
Address Book Framework Reference

Data Management: Contact Data



2010-03-22



Apple Inc.
© 2010 Apple Inc.
All rights reserved.

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

The Apple logo is a trademark of Apple Inc.

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

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

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

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

Apple, the Apple logo, and iPhone are trademarks of Apple Inc., registered in the United States and other countries.

MobileMe is a trademark of Apple Inc.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

Simultaneously published in the United States and Canada.

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

ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

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

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

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

Contents

Introduction **Introduction** 5

Part I **Opaque Types** 7

Chapter 1 **ABAddressBook Reference** 9

Overview 9
Functions by Task 9
Functions 10
Callbacks 14
Data Types 15
Constants 15

Chapter 2 **ABMultiValue Reference** 17

Overview 17
Functions by Task 17
Functions 18
Data Types 21
Constants 22

Chapter 3 **ABMutableMultiValue Reference** 23

Overview 23
Functions by Task 23
Functions 24
Data Types 28

Chapter 4 **ABRecord Reference** 29

Overview 29
Functions by Task 29
Functions 30
Data Types 33
Constants 34

Part II **Other References** 39

Chapter 5 **ABGroup Reference** 41

Overview 41

Functions by Task 41
Functions 42
Constants 47

Chapter 6 **ABPerson Reference 49**

Overview 49
Functions by Task 49
Functions 51
Data Types 59
Constants 60

Chapter 7 **ABSource Reference 71**

Overview 71
Functions 71
Data Types 72
Constants 73

Document Revision History 75

Introduction

Framework	/System/Library/Frameworks/AddressBook.framework
Header file directories	/System/Library/Frameworks/AddressBook.framework/Headers
Declared in	ABAddressBook.h ABGroup.h ABMultiValue.h ABPerson.h ABRecord.h ABSource.h

The Address Book framework provides access to a centralized contacts database, called the Address Book database, that stores a user's contacts. Applications such as Mail and Text use this database to present information about known and unknown persons.

Opaque Types

ABAddressBook Reference

Derived From:	CTypeRef
Framework:	AddressBook/AddressBook.h
Declared in	ABAddressBook.h

Overview

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

The `ABAddressBook` opaque type provides functions for creating references to the Address Book database, saving changes, discarding changes, and registering for changes made externally (by other threads or processes) to the database.

Functions by Task

Managing Address Books

[ABAddressBookCreate](#) (page 11)

Creates a new address book with data from the Address Book database.

[ABAddressBookHasUnsavedChanges](#) (page 11)

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

[ABAddressBookSave](#) (page 13)

Saves any unsaved changes to the Address Book database.

[ABAddressBookRevert](#) (page 13)

Discards unsaved changes in an address book.

Managing Address Book Records

[ABAddressBookAddRecord](#) (page 10)

Adds a record to an address book.

[ABAddressBookRemoveRecord](#) (page 12)

Removes a record from an address book.

Getting Address Book Change Notifications

[ABAddressBookRegisterExternalChangeCallback](#) (page 12)

Registers a callback to receive notifications when the Address Book database is modified.

[ABAddressBookUnregisterExternalChangeCallback](#) (page 14)

Unregisters a callback.

Localizing Text

[ABAddressBookCopyLocalizedLabel](#) (page 10)

Returns a localized version of a record-property label.

Functions

ABAddressBookAddRecord

Adds a record to an address book.

```
bool ABAddressBookAddRecord (
    ABAddressBookRef addressBook,
    ABRecordRef record,
    CFErrorRef *error
);
```

Parameters

addressBook

The address book to which *record* is added.

record

The record to add to *addressBook*.

error

On error, contains error information. See “[Address Book Errors](#)” (page 16).

Return Value

true when successful, false otherwise.

Availability

Available in iOS 2.0 and later.

Declared In

ABAddressBook.h

ABAddressBookCopyLocalizedLabel

Returns a localized version of a record-property label.

```
CFStringRef ABAddressBookCopyLocalizedLabel (
    CFStringRef label
);
```

Parameters*label*

The label to localize.

Return Value

The label localized to the user's locale.

Availability

Available in iOS 2.0 and later.

Declared In

ABAddressBook.h

ABAddressBookCreate

Creates a new address book with data from the Address Book database.

```
ABAddressBookRef ABAddressBookCreate (
    void
);
```

DiscussionChanges made to the returned address book are reflected in the Address Book database only after saving the address book with [ABAddressBookSave](#) (page 13).

Important: You must ensure that an instance of `ABAddressBookRef` is used by only one thread.

Availability

Available in iOS 2.0 and later.

Declared In

ABAddressBook.h

ABAddressBookHasUnsavedChanges

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

```
bool ABAddressBookHasUnsavedChanges (
    ABAddressBookRef addressBook
);
```

Parameters*addressBook*

The address book in question.

Return Value`true` when *addressBook* contains unsaved changes, `false` otherwise.**Availability**

Available in iOS 2.0 and later.

Declared In

ABAddressBook.h

ABAddressBookRegisterExternalChangeCallback

Registers a callback to receive notifications when the Address Book database is modified.

```
void ABAddressBookRegisterExternalChangeCallback (
    ABAddressBookRef addressBook,
    ABExternalChangeCallback callback,
    void *context
);
```

Parameters*addressBook*

The address book used to interact with the Address Book database.

callback

The function to invoke when the Address Book database changes.

context

The object to pass to the callback function.

Availability

Available in iOS 2.0 and later.

See Also[ABExternalChangeCallback](#) (page 14)[ABAddressBookUnregisterExternalChangeCallback](#) (page 14)**Declared In**

ABAddressBook.h

ABAddressBookRemoveRecord

Removes a record from an address book.

```
bool ABAddressBookRemoveRecord (
    ABAddressBookRef addressBook,
    ABRecordRef record,
    CFErrorRef *error
);
```

Parameters*addressBook*The address book from which *record* is to be removed.*record*The record to remove from *addressBook*.*error*On error, contains error information. See [“Address Book Errors”](#) (page 16).**Return Value**

true when successful, false otherwise.

Availability

Available in iOS 2.0 and later.

Declared In

ABAddressBook.h

ABAddressBookRevert

Discards unsaved changes in an address book.

```
void ABAddressBookRevert (
    ABAddressBookRef addressBook
);
```

Parameters

addressBook

The address book to revert.

Discussion

The address book is loaded with the information in the Address Book database.

Availability

Available in iOS 2.0 and later.

Declared In

ABAddressBook.h

ABAddressBookSave

Saves any unsaved changes to the Address Book database.

```
bool ABAddressBookSave (
    ABAddressBookRef addressBook,
    CFErrorRef *error
);
```

Parameters

addressBook

The address book to save.

error

On error, contains error information. See [“Address Book Errors”](#) (page 16).

Return Value

true when successful, false otherwise.

Availability

Available in iOS 2.0 and later.

Declared In

ABAddressBook.h

ABAddressBookUnregisterExternalChangeCallback

Unregisters a callback.

```
void ABAddressBookUnregisterExternalChangeCallback (
    ABAddressBookRef addressBook,
    ABExternalChangeCallback callback,
    void *context
);
```

Parameters

addressBook

The address book used to interact with the Address Book database.

callback

The function to invoke when the Address Book database changes. The signature of the function must match [ABExternalChangeCallback](#) (page 14).

context

The object to pass to the callback function.

Availability

Available in iOS 2.0 and later.

See Also

[ABExternalChangeCallback](#) (page 14)

[ABAddressBookRegisterExternalChangeCallback](#) (page 12)

Declared In

ABAddressBook.h

Callbacks

ABExternalChangeCallback

Prototype for a function callback invoked on an address book when the Address Book database is modified by another address book instance.

```
typedef void (*ABExternalChangeCallback) (
    ABAddressBookRef addressBook,
    CFDictionaryRef info,
    void *context
);
```

If you name your callback function `MyAddressBookExternalChangeCallback`, you declare it like this:

```
void MyAddressBookExternalChangeCallback (
    ABAddressBookRef addressBook,
    CFDictionaryRef info,
    void *context
);
```

Parameters*addressBook*

An address book used to interact with the Address Book database.

info

Always NULL.

context

The object to pass to the callback function.

