What’s New in Cocoa

Session 204

Ali Ozer
Director of Cocoa Frameworks

Agenda

High-level coverage of changes in Cocoa in Yosemite

Pointers to related sessions

Topics

New Look

Extensions

Handoff

Storyboards and View Controllers

API Modernization

Swift

And others…

Topics

New Look

Extensions

Handoff

Storyboards and View Controllers

API Modernization

Swift

And others…

New Look

New Look

Updated look for controls

Translucency

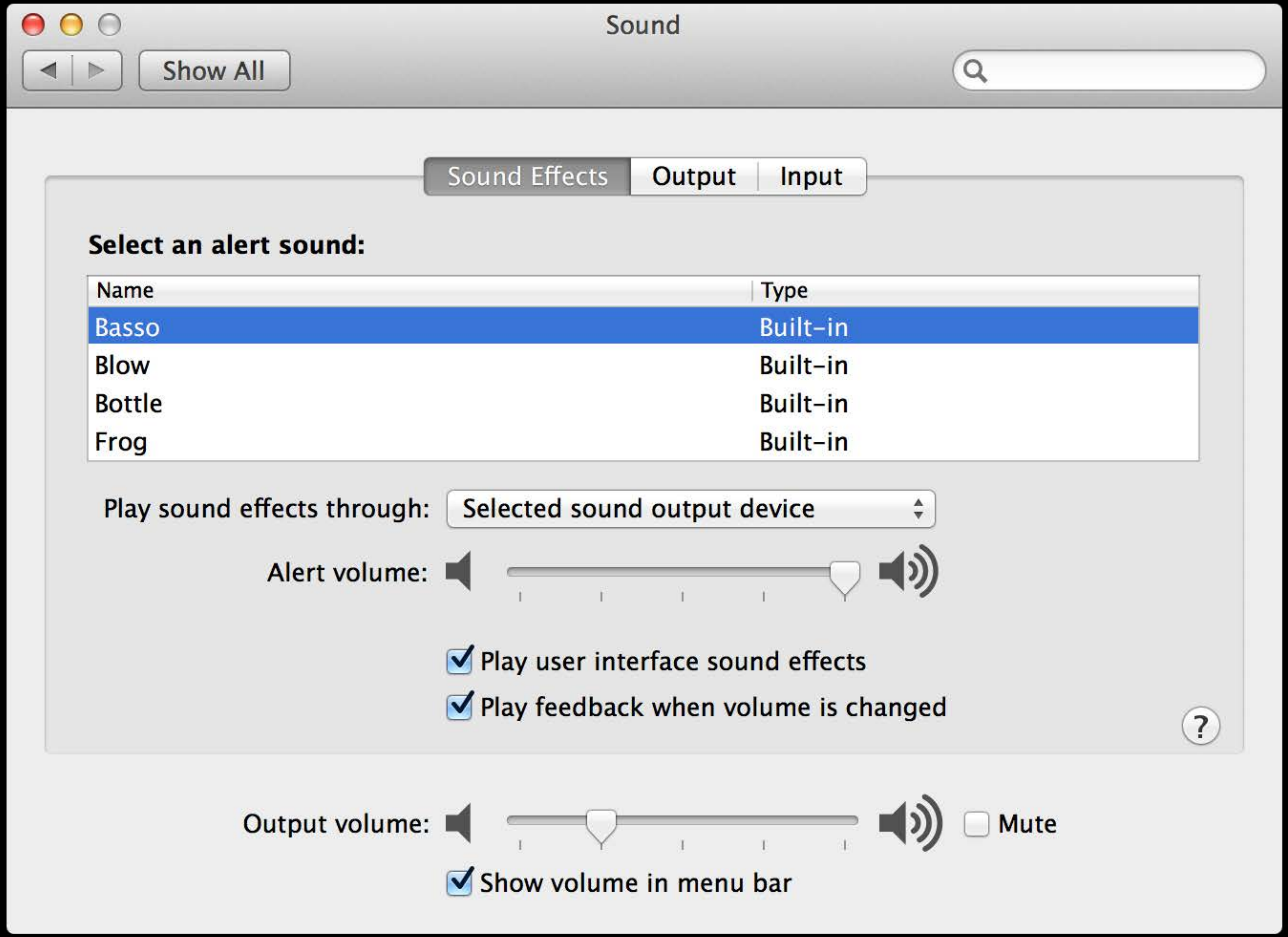
Vibrancy

New window styles

New font

Updated Look for Controls

Updated Look for Controls



Select an alert sound:

Name	Type
Basso	Built-in
Blow	Built-in
Bottle	Built-in
Frog	Built-in

Play sound effects through: Selected sound output device

Alert volume:

✓ Play user interface sound effects

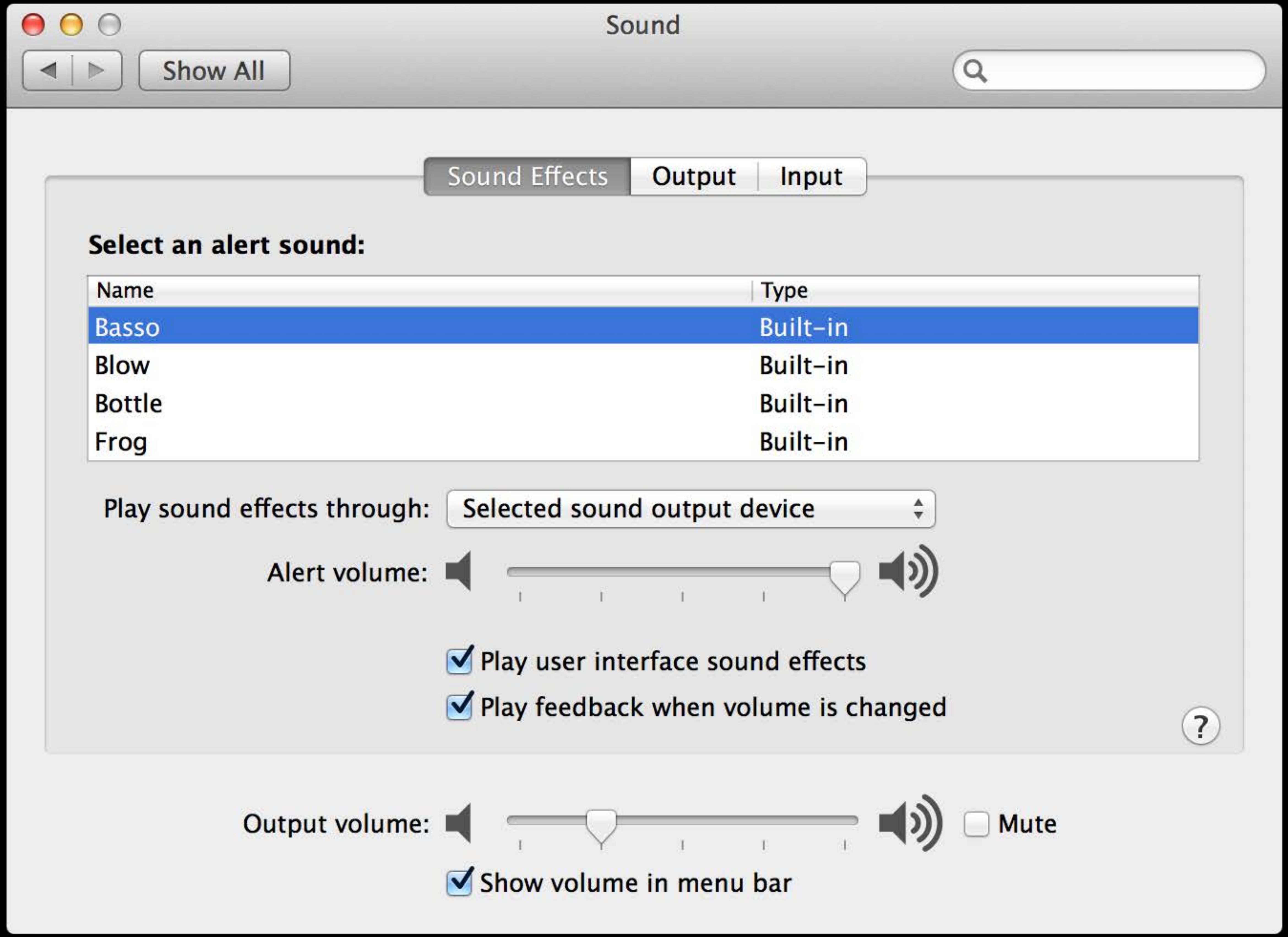
✓ Play feedback when volume is changed

Output volume: Mute

✓ Show volume in menu bar

Mavericks

Updated Look for Controls



Select an alert sound:

Name	Type
Basso	Built-in
Blow	Built-in
Bottle	Built-in
Frog	Built-in

Play sound effects through: Selected sound output device

Alert volume:

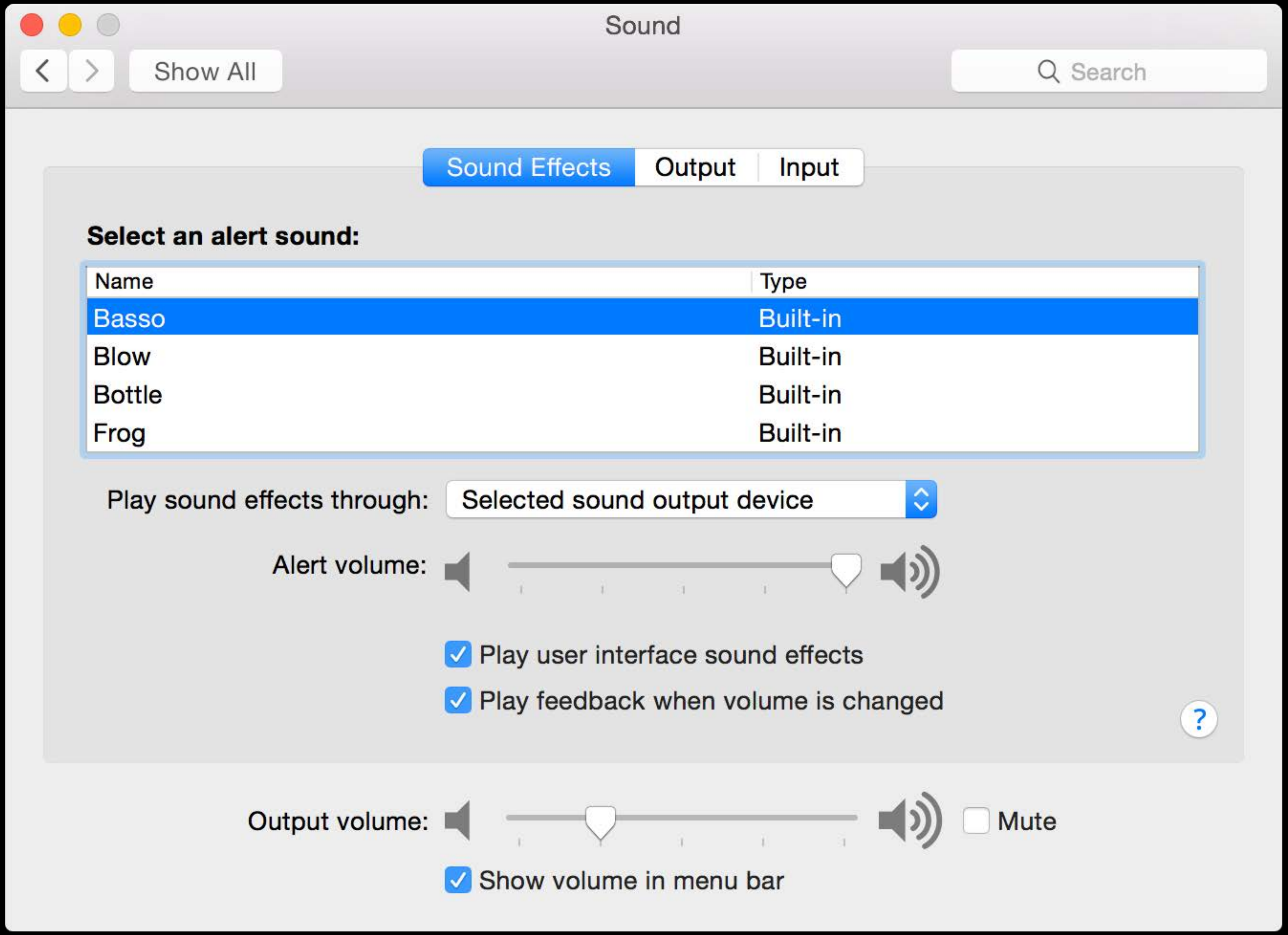
✓ Play user interface sound effects

✓ Play feedback when volume is changed

Output volume: ☐ Mute

✓ Show volume in menu bar

Mavericks



Select an alert sound:

Name	Type
Basso	Built-in
Blow	Built-in
Bottle	Built-in
Frog	Built-in

Play sound effects through: Selected sound output device

Alert volume:

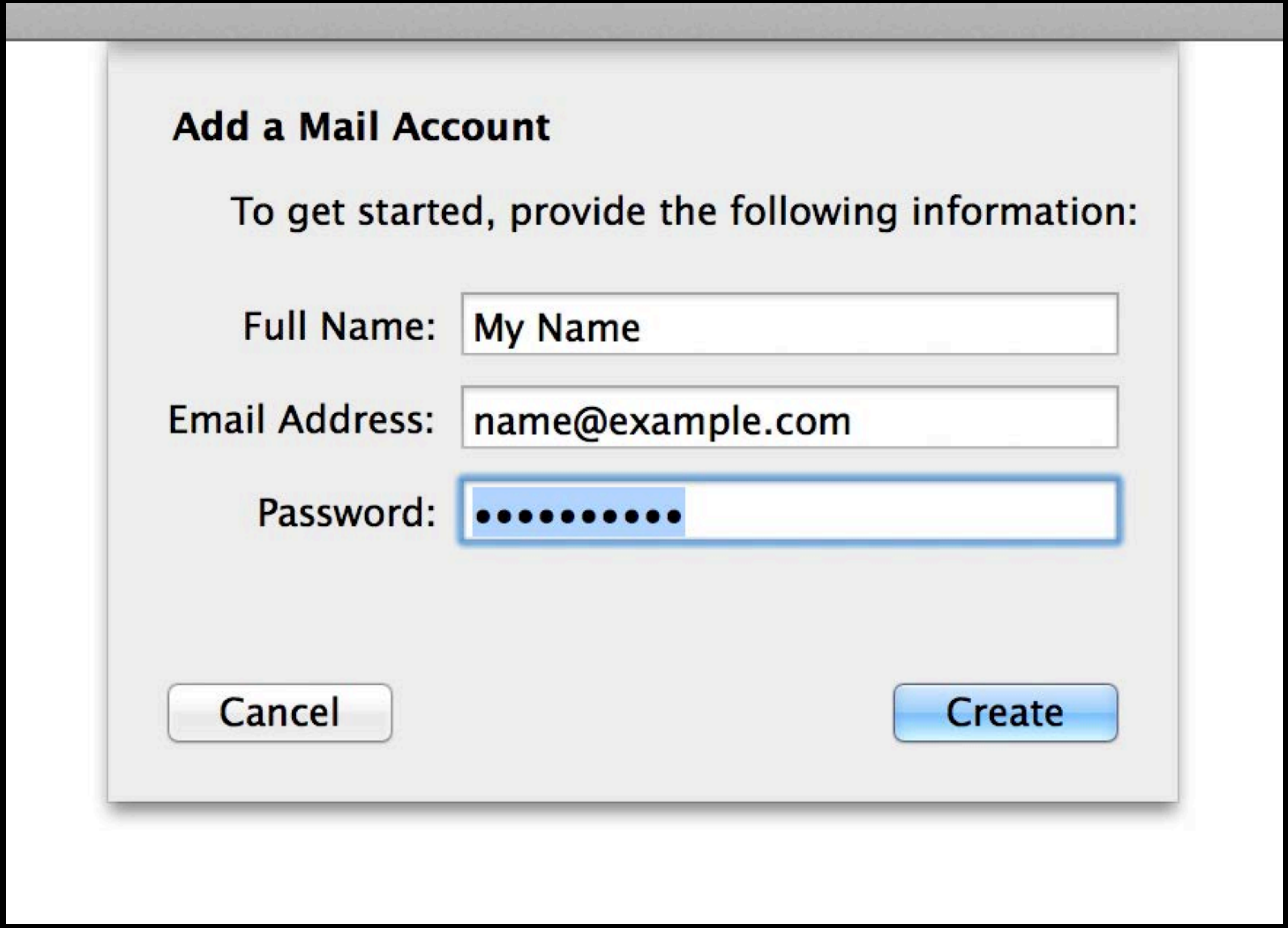
✓ Play user interface sound effects

✓ Play feedback when volume is changed

Output volume: ☐ Mute

✓ Show volume in menu bar

Yosemite

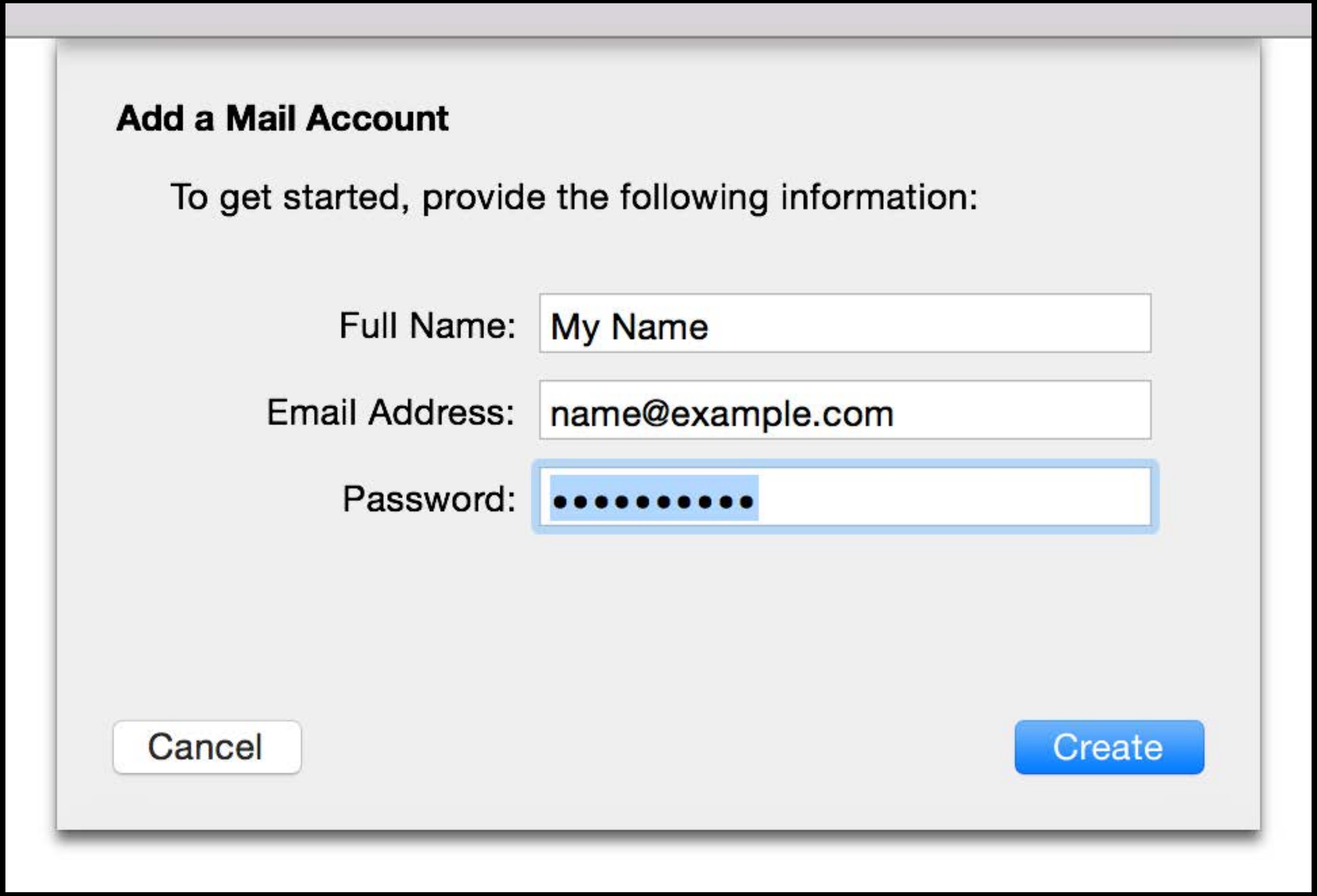


Add a Mail Account

To get started, provide the following information:

Full Name:	My Name
Email Address:	name@example.com
Password:	••••••••••

Mavericks

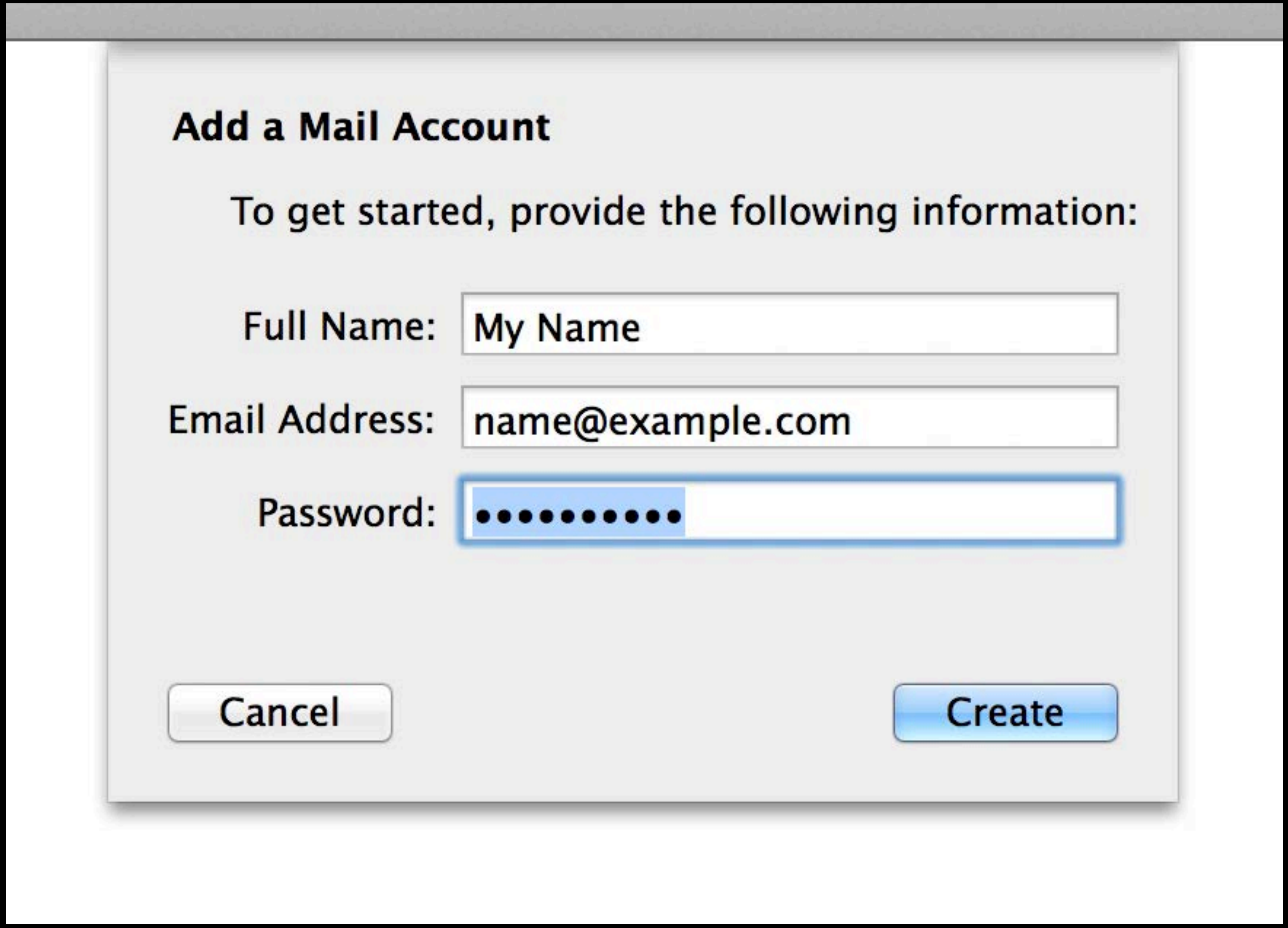


Add a Mail Account

To get started, provide the following information:

Full Name:	My Name
Email Address:	name@example.com
Password:	••••••••••

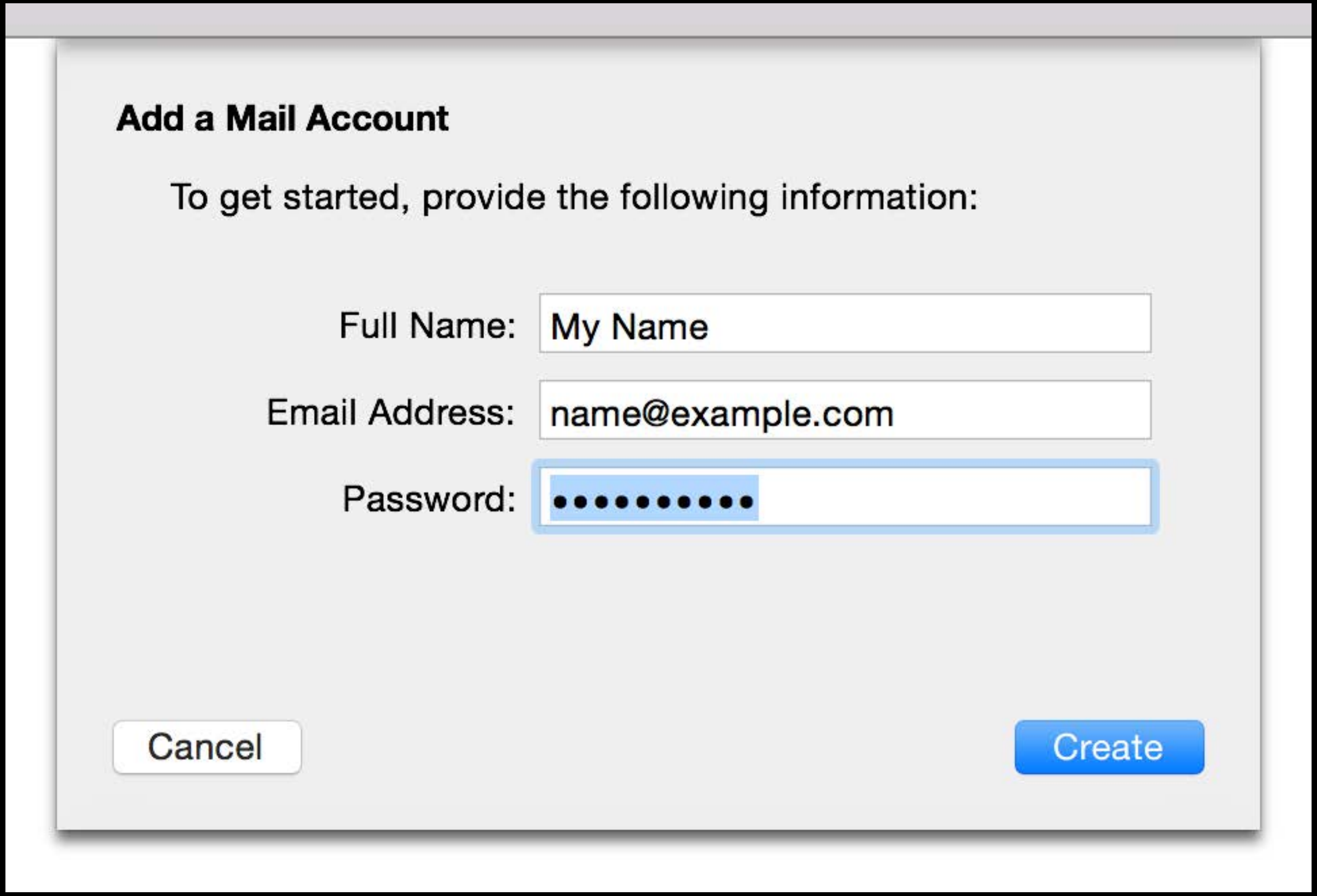
Yosemite


Add a Mail Account

To get started, provide the following information:

Full Name:	My Name
Email Address:	name@example.com
Password:	••••••••••

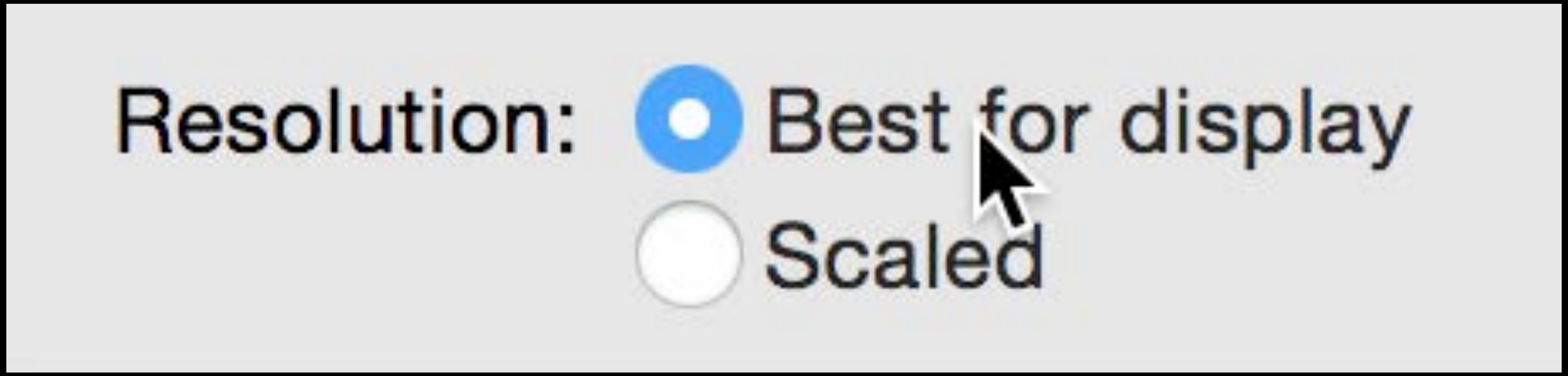
Mavericks


Add a Mail Account

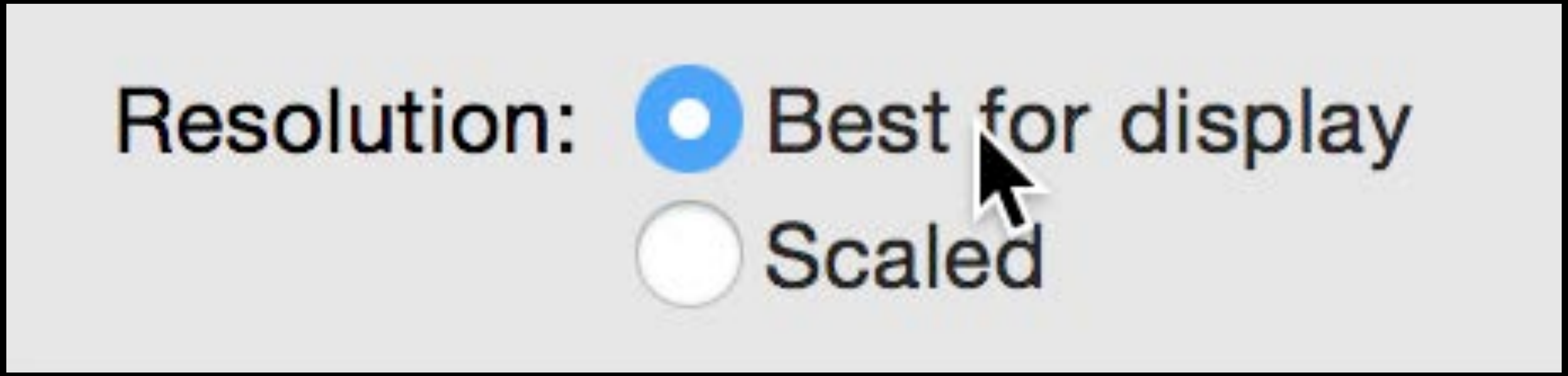
To get started, provide the following information:

