

What's New in Cocoa

Session 205

Ali Ozer

Director of Cocoa Frameworks

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

Topics

Topics

- High level coverage of updates to Cocoa in 10.9

Topics

- High level coverage of updates to Cocoa in 10.9
- Pointers to related sessions and labs

Topics

- High level coverage of updates to Cocoa in 10.9
- Pointers to related sessions and labs
- Pretty much everything we talk about is new

Topics



- High level coverage of updates to Cocoa in 10.9
- Pointers to related sessions and labs
- Pretty much everything we talk about is new

Topics



- High level coverage of updates to Cocoa in 10.9
- Pointers to related sessions and labs
- Pretty much everything we talk about is new
- Not a complete reference to all new or changed APIs!
 - Highlights
 - See headers files or release notes for details

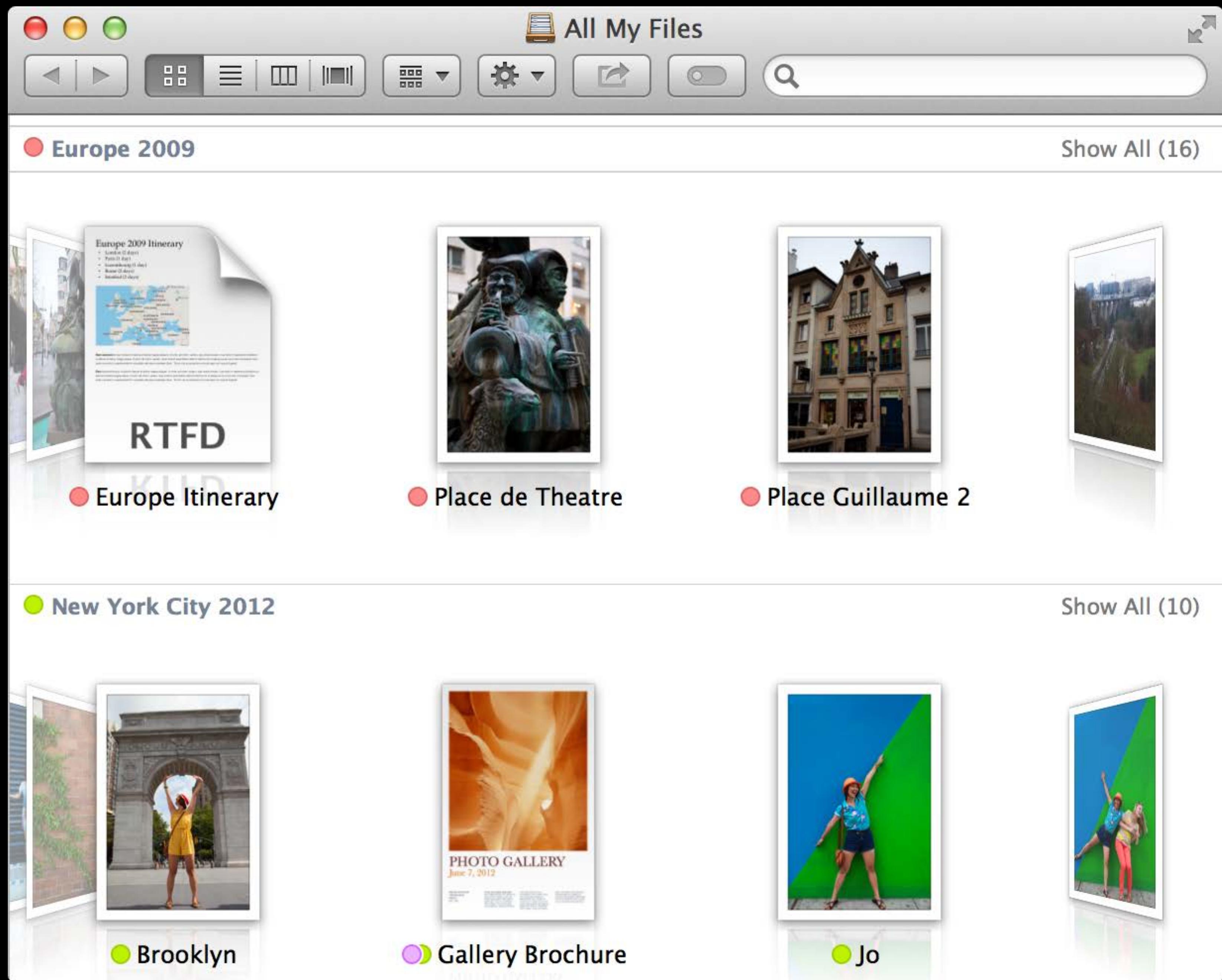
Tags

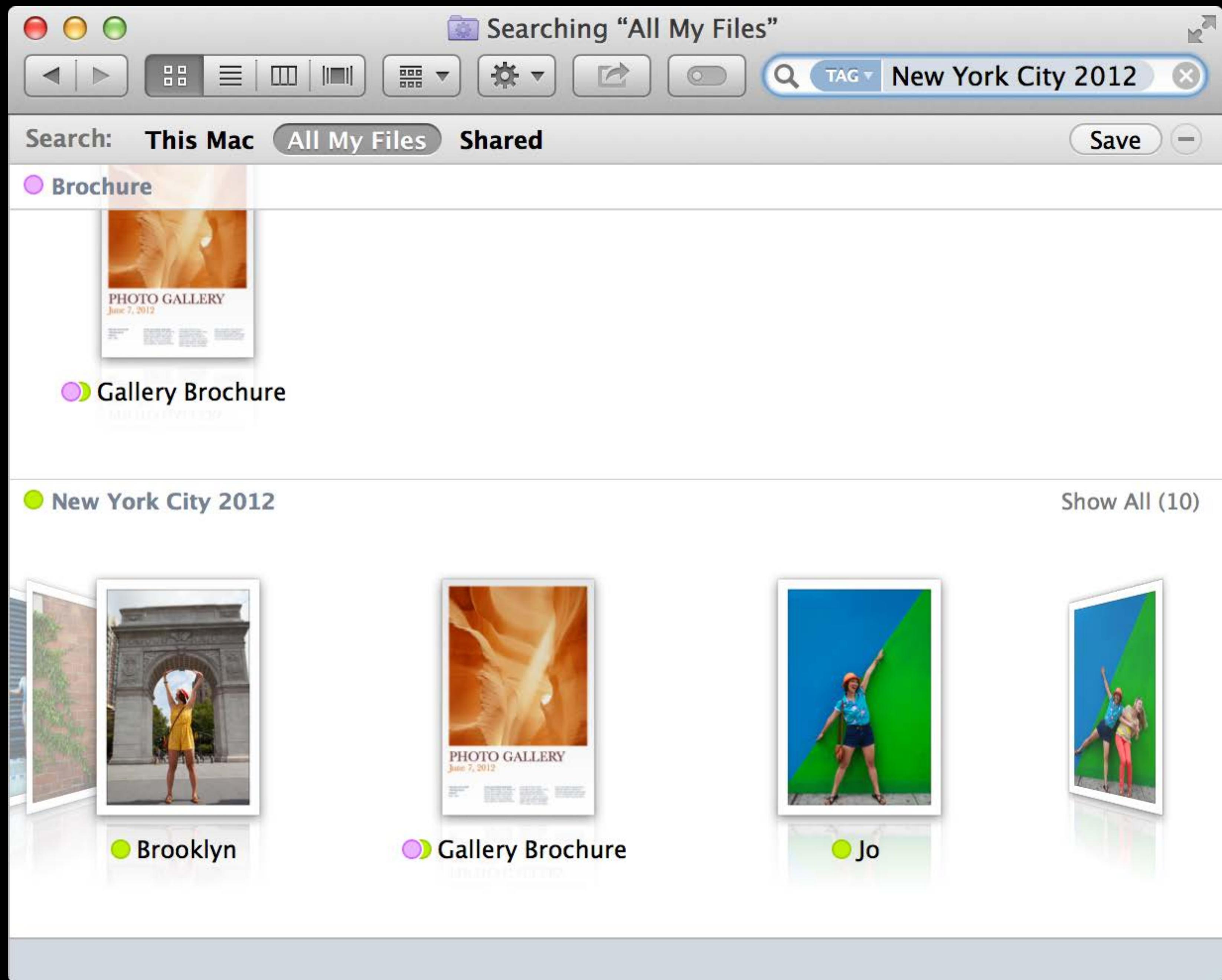
Tags

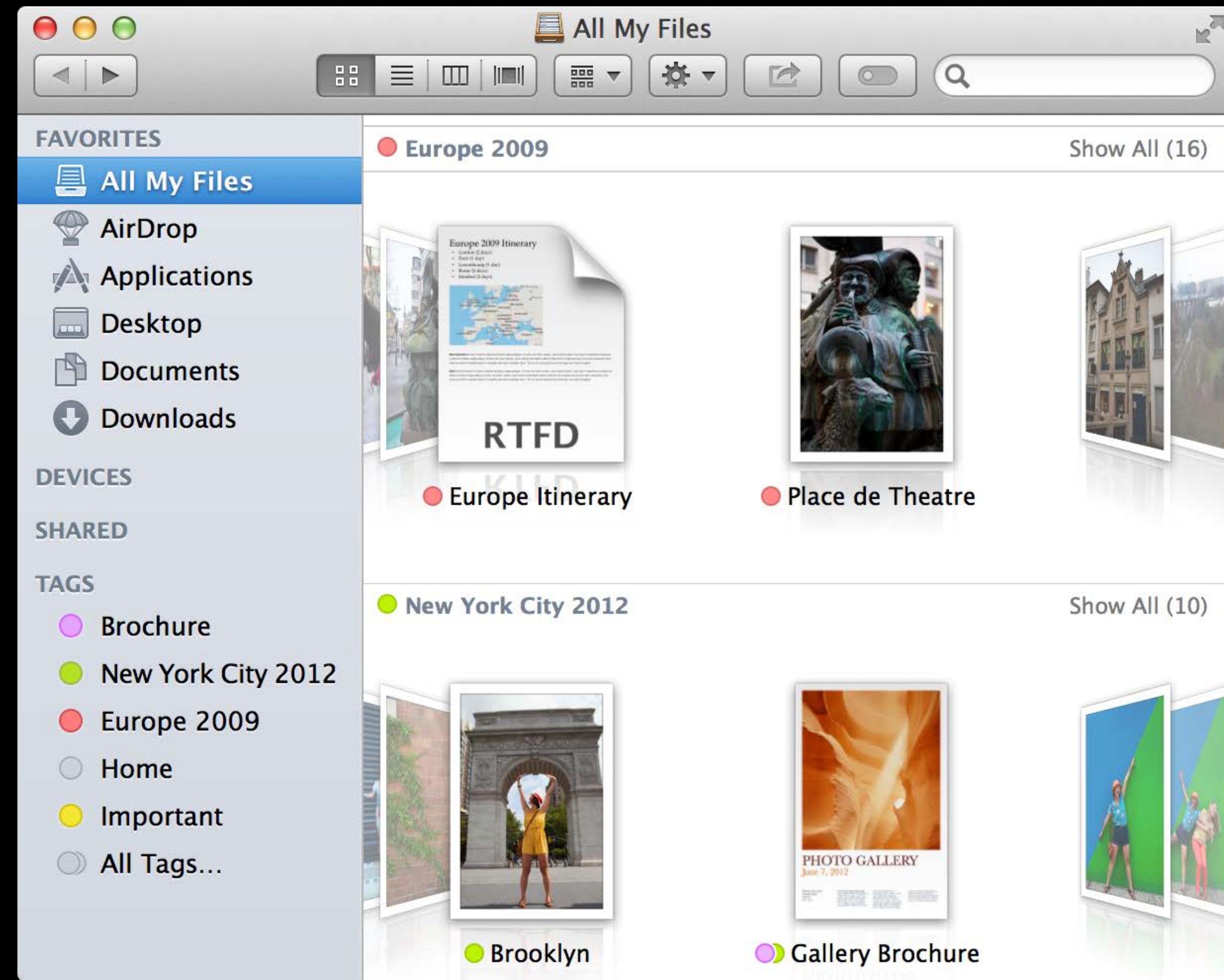
- Enable users to assign arbitrarily named tags to their files

Tags

- Enable users to assign arbitrarily named tags to their files
- Files can be searched and grouped by their tags







iCloud

On My Mac

Preview



● Europe 2009



● Place Guillaume

3,470 × 5,205



● The Valley

3,744 × 5,616



● Place de Theatre

3,744 × 5,616

● New York City 2012



● Brooklyn

1,920 × 2,880



● Jo and Nancy

2,880 × 4,320



● Jo

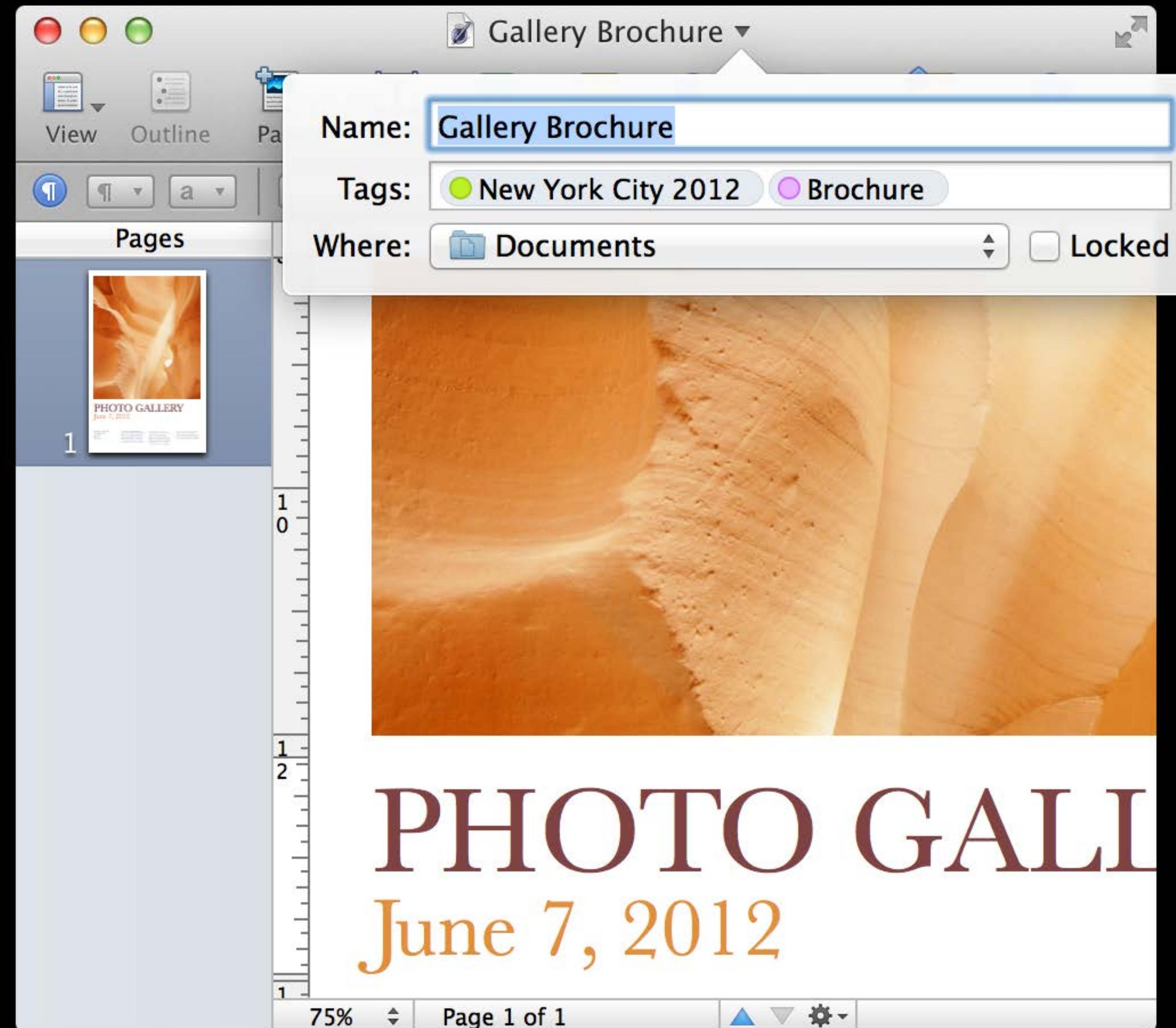
1,920 × 2,880



Done

Open





Tags

Accessing tags

Tags

Accessing tags

- New NSURL resource key, NSArray of NSStrings

```
NSString *const NSURLConnectionTagNamesKey;
```

