



Address Book for iPhone

The people's framework

Alexandre Aybes
iPhone Software Engineer



Inbox

Get Mail Delete Junk Reply Reply All Forward New Message Note To Do

Search


| Subject | From | Date Received | Mailbox | Size |
|--|------------------------|---------------------|----------------------|---------------|
| Fwd: Full conference schedule now available | Evan Doll | Yesterday | 4:32 PM Inbox | 15.0 KB |
| Fwd: Oscar 25 mai 2010 | Dominique Gille | May 30, 2010 | 4:00 PM Inbox | 88.0 KB |
| Testing | Alexandre Aybes | May 26, 2010 | 4:23 PM Inbox | 351 KB |
| Wwdc: | Alexandre Aybes | May 24, 2010 | 5:18 PM Notes | 0.9 KB |

Subject: **Fwd: Full conference schedule now available**
 From: Evan Doll
 Date: June 2, 2010 4:32:02 PM PDT
 To: Alex Aybes

You're so dreamy, Alex!

Begin forwarded message:

From: Apple Developer <noreply@adc.apple.com>
Date: June 2, 2010 4:14:03 PM PDT
To: [evan](#)
Subject: Full conference schedule now available



Full conference schedule now available

[Log in](#) to the WWDC site and view the complete conference schedule of technical sessions, hands-on labs, lunchtime presentations, and special events for the Apple Worldwide Developers Conference 2010.

When you're logged in to the WWDC site, you can also:



Address Book for iPhone

The people's framework

Alexandre Aybes
iPhone Software Engineer



What We Will Cover Today

- Overview of Address Book
 - Integrating people into your application
 - Adapting your application for the iPad
 - Improving the user experience: responsiveness and multitasking
-
- No extensive API listings

What You Need to Do:

RTFD

Read the fantastic documentation

Overview

What's Address Book?