Full Name:	My Name
Email Address:	name@example.com
Password:	••••••••••

Yosemite



Resolution: [● Best for display] ○ Scaled


Resolution: [✓ Best for display] Scaled

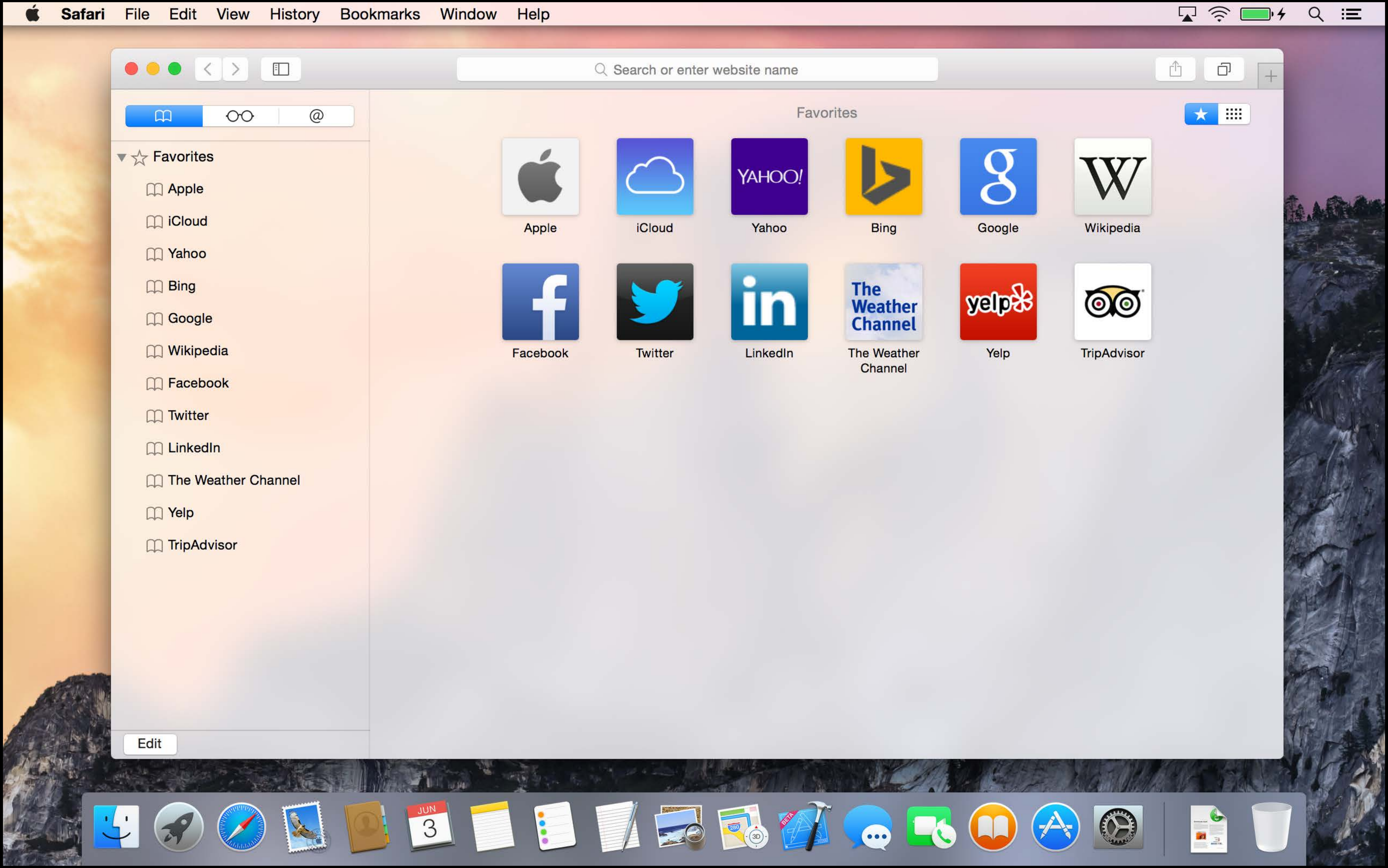
Updated Look for Controls

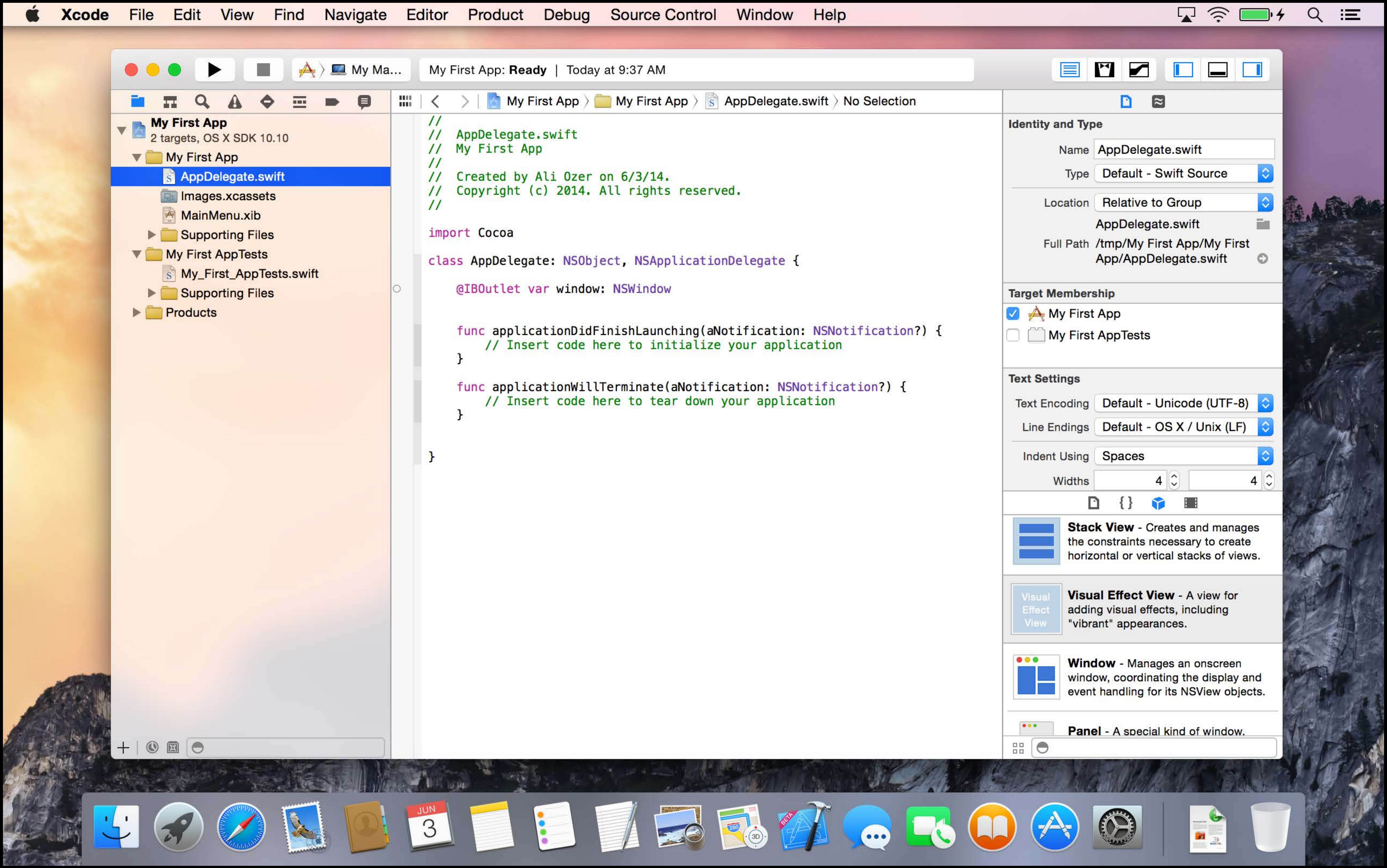
Automatic!

Translucency

Transparency + Blur

Finder
File
Edit
View
Go
Window
Help
JUN
3


Safari
File
Edit
View
History
Bookmarks
Window
Help
Search or enter website name
Favorites
Apple
iCloud
Yahoo
Bing
Google
Wikipedia
Facebook
Twitter
LinkedIn
The Weather Channel
Yelp
TripAdvisor
Edit
Favorites
Apple
iCloud
Yahoo
Bing
Google
Wikipedia
Facebook
Twitter
LinkedIn
The Weather Channel
Yelp
TripAdvisor


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
My Ma...
My First App: Ready | Today at 9:37 AM
My First App 〉 My First App 〉 AppDelegate.swift 〉 No Selection
My First App
2 targets, OS X SDK 10.10
My First App
AppDelegate.swift
Images.xcassets
MainMenu.xib
Supporting Files
My First AppTests
My_First_AppTests.swift
Supporting Files
Products

```
//
//  AppDelegate.swift
//  My First App
//
//  Created by Ali Ozer on 6/3/14.
//  Copyright (c) 2014. All rights reserved.
//

import Cocoa

class AppDelegate: NSObject, NSApplicationDelegate {

    @IBOutlet var window: NSWindow


    func applicationDidFinishLaunching(aNotification: NSNotification?) {
        // Insert code here to initialize your application
    }

    func applicationWillTerminate(aNotification: NSNotification?) {
        // Insert code here to tear down your application
    }


}
```

Identity and Type

Name: AppDelegate.swift
Type: Default - Swift Source
Location: Relative to Group
AppDelegate.swift
Full Path: /tmp/My First App/My First App/AppDelegate.swift

Target Membership

✓ My First App
☐ My First AppTests

Text Settings

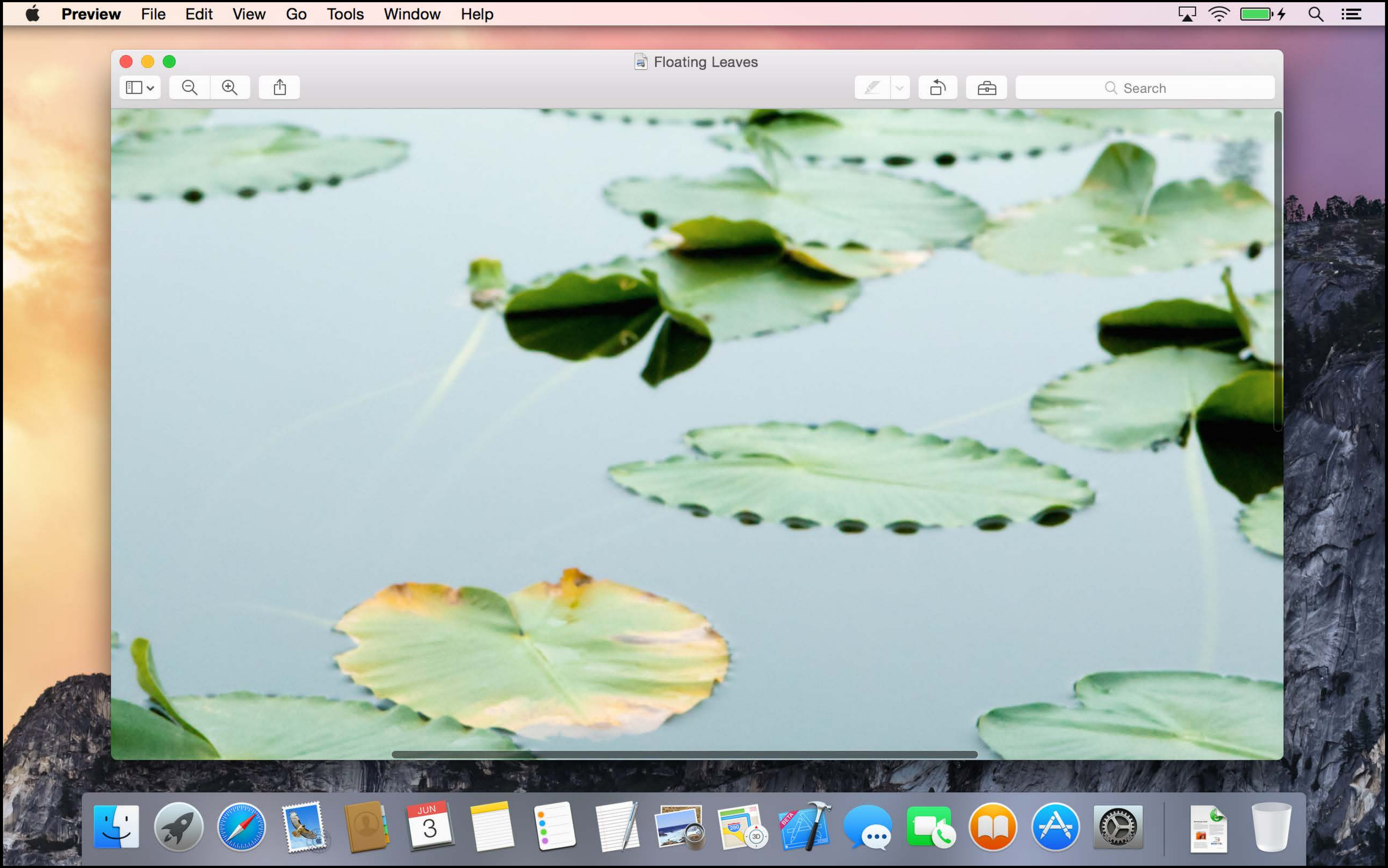
Text Encoding: Default - Unicode (UTF-8)
Line Endings: Default - OS X / Unix (LF)
Indent Using: Spaces
Widths: 4 4

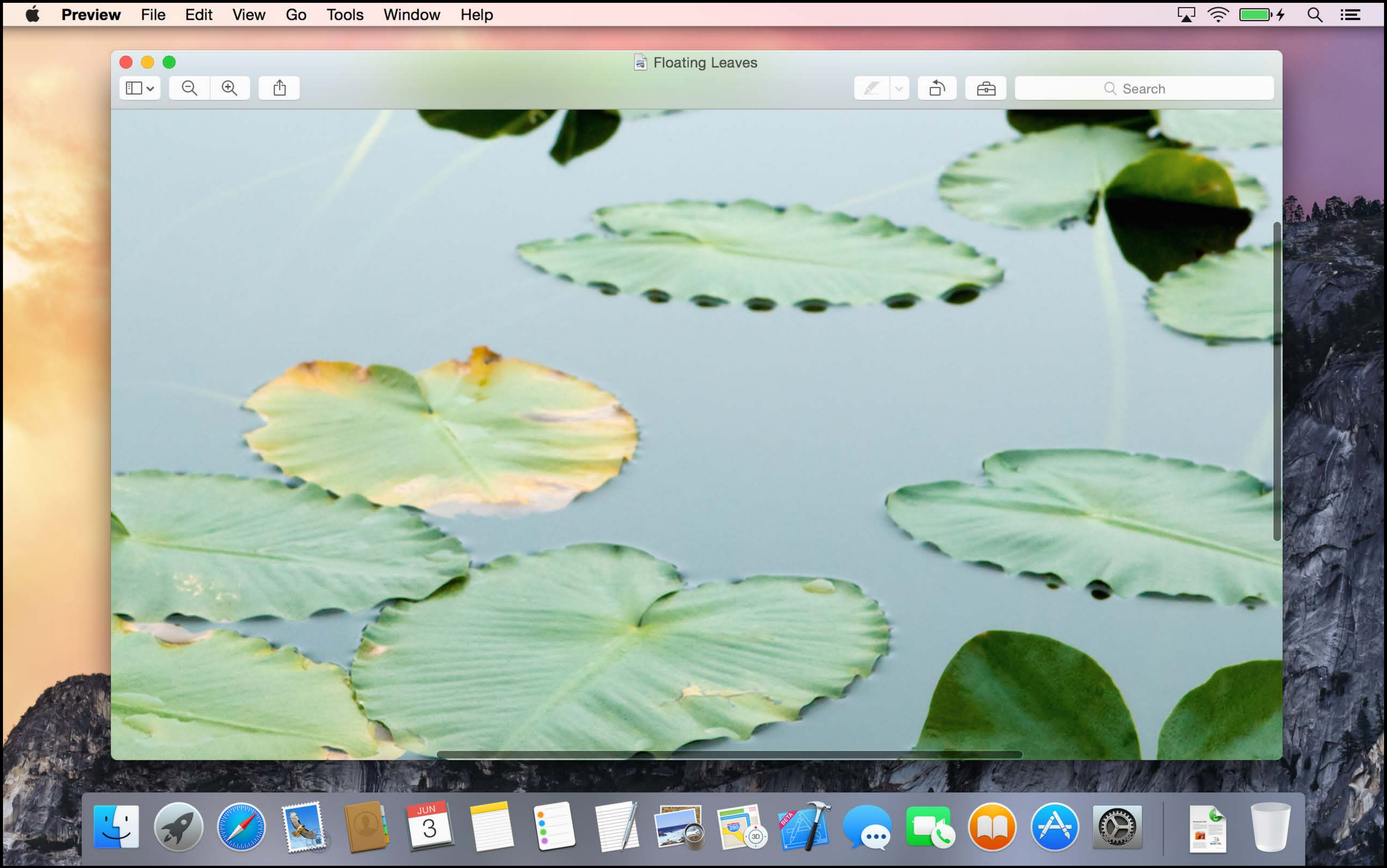
Stack View - Creates and manages the constraints necessary to create horizontal or vertical stacks of views.

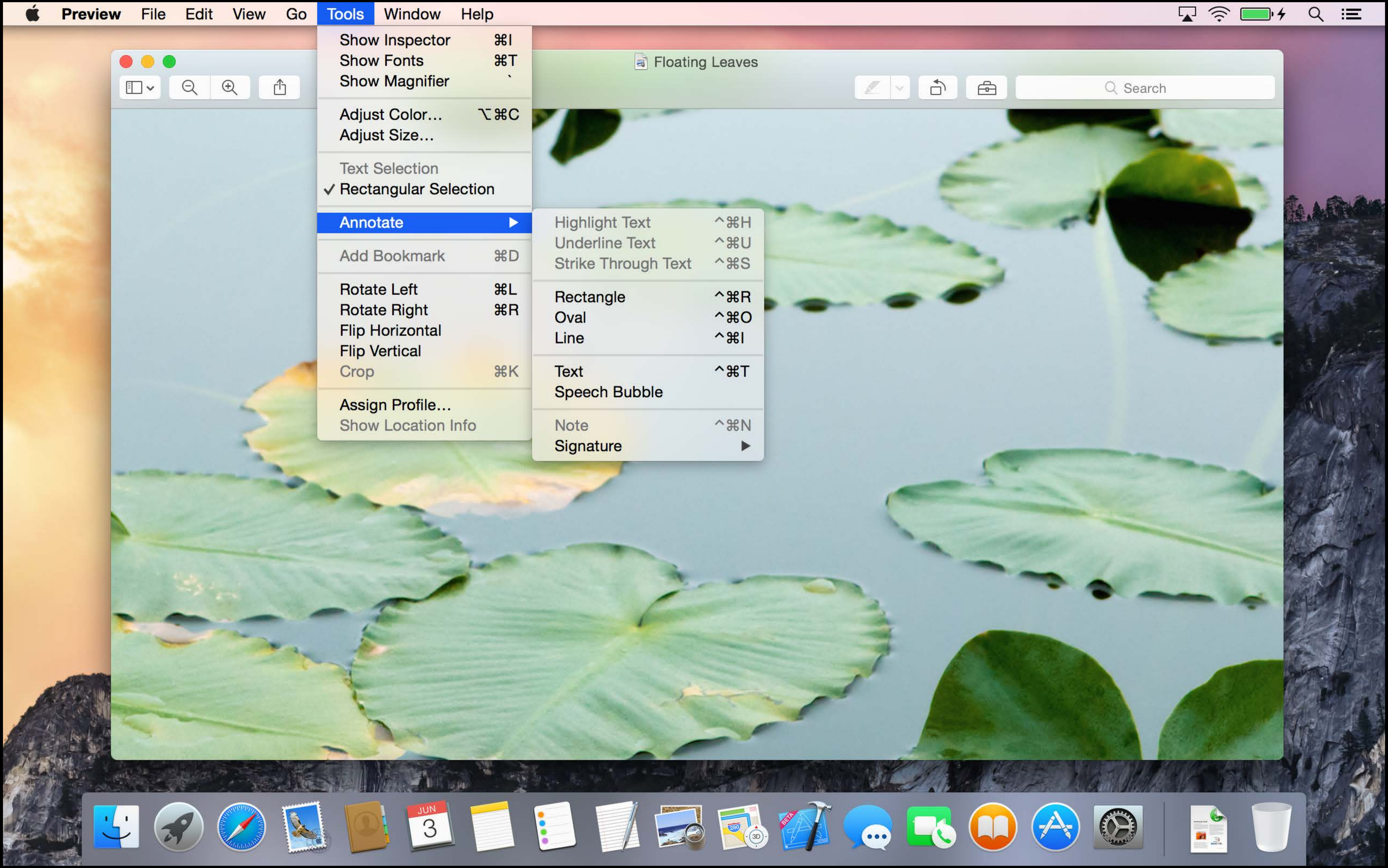
Visual Effect View - A view for adding visual effects, including "vibrant" appearances.

Window - Manages an onscreen window, coordinating the display and event handling for its NSView objects.

Panel - A special kind of window.


Preview
File
Edit
View
Go
Tools
Window
Help
Floating Leaves
Search
JUN
3


Preview
File
Edit
View
Go
Tools
Window
Help
Floating Leaves
Search
JUN
3


Preview File Edit View Go Tools Window Help
Floating Leaves
Search
Show Inspector ⌘I
Show Fonts ⌘T
Show Magnifier `
Adjust Color… ⌥⌘C
Adjust Size…
Text Selection
✓ Rectangular Selection
Annotate ▶
Add Bookmark ⌘D
Rotate Left ⌘L
Rotate Right ⌘R
Flip Horizontal
Flip Vertical
Crop ⌘K
Assign Profile…
Show Location Info
Highlight Text ^⌘H
Underline Text ^⌘U
Strike Through Text ^⌘S
Rectangle ^⌘R
Oval ^⌘O
Line ^⌘I
Text ^⌘T
Speech Bubble
Note ^⌘N
Signature ▶
JUN 3

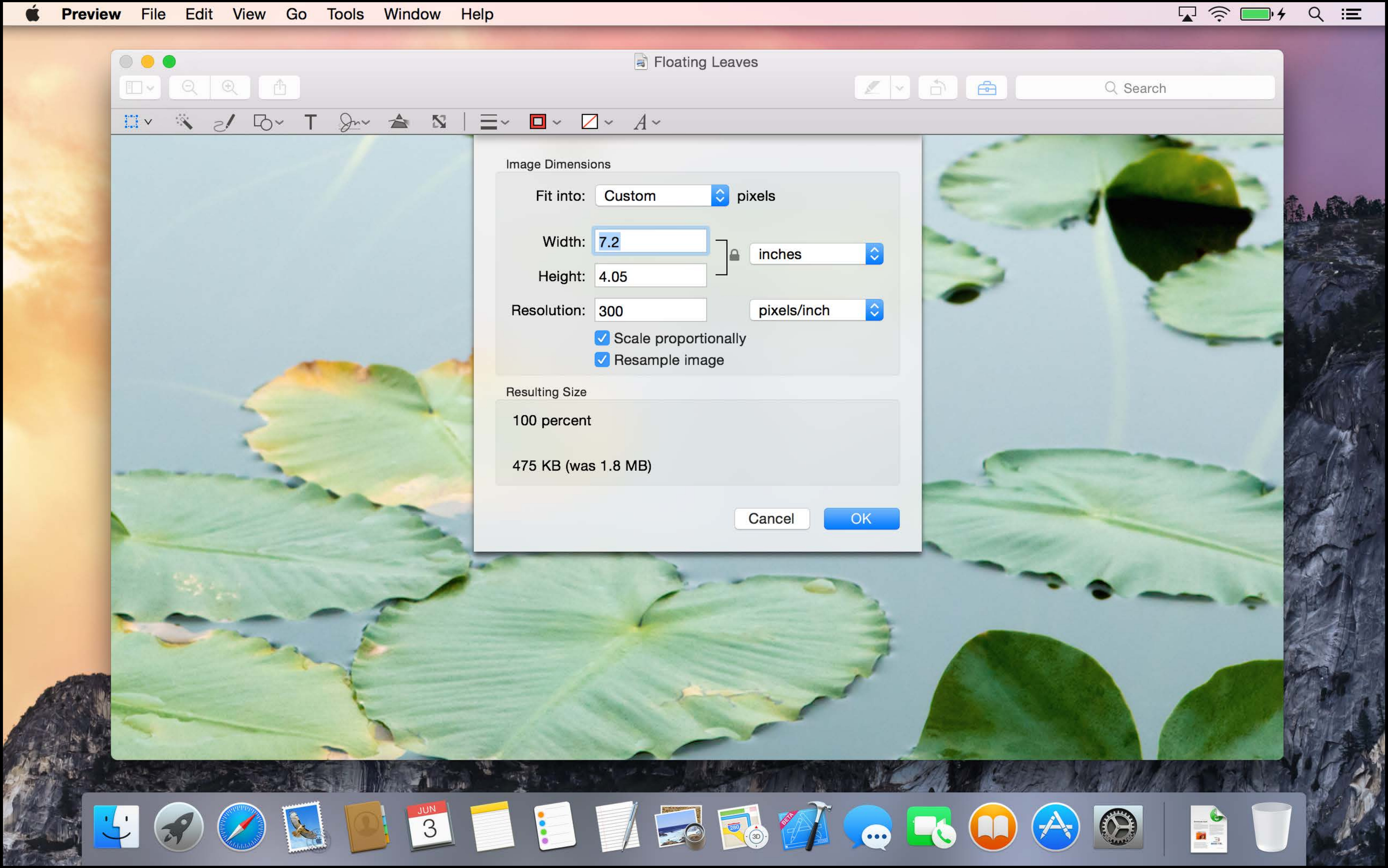


Image Dimensions

Fit into: Custom pixels

Width: 7.2

Height: 4.05

inches

Resolution: 300 pixels/inch

☑ Scale proportionally

☑ Resample image

Resulting Size

100 percent

475 KB (was 1.8 MB)

Translucency

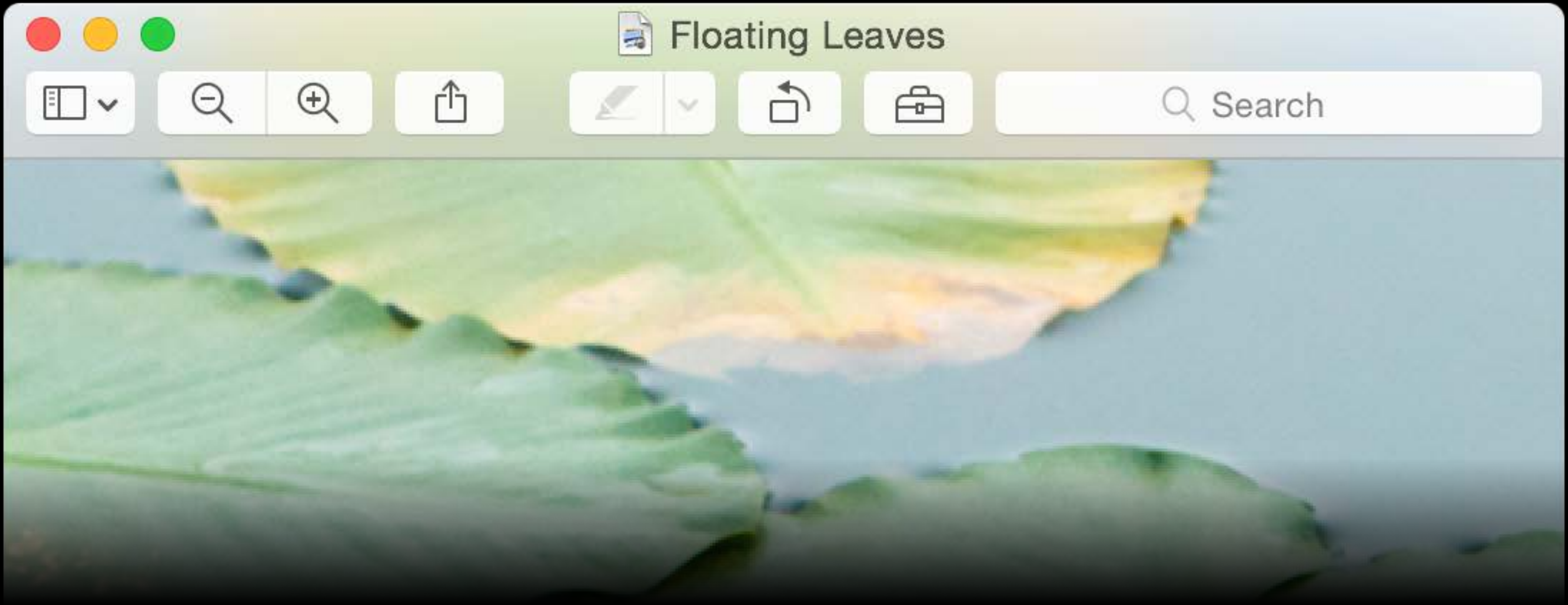
Automatic in many cases

- Sheets, menus, popovers
- Source lists
- Titlebars/toolbars

NSWindow

Translucent Titlebar/Toolbar

Automatic for any NSScrollView next to title bar



NSWindow

Translucent Titlebar/Toolbar

For other cases, use new style NSFullSizeContentViewWindowMask

NSWindow

Translucent Titlebar/Toolbar

For other cases, use new style NSFullSizeContentViewWindowMask

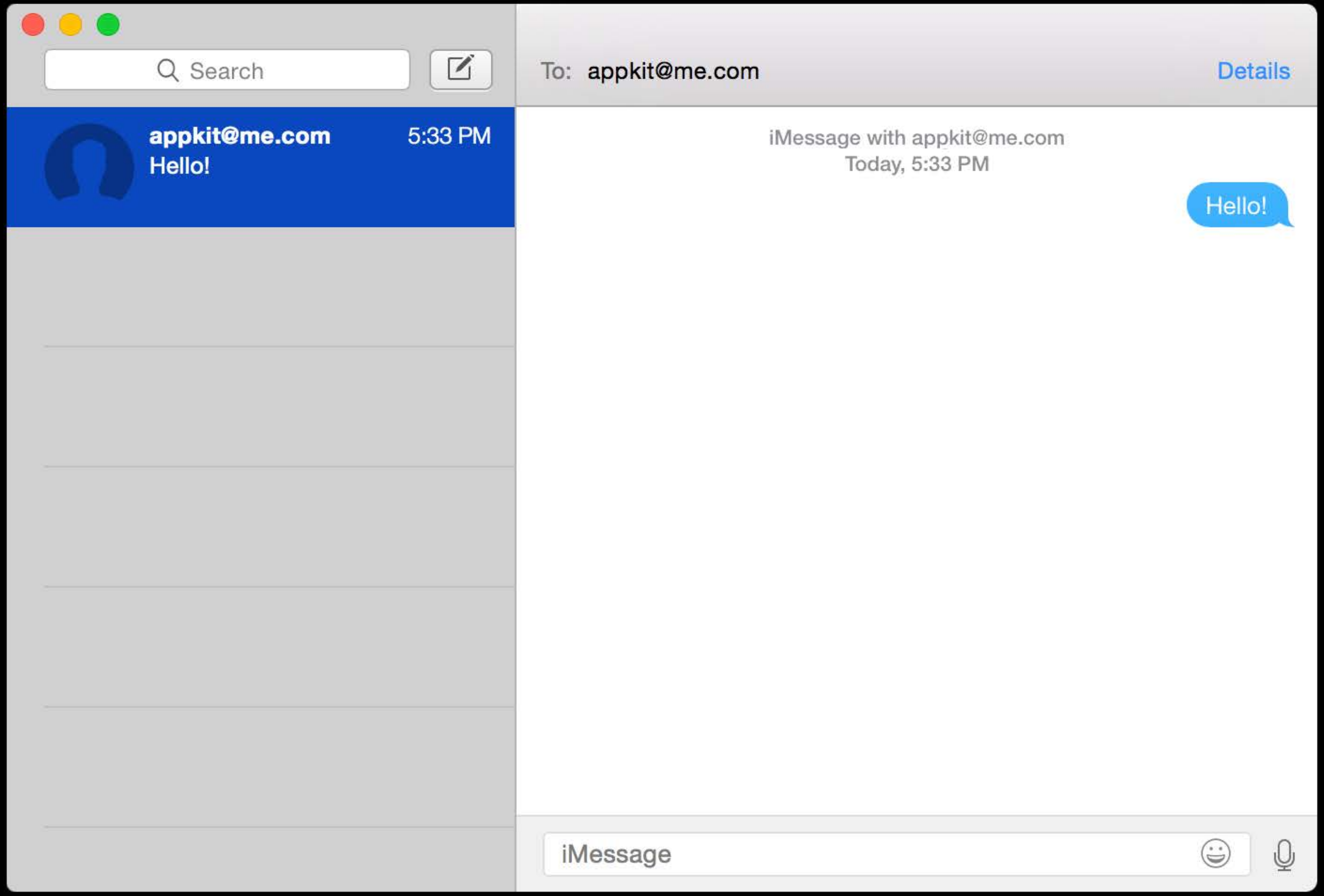
Further customize the title bar with titleVisibility and titlebarAppearsTransparent