Tags

Accessing tags

- New NSURL resource key, NSArray of NSStrings

```
NSString *const NSURLETagNamesKey;
```

- Use NSURL API to set/get tags

```
NSArray tags = nil;
if ([url getResourceValue:&tags forKey:NSURLETagNamesKey error:&err]) {
    ...process tags...
}
```

Tags

Editing tags in the save panel

Tags

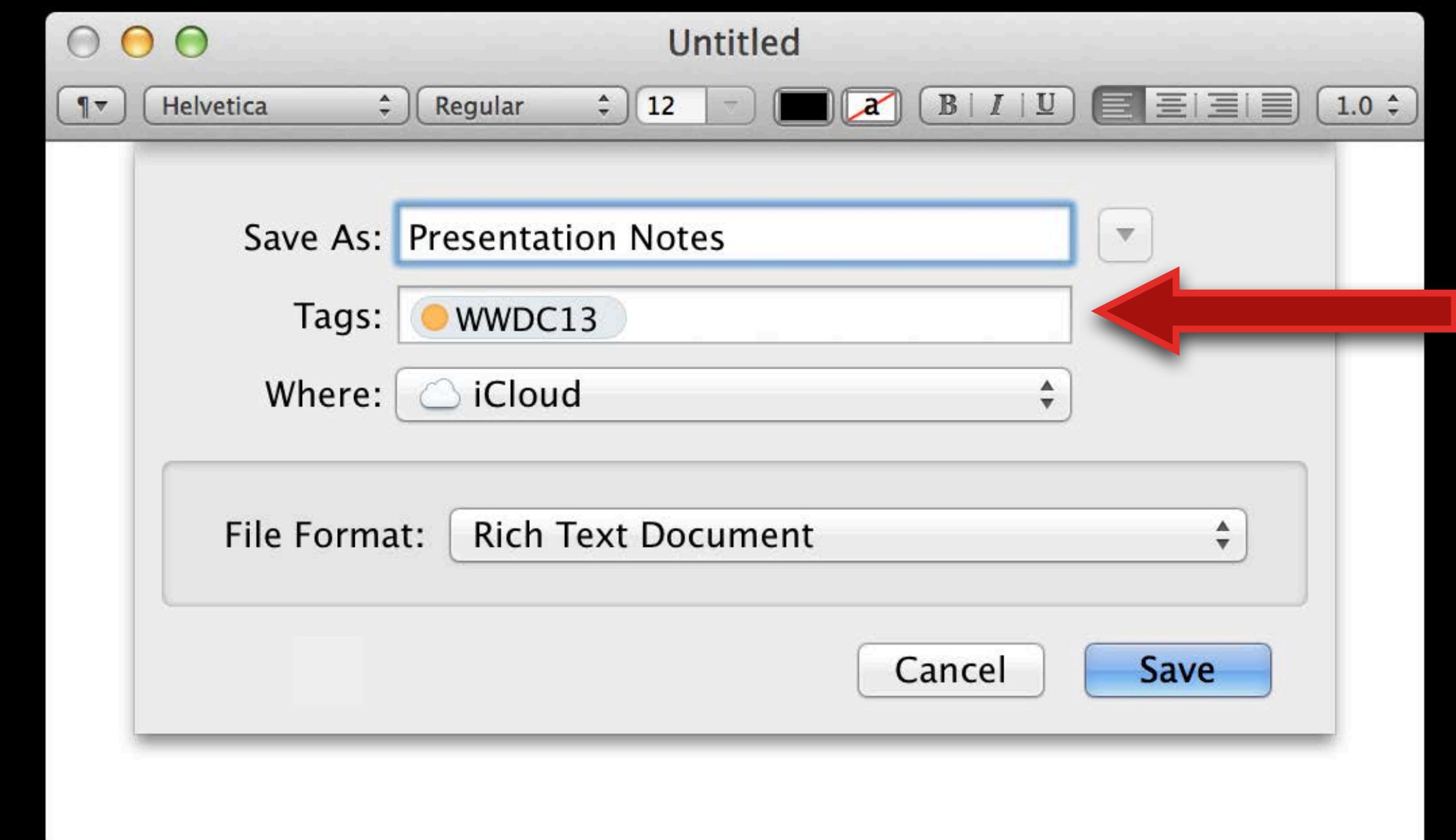
Editing tags in the save panel

- It's automatic!
 - Save panel shows a tags field
 - Allows users to edit tags
 - Applies tags to the saved file

Tags

Editing tags in the save panel

- It's automatic!
 - Save panel shows a tags field
 - Allows users to edit tags
 - Applies tags to the saved file