Address Book

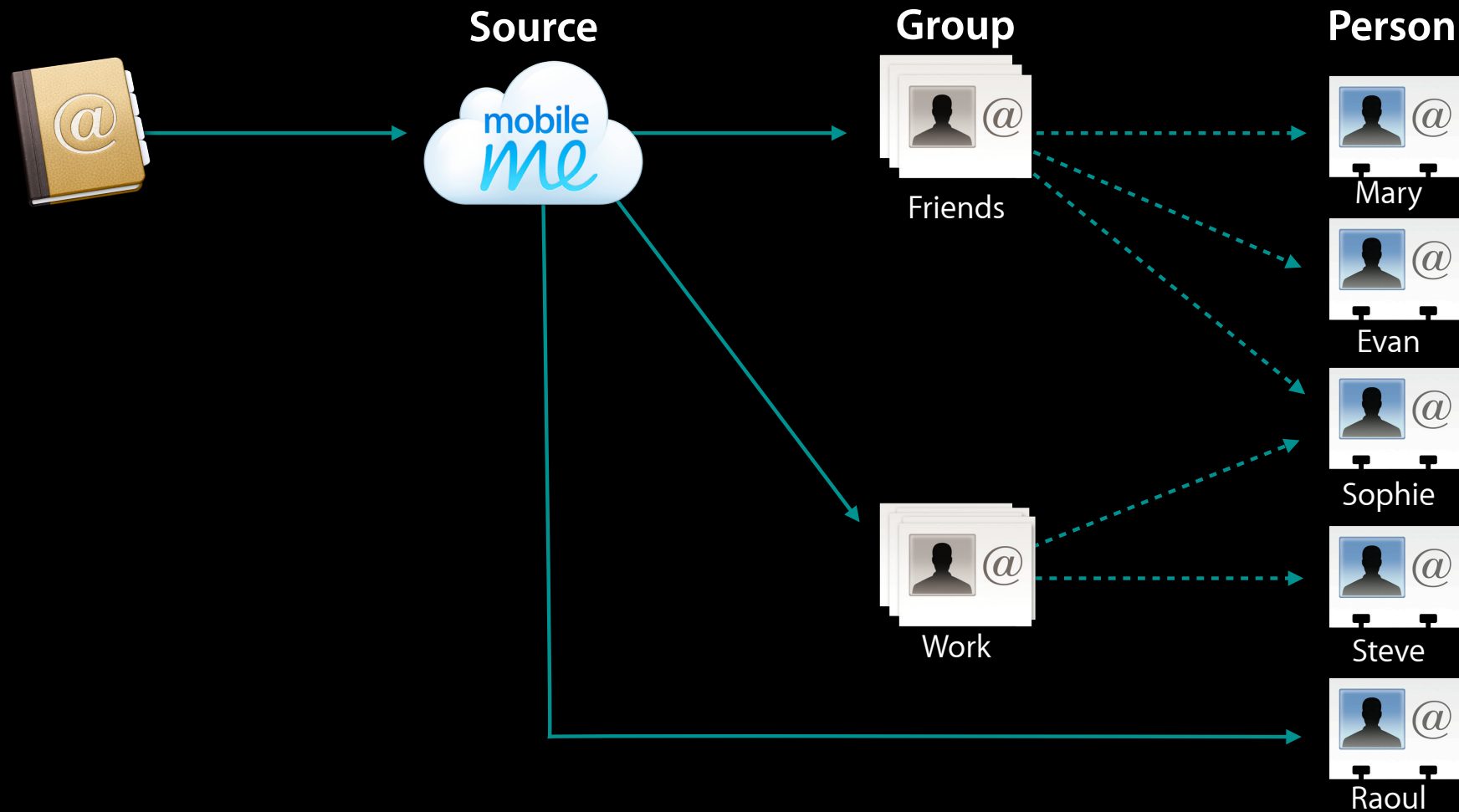
Address Book UI Framework



Address Book Framework



Address Book Framework



Address Book Framework



Source



Group

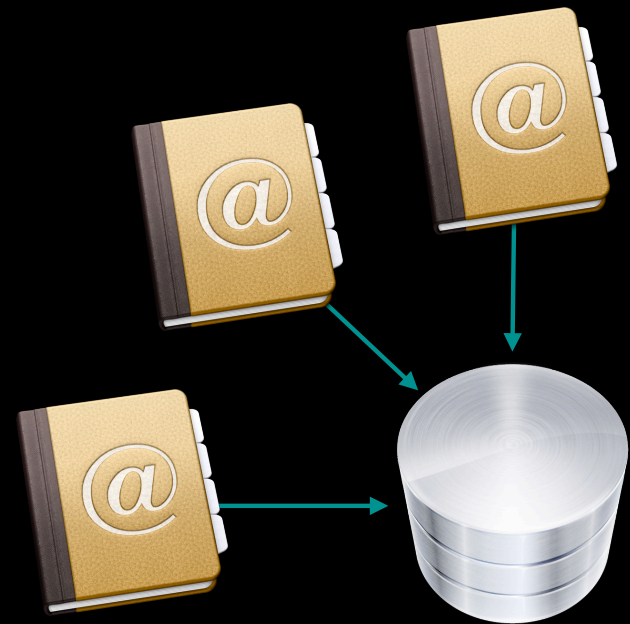


Person



Address Book

- Central point for all things Address Book
- Your connection to the database
- Multiple instances, a single database



Address Book Framework



Source



Group



Person



Source

- Identifies where contacts come from
 - On My iPhone (Local)
 - MobileMe
 - Microsoft Exchange
 - CardDAV
- One source to many contacts relationship