Discussion

Use [ABAddressBookRegisterExternalChangeCallback](#) (page 12) to register and [ABAddressBookUnregisterExternalChangeCallback](#) (page 14) to unregister the callback function.

You can register for a callback with different contexts or callback functions. The run loop on the thread that registered the callback invokes the callback.

The *addressBook* object does not take any action to flush or synchronize cached state with the Address Book database. If you want to ensure that *addressBook* doesn't contain stale values, use [ABAddressBookRevert](#) (page 13).

Availability

Available in iOS 2.0 and later.

Declared In

ABAddressBook.h

Data Types

ABAddressBookRef

Reference to an object used to interact with the Address Book database.

```
typedef CTypeRef ABAddressBookRef;
```

Availability

Available in iOS 2.0 and later.

Declared In

ABAddressBook.h

Constants

Address Book Error Domain

This constant identifies the error domain under which Address Book errors are grouped.

```
const CFStringRef ABAddressBookErrorDomain;
```

Constants

ABAddressBookErrorDomain

The main error domain for Address Book framework operations.

Available in iOS 2.0 and later.

Declared in `ABAddressBook.h`.

Address Book Errors

Errors that can be raised under the Address Book error domain.

```
enum {  
    kABOperationNotPermittedByStoreError = 0  
};
```

Constants

kABOperationNotPermittedByStoreError

The operation is not allowed by the Address Book database.

Available in iOS 2.0 and later.

Declared in `ABAddressBook.h`.

ABMultiValue Reference

Derived From:	CTypeRef
Framework:	AddressBook/AddressBook.h
Declared in	ABMultiValue.h

Overview

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

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

Multivalued properties are immutable. To use mutable multivalued properties, see *ABMutableMultiValue Reference*.

Functions by Task

Getting Values and Labels

[ABMultiValueCopyValueAtIndex](#) (page 19)

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

[ABMultiValueCopyArrayOfAllValues](#) (page 18)

Returns an array with the values in a multivalued property.

[ABMultiValueGetCount](#) (page 19)

Returns the number of values in a multivalued property.

[ABMultiValueGetFirstIndexOfValue](#) (page 20)

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

[ABMultiValueCopyLabelAtIndex](#) (page 18)

Returns the label for a value in a multivalued property.

Getting Value Identifiers

[ABMultiValueGetIdentifierAtIndex](#) (page 20)

Returns the identifier of a value in a multivalued property.

[ABMultiValueGetIndexForIdentifier](#) (page 20)

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

Getting Property Information

[ABMultiValueGetPropertyType](#) (page 21)

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

Functions

ABMultiValueCopyArrayOfAllValues

Returns an array with the values in a multivalue property.

```
CFArrayRef ABMultiValueCopyArrayOfAllValues (
    ABMultiValueRef multiValue
);
```

Parameters

multiValue

The multivalue property whose values are being return.

Return Value

Array containing the values in *multiValue*.

Availability

Available in iOS 2.0 and later.

Declared In

ABMultiValue.h

ABMultiValueCopyLabelAtIndex

Returns the label for a value in a multivalue property.

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

Parameters

multiValue

The multivalue property with the value whose label to return.

index

Location of the value within *multiValue* whose label to return.

Raises an exception when out of bounds.

Return Value

Label for the value at *index* within *multiValue*.

Discussion

If there is no label for the value at `index`, returns `NULL`.

Availability

Available in iOS 2.0 and later.

Declared In

ABMultiValue.h

ABMultiValueCopyValueAtIndex

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

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

Parameters

multiValue

The multivalue property from which to obtain the value.

index

Location of the desired value within *multiValue*.

Raises an exception when out of bounds.

Return Value

The value at *index* in *multiValue*.

Availability

Available in iOS 2.0 and later.

Declared In

ABMultiValue.h

ABMultiValueGetCount

Returns the number of values in a multivalue property.

```
CFIndex ABMultiValueGetCount (
    ABMultiValueRef multiValue
);
```

Parameters

multiValue

The multivalue property whose value are being counted.

Return Value

The number of values in *multiValue*.

Availability

Available in iOS 2.0 and later.

Declared In

ABMultiValue.h

ABMultiValueGetFirstIndexOfValue

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

```
CFIndex ABMultiValueGetFirstIndexOfValue (
    ABMultiValueRef multiValue,
    CTypeRef value
);
```

Parameters

multiValue

The multivalue property in which to search for *value*.

value

The value to search for in *multiValue*.

Return Value

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

Availability

Available in iOS 2.0 and later.

Declared In

ABMultiValue.h

ABMultiValueGetIdentifierAtIndex

Returns the identifier of a value in a multivalue property.

```
ABMultiValueIdentifier ABMultiValueGetIdentifierAtIndex (
    ABMultiValueRef multiValue,
    CFIndex index
);
```

Parameters

multiValue

The multivalue property with the identifier to return.

index

Location of the value within *multiValue* whose identifier is being returned.

Return Value

Identifier of the value at *index* within *multiValue*.

Availability

Available in iOS 2.0 and later.

Declared In

ABMultiValue.h

ABMultiValueGetIndexForIdentifier

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

CHAPTER 2

ABMultiValue Reference

```
CFIndex ABMultiValueGetIndexForIdentifier (
    ABMultiValueRef multiValue,
    ABMultiValueIdentifier identifier
);
```

Parameters

multiValue

The multivalue property in which to look for the *identifier* value identifier.

identifier

The identifier of the value whose location within *multiValue* is being returned.

Return Value

Location within *multiValue* that contains the value with *identifier* as its identifier.

Availability

Available in iOS 2.0 and later.

Declared In

ABMultiValue.h

ABMultiValueGetPropertyType

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

```
ABPropertyType ABMultiValueGetPropertyType (
    ABMultiValueRef multiValue
);
```

Parameters

multiValue

The multivalue property whose type is being returned.

Return Value

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

Availability

Available in iOS 2.0 and later.

See Also

[ABMultiValueCreateMutable](#) (page 24)

Declared In

ABMultiValue.h

Data Types

ABMultiValueRef

Reference to a multivalue property.

```
typedef CTypeRef ABMultiValueRef;
```

Availability

Available in iOS 2.0 and later.

Declared In

ABMultiValue.h

ABMultiValueIdentifier

Identifies multivalue properties.

```
typedef int32_t ABMultiValueIdentifier;
```

Discussion

See [“Invalid Multivalue-Property Identifier”](#) (page 22).

Availability

Available in iOS 2.0 and later.

Declared In

ABMultiValue.h

Constants

Invalid Multivalue-Property Identifier

This preprocessor definition identifies an multivalue property with values whose type does not match its declared type.

```
#define kABMultiValueInvalidIdentifier -1
```

Constants

kABMultiValueInvalidIdentifier

Invalid multivalue property.

Available in iOS 2.0 and later.

Declared in ABMultiValue.h.

Discussion

See [ABMultiValueIdentifier](#) (page 22).

ABMutableMultiValue Reference

Derived From:	CTypeRef
Framework:	AddressBook/AddressBook.h
Declared in	ABMultiValue.h

Overview

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

Functions by Task

Creating Properties

[ABMultiValueCreateMutable](#) (page 24)
Returns a new, empty, mutable multivalued property.

Getting Properties

[ABMultiValueCreateMutableCopy](#) (page 25)
Returns a mutable copy of a multivalued property.

Managing Property Values

[ABMultiValueAddValueAndLabel](#) (page 24)
Adds a value and its corresponding label to a multivalued property.

[ABMultiValueReplaceValueAtIndex](#) (page 27)
Replaces a value in a multivalued property with another value.

[ABMultiValueReplaceLabelAtIndex](#) (page 27)
Replaces a label in a multivalued property with another label.

[ABMultiValueInsertValueAndLabelAtIndex](#) (page 25)
Inserts a value and a label into a multivalued property.

