
Address Book UI Framework Reference for iOS

User Experience: Windows & Views



2010-07-07



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

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

[Address Book UI Framework Overview](#) 5

Part I **Classes** 7

Chapter 1 **ABNewPersonViewController Class Reference** 9

[Overview](#) 9
[Tasks](#) 9
[Properties](#) 10

Chapter 2 **ABPeoplePickerNavigationController Class Reference** 13

[Overview](#) 13
[Tasks](#) 13
[Properties](#) 13

Chapter 3 **ABPersonViewController Class Reference** 15

[Overview](#) 15
[Tasks](#) 15
[Properties](#) 16
[Instance Methods](#) 18

Chapter 4 **ABUnknownPersonViewController Class Reference** 19

[Overview](#) 19
[Tasks](#) 19
[Properties](#) 20

Part II **Protocols** 23

Chapter 5 **ABNewPersonViewControllerDelegate Protocol Reference** 25

[Overview](#) 25
[Tasks](#) 25
[Instance Methods](#) 25

Chapter 6 **ABPeoplePickerNavigationControllerDelegate Protocol Reference** 27

Overview 27
Tasks 27
Instance Methods 27

Chapter 7 **ABPersonViewControllerDelegate Protocol Reference** 31

Overview 31
Tasks 31
Instance Methods 31

Chapter 8 **ABUnknownPersonViewControllerDelegate Protocol Reference** 33

Overview 33
Tasks 33
Instance Methods 33

Part III **Functions** 35

Chapter 9 **Address Book UI Functions Reference** 37

Overview 37
Functions 37

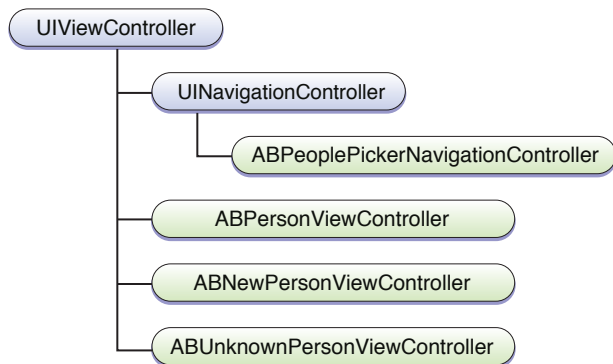
Document Revision History 39

Introduction

Framework	/System/Library/Frameworks/AddressBookUI.framework
Header file directories	/System/Library/Frameworks/AddressBookUI.framework/Headers
Declared in	ABAddressFormatting.h ABNewPersonViewController.h ABPeoplePickerNavigationController.h ABPersonViewController.h ABUnknownPersonViewController.h

The Address Book UI framework provides controllers that facilitate displaying, editing, selecting, and creating records in the Address Book database.

Address Book UI Framework Overview



Address Book UI framework inherits from UIKit framework

Classes

ABNewPersonViewController Class Reference

Inherits from	UIViewController : UIResponder : NSObject
Conforms to	NSCoding (UIViewController) NSObject (NSObject)
Framework	/System/Library/Frameworks/AddressBookUI.framework
Availability	Available in iOS 2.0 and later.
Declared in	ABNewPersonViewController.h

Overview

The `ABNewPersonViewController` class (whose instances are known as **new-person view controllers**) implements the view controller used to create a contact.

New-person view controllers are modal view controllers that manage a set of view controllers used to create a contact (`ABPersonRef`) and edit its properties.

Important: New-person view controllers must be used with a navigation controller in order to function properly. It is recommended that you present a new-person view controller modally.

Tasks

Displaying Person Properties

`displayedPerson` (page 10) *property*

Optional. Specifies the person properties that the new-person view controller pre-fills in its views.

Configuring New Person Views

`addressBook` (page 10) *property*

Optional. The address book to which the new contact is added.

`parentGroup` (page 11) *property*

Optional. Specifies the group to which to add the new contact on save.

[newPersonViewDelegate](#) (page 10) *property*
The delegate of a new-person view controller.

Properties

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

addressBook

Optional. The address book to which the new contact is added.

```
@property(nonatomic, readonly) ABAddressBookRef addressBook
```

Discussion

When unset, an address book is created and assigned to this property when needed.