Address Book Framework



Source



Group

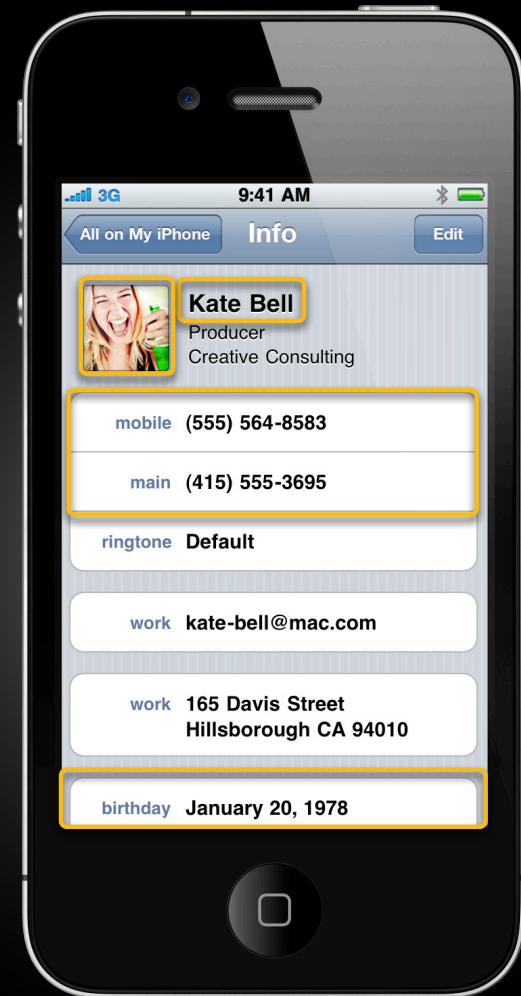


Person

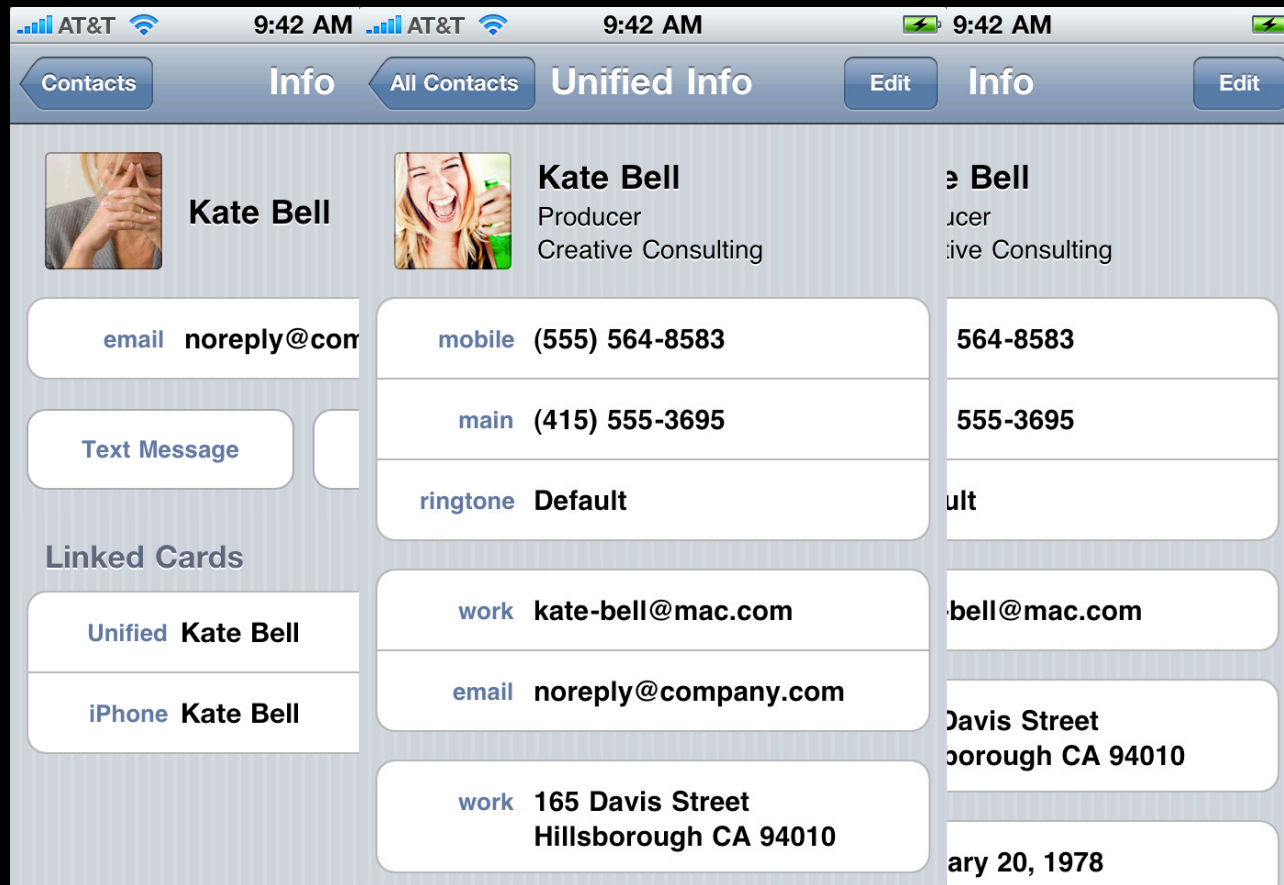


Person

- Represents an individual or organization
- Properties
 - Single value
 - Multiple values