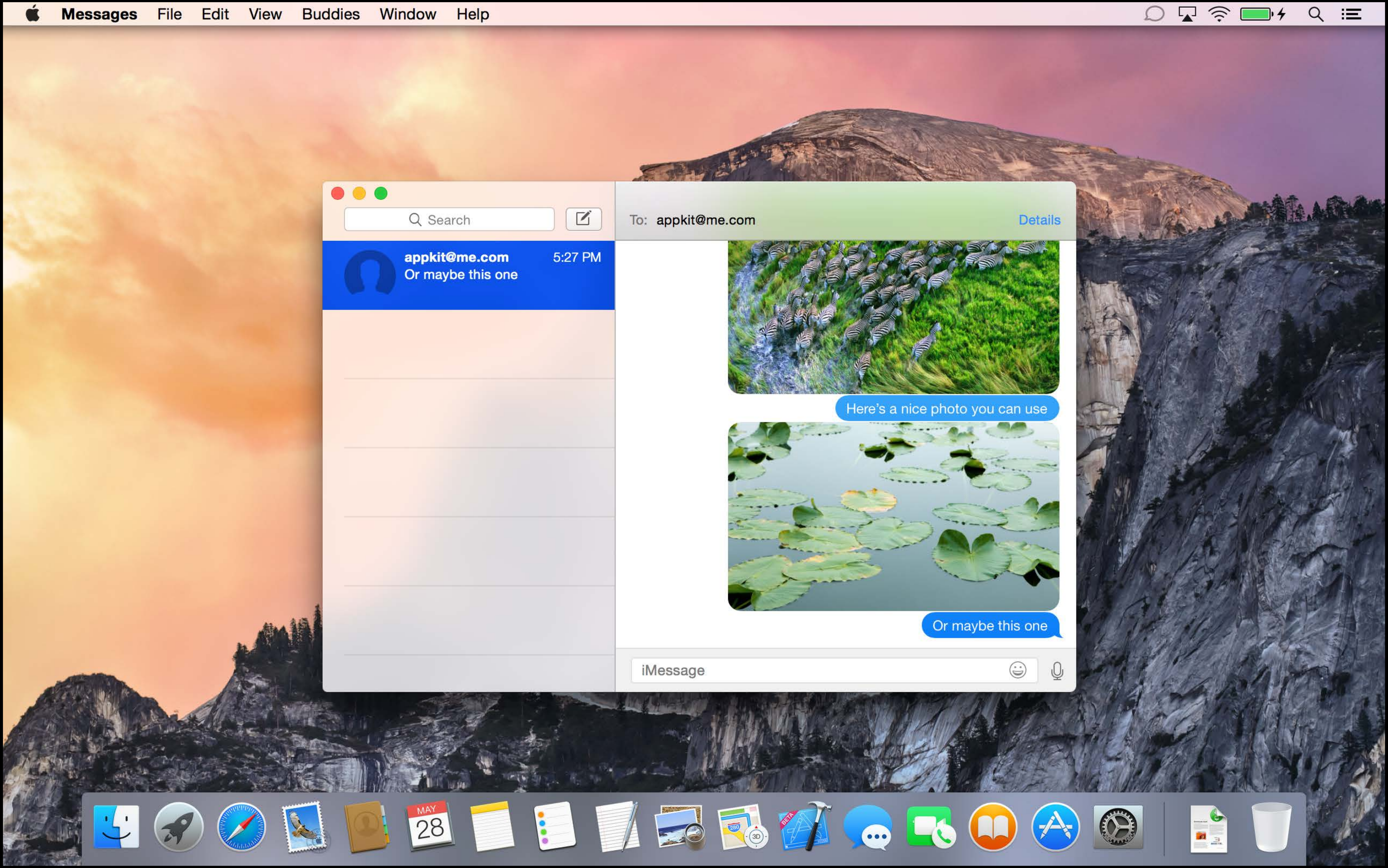
NSWindow

Translucent Titlebar/Toolbar

For other cases, use new style NSFullSizeContentViewWindowMask

Further customize the title bar with titleVisibility and titlebarAppearsTransparent





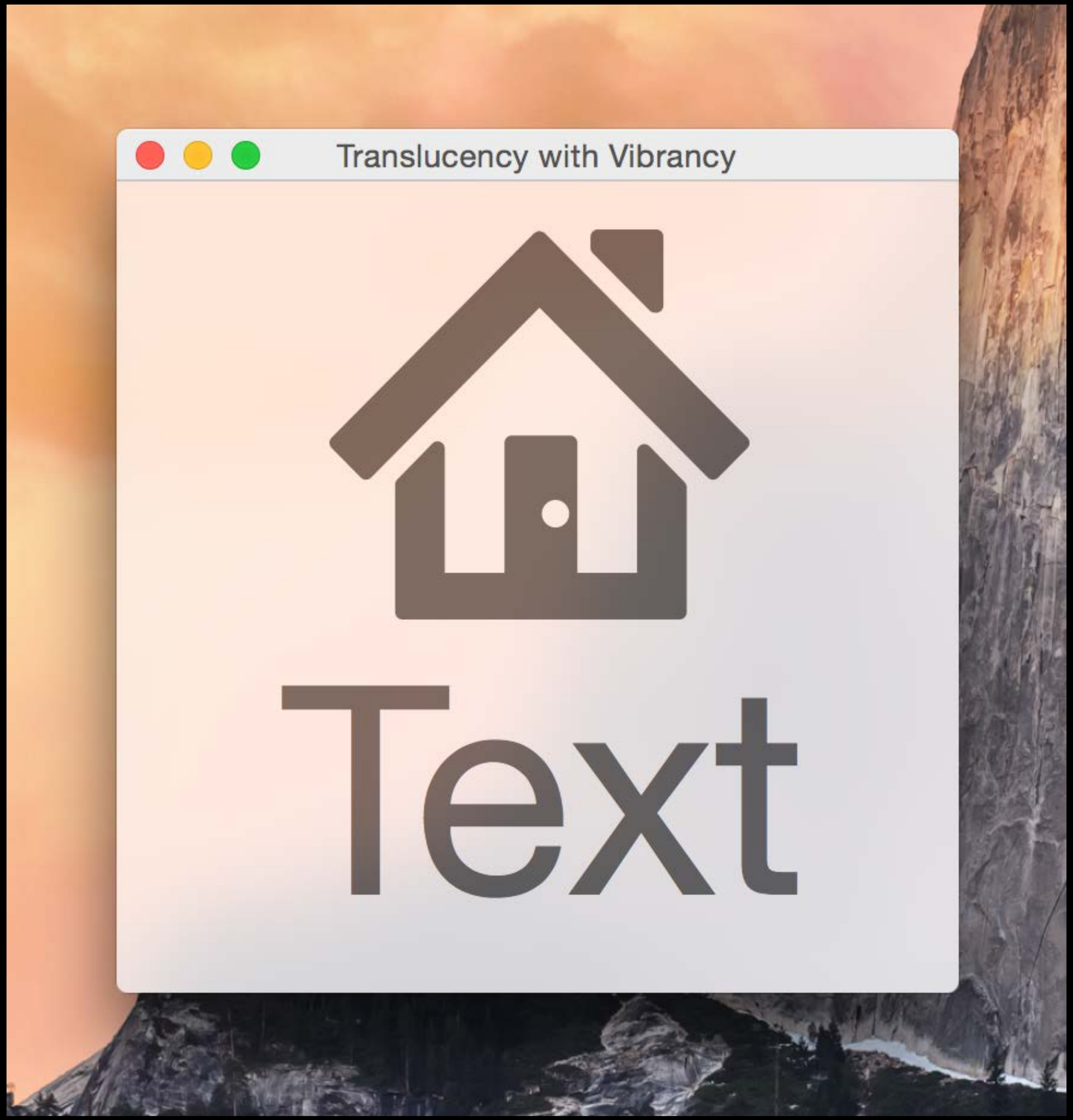
appkit@me.com 5:27 PM
Or maybe this one

Here's a nice photo you can use

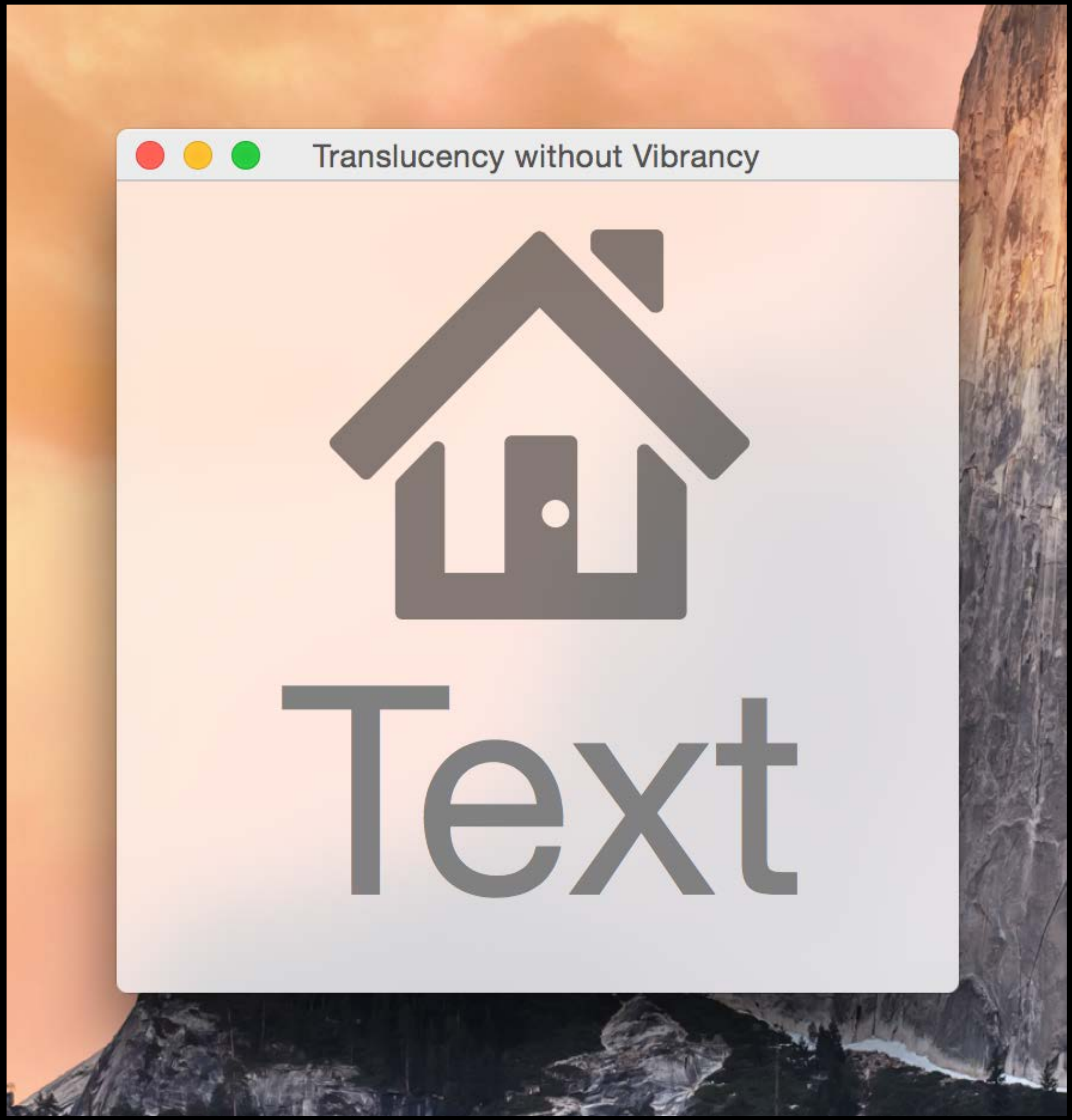
Or maybe this one

Vibrancy

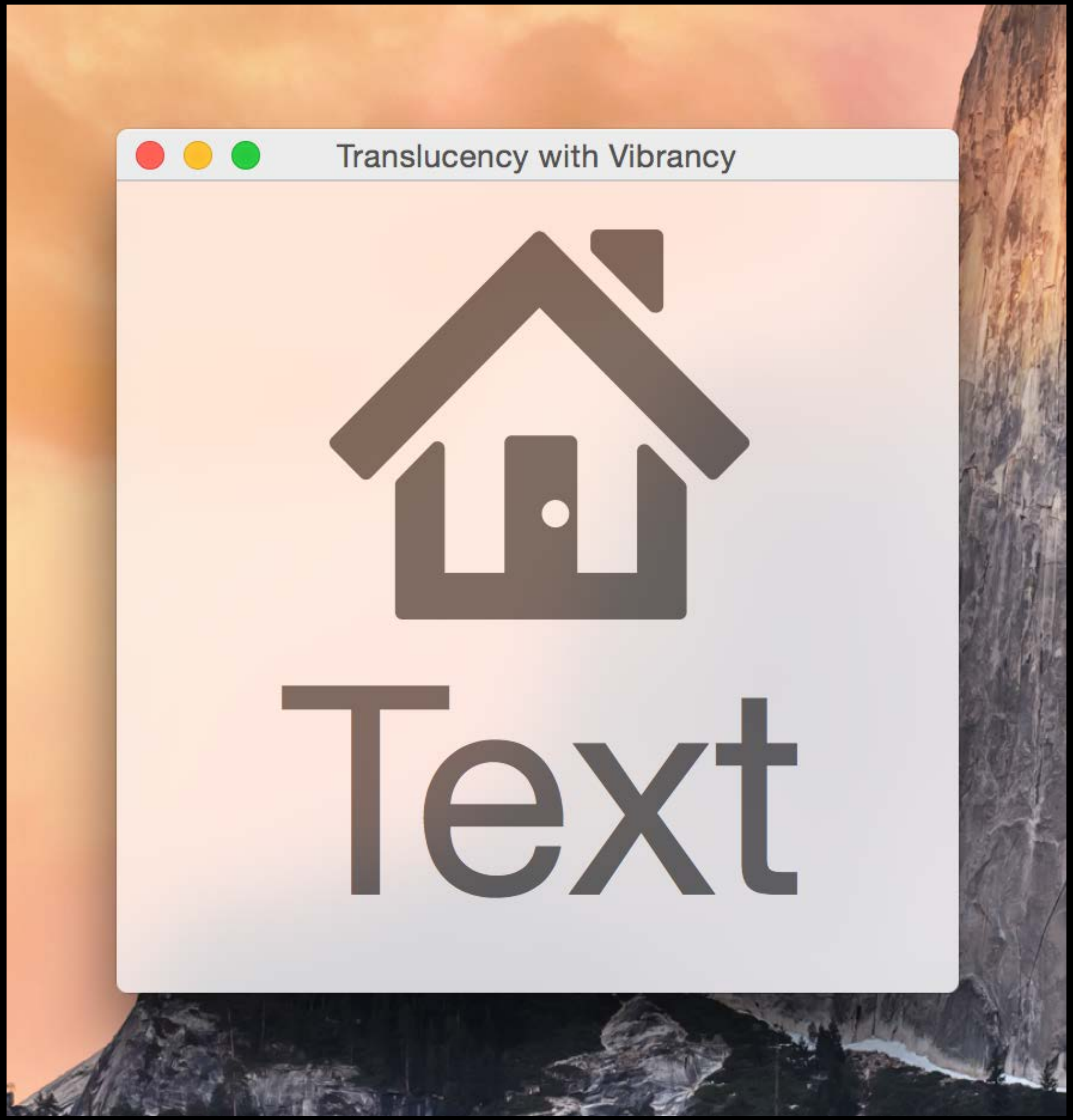
Vibrancy



Vibrancy



Vibrancy



Vibrancy

Automatic in contexts where we apply translucency

- For controls and other NSViews when appropriate

Vibrancy

Enabling it explicitly

Vibrancy

Enabling it explicitly

Create a top-level NSVisualEffectView

- Specify in-window or behind-window translucency with blendingMode

Vibrancy

Enabling it explicitly

Create a top-level NSVisualEffectView

- Specify in-window or behind-window translucency with blendingMode

Specify a vibrant appearance

- NSAppearanceNameVibrantLight or NSAppearanceNameVibrantDark

Vibrancy

Enabling it explicitly

Create a top-level NSVisualEffectView

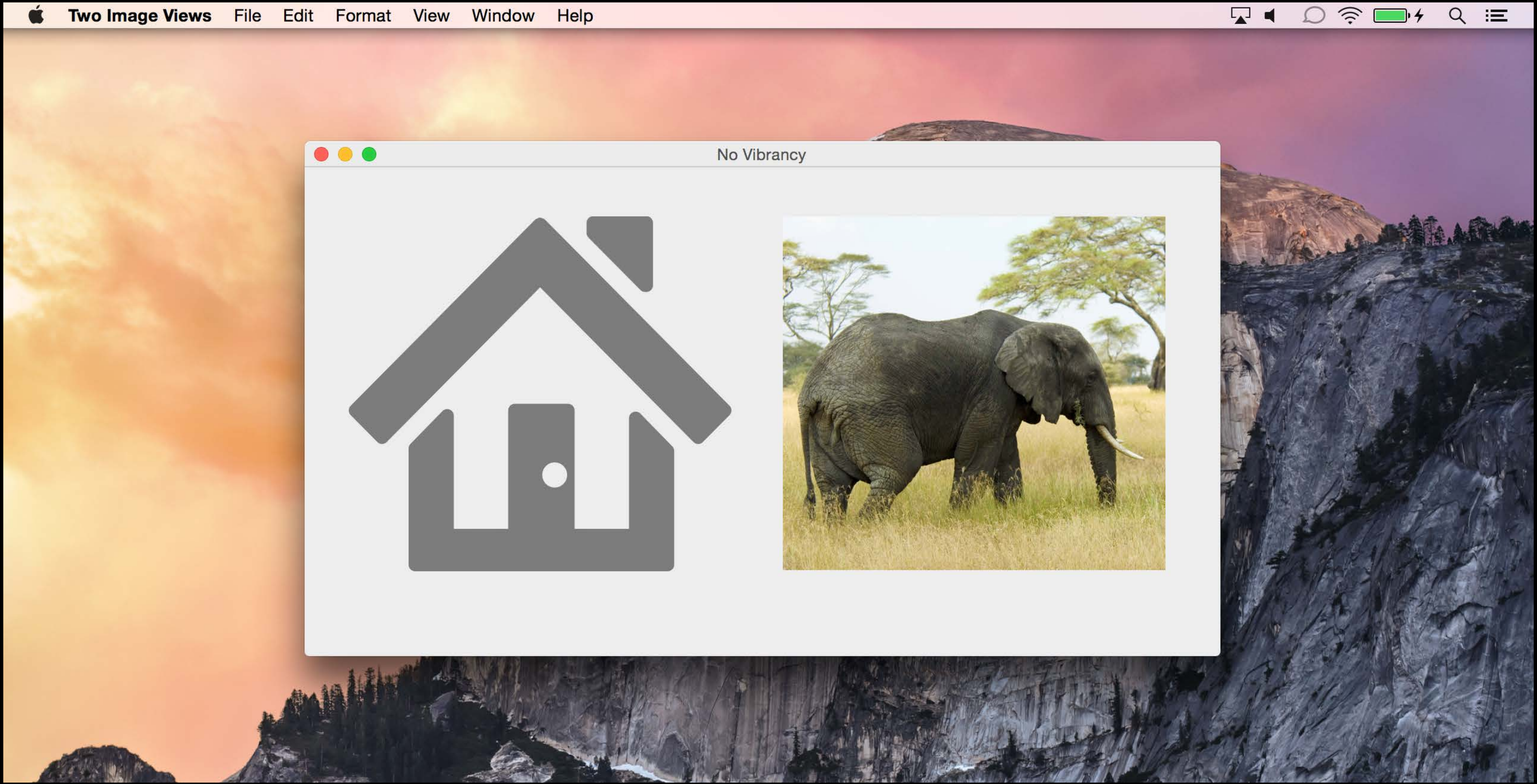
- Specify in-window or behind-window translucency with blendingMode

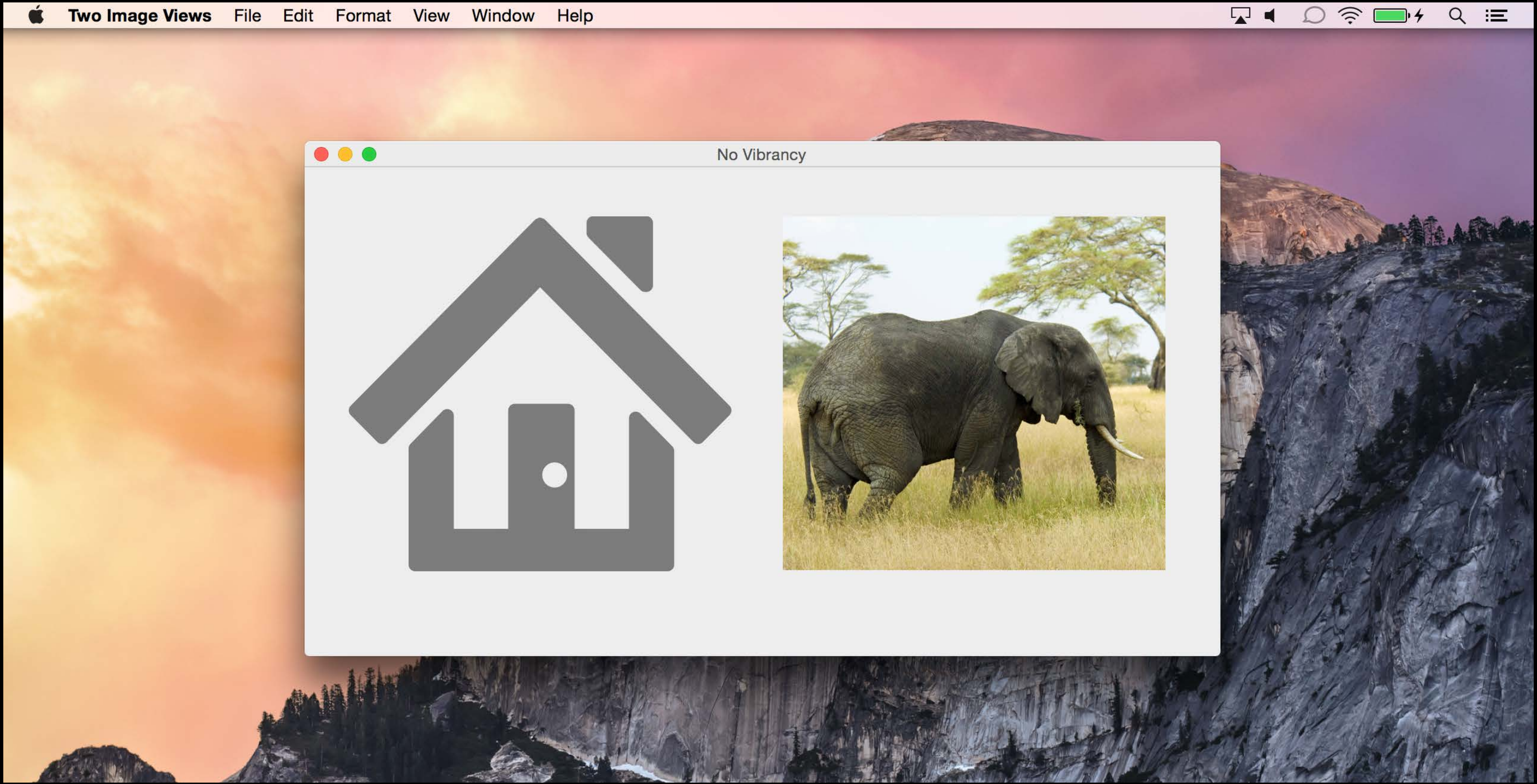
Specify a vibrant appearance

- NSAppearanceNameVibrantLight or NSAppearanceNameVibrantDark

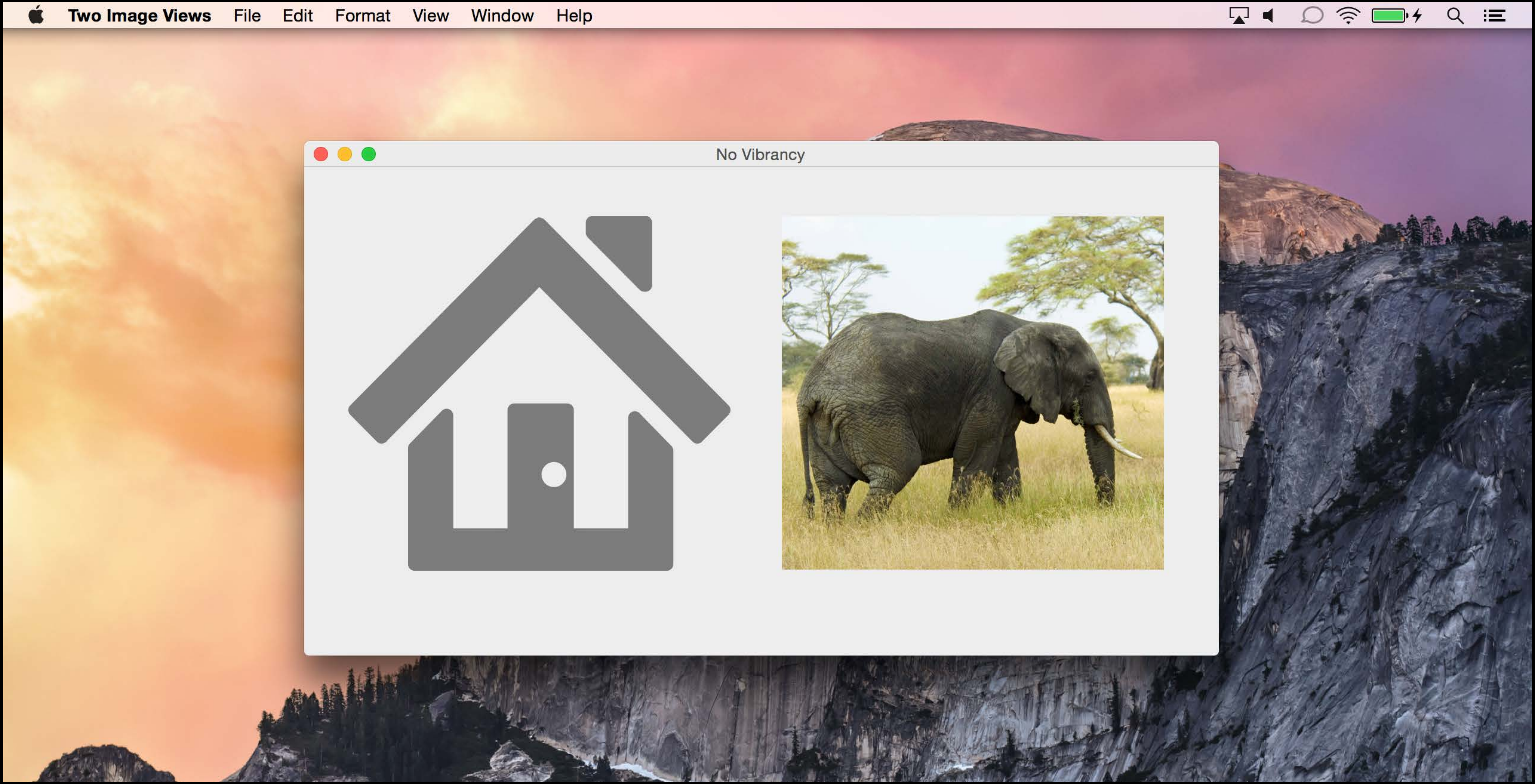
Populate with vibrant-capable controls

- Such controls return YES from allowsVibrancy
- Some controls opt-in to vibrancy dynamically


Two Image Views
File
Edit
Format
View
Window
Help
No Vibrancy

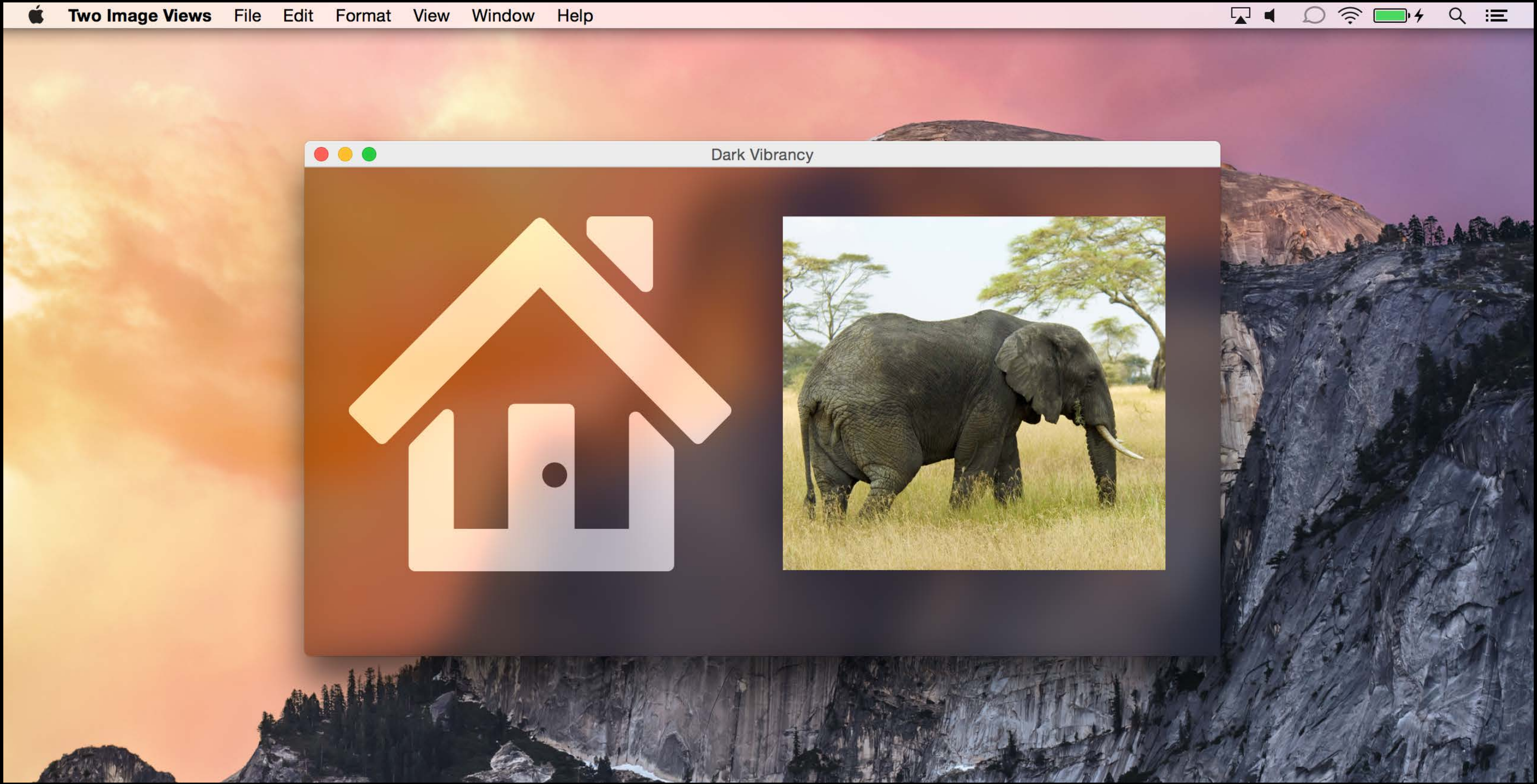


NSImageView with
template image



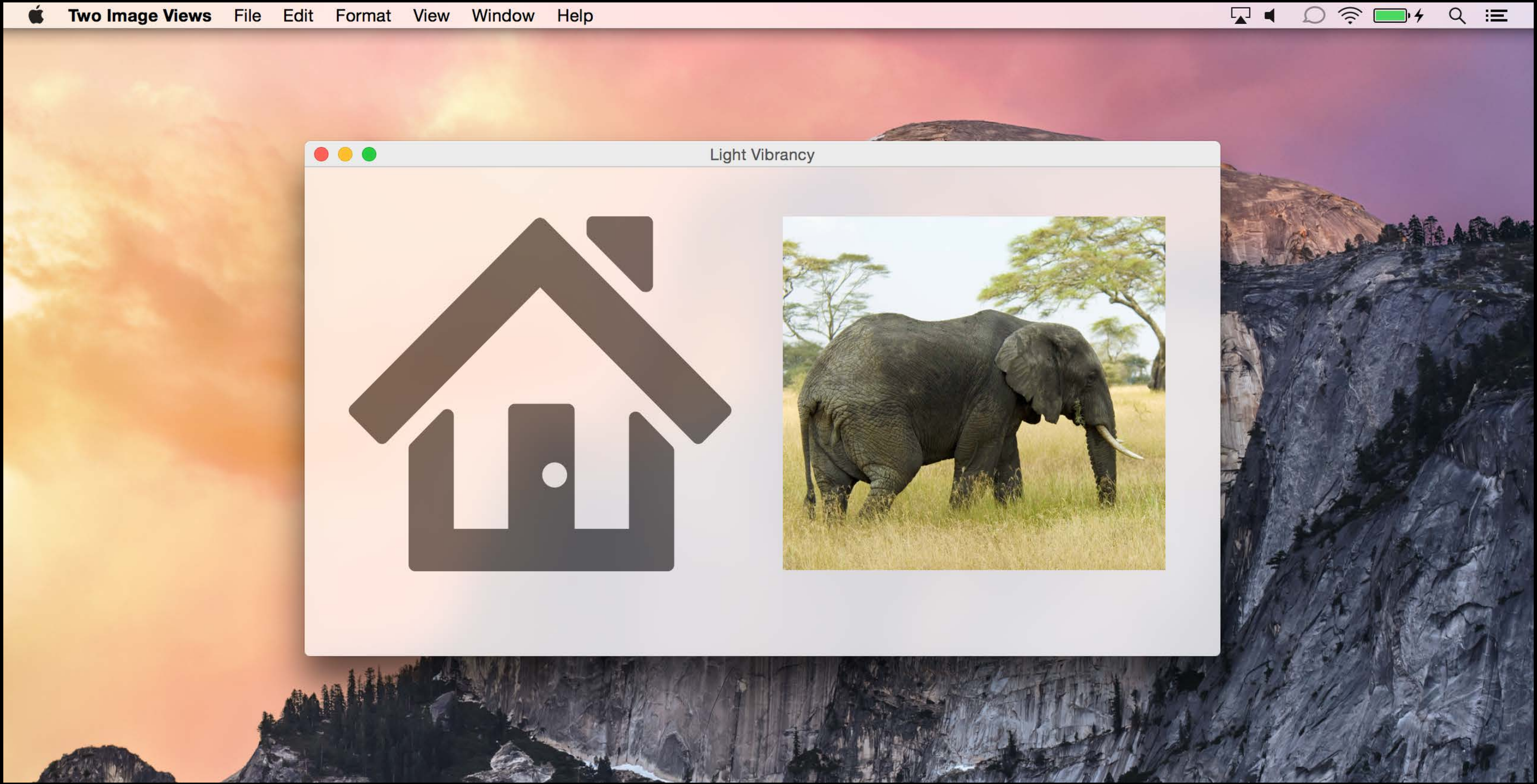
NSImageView with template image

NSImageView with regular image



NSImageView with template image

NSImageView with regular image



NSImageView with template image

NSImageView with regular image

NSColor

Existing system colors have been updated

- And many have appearance-specific variants

NSColor

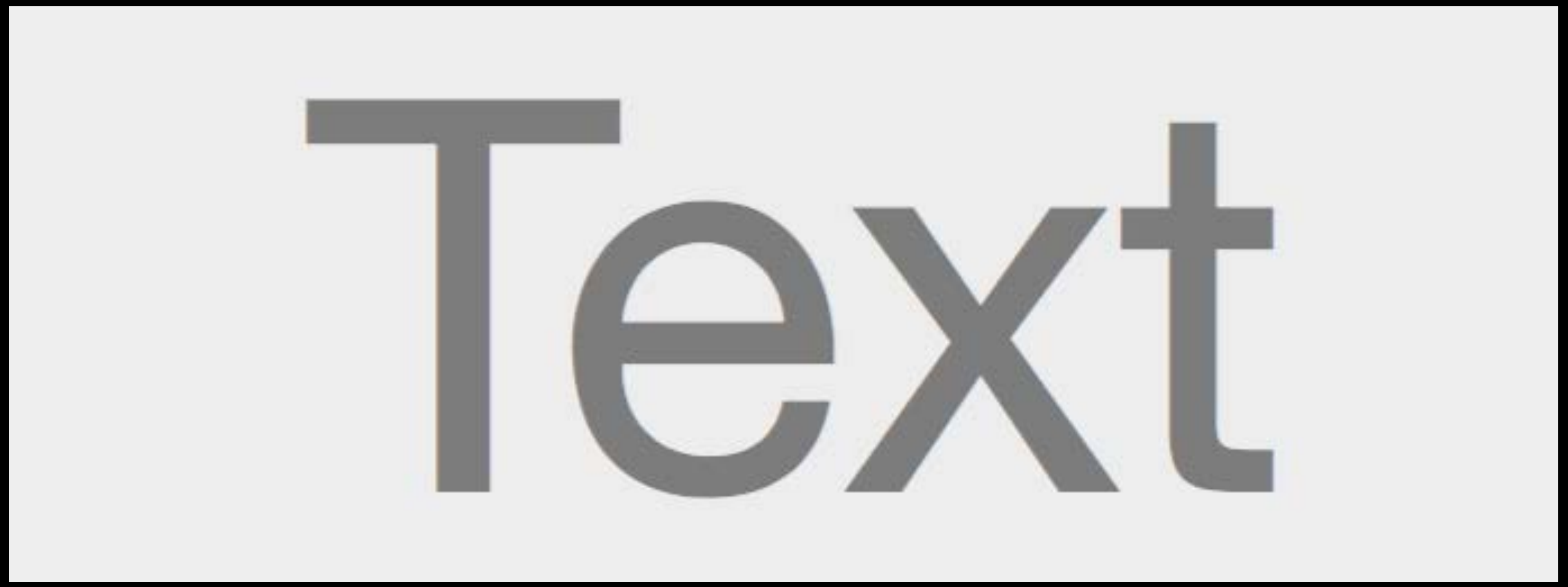
Existing system colors have been updated

