

Returns the index for the given identifier.

```
CFIndex ABMultiValueIndexForIdentifier (
   ABMultiValueRef multiValue,
   CFStringRef identifier
);
```

Parameters

multiValue

The multi-value list that you wish to access.

identifier

The identifier whose index you wish to obtain.

Return Value

The index of *identifier*.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

ABMultiValuePropertyType

Returns the type for the values in a multi-value list.

```
ABPropertyType ABMultiValuePropertyType (
   ABMultiValueRef multiValue
);
```

Parameters

multiValue
The multi-value list whose property type you wish to obtain.

Return Value
The property type of *multiValue*. If the list is empty or its values are of different types, returns `kABErrorInProperty`.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

Data Types

ABMultiValueRef

A reference to an ABMultiValue or ABMutableMultiValue object.

```
typedef const struct __ABMultiValue *ABMultiValueRef;
```

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

ABMutableMultiValue Reference for C

Derived From:	ABMultiValue : CFType
Framework:	AddressBook/ABAddressBookC.h
Declared in	ABAddressBookC.h
Companion guide	Address Book Programming Guide for Mac OS X

Overview

The ABMultiValue and ABMutableMultiValue opaque types are used to represent properties that might have multiple values. Each value in a multi-value list must be the same type, and has an associated pre-defined or user-defined label, and unique identifier. The labels, however, need not be unique. For example, you can have multiple “Home” phone numbers. Each multi-value 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 functions that access the content of a multi-value list.

You can use either the `ABMultiValueAdd` (page 37) or `ABMultiValueInsert` (page 39) functions to add value/label pairs to a multi-value list. You can remove an entry in a multi-value list using the `ABMultiValueRemove` (page 39) function. You can also replace values and labels using the `ABMultiValueReplaceLabel` (page 40)and `ABMultiValueReplaceValue` (page 40) functions.

Use the `ABMultiValueSetPrimaryIdentifier` (page 41) function to set the primary identifier—that is, designate the corresponding value as the default value for a multi-value list. Use the `ABMultiValueCopyIdentifierAtIndex` (page 31) function to get the unique identifier for a value/label pair.

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

Functions

ABMultiValueAdd

Adds a value and its label to a multi-value list.

```
bool ABMultiValueAdd (
   ABMutableMultiValueRef multiValue,
   CFTypeRef value,
   CFStringRef label,
   CFStringRef *outIdentifier
);
```

Parameters

multiValue

The multi-value list you wish to modify.

value

An object representing a value in a multi-value list--it must be of the correct type. For example, if *multiValue* is the value for a property of type kABMultiStringProperty, then *value* needs to be a CFString object. See Property Types for a list of supported types in a multi-value list (see descriptions of the kABMulti... constants). If *value* is NULL, this function raises an exception.

label

The label for *value*—it need not be unique. If *label* is NULL, this function raises an exception.

outIdentifier

If *value* is added successfully, this parameter returns the new identifier.

Return Value

true if successfully, false otherwise.

Discussion

This function 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 multi-value list whose values are not all of the same type, functions, such as the ABRecord ABRecordSetValue (page 82) function, will returns NULL or kABErrorInProperty.

Availability

Available in Mac OS X v10.2 and later.

Related Sample Code

AddressBookCarbon

Declared In

ABAddressBookC.h

ABMultiValueCreateMutable

Returns a newly created mutable multi-value list object.

```
ABMutableMultiValueRef ABMultiValueCreateMutable (
   void
);
```

Return Value

A newly created ABMutableMultiValue object. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.2 and later.

Related Sample Code

AddressBookCarbon

Declared In

`ABAddressBookC.h`

ABMultiValueInsert

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

```
bool ABMultiValueInsert (
   ABMutableMultiValueRef multiValue,
   CFTypeRef value,
   CFStringRef label,
   CFIndex index,
   CFStringRef *outIdentifier
);
```

Parameters

multiValue

The multi-value list you wish to modify.

value

An object representing a value in a multi-value list--it must be of the correct type. For example, if *multiValue* is the value for a property of type kABMultiStringProperty, then *value* needs to be a CFString object. See `Property Types` for a list of supported types in a multi-value list (see descriptions of the `kABMulti...` constants). If *value* is `NULL`, this function raises an exception.

label

The label for *value*—it need not be unique. If *label* is `NULL`, this function raises an exception.

index

The index to insert *value* at. If *index* is out of bounds, this function raises an exception.

outIdentifier

If *value* is added successfully, this parameter returns the new identifier.

Return Value

`true` if successfully, `false` otherwise.

Discussion

This function 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 multi-value list whose values are not all of the same type, functions, such as the ABRecord `ABRecordSetValue` (page 82) function, will returns `NULL` or `kABErrorProperty`.

Version Notes

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABMultiValueRemove

Removes the value and label at the given index.

```
bool ABMultiValueRemove (
   ABMutableMultiValueRef multiValue,
   CFIndex index
);
```

Parameters

multiValue
: The multi-value list you wish to modify.

index
: The index of the entry to be removed. If *index* is out of bounds, this function raises an exception.

Return Value
`true` if successfully, `false` otherwise.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

ABMultiValueReplaceLabel

Replaces the label at the given index.

```
bool ABMultiValueReplaceLabel (
   ABMutableMultiValueRef multiValue,
   CFStringRef label,
   CFIndex index
);
```

Parameters

multiValue
: The multi-value list you wish to modify.

label
: The new label at *index*—it need not be unique. If *label* is `NULL`, this function raises an exception.

index
: The index of the entry to be modified. If *index* is out of bounds, this function raises an exception.

Return Value
`true` if successfully, `false` otherwise.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

ABMultiValueReplaceValue

Replaces the value at the given index.

```
bool ABMultiValueReplaceValue (
   ABMutableMultiValueRef multiValue,
   CFTypeRef value,
   CFIndex index
);
```

Parameters

multiValue

The multi-value list you wish to modify.

value

An object representing the new value in a multi-value list--it must be of the correct type. For example, if *multiValue* is the value for a property of type kABMultiStringProperty, then *value* needs to be a CFString object. See `Property Types` for a list of supported types in a multi-value list (see descriptions of the `kABMulti...` constants). If *value* is `NULL`, this function raises an exception.

index

The index of the entry to be modified. If *index* is out of bounds, this function raises an exception.