[ABMultiValueRemoveValueAndLabelAtIndex](#) (page 26)

Removes a value from a multivalue property.

Functions

ABMultiValueAddValueAndLabel

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

```
bool ABMultiValueAddValueAndLabel (
    ABMutableMultiValueRef multiValue,
    CTypeRef value,
    CFStringRef label,
    ABMultiValueIdentifier *outIdentifier
);
```

Parameters

multiValue

The multivalue property to add *value* and *label* to.

value

The value to add to *multiValue*.

label

The label for *value*.

outIdentifier

The address at which to place the identifier of the added value. Pass NULL to ignore the identifier.

Return Value

true when value is added to multiValue successfully, false otherwise.

Discussion

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

Availability

Available in iOS 2.0 and later.

See Also

[ABMultiValueCreateMutable](#) (page 24)

Declared In

ABMultiValue.h

ABMultiValueCreateMutable

Returns a new, empty, mutable multivalue property.


```
ABMutableMultiValueRef ABMultiValueCreateMutable (
    ABPropertyType type
);
```

Parameters

type

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

Return Value

An empty multivalued property whose values will be of type *type*.

Availability

Available in iOS 2.0 and later.

See Also

[ABMultiValueGetPropertyType](#) (page 21)

Declared In

ABMultiValue.h

ABMultiValueCreateMutableCopy

Returns a mutable copy of a multivalued property.

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

Parameters

multiValue

The multivalued property being copied.

Return Value

A mutable copy of *multiValue*.

Availability

Available in iOS 2.0 and later.

Declared In

ABMultiValue.h

ABMultiValueInsertValueAndLabelAtIndex

Inserts a value and a label into a multivalued property.

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

Parameters*multiValue*The multivalue property into which to insert *value*.*value*

The value to insert.

label

The label to insert.

*index*The location, in *multiValue*, at which to insert *value* and *label*.

Raises an exception when out of bounds.

outIdentifier

On output, the identifier of the added value.

Return Value`true` when value and label are inserted successfully into *multiValue*, `false` otherwise.**Discussion**

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

Availability

Available in iOS 2.0 and later.

See Also[ABMultiValueCreateMutable](#) (page 24)**Declared In**

ABMultiValue.h

ABMultiValueRemoveValueAndLabelAtIndex

Removes a value from a multivalue property.

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

Parameters*multiValue*The multivalue property from which to remove the value at *index*.*index*The location, in *multiValue*, of the value being removed.

Raises an exception when out of bounds.

Return Value

true when successful, false otherwise.

Availability

Available in iOS 2.0 and later.

Declared In

ABMultiValue.h

ABMultiValueReplaceLabelAtIndex

Replaces a label in a multivalue property with another label.

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

Parameters

multiValue

The multivalue property containing the label to replace.

label

The new label.

index

The location, in *multiValue*, of the label to replace.

Raises an exception when out of bounds.

Return Value

true when successful, false otherwise.

Availability

Available in iOS 2.0 and later.

Declared In

ABMultiValue.h

ABMultiValueReplaceValueAtIndex

Replaces a value in a multivalue property with another value.

```
bool ABMultiValueReplaceValueAtIndex (
    ABMutableMultiValueRef multiValue,
    CFTyperef value,
    CFIndex index
);
```

Parameters

multiValue

The multivalue property containing the value to replace.

value

The new value.

index

The location, in *multiValue*, of the value to replace.

Raises an exception when out of bounds.

Return Value

true when successful, false otherwise.

Availability

Available in iOS 2.0 and later.

Declared In

ABMultiValue.h

Data Types

ABMutableMultiValueRef

Reference to a mutable multivalued property.

```
typedef CTypeRef ABMutableMultiValueRef;
```

Availability

Available in iOS 2.0 and later.

Declared In

ABMultiValue.h

ABRecord Reference

Derived From:	CTypeRef
Framework:	AddressBook/AddressBook.h
Declared in	ABRecord.h

Overview

The `ABRecord` opaque type (whose objects are known as **records**) provides a common interface to the records in the Address Book database. It defines common properties for all records. A **property** is a field in a record such as the first or last name of a person record.

Each record in the Address Book database has a unique ID, which you obtain using the [ABRecordGetRecordID](#) (page 31) function.

Functions by Task

Getting Record Information

- [ABRecordGetRecordID](#) (page 31)
Returns the unique ID of a record.
- [ABRecordGetRecordType](#) (page 31)
Returns the type of a record.

Managing Property Values

- [ABRecordSetValue](#) (page 32)
Sets the value of a record property.
- [ABRecordCopyValue](#) (page 30)
Returns the value of a record property.
- [ABRecordRemoveValue](#) (page 32)
Removes the value of a record property.
- [ABRecordCopyCompositeName](#) (page 30)
Returns an appropriate, human-friendly name for the record.

Functions

ABRecordCopyCompositeName

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

```
CFStringRef ABRecordCopyCompositeName (
    ABRecordRef record
);
```

Parameters

record

The record whose name is being returned.

Return Value

- For person records: The concatenated value of these properties: Prefix, Suffix, Organization, First name, and Last name.
- For group records: The value of the group name property ([kABGroupNameProperty](#) (page 47)).

Discussion

The behavior of this function is not defined for source records.

Availability

Available in iOS 2.0 and later.

Declared In

ABRecord.h

ABRecordCopyValue

Returns the value of a record property.

```
CTypeRef ABRecordCopyValue (
    ABRecordRef record,
    ABPropertyID property
);
```

Parameters

record

The record containing the property in question.

property

The property of *record* whose value is being returned. See properties in [“Constants”](#) (page 60) and [“Constants”](#) (page 47).

Return Value

The value of *property* in *record*.

Availability

Available in iOS 2.0 and later.

Declared In

ABRecord.h

ABRecordGetRecordID

Returns the unique ID of a record.

```
ABRecordID ABRecordGetRecordID (  
    ABRecordRef record  
);
```

Parameters

record

The record whose ID is being returned.

Return Value

The unique ID of *record* when the record exists in the Address Book database.

[kABRecordInvalidID](#) (page 37) when the record hasn't been saved to the database.

Availability

Available in iOS 2.0 and later.

Declared In

ABRecord.h

ABRecordGetRecordType

Returns the type of a record.

```
ABRecordType ABRecordGetRecordType (  
    ABRecordRef record  
);
```

Parameters

record

The record whose type is being returned.

Return Value

The record type of *record*, which can be one of these:

- `kABPersonType` for person records
- `kABGroupType` for group records.
- `kABSourceType` for source records.

Availability

Available in iOS 2.0 and later.

See Also

[ABPersonCreate](#) (page 56)

[ABGroupCreate](#) (page 45)

Declared In

ABRecord.h

ABRecordRemoveValue

Removes the value of a record property.

```
bool ABRecordRemoveValue (
    ABRecordRef record,
    ABPropertyID property,
    CFErrorRef *error
);
```

Parameters

record

The record with the property in question.

property

The property, in *record*, whose value is being removed. See properties in “Constants” (page 60) and “Constants” (page 47).

error

On failure, information about its cause.

Return Value

true on success, false otherwise.

Availability

Available in iOS 2.0 and later.

Declared In

ABRecord.h

ABRecordSetValue

Sets the value of a record property.

```
bool ABRecordSetValue (
    ABRecordRef record,
    ABPropertyID property,
    CFTyperef value,
    CFErrorRef *error
);
```

Parameters

record

The record containing the property in question.

property

The property whose value is being set. See properties in “Constants” (page 60) and “Constants” (page 47).

value

The new value. Pass NULL to remove the property from *record*.

error

On failure, information about its cause.

Return Value

true when successful, false otherwise.

Availability

Available in iOS 2.0 and later.

Declared In

ABRecord.h

Data Types

ABRecordRef

Reference to a record.

```
typedef CTypeRef ABRecordRef;
```

Availability

Available in iOS 2.0 and later.