Linking



Corporate

Unified

Personal

Linking

iOS 4

- Multiple records for the same individual
 - Corporate source (MS Exchange)
 - Personal source (MobileMe)
- Automatically linked when added
- You can access the linked people

```
CFArrayRef people =  
    ABPersonCopyArrayOfAllLinkedPeople(person);
```



Address Book Framework



Source



Group

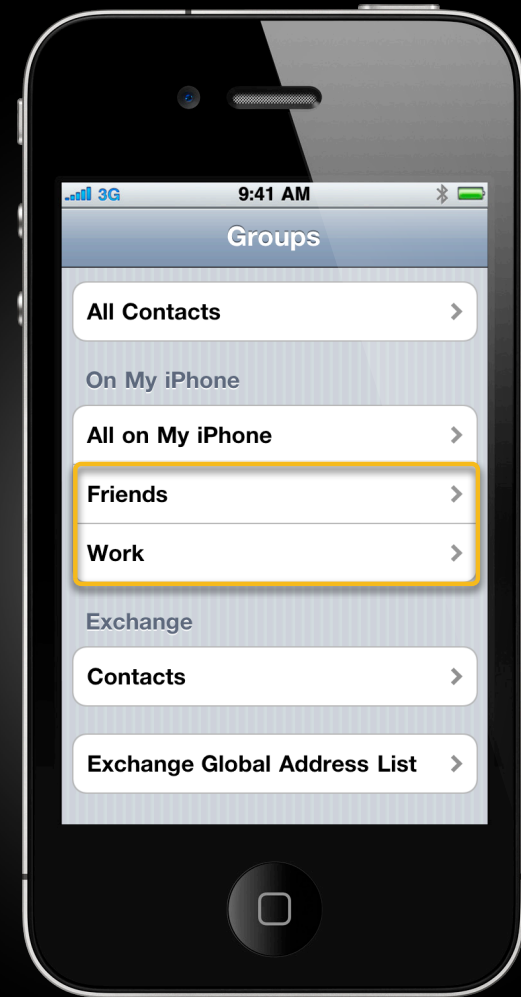


Person

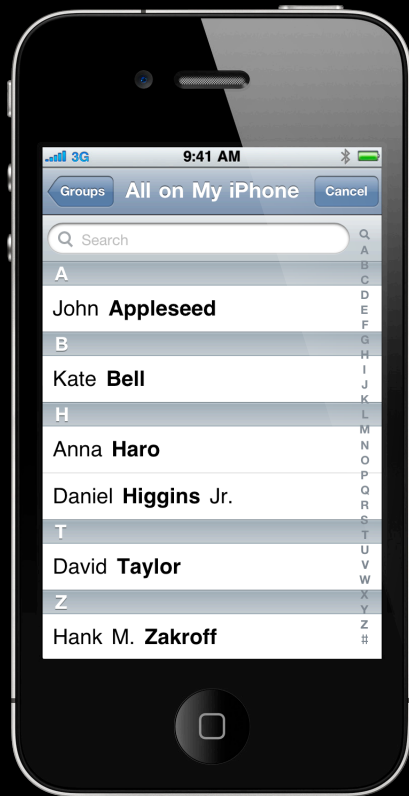


Group

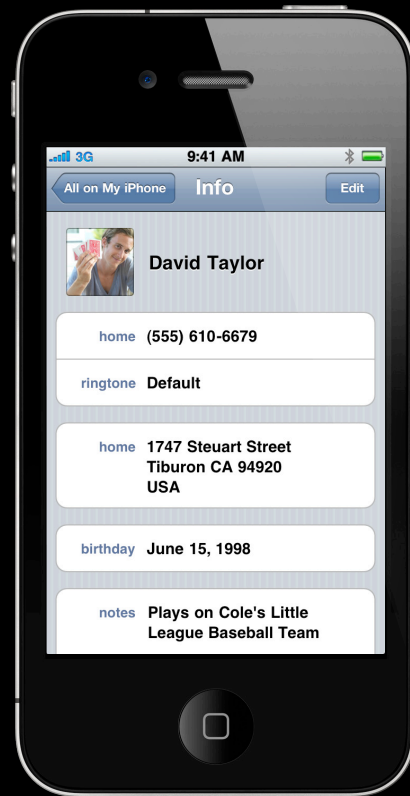
- Many-to-many relationship to person
- No ownership relation
- Not supported by all sources