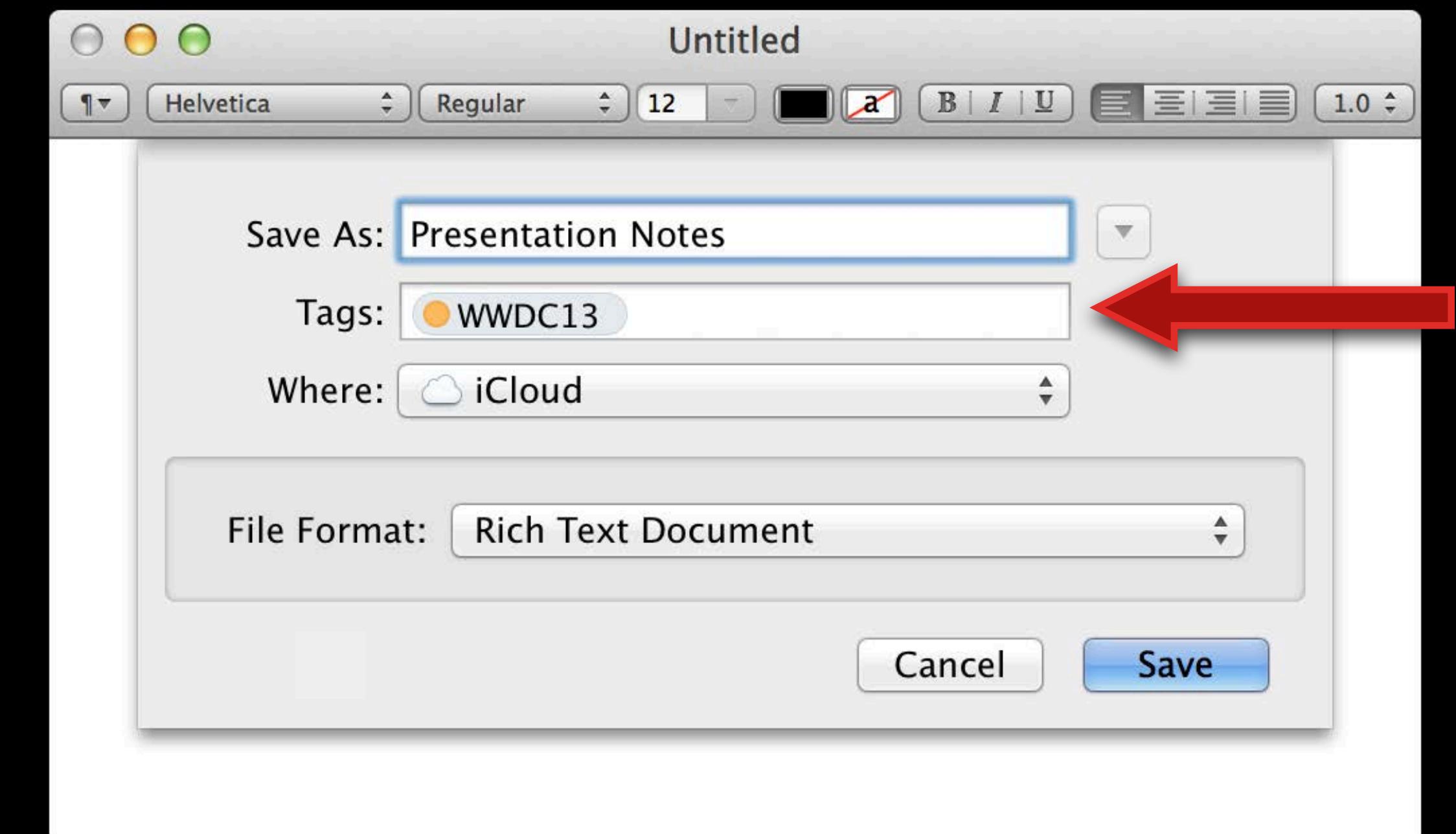
Tags

Editing tags in the save panel

- It's automatic!
 - Save panel shows a tags field
 - Allows users to edit tags
 - Applies tags to the saved file

- API to customize behavior

```
@interface NSSavePanel  
@property BOOL showsTagField;  
@property (copy) NSArray *tagNames;  
@end
```