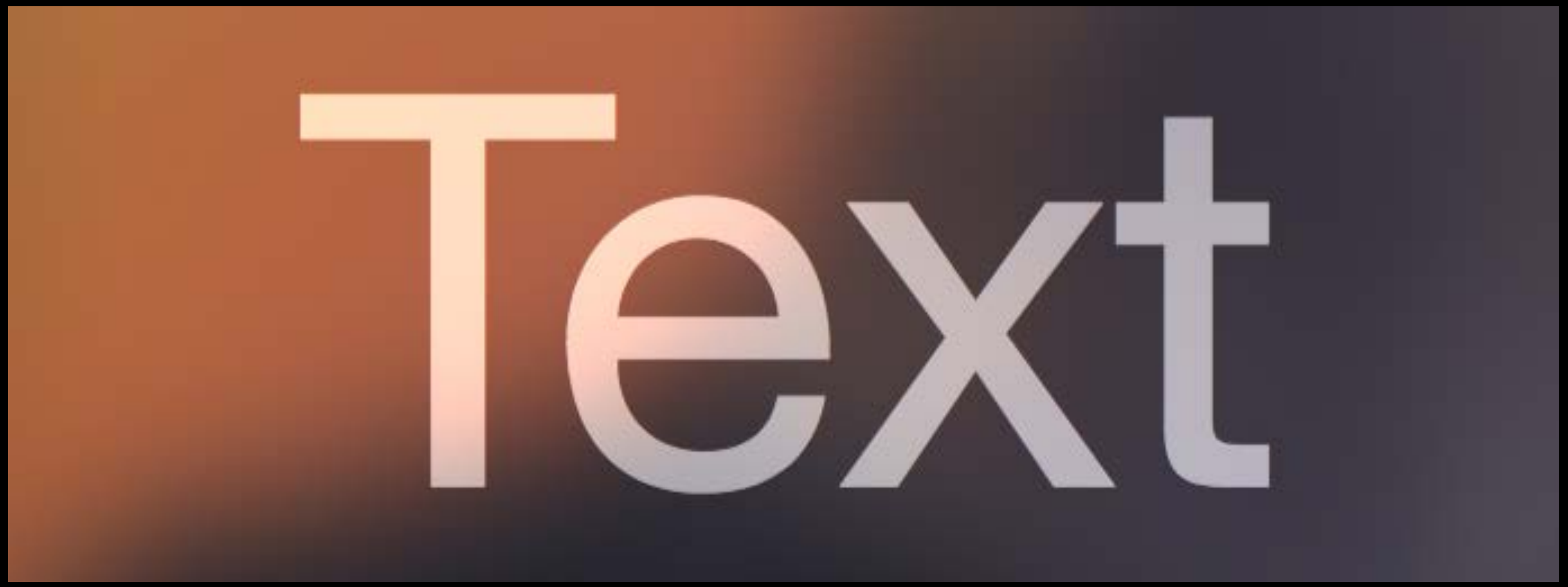
- And many have appearance-specific variants

NSColor

Existing system colors have been updated

- And many have appearance-specific variants





NSColor

Existing system colors have been updated

- And many have appearance-specific variants

```
Text color = [NSColor secondaryLabelColor];
```

New Font

New Font

Helvetica Neue optimized for OS X

- Metrics similar to Lucida

New Font

Helvetica Neue optimized for OS X

- Metrics similar to Lucida

Obtain via methods such as `systemFontOfSize:`

New Font

Helvetica Neue optimized for OS X

- Metrics similar to Lucida

Obtain via methods such as systemFontOfSize:

In applications linked against 10.9 SDK and earlier

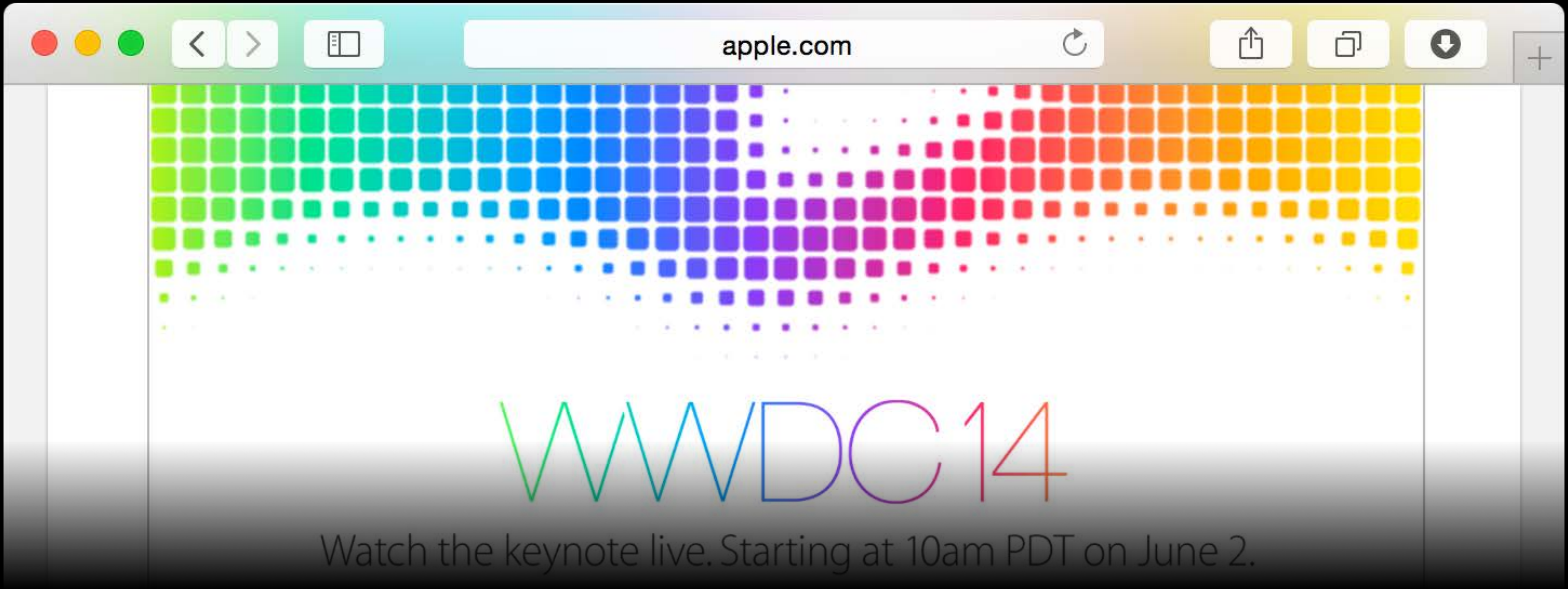
- Explicit references to Lucida Grande in UI elements will be replaced at runtime
- Text will be compressed if too tight

NSSegmentedControl

New style for back/forward buttons, NSSegmentStyleSeparated

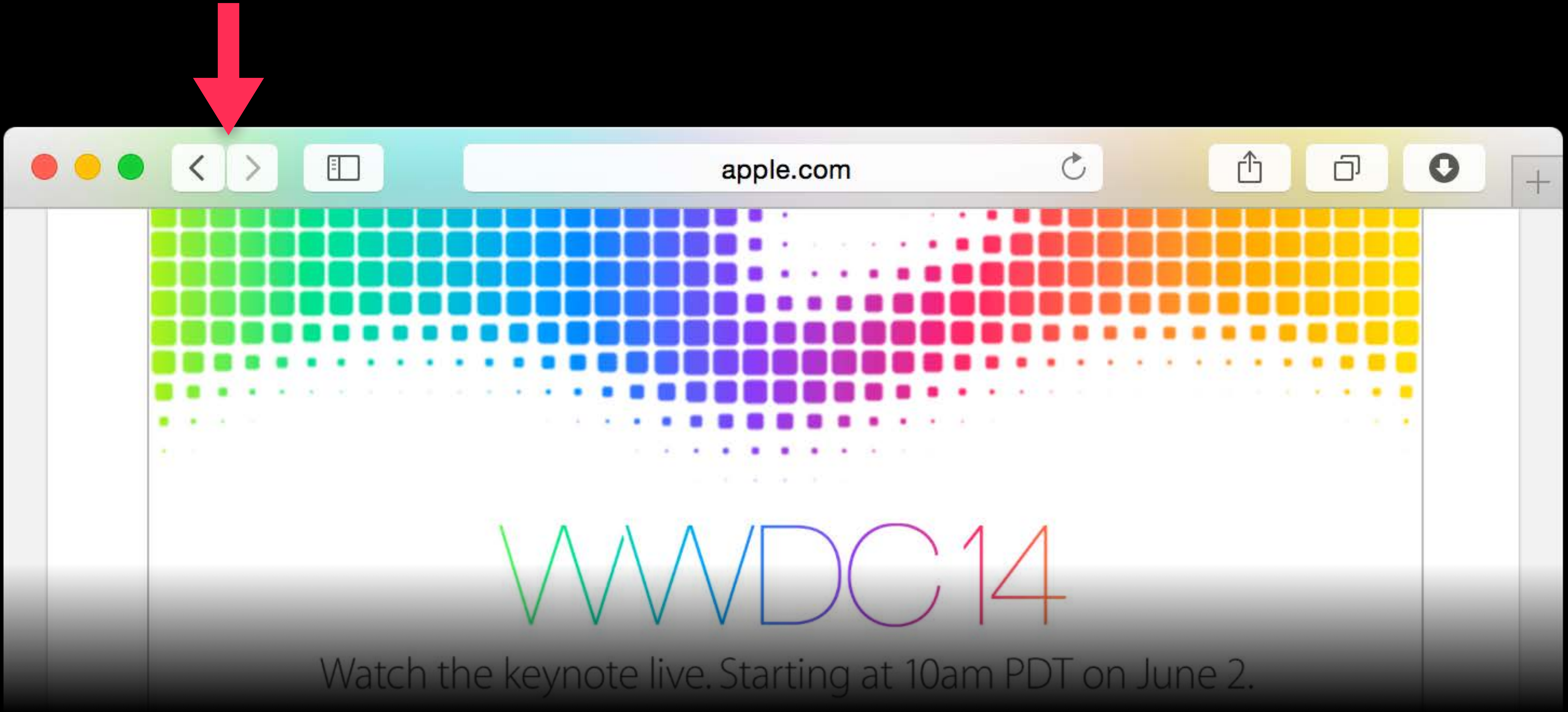
NSSegmentedControl

New style for back/forward buttons, NSSegmentStyleSeparated



NSSegmentedControl

New style for back/forward buttons, NSSegmentStyleSeparated



Related Sessions

• Adapting Your App to the New UI of OS X Yosemite	Pacific Heights	Tuesday 3:15PM
• Adopting Advanced Features of the New UI	Marina	Wednesday 2:00PM

Labs

Cocoa Lab	Frameworks Lab B	Tuesday 12:30PM
New UI and Cocoa Lab	Frameworks Lab B	Wednesday 3:15PM

Extensions

Extensions

Provide access to your app’s functionality and content in other apps

Extensions

Provide access to your app’s functionality and content in other apps

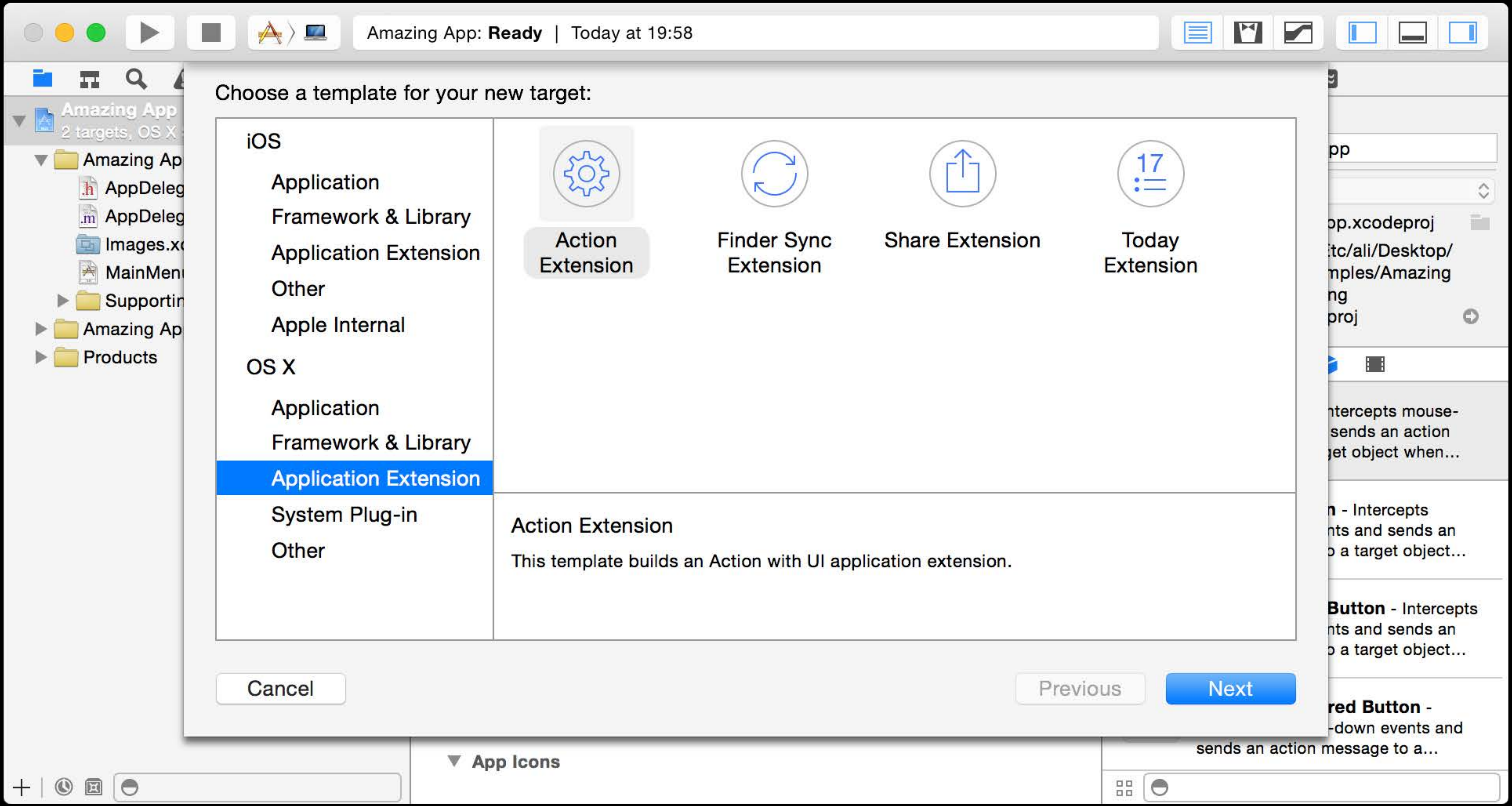
- Run in a separate process from the app in which they’re invoked

Extensions

Provide access to your app’s functionality and content in other apps

- Run in a separate process from the app in which they’re invoked

Are delivered with apps as distinct bundles within the app bundle



Amazing App: **Ready** | Today at 19:58

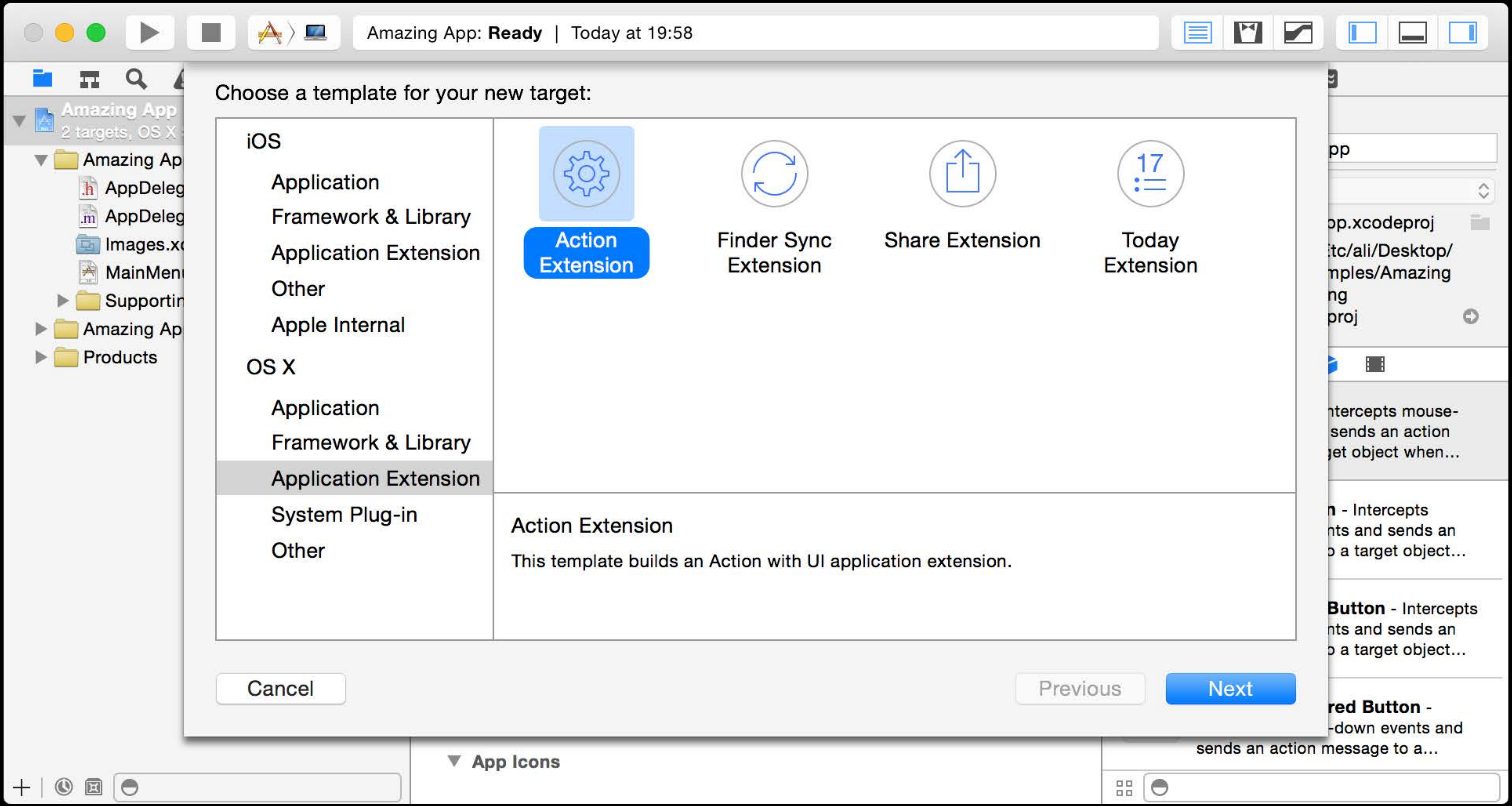
Choose a template for your new target:

iOS
- Application
- Framework & Library
- Application Extension
- Other
- Apple Internal

OS X
- Application
- Framework & Library
- Application Extension
- System Plug-in
- Other

Action Extension

This template builds an Action with UI application extension.



Choose a template for your new target:

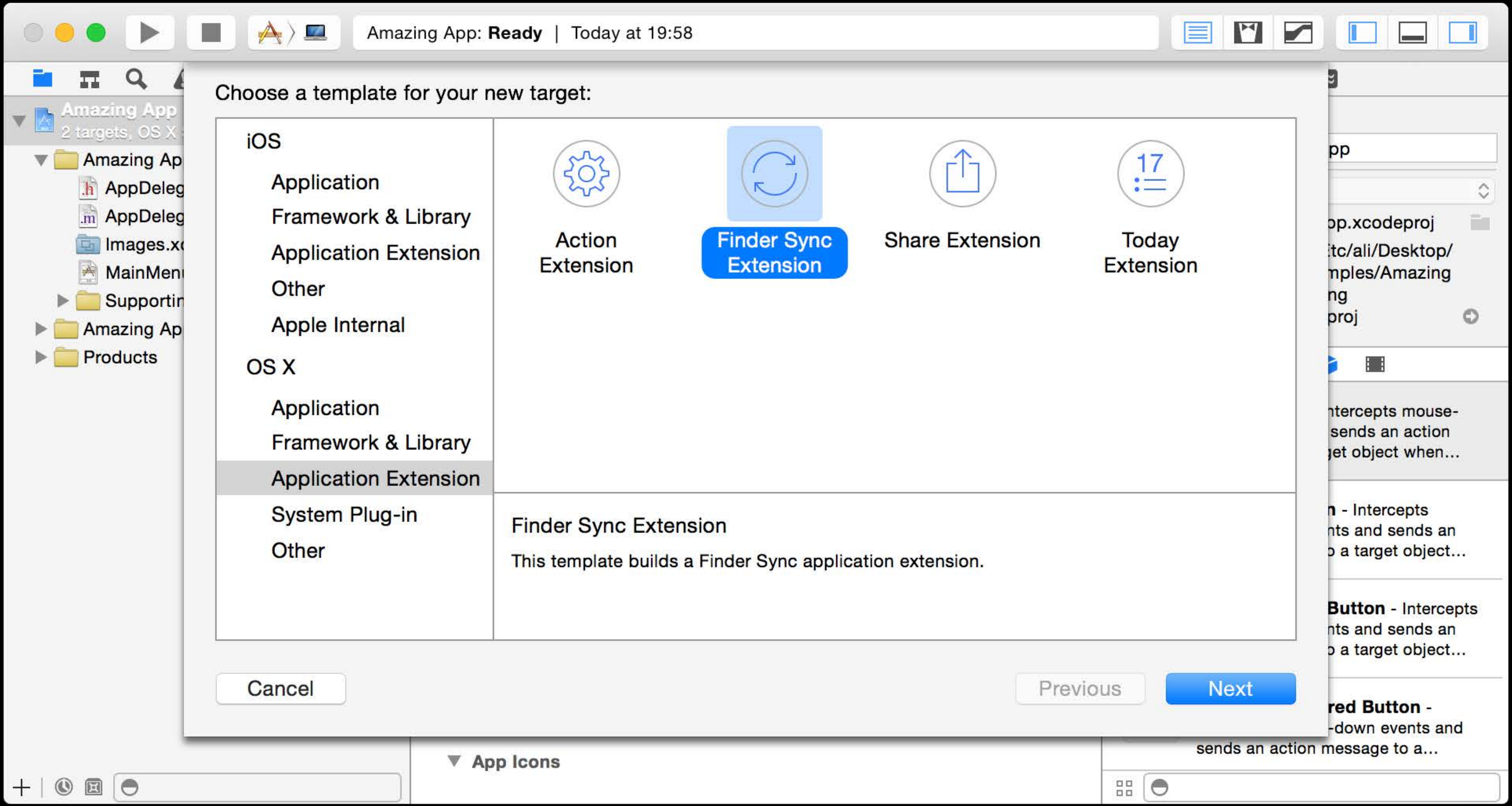
iOS
- Application
- Framework & Library
- Application Extension
- Other
- Apple Internal

OS X
- Application
- Framework & Library
- Application Extension
- System Plug-in
- Other

Action Extension | Finder Sync Extension | Share Extension | Today Extension

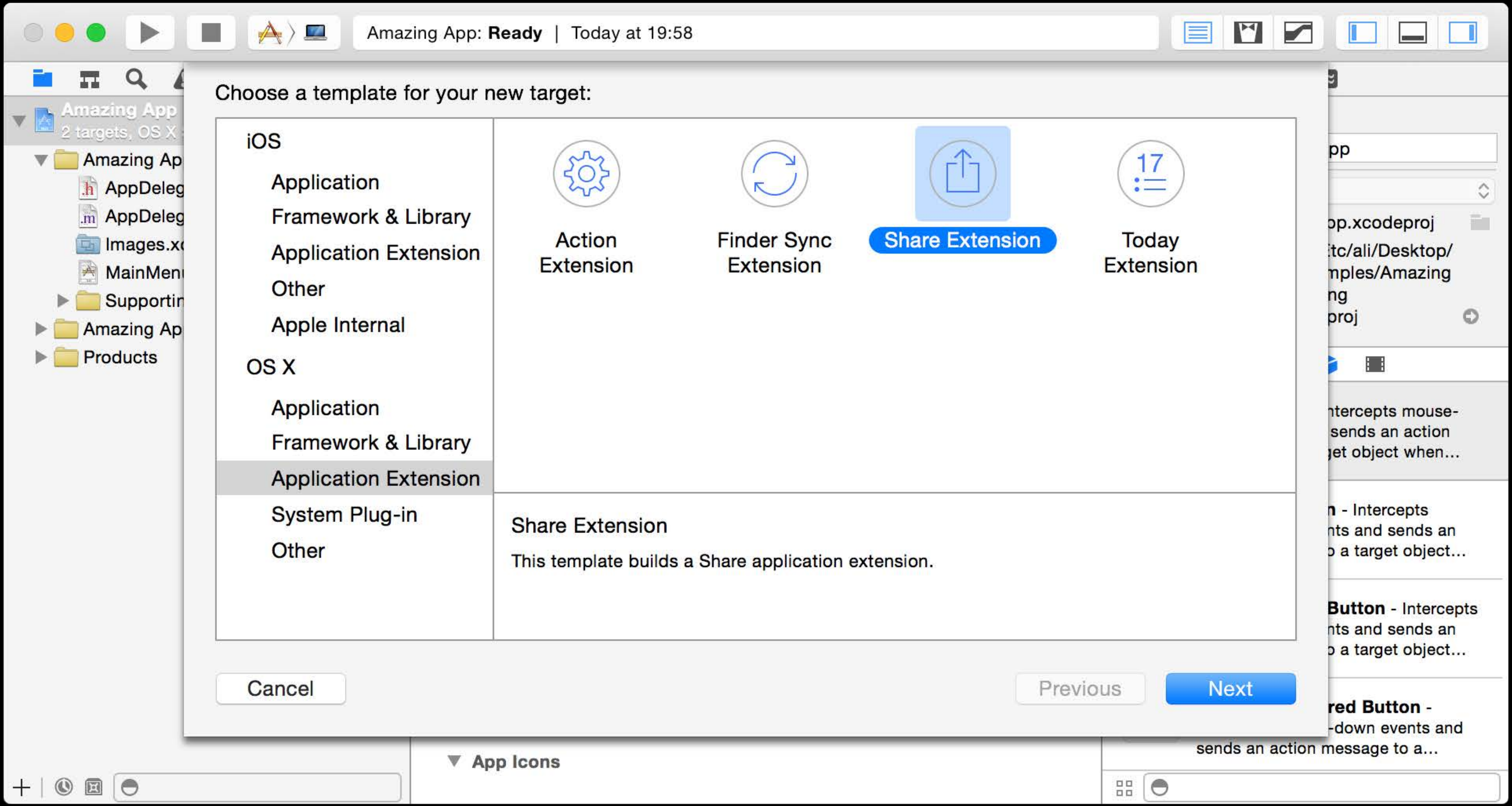
Action Extension

This template builds an Action with UI application extension.