Availability

Available in iOS 2.0 and later.

Declared In

ABNewPersonViewController.h

displayedPerson

Optional. Specifies the person properties that the new-person view controller pre-fills in its views.

```
@property(nonatomic, readonly) ABRecordRef displayedPerson
```

Discussion

When this property has no person properties, the new-person view controller does not pre-fill properties in its views.

Availability

Available in iOS 2.0 and later.

Declared In

ABNewPersonViewController.h

newPersonViewDelegate

The delegate of a new-person view controller.

```
@property(nonatomic, assign) id<ABNewPersonViewControllerDelegate>  
newPersonViewDelegate
```

Discussion

The delegate must adopt the `ABNewPersonViewControllerDelegate` formal protocol.

Availability

Available in iOS 2.0 and later.

Declared In

ABNewPersonViewController.h

parentGroup

Optional. Specifies the group to which to add the new contact on save.

@property(nonatomic, readonly) ABRecordRef parentGroup

Availability

Available in iOS 2.0 and later.

Declared In

ABNewPersonViewController.h

ABPeoplePickerNavigationController Class Reference

Inherits from	UINavigationController : UIViewController : UIResponder : NSObject
Conforms to	NSCoding (UIViewController) NSObject (NSObject)
Framework	/System/Library/Frameworks/AddressBookUI.framework
Availability	Available in iOS 2.0 and later.
Declared in	ABPeoplePickerNavigationController.h

Overview

The `ABPeoplePickerNavigationController` class (whose instances are known as **people-picker navigation controllers**) implements a view controller that manages a set of views that allow the user to select a contact or one of its contact-information items from an address book.

Tasks

Displaying Person Properties

[displayedProperties](#) (page 14) *property*

Array of `NSNumber` objects representing `ABPropertyID` values identifying properties (such as name or telephone number) the picker displays when it shows a person.

Configuring People Pickers

[addressBook](#) (page 14) *property*

Optional. The address book from which to obtain the list of contacts.

[peoplePickerDelegate](#) (page 14) *property*

The people-picker navigation controller delegate.

Properties

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

addressBook

Optional. The address book from which to obtain the list of contacts.

```
@property(n nonatomic, readwrite) ABAddressBookRef addressBook
```

Discussion

When unset, an address book is created and assigned to this property when needed.

Availability

Available in iOS 2.0 and later.

Declared In

ABPeoplePickerNavigationController.h

displayedProperties

Array of `NSNumber` objects representing `ABPropertyID` values identifying properties (such as name or telephone number) the picker displays when it shows a person.

```
@property(n nonatomic, copy) NSArray *displayedProperties
```

Discussion

To have the picker display a single property for the person displayed, such as telephone number, set `displayedProperties` to an array with a single value, such as `kABPersonPhoneProperty`.

Availability

Available in iOS 2.0 and later.

Declared In

ABPeoplePickerNavigationController.h

peoplePickerDelegate

The people-picker navigation controller delegate.

```
@property(n nonatomic, assign) id<ABPeoplePickerNavigationControllerDelegate>
    peoplePickerDelegate
```

Discussion

The delegate must adopt the `ABPeoplePickerNavigationControllerDelegate` formal protocol.

Availability

Available in iOS 2.0 and later.

Declared In

ABPeoplePickerNavigationController.h

ABPersonViewController Class Reference

Inherits from	UIViewController : UIResponder : NSObject
Conforms to	NSCoding (UIViewController) NSObject (NSObject)
Framework	/System/Library/Frameworks/AddressBookUI.framework
Availability	Available in iOS 2.0 and later.
Declared in	ABPersonViewController.h

Overview

The `ABPersonViewController` class (whose instances are known as **person view controllers**) implements the view used to display a person record (`ABPersonRef`).

Important: Person view controllers must be used with a navigation controller in order to function properly.

Tasks

Displaying Person Properties

[displayedPerson](#) (page 16) *property*

The person displayed by the person view.

[displayedProperties](#) (page 17) *property*

Optional. Identifies the set of properties (such as name or telephone number) of [displayedPerson](#) (page 16) the receiver displays.