Return Value

`true` if successfully, `false` otherwise.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABMultiValueSetPrimaryIdentifier

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

```
bool ABMultiValueSetPrimaryIdentifier (
   ABMutableMultiValueRef multiValue,
   CFStringRef identifier
);
```

Parameters

multiValue

The multi-value list you wish to modify.

identifier

The identifier corresponding to the value you wish to designate as the primary value for this multi-value list. Use the `ABMultiValueCopyIdentifierAtIndex` (page 31) function to get the identifier given the index. If *identifier* is `NULL`, this function raises an exception.

Return Value

`true` if successfully, `false` otherwise.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

Data Types

ABMutableMultiValueRef

A reference to an ABMutableMultiValue object.

```
typedef struct __ABMultiValue *ABMutableMultiValueRef;
```

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

2006-05-23 | © 2006 Apple Computer, Inc. All Rights Reserved.

ABPerson C Reference

Derived From:	ABRecord : CFType
Framework:	AddressBook/ABAddressBookC.h
Declared in	ABAddressBookC.h ABGlobalsC.h
Companion guide	Address Book Programming Guide for Mac OS X

Overview

The ABPerson opaque type 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 opaque type defines properties such as the person's name, company, address, email addresses, and phone numbers.

You get a person's property value using the `ABRecordCopyValue` (page 80) function. See ABRecord for more functions that get and set properties. See the "`Constants`" (page 50) section for a list of all the properties, labels, and keys used to access fields in a person record.

Some of these properties have multiple values that are accessed via standard and user-defined labels. For example, a person may have a home, work, mobile, and fax phone numbers. Therefore, the phone attribute is defined as an ABMultiValue object containing NSString objects for each number. See ABMultiValue for more details on multi-value lists and how primary values work.

You can add your own properties to person records too using the `ABAddPropertiesAndTypes` (page 10) function—that is, attach additional program-defined data to each person record. Because the Address Book database is stored as a property list, these program-defined properties can be ignored by other applications. Note that the AddressBook database is accessed by multiple application and is not encrypted so your application should not store any sensitive information in the database like credit card numbers.

A person may also have an associated picture or image. The image in not actually stored in the Address Book database (a property list)—it's stored in a separate image file. You can set a person's image using the `ABPersonSetImageData` (page 48) function, or get an image using the `ABPersonCopyImageData` (page 45)function.

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

Use the `ABBeginLoadingImageDataForClient` (page 44) function if an image file is not local and you want to perform an asynchronous fetch. The `ABBeginLoadingImageDataForClient` (page 44) function will return an image tracking number. The tracking number and the fetched image will be passed to your callback function. Implement your callback function to handle the fetched image. Use the `ABCancelLoadingImageDataForTag` (page 45) function if for some reason you want to cancel an asynchronous fetch.

Person records may belong to multiple groups. Use the `ABPersonCopyParentGroups` (page 45) function to get the groups a person belongs to. See ABGroup for more information about groups.

You can also search for records matching a particular "query" you specify by creating an ABSearchElement object. Use the `ABPersonCreateSearchElement` (page 47) function to create an ABSearchElement object containing your query. Then use the ABAddressBook `ABCopyArrayOfMatchingRecords` (page 12) function, passing the ABSearchElement as the argument, to query the database. See ABSearchElement for more functions that create compound queries.

Your application can also import and export persons in the vCard file format using the `ABPersonCreateWithVCardRepresentation` (page 47) and `ABPersonCopyVCardRepresentation` (page 46) functions.

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

Functions

ABBeginLoadingImageDataForClient

Starts an asynchronous fetch for image data in all locations, and returns a non-zero tag for tracking.

```
CFIndex ABBeginLoadingImageDataForClient (
   ABPersonRef person,
   ABImageClientCallback callback,
   void *refcon
);
```

Parameters

`person`

The person whose image data you wish to fetch.

`callback`

The function to call when the fetch is completed.

`info`

An untyped pointer to program-defined data that will be passed to the callback.

Return Value

A non-zero tag for tracking

Discussion

Use this function to begin an asynchronous fetch. Implement your callback function to receive the fetched image. Use the `ABCancelLoadingImageDataForTag` (page 45) function to cancel an asynchronous fetch.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABCancelLoadingImageDataForTag

Cancels an asynchronous fetch of an image for the given tag.

```
void ABCancelLoadingImageDataForTag (
   CFIndex tag
);
```

Parameters

tag

Used to track an asynchronous fetch. This parameter should have been returned from a previous call to the ABBeginLoadingImageDataForClient (page 44) function.

Discussion

Use the ABBeginLoadingImageDataForClient (page 44) function to begin an asynchronous fetch. Implement your callback function to receive the fetched image. Use this function to cancel an asynchronous fetch.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABAddressBookC.h

ABPersonCopyImageData

Returns data that contains a picture of a person.

```
CFDataRef ABPersonCopyImageData (
   ABPersonRef person
);
```

Parameters

person

The person whose image you wish to obtain.

Return Value

The data representing an image of *person*. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABAddressBookC.h

ABPersonCopyParentGroups

Returns an array of groups that a person belongs to.

```
CFArrayRef ABPersonCopyParentGroups (
   ABPersonRef person
);
```

Parameters

person

The person whose parent groups you wish to obtain.

Return Value

An array of ABGroup objects which *person* belongs to. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABPersonCopyVCardRepresentation

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

```
CFDataRef ABPersonCopyVCardRepresentation (
   ABPersonRef person
);
```

Parameters

person

The person whose vCard representation you wish to obtain.

Return Value

The vCard representation of *person* as a data object in vCard format. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABPersonCreate

Returns a newly created person object.

```
ABPersonRef ABPersonCreate (
   void
);
```

Return Value

A newly created person object. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.2 and later.

Related Sample Code

AddressBookCarbon

Declared In

`ABAddressBookC.h`

ABPersonCreateSearchElement

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

```
ABSearchElementRef ABPersonCreateSearchElement (
   CFStringRef property,
   CFStringRef label,
   CFStringRef key,
   CFTypeRef value,
   ABSearchComparison comparison
);
```

Parameters

property

The name of the property to search on. It cannot be `NULL`. For a full list of the properties, see "`Person Properties`" (page 50) and `Common Properties` (page 83) in ABRecord.

label

The label name for a multi-value list. If *property* does not have multiple values, pass `NULL`. If *property* does have multiple values, pass `NULL` to search all the values.

key

The key name for a dictionary. If *property* is not a dictionary, pass `NULL`. If *property* is a dictionary, pass `NULL` to search all keys.

value

The value you are searching for. It cannot be `NULL`

comparison

Specifies the type of comparison to perform, such as `kABEqual` or `kABPrefixMatchCaseInsensitive`. For a full list, see `ABSearchComparison` (page 87).