Declared In

ABRecord.h

ABRecordID

Integer that identifies a record.

```
typedef int32_t ABRecordID;
```

Discussion

Valid record IDs are positive integers.

See [“Invalid Record ID”](#) (page 36).

Availability

Available in iOS 2.0 and later.

Declared In

ABRecord.h

ABPropertyID

Integer that identifies a record property.

```
typedef int32_t ABPropertyID;
```

Discussion

See properties listed in [“Constants”](#) (page 60) and [“Constants”](#) (page 47)

Availability

Available in iOS 2.0 and later.

Declared In

ABRecord.h

ABRecordType

Integer that identifies a record type.

```
typedef uint32_t ABRecordType;
```

Discussion

See “[Record Types](#)” (page 34).

Availability

Available in iOS 2.0 and later.

Declared In

ABRecord.h

ABPropertyType

Integer that identifies a record property type.

```
typedef uint16_t ABPropertyType;
```

Discussion

See “[Record Property Types](#)” (page 35).

Availability

Available in iOS 2.0 and later.

Declared In

ABRecord.h

Constants

Record Types

These constants identify record types.

```
enum {  
    kABPersonType = 0,  
    kABGroupType = 1  
};
```

Constants

kABPersonType

Person record.

Available in iOS 2.0 and later.

Declared in ABRecord.h.

kABGroupType

Group record.

Available in iOS 2.0 and later.

Declared in ABRecord.h.

Discussion

See [ABRecordType](#) (page 34).

Multivalue Properties

Indicates a multivalue property.

```
#define kABMultiValueMask (1 << 8)
```

Constants

`kABMultiValueMask`

Indicates a multivalue property.

Available in iOS 2.0 and later.

Declared in `ABRecord.h`.

Record Property Types

These constants identify record property types.

```
enum {
    kABInvalidPropertyType           = 0x0,
    kABStringPropertyType            = 0x1,
    kABIntegerPropertyType           = 0x2,
    kABRealPropertyType              = 0x3,
    kABDateTimePropertyType          = 0x4,
    kABDictionaryPropertyType        = 0x5,
    kABMultiStringPropertyType       = kABMultiValueMask | kABStringPropertyType
    kABMultiIntegerPropertyType      = kABMultiValueMask | kABIntegerPropertyType
    kABMultiRealPropertyType         = kABMultiValueMask | kABRealPropertyType
    kABMultiDateTimePropertyType     = kABMultiValueMask | kABDateTimePropertyType
    kABMultiDictionaryPropertyType   = kABMultiValueMask | kABDictionaryPropertyType
};
```

Constants

`kABInvalidPropertyType`

Invalid property.

Available in iOS 2.0 and later.

Declared in `ABRecord.h`.

`kABStringPropertyType`

String property. Returned as a `CFStringRef` object.

Available in iOS 2.0 and later.

Declared in `ABRecord.h`.

`kABIntegerPropertyType`

Integer property. Returned as a `CFNumberRef` object.

Available in iOS 2.0 and later.

Declared in `ABRecord.h`.

`kABRealPropertyType`

Real property. Returned as a `CFNumberRef` object.

Available in iOS 2.0 and later.

Declared in `ABRecord.h`.

`kABDateTimePropertyType`

Date-time property. Returned as a `CFDateRef` object.

Available in iOS 2.0 and later.

Declared in `ABRecord.h`.

`kABDictionaryPropertyType`

Dictionary property. Returned as a `CFDictionaryRef` object.

Available in iOS 2.0 and later.

Declared in `ABRecord.h`.

`kABMultiStringPropertyType`

Multistring property. Returned as an [ABMultiValueIdentifier](#) (page 22) of `CFStringRef` objects.

Available in iOS 2.0 and later.

Declared in `ABRecord.h`.

`kABMultiIntegerPropertyType`

Multiinteger property. Returned as an [ABMultiValueIdentifier](#) (page 22) of `CFNumberRef` objects.

Available in iOS 2.0 and later.

Declared in `ABRecord.h`.

`kABMultiRealPropertyType`

Multireal property. Returned as an [ABMultiValueIdentifier](#) (page 22) of `CFNumberRef` objects.

Available in iOS 2.0 and later.

Declared in `ABRecord.h`.

`kABMultiDateTimePropertyType`

Multi-date-time property. Returned as an [ABMultiValueIdentifier](#) (page 22) of `CFDateRef` objects.

Available in iOS 2.0 and later.

Declared in `ABRecord.h`.

`kABMultiDictionaryPropertyType`

Multidictionary property. Returned as an [ABMultiValueIdentifier](#) (page 22) of `CFDictionaryRef` objects.

Available in iOS 2.0 and later.

Declared in `ABRecord.h`.

Discussion

See [ABPropertyType](#) (page 34).

Invalid Values

These macros identify invalid values for record and property identifiers.

```
#define kABRecordInvalidID -1  
#define kABPropertyInvalidID -1
```

Constants

`kABRecordInvalidID`

Records with this ID have not been saved to the Address Book database.

Available in iOS 2.0 and later.

Declared in `ABRecord.h`.

`kABPropertyInvalidID`

Indicates an invalid value for a property ID.

Available in iOS 3.0 and later.

Declared in `ABRecord.h`.

Other References

ABGroup Reference

Framework:	AddressBook/AddressBook.h
Declared in	ABGroup.h

Overview

Group records are represented by instances of the [ABRecordRef](#) (page 33) opaque type, whose record type is `kABGroupType`. Person records may belong to multiple groups. Groups may not contain other groups.

Functions by Task

Creating Groups

[ABGroupCreate](#) (page 45)

Creates a group record.

[ABGroupCreateInSource](#) (page 46)

Creates a group in a particular source.

Managing Group Members

[ABGroupCopyArrayOfAllMembers](#) (page 44)

Returns the records in a group.

[ABGroupCopyArrayOfAllMembersWithSortOrdering](#) (page 44)

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

[ABGroupAddMember](#) (page 44)

Adds a person record to a group.

[ABGroupRemoveMember](#) (page 46)

Removes a record from a group.

Accessing the Groups in an Address Book

[ABAddressBookGetGroupWithRecordID](#) (page 43)

Returns the group with a given record ID.

[ABAddressBookGetGroupCount](#) (page 43)

Returns the number of groups in an address book.

[ABAddressBookCopyArrayOfAllGroups](#) (page 42)

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

[ABAddressBookCopyArrayOfAllGroupsInSource](#) (page 42)

Returns an array of all groups from a particular source.

[ABGroupCopySource](#) (page 45)

Returns the source that the group is from.

Functions

ABAddressBookCopyArrayOfAllGroups

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

```
CFArrayRef ABAddressBookCopyArrayOfAllGroups (
    ABAddressBookRef addressBook
);
```

Parameters

addressBook

The address book from which to return the groups.

Return Value

An array containing the groups in *addressBook*.

Availability

Available in iOS 2.0 and later.

Declared In

ABGroup.h

ABAddressBookCopyArrayOfAllGroupsInSource

Returns an array of all groups from a particular source.

```
CFArrayRef ABAddressBookCopyArrayOfAllGroupsInSource (
    ABAddressBookRef addressBook,
    ABRecordRef source
);
```

Parameters

addressBook

The address book whose records are being returned.

source

The source whose groups are being returned.

Return Value

An array of all groups from a particular source.

Availability

Available in iOS 4.0 and later.

Declared In

ABGroup.h

ABAddressBookGetGroupCount

Returns the number of groups in an address book.

```
CFIndex ABAddressBookGetGroupCount (
    ABAddressBookRef addressBook
);
```

Parameters

addressBook

The address book whose record count is being returned.

Return Value

The number of groups in *addressBook*.

Availability

Available in iOS 2.0 and later.

Declared In

ABGroup.h

ABAddressBookGetGroupWithRecordID

Returns the group with a given record ID.

```
ABRecordRef ABAddressBookGetGroupWithRecordID (
    ABAddressBookRef addressBook,
    ABRecordID recordID
);
```

Parameters

addressBook

The address book in which to look for the record.

recordID

The record ID of the group being returned.