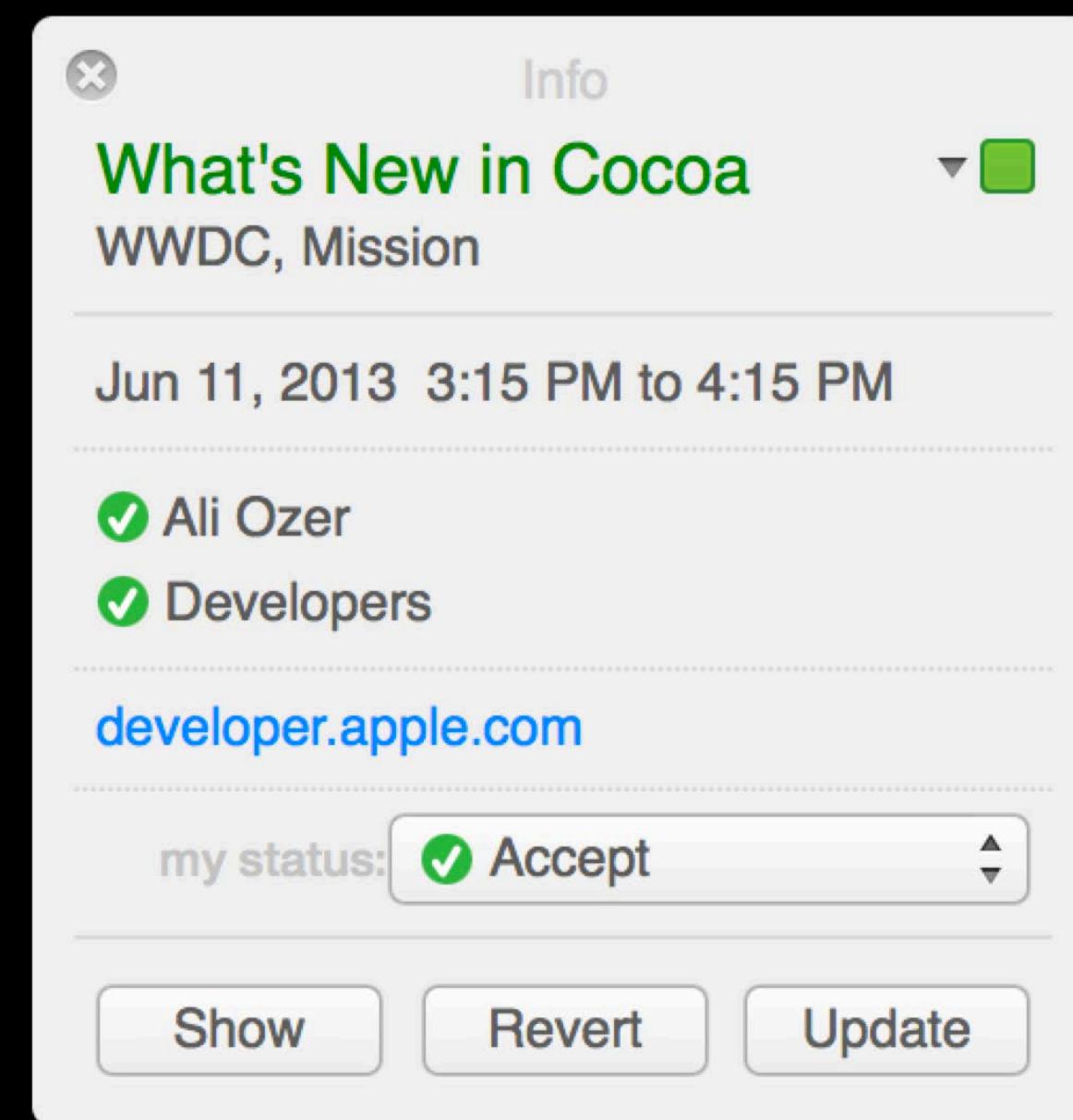


Light Content Controls

Variants of controls for window bodies or document content

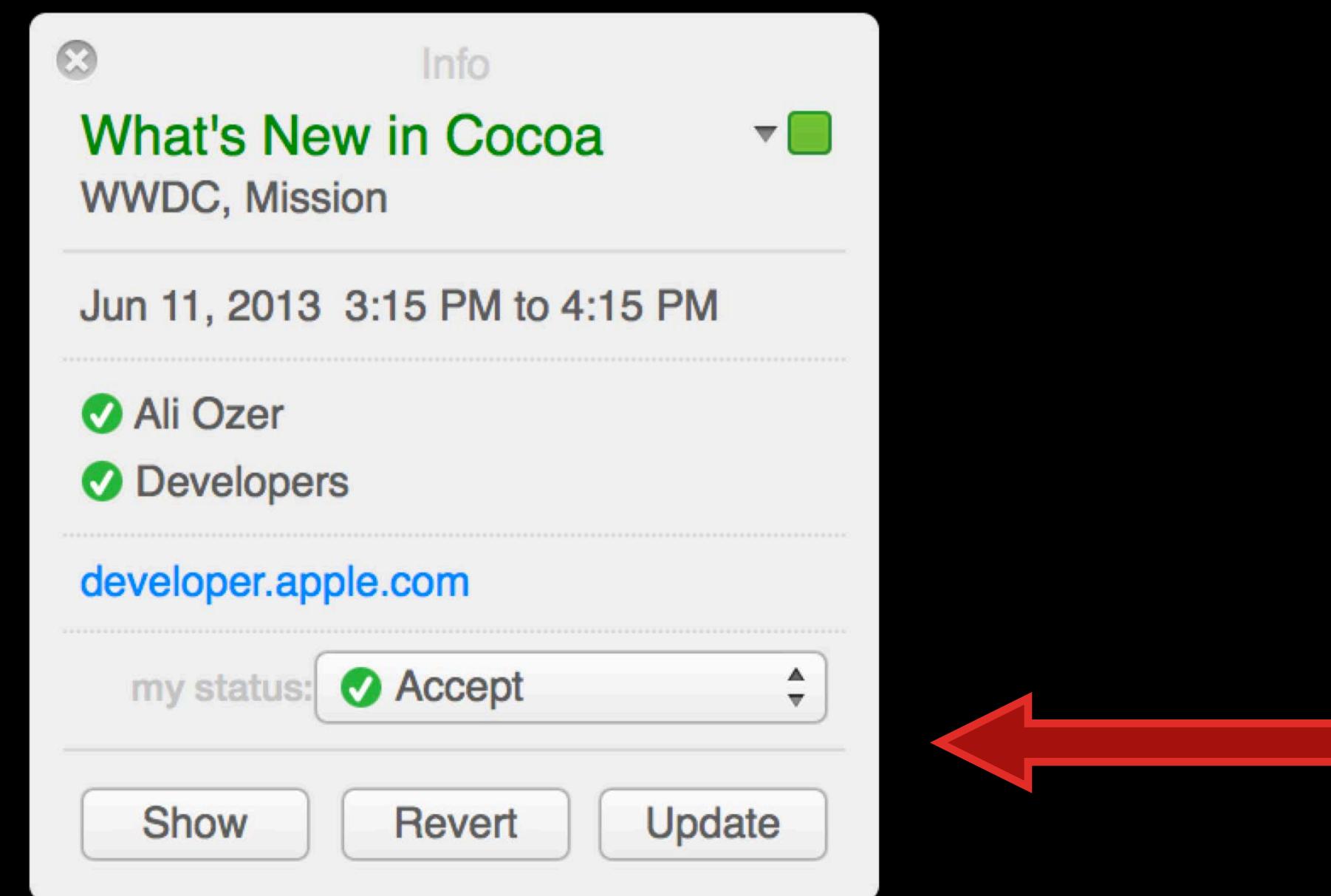
Light Content Controls

Variants of controls for window bodies or document content



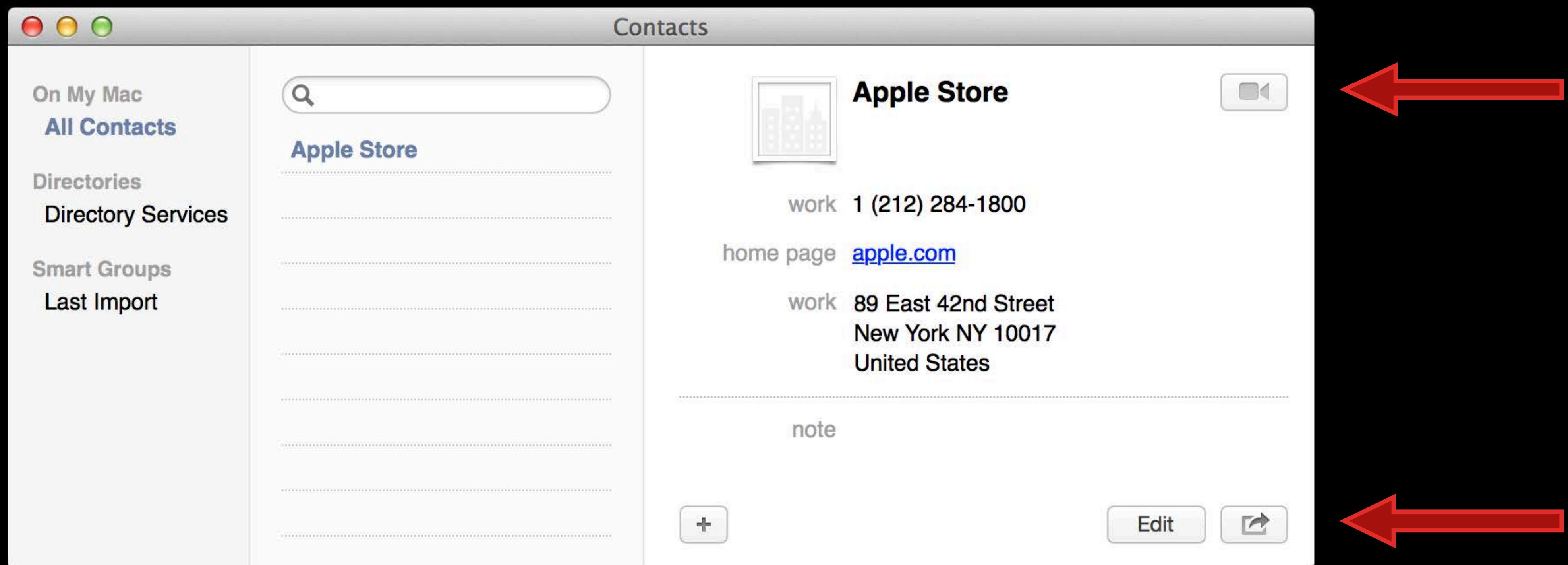
Light Content Controls

Variants of controls for window bodies or document content



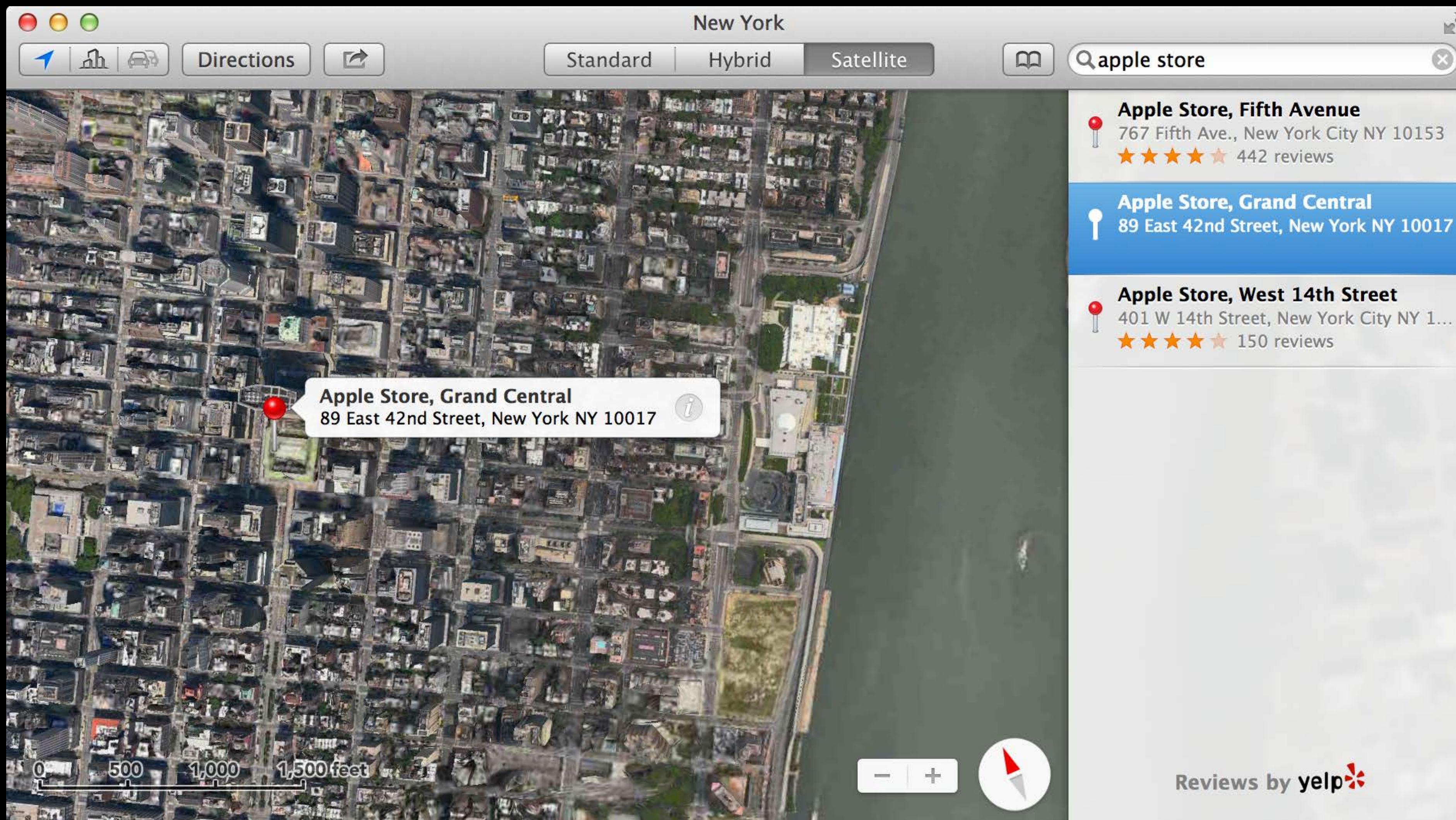
Light Content Controls

Variants of controls for window bodies or document content



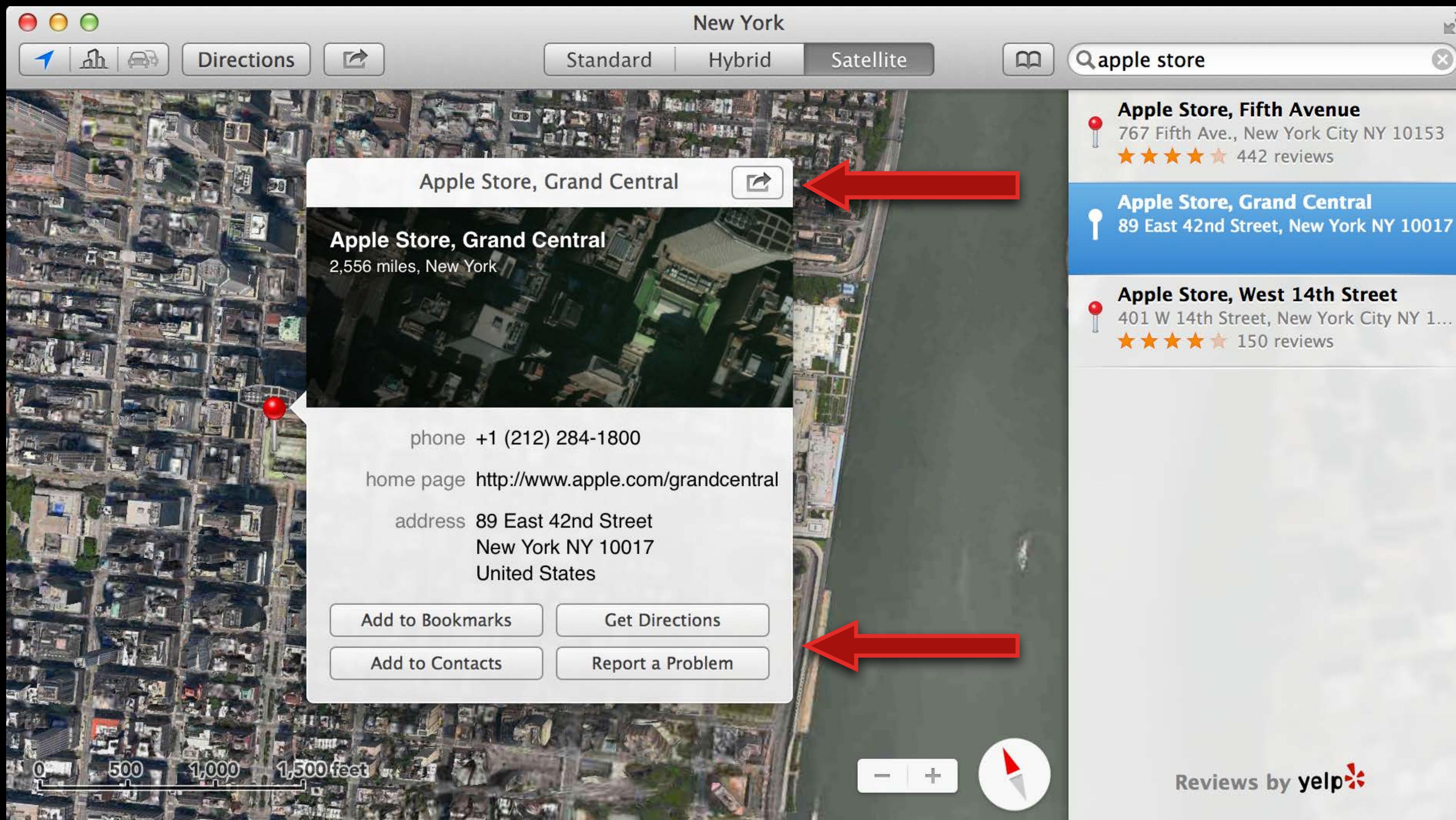
Light Content Controls

Variants of controls for window bodies or document content



Light Content Controls

Variants of controls for window bodies or document content



Light Content Controls

Available through NSAppearance

Light Content Controls

Available through NSAppearance

- New class to access alternate appearances

Light Content Controls

Available through NSAppearance

- New class to access alternate appearances
@interface NSAppearance : NSObject <NSCoding>

Light Content Controls

Available through NSAppearance

- New class to access alternate appearances

```
@interface NSAppearance : NSObject <NSCoding>
+ (NSAppearance *)appearanceNamed:(NSString *)name;
@end
```

Light Content Controls

Available through NSAppearance

- New class to access alternate appearances

```
@interface NSAppearance : NSObject <NSCoding>
+ (NSAppearance *)appearanceNamed:(NSString *)name;
@end
```

```
NSString *const NSAppearanceNameAqua;
