



Designing Apps with Interface Builder

Rob Marini

Interface Builder Engineer

What a Year It's Been...



iOS 4

Roadmap

Building Universal Applications

- Universal or independent apps
- Organizing your project

- Adopting iPad features
- Adopting iOS 4 features

Roadmap

Building Universal Applications

- Universal or independent apps
- Organizing your project

- Adopting iPad features
- Adopting iOS 4 features

Universal Apps

One app, everywhere

- Build upon existing customer base
- Exposure to a larger audience
- Single App Store review



Independent Apps

One source base, two apps

- Highly platform specific
- Per-device pricing
- Reduced weight for resource intensive apps



“Universal Apps are far more appealing to users than apps that target just one platform”

Vishal Kapur, iTeleport

<http://blog.iteleportmobile.com/the-case-for-universal-apps>

Roadmap

Building Universal Applications

- Universal or independent apps
- Organizing your project

- Adopting iPad features
- Adopting iOS 4 features

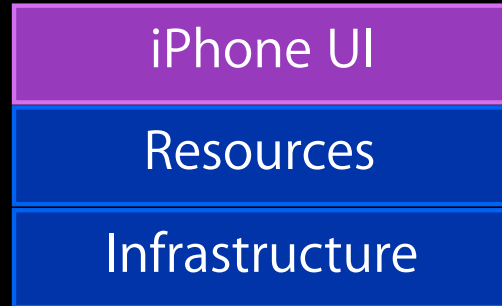
Independent Application Architecture