Return Value

If found, the group with *recordID* as its identifier; otherwise, NULL.

Availability

Available in iOS 2.0 and later.

Declared In

ABGroup.h

ABGroupAddMember

Adds a person record to a group.

```
bool ABGroupAddMember (  
    ABRecordRef group,  
    ABRecordRef person,  
    CFErrorRef *error  
);
```

Parameters

group

The group that the person record is being added to.

person

The person record to add to *group*.

error

On error, information about the cause.

Return Value

true when person is added to group; otherwise false.

Discussion

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

Availability

Available in iOS 2.0 and later.

Declared In

ABGroup.h

ABGroupCopyArrayOfAllMembers

Returns the records in a group.

```
CFArrayRef ABGroupCopyArrayOfAllMembers (  
    ABRecordRef group  
);
```

Parameters

group

The group whose records are being returned.

Return Value

An array containing the records in *group*.

Availability

Available in iOS 2.0 and later.

Declared In

ABGroup.h

ABGroupCopyArrayOfAllMembersWithSortOrdering

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

```

NSArrayRef ABGroupCopyArrayOfAllMembersWithSortOrdering (
    ABRecordRef group,
    ABPersonSortOrdering sortOrdering
);

```

Parameters*group*

The group whose records are being returned.

*sortOrdering*The order in which you want the returned records. See [“Sort Order”](#) (page 60).**Return Value**An array containing the records in *group* using *sortOrdering*.**Availability**

Available in iOS 2.0 and later.

Declared In

ABGroup.h

ABGroupCopySource

Returns the source that the group is from.

```

ABRecordRef ABGroupCopySource (
    ABRecordRef group
);

```

Parameters*group*

The group whose source is being returned.

Return Value

The source that the group record is from.

Availability

Available in iOS 4.0 and later.

Declared In

ABGroup.h

ABGroupCreate

Creates a group record.

```

ABRecordRef ABGroupCreate (
    void
);

```

Availability

Available in iOS 2.0 and later.

Declared In

ABGroup.h

ABGroupCreateInSource

Creates a group in a particular source.

```
ABRecordRef ABGroupCreateInSource (
    ABRecordRef source
);
```

Parameters

source

The source to create the group in.

Return Value

A new group in the specified source.

Discussion

If *source* is `NULL`, the group is created in the default source.

Availability

Available in iOS 4.0 and later.

Declared In

ABGroup.h

ABGroupRemoveMember

Removes a record from a group.

```
bool ABGroupRemoveMember (
    ABRecordRef group,
    ABRecordRef member,
    CFErrorRef *error
);
```

Parameters

group

The group from which to remove record.

record

The record to remove from *group*.

error

On error, information about the cause.

Return Value

`true` when *record* is removed from *group*; otherwise `false`.

Availability

Available in iOS 2.0 and later.

Declared In

ABGroup.h

Constants

Group Properties

This constant identifies the properties of groups.

```
const int kABGroupNameProperty;
```

Constants

`kABGroupNameProperty`

Group name property.

Available in iOS 2.0 and later.

Declared in `ABGroup.h`.

ABPerson Reference

Framework:	AddressBook/AddressBook.h
Declared in	ABPerson.h

Overview

Person records are represented by instances of the [ABRecordRef](#) (page 33) opaque type, whose record type is `kABPersonType`. Person records store information about a person, such as first name, last name, address, email, and phone number.

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

Functions by Task

Creating Persons

[ABPersonCreate](#) (page 56)

Creates a new person record in the default source.

[ABPersonCreateInSource](#) (page 56)

Creates a new person record in a particular source.

Sorting Person Records

[ABPersonComparePeopleByName](#) (page 53)

Indicates how two person records get sorted.

Getting Person Property Information

[ABPersonGetTypeOfProperty](#) (page 57)

Returns the type of a person property.

[ABPersonCopyLocalizedPropertyName](#) (page 55)

Returns the localized name of a person property

Managing Pictures

- [ABPersonSetImageData](#) (page 58)
Sets a person's picture.
- [ABPersonCopyImageData](#) (page 54)
Returns the picture for a person.
- [ABPersonHasImageData](#) (page 57)
Indicates whether a person has a picture.
- [ABPersonRemoveImageData](#) (page 58)
Removes a person's picture.

Accessing the Persons in an Address Book

- [ABAddressBookGetPersonCount](#) (page 52)
Returns the number of person records in an address book.
- [ABAddressBookGetPersonWithRecordID](#) (page 53)
Returns the person record with a given record ID.
- [ABAddressBookCopyArrayOfAllPeople](#) (page 51)
Returns all the person records in an address book.
- [ABAddressBookCopyArrayOfAllPeopleInSource](#) (page 51)
Returns an array of all person records from a particular source.
- [ABAddressBookCopyArrayOfAllPeopleInSourceWithSortOrdering](#) (page 51)
Returns an array of all person records in the address book, sorted with the specified order.
- [ABAddressBookCopyPeopleWithName](#) (page 52)
Performs a prefix search on the composite names of people in an address book and returns an array of persons that match the search criteria.
- [ABPersonCopyArrayOfAllLinkedPeople](#) (page 54)
Returns an array of all person records in the address book database that are linked to the given person record.
- [ABPersonCopySource](#) (page 55)
Returns the source that the person record is from.

Getting Name Format and Sort Preferences

- [ABPersonGetSortOrdering](#) (page 57)
Returns the user's sort-ordering preference for lists of persons.
- [ABPersonGetCompositeNameFormat](#) (page 56)
Returns the person-name display format.

Functions

ABAddressBookCopyArrayOfAllPeople

Returns all the person records in an address book.

```
CFArrayRef ABAddressBookCopyArrayOfAllPeople (
    ABAddressBookRef addressBook
);
```

Parameters

addressBook

The address book whose person records are being returned.

Return Value

Array containing the person records in *addressBook*.

Availability

Available in iOS 2.0 and later.

Declared In

ABPerson.h

ABAddressBookCopyArrayOfAllPeopleInSource

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

```
CFArrayRef ABAddressBookCopyArrayOfAllPeopleInSource (
    ABAddressBookRef addressBook,
    ABRecordRef source
);
```

Parameters

addressBook

The address book whose records are being returned.

source

The source whose records are being returned.

Return Value

An array of all person records from a particular source.

Availability

Available in iOS 4.0 and later.

Declared In

ABPerson.h

ABAddressBookCopyArrayOfAllPeopleInSourceWithSortOrdering

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

```

NSArrayRef ABAddressBookCopyArrayOfAllPeopleInSourceWithSortOrdering (
    ABAddressBookRef addressBook,
    ABRecordRef source,
    ABPersonSortOrdering sortOrdering
);

```

Parameters*addressBook*

The address book whose person records are being returned.

source

The source whose records are being returned.

*sortOrdering*Indicates whether to sort by first name or by last name. See “[Sort Order](#)” (page 60).**Return Value**An array of all person records in the address book database, sorted by *sortOrdering*.**Availability**

Available in iOS 4.0 and later.

Declared In

ABPerson.h

ABAddressBookCopyPeopleWithName

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

```

NSArrayRef ABAddressBookCopyPeopleWithName (
    ABAddressBookRef addressBook,
    CFStringRef name
);

```

Parameters*addressBook*

The address book in which to search.

name

The search term. Examples: "Mary", "Smi", "Hu Yan".

Return ValueArray containing person records (ABRecordRef objects) that match *name*.**Availability**

Available in iOS 2.0 and later.

Declared In

ABPerson.h

ABAddressBookGetPersonCount

Returns the number of person records in an address book.

```
CFIndex ABAddressBookGetPersonCount (
    ABAddressBookRef addressBook
);
```

Parameters

addressBook

The address book whose person records are being counted.

Return Value

The number of person records in *addressBook*.

Availability

Available in iOS 2.0 and later.