Finder Sync Extension

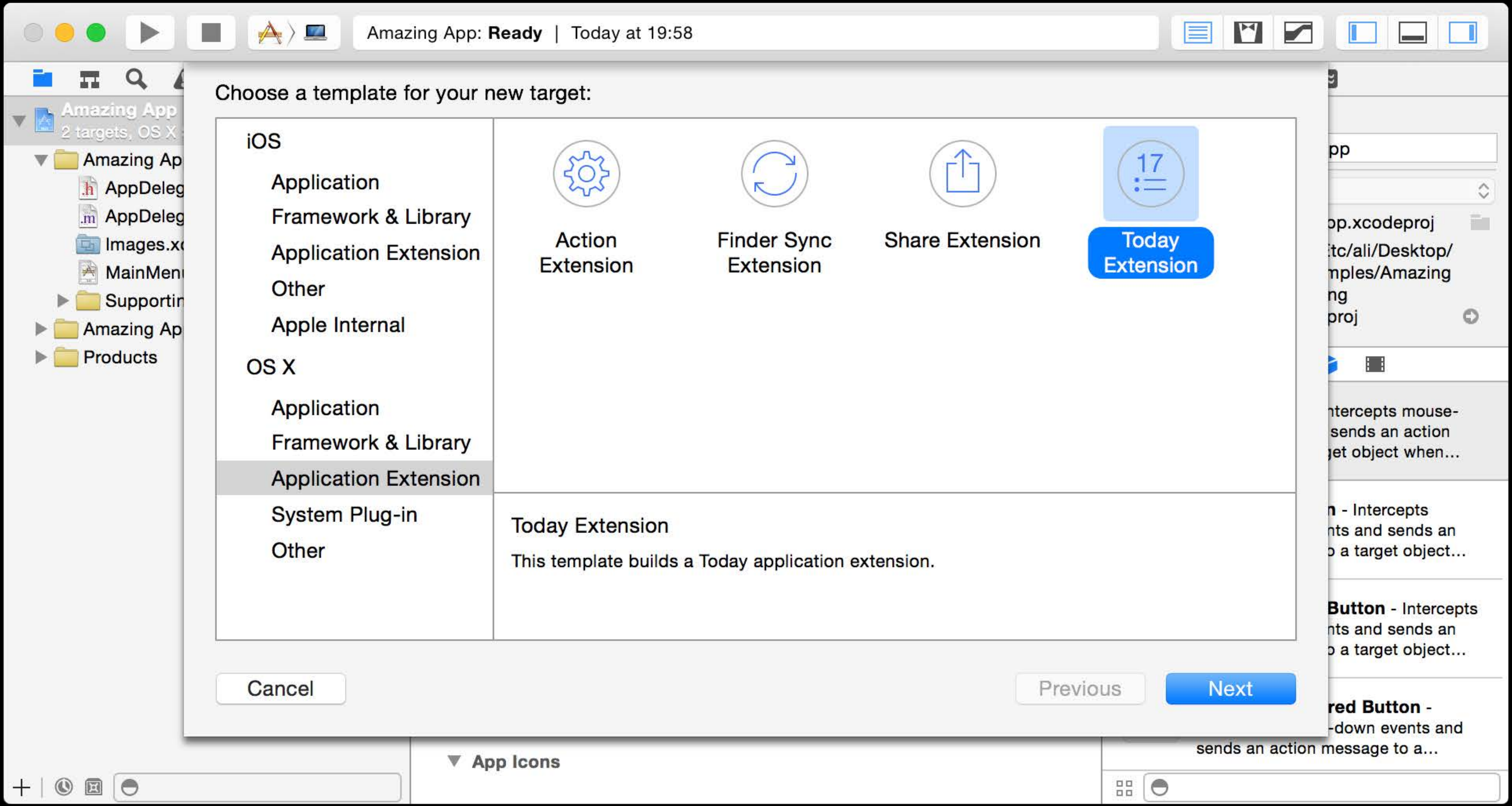
This template builds a Finder Sync application extension.


Amazing App: **Ready** | Today at 19:58

Choose a template for your new target:

Share Extension

This template builds a Share application extension.


Amazing App: **Ready** | Today at 19:58

Choose a template for your new target:

Today Extension

This template builds a Today application extension.

Extensions

NSExtensionContext

NSExtensionItem

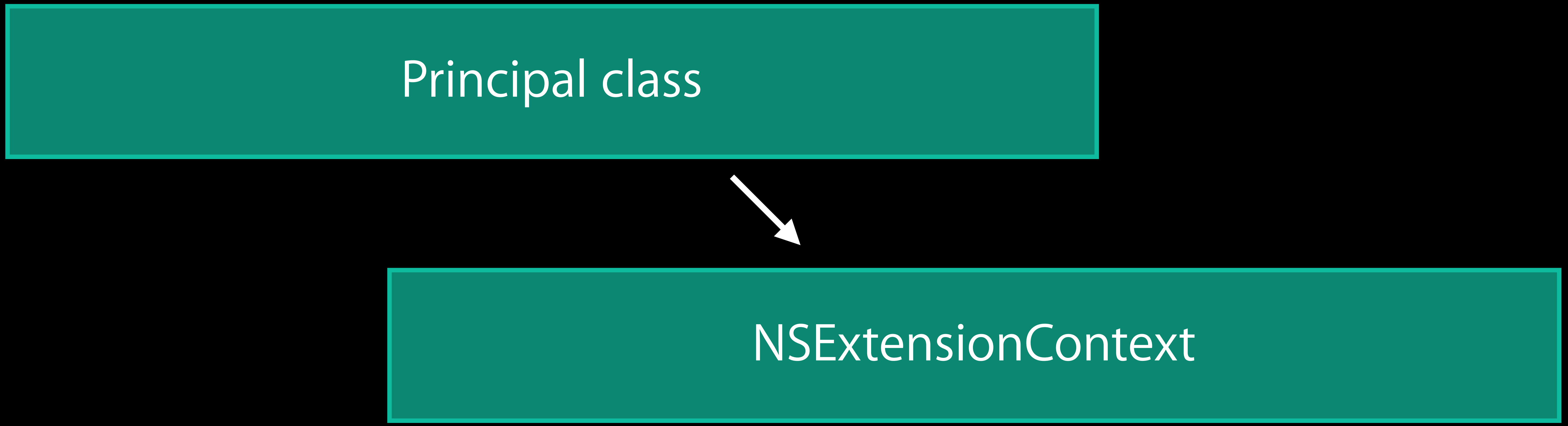
NSItemProvider

Inside the Extension

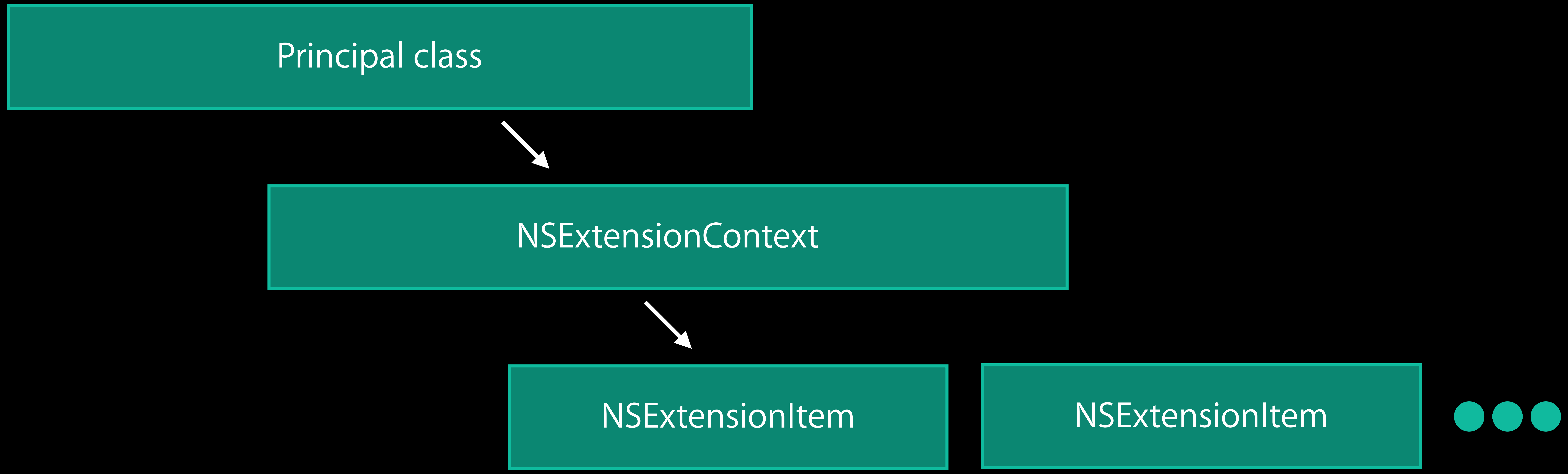
Inside the Extension

Principal class

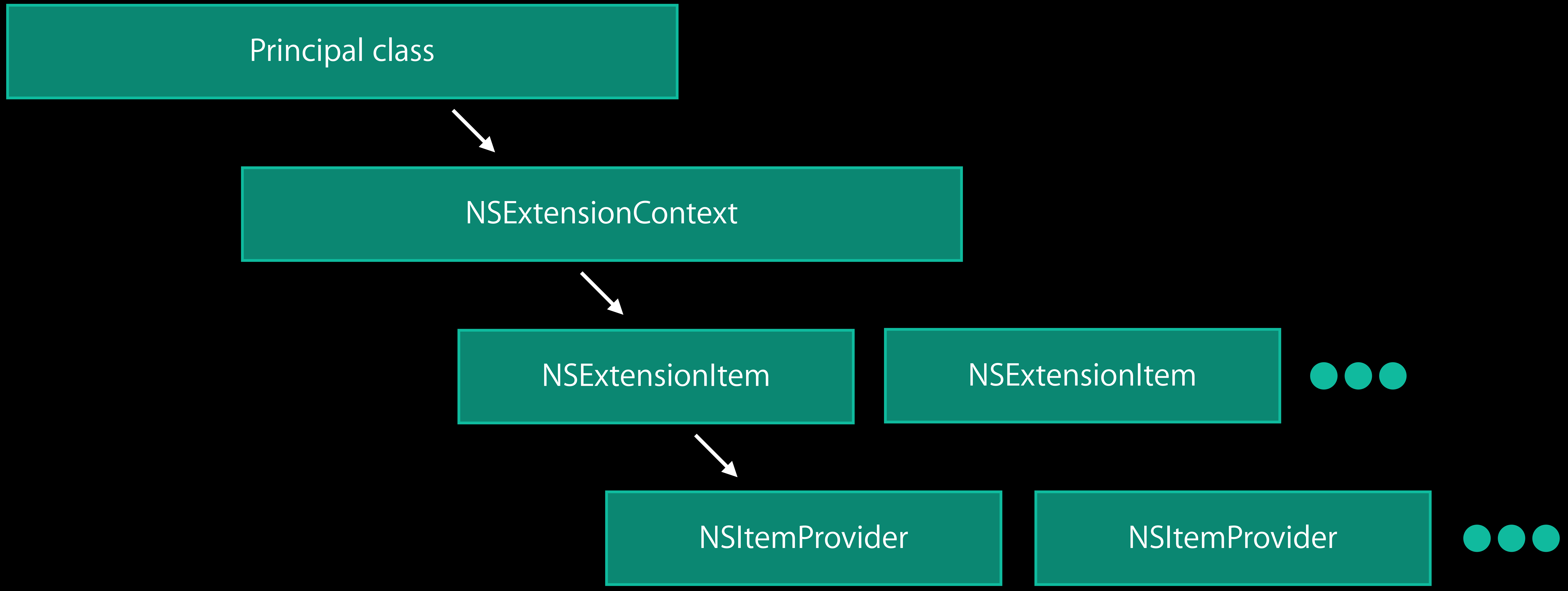
Inside the Extension



Inside the Extension



Inside the Extension



NSExtensionContext

NSExtensionContext

Get the data to be worked on from the instance of the principal class

```
    NSArray *extensionItems = self.extensionContext.inputItems;
```

NSExtensionContext

Get the data to be worked on from the instance of the principal class

```
NSArray *extensionItems = self.extensionContext.inputItems;
```

When done, return the results

```
NSArray *processedItems = << array of NSExtensionItem >> ;
[self.extensionContext completeRequestReturningItems:processedItems
                                   completionHandler:nil];
```

NSExtensionContext

Get the data to be worked on from the instance of the principal class

```
NSArray *extensionItems = self.extensionContext.inputItems;
```

When done, return the results

```
NSArray *processedItems = << array of NSExtensionItem >> ;
[self.extensionContext completeRequestReturningItems:processedItems
                                   completionHandler:nil];
```

Or indicate error/cancellation

```
[self.extensionContext cancelRequestWithError:error];
```

Related Sessions

Creating Extensions for iOS and OS X, Part 1	Mission	Tuesday 2:00PM
Creating Extensions for iOS and OS X, Part 2	Mission	Wednesday 11:30AM

Labs

● Extensions Lab	Frameworks Lab A	Tuesday 3:15PM
● Extensions Lab	Frameworks Lab B	Thursday 2:00PM

Handoff

Handoff

Enables users to seamlessly transition activities between devices

Handoff

Enables users to seamlessly transition activities between devices

Simple base API

- NSUserActivity

Handoff

Enables users to seamlessly transition activities between devices

Simple base API

- NSUserActivity

Related API

- NSApplication
- NSDocument
- NSResponder

NSUserActivity

Encapsulates hand-off information about a single user activity

NSUserActivity

Encapsulates hand-off information about a single user activity

```
NSUserActivity *activity = [[NSUserActivity alloc]
                  initWithActivityType:@"com.company.somegame.playing"];
```

NSUserActivity

Encapsulates hand-off information about a single user activity

```
NSUserActivity *activity = [[NSUserActivity alloc]
                  initWithActivityType:@"com.company.somegame.playing"];
activity.userInfo = @{@"Level":@1, @"Location":@"Start", @"Score":@0};
```

NSUserActivity

Encapsulates hand-off information about a single user activity

```
NSUserActivity *activity = [[NSUserActivity alloc]
                   initWithActivityType:@"com.company.somegame.playing"];
activity.userInfo = @{@"Level":@1, @"Location":@"Start", @"Score":@0};
activity.title = @"Playing Some Game”;
```

NSUserActivity

Encapsulates hand-off information about a single user activity

```
NSUserActivity *activity = [[NSUserActivity alloc]
                   initWithActivityType:@"com.company.somegame.playing"];
activity.userInfo = @{@"Level":@1, @"Location":@"Start", @"Score":@0};
activity.title = @"Playing Some Game”;

[activity becomeCurrent];
```

NSApplication

Controls continuation of user activities

```
- (BOOL)application:(NSApplication *)application
        willContinueUserActivityWithType:(NSString *)type;

- (BOOL)application:(NSApplication *)application
        continueUserActivity:(NSUserActivity *)userActivity
        restorationHandler:(void(^)(NSArray *restorableObjects))handler;
```

NSDocument

Easy handoff support for iCloud documents

Add to Info.plist

```
<key>CFBundleDocumentTypes</key>
<array>
   <dict>
      …
      <key>NSUbiquitousDocumentUserActivityType</key>
      <string>com.apple.TextEdit.editing</string>
   </dict>
</array>
```

Access the userActivity object via

```
@property (strong) NSUserActivity *userActivity;
```

Related Sessions

● Adopting Handoff on iOS and OS X	Mission	Wednesday 2:00PM

Labs

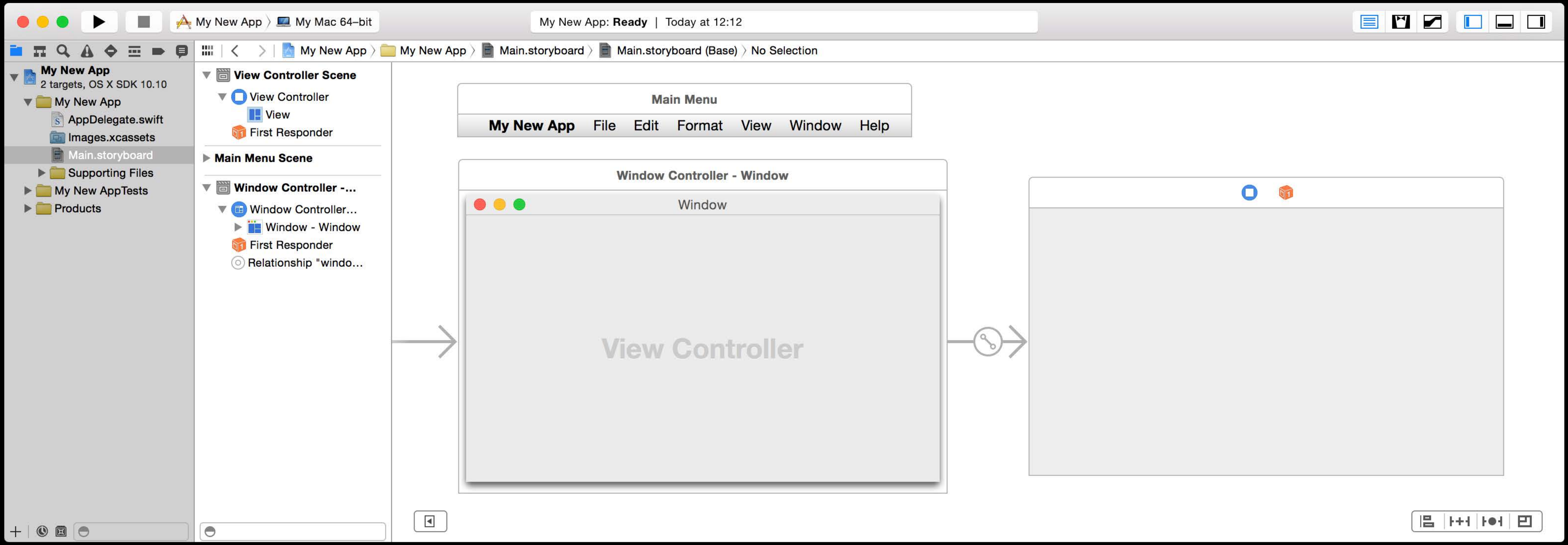
● Handoff Lab	Frameworks Lab B	Thursday 9:00AM

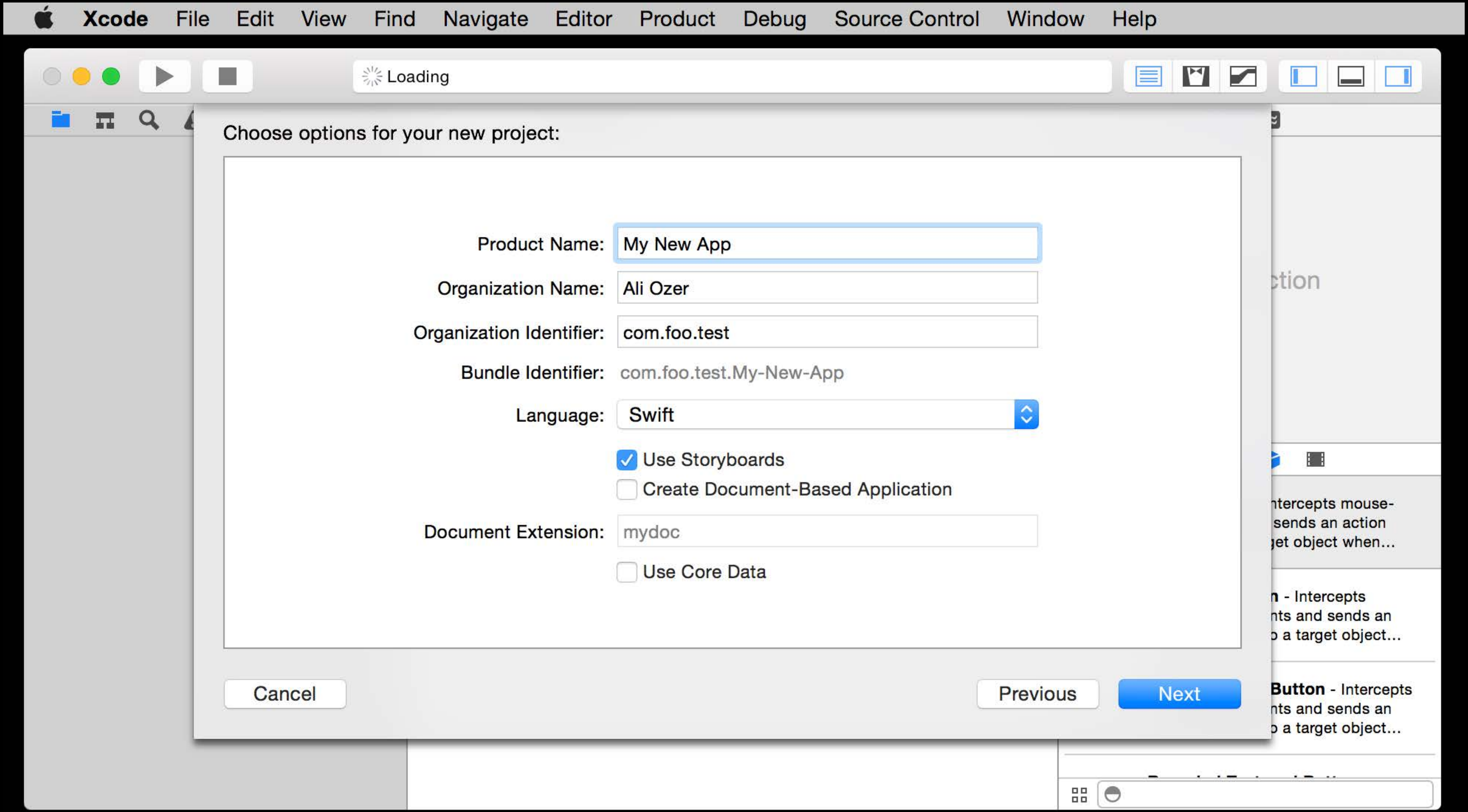
Storyboards and View Controllers

Storyboard

Now on OS X

Visual representation of the user interface




Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Loading

Choose options for your new project:

Product Name: My New App

Organization Name: Ali Ozer

Organization Identifier: com.foo.test

Bundle Identifier: com.foo.test.My-New-App

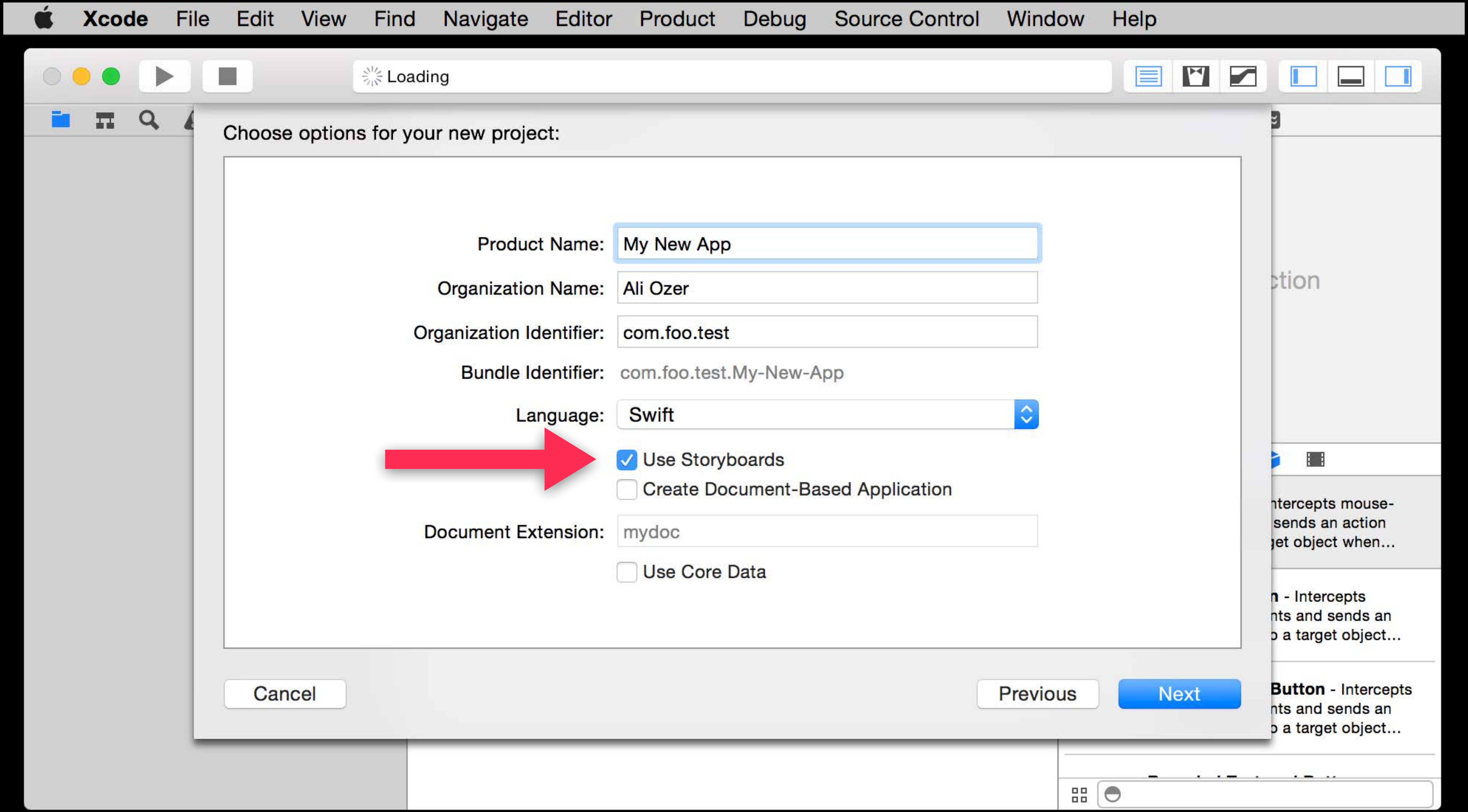
Language: Swift

[✓] Use Storyboards

[ ] Create Document-Based Application

Document Extension: mydoc

[ ] Use Core Data


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Loading
Cancel
Previous
Next

Choose options for your new project:

Product Name: My New App
Organization Name: Ali Ozer
Organization Identifier: com.foo.test
Bundle Identifier: com.foo.test.My-New-App
Language: Swift
[✓] Use Storyboards
[ ] Create Document-Based Application
Document Extension: mydoc
[ ] Use Core Data

Storyboard

Specify the different parts of your UI as different scenes

Use segues to connect or transition between them

Storyboard

New classes

- NSStoryboard
- NSStoryboardSegue

Storyboard

New classes

- NSStoryboard
- NSStoryboardSegue

New protocol

- NSSeguePerforming

Storyboard

New classes

- NSStoryboard
- NSStoryboardSegue

New protocol

- NSSeguePerforming

New APIs on NSViewController, NSWindowController

- NSSeguePerforming conformance
- Access to storyboard

New View Controllers

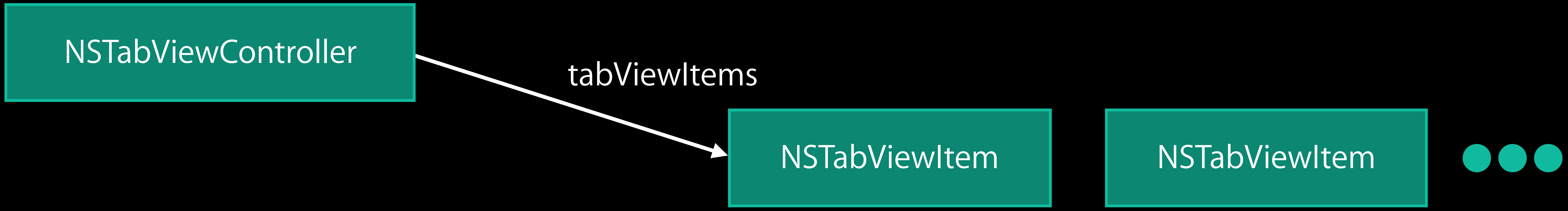
NSTabViewController

NSSplitViewController

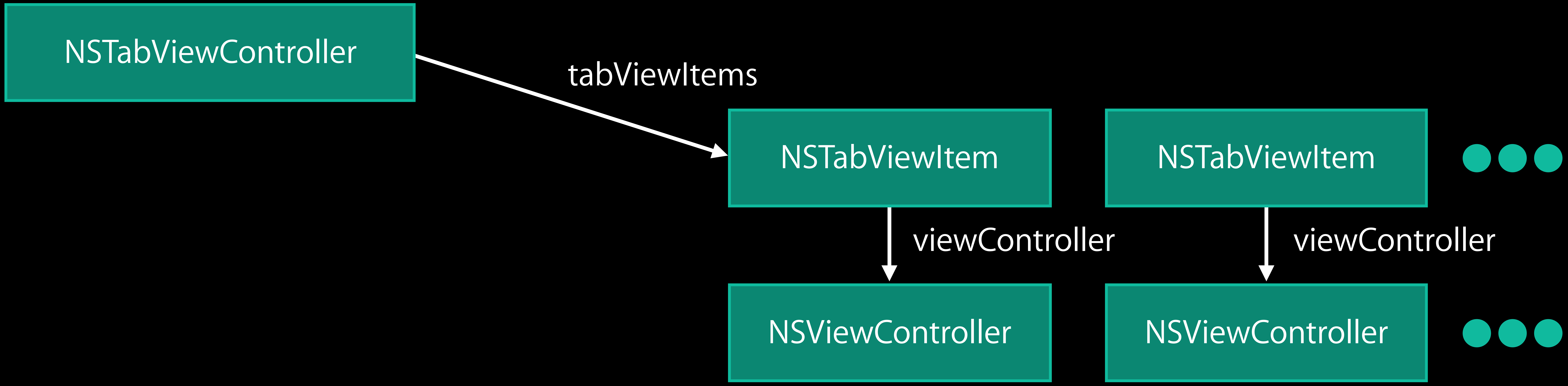
NSTabViewController

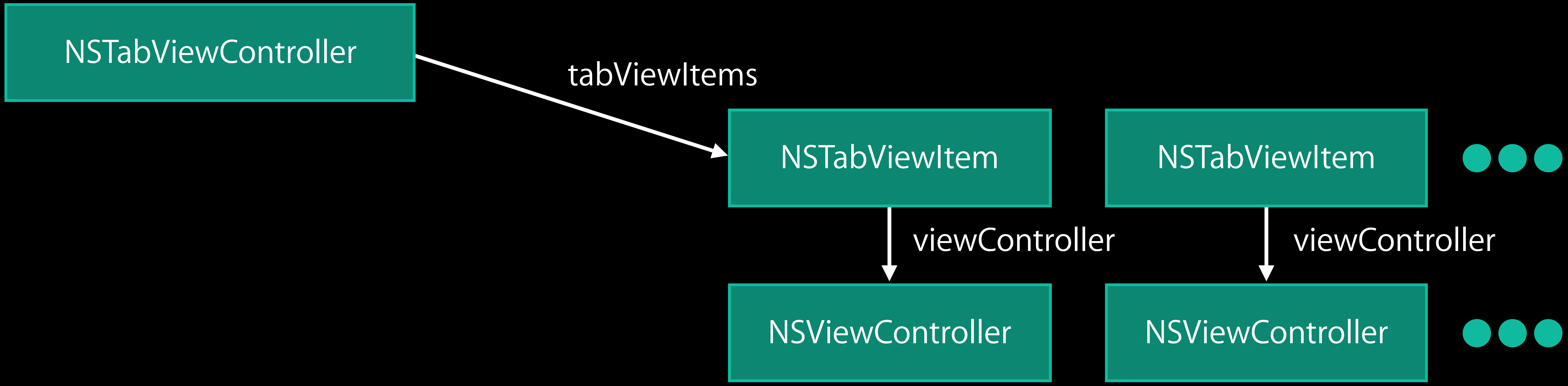
NSTabViewController

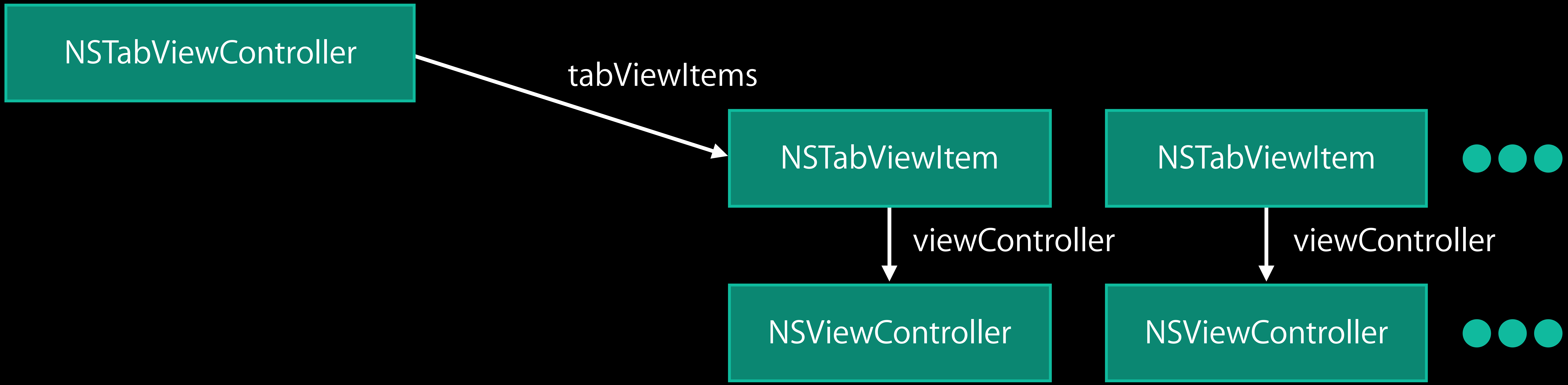
NSTabViewController

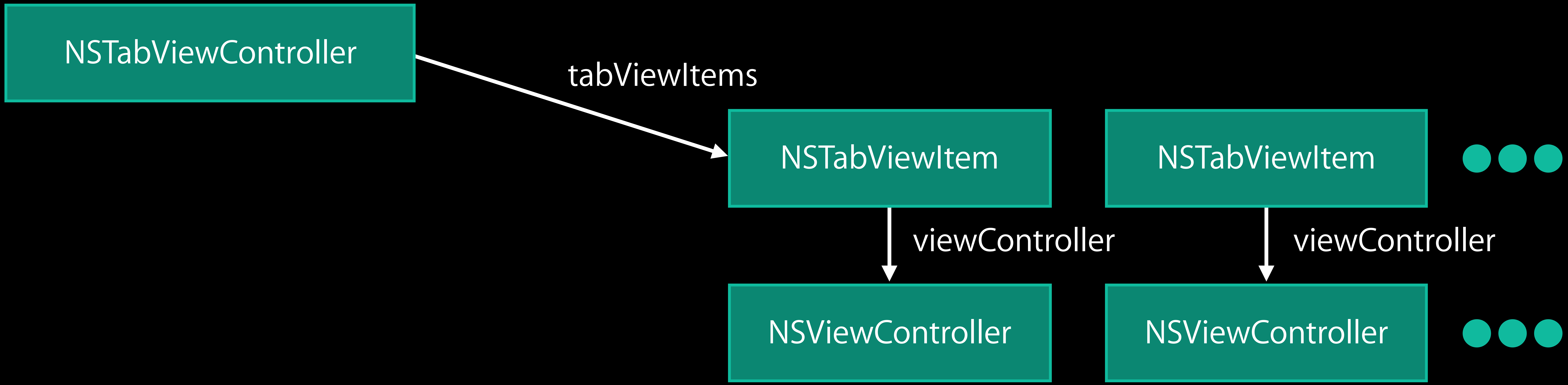


NSTabViewController

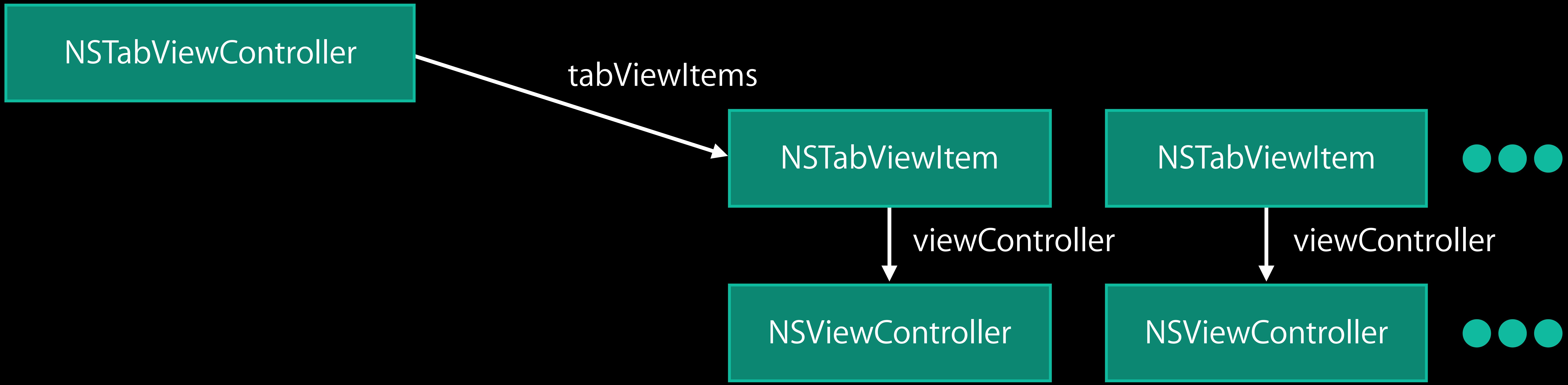



NSTabViewController
tabViewItems
NSTabViewItem
NSTabViewItem
viewController
viewController
NSViewController
NSViewController