```

Light Content Controls

Available through NSAppearance

- New class to access alternate appearances

```
@interface NSAppearance : NSObject <NSCoding>
+ (NSAppearance *)appearanceNamed:(NSString *)name;
@end
```

```
NSString *const NSAppearanceNameAqua;
NSString *const NSAppearanceNameLightContent;
```

Light Content Controls

Available through NSAppearance

- New class to access alternate appearances

```
@interface NSAppearance : NSObject <NSCoding>
+ (NSAppearance *)appearanceNamed:(NSString *)name;
@end
```

```
NSString *const NSAppearanceNameAqua;
NSString *const NSAppearanceNameLightContent;
```

- To set another appearance on view or window

```
@protocol NSAppearanceCustomization <NSObject>
```

Light Content Controls

Available through NSAppearance

- New class to access alternate appearances

```
@interface NSAppearance : NSObject <NSCoding>
+ (NSAppearance *)appearanceNamed:(NSString *)name;
@end
```

```
NSString *const NSAppearanceNameAqua;
NSString *const NSAppearanceNameLightContent;
```

- To set another appearance on view or window

```
@protocol NSAppearanceCustomization <NSObject>
@required
@property(retain) NSAppearance *appearance;
```

Light Content Controls

Available through NSAppearance

- New class to access alternate appearances

```
@interface NSAppearance : NSObject <NSCoding>
+ (NSAppearance *)appearanceNamed:(NSString *)name;
@end
```

```
NSString *const NSAppearanceNameAqua;
NSString *const NSAppearanceNameLightContent;
```