Declared In

ABPerson.h

ABAddressBookGetPersonWithRecordID

Returns the person record with a given record ID.

```
ABRecordRef ABAddressBookGetPersonWithRecordID (
    ABAddressBookRef addressBook,
    ABRecordID recordID
);
```

Parameters

addressBook

The address book in which to search for *recordID*.

recordID

The record identifier to seek in *addressBook*.

Return Value

If found, the person record with *recordID* as its identifier, otherwise NULL.

Availability

Available in iOS 2.0 and later.

Declared In

ABPerson.h

ABPersonComparePeopleByName

Indicates how two person records get sorted.

```
CFComparisonResult ABPersonComparePeopleByName (
    ABRecordRef person1,
    ABRecordRef person2,
    ABPersonSortOrdering ordering
);
```

Parameters

person1

The first person.

person2

The second person.

ordering

Indicates whether to sort by first name or by last name. See “Sort Order” (page 60).

Return Value

- `kCFCompareLessThan` when *person1* goes before *person2*.
- `kCFCompareEqualTo` when *person1* and *person2* have the same name.
- `kCFCompareGreaterThan` when *person1* goes after *person2*.

Availability

Available in iOS 2.0 and later.

Declared In

ABPerson.h

ABPersonCopyArrayOfAllLinkedPeople

Returns an array of all person records in the address book database that are linked to the given person record.

```
CFArrayRef ABPersonCopyArrayOfAllLinkedPeople (
    ABRecordRef person
);
```

Parameters

person

The person record whose linked records are being returned.

Return Value

An array of all person records in the address book database that are linked to the given person record

Discussion

Linked records contain information about the same person, and usually come from different sources.

Availability

Available in iOS 4.0 and later.

Declared In

ABPerson.h

ABPersonCopyImageData

Returns the picture for a person.

```
CFDataRef ABPersonCopyImageData (
    ABRecordRef person
);
```

Parameters

person

The person record whose picture is being returned.

Return Value

The picture for *person*. NULL when the person has no picture.

Availability

Available in iOS 2.0 and later.

Declared In

ABPerson.h

ABPersonCopyLocalizedPropertyName

Returns the localized name of a person property

```
CFStringRef ABPersonCopyLocalizedPropertyName (  
    ABPropertyID property  
);
```

Parameters

property

The person property whose localized name is being returned.

Return Value

The localized name of *property*.

Availability

Available in iOS 2.0 and later.

Declared In

ABPerson.h

ABPersonCopySource

Returns the source that the person record is from.

```
ABRecordRef ABPersonCopySource (  
    ABRecordRef person  
);
```

Parameters

person

The person whose source is being returned.

Return Value

The source of the person record.

Availability

Available in iOS 4.0 and later.

Declared In

ABPerson.h

ABPersonCreate

Creates a new person record in the default source.

```
ABRecordRef ABPersonCreate (
    void
);
```

Discussion

To add the newly created person record to the Address Book database, use [ABAddressBookAddRecord](#) (page 10).

Availability

Available in iOS 2.0 and later.

Declared In

ABPerson.h

ABPersonCreateInSource

Creates a new person record in a particular source.

```
ABRecordRef ABPersonCreateInSource (
    ABRecordRef source
);
```

Parameters

source

The source to create the person record in.

Return Value

A new person record in the specified source.

Discussion

To add the newly created person record to the Address Book database, use [ABAddressBookAddRecord](#) (page 10).

If *source* is NULL, the group is created in the default source.

Availability

Available in iOS 4.0 and later.

Declared In

ABPerson.h

ABPersonGetCompositeNameFormat

Returns the person-name display format.

```
ABPersonCompositeNameFormat ABPersonGetCompositeNameFormat (
    void
);
```

Return Value

A constant indicating person-name display order; see “[Composite Name Format](#)” (page 60).

Availability

Available in iOS 2.0 and later.

Declared In

ABPerson.h

ABPersonGetSortOrdering

Returns the user's sort-ordering preference for lists of persons.

```
ABPersonSortOrdering ABPersonGetSortOrdering (
    void
);
```

Return Value

The user's sort ordering preference; see “Sort Order” (page 60).

Availability

Available in iOS 2.0 and later.

Declared In

ABPerson.h

ABPersonGetTypeOfProperty

Returns the type of a person property.

```
ABPropertyType ABPersonGetTypeOfProperty (
    ABPropertyID property
);
```

Parameters

property

The person property whose type is being returned.

Return Value

The type of *property*.

Availability

Available in iOS 2.0 and later.

Declared In

ABPerson.h

ABPersonHasImageData

Indicates whether a person has a picture.

```
bool ABPersonHasImageData (  
    ABRecordRef person  
);
```

Parameters

person

The person record in question.

Return Value

true when *person* has a picture, false otherwise.

Availability

Available in iOS 2.0 and later.

Declared In

ABPerson.h

ABPersonRemoveImageData

Removes a person's picture.

```
bool ABPersonRemoveImageData (  
    ABRecordRef person,  
    CFErrorRef *error  
);
```

Parameters

person

The person record whose picture is being removed.

error

On error, information about the cause.

Return Value

true when successful, false otherwise.

Availability

Available in iOS 2.0 and later.

Declared In

ABPerson.h

ABPersonSetImageData

Sets a person's picture.

```
bool ABPersonSetImageData (  
    ABRecordRef person,  
    CFDataRef imageData,  
    CFErrorRef *error  
);
```

Parameters

person

The person record whose picture is being set.

imageData

The picture to assign to *person*.

error

On error, information about the cause.

Return Value

true when successful, false otherwise.

Availability

Available in iOS 2.0 and later.

Declared In

ABPerson.h

Data Types

ABPersonSortOrdering

Indicates a person sort ordering.

```
typedef uint32_t ABPersonSortOrdering;
```

Discussion

See “[Sort Order](#)” (page 60).

Availability

Available in iOS 2.0 and later.

Declared In

ABPerson.h

ABPersonCompositeNameFormat

Indicates a person-name display format.

```
typedef uint32_t ABPersonCompositeNameFormat;
```

Discussion

See “[Composite Name Format](#)” (page 60).

Availability

Available in iOS 2.0 and later.

Declared In

ABPerson.h

Constants

Sort Order

These constants identify sort order for person lists.

```
enum {
    kABPersonSortByFirstName = 0,
    kABPersonSortByLastName = 1
};
```

Constants

`kABPersonSortByFirstName`

Order by first name.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonSortByLastName`

Order by last name.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

Discussion

The value of these constants is undefined until one of the following functions has been called: [ABAddressBookCreate](#) (page 11), [ABPersonCreate](#) (page 56), [ABGroupCreate](#) (page 45).

See [ABPersonSortOrdering](#) (page 59) and [ABPersonGetSortOrdering](#) (page 57).

Composite Name Format

These constants identify the format to use for a person's composite name order for persons.

```
enum {
    kABPersonCompositeNameFormatFirstNameFirst = 0,
    kABPersonCompositeNameFormatLastNameFirst = 1
};
```

Constants

`kABPersonCompositeNameFormatFirstNameFirst`

First name first.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonCompositeNameFormatLastNameFirst`

Last name first.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

Discussion

The value of these constants is undefined until one of the following functions has been called: [ABAddressBookCreate](#) (page 11), [ABPersonCreate](#) (page 56), [ABGroupCreate](#) (page 45).

See [ABPersonCompositeNameFormat](#) (page 59) and [ABRecordCopyCompositeName](#) (page 30).