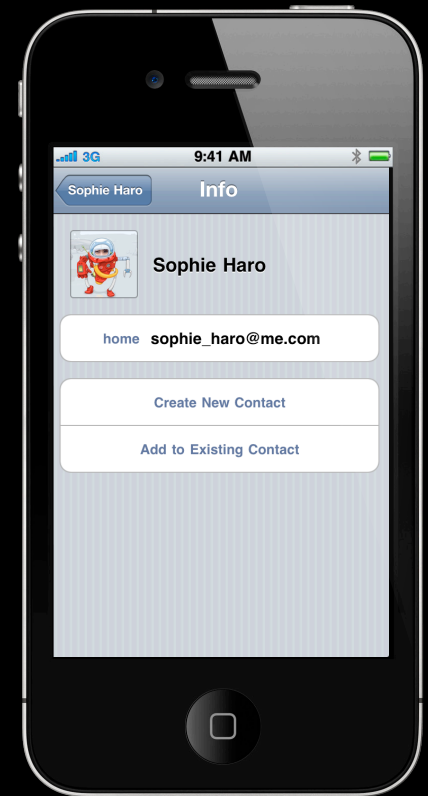
Address Book UI Framework



People Picker
Navigation Controller



Person View Controller



Unknown Person
View Controller

Let's Put It All Together

- Photo cutouts
- Pick a contact
- Get the contact's photo
- Present the contact's information



Photo Cutouts

A fair in your iPhone!

Clare Kasemset

iPhone Software Engineer

What You Just Saw

- Ready to use view controllers
 - ABPeoplePickerNavigationController
 - ABPersonViewController
- Linked people

```
NSArrayRef people = ABPersonCopyArrayOfAllLinkedPeople(person);
```

The Next Step

- Saving the user's creations
- Sharing the user's creations

Saving and Sharing

The more the merrier

Eric Gillum

iPhone Software Engineer

What You Just Saw

- Save record IDs
- Save extra info just in case
- Record IDs do not carry over to other devices
- Match using the name

```
CFArrayRef people = ABAddressBookCopyPeopleWithName(addressBook, name);
```

- Fine tune the matching using other fields
- ABUnknownPersonViewController for contacts not in Address Book

Bigger Is Better

Photo cutouts on the iPad



Popovers



Very Little Work

- Very few Address Book-related changes
- Use popovers when necessary
- Let the frameworks do the rest

Improving The User Experience

- Making your app more responsive
- Responding to background changes

Responsiveness

- Matching on a background thread
 - Use a different instance of Address Book
 - Send ABRecordID back to the main thread

```
+ (void)detachNewThreadSelector:@selector(backgroundMatching)...
```

```
- (void)backgroundMatching {  
    ABAddressBookRef ab = ABAddressBookCreate();  
    // match people  
    NSArray *people = [self matchPeopleWithAddressBook:ab];  
    // extract recordIDs  
    NSArray *recordIDs = [self recordIDsOfPeople:people];  
    [self performSelectorOnMainThread:@selector(found:) withObject:recordIDs];  
    CFRelease(ab);  
}
```

```
person = ABAddressBookGetPersonWithRecordID(mainAddressBook, recordID);
```

Multitasking

- Register to hear about external changes

```
ABAddressBookRegisterExternalChangeCallback(ab, myFunctionPointer, context);
```

- Revert to get the latest changes

```
ABAddressBookRevert(ab);
```

- Redo people matching
- Update the UI
- Save your changes

```
ABAddressBookSave(ab, &error);
```

Summary

- Take advantage of the new linking features
- Multitasking means more background changes
- With just a little code you get a lot of functionality

More Information

Mark Malone

Integration Technologies Evangelist
mgm@apple.com

Documentation

Address Book Documentation
<http://developer.apple.com/iphone>

Apple Developer Forums

<http://devforums.apple.com>

Related Sessions

Adopting Multitasking on iPhone OS, Part 1 (Repeat)

Marina
Friday 9:00AM

Adopting Multitasking on iPhone OS, Part 2 (Repeat)

Marina
Friday 10:15AM

Labs

Address Book and Contacts Lab

Application Frameworks Lab A
Wednesday 4:30PM – 6:30PM