NSTabViewController
tabViewItems
NSTabViewItem
NSTabViewItem
viewController
viewController
NSViewController
NSViewController

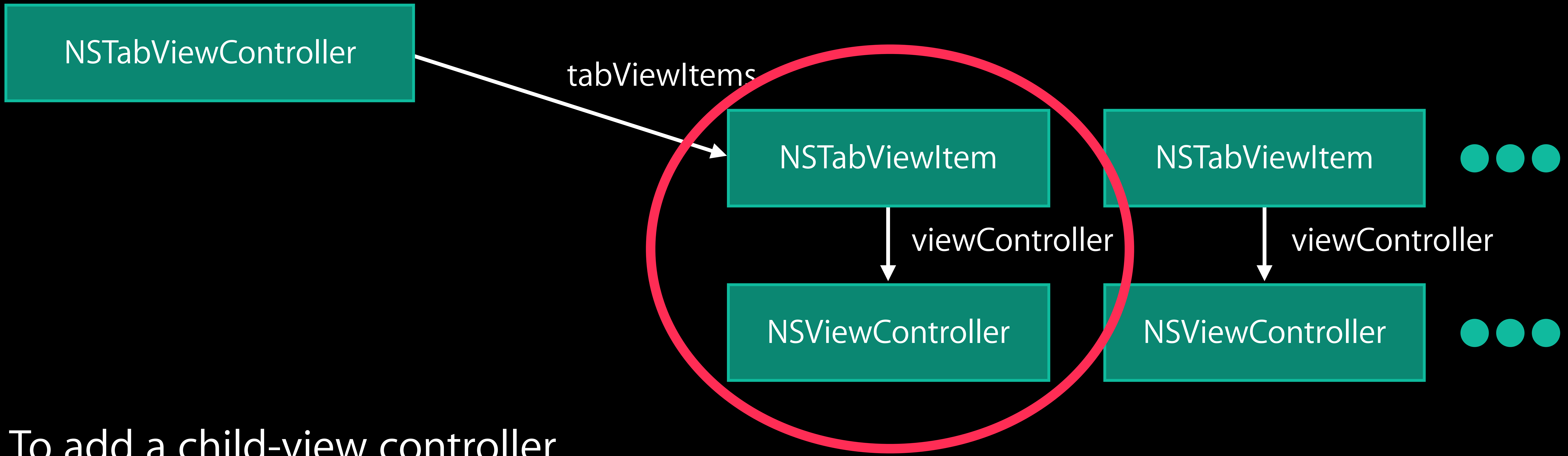


To add a child-view controller



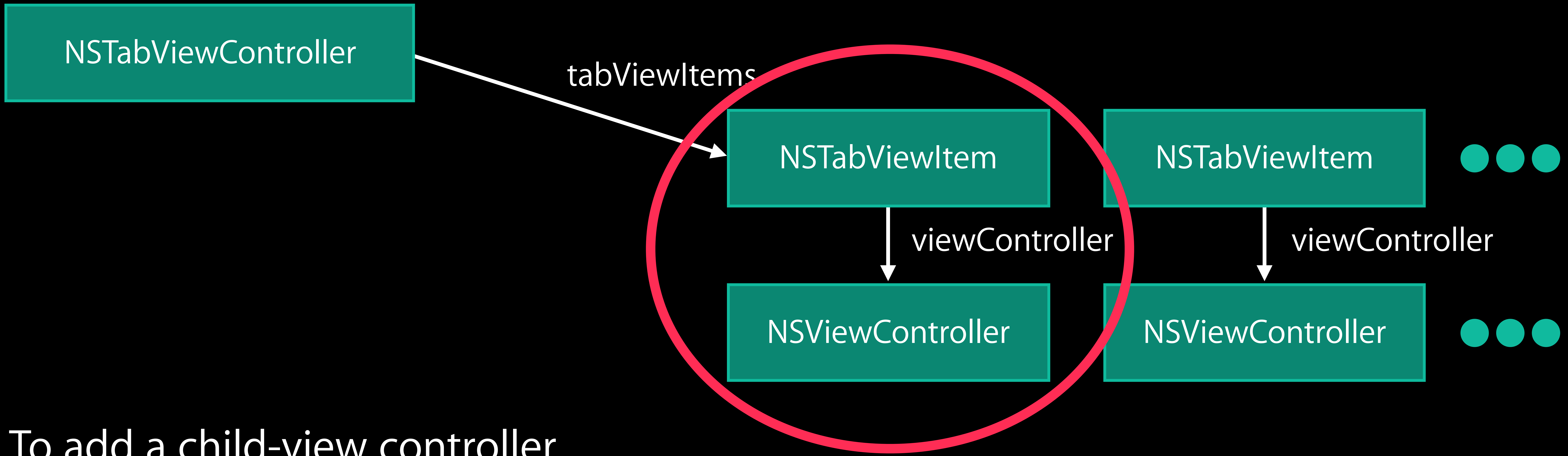
To add a child-view controller

```
    NSTabViewItem *item = [NSTabViewItem tabViewItemWithViewController:child];
```



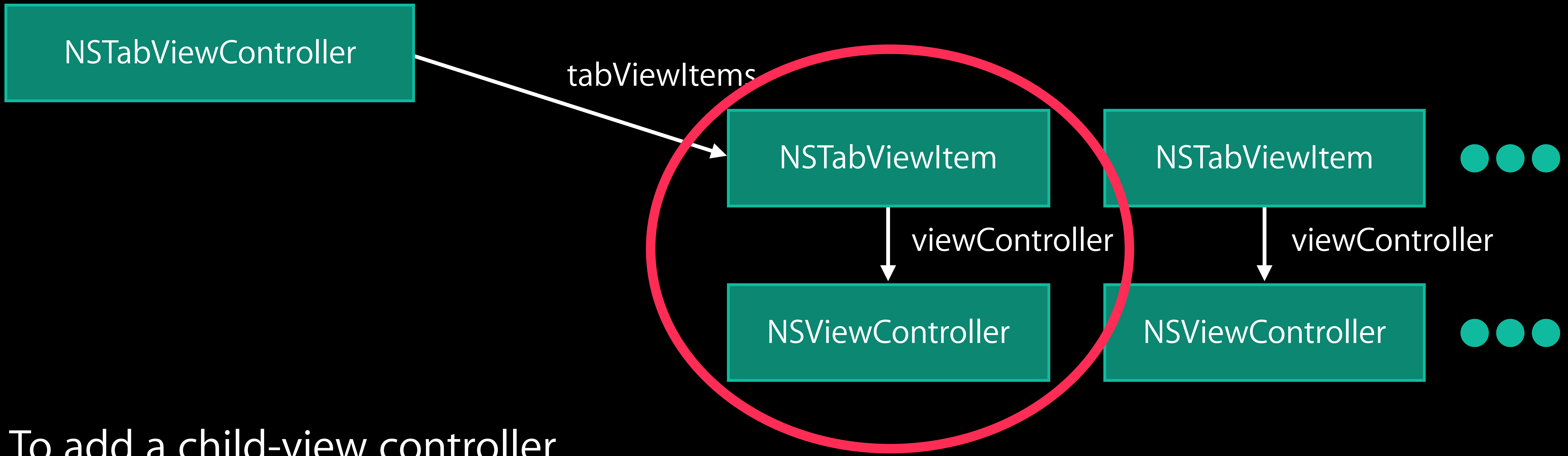
To add a child-view controller

```
    NSTabViewItem *item = [NSTabViewItem tabViewItemWithViewController:child];
```



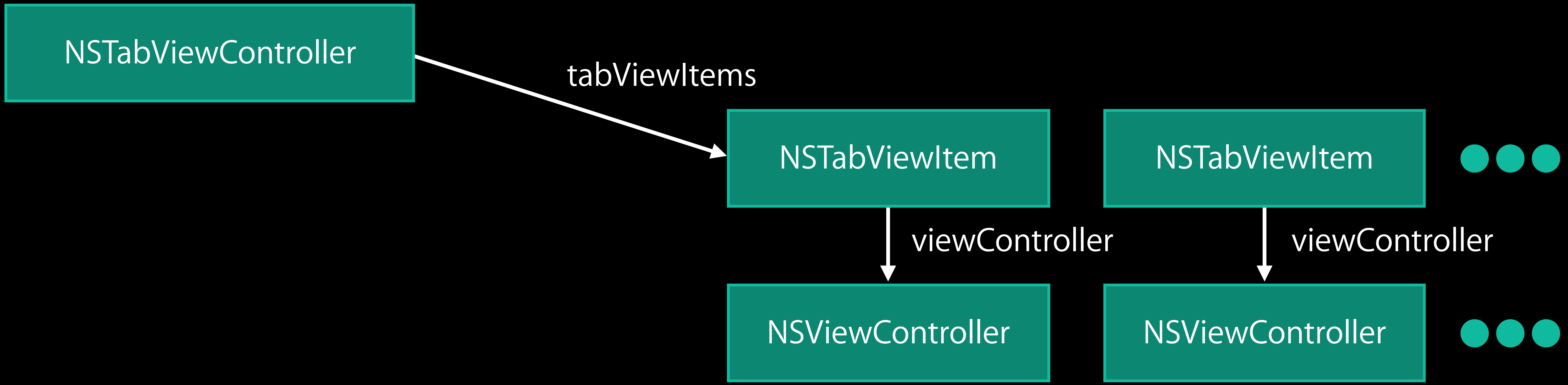
To add a child-view controller

```
NSTabViewItem *item = [NSTabViewItem tabViewItemWithViewController:child];
item.label = @“Tab Label”;   // Configure item as needed
```



To add a child-view controller

```
NSTabViewItem *item = [NSTabViewItem tabViewItemWithViewController:child];
item.label = @“Tab Label”;   // Configure item as needed
[tabViewController addTabViewItem:item];
```

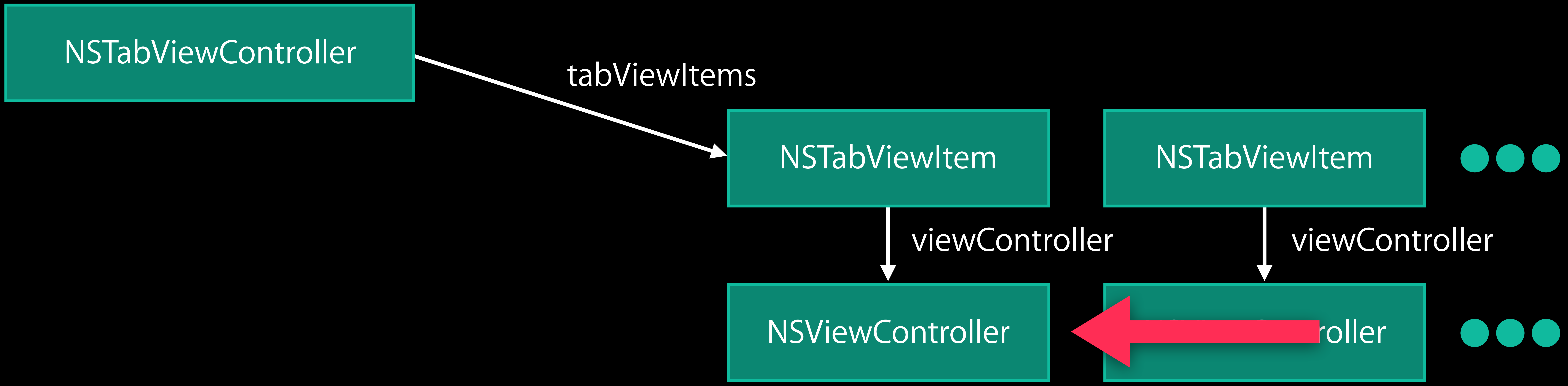


To add a child-view controller

```
NSTabViewItem *item = [NSTabViewItem tabViewItemWithViewController:child];
item.label = @“Tab Label”;   // Configure item as needed
[tabViewController addTabViewItem:item];
```

or

```
[tabViewController addChildViewController:child];
```

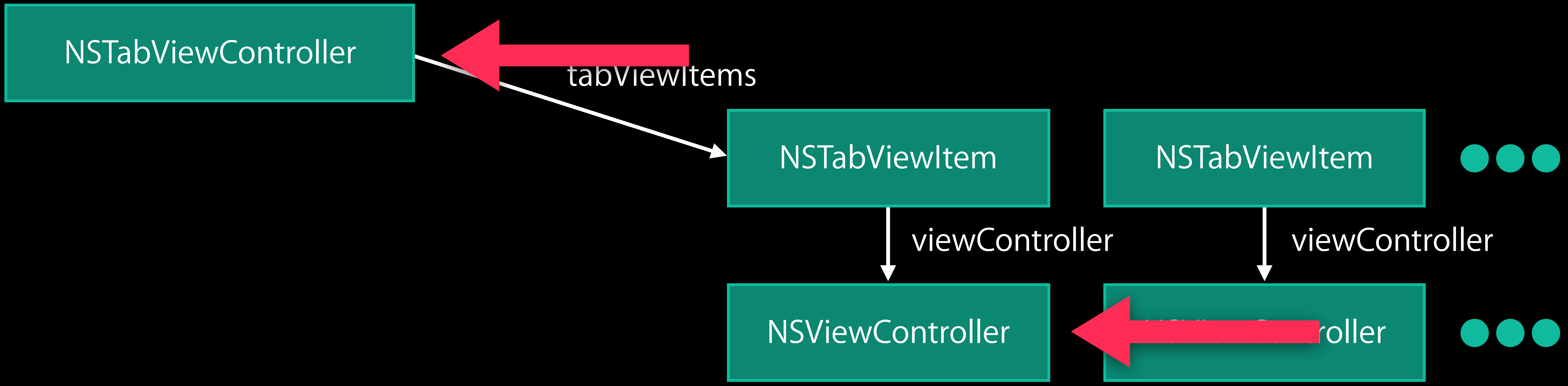


To add a child-view controller

```
NSTabViewItem *item = [NSTabViewItem tabViewItemWithViewController:child];
item.label = @“Tab Label”;   // Configure item as needed
[tabViewController addTabViewItem:item];
```

or

```
[tabViewController addChildViewController:child];
```

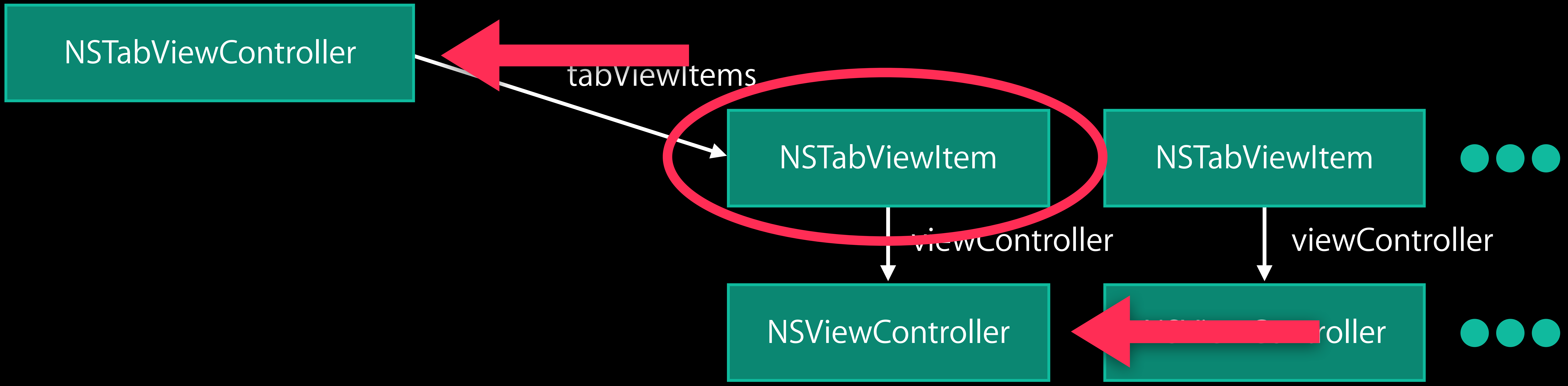


To add a child-view controller

```
NSTabViewItem *item = [NSTabViewItem tabViewItemWithViewController:child];
item.label = @“Tab Label”;   // Configure item as needed
[tabViewController addTabViewItem:item];
```

or

```
[tabViewController addChildViewController:child];
```

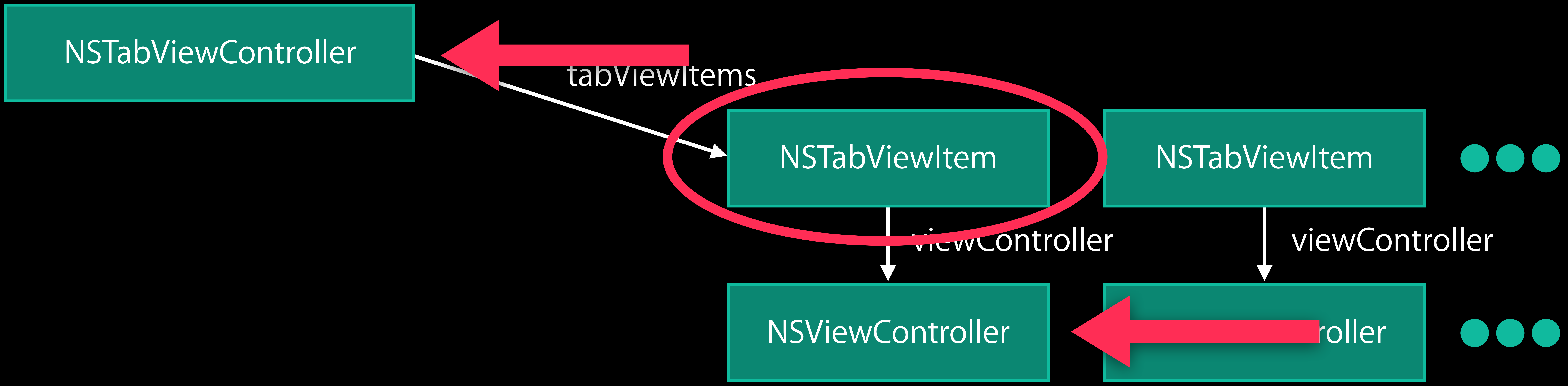


To add a child-view controller

```
    NSTabViewItem *item = [NSTabViewItem tabViewItemWithViewController:child];
    item.label = @“Tab Label”;   // Configure item as needed
    [tabViewController addTabViewItem:item];
```

or

```
    [tabViewController addChildViewController:child];
```



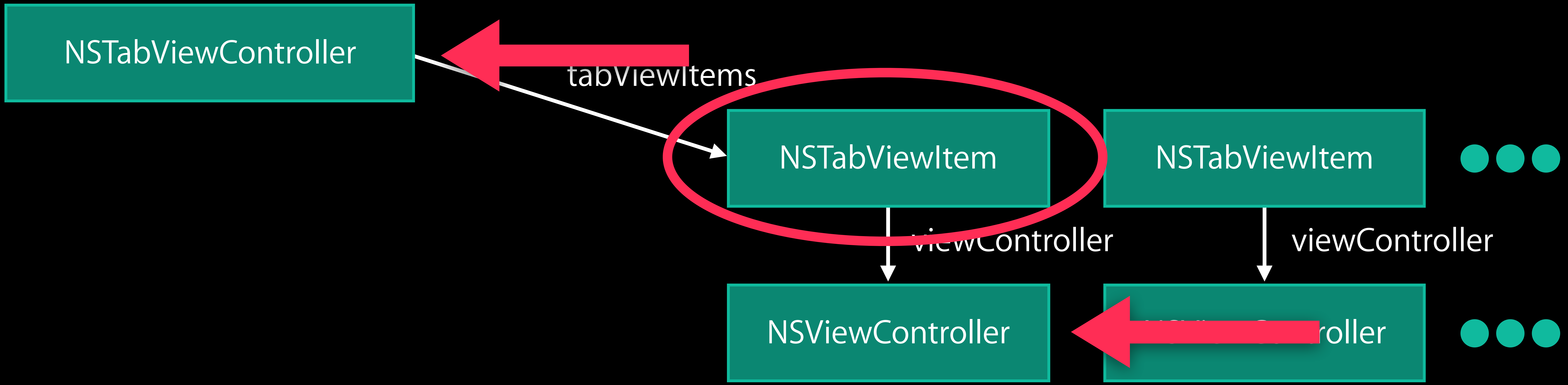
To add a child-view controller

```
NSTabViewItem *item = [NSTabViewItem tabViewItemWithViewController:child];
item.label = @“Tab Label”;   // Configure item as needed
[tabViewController addTabViewItem:item];
```

or

```
[tabViewController addChildViewController:child];

// If needed, to configure the child tabViewItem further:
item = [tabViewController tabViewItemForViewController:child];
```



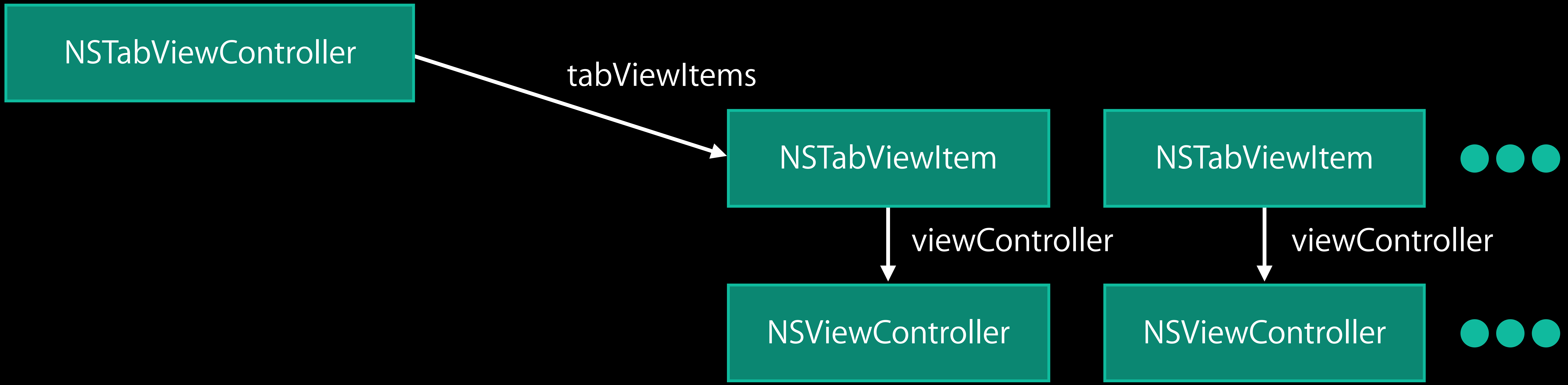
To add a child-view controller

```
NSTabViewItem *item = [NSTabViewItem tabViewItemWithViewController:child];
item.label = @“Tab Label”;   // Configure item as needed
[tabViewController addTabViewItem:item];
```

or

```
[tabViewController addChildViewController:child];

// If needed, to configure the child tabViewItem further:
item = [tabViewController tabViewItemForViewController:child];
item.label = @“Tab Label”;
```



To add a child-view controller

```
NSTabViewItem *item = [NSTabViewItem tabViewItemWithViewController:child];
item.label = @"Tab Label";   // Configure item as needed
[tabViewController addTabViewItem:item];
```

or

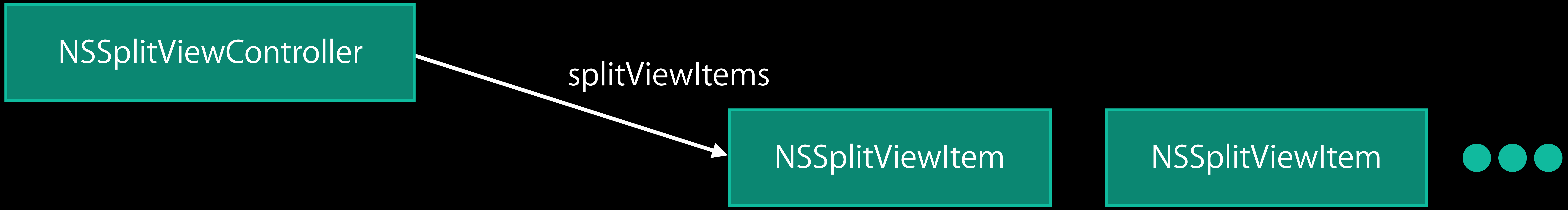
```
[tabViewController addChildViewController:child];

// If needed, to configure the child tabViewItem further:
item = [tabViewController tabViewItemForViewController:child];
item.label = @"Tab Label";
```

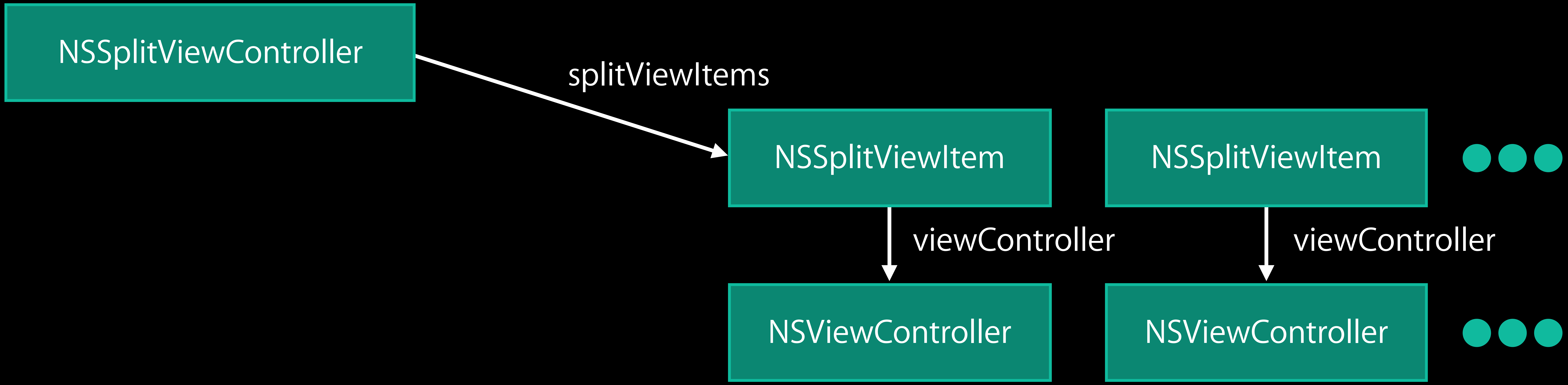
NSSplitViewController

NSSplitViewController

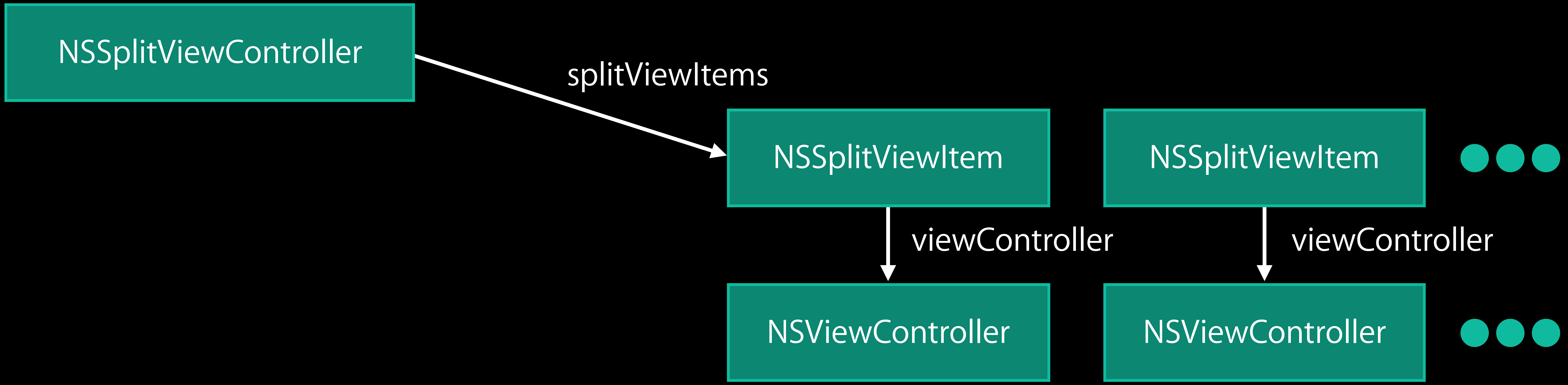
NSSplitViewController



NSSplitViewController



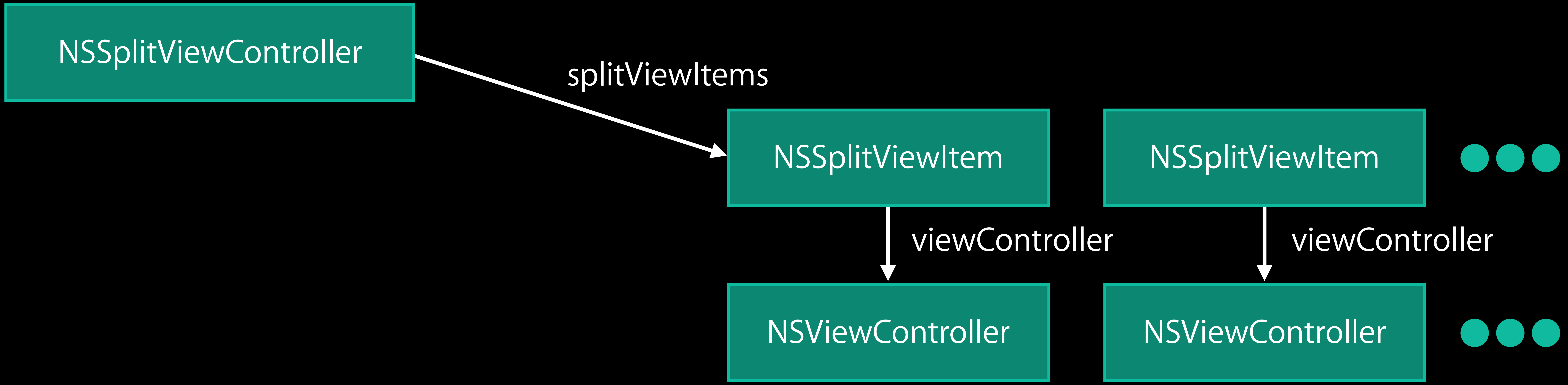
NSSplitViewController



To add a child-view controller

```
NSSplitViewItem *item = [NSSplitViewItem
                            splitViewItemWithViewController:child];
item.canCollapse = YES;   // initialize item as needed
[splitViewController addSplitViewItem:item];
```