Personal Information Properties

These constants identify person properties used to store personal information.

```
const ABPropertyID kABPersonFirstNameProperty;
const ABPropertyID kABPersonLastNameProperty;
const ABPropertyID kABPersonMiddleNameProperty;
const ABPropertyID kABPersonPrefixProperty;
const ABPropertyID kABPersonSuffixProperty;
const ABPropertyID kABPersonNicknameProperty;
const ABPropertyID kABPersonFirstNamePhoneticProperty;
const ABPropertyID kABPersonLastNamePhoneticProperty;
const ABPropertyID kABPersonMiddleNamePhoneticProperty;
const ABPropertyID kABPersonOrganizationProperty;
const ABPropertyID kABPersonJobTitleProperty;
const ABPropertyID kABPersonDepartmentProperty;
const ABPropertyID kABPersonEmailProperty;
const ABPropertyID kABPersonBirthdayProperty;
const ABPropertyID kABPersonNoteProperty;
const ABPropertyID kABPersonCreationDateProperty;
const ABPropertyID kABPersonModificationDateProperty;
```

Constants

`kABPersonFirstNameProperty`

First name. Type: `kABStringPropertyType`.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonLastNameProperty`

Last name. Type: `kABStringPropertyType`.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonMiddleNameProperty`

Middle name. Type: `kABStringPropertyType`.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonPrefixProperty`

Prefix (“Sir,” “Duke,” “General”). Type: `kABStringPropertyType`.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonSuffixProperty`

Suffix (“Jr.,” “Sr.,” “III”). Type: `kABStringPropertyType`.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

- `kABPersonNicknameProperty`
Nickname. Type: `kABStringPropertyType`.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.
- `kABPersonFirstNamePhoneticProperty`
First name phonetic. Type: `kABStringPropertyType`.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.
- `kABPersonLastNamePhoneticProperty`
Last name phonetic. Type: `kABStringPropertyType`.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.
- `kABPersonMiddleNamePhoneticProperty`
Middle name phonetic. Type: `kABStringPropertyType`.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.
- `kABPersonOrganizationProperty`
Organization name. Type: `kABStringPropertyType`.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.
- `kABPersonJobTitleProperty`
Job title. Type: `kABStringPropertyType`.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.
- `kABPersonDepartmentProperty`
Department. Type: `kABStringPropertyType`.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.
- `kABPersonEmailProperty`
Email address. Type: `kABMultiStringPropertyType`.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.
- `kABPersonBirthdayProperty`
Birthday. Type: `kABDateTimePropertyType`.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.
- `kABPersonNoteProperty`
Note. Type: `kABStringPropertyType`.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.

`kABPersonCreationDateProperty`
Creation date. Type: `kABDateTimePropertyType`.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.

`kABPersonModificationDateProperty`
Modification date. Type: `kABDateTimePropertyType`.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.

Discussion

The value of these constants is undefined until one of the following functions has been called: [ABAddressBookCreate](#) (page 11), [ABPersonCreate](#) (page 56), [ABGroupCreate](#) (page 45).

Address Property

These constants implement the person address property (a multivalue property of type `kABMultiDictionaryPropertyType`) and the labels used to identify each of the property's values.

```
const ABPropertyID kABPersonAddressProperty;
const CFStringRef kABPersonAddressStreetKey;
const CFStringRef kABPersonAddressCityKey;
const CFStringRef kABPersonAddressStateKey;
const CFStringRef kABPersonAddressZIPKey;
const CFStringRef kABPersonAddressCountryKey;
const CFStringRef kABPersonAddressCountryCodeKey;
```

Constants

`kABPersonAddressProperty`
Identifier for the address multivalue property.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.

`kABPersonAddressStreetKey`
Street.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.

`kABPersonAddressCityKey`
City.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.

`kABPersonAddressStateKey`
State.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.

`kABPersonAddressZIPKey`
Zip code.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.

`kABPersonAddressCountryKey`
Country.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonAddressCountryCodeKey`

Country code. The supported values are listed in “Country Codes”.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

Discussion

The value of these constants is undefined until one of the following functions has been called:

[ABAddressBookCreate](#) (page 11), [ABPersonCreate](#) (page 56), [ABGroupCreate](#) (page 45).

Date Properties

These constants implement the person dates property (a multivalue property of type `kABMultiDateTimePropertyType`) and the labels used to identify each of the property’s values.

```
const ABPropertyID kABPersonDateProperty;
const CFStringRef kABPersonAnniversaryLabel;
```

Constants

`kABPersonDateProperty`

Identifier for the dates multivalue property.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonAnniversaryLabel`

Birthdate.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

Discussion

The value of these constants is undefined until one of the following functions has been called:

[ABAddressBookCreate](#) (page 11), [ABPersonCreate](#) (page 56), [ABGroupCreate](#) (page 45).

Person Type Property

These constants implement the person type property (a property of type `kABIntegerPropertyType`), which indicates whether a person record represents a human being or an organization.

```
const ABPropertyID kABPersonKindProperty;
const CFNumberRef kABPersonKindPerson;
const CFNumberRef kABPersonKindOrganization;
```

Constants

`kABPersonKindProperty`

Identifier for the type property.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonKindPerson`
Identifies a person.
 Available in iOS 2.0 and later.
 Declared in `ABPerson.h`.

`kABPersonKindOrganization`
Identifies an organization.
 Available in iOS 2.0 and later.
 Declared in `ABPerson.h`.

Discussion

The value of these constants is undefined until one of the following functions has been called: [ABAddressBookCreate](#) (page 11), [ABPersonCreate](#) (page 56), [ABGroupCreate](#) (page 45).

Phone Number Property

These constants implement the person phone number property (a multivalue property of type `kABMultiStringPropertyType`) and the labels used to identify each of the property's values.

```
const ABPropertyID kABPersonPhoneProperty;
const CFStringRef kABPersonPhoneMobileLabel;
const CFStringRef kABPersonPhoneIPhoneLabel;
const CFStringRef kABPersonPhoneMainLabel;
const CFStringRef kABPersonPhoneHomeFAXLabel;
const CFStringRef kABPersonPhoneWorkFAXLabel;
const CFStringRef kABPersonPhonePagerLabel;
```

Constants

`kABPersonPhoneProperty`
Identifier for the phone number multivalue property.
 Available in iOS 2.0 and later.
 Declared in `ABPerson.h`.

`kABPersonPhoneMobileLabel`
Mobile phone number.
 Available in iOS 2.0 and later.
 Declared in `ABPerson.h`.

`kABPersonPhoneIPhoneLabel`
iPhone number.
 Available in iOS 3.0 and later.
 Declared in `ABPerson.h`.

`kABPersonPhoneMainLabel`
Main phone number.
 Available in iOS 2.0 and later.
 Declared in `ABPerson.h`.

`kABPersonPhoneHomeFAXLabel`
Fax phone number.
 Available in iOS 2.0 and later.
 Declared in `ABPerson.h`.

`kABPersonPhoneWorkFAXLabel`
Work phone number.
 Available in iOS 2.0 and later.
 Declared in `ABPerson.h`.

`kABPersonPhonePagerLabel`
Pager phone number.
 Available in iOS 2.0 and later.
 Declared in `ABPerson.h`.

Discussion

The value of these constants is undefined until one of the following functions has been called: [ABAddressBookCreate](#) (page 11), [ABPersonCreate](#) (page 56), [ABGroupCreate](#) (page 45).

Instant Message Property

These constants implement the person instant message property (a multivalue property of type `ABMultiDictionaryPropertyType`) and the labels used to identify each of the property's values.

```
const ABPropertyID kABPersonInstantMessageProperty;
const CFStringRef kABPersonInstantMessageServiceKey;
const CFStringRef kABPersonInstantMessageServiceYahoo;
const CFStringRef kABPersonInstantMessageServiceJabber;
const CFStringRef kABPersonInstantMessageServiceMSN;
const CFStringRef kABPersonInstantMessageServiceICQ;
const CFStringRef kABPersonInstantMessageServiceAIM;
const CFStringRef kABPersonInstantMessageUsernameKey;
```

Constants

`kABPersonInstantMessageProperty`
Identifier for the instant message multivalue property.
 Available in iOS 2.0 and later.
 Declared in `ABPerson.h`.

`kABPersonInstantMessageServiceKey`
Instant message service.
 Available in iOS 2.0 and later.
 Declared in `ABPerson.h`.

`kABPersonInstantMessageServiceYahoo`
Yahoo instant message service.
 Available in iOS 2.0 and later.
 Declared in `ABPerson.h`.

`kABPersonInstantMessageServiceJabber`
Jabber instant message service.
 Available in iOS 2.0 and later.
 Declared in `ABPerson.h`.

