Class

ABAddressBook

The main object you use to access the Address Book database.

macOS

```
class ABAddressBook
```

Overview

The `ABAddressBook` class provides a programming interface to the Address Book—a centralized database used by multiple applications to store contact and other personal information about people. The Address Book database also supports the notion of a “group” containing one or more persons. People may belong to multiple groups, and groups may also belong to other groups with some restrictions (for example, no circular references are allowed).

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

Topics

Creating and Initializing an Address Book

```
class func shared() -> ABAddressBook!
```

Returns the unique shared instance of `ABAddressBook`, or `nil` if the Address Book database can’t be initialized.

Retrieving Groups and People

```
func groups() -> [Any]!
```

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

```
func people() -> [Any]!
```

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

Setting and Retrieving the Logged-in User's Record

```
func me() -> ABPerson!
```

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

```
func setMe(ABPerson!)
```

Sets the record that represents the logged-in user.

Retrieving a Specific Record

```
func record(forUniqueId: String!) -> ABRecord!
```

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

Retrieving the Class of a Record

```
func recordClass(fromUniqueId: String!) -> String!
```

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

Retrieving a Formatted Address

```
func formattedAddress(from: [AnyHashable : Any]!) -> NSAttributedString
!
```

Returns an attributed string containing the formatted address.

Retrieving Default Values

```
func defaultCountryCode() -> String!
```

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

```
func defaultNameOrdering() -> Int
```

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

Adding and Removing Records

`func add(ABRecord!, error: ()) throws`

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

`func add(ABRecord!) -> Bool`

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

`func remove(ABRecord!, error: ()) throws`

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

`func remove(ABRecord!) -> Bool`

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

Searching

`func records(matching: ABSearchElement!) -> [Any]!`

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

Saving and Detecting Changes

`func hasUnsavedChanges() -> Bool`

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

`func save() -> Bool`

Saves all the changes made since the last save.

`func saveAndReturnError() throws`

Saves all the changes made since the last save.

Constants

Database change notification keys

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

Errors

Errors codes returned by the Address Book Framework.

Notifications

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

`static let abDatabaseChanged: NSNotification.Name`

Posted when this process has changed the Address Book database.

`static let abDatabaseChangedExternally: NSNotification.Name`

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

Relationships

Inherits From

`NSObject`

Conforms To

`CVarArg`
`CustomDebugStringConvertible`
`CustomStringConvertible`
`Equatable`
`Hashable`
`NSObjectProtocol`