Return Value

A search element object that specifies a query according to the above parameters. You are responsible for releasing this object.

Discussion

Use the ABAddressBook `ABCopyArrayOfMatchingRecords` (page 12) function to actually perform the query. Also, see ABSearchElement for more functions that create compound queries.

Availability

Available in Mac OS X v10.2 and later.

Related Sample Code

AddressBookCarbon

Declared In

`ABAddressBookC.h`

ABPersonCreateWithVCardRepresentation

Returns a new ABPerson object initialized with the given data in vCard format.

```
ABPersonRef ABPersonCreateWithVCardRepresentation (
   CFDataRef vCard
);
```

Parameters

vCard

The data in vCard format to initialize the new ABPerson object with.

Return Value

A new ABPerson object initialized with the given data in vCard format. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABPersonSetImageData

Sets the image for this person to the given data.

```
bool ABPersonSetImageData (
   ABPersonRef person,
   CFDataRef imageData
);
```

Parameters

person

The person whose image data you wish to set.

imageData

The image data to use as the image for *person*.

Return Value

`true` if successful, `false` otherwise.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

Callbacks

ABImageClientCallback

Prototype of a callback function used to notify an application when an asynchronous image fetch is complete.

```
typedef void (*ABImageClientCallback) (
   CFDataRef imageData,
   int tag,
   void *info
);
```

If you name your function MyCallBack, you would declare it like this:

```
const void MyCallback (
   CFDataRef imageData,
   int tag,
   void *info
);
```

Parameters

imageData

The image data in Quicktime compatible format that was loaded from an asynchronous fetch. NULL if the fetch failed.

tag

The tracking number for this fetch that should have been obtained from a previous call to the ABBeginLoadingImageDataForClient (page 44) function.

info

An untyped pointer to program-defined data that was passed to the ABBeginLoadingImageDataForClient (page 44) function.

Discussion

Use the ABBeginLoadingImageDataForClient (page 44) function to begin an asynchronous fetch, and the ABCancelLoadingImageDataForTag (page 45) function to cancel an asynchronous fetch.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABAddressBookC.h

Data Types

ABPersonRef

A reference to an ABPerson object.

```
typedef struct __ABPerson *ABPersonRef;
```

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABAddressBookC.h

Constants

Person Properties

CFPlugIn defines the following domain qualifier constants.

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

Constants

`kABFirstNameProperty`

First name (string).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABLastNameProperty`

Last name (string).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABFirstNamePhoneticProperty`

First name phonetic (string).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABLastNamePhoneticProperty`
Last name phonetic (string).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABBirthdayProperty`
Birth date (date).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABOrganizationProperty`
Company name (string)

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABJobTitleProperty`
Job Title (string).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABHomePageProperty`
Home Web page (string). *Deprecated in Mac OS X version 10.4.*

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABURLsProperty`
Web pages (multi-string).

Available in Mac OS X v10.4 and later.

Declared in `ABGlobals.h`.

`kABCalendarURIsProperty`
Calendar URIs(multi-string).

Available in Mac OS X v10.5 and later.

Declared in `ABGlobals.h`.

`kABEmailProperty`
email(s) (multi-string).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABAddressProperty`
Street Addresses (multi-dictionary).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABPhoneProperty`
Generic phone number (multi-string).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

kABAIMInstantProperty
: AIM Instant Messaging (multi-string).

  Available in Mac OS X v10.2 and later.

  Declared in ABGlobals.h.

kABJabberInstantProperty
: Jabber Instant Messaging (multi-string).

  Available in Mac OS X v10.2 and later.

  Declared in ABGlobals.h.

kABMSNInstantProperty
: MSN Instant Messaging (multi-string).

  Available in Mac OS X v10.2 and later.

  Declared in ABGlobals.h.

kABYahooInstantProperty
: Yahoo Instant Messaging (multi-string).

  Available in Mac OS X v10.2 and later.

  Declared in ABGlobals.h.

kABICQInstantProperty
: ICQ Instant Messaging (multi-string).

  Available in Mac OS X v10.2 and later.

  Declared in ABGlobals.h.

kABNoteProperty
: Note (string).

  Available in Mac OS X v10.2 and later.

  Declared in ABGlobals.h.

kABMiddleNameProperty
: Middle name (string). Not supported in the AddressBook UI.

  Available in Mac OS X v10.2 and later.

  Declared in ABGlobals.h.

kABMiddleNamePhoneticProperty
: Middle name phonetic (string). Not supported in the AddressBook UI.

  Available in Mac OS X v10.2 and later.

  Declared in ABGlobals.h.

kABTitleProperty
: Title as in “Sir”, “Duke”, “General”, “Cardinal”, or “Lord” (string). Not supported in the AddressBook UI.

  Available in Mac OS X v10.2 and later.

  Declared in ABGlobals.h.

kABSuffixProperty
: Suffix as in “Sr.”, “Jr.”, “III”, or “Esq.” (string). Not supported in the AddressBook UI.

  Available in Mac OS X v10.2 and later.

  Declared in ABGlobals.h.

`kABNicknameProperty`
: Nickname (string). Not supported in the AddressBook UI.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABMaidenNameProperty`
: Maiden name (string). Not supported in the AddressBook UI.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABOtherDatesProperty`
: Dates associated with a person (ABMultiDateProperty containing dates).

Available in Mac OS X v10.3 and later.

Declared in `ABGlobals.h`.

`kABRelatedNamesProperty`
: Names of people related to a person (ABMultiStringProperty containing names).

Available in Mac OS X v10.3 and later.

Declared in `ABGlobals.h`.

`kABDepartmentProperty`
: Department name (string).

Available in Mac OS X v10.3 and later.

Declared in `ABGlobals.h`.

`kABPersonFlags`
: Property that specifies the name ordering and user configuration of a record in the Address Book application.

Available in Mac OS X v10.3 and later.

Declared in `ABGlobals.h`.

Person Flags

The ABPersonFlags property is used to access the following settings:

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

Constants

`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`.

`kABShowAsMask`

Used in conjunction with `kABShowAsPerson` and `kABShowAsCompany` to determine record configuration.

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) in 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`.

`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`.

Availability

Available in Mac OS X v10.3 and later.

Email Labels

Labels used by the email property.

```
CFStringRef kABEmailWorkLabel;
CFStringRef kABEmailHomeLabel;
```

Constants

`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`.