[shouldShowLinkedPeople](#) (page 17) *property*

Indicates whether the person view should display data from person records that are linked with the person record being displayed.

Configuring Person Views

[addressBook](#) (page 16) *property*

Optional. The address book from which to obtain the contact to display.

- `allowsEditing` (page 16) *property*
Specifies whether the user can edit the person's information.
- `setHighlightedItemForProperty:withIdentifier:` (page 18)
Specifies whether to highlight a particular property of the displayed person.
- `personViewDelegate` (page 17) *property*
The person-view controller delegate.

Properties

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

addressBook

Optional. The address book from which to obtain the contact to display.

```
@property(nonatomic, readonly) ABAddressBookRef addressBook
```

Discussion

When unset, an address book is created and assigned to this property when needed.

Availability

Available in iOS 2.0 and later.

Declared In

ABPersonViewController.h

allowsEditing

Specifies whether the user can edit the person's information.

```
@property(nonatomic) BOOL allowsEditing
```

Discussion

When editing a person's information, all person properties are visible.

Availability

Available in iOS 2.0 and later.

Declared In

ABPersonViewController.h

displayedPerson

The person displayed by the person view.


```
@property(n nonatomic, readwrite) ABRecordRef displayedPerson
```

Discussion

The receiver displays the properties of this person record that are present in `displayedProperties` (page 17).

Availability

Available in iOS 2.0 and later.

Declared In

ABPersonViewController.h

displayedProperties

Optional. Identifies the set of properties (such as name or telephone number) of `displayedPerson` (page 16) the receiver displays.

```
@property(n nonatomic, copy) NSArray *displayedProperties
```

Discussion

The default value of this property is NULL.

The properties are specified using an array of NSNumber objects representing ABPropertyID values.

To have the receiver display a single property for `displayedPerson` (page 16), such as telephone number, set `displayedProperties` to an array with a single value, such as `kABPersonPhoneProperty`.

Availability

Available in iOS 2.0 and later.

Declared In

ABPersonViewController.h

personViewDelegate

The person-view controller delegate.

```
@property(n nonatomic, assign) id<ABPersonViewControllerDelegate> personViewDelegate
```

Discussion

The delegate must adopt the `ABPersonViewControllerDelegate` protocol.

Availability

Available in iOS 2.0 and later.

Declared In

ABPersonViewController.h

shouldShowLinkedPeople

Indicates whether the person view should display data from person records that are linked with the person record being displayed.

@property(nonatomic) BOOL shouldShowLinkedPeople

Discussion

Linked records represent the same actual person. They often come from different sources, but may also come from the same source.

Availability

Available in iOS 4.0 and later.

Declared In

ABPersonViewController.h

Instance Methods

setHighlightedItemForProperty:withIdentifier:

Specifies whether to highlight a particular property of the displayed person.

```
- (void)setHighlightedItemForProperty:(ABPropertyID)property  
withIdentifier:(ABMultiValueIdentifier)identifier
```

Parameters

property

The property to highlight.

identifier

When *property* is a multivalued property, the value to highlight.

Availability

Available in iOS 2.0 and later.

See Also

[@property displayedPerson](#) (page 16)

Declared In

ABPersonViewController.h

ABUnknownPersonViewController Class Reference

Inherits from	UIViewController : UIResponder : NSObject
Conforms to	NSCoding (UIViewController) NSObject (NSObject)
Framework	/System/Library/Frameworks/AddressBookUI.framework
Availability	Available in iOS 2.0 and later.
Declared in	ABUnknownPersonViewController.h

Overview

The `ABUnknownPersonViewController` class (whose instances are known as **unknown-person view controllers**) implements a view controller used to create a contact from a set of person properties.

Unknown-person view controllers display contact information that can be saved to the Address Book database. From instances of this class, users may also initiate standard actions, such as:

- Phone call
- Text message
- Create contact
- Add to contact

Performing any of the standard actions may result in your application quitting.

Important: Unknown-person view controllers must be used with a navigation controller in order to function properly.

Tasks

Displaying Person Properties

`alternateName` (page 21) *property*

Provides a default value for the First Name and Last Name properties.