iPhone UI	iPad UI
Resources	Resources
Infrastructure	Infrastructure

Independent Application Architecture

iPhone UI	iPad UI
Resources	Resources
Infrastructure	Infrastructure

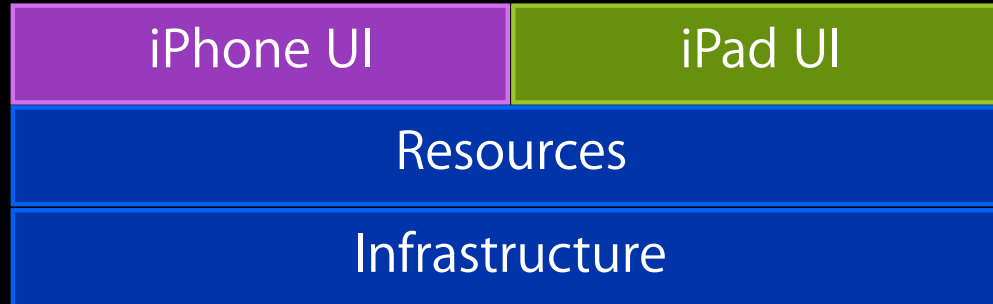
Independent Application Architecture



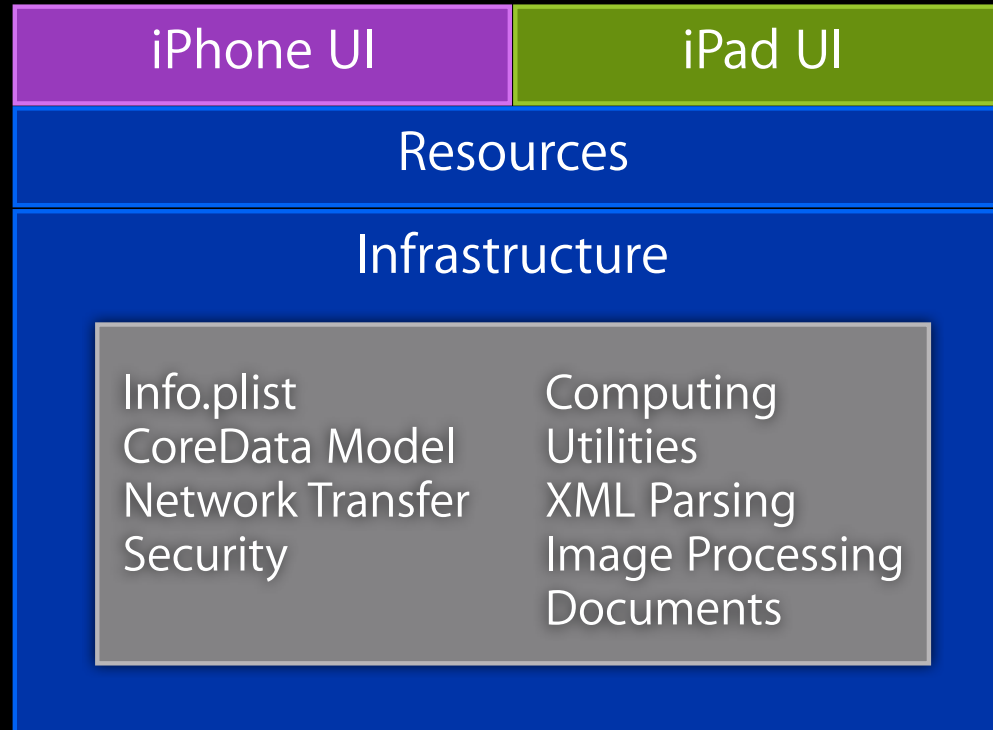
Independent Application Architecture

iPhone UI	iPad UI
Resources	Resources
Infrastructure	Infrastructure

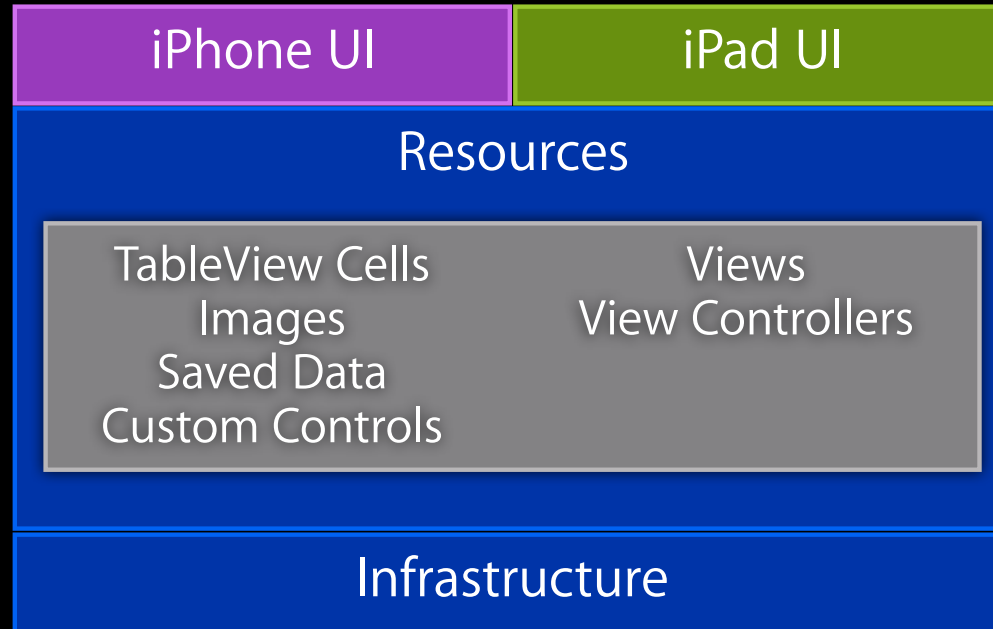
Architecture of a Universal App



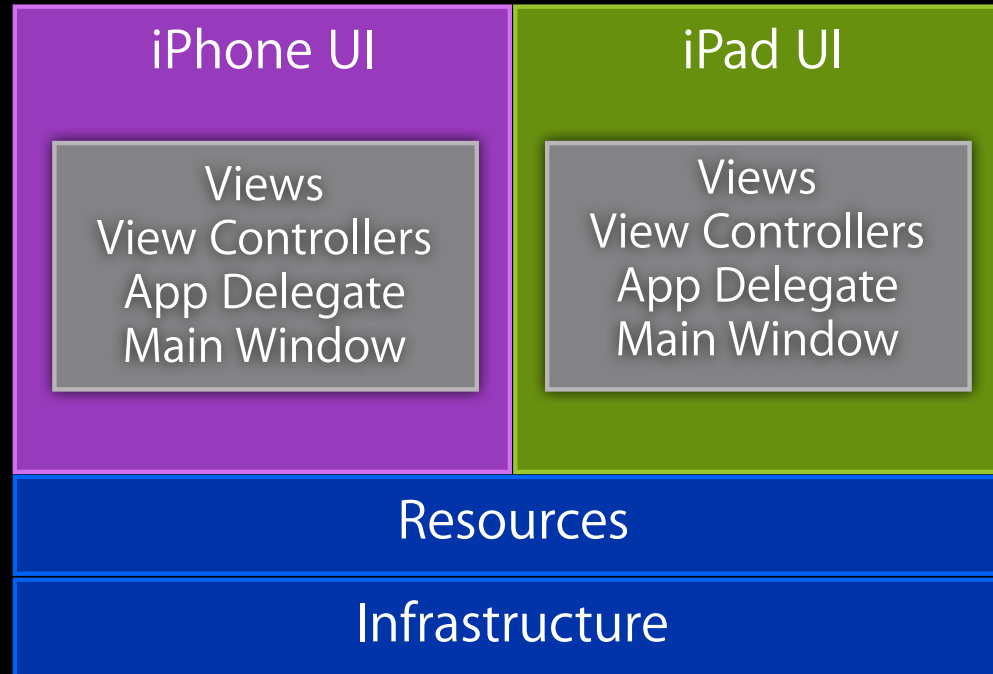
Architecture of a Universal App



Architecture of a Universal App



Architecture of a Universal App



When to Use Per-Platform Classes

Organizing your project

- Serve a specific purpose for each platform
- Highly customized for particular device capabilities
- Ensure it only executes on target platform

When to Use Shared Classes

Organizing your project

- Very little or no platform dependent code
- Minor behavior differences keyed off runtime checks

Roadmap

Building Universal Applications

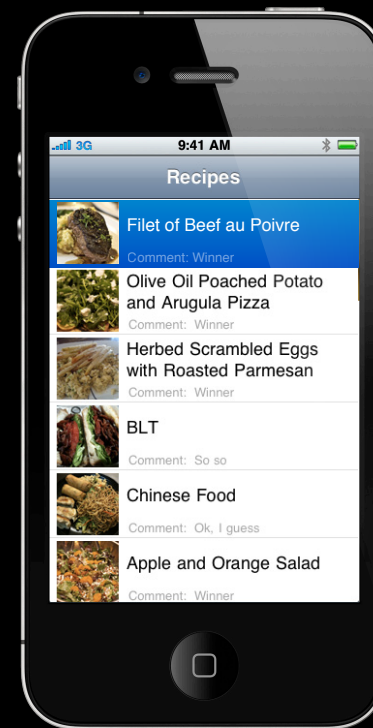
- Universal or independent apps
- Organizing your project