Address Labels

Labels used by the address property.

```
CFStringRef kABAddressWorkLabel;
CFStringRef kABAddressHomeLabel;
```

Constants

`kABAddressWorkLabel`

Work address.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABAddressHomeLabel`

Home address.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

Address Keys

Keys used by the address property.

```
CFStringRef kABAddressStreetKey;
CFStringRef kABAddressCityKey;
CFStringRef kABAddressStateKey;
CFStringRef kABAddressZIPKey;
CFStringRef kABAddressCountryKey;
CFStringRef kABAddressCountryCodeKey;
```

Constants

`kABAddressStreetKey`

Street (string).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABAddressCityKey`

City (string).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABAddressStateKey`

State (string).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABAddressZIPKey`

Zip (string).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABAddressCountryKey`

Country (string).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABAddressCountryCodeKey`

Country Code (string).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

Phone Labels

Labels used by the phone property.

```
CFStringRef kABPhoneWorkLabel;
CFStringRef kABPhoneHomeLabel;
CFStringRef kABPhoneMobileLabel;
CFStringRef kABPhoneMainLabel;
CFStringRef kABPhoneHomeFAXLabel;
CFStringRef kABPhoneWorkFAXLabel;
CFStringRef kABPhonePagerLabel;
```

Constants

`kABPhoneWorkLabel`

Work phone.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABPhoneHomeLabel`

Home phone.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABPhoneMobileLabel`

Cell phone.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABPhoneMainLabel`

Main phone.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABPhoneHomeFAXLabel`

FAX number.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABPhoneWorkFAXLabel`

FAX number.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABPhonePagerLabel`

Pager number.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

Web Page Labels

Labels used by the `kABURLsProperty` property.

```
CFStringRef kABHomePageLabel;
```

Constants

`kABHomePageLabel`

Web page URL.

Available in Mac OS X v10.4 and later.

Declared in `ABGlobals.h`.

Related Names Labels

Labels used by the related-names property.

```
CFStringRef kABMotherLabel;
CFStringRef kABFatherLabel;
CFStringRef kABParentLabel;
CFStringRef kABSisterLabel;
CFStringRef kABBrotherFAXLabel;
CFStringRef kABChildLabel;
CFStringRef kABFriendLabel;
CFStringRef kABSpouseLabel;
CFStringRef kABPartnerLabel;
CFStringRef kABAssistantLabel;
CFStringRef kABManagerLabel;
```

Constants

`kABMotherLabel`

Mother.

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`.

`kABParentLabel`
: Parent.

  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`.

`kABBrotherLabel`
: Brother.

  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`.

Availability
Available in Mac OS X v10.3 and later.

AIM Instant Labels

Labels used by the AIM instance property.

```
CFStringRef kABAIMWorkLabel;
CFStringRef kABAIMHomeLabel;
```

Constants

`kABAIMWorkLabel`

Work AIM.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABAIMHomeLabel`

Home AIM.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

Jabber Instant Labels

Labels used by the Jabber instance property.

```
CFStringRef kABJabberWorkLabel;
CFStringRef kABJabberHomeLabel;
```

Constants

`kABJabberWorkLabel`

Work Jabber.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABJabberHomeLabel`

Home Jabber.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

MSN Instant Labels

Labels used by the MSN instance property.

```
CFStringRef kABMSNWorkLabel;
CFStringRef kABMSNHomeLabel;
```

Constants

`kABMSNWorkLabel`

Work MSN.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABMSNHomeLabel`

Home MSN.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

Yahoo Instant Labels

Labels used by the Yahoo instance property.

```
CFStringRef kABYahooWorkLabel;
CFStringRef kABYahooHomeLabel;
```

Constants

`kABYahooWorkLabel`

Work Yahoo.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABYahooHomeLabel`

Home Yahoo.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

ICQ Instant Labels

Labels used by the ICQ instance property.

```
CFStringRef kABICQWorkLabel;
CFStringRef kABICQHomeLabel;
```

Constants

`kABICQWorkLabel`

Work ICQ.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABICQHomeLabel`

Home ICQ.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

Other Dates Labels

Labels for values contained in ABOtherDatesProperty multi-value properties.

```
kABAnniversaryLabel
```

Constants

`kABAnniversaryLabel`

Anniversary date.

Available in Mac OS X v10.3 and later.

Declared in `ABGlobals.h`.

Availability

Available in Mac OS X v10.3 and later.

Generic Labels

Generic labels that may apply to any multi-value list property.

```
CFStringRef kABWorkLabel;
CFStringRef kABHomeLabel;
CFStringRef kABOtherLabel;
```

Constants

`kABWorkLabel`

All `kABXXXXWorkLabel` constants are equivalent to this label.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABHomeLabel`

All `kABXXXXHomeLabel` constants are equivalent to this label.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABOtherLabel`

Can be used with any multi-value property.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

Record Type

Constants used to indicate a specific type of record.

```
CFStringRef kABPersonRecordType;
```

Constants

`kABPersonRecordType`

Indicates record of an ABPerson object.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobalsC.h`.

ABPicker Reference for C

Derived From:	CFType
Framework:	AddressBook/ABAddressBookC.h
Declared in	ABPeoplePickerC.h ABGlobalsC.h
Companion guide	Address Book Programming Guide for Mac OS X

Overview

The ABPicker opaque type allows you create and manipulate people-picker windows in an application's user interface.

Use ABPickerCreate (page 67) to create a people-picker window and ABPickerSetVisibility (page 75) to make it visible. Use ABPickerAddProperty (page 63) to add properties to the record list. Users can specify which property to display by clicking the property column in the record list and choosing the desired property.

Functions

ABPickerAddProperty

Adds a property to the group of properties available in the record list. Use ABPickerRemoveProperty (page 71) to remove a property from the list and ABPickerCopyProperties (page 65) to obtain the list of properties available in the list.

```
void ABPickerAddProperty (
   ABPickerRef inPicker,
   CFStringRef inProperty
);
```

Parameters

inPicker

The people-picker window to manipulate.

inProperty

The property to add, specified using one of the kAB...Property constants (defined in ABPerson and ABRecord) or a custom string.

Availability

Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerChangeAttributes

Specifies the selection behaviors for a people-picker window. Use `ABPickerGetAttributes` (page 70) to obtain the selection behaviors specified for the window.

```
void ABPickerChangeAttributes (
   ABPickerRef inPicker,
   ABPickerAttributes inAttributesToSet,
   ABPickerAttributes inAttributesToClear
);
```

Parameters

`inPicker`
The people-picker window to manipulate.