[message](#) (page 22) *property*

Text displayed below [alternateName](#) (page 21).

[displayedPerson](#) (page 21) *property*

Specifies a person record whose properties are displayed by the view controller.

Managing Delegates

[addressBook](#) (page 20) *property*

Optional. The address book into which to add the new contact create a contact using the information the receiver displays.

[allowsActions](#) (page 21) *property*

Specifies whether the user can perform property-based actions, such as making calls, sending text messages, and opening URLs.

[allowsAddingToAddressBook](#) (page 21) *property*

Specifies whether the user can add the properties displayed by the unknown-person view controller to the Address Book as a new contact or to an existing contact.

[unknownPersonViewDelegate](#) (page 22) *property*

The unknown-person view controller delegate.

Properties

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

addressBook

Optional. The address book into which to add the new contact create a contact using the information the receiver displays.

```
@property(nonatomic, readonly) ABAddressBookRef addressBook
```

Discussion

When unspecified, this view controller sets the value of this property by creating an `ABAddressBookRef` object.

Availability

Available in iOS 2.0 and later.

See Also

[@property allowsAddingToAddressBook](#) (page 21)

Declared In

`ABUnknownPersonViewController.h`

allowsActions

Specifies whether the user can perform property-based actions, such as making calls, sending text messages, and opening URLs.

```
@property(n nonatomic) BOOL allowsActions
```

Availability

Available in iOS 2.0 and later.

Declared In

ABUnknownPersonViewController.h

allowsAddingToAddressBook

Specifies whether the user can add the properties displayed by the unknown-person view controller to the Address Book as a new contact or to an existing contact.

```
@property(n nonatomic) BOOL allowsAddingToAddressBook
```

Availability

Available in iOS 2.0 and later.

Declared In

ABUnknownPersonViewController.h

alternateName

Provides a default value for the First Name and Last Name properties.

```
@property(n nonatomic, copy) NSString *alternateName
```

Discussion

The receiver uses this value for First Name and Last Name properties when they're empty.

Availability

Available in iOS 2.0 and later.

Declared In

ABUnknownPersonViewController.h

displayedPerson

Specifies a person record whose properties are displayed by the view controller.

```
@property(n nonatomic, readwrite) ABRecordRef displayedPerson
```

Discussion

All the properties of `displayedPerson` are displayed by the view controller.

Availability

Available in iOS 2.0 and later.

Declared In

ABUnknownPersonViewController.h

messageText displayed below [alternateName](#) (page 21).

```
@property(n nonatomic, copy) NSString *message
```

Availability

Available in iOS 2.0 and later.

Declared In

ABUnknownPersonViewController.h

unknownPersonViewDelegate

The unknown-person view controller delegate.

```
@property(n nonatomic, assign) id<ABUnknownPersonViewControllerDelegate>  
    unknownPersonViewDelegate
```

DiscussionThe delegate must adopt the `ABUnknownPersonViewControllerDelegate` protocol.**Availability**

Available in iOS 2.0 and later.

Declared In

ABUnknownPersonViewController.h

Protocols

ABNewPersonViewControllerDelegate Protocol Reference

Conforms to	NSObject
Framework	/System/Library/Frameworks/AddressBookUI.framework
Availability	Available in iOS 2.0 and later.
Declared in	ABNewPersonViewController.h

Overview

The `ABNewPersonViewControllerDelegate` protocol declares the interface that `ABNewPersonViewController` delegates must implement.

Tasks

Responding to User Events

- `newPersonViewController:didCompleteWithNewPerson:` (page 25) *required method*
Sent when the user taps Save or Cancel. If the user tapped Save, the current address book has been saved to the Address Book database. (required)

Instance Methods

newPersonViewController:didCompleteWithNewPerson:

Sent when the user taps Save or Cancel. If the user tapped Save, the current address book has been saved to the Address Book database. (required)

```
- (void)newPersonViewController:(ABNewPersonViewController *)newPersonViewController
  didCompleteWithNewPerson:(ABRecordRef)person
```

Parameters

newPersonViewController

The new-person-view controller that manages the new-person interface.

person

On Save, the newly created (and saved) person record.

On Cancel, NULL.