- Adopting iPad features
- Adopting iOS 4 features

The Recipes Application

iPhone version

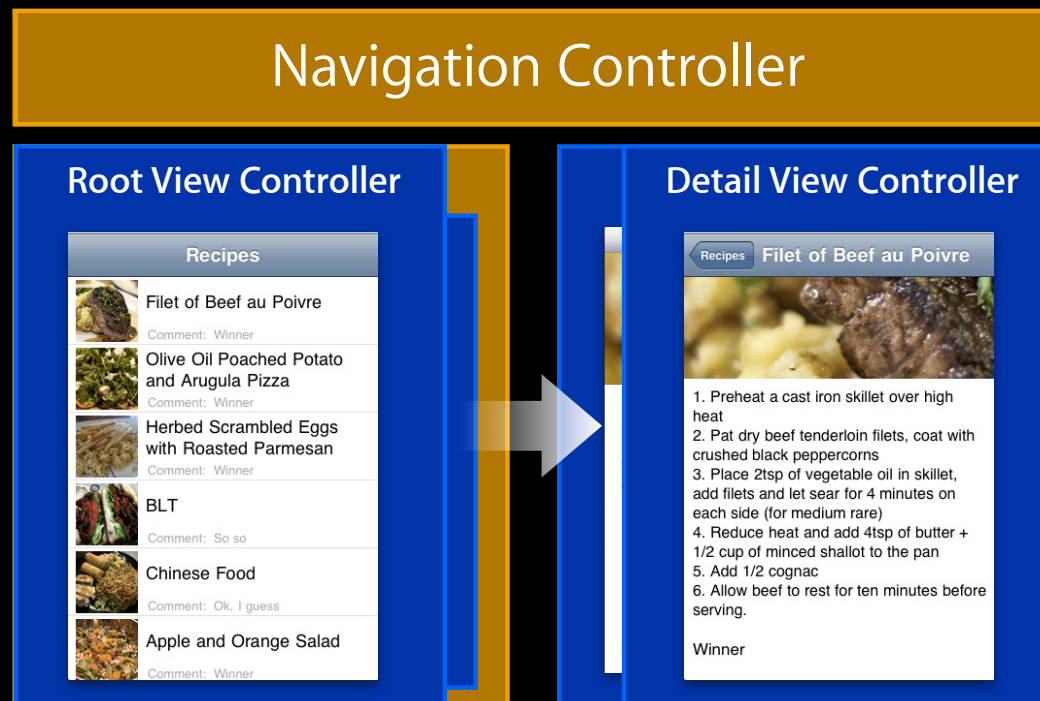
- Navigation based application



The Recipes Application

iPad version

- Split view wrapped application



The Recipes Application

iPad version

- Split view application



The Recipes Application

Breakdown

Shared

- RecipeObject
- RecipeTableViewCell
- RecipeTableViewController
- Navigation Controller

Per-Platform

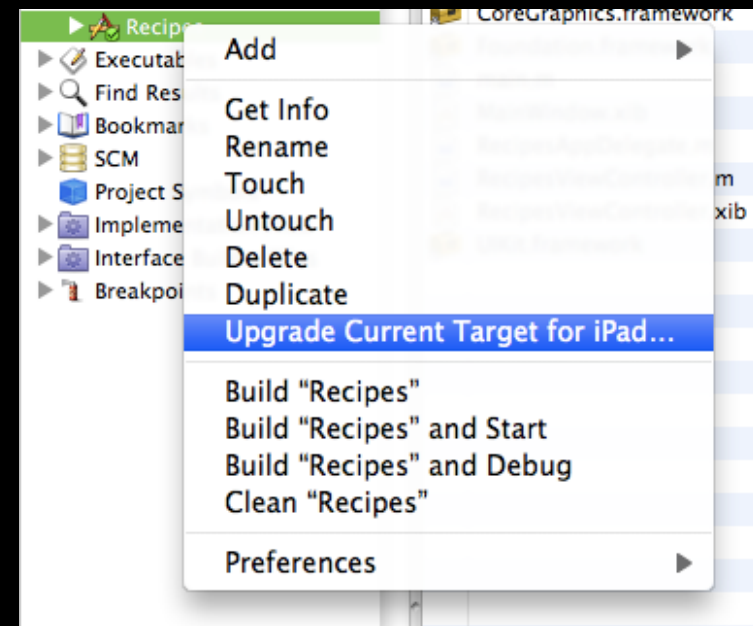
- MainWindow
- RecipeAppDelegate
- RecipeDetailViewController

iPad Only

- Split View Controller

Adopting iPad Features

Upgrading a target



Adopting iPad Features

Upgrade a target

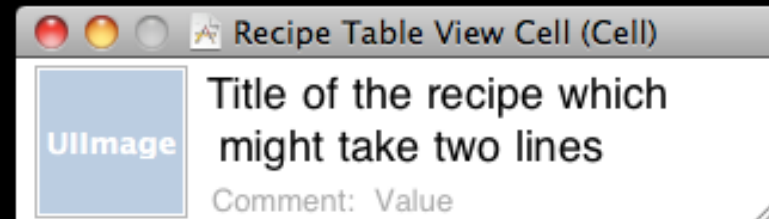
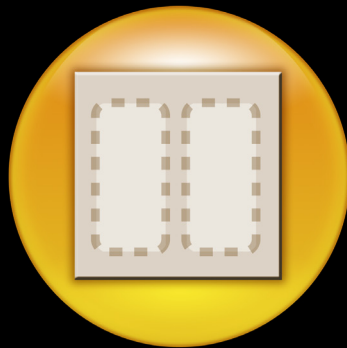
- Upgrade XIBs one at a time in Interface Builder
- Batch process using command line



Adopting iPad Features

Rethinking the user interface

- Adopt new UI paradigms
- Consider what views can be shared between platforms



Demo

Making an application universal

Recap

What we just showed you

- Upgrading an iPhone target to be universal
- Adopting new technologies tailored for iPad
- Adding iPad-specific features while continuing to support iPhone