`inAttributesToSet`
The attributes to set for the window. The possible selection behaviors are described in `ABPickerAttributes` (page 76).

`inAttributesToClear`
The attributes to unset for the window. The possible selection behaviors are described in `ABPickerAttributes` (page 76).

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerClearSearchField

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

```
void ABPickerClearSearchField (
   ABPickerRef inPicker
);
```

Parameters

`inPicker`
The people-picker window to manipulate.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerCopyColumnTitle

Obtains the title of a custom property.

```
CFStringRef ABPickerCopyColumnTitle (
   ABPickerRef inPicker,
   CFStringRef inProperty
);
```

Parameters

inPicker

The people-picker window in question.

Return Value

The title of the column that displays the custom property in the record list.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ABPeoplePickerC.h`

ABPickerCopyDisplayedProperty

Returns the name of the property currently displayed in the record list.

```
CFStringRef ABPickerCopyDisplayedProperty (
   ABPickerRef inPicker
);
```

Parameters

inPicker

The people-picker window in question.

Return Value

The name of the property displayed.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ABPeoplePickerC.h`

ABPickerCopyProperties

Obtains the list of properties available in the record list. Use `ABPickerAddProperty` (page 63) to add a property to the record list and `ABPickerRemoveProperty` (page 71) to remove a property from the list.

```
CFArrayRef ABPickerCopyProperties (
   ABPickerRef inPicker
);
```

Parameters

inPicker

The people-picker window in question.

Return Value

An array with the list of properties available in the record list.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerCopySelectedGroups

Returns the groups selected in the group list as an array of ABGroup objects.

```
CFArrayRef ABPickerCopySelectedGroups (
   ABPickerRef inPicker
);
```

Parameters

inPicker
The people-picker window in question.

Return Value
An array with the groups selected in the group list.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerCopySelectedIdentifiers

Returns the identifiers of the selected values in a multi-value property or an empty array if the property displayed is a single-value property.

```
CFArrayRef ABPickerCopySelectedIdentifiers (
   ABPickerRef inPicker,
   ABPersonRef inPerson
);
```

Parameters

inPicker
The people-picker window in question.

inPerson
The ABPerson that contains the multi-value property in question.

Return Value
An array of CFString objects representing the selected identifiers.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerCopySelectedRecords

Returns the selection in the record list as an array of ABGroup or ABPerson objects.

```
CFArrayRef ABPickerCopySelectedRecords (
   ABPickerRef inPicker
);
```

Parameters

inPicker

The people-picker window in question.

Return Value

An array with the groups or records selected in the record list.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ABPeoplePickerC.h`

ABPickerCopySelectedValues

Returns the selected values in a multi-value property or an empty array if no values are selected or the property displayed is a single-value property.

```
CFArrayRef ABPickerCopySelectedValues (
   ABPickerRef inPicker
);
```

Parameters

inPicker

The people-picker window in question.

Return Value

An array of the values selected.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ABPeoplePickerC.h`

ABPickerCreate

Creates an ABPickerRef. The corresponding window is hidden. Invoke `ABPickerSetVisibility` (page 75) to show it. Release with `CFRelease`.

```
ABPickerRef ABPickerCreate (
   void
);
```

Return Value

The object that represents the people-picker window.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerDeselectAll

Deselects all selected groups, records, and values in multi-value properties.

```
void ABPickerDeselectAll (
   ABPickerRef inPicker
);
```

Parameters

inPicker

The people-picker window to manipulate.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerDeselectGroup

Deselects a group in the group list.

```
void ABPickerDeselectGroup (
   ABPickerRef inPicker,
   ABGroupRef inGroup
);
```

Parameters

inPicker

The people-picker window to manipulate.

inGroup

The group to deselect in the list.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerDeselectIdentifier

Deselects a value in multi-value property currently displayed in the record list.

```
void ABPickerDeselectIdentifier (
   ABPickerRef inPicker,
   ABPersonRef inPerson,
   CFStringRef inIdentifier
);
```

Parameters

inPicker

The people-picker window to manipulate.

inPerson

The ABPerson that contains the multi-value property in question.

inIdentifier

The identifier of the value to deselect in the multi-value property.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ABPeoplePickerC.h`

ABPickerDeselectRecord

Deselects a group in the record list.

```
void ABPickerDeselectRecord (
   ABPickerRef inPicker,
   ABRecordRef inRecord
);
```

Parameters

inPicker

The people-picker window to manipulate.

inRecord

The record to deselect in the list.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ABPeoplePickerC.h`

ABPickerEditInAddressBook

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

```
void ABPickerEditInAddressBook (
   ABPickerRef inPicker
);
```

Parameters

inPicker

The people-picker window in question.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ABPeoplePickerC.h`

ABPickerGetAttributes

Indicates the selection behaviors selected a people-picker window. Use `ABPickerChangeAttributes` (page 64) to specify selection behaviors for the window.

```
ABPickerAttributes ABPickerGetAttributes (
   ABPickerRef inPicker
);
```

Parameters

inPicker

The people-picker window in question.

Return Value

An OptionBits object with the selection behaviors selected for the window. The possible selection behaviors are described in `ABPickerAttributes` (page 76).

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ABPeoplePickerC.h`

ABPickerGetDelegate

Obtains the delegate for a people-picker window.

```
EventTargetRef ABPickerGetDelegate (
   ABPickerRef inPicker
);
```

Parameters

inPicker

The people-picker window in question.

Return Value

The window’s delegate.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ABPeoplePickerC.h`

ABPickerGetFrame

Returns the position and size of the people-picker window.

```
void ABPickerGetFrame (
   ABPickerRef inPicker,
   HIRect *outFrame
);
```

Parameters

inPicker

The people-picker window in question.

outFrame

On output, the position and size of the window in screen coordinates.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABPeoplePickerC.h

ABPickerIsVisible

Indicates whether the people-picker window is visible.

```
bool ABPickerIsVisible (
   ABPickerRef inPicker
);
```

Parameters

inPicker

The people-picker window in question.

Return Value

true when the window is visible, false otherwise.

Availability

Available in Mac OS X v10.3 and later.