Discussion

If the user tapped Save, pending changes in the current address book (ABAddressBook) have been saved by the time this message is sent to the receiver.

The receiver must dismiss *newPersonViewController*.

Availability

Available in iOS 2.0 and later.

Declared In

ABNewPersonViewController.h

ABPeoplePickerNavigationControllerDelegate Protocol Reference

Conforms to	NSObject
Framework	/System/Library/Frameworks/AddressBookUI.framework
Availability	Available in iOS 2.0 and later.
Declared in	ABPeoplePickerNavigationController.h

Overview

The `ABPeoplePickerNavigationControllerDelegate` protocol describes the interface `ABPeoplePickerNavigationController` delegates must adopt to respond to people-picker user events.

Tasks

Responding to User Events

- `peoplePickerNavigationController:shouldContinueAfterSelectingPerson:` (page 27) *required method*
Sent when the user selects a contact. (required)
- `peoplePickerNavigationController:shouldContinueAfterSelectingPerson:property:identifier:` (page 28) *required method*
Sent when the user selects one of a person's properties. (required)
- `peoplePickerNavigationControllerDidCancel:` (page 28) *required method*
Sent when the user taps Cancel. (required)

Instance Methods

`peoplePickerNavigationController:shouldContinueAfterSelectingPerson:`

Sent when the user selects a contact. (required)

- `(BOOL)peoplePickerNavigationController:(ABPeoplePickerNavigationController *)peoplePicker shouldContinueAfterSelectingPerson:(ABRecordRef)person`

Parameters*peoplePicker*

The people-picker navigation controller with which the user interacted.

person

The person the user selected.

Return Value

- YES to display the contact and dismiss the picker.
- NO to do nothing.

Availability

Available in iOS 2.0 and later.

Declared In

ABPeoplePickerNavigationController.h

peoplePickerNavigationController:shouldContinueAfterSelectingPerson:property:identifier:

Sent when the user selects one of a person's properties. (required)

```
- (BOOL)peoplePickerNavigationController:(ABPeoplePickerNavigationController *)peoplePicker shouldContinueAfterSelectingPerson:(ABRecordRef)person property:(ABPropertyID)property identifier:(ABMultiValueIdentifier)identifier
```

Parameters*peoplePicker*

The people-picker navigation controller with which the user interacted.

person

The person whose contact information item the user selected.

property

The property the user selected.

*identifier*The identifier for the value the user selected if *property* is a multivalued property; otherwise, `kABMultiValueInvalidIdentifier`.**Return Value**

- YES to perform the action for the property selected and dismiss the picker.
- NO to show the person in the picker.

Availability

Available in iOS 2.0 and later.

Declared In

ABPeoplePickerNavigationController.h

peoplePickerNavigationControllerDidCancel:

Sent when the user taps Cancel. (required)

```
- (void)peoplePickerNavigationControllerDidCancel:(ABPeoplePickerNavigationController *)peoplePicker
```

Parameters

peoplePicker

The people-picker navigation controller with which the user interacted.

Discussion

The receiver is responsible for dismissing *peoplePicker*.

Availability

Available in iOS 2.0 and later.

Declared In

ABPeoplePickerNavigationController.h

ABPersonViewControllerDelegate Protocol Reference

Conforms to	NSObject
Framework	/System/Library/Frameworks/AddressBookUI.framework
Availability	Available in iOS 2.0 and later.
Declared in	ABPersonViewController.h

Overview

The `ABPersonViewControllerDelegate` protocol declares the interface that must be implemented by `ABPersonViewController` delegates.

Tasks

Responding to User Events

- `personViewController:shouldPerformDefaultActionForPerson:property:identifier:` (page 31) *required method*

Sent when the user selects a property value of the person displayed in a person view controller. (required)

Instance Methods

personViewController:shouldPerformDefaultActionForPerson:property:identifier:

Sent when the user selects a property value of the person displayed in a person view controller. (required)

```
- (BOOL)personViewController:(ABPersonViewController *)personViewController
    shouldPerformDefaultActionForPerson:(ABRecordRef)person
    property:(ABPropertyID)property
    identifier:(ABMultiValueIdentifier)identifierForValue
```

Parameters

personViewController
The sender.

person

The person *personViewController* is displaying.

property

The property whose value the user selected.

valueIdentifier

When *property* is a multivalued property, the value the user selected.