NSSplitViewController



To add a child-view controller

```
NSSplitViewItem *item = [NSSplitViewItem
                           splitViewItemWithViewController:child];
item.canCollapse = YES;   // initialize item as needed
[splitViewController addSplitViewItem:item];
```

or

```
[splitViewController addChildViewController:child];
```

View Controller Presentation

View Controller Presentation

```
- (void)presentViewControllerAsSheet:(NSViewController *)controller;
```

View Controller Presentation

```
- (void)presentViewControllerAsSheet:(NSViewController *)controller;

- (void)presentViewControllerAsModalWindow:(NSViewController *)controller;
```

View Controller Presentation

```
- (void)presentViewControllerAsSheet:(NSViewController *)controller;

- (void)presentViewControllerAsModalWindow:(NSViewController *)controller;

- (void)presentViewController:(NSViewController *)controller
      asPopoverRelativeToRect:(NSRect)positioningRect
                       ofView:(NSView *)positioningView
                preferredEdge:(NSRectEdge)preferredEdge
                     behavior:(NSPopoverBehavior)behavior;
```

View Controller Presentation

```
- (void)presentViewControllerAsSheet:(NSViewController *)controller;

- (void)presentViewControllerAsModalWindow:(NSViewController *)controller;

- (void)presentViewController:(NSViewController *)controller
     asPopoverRelativeToRect:(NSRect)positioningRect
                      ofView:(NSView *)positioningView
               preferredEdge:(NSRectEdge)preferredEdge
                    behavior:(NSPopoverBehavior)behavior;

- (void)dismissViewController:(NSViewController *)controller;
```

View Controller Presentation

```
- (void)transitionFromViewController:(NSViewController *)source
                    toViewController:(NSViewController *)dest
                             options:(NSViewControllerTransitionOptions)opts
                   completionHandler:(void (^)(void))completion;
```

View Controller Presentation

```
- (void)viewDidLoad;
```

View Controller Presentation

```
- (void)viewDidLoad;

- (void)viewWillAppear;
- (void)viewDidAppear;
- (void)viewWillDisappear;
- (void)viewDidDisappear;
```

View Controller Presentation

```
- (void)viewDidLoad;

- (void)viewWillAppear;
- (void)viewDidAppear;
- (void)viewWillDisappear;
- (void)viewDidDisappear;

- (void)viewWillLayout;
- (void)viewDidLayout;
```

View Controller

View Controller

Now automatically added into the responder chain

- View controller’s next responder is set to that of its view
- View controller becomes the next responder of the view

View Controller

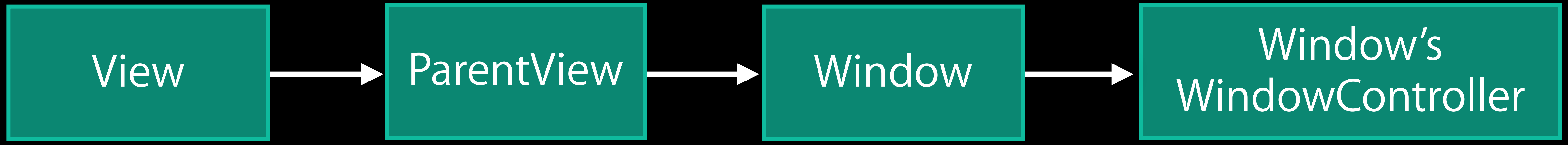
Now automatically added into the responder chain

- View controller's next responder is set to that of its view
- View controller becomes the next responder of the view

View Controller

Now automatically added into the responder chain

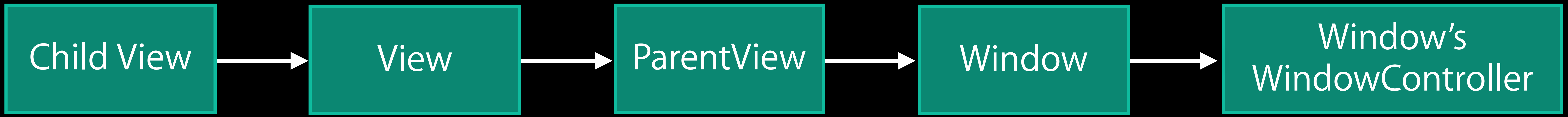
- View controller’s next responder is set to that of its view
- View controller becomes the next responder of the view


$$\begin{tikzcd}\text{View} \arrow[r] & \text{ParentView} \arrow[r] & \text{Window} \arrow[r] & \text{Window's WindowController}\end{tikzcd}$$

View Controller

Now automatically added into the responder chain

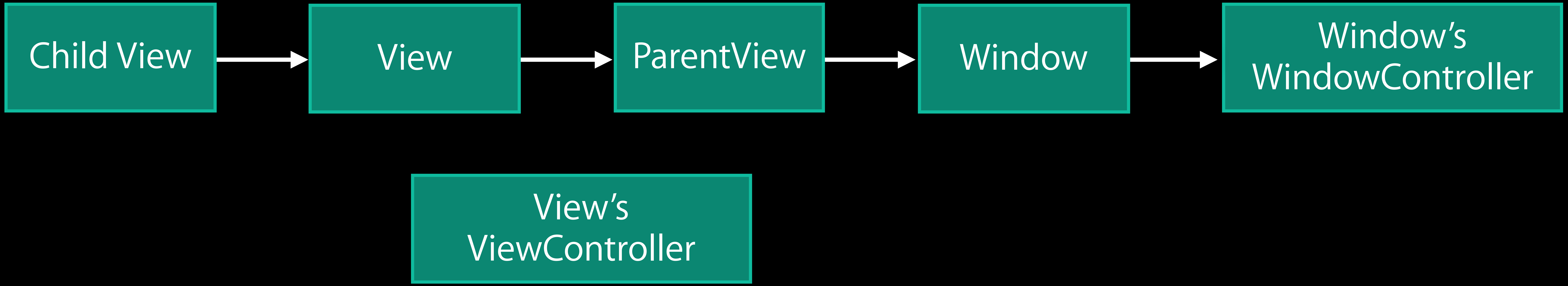
- View controller's next responder is set to that of its view
- View controller becomes the next responder of the view



View Controller

Now automatically added into the responder chain

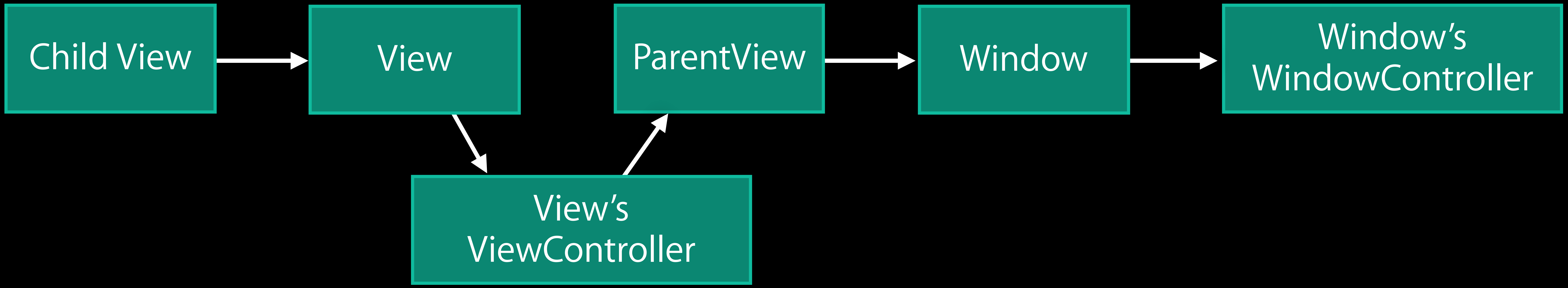
- View controller’s next responder is set to that of its view
- View controller becomes the next responder of the view



View Controller

Now automatically added into the responder chain

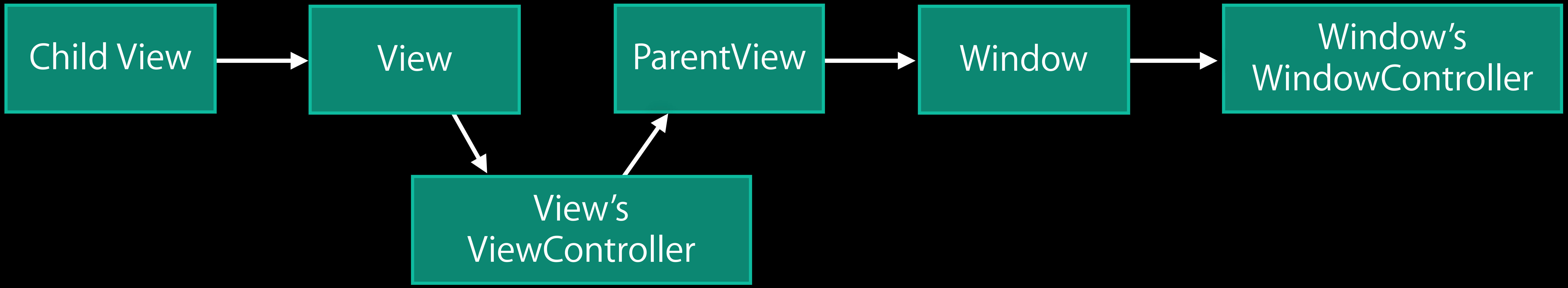
- View controller’s next responder is set to that of its view
- View controller becomes the next responder of the view



View Controller

Now automatically added into the responder chain

- View controller's next responder is set to that of its view
- View controller becomes the next responder of the view



This change is effective only in apps built against the 10.10 SDK

Related Sessions

● Storyboards and Controllers on OS X	Pacific Heights	Tuesday 4:30PM

Labs

• View Controllers and Cocoa Lab	Frameworks Lab B Thursday 11:30AM

API Modernization

API Modernization

Many advances in Objective-C in recent years

API Modernization

Many advances in Objective-C in recent years

@property

API Modernization

Many advances in Objective-C in recent years

@property

instancetype

API Modernization

Many advances in Objective-C in recent years

@property

instancetype

enums with explicit underlying types, NS_ENUM and NS_OPTIONS

API Modernization

Many advances in Objective-C in recent years

@property

instancetype

enums with explicit underlying types, NS_ENUM and NS_OPTIONS

NS_REQUIRES_SUPER

API Modernization

Many advances in Objective-C in recent years

@property

instancetype

enums with explicit underlying types, NS_ENUM and NS_OPTIONS

NS_REQUIRES_SUPER

And a new one, NS_DESIGNATED_INITIALIZER

API Modernization

Many advances in Objective-C in recent years

@property

instancetype

enums with explicit underlying types, NS_ENUM and NS_OPTIONS

NS_REQUIRES_SUPER

And a new one, NS_DESIGNATED_INITIALIZER

In 10.10 and iOS 8 SDKs, increased adoption of these facilities

API Modernization

Why?

API Modernization

Why?

Allow stating the APIs more correctly and precisely

API Modernization

Why?

Allow stating the APIs more correctly and precisely

This enables

- The APIs to be more self-documenting

API Modernization

Why?

Allow stating the APIs more correctly and precisely

This enables

- The APIs to be more self-documenting
- The APIs to be more consistent

API Modernization

Why?

Allow stating the APIs more correctly and precisely

This enables

- The APIs to be more self-documenting
- The APIs to be more consistent
- Xcode to be more helpful

API Modernization

Why?

Allow stating the APIs more correctly and precisely

This enables

- The APIs to be more self-documenting
- The APIs to be more consistent
- Xcode to be more helpful
- The compiler to detect and warn about potential bugs

API Modernization

Why?

Allow stating the APIs more correctly and precisely

This enables

- The APIs to be more self-documenting
- The APIs to be more consistent
- Xcode to be more helpful
- The compiler to detect and warn about potential bugs
- Better exposure of APIs in Swift

API Modernization

Why?

Allow stating the APIs more correctly and precisely

This enables

- The APIs to be more self-documenting
- The APIs to be more consistent
- Xcode to be more helpful
- The compiler to detect and warn about potential bugs
- Better exposure of APIs in Swift

You may see new warnings or errors in your builds

Using @property

Using @property

Many accessors in APIs converted to @property

Using @property

Many accessors in APIs converted to @property

Obvious ones, for instance in NSControl

```
@property (weak) id target;
@property (copy) id objectValue;
```

Using @property

Many accessors in APIs converted to @property

Obvious ones, for instance in NSControl

```
@property (weak) id target;
@property (copy) id objectValue;
```

But also possibly “computed” properties

```
@property NSInteger integerValue;
@property (copy) NSString *stringValue;
@property (readonly, copy) NSString *description;
```

Using @property

Many accessors in APIs converted to @property

Obvious ones, for instance in NSControl

```
    @property (weak) id target;
    @property (copy) id objectValue;
```

But also possibly "computed" properties

```
    @property NSInteger integerValue;
    @property (copy) NSString *stringValue;
    @property (readonly, copy) NSString *description;
```

Use @property for anything that is about the value or state of an object or its relationship to other objects

When Not to Use @property

Not every method which can be expressed as a property should be

When Not to Use @property

Not every method which can be expressed as a property should be

When Not to Use @property

- Not every method which can be expressed as a property should be
- Bad candidates for @property

When Not to Use @property

Not every method which can be expressed as a property should be

Bad candidates for @property

- Methods which do things: load, parse, toggle, …

When Not to Use @property

Not every method which can be expressed as a property should be

Bad candidates for @property

- Methods which do things: load, parse, toggle, …
  - Generally these have a verb prefix on the name

When Not to Use @property

Not every method which can be expressed as a property should be

Bad candidates for @property

- Methods which do things: load, parse, toggle, …
    - Generally these have a verb prefix on the name
- Generators: init, copy, objectEnumerator

When Not to Use @property

Not every method which can be expressed as a property should be

Bad candidates for @property

- Methods which do things: load, parse, toggle, …
  - Generally these have a verb prefix on the name
- Generators: init, copy, objectEnumerator
- Methods which change state: nextObject

Weak @property

Weak @property

Use “weak” (zeroing weak) for targets

```
    @property (weak) id target;
```

Weak @property

Use “weak” (zeroing weak) for targets

```
    @property (weak) id target;
```

- Effective only in applications linked against 10.10 SDK

Weak @property

Use “weak” (zeroing weak) for targets

```
    @property (weak) id target;
```

- Effective only in applications linked against 10.10 SDK

Newly introduced delegates and data sources will also be weak

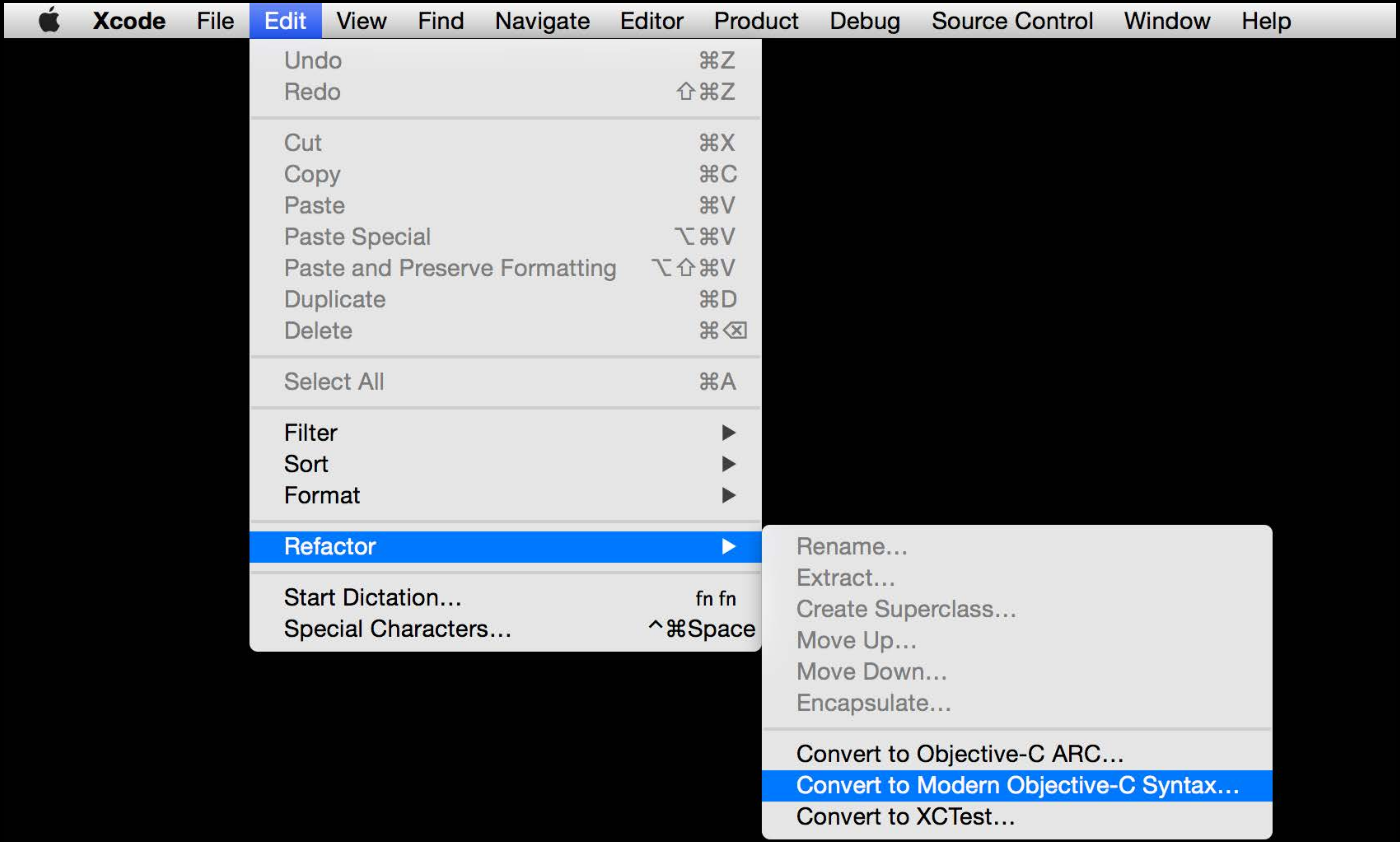
```
    @property (weak) id <NSGestureRecognizerDelegate> delegate;
```

Xcode Modernizer

If you want to modernize your code

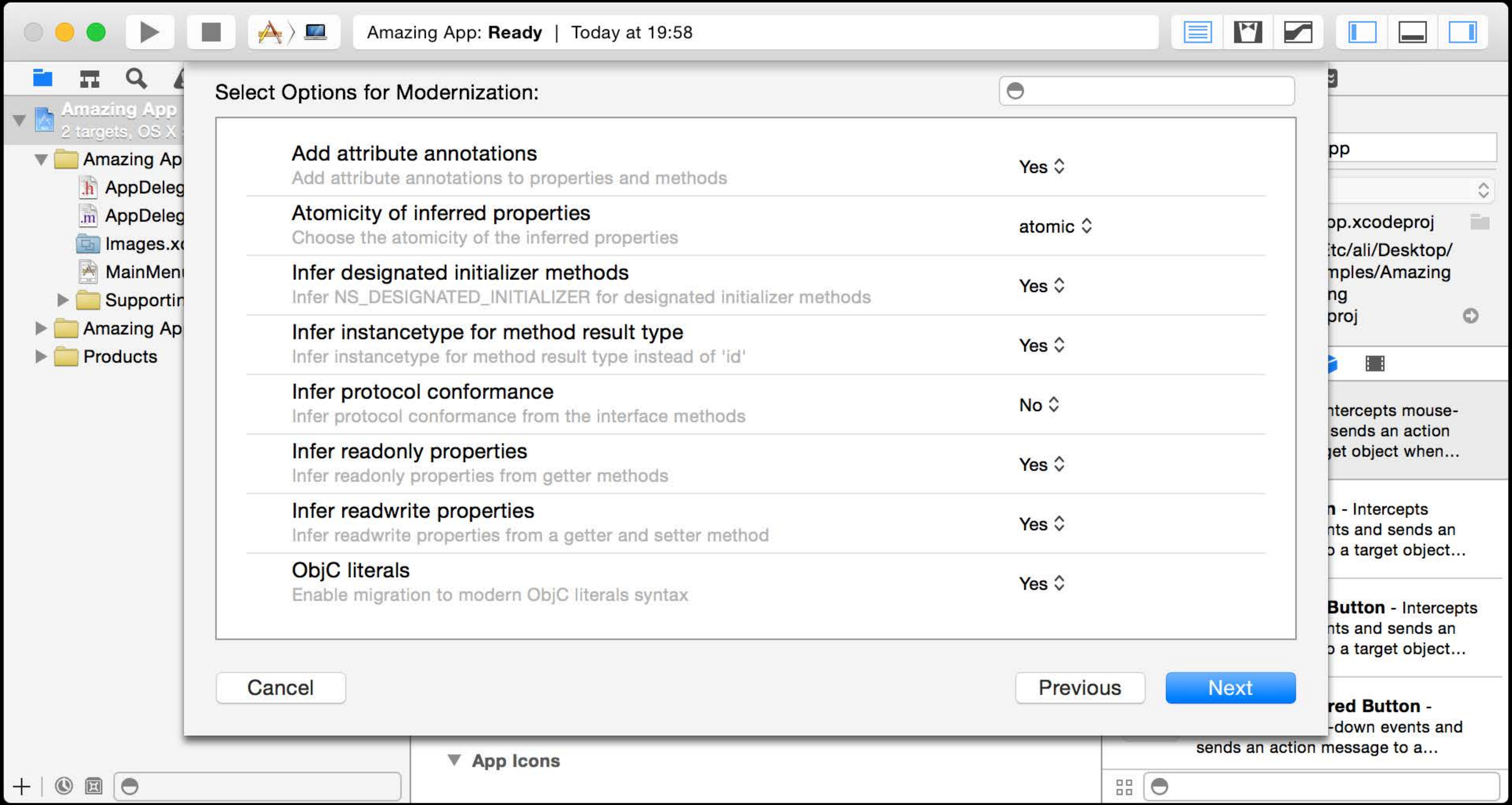
Xcode Modernizer

If you want to modernize your code



Xcode Modernizer

If you want to modernize your code



Select Options for Modernization:

Add attribute annotations Add attribute annotations to properties and methods	Yes
Atomicity of inferred properties Choose the atomicity of the inferred properties	atomic
Infer designated initializer methods Infer NS_DESIGNATED_INITIALIZER for designated initializer methods	Yes
Infer instancetype for method result type Infer instancetype for method result type instead of 'id'	Yes
Infer protocol conformance Infer protocol conformance from the interface methods	No
Infer readonly properties Infer readonly properties from getter methods	Yes
Infer readwrite properties Infer readwrite properties from a getter and setter method	Yes
ObjC literals Enable migration to modern ObjC literals syntax	Yes

Swift

Swift

A new language for Cocoa

Seamless interoperability with Cocoa APIs and Objective-C code

Cocoa APIs in Swift

Existing API guidelines for Cocoa Objective-C APIs apply to Swift

No changes in APIs as exposed in Swift

Cocoa Properties in Swift

Cocoa Properties in Swift

```
@property NSRect frame;
```

Cocoa Properties in Swift

```
@property NSRect frame;

var frame : NSRect
```

Cocoa Properties in Swift

```
@property NSRect frame;

var frame : NSRect


@property (readonly, strong) NSStoryboard *storyboard;
```

Cocoa Properties in Swift

```
@property NSRect frame;

var frame : NSRect


@property (readonly, strong) NSStoryboard *storyboard;

var storyboard : NSStoryboard! { get }
```

Cocoa Properties in Swift

```
@property NSRect frame;

var frame : NSRect


@property (readonly, strong) NSStoryboard *storyboard;

var storyboard : NSStoryboard! { get }


@property (copy) NSArray *subviews;
```

Cocoa Properties in Swift

```
@property NSRect frame;

var frame : NSRect


@property (readonly, strong) NSStoryboard *storyboard;

var storyboard : NSStoryboard! { get }


@property (copy) NSArray *subviews;

var subviews : AnyObject[]!
```

Cocoa APIs in Swift

Methods with no arguments

Cocoa APIs in Swift

Methods with no arguments

Declaration

```
    - (void)displayIfNeeded;
```

Cocoa APIs in Swift

Methods with no arguments

Declaration

```
    - (void)displayIfNeeded;

    func displayIfNeeded()
```

Cocoa APIs in Swift

Methods with no arguments

Declaration

```
    - (void)displayIfNeeded;

    func displayIfNeeded()
```

Invocation

Cocoa APIs in Swift

Methods with no arguments

Declaration

```
- (void)displayIfNeeded;

func displayIfNeeded()
```

Invocation

```
view.displayIfNeeded()
```

Cocoa APIs in Swift

Methods with one argument

Cocoa APIs in Swift

Methods with one argument

Declaration

```
    - (void)addSubview:(NSView *)aView;
```

Cocoa APIs in Swift

Methods with one argument

Declaration

```
    - (void)addSubview:(NSView *)aView;

    func addSubview(aView:NSView?)
```

Cocoa APIs in Swift

Methods with one argument

Declaration

```
    - (void)addSubview:(NSView *)aView;

    func addSubview(aView:NSView?)
```

Cocoa APIs in Swift

Methods with one argument

Declaration

```
    - (void)addSubview:(NSView *)aView;

    func addSubview(aView:NSView?)
```

Cocoa APIs in Swift

Methods with one argument

Declaration

```
    - (void)addSubview:(NSView *)aView;

    func addSubview(aView:NSView?)
```

Invocation

Cocoa APIs in Swift

Methods with one argument

Declaration

```
    - (void)addSubview:(NSView *)aView;

    func addSubview(aView:NSView?)
```

Invocation

```
    view.addSubview(anotherView)
```

Cocoa APIs in Swift

Methods with multiple arguments

Cocoa APIs in Swift

Methods with multiple arguments

Declaration

```
    - (void)performSegueWithIdentifier:(NSString)segueID sender:(id)sender;
```

Cocoa APIs in Swift

Methods with multiple arguments

Declaration

```
    - (void)performSegueWithIdentifier:(NSString)segueID sender:(id)sender;

    func performSegueWithIdentifier(segueID:String?, sender:AnyObject?)
```

Cocoa APIs in Swift

Methods with multiple arguments

Declaration

```
    - (void)performSegueWithIdentifier:(NSString)segueID sender:(id)sender;
                        ↙
    func performSegueWithIdentifier(segueID:String?, sender:AnyObject?)
```

Cocoa APIs in Swift

Methods with multiple arguments

Declaration

```
    - (void)performSegueWithIdentifier:(NSString)segueID sender:(id)sender;
                  ↙                                 ↙
    func performSegueWithIdentifier(segueID:String?, sender:AnyObject?)
```

Cocoa APIs in Swift

Methods with multiple arguments

Declaration

```
    - (void)performSegueWithIdentifier:(NSString)segueID sender:(id)sender;

    func performSegueWithIdentifier(segueID:String?, sender:AnyObject?)
```

Cocoa APIs in Swift

Methods with multiple arguments

Declaration

```
    - (void)performSegueWithIdentifier:(NSString)segueID sender:(id)sender;

    func performSegueWithIdentifier(segueID:String?, sender:AnyObject?)
```

Cocoa APIs in Swift

Methods with multiple arguments

Declaration

```
    - (void)performSegueWithIdentifier:(NSString)segueID sender:(id)sender;

    func performSegueWithIdentifier(segueID:String?, sender:AnyObject?)
```

Cocoa APIs in Swift

Methods with multiple arguments

Declaration

```
- (void)performSegueWithIdentifier:(NSString)segueID sender:(id)sender;

func performSegueWithIdentifier(segueID:String?, sender:AnyObject?)
```

Cocoa APIs in Swift

Methods with multiple arguments

Declaration

```
- (void)performSegueWithIdentifier:(NSString)segueID sender:(id)sender;

func performSegueWithIdentifier(segueID:String?, sender:AnyObject?)

func performSegueWithIdentifier(String?, sender:AnyObject?)
```

Cocoa APIs in Swift

Methods with multiple arguments

Declaration

```
- (void)performSegueWithIdentifier:(NSString)segueID sender:(id)sender;

func performSegueWithIdentifier(segueID:String?, sender:AnyObject?)

func performSegueWithIdentifier(String?, sender:AnyObject?)
```

Invocation

```
viewController.performSegueWithIdentifier("Next", sender:nil)
```

Cocoa APIs in Swift

Cocoa APIs omit explicit label on the first argument

Cocoa APIs in Swift

Cocoa APIs omit explicit label on the first argument

Name of first argument is part of the base name

Cocoa APIs in Swift

Cocoa APIs omit explicit label on the first argument

Name of first argument is part of the base name

```
    viewController.performSegueWithIdentifier(“Next”, sender:nil)
```

Cocoa APIs in Swift

Cocoa APIs omit explicit label on the first argument

Name of first argument is part of the base name

```
    viewController.performSegueWithIdentifier(“Next”, sender:nil)

    canCollapse = delegate.splitView(self, canCollapseSubview:subview)
```

Cocoa APIs in Swift

Cocoa APIs omit explicit label on the first argument

Name of first argument is part of the base name

```
viewController.performSegueWithIdentifier(“Next”, sender:nil)

canCollapse = delegate.splitView(self, canCollapseSubview:subview)

success = url.setResourceValue(value, forKey:NSURLNameKey, error:&err)
```

Cocoa APIs in Swift

When to consider using label on first argument

Cocoa APIs in Swift

When to consider using label on first argument

Cases where arguments are “equally weighted” subparts of a whole

Cocoa APIs in Swift

When to consider using label on first argument

Cases where arguments are "equally weighted" subparts of a whole

Instead of

```
    func moveToX(CGFloat, y:CGFloat)
```

Cocoa APIs in Swift

When to consider using label on first argument

Cases where arguments are “equally weighted” subparts of a whole

Instead of

```
    func moveToX(CGFloat, y:CGFloat)
```

Perhaps

```
    func move(x:CGFloat, y:CGFloat), or moveToLocation(x:CGFloat, y:CGFloat)
```

Cocoa APIs in Swift

When to consider using label on first argument

Cases where arguments are “equally weighted” subparts of a whole

Instead of

```
    func moveToX(CGFloat, y:CGFloat)
```

Perhaps

```
    func move(x:CGFloat, y:CGFloat), or moveToLocation(x:CGFloat, y:CGFloat)
```

Better yet do

```
    func moveToLocation(NSPoint)
```

Cocoa APIs in Swift

When to consider using label on first argument

Cases where arguments are “equally weighted” subparts of a whole

Instead of

```
    func moveToX(CGFloat, y:CGFloat)
```

Perhaps

```
    func move(x:CGFloat, y:CGFloat), or moveToLocation(x:CGFloat, y:CGFloat)
```

Better yet do

```
    func moveToLocation(NSPoint)
```

Use a “combined” type where appropriate

- NSDate/NSDateComponents, NS/UIColor, NSRange, CGRect, SCNVector3, …

Cocoa APIs in Swift

Init methods

Cocoa APIs in Swift

Init methods

Declaration

```
    - (instancetype)initWithFrame:(NSRect)frameRect;
```

Cocoa APIs in Swift

Init methods

Declaration

```
    - (instancetype)initWithFrame:(NSRect)frameRect;

    init(frame frameRect:NSRect)
```

Cocoa APIs in Swift

Init methods

Declaration

```
- (instancetype)initWithFrame:(NSRect)frameRect;

init(frame frameRect:NSRect)
```

Cocoa APIs in Swift

Init methods

Declaration

```
    - (instancetype)initWithFrame:(NSRect)frameRect;

    init(frame frameRect:NSRect)
```

Invocation

```
    view = NSView(frame:rect)
```

Cocoa APIs in Swift

Init methods

Declaration

```
    - (instancetype)initWithFrame:(NSRect)frameRect;

    init(frame frameRect:NSRect)
```

Invocation

```
    view = NSView(frame:rect)
```

Cocoa APIs in Swift

Convenience constructors

Cocoa APIs in Swift

Convenience constructors

Declaration

```
    + (NSColor *)colorWithPatternImage:(NSImage *)image;
```

Cocoa APIs in Swift

Convenience constructors

Declaration

```
    + (NSColor *)colorWithPatternImage:(NSImage *)image;

    init(patternImage image:NSImage?)
```

Cocoa APIs in Swift

Convenience constructors

Declaration

```
    + (NSColor *)colorWithPatternImage:(NSImage *)image;

    init(patternImage image:NSImage?)
```

Invocation

```
    color = NSColor(patternImage:image)
```

Cocoa APIs in Swift

Enumerated types

Cocoa APIs in Swift

Enumerated types

```
typedef NS_ENUM(NSInteger, NSByteCountFormatterCountStyle) {
    NSByteCountFormatterCountStyleFile,
    NSByteCountFormatterCountStyleMemory,
    NSByteCountFormatterCountStyleDecimal,
    NSByteCountFormatterCountStyleBinary
};
```