Declared In

ABPeoplePickerC.h

ABPickerRemoveProperty

Removes a property from the group of properties whose values are shown in the record list. Use ABPickerAddProperty (page 63) to add a property to the record list and ABPickerCopyProperties (page 65) to obtain the list of properties shown in the record list.

```
void ABPickerRemoveProperty (
   ABPickerRef inPicker,
   CFStringRef inProperty
);
```

Parameters

inPicker

The people-picker window to manipulate.

2006-05-23 | © 2006 Apple Computer, Inc. All Rights Reserved.

inProperty
The property to remove, specified using one of the `kAB...Property` constants or a custom string.

Return Value

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerSelectGroup

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

```
void ABPickerSelectGroup (
   ABPickerRef inPicker,
   ABGroupRef inGroup,
   bool inExtendSelection
);
```

Parameters

inPicker
The people-picker window to manipulate.

inGroup
The group to select.

inExtendSelection
`true` if you want to add *inGroup* to the list of selected groups in the group list; `false` if you want *inGroup* to be the only group selected in the list.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerSelectIdentifier

Selects a value or a set of values in a multi-value property.

```
void ABPickerSelectIdentifier (
   ABPickerRef inPicker,
   ABPersonRef inPerson,
   CFStringRef inIdentifier,
   bool inExtendSelection
);
```

Parameters

inPicker
The people-picker window to manipulate.

inPerson
The person with the multi-value property with the value to select.

inIdentifier

The identifier of the value to select in the multi-value property.

inExtendSelection

`true` if you want to add the value to the list of selected values in the multi-value property of the desired person; `false` if you want the value to be the only value selected in the list.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerSelectInAddressBook

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

```
void ABPickerSelectInAddressBook (
   ABPickerRef inPicker
);
```

Parameters

inPicker

The people-picker window in question.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerSelectRecord

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

```
void ABPickerSelectRecord (
   ABPickerRef inPicker,
   ABRecordRef inRecord,
   bool inExtendSelection
);
```

Parameters

inPicker

The people-picker window to manipulate.

inRecord

The record to select.

inExtendSelection

`true` if you want to add *inRecord* to the list of selected records in the record list; `false` if you want *inRecord* to be the only record selected in the list.

Availability
Available in Mac OS X v10.3 and later.

Declared In

`ABPeoplePickerC.h`

ABPickerSetColumnTitle

Sets the title for a custom property.

```
void ABPickerSetColumnTitle (
   ABPickerRef inPicker,
   CFStringRef inTitle,
   CFStringRef inProperty
);
```

Parameters

inPicker

The people-picker window to manipulate.

inTitle

The new column title.

inProperty

The property whose column's title to set.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ABPeoplePickerC.h`

ABPickerSetDelegate

Sets the event handler for people-picker events.

```
void ABPickerSetDelegate (
   ABPickerRef inPicker,
   EventTargetRef inDelegate
);
```

Parameters

inPicker

The people-picker window in question.

inDelegate

The delegate for the window.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ABPeoplePickerC.h`

ABPickerSetDisplayedProperty

Displays one of the properties whose values are shown in the record list.

```
void ABPickerSetDisplayedProperty (
   ABPickerRef inPicker,
   CFStringRef inProperty
);
```

Parameters

inPicker
The people-picker window to manipulate.

inProperty
The property to display.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerSetFrame

Specifies the position and size of the people-picker window.

```
void ABPickerSetFrame (
   ABPickerRef inPicker,
   const HIRect *inFrame
);
```

Parameters

inPicker
The people-picker window to manipulate.

inFrame
The size and position of the window in screen coordinates.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerSetVisibility

Shows or hides a people-picker window.

```
void ABPickerSetVisibility (
   ABPickerRef inPicker,
   bool visible
);
```

Parameters

inPicker
The people-picker window to manipulate.

visible
`true` to show the window and `false` to hide it.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

Data Types

ABAddressBookRef

A reference to an ABPicker object.

```
typedef struct OpaqueABPicker *ABPickerRef;
```

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

ABPickerAttributes

These constants specify the selection behavior for the values of multi-value properties.

```
typedef OptionBits ABPickerAttributes;
```

Discussion
When multiple behaviors are selected, the most restrictive behavior is used. The default behavior is single-value selection (`kABPickerSingleValueSelection`).

Constant	Description
`kABPickerSingleValueSelection` Available in Mac OS X v10.3 and later. Declared in `ABPeoplePickerC.h`.	Allows the user to select a single value.
`kABPickerMultipleValueSelection` Available in Mac OS X v10.3 and later. Declared in `ABPeoplePickerC.h`.	Allows the user to select multiple values.
`kABPickerAllowGroupSelection`	Allows the user to select entire groups in the groups list. Otherwise, at least one record is selected when the user selectes a group.
`kABPickerAllowMultipleSelection`	Allows the user to select more than one group or record at a time.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABPeoplePickerC.h`

Constants

People-Picker Event Class

This is the People Picker event class.

```
enum {
    kEventClassABPeoplePicker = 'abpp'
};
```

Constants

`kEventClassABPeoplePicker`

The class of people-picker events.

Available in Mac OS X v10.3 and later.

Declared in `ABPeoplePickerC.h`.

Availability
Available in Mac OS X v10.3 and later.

People-Picker Event Kinds

Constants used to specify People Picker event types.

```
enum {
    kEventABPeoplePickerGroupSelectionChanged = 1,
    kEventABPeoplePickerNameSelectionChanged = 2,
    kEventABPeoplePickerValueSelectionChanged = 3,
    kEventABPeoplePickerDisplayedPropertyChanged = 4,
    kEventABPeoplePickerGroupDoubleClicked = 5,
    kEventABPeoplePickerNameDoubleClicked = 6,
};
```

Constants

`kEventABPeoplePickerGroupSelectionChanged`

The selection in the group list changed.

Available in Mac OS X v10.3 and later.

Declared in `ABPeoplePickerC.h`.

`kEventABPeoplePickerNameSelectionChanged`

The selection in the name list changed.

Available in Mac OS X v10.3 and later.

Declared in `ABPeoplePickerC.h`.

`kEventABPeoplePickerValueSelectionChanged`
    The selection in a multi-value property changed.

    Available in Mac OS X v10.3 and later.

    Declared in `ABPeoplePickerC.h`.

`kEventABPeoplePickerDisplayedPropertyChanged`
    The displayed property in the record list changed.

    Available in Mac OS X v10.3 and later.

    Declared in `ABPeoplePickerC.h`.

`kEventABPeoplePickerGroupDoubleClicked`
    A group in the group list was double-clicked.

    Available in Mac OS X v10.3 and later.

    Declared in `ABPeoplePickerC.h`.