Return Value

- YES if *personViewController* should perform its default action. Your application may quit as a result of this action.
- NO: if *personViewController* should do nothing. The delegate may perform custom action processing.

Availability

Available in iOS 2.0 and later.

Declared In

ABPersonViewController.h

ABUnknownPersonViewControllerDelegate Protocol Reference

Conforms to	NSObject
Framework	/System/Library/Frameworks/AddressBookUI.framework
Availability	Available in iOS 2.0 and later.
Declared in	ABUnknownPersonViewController.h

Overview

The `ABUnknownPersonViewControllerDelegate` protocol describes the interface `ABUnknownPersonViewController` delegates must adopt to respond to unknown-person view user events.

Tasks

Responding to User Events

- `unknownPersonViewController:didResolveToPerson:` (page 33) *required method*
Sent when the user finishes creating a contact or adding the displayed person properties to an existing contact. (required)
- `unknownPersonViewController:shouldPerformDefaultActionForPerson:property:identifier:` (page 34)
Sent when the user selects a property value of the person displayed in a person view controller.

Instance Methods

unknownPersonViewController:didResolveToPerson:

Sent when the user finishes creating a contact or adding the displayed person properties to an existing contact. (required)

- (void)unknownPersonViewController:(ABUnknownPersonViewController *)unknownPersonView didResolveToPerson:(ABRecordRef)person

Parameters*unknownPersonView*

The unknown-person view controller that sent the message.

The receiver must dismiss this view controller.

person

The contact the user created or to which they added information. This record is saved in the Address Book database.

NULL when the user cancelled the interaction.

Availability

Available in iOS 2.0 and later.

Declared In

ABUnknownPersonViewController.h

unknownPersonViewController:shouldPerformDefaultActionForPerson:property:identifier:

Sent when the user selects a property value of the person displayed in a person view controller.

```
- (BOOL)unknownPersonViewController:(ABUnknownPersonViewController *)personViewController shouldPerformDefaultActionForPerson:(ABRecordRef)person property:(ABPropertyID)property identifier:(ABMultiValueIdentifier)identifier
```

Parameters*unknownPersonViewController*

The sender.

person

The person *unknownPersonViewController* is displaying.

property

The property whose value the user selected.

identifier

The identifier for the value the user selected if *property* is a multivalue property; otherwise, `kABMultiValueInvalidIdentifier`.

Return Value

- YES if *unknownPersonViewController* should perform its default action. Your application may quit as a result of this action.
- NO: if *unknownPersonViewController* should do nothing. The delegate may perform custom action processing.

Availability

Available in iOS 3.0 and later.

Declared In

ABUnknownPersonViewController.h

Functions

Address Book UI Functions Reference

Framework:	AddressBookUI/AddressBookUI.h
Declared in	ABAddressFormatting.h

Overview

The Address Book UI functions provide functionality for formatting contact information.

Functions

ABCreateStringWithAddressDictionary

Returns a formatted address from an address property.

```
NSString *ABCreateStringWithAddressDictionary (
    NSDictionary *address,
    BOOL addCountryName
);
```

Parameters

address

A dictionary representing the address property to format.

addCountryName

Specifies whether to include the name of the country in the returned formatted address.

When *YES* and *address* doesn't include a country name, the country name is added to the return value. (The country name is generated from the country code entry in *address*; see "Address Property" in *ABPerson Reference*.)

Return Value

The formatted address (may include line endings).

Discussion

The address is formatted based on the address's country code (`kABPersonAddressCountryCodeKey`). In general, the country code should be set to correspond with the country name (`kABPersonAddressCountryKey`).

Availability

Available in iOS 2.0 and later.

Declared In

ABAddressFormatting.h

Document Revision History

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

Date	Notes
2010-07-07	Changed the title from "Address Book UI Framework Reference for iPhone OS."
2008-11-19	Added class hierarchy diagram.
2008-06-06	New document that describes the API for displaying contact data.

REVISION HISTORY

Document Revision History