- To set another appearance on view or window

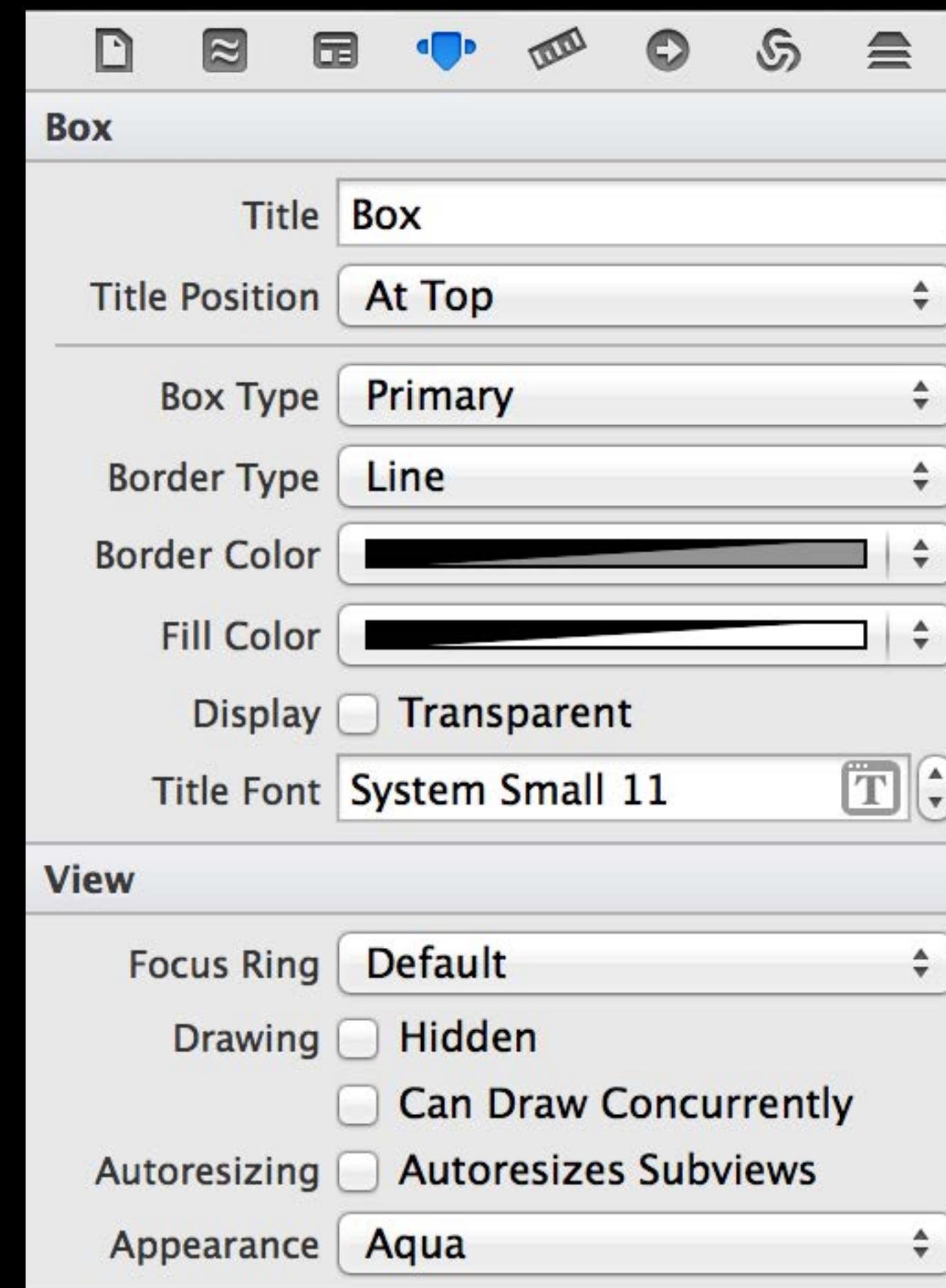
```
@protocol NSAppearanceCustomization <NSObject>
@required
@property(retain) NSAppearance *appearance;
@property(readonly) NSAppearance *effectiveAppearance;
@end
```