`kEventABPeoplePickerNameDoubleClicked`
    A record in the record list was double-clicked.

    Available in Mac OS X v10.3 and later.

    Declared in `ABPeoplePickerC.h`.

Discussion

A people-picker window delegate is notified when the events defined earlier occur. People Picker events contain an event parameter, which contains the ABPicker object. Use the following code to obtain the ABPicker object:

```
GetEventParameter ( inEvent, kEventParamABPickerRef,
                    typeCFTypeRef, NULL, sizeof(ABPickerRef),
                    NULL, &outPickerRef );
```

Availability

Available in Mac OS X v10.3 and later.

People-Picker Event Parameter Name

Use this constant to obtain the ABPicker object from a People Picker event.

```
enum {
    kEventParamABPickerRef = 'abpp'
};
```

Constants

`kEventParamABPickerRef`
    The parameter name of people-picker events.

    Available in Mac OS X v10.3 and later.

    Declared in `ABPeoplePickerC.h`.

Availability

Available in Mac OS X v10.3 and later.

ABRecord C Reference

Derived From:	CFType
Framework:	AddressBook/ABAddressBookC.h
Declared in	ABAddressBookC.h ABGlobalsC.h
Companion guide	Address Book Programming Guide for Mac OS X

Overview

ABRecord is an abstract opaque type providing a common interface to and defining common properties for all records in the Address Book database. 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 functions for getting, setting, and removing property values.

Use `ABRecordCopyValue` (page 80) to get a record's property value, use `ABRecordSetValue` (page 82) to set a value, and `ABRecordRemoveValue` (page 81) to remove a value.

Each record in the Address Book database has a corresponding unique ID obtained using the `ABRecordCopyUniqueId` (page 80) function. The unique ID is used by other functions in the AddressBook framework.

You can check if a record is read-only by using the `ABRecordIsReadOnly` (page 81) function.

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

Functions

ABRecordCopyRecordType

Returns the type of the given record.

```
CFStringRef ABRecordCopyRecordType (
   ABRecordRef record
);
```

Parameters

record

The record whose type you wish to obtain.

Return Value
The type of *record*, one of the kAB...RecordType constants. You are responsible for releasing this object.

Availability
Available in Mac OS X v10.2 and later.

Declared In
ABAddressBookC.h

ABRecordCopyUniqueId

Returns the unique ID of the receiver.

```
CFStringRef ABRecordCopyUniqueId (
   ABRecordRef record
);
```

Parameters

record
The record whose unique ID you wish to obtain.

Return Value
The unique ID corresponding to *record*. You are responsible for releasing this object.

Availability
Available in Mac OS X 10.2 and later.

Declared In
ABAddressBookC.h

ABRecordCopyValue

Returns the value of the given property.

```
CFTypeRef ABRecordCopyValue (
   ABRecordRef record,
   CFStringRef property
);
```

Parameters

record
The record whose value you wish to obtain.

property
The property name in *record* whose value you wish to obtain. May be a pre-defined or program-defined property. See Common Properties (page 83) for a list of properties all records have, and specific ABRecord derived opaque types for any additional properties.

Return Value
The value for *property* in *record*. The type of the returned value depends on the property type (see Property Types for a list of possible property types). You are responsible for releasing this object.

Availability
Available in Mac OS X v10.2 and later.

Related Sample Code
AddressBookCarbon

MyFirstJNIProject

Declared In
`ABAddressBookC.h`

ABRecordCreateCopy

Returns a copy of the given record.

```
ABRecordRef ABRecordCreateCopy (
   ABRecordRef record
);
```

Parameters

`record`

The record you wish to copy.

Return Value
A copy of the specified ABRecordRef.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`ABAddressBookC.h`

ABRecordIsReadOnly

Returns whether or not the record is read-only.

```
bool ABRecordIsReadOnly (
   ABRecordRef record
);
```

Parameters

`record`

The record you wish to check.

Return Value
`true` if record is read-only, `false` otherwise.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`ABAddressBookC.h`

ABRecordRemoveValue

Removes the value of the given property.

```
bool ABRecordRemoveValue (
   ABRecordRef record,
   CFStringRef property
);
```

Parameters

record

The record whose value you wish to remove.

property

The property name in *record* whose value you wish to remove. May be a pre-defined or program-defined property. See Common Properties (page 83) for a list of properties all records have, and specific ABRecord derived opaque types for any additional properties.

Return Value

The value for *property* in *record*. The type of the returned value depends on the property type (see Property Types for a list of possible property types). You are responsible for releasing this object.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABAddressBookC.h

ABRecordSetValue

Sets the value of a given property for a record.

```
bool ABRecordSetValue (
   ABRecordRef record,
   CFStringRef property,
   CFTypeRef value
);
```

Parameters

record

The record you wish to modify.

property

The property whose value you wish to set. May be a pre-defined or program-defined property. See Common Properties (page 83) for a list of properties all records have, and specific ABRecord derived opaque types for any additional properties. If NULL, this function raises an exception.

value

The new value for *property* in *record*. If NULL or not the correct type, this function raises an exception.

Return Value

If *property* is a multi-value list property, this method checks to see if the values in the multi-value list are the same type. If the multi-value list contains mixed types, this method returns false. Returns true if successful, false otherwise.

Availability

Available in Mac OS X v10.2 and later.

Related Sample Code

AddressBookCarbon

Declared In
`ABAddressBookC.h`

Data Types

ABRecordRef

A reference to an ABRecord object or any of its derived opaque types.

```
typedef void *ABRecordRef;
```

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

Constants

Common Properties

Properties common to all record types.

```
CFStringRef kABUIDProperty;
CFStringRef kABCreationDateProperty;
CFStringRef kABModificationDateProperty
```

Constants

`kABUIDProperty`

The UID property.

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABCreationDateProperty`

Creation date (when first saved).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

`kABModificationDateProperty`

Modification date (when last saved).

Available in Mac OS X v10.2 and later.

Declared in `ABGlobals.h`.

ABSearchElement C Reference

Derived From:	CFType
Framework:	AddressBook/ABAddressBookC.h
Declared in	ABAddressBookC.h ABGlobalsC.h
Companion guide	Address Book Programming Guide for Mac OS X

Overview

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

You can create a simple query by creating an ABSearchElement object using either the ABGroupCreateSearchElement (page 27) or ABPersonCreateSearchElement (page 47) function for the corresponding record. Then you use the ABAddressBook ABCopyArrayOfMatchingValues (page 12) function, passing the ABSearchElement as the parameter, to query the database.