Adopting iOS 4 Features

Barry Langdon-Lassagne
Interface Builder Engineer

Roadmap

Building Universal Applications

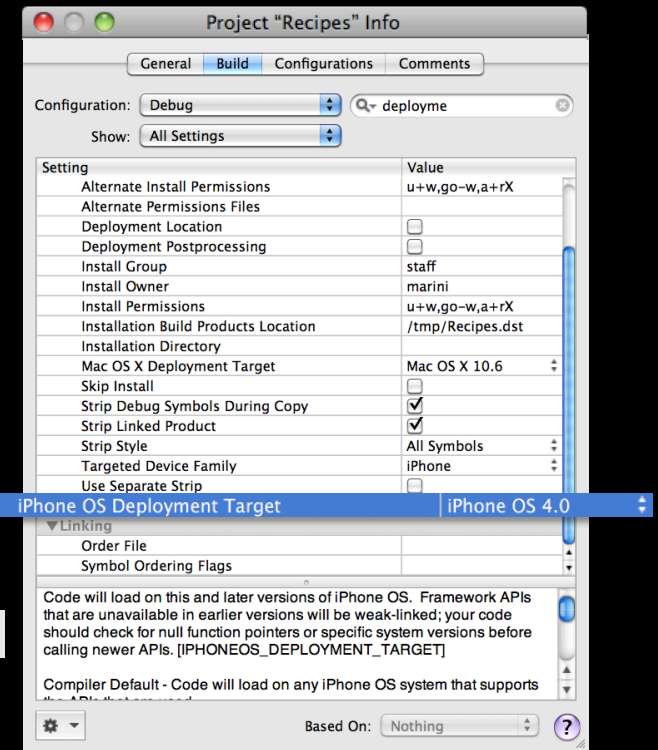
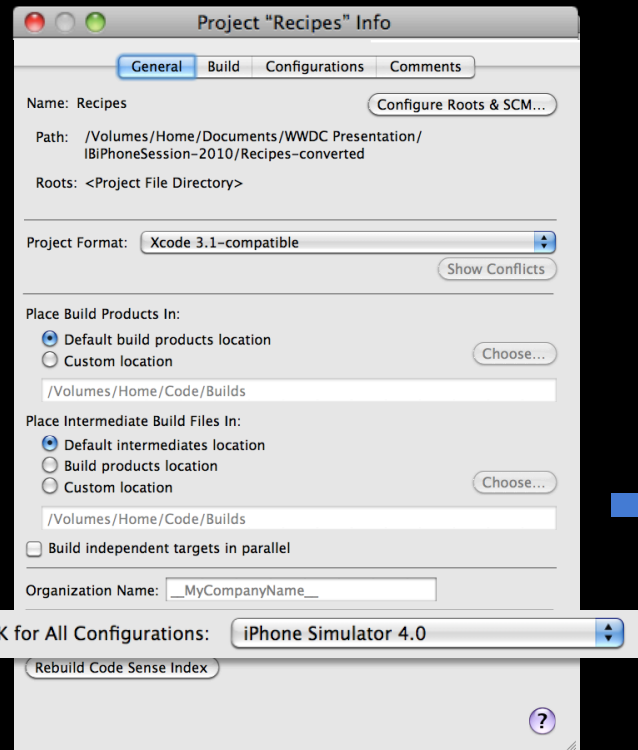
- Universal or independent apps
- Organizing your project