`kABPersonInstantMessageServiceMSN`
MSN instant message service.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonInstantMessageServiceICQ`
ICQ instant message service.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonInstantMessageServiceAIM`
AIM instant message service.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonInstantMessageUsernameKey`
Instant message service username.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

Discussion

The value of these constants is undefined until one of the following functions has been called: [ABAddressBookCreate](#) (page 11), [ABPersonCreate](#) (page 56), [ABGroupCreate](#) (page 45).

An instant message multivalue property may have the following structure:

```
kABPersonInstantMessageServiceKey
    Yahoo
kABPersonInstantMessageUsernameKey
    <#username#>@yahoo.com
kABPersonInstantMessageServiceKey
    AIM
kABPersonInstantMessageUsernameKey
    <#username#>@me.com
```

URL Property

This constant implements the person URL property (a multivalue property of type `kABMultiStringPropertyType`) and the label used to identify the property's sole value.

```
const ABPropertyID kABPersonURLProperty;
const CFStringRef kABPersonHomePageLabel;
```

Constants

`kABPersonURLProperty`
Identifier for the URL multivalue property.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.

`kABPersonHomePageLabel`

Home page.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

Discussion

The value of these constants is undefined until one of the following functions has been called: [ABAddressBookCreate](#) (page 11), [ABPersonCreate](#) (page 56), [ABGroupCreate](#) (page 45).

Related Name Property

These constants implement the person related name property (a multivalued property of type `kABMultiStringPropertyType`) and the labels used to identify each of the property's values.

```
const ABPropertyID kABPersonRelatedNamesProperty;
const CFStringRef kABPersonMotherLabel;
const CFStringRef kABPersonFatherLabel;
const CFStringRef kABPersonParentLabel;
const CFStringRef kABPersonSisterLabel;
const CFStringRef kABPersonBrotherLabel;
const CFStringRef kABPersonChildLabel;
const CFStringRef kABPersonFriendLabel;
const CFStringRef kABPersonSpouseLabel;
const CFStringRef kABPersonPartnerLabel;
const CFStringRef kABPersonManagerLabel;
const CFStringRef kABPersonAssistantLabel;
```

Constants

`kABPersonRelatedNamesProperty`

Identifier for the related name multivalued property.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonMotherLabel`

Mother.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonFatherLabel`

Father.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonParentLabel`

Parent.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABPersonSisterLabel`

Sister.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

- `kABPersonBrotherLabel`
Brother.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.
- `kABPersonChildLabel`
Child.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.
- `kABPersonFriendLabel`
Friend.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.
- `kABPersonSpouseLabel`
Spouse.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.
- `kABPersonPartnerLabel`
Partner.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.
- `kABPersonManagerLabel`
Manager.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.
- `kABPersonAssistantLabel`
Assistant.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.

Discussion

The value of these constants is undefined until one of the following functions has been called: [ABAddressBookCreate](#) (page 11), [ABPersonCreate](#) (page 56), [ABGroupCreate](#) (page 45).

Generic Property Labels

These constants contain generic property labels.

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

Constants

- `kABWorkLabel`
Work label.
Available in iOS 2.0 and later.
Declared in `ABPerson.h`.

`kABHomeLabel`

Home label.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

`kABOtherLabel`

Other label.

Available in iOS 2.0 and later.

Declared in `ABPerson.h`.

Discussion

The value of these constants is undefined until one of the following functions has been called: [ABAddressBookCreate](#) (page 11), [ABPersonCreate](#) (page 56), [ABGroupCreate](#) (page 45).

ABSource Reference

Framework:	AddressBook/AddressBook.h
Declared in	ABSource.h

Overview

Sources of contact information are represented by instances of the [ABRecordRef](#) (page 33) opaque type, whose record type is `kABSourceType`. Examples of sources include the local address book database, CardDAV servers, and social networking sites.

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

Functions

ABAddressBookCopyArrayOfAllSources

Returns an array of all sources in the address book.

```
CFArrayRef ABAddressBookCopyArrayOfAllSources (
    ABAddressBookRef addressBook
);
```

Parameters

addressBook

The address book whose source records are being returned.

Return Value

Array containing the source records in *addressBook*.

Availability

Available in iOS 4.0 and later.

Declared In

ABSource.h

ABAddressBookCopyDefaultSource

Returns the default source.

```
ABRecordRef ABAddressBookCopyDefaultSource (
    ABAddressBookRef addressBook
);
```

Parameters

addressBook

The address book whose default source is being returned.

Return Value

The default source record.

Discussion

This is the source used when creating new records if no source is specified.

Availability

Available in iOS 4.0 and later.

Declared In

ABSource.h

ABAddressBookGetSourceWithRecordID

Returns the source record with the given record ID.

```
ABRecordRef ABAddressBookGetSourceWithRecordID (
    ABAddressBookRef addressBook,
    ABRecordID sourceID
);
```

Parameters

addressBook

The address book in which to search for *recordID*.

sourceID

The record identifier to seek in *addressBook*.

Return Value

If found, the source record with *recordID* as its identifier, otherwise NULL.

Availability

Available in iOS 4.0 and later.

Declared In

ABSource.h

Data Types

ABSourceType

Indicates a source type. See [“Source Properties”](#) (page 74).


```
typedef int ABSourceType;
```

Availability

Available in iOS 4.0 and later.

Declared In

ABSource.h

Constants

Searchable Sources

Indicates that a source is searchable.

```
#define kABSourceTypeSearchableMask 0x01000000
```

Constants

`kABSourceTypeSearchableMask`
Indicates that a source is searchable.
 Available in iOS 4.0 and later.
 Declared in `ABSource.h`.

Source Types

These constants identify the type of a source.

```
enum {
    kABSourceTypeLocal          = 0x0,
    kABSourceTypeExchange       = 0x1,
    kABSourceTypeExchangeGAL    = kABSourceTypeExchange | kABSourceTypeSearchableMask,
    kABSourceTypeMobileMe       = 0x2,
    kABSourceTypeLDAP           = 0x3 | kABSourceTypeSearchableMask,
    kABSourceTypeCardDAV        = 0x4,
    kABSourceTypeCardDAVSearch  = kABSourceTypeCardDAV | kABSourceTypeSearchableMask,
};
typedef int ABSourceType;
```

Constants

`kABSourceTypeLocal`
Local address book database.
 Available in iOS 4.0 and later.
 Declared in `ABSource.h`.

`kABSourceTypeExchange`
Exchange server.
 Available in iOS 4.0 and later.
 Declared in `ABSource.h`.

`kABSourceTypeExchangeGAL`
Exchange Global Address List.
Available in iOS 4.0 and later.
Declared in `ABSource.h`.

`kABSourceTypeMobileMe`
MobileMe.
Available in iOS 4.0 and later.
Declared in `ABSource.h`.

`kABSourceTypeLDAP`
LDAP server.
Available in iOS 4.0 and later.
Declared in `ABSource.h`.

`kABSourceTypeCardDAV`
CardDAV server.
Available in iOS 4.0 and later.
Declared in `ABSource.h`.

`kABSourceTypeCardDAVSearch`
Searchable CardDAV server.
Available in iOS 4.0 and later.
Declared in `ABSource.h`.

Source Properties

These constants identify source properties.

```
const ABPropertyID kABSourceNameProperty;  
const ABPropertyID kABSourceTypeProperty;
```

Constants

`kABSourceNameProperty`
The name of the source. Type: `kABStringPropertyType`.
Available in iOS 4.0 and later.
Declared in `ABSource.h`.

`kABSourceTypeProperty`
The type of the source. Type: [ABSourceType](#) (page 72)
Available in iOS 4.0 and later.
Declared in `ABSource.h`.

Document Revision History

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

Date	Notes
2010-03-22	Updated for iOS 4.0.
2009-05-15	Removed preliminary document boilerplate.
2008-07-08	New document that describes the Address Book API for accessing contact data.

REVISION HISTORY

Document Revision History