ABSearchElement also provides a function for creating compound queries. Use the ABSearchElementCreateWithConjunction (page 85) function to combine two simple or complex queries into a compound query using either the kABSearchAnd or kABSearchOr conjunction constants.

Use the ABSearchElementMatchesRecord (page 86) function to test whether or not a specific record matches a query.

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

Functions

ABSearchElementCreateWithConjunction

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

```
ABSearchElementRef ABSearchElementCreateWithConjunction (
   ABSearchConjunction conjunction,
   CFArrayRef childrenSearchElement
);
```

Parameters

conjunction

The conjunction used to join the search elements in *children*. Can be either `kABSearchAnd` or `kABSearchOr`.

children

An array containing ABSearchElement objects to be joined using *conjunction*. If `NULL` this function raises an exception.

Return Value

A new compound search element joining the search elements in *children* using *conjunction*. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABSearchElementMatchesRecord

Tests whether or not a record matches a search element.

```
bool ABSearchElementMatchesRecord (
   ABSearchElementRef searchElement,
   ABRecordRef record
);
```

Parameters

searchElement

The search element containing the query you wish to test *record* with.

record

The record you wish to test.

Return Value

Returns `true` if the *record* parameter satisfies the conditions in the *searchElement*, `false` otherwise.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

Data Types

ABSearchElementRef

A reference to an ABSearchElement object.

```
typedef struct __ABSearchElementRef *ABSearchElementRef;
```

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

Constants

ABSearchComparison

Constants used to specify the type of comparison being made.

```
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
};
```

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`.

`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`.

`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`.

`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`.

`kABBitsInBitFieldMatch`

Search for elements that match the bits in ABPersonFlags.

Available in Mac OS X v10.3 and later.

Declared in `ABTypedefs.h`.

2006-05-23 | © 2006 Apple Computer, Inc. All Rights Reserved.

`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`.

`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 in a call to the ABGroupCreateSearchElement (page 27) or ABPersonCreateSearchElement (page 47) function to specify the type of comparison being made.

Version Notes

ABSearchConjunction

Constants used to create compound search elements.

```
enum ABSearchConjunction {
    kABSearchAnd = 0,
    kABSearchOr = 1
};
```

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 to create compound search elements with the ABSearchElementCreateWithConjunction (page 85) function.

Version Notes

PART II

Managers

Address Book Utilities Reference

Framework: AddressBook/ABAddressBookC.h

Overview

Note: This document was previously titled *ABUtilities Reference for C*.

Functions

ABCopyLocalizedPropertyOrLabel

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

```
CFStringRef ABCopyLocalizedPropertyOrLabel (
   CFStringRef labelOrProperty
);
```

Parameters

propertyOrLabel

The property, label, or key to be localized.

Return Value

The localized version of *propertyOrLabel*, or *propertyOrLabel* if a localized string can not be found. You are responsible for releasing this object.

Availability

Available in Mac OS X v10.2 and later.

Declared In

ABAddressBookC.h

PART III

Other References

Address Book Actions Reference

Framework:	AddressBook/ABActionsC.h
Declared in	ABActionsC.h
Companion guide	Address Book Programming Guide for Mac OS X

Overview

Note: This document was previously titled *ABActions Reference for C*.

The Address Book action callbacks allow you to populate the rollover menus of the Address Book application with custom items. You do this by writing an Address Book action plug-in that implements a function named `ABActionRegisterCallbacks`. This function registers a set of callback functions that are invoked by Address Book. The plug-in's CFBundle must also implement the callback functions.

This is an example implementation of the ABActionRegisterCallbacks function:

```
ABActionCallbacks* ABActionRegisterCallbacks(void)
{
    ABActionCallbacks *callbacks;
    callbacks = malloc(sizeof(ABActionCallbacks));
    if (callbacks == NULL)
        return NULL;
    callbacks->version = 0;
    callbacks->property = actionProperty;
    callbacks->title = actionTitle;
    callbacks->enabled = actionEnabled;
    callbacks->selected = actionSelected;
    return callbacks;
}
```

Each action plug-in can implement only one action. Actions can only apply to items with labels.

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.

Callbacks

ABActionCopyTitleCallback

Return the title of the menu item for the action. If the property returned by `ABActionGetPropertyCallback` (page 100) is a multi-value property, *`identifier`* contains the unique identifier of the value selected.

```
typedef CFStringRef (*ABActionCopyTitleCallback) (
   ABPersonRef person,
   CFString CFStringRef identifier
);
```

You may implement the function like this:

```
CFStringRef actionTitle(ABPersonRef person,
CFStringRef identifier)
```

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABActionsC.h`

ABActionEnabledCallback

Return `true` if the action menu item should be enabled, `false` otherwise. If the property returned by `ABActionGetPropertyCallback` (page 100) is a multi-value property, *`identifier`* contains the unique identifier of the value selected.

```
typedef Boolean (*ABActionEnabledCallback) (
   ABPersonRef person,
   CFStringRef identifier
);
```

You may implement the function like this:

```
Boolean actionEnabled(ABPersonRef person,
CFStringRef identifier)
```

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABActionsC.h`

ABActionGetPropertyCallback

Return the property the action applies to.

```
typedef CFStringRef (*ABActionGetPropertyCallback)
(void);
```

You may implement the function like this:

```
CFStringRef actionProperty(void)
```

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABActionsC.h`

ABActionSelectedCallback

Execute the action. If the property returned by `ABActionGetPropertyCallback` (page 100) is a multi-value property, *`identifier`* contains the unique identifier of the value selected; otherwise, identifier is `NULL`.

```
typedef void (*ABActionSelectedCallback) (
   ABPersonRef person,
   CFStringRef identifier
);
```

You may implement the function like this:

```
void actionSelected(ABPersonRef person,
CFStringRef identifier)
```

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ABActionsC.h`

Data Types

ABActionCallbacks

This is the structure that the `ABActionRegisterCallbacks` returns to tell Address Book about the action the plug-in implements.

```
typedef struct {
    CFIndex version;
    ABActionGetpropertyCallback property;
    ABActionCopyTitleCallback title;
    ABActionEnabledCallback enabled;
    ABActionSelectedCallback selected;
} ABActionCallbacks;
```

Availability
Available in Mac OS X v10.3 and later.

Declared In

`ABActionsC.h`

Document Revision History

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

Date	Notes
2006-05-23	First publication of this content as a collection of previously published documents.

Index