Cocoa APIs in Swift

Enumerated types

```
typedef NS_ENUM(NSInteger, NSByteCountFormatterCountStyle) {
   NSByteCountFormatterCountStyleFile,
   NSByteCountFormatterCountStyleMemory,
   NSByteCountFormatterCountStyleDecimal,
   NSByteCountFormatterCountStyleBinary
};

enum NSByteCountFormatterCountStyle : Int {
    case File
    case Memory
    case Decimal
    case Binary
}
```

Cocoa APIs in Swift

Enumerated types

```
enum NSByteCountFormatterCountStyle : Int {
    case File
    case Memory
    case Decimal
    case Binary
}
```

Cocoa APIs in Swift

Enumerated types

```
enum NSByteCountFormatterCountStyle : Int {
    case File
    case Memory
    case Decimal
    case Binary
}

NSByteCountFormatter.stringFromByteCount(numBytes,
                                         countStyle:NSByteCountFormatterCountStyle.File)
```

Cocoa APIs in Swift

Enumerated types

```
enum NSByteCountFormatterCountStyle : Int {
    case File
    case Memory
    case Decimal
    case Binary
}

NSByteCountFormatter.stringFromByteCount(numBytes,
                                         countStyle:NSByteCountFormatterCountStyle.File)

NSByteCountFormatter.stringFromByteCount(numBytes, countStyle:.File)
```

Related Sessions

●	Introduction to Swift	Presidio	Tuesday 2:00PM
●	Integrating Swift with Objective-C	Presidio	Wednesday 9:00AM
●	Swift Interoperability In Depth	Presidio	Wednesday 3:15PM
●	Creating Modern Cocoa Apps	Marina	Thursday 10:15AM

Labs

● Swift Lab	Tools Lab A	Every day 9:00AM

Gesture Recognizers

Gesture Recognizers

Gesture Recognizers

New class NSGestureRecognizer

Gesture Recognizers

New class NSGestureRecognizer

Subclasses

- NSClickGestureRecognizer
- NSMagnificationGestureRecognizer
- NSPanGestureRecognizer
- NSPressGestureRecognizer
- NSRotationGestureRecognizer

Gesture Recognizers

New class NSGestureRecognizer

Subclasses

- NSClickGestureRecognizer
- NSMagnificationGestureRecognizer
- NSPanGestureRecognizer
- NSPressGestureRecognizer
- NSRotationGestureRecognizer

APIs to create custom subclasses

Related Sessions

● Storyboards and Controllers on OS X	Pacific Heights	Tuesday 4:30PM

Block-based Event Tracking

Block-based Event Tracking

```
@interface NSWindow …

- (void)trackEventsMatchingMask:(NSEventMask)mask
                        timeout:(NSTimeInterval)timeout
                           mode:(NSString *)mode
                        handler:(void(^)(NSEvent *event, BOOL *stop))tracker;
```

Block-based Event Tracking

```
@interface NSWindow …

- (void)trackEventsMatchingMask:(NSEventMask)mask
                        timeout:(NSTimeInterval)timeout
                           mode:(NSString *)mode
                        handler:(void(^)(NSEvent *event, BOOL *stop))tracker;
```

Continuously track events until stopped or timeout is reached

Accessibility

New Accessibility APIs

Simpler

New Accessibility APIs

Simpler

Accessibility values expressed directly as properties

New Accessibility APIs

Simpler

Accessibility values expressed directly as properties

No need to subclass

New Accessibility APIs

Simpler

Accessibility values expressed directly as properties

No need to subclass

Better compile time warnings

New Accessibility APIs

Before

```
- (id)accessibilityAttributeValue:(NSString *)attr {
    if ([attr isEqualToString:NSAccessibilityDescriptionAttribute]) {
        return NSLocalizedString(@"Take Screenshot", @“…”);
    } else {
        return [super accessibilityAttributeValue:attr];
    }
}
```

Now

```
- (NSString *)accessibilityLabel {
    return NSLocalizedString(@"Take Screenshot", @“…”);
}
```

Related Sessions

● Accessibility on OS X	Russian Hill	Tuesday 2:00PM

Labs

● Accessibility and Speech Lab	Frameworks Lab B Wednesday 10:15AM

Power

Quality of Service

New property on NSOperation, NSOperationQueue, NSThread, …

```
    @property NSQualityOfService qualityOfService;
```

Allows indicating the nature and importance of work

Lets the system manage resources

Quality of Service

```
typedef NS_ENUM(NSInteger, NSQualityOfService) {
    NSQualityOfServiceUserInteractive,
    NSQualityOfServiceUserInitiated,
    NSQualityOfServiceUtility,
    NSQualityOfServiceBackground,
    NSQualityOfServiceDefault
};
```

Quality of Service

```
typedef NS_ENUM(NSInteger, NSQualityOfService) {
    NSQualityOfServiceUserInteractive,     // Scrolling email message
    NSQualityOfServiceUserInitiated,
    NSQualityOfServiceUtility,
    NSQualityOfServiceBackground,
    NSQualityOfServiceDefault
};
```

Quality of Service

```
typedef NS_ENUM(NSInteger, NSQualityOfService) {
   NSQualityOfServiceUserInteractive,     // Scrolling email message
   NSQualityOfServiceUserInitiated,       // Showing an email message
   NSQualityOfServiceUtility,
   NSQualityOfServiceBackground,
   NSQualityOfServiceDefault
};
```

Quality of Service

```
typedef NS_ENUM(NSInteger, NSQualityOfService) {
   NSQualityOfServiceUserInteractive,     // Scrolling email message
   NSQualityOfServiceUserInitiated,       // Showing an email message
   NSQualityOfServiceUtility,             // Periodic mail fetch
   NSQualityOfServiceBackground,
   NSQualityOfServiceDefault
};
```

Quality of Service

```
typedef NS_ENUM(NSInteger, NSQualityOfService) {
   NSQualityOfServiceUserInteractive,     // Scrolling email message
   NSQualityOfServiceUserInitiated,       // Showing an email message
   NSQualityOfServiceUtility,             // Periodic mail fetch
   NSQualityOfServiceBackground,          // Indexing
   NSQualityOfServiceDefault
};
```

Quality of Service

```
typedef NS_ENUM(NSInteger, NSQualityOfService) {
   NSQualityOfServiceUserInteractive,     // Scrolling email message
   NSQualityOfServiceUserInitiated,       // Showing an email message
   NSQualityOfServiceUtility,             // Periodic mail fetch
   NSQualityOfServiceBackground,          // Indexing
   NSQualityOfServiceDefault              // Inferred from environment
};
```

NSBackgroundActivityScheduler

Cocoa-level interface to the XPC Activity API

Schedule maintenance or background kinds of tasks

Related Sessions

• Writing Energy Efficient Code, Part 1	Russian Hill	Wednesday 10:15AM
• Power, Performance and Diagnostics: What's new in GCD and XPC	Russian Hill	Thursday 2:00PM

NSString

NSString Encoding Detector

NSString Encoding Detector

API to detect string encoding of a sequence of bytes

```
+ (NSStringEncoding)stringEncodingForData:(NSData *)data
                          encodingOptions:(NSDictionary *)opts
                          convertedString:(NSString **)string
                      usedLossyConversion:(BOOL *)usedLossyConversion;
```

NSString Encoding Detector

API to detect string encoding of a sequence of bytes

```
+ (NSStringEncoding)stringEncodingForData:(NSData *)data
                          encodingOptions:(NSDictionary *)opts
                          convertedString:(NSString **)string
                      usedLossyConversion:(BOOL *)usedLossyConversion;
```

Options include

NSString Encoding Detector

API to detect string encoding of a sequence of bytes

```
+ (NSStringEncoding)stringEncodingForData:(NSData *)data
                          encodingOptions:(NSDictionary *)opts
                          convertedString:(NSString **)string
                      usedLossyConversion:(BOOL *)usedLossyConversion;
```

Options include

- Encodings to be considered or not

NSString Encoding Detector

API to detect string encoding of a sequence of bytes

```
+ (NSStringEncoding)stringEncodingForData:(NSData *)data
                          encodingOptions:(NSDictionary *)opts
                          convertedString:(NSString **)string
                      usedLossyConversion:(BOOL *)usedLossyConversion;
```

Options include

- Encodings to be considered or not
- Whether to allow lossy conversion

NSString Encoding Detector

API to detect string encoding of a sequence of bytes

```
+ (NSStringEncoding)stringEncodingForData:(NSData *)data
                          encodingOptions:(NSDictionary *)opts
                          convertedString:(NSString **)string
                      usedLossyConversion:(BOOL *)usedLossyConversion;
```

Options include

- Encodings to be considered or not
- Whether to allow lossy conversion
- Language hint

NSString

Two other small new APIs

```
- (BOOL)containsString:(NSString *)str;
- (BOOL)localizedCaseInsensitiveContainsString:(NSString *)str;
```

Tagged Pointer Strings

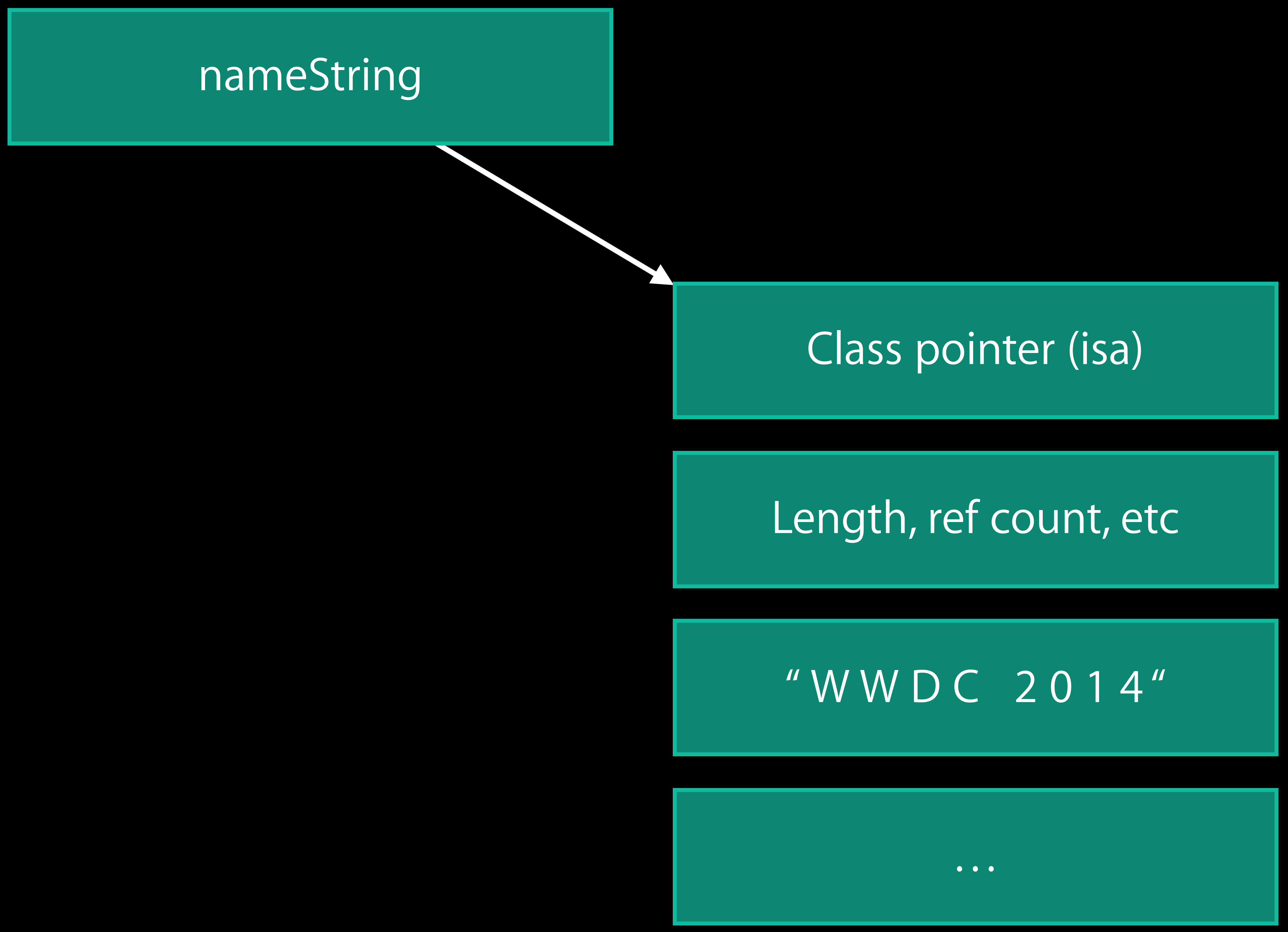
Tagged Pointer Strings

Where possible, we stuff the whole NSString into the object pointer itself

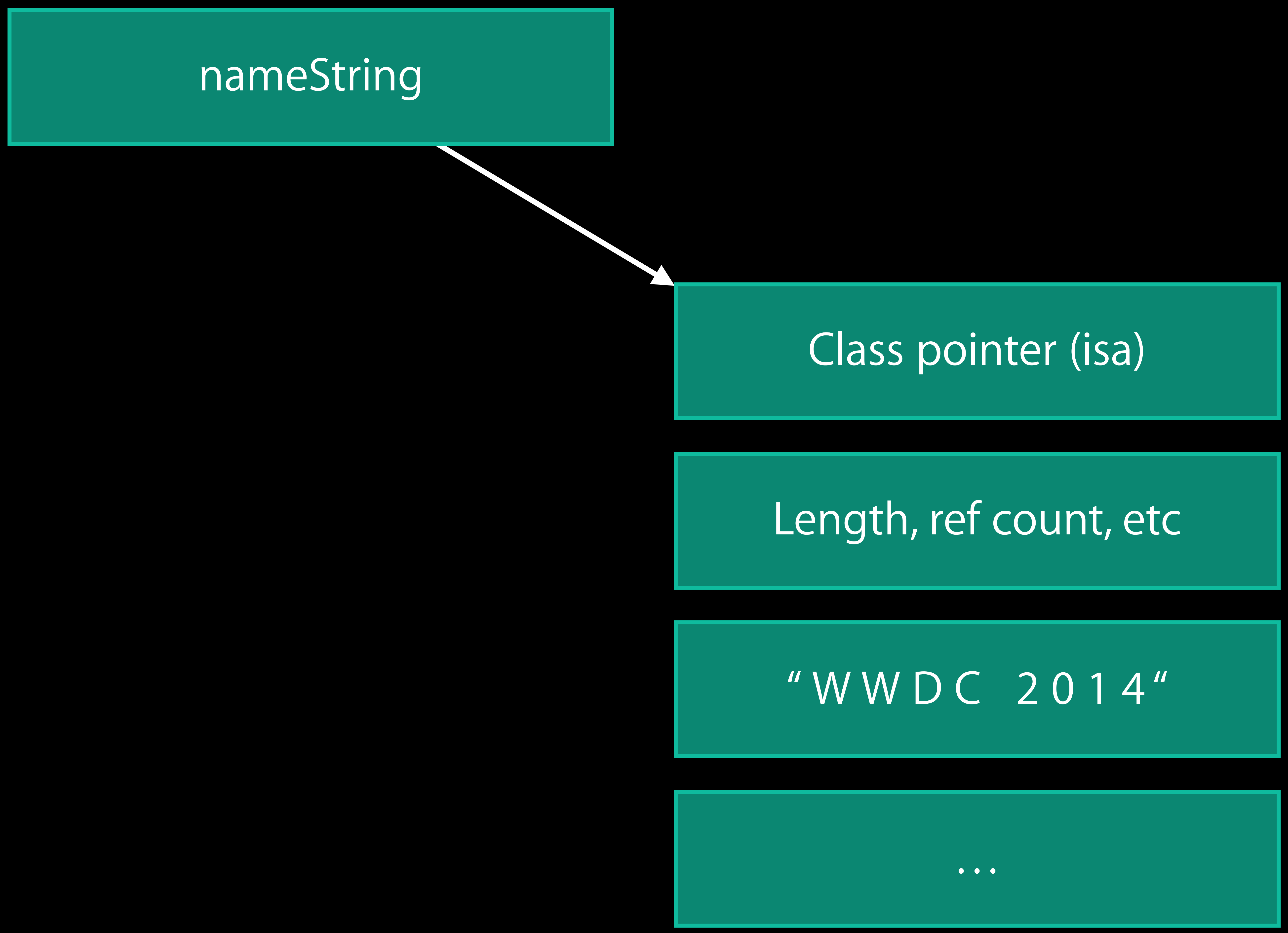
```
NSString *nameString = [NSString stringWithUTF8String:“WWDC 2014”];
NSString *nameString = [NSString stringWithUTF8String:“WWDC 2014”];
```

nameString

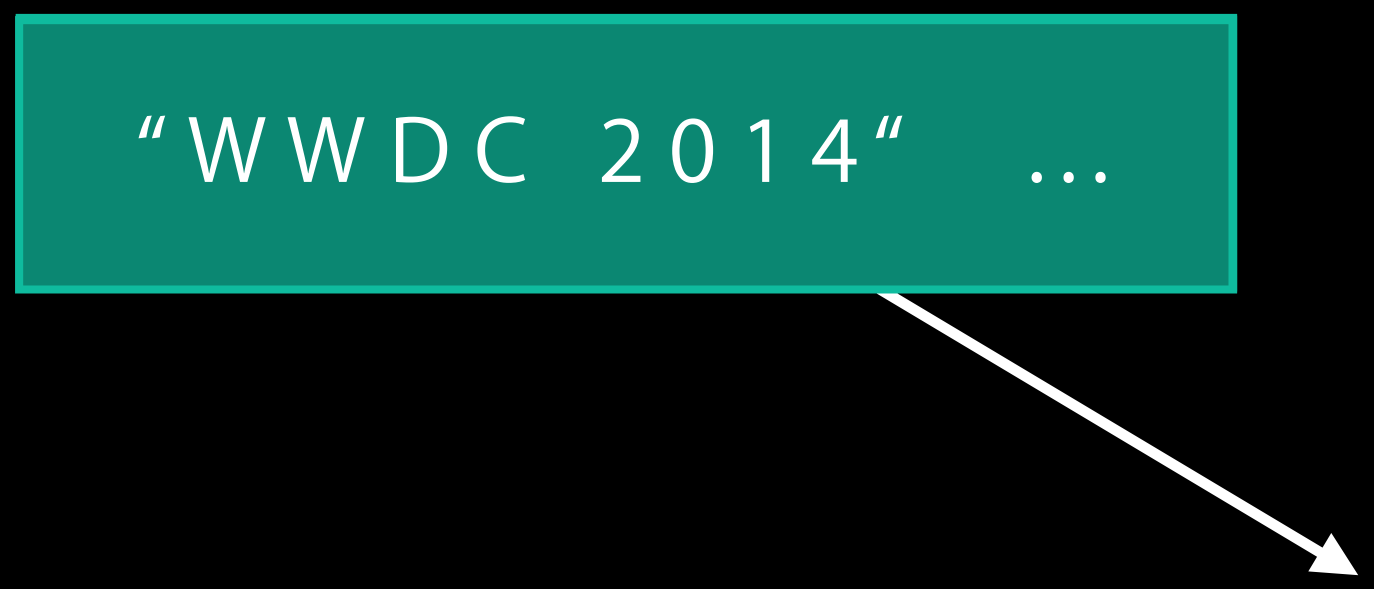
```
NSString *nameString = [NSString stringWithUTF8String:“WWDC 2014”];
```



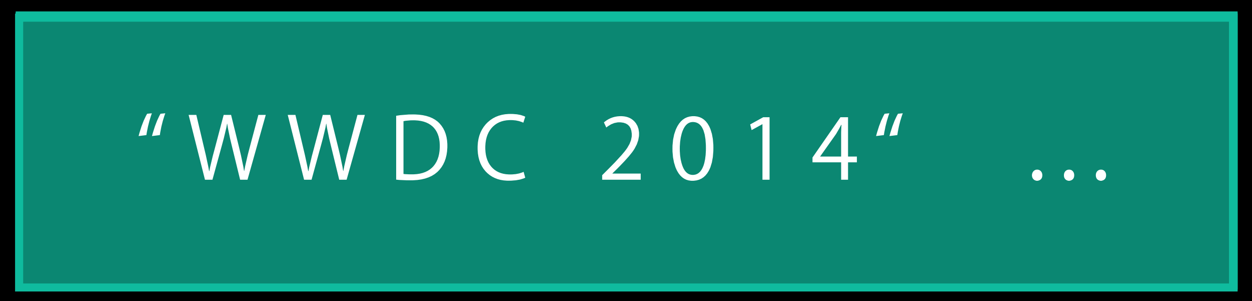
```
NSString *nameString = [NSString stringWithUTF8String:“WWDC 2014”];
```



```
NSString *nameString = [NSString stringWithUTF8String:“WWDC 2014”];
```



```
NSString *nameString = [NSString stringWithUTF8String:“WWDC 2014”];
```



Tagged Pointer Strings

Things to watch for

Tagged Pointer Strings

Things to watch for

No isa pointer!

Tagged Pointer Strings

Things to watch for

No isa pointer!

Different performance characteristics

Tagged Pointer Strings

Things to watch for

No isa pointer!

Different performance characteristics

Better out-of-bounds checking

Tagged Pointer Strings

Things to watch for

No isa pointer!

Different performance characteristics

Better out-of-bounds checking

Automatically enabled in 64-bit apps linked against 10.10 SDK

Formatters

New NSFormatters

New NSFormatters

NSMassFormatter

- 25.8 pounds

New NSFormatters

NSMassFormatter

NSEnergyFormatter

- 800 kcal

New NSFormatters

NSMassFormatter

NSEnergyFormatter

NSLengthFormatter

- 42.5 miles

New NSFormatters

NSMassFormatter

NSEnergyFormatter

NSLengthFormatter

NSDateIntervalFormatter

- Jun 3, 2014, 11:30 AM-12:30 PM

New NSFormatters

NSMassFormatter

NSEnergyFormatter

NSLengthFormatter

NSDateIntervalFormatter

- Jun 3, 2014, 11:30 AM-12:30 PM

NSDateComponentsFormatter

- 3 hours, 25 minutes, 42 seconds

New NSFormatters

NSMassFormatter

NSEnergyFormatter

NSLengthFormatter

NSDateIntervalFormatter

- Jun 3, 2014, 11:30 AM-12:30 PM

NSDateComponentsFormatter

- 3 hours, 25 minutes, 42 seconds
- About 10 minutes remaining

New NSFormatters

NSMassFormatter

NSEnergyFormatter

NSLengthFormatter

NSDateIntervalFormatter

NSDateComponentsFormatter

Various customization options

New NSFormatters

NSMassFormatter

NSEnergyFormatter

NSLengthFormatter

NSDateIntervalFormatter

NSDateComponentsFormatter

Various customization options

Formatting only, no parsing

Formatting Context

Formatting Context

```
typedef NS_ENUM(NSInteger, NSFormattingContext) {
    NSFormattingContextUnknown,
    NSFormattingContextDynamic,
    NSFormattingContextStandalone,
    NSFormattingContextListItem,
    NSFormattingContextBeginningOfSentence,
    NSFormattingContextMiddleOfSentence
};
```

Formatting Context

```
enum NSFormattingContext : Int {
    Unknown,
    Dynamic,
    Standalone,
    ListItem,
    BeginningOfSentence,
    MiddleOfSentence
}
```

Formatting Context

```
enum NSFormattingContext : Int {
    Unknown,
    Dynamic,
    Standalone,
    ListItem,
    BeginningOfSentence,    //  “Juin 2014 …”
    MiddleOfSentence
}
```

Formatting Context

```
enum NSFormattingContext : Int {
    Unknown,
    Dynamic,
    Standalone,
    ListItem,
    BeginningOfSentence,    //  “Juin 2014 …”
    MiddleOfSentence        //  “… juin 2014 …”
}
```

Formatting Context

```
enum NSFormattingContext : Int {
    Unknown,
    Dynamic,                   //  Chooses automatically
    Standalone,
    ListItem,
    BeginningOfSentence,       //  “Juin 2014 …”
    MiddleOfSentence           //  “… juin 2014 …”
}
```

Related Sessions

Advanced Topics in Internationalization	Russian Hill	Tuesday 9:00AM

Labs

Internationalization Lab	Frameworks Lab B	Tuesday 3:15PM
Foundation Lab	Frameworks Lab B	Wednesday 9:00AM

iCloud

CloudKit

New framework for managing structured data on iCloud and sharing between users

Back-end for iCloud document storage

Related Sessions

• Introducing CloudKit	Mission	Tuesday 3:15PM
• Advanced CloudKit	Mission	Thursday 3:15PM

iCloud Document Storage

iCloud Document Storage

New back-end

iCloud Document Storage

New back-end

Document versions available on iCloud

iCloud Document Storage

New back-end

Document versions available on iCloud

iCloud Drive available to all applications

iCloud Document Storage

Handling of non-downloaded files has changed

- Now tracked with invisible files with different names

iCloud Document Storage

Handling of non-downloaded files has changed

- Now tracked with invisible files with different names

Use NSMetadataQuery, NSMetadataItem, and NSFileCoordinator to access iCloud files

iCloud Document Storage

Handling of non-downloaded files has changed

- Now tracked with invisible files with different names

Use NSMetadataQuery, NSMetadataItem, and NSFileCoordinator to access iCloud files

Do not enumerate iCloud container contents directly

- If you do, ignore hidden or unrecognized files

iCloud Document Storage

Handling of non-downloaded files has changed

- Now tracked with invisible files with different names

Use NSMetadataQuery, NSMetadataItem, and NSFileCoordinator to access iCloud files

Do not enumerate iCloud container contents directly

- If you do, ignore hidden or unrecognized files

To get metadata, use the new "promised item" APIs on NSURL

```
- (BOOL)getPromisedItemResourceValue:(id *)value
                              forKey:(NSString *)key
                               error:(NSError **)error;
```

Related Sessions

● Building a Document-based App	Marina	Thursday 11:30AM

Core Data

Core Data

Core Data

Batch updates with NSBatchUpdateRequest

Core Data

Batch updates with NSBatchUpdateRequest

Asynchronous fetching with NSAsynchronousFetchRequest

- Also provides NSProgress support

Core Data

Batch updates with NSBatchUpdateRequest

Asynchronous fetching with NSAsynchronousFetchRequest

- Also provides NSProgress support

iCloud

- Infrastructure improvements

Related Sessions

● What's New in Core Data	Pacific Heights	Thursday 9:00AM

Labs

Core Data Lab	Services Lab B	Wednesday 9:00AM
Core Data Lab	Services Lab B	Thursday 10:15AM
Core Data Lab	Services Lab B	Friday 9:00AM

Auto Layout

Auto Layout

New APIs to activate NSLayoutConstraints directly

```
+ (void)activateConstraints:(NSArray *)constraints;
+ (void)deactivateConstraints:(NSArray *)constraints;
@property (getter=isActive) BOOL active;
```

Auto Layout

New APIs to activate NSLayoutConstraints directly

```
+ (void)activateConstraints:(NSArray *)constraints;
+ (void)deactivateConstraints:(NSArray *)constraints;
@property (getter=isActive) BOOL active;
```

These replace the existing APIs on NSView

```
- (void)addConstraint:(NSLayoutConstraint *)constraint;
- (void)addConstraints:(NSArray *)constraints;
- (void)removeConstraint:(NSLayoutConstraint *)constraint;
- (void)removeConstraints:(NSArray *)constraints;
```

NSCell

On its way to formal deprecation

NSCell

On its way to formal deprecation

Some NSCell APIs promoted to their corresponding NSControl subclasses

- NSControl, NSTextField, NSSearchField, NSLevelIndicator, NSSlider, NSPathControl
- Use the controls where possible

NSCell

On its way to formal deprecation

Some NSCell APIs promoted to their corresponding NSControl subclasses

- NSControl, NSTextField, NSSearchField, NSLevelIndicator, NSSlider, NSPathControl
- Use the controls where possible

NSCell-based NSTableView deprecated

- Use view-based NSTableView

NSCell

On its way to formal deprecation

Some NSCell APIs promoted to their corresponding NSControl subclasses

- NSControl, NSTextField, NSSearchField, NSLevelIndicator, NSSlider, NSPathControl
- Use the controls where possible

NSCell-based NSTableView deprecated

- Use view-based NSTableView

NSMatrix-based NSBrowser deprecated

- Use item-based NSBrowser

NSCell

On its way to formal deprecation

Some NSCell APIs promoted to their corresponding NSControl subclasses

- NSControl, NSTextField, NSSearchField, NSLevelIndicator, NSSlider, NSPathControl
- Use the controls where possible

NSCell-based NSTableView deprecated

- Use view-based NSTableView

NSMatrix-based NSBrowser deprecated

- Use item-based NSBrowser

NSMatrix on its way out too

- Sibling radio buttons with same action will now operate as a group

More New Stuff

More New Stuff

NSTableView

- Create statically

More New Stuff

NSTableView

- Create statically

NSImage

- Specify fancy resizing behaviors with capInsets and resizingMode

More New Stuff

NSTableView

- Create statically

NSImage

- Specify fancy resizing behaviors with capInsets and resizingMode

NSBitmapImageRep

- More bitmap formats, for instance BGRA

More New Stuff

NSTableView

- Create statically

NSImage

- Specify fancy resizing behaviors with capInsets and resizingMode

NSBitmapImageRep

- More bitmap formats, for instance BGRA

Asset catalogs

- Support for more formats and slicing

Some More New Stuff

NSAttributedString

- Apply letterpress text effect

Some More New Stuff

NSAttributedString

- Apply letterpress text effect

NSPopover

- Delegate method popoverShouldDetach:

Some More New Stuff

NSAttributedString

- Apply letterpress text effect

NSPopover

- Delegate method popoverShouldDetach:

NSComboBox, NSDatePicker, NSPopupButton, NSSearchField, and NSSplitView

- Flip as expected for right-to-left

Some More New Stuff

NSAttributedString

- Apply letterpress text effect

NSPopover

- Delegate method popoverShouldDetach:

NSComboBox, NSDatePicker, NSPopupButton, NSSearchField, and NSSplitView

- Flip as expected for right-to-left

NSNibLoading

- Do custom setup in your view subclass for live views support in Interface Builder

Other New Stuff

NSOpenGLContext

- Query the pixelFormat and lock the context without going down to CGLContext

Other New Stuff

NSOpenGLContext

- Query the pixelFormat and lock the context without going down to CGLContext

NSFileCoordinator

- Asynchronous waiting with coordinateAccessWithIntents:queue:byAccessor:

Other New Stuff

NSOpenGLContext

- Query the pixelFormat and lock the context without going down to CGLContext

NSFileCoordinator

- Asynchronous waiting with coordinateAccessWithIntents:queue:byAccessor:

NSWorkspace

- Specify which apps to use and how when opening URLs

Other New Stuff

NSOpenGLContext

- Query the pixelFormat and lock the context without going down to CGLContext

NSFileCoordinator

- Asynchronous waiting with coordinateAccessWithIntents:queue:byAccessor:

NSWorkspace

- Specify which apps to use and how when opening URLs

NSURL

- Resolve alias files with URLByResolvingAliasFileAtURL:options:error:

And Even More New Stuff

NSProcessInfo

- operatingSystemVersion, and isOperatingSystemAtLeastVersion:

And Even More New Stuff

NSProcessInfo

- operatingSystemVersion, and isOperatingSystemAtLeastVersion:

NSXPCConnection

- NSProgress support across processes

Summary

Summary

New Look

Extensions

Handoff

Storyboards and View Controllers

API Modernization

Swift

And many others…

More Information

Jake Behrens
Frameworks Evangelist
behrens@apple.com

Documentation
Mac Dev Center
https://developer.apple.com/devcenter/mac

Release Notes
Application Kit Release Notes, Foundation Kit Release Notes
http://developer.apple.com/mac

Apple Developer Forums
http://devforums.apple.com

Related Sessions

Creating Modern Cocoa Apps	Marina	Thursday 10:15AM
Adapting Your App to the New UI of OS X Yosemite	Pacific Heights	Tuesday 3:15PM
Adopting Advanced Features of the New UI of OS X Yosemite	Marina	Wednesday 2:00PM
Storyboards and Controllers on OS X	Pacific Heights	Tuesday 4:30PM
Creating Extensions for iOS and OS X, Part 1	Mission	Tuesday 2:00PM
Adopting Handoff on iOS and OS X	Marina	Wednesday 2:00PM
What's New in Interface Builder	Mission	Wednesday 3:15PM

Labs

Cocoa Lab	Frameworks Lab B	Tuesday 12:30PM
Internationalization Lab	Frameworks Lab B	Tuesday 3:15PM
Foundation Lab	Frameworks Lab B	Wednesday 9:00AM
New UI and Cocoa Lab	Frameworks Lab B	Wednesday 3:15PM
View Controllers and Cocoa Lab	Frameworks Lab B	Thursday 11:30AM
Cocoa Lab	Frameworks Lab B	Thursday 4:30PM

WWDC14