- Adopting iPad features
- Adopting iOS 4 features

Maintaining Compatibility

Deployment target and base SDK

- Range of versions



Device and Version Runtime Checks

Maintaining compatibility

- `-[UIDevice userInterfaceIdiom]`

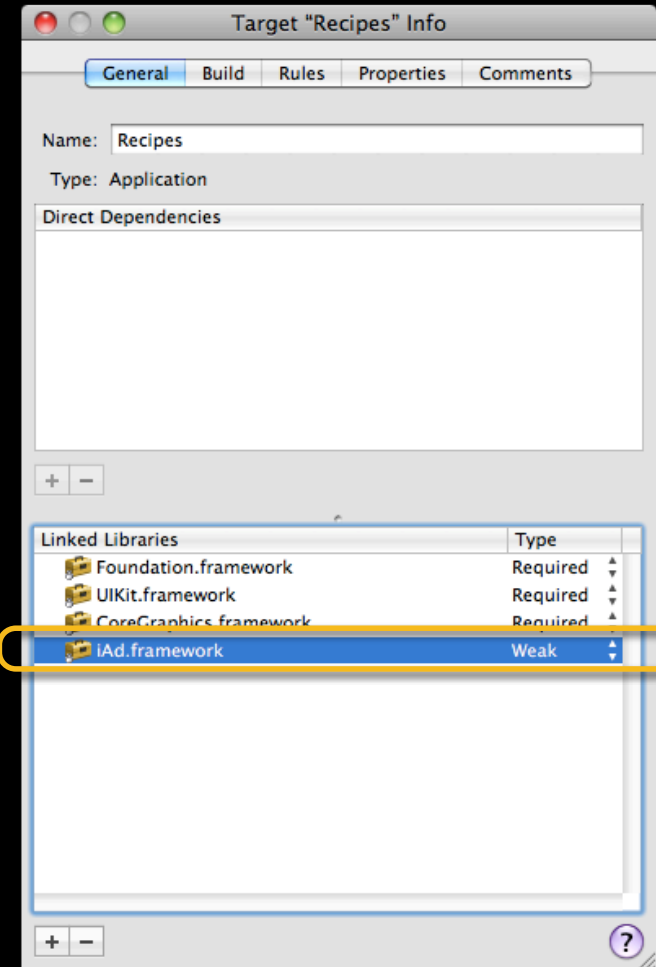
- `NSClassFromString()`

- `-[NSObject respondsToSelector:]`

- `-[UIDevice systemVersion]`

Maintaining Compatibility

Weak linking frameworks



iOS 4 and Interface Builder

The logo for iOS 4, featuring the text "iOS 4" in a white, sans-serif font inside a dark, rounded rectangular button with a subtle gradient and shadow.

What we'll be doing to our Universal App

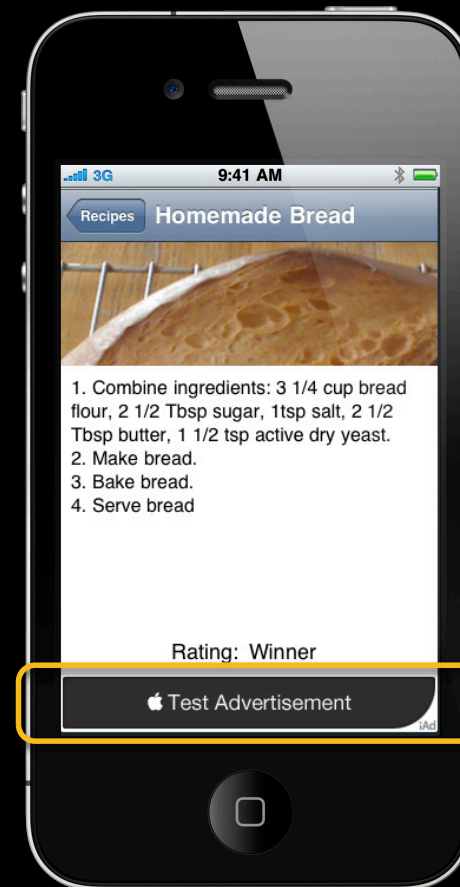
- Adding iAd support
- Migrating to UINib
- Leveraging IBOutletCollection
- UI automation testing

Adding iAd Support

Integrating new UI Elements

iOS 4

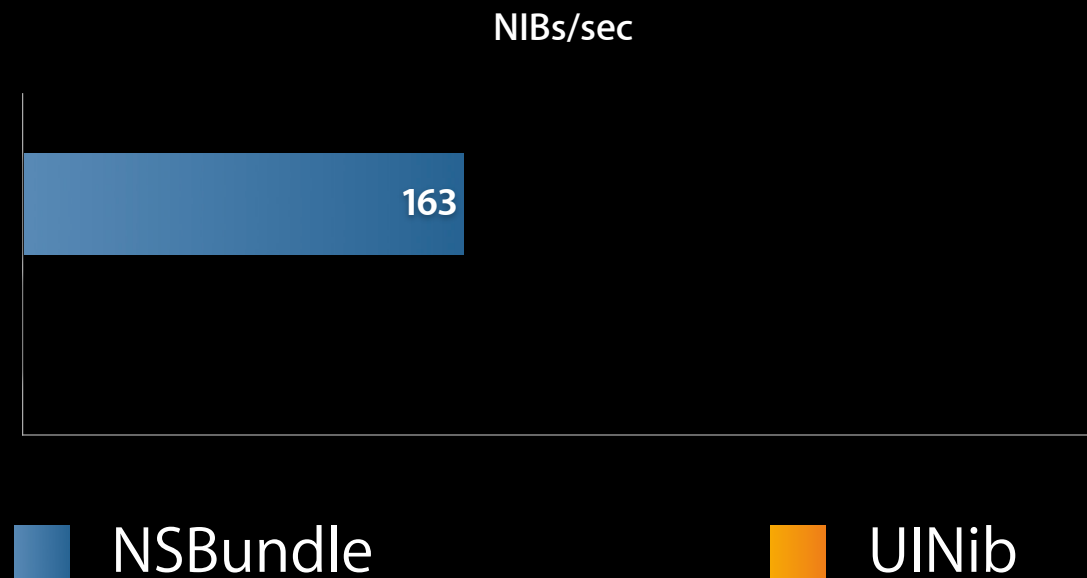
- Drag and drop in Interface Builder
- Weak link the framework
- Add code to handle rotation



Migrating to UINib

Performance enhancement

- High-performance loading of frequently reused NIB data
 - Load twice as many NIBs in the same time

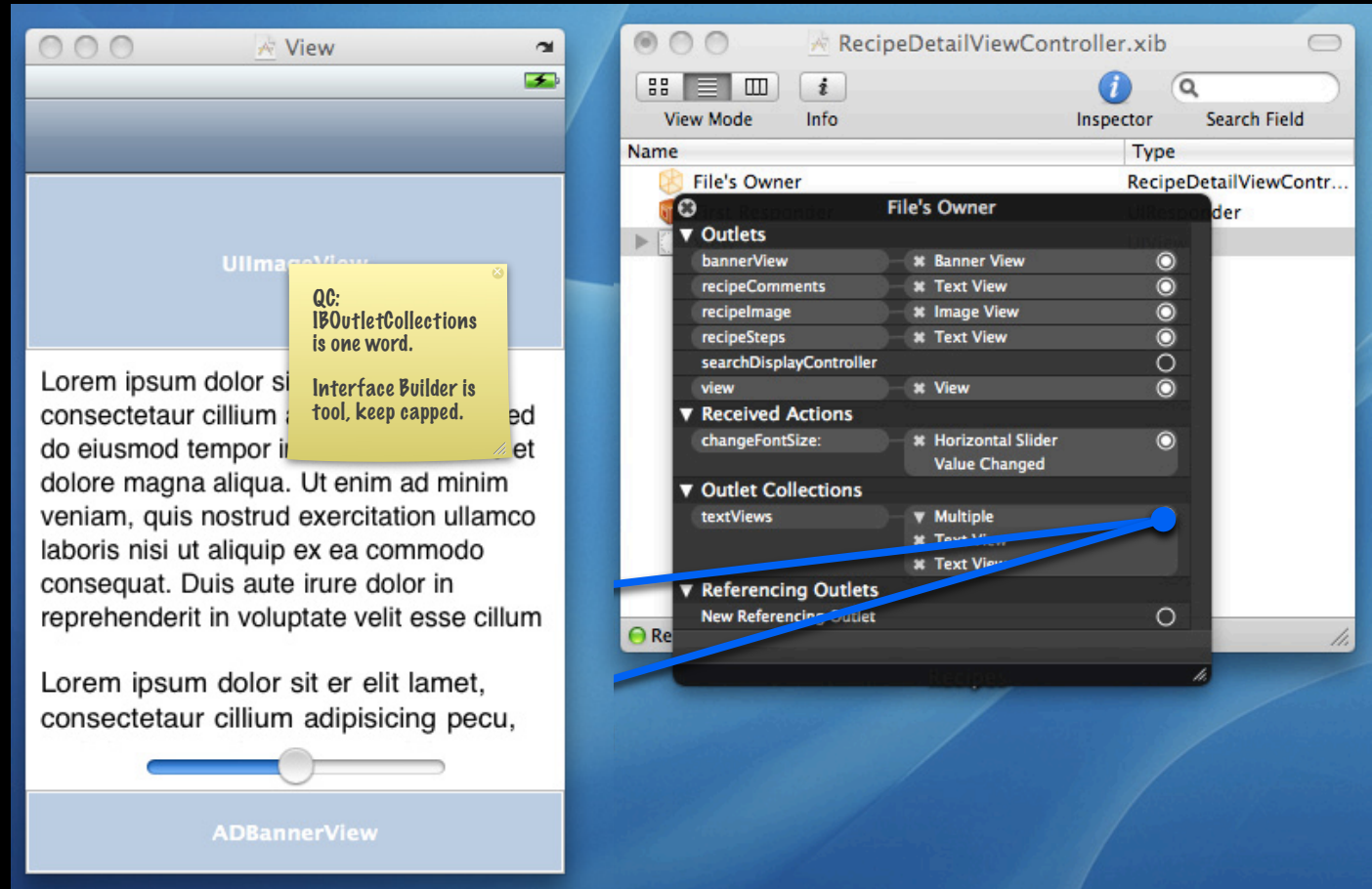


Leveraging IBOutletCollections

iOS 4

Integrating new features

- New type of Interface Builder outlet
- Refer to multiple UI elements simultaneously



Leveraging IBOutletCollections

Integrating new features

- Outlets

```
IBOutlet UILabel *label;
```

Leveraging IBOutletCollections

Integrating new features

- Outlet Collections

```
IBOutletCollection(UILabel) NSArray *labels;
```


Leveraging IBOutletCollections

Integrating new features

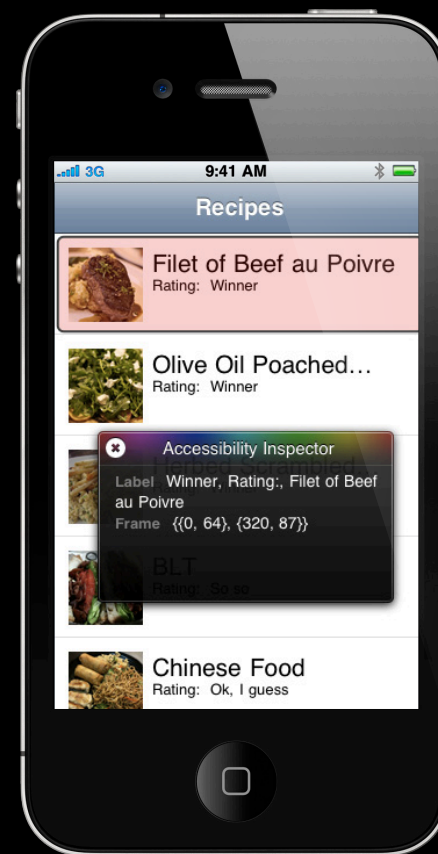
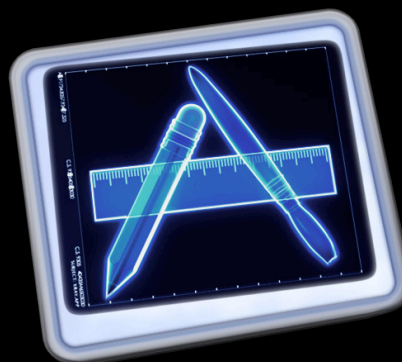
- Outlet Collections

```
IBOutletCollection(id) NSArray *objects;
```

UI Automation Testing

iOS 4

- Run from Instruments
- Based on JavaScript
- Uses accessibility traits set in Interface Builder



Demo

Adding iOS 4 features to our Universal App

Recap

What we just showed you

- iAd
- UINavigationController
- Outlet collections
- Accessibility and UI automation testing

Key Points from Today's Session

- Update your app to be Universal
- Integrate new features
- Maintain compatibility

Related Sessions

Automating User Interface Testing with Instruments

Marina
Wednesday 2:00PM

Using Interface Builder in Xcode 4

Pacific Heights
Thursday 4:30PM

Integrating Ads with iAd (Repeat)

Pacific Heights
Friday 9:00AM

Labs

Automated User Interface Testing Lab

Developer Tools Lab A
Thursday 9:00AM

Xcode for iPhone Development Lab

Developer Tools Lab B
Tuesday 3:15PM

More Information

Michael Jurewitz

Developer Tools Evangelist

jurewitz@apple.com

Documentation

iOS 4 Release Notes

<http://developer.apple.com/iphone>

Apple Developer Forums

<http://devforums.apple.com>



The last slide
after the logo is
intentionally
left blank for
all