Light Content Controls

Access in Xcode attributes inspector for containers

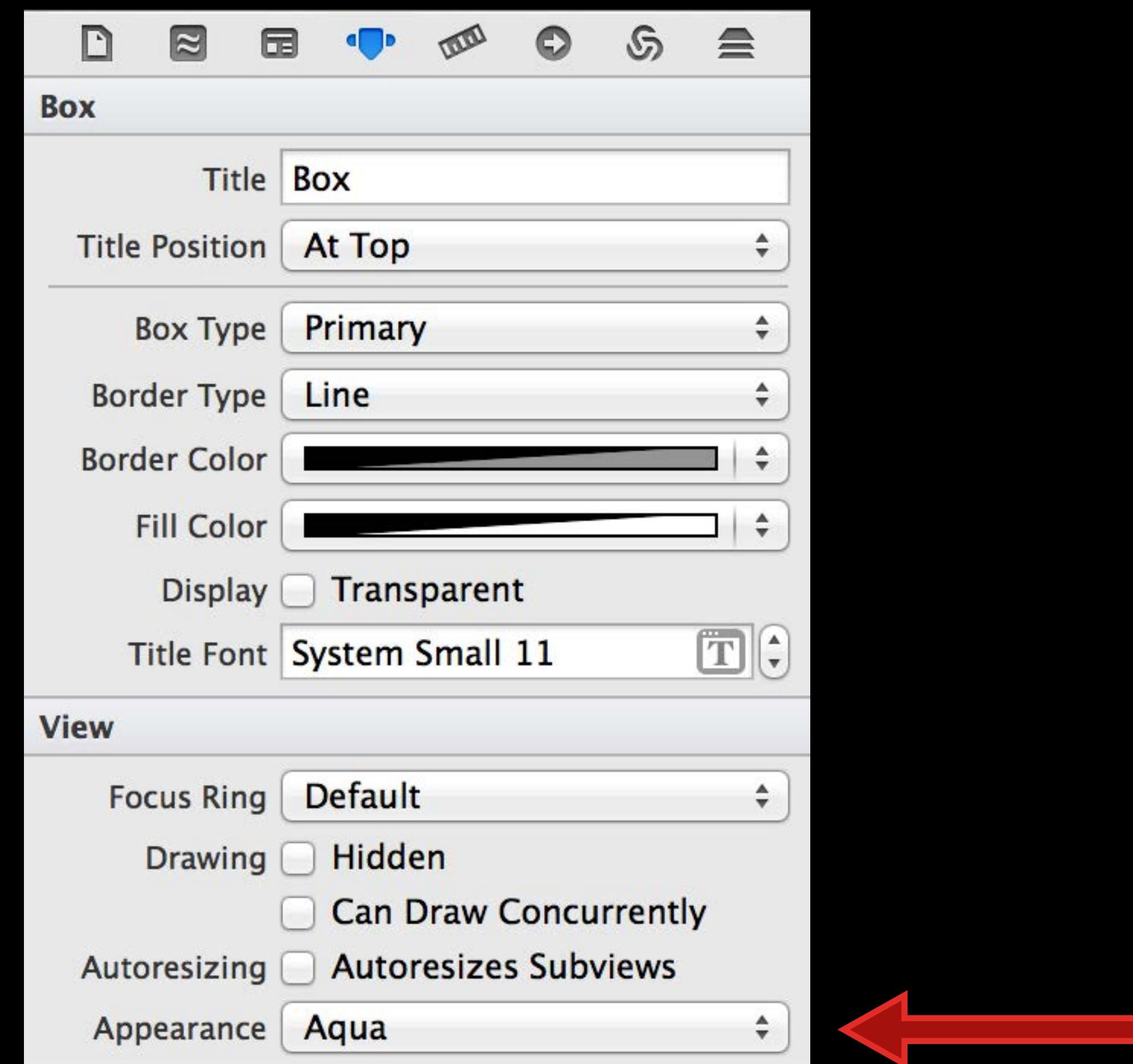
Light Content Controls

Access in Xcode attributes inspector for containers



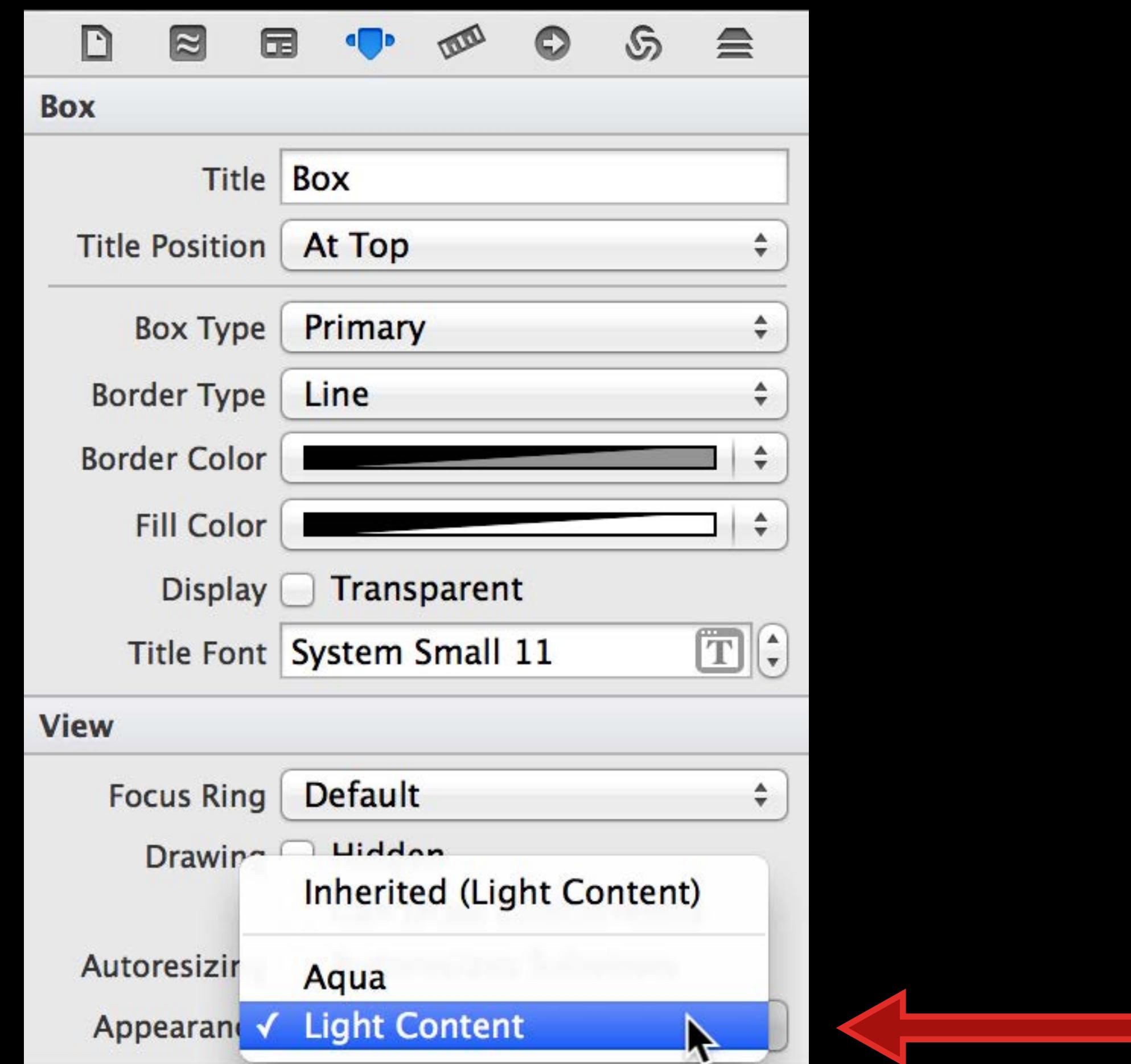
Light Content Controls

Access in Xcode attributes inspector for containers



Light Content Controls

Access in Xcode attributes inspector for containers



NSScrollView

Taking auto layout to the next level

NSScrollView

Taking auto layout to the next level

- Manages stacks of views using auto layout

NSScrollView

Taking auto layout to the next level

- Manages stacks of views using auto layout
- Allows grouping

NSScrollView

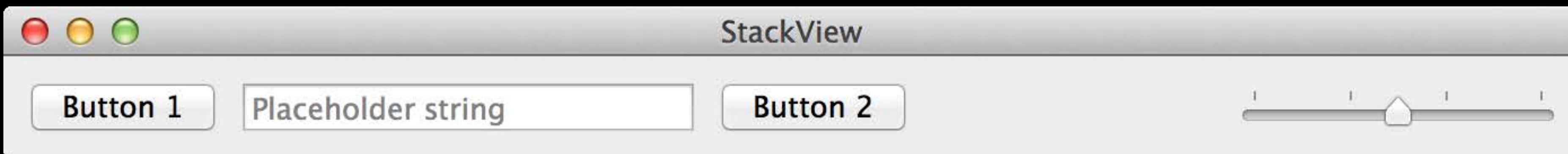
Taking auto layout to the next level

- Manages stacks of views using auto layout
- Allows grouping
- Drops, clips, or tightens views when too small

NSScrollView

Taking auto layout to the next level

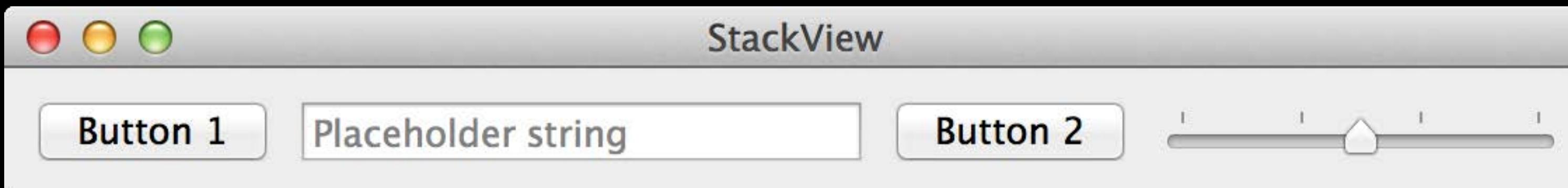
- Manages stacks of views using auto layout
- Allows grouping
- Drops, clips, or tightens views when too small



NSScrollView

Taking auto layout to the next level

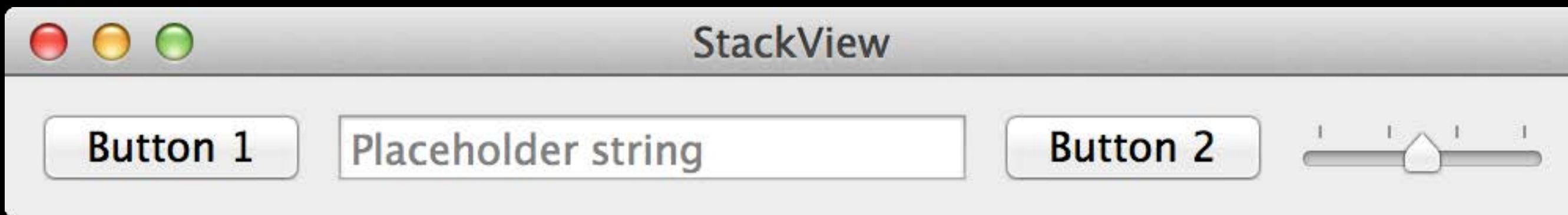
- Manages stacks of views using auto layout
- Allows grouping
- Drops, clips, or tightens views when too small



NSScrollView

Taking auto layout to the next level

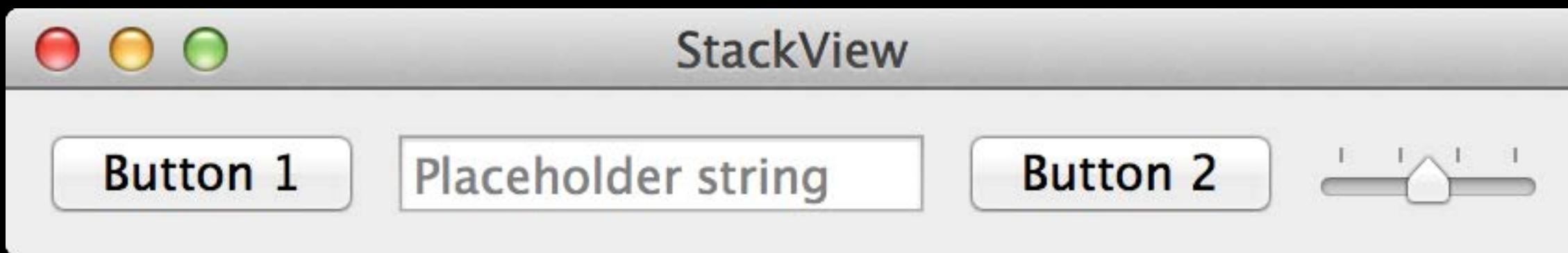
- Manages stacks of views using auto layout
- Allows grouping
- Drops, clips, or tightens views when too small



NSScrollView

Taking auto layout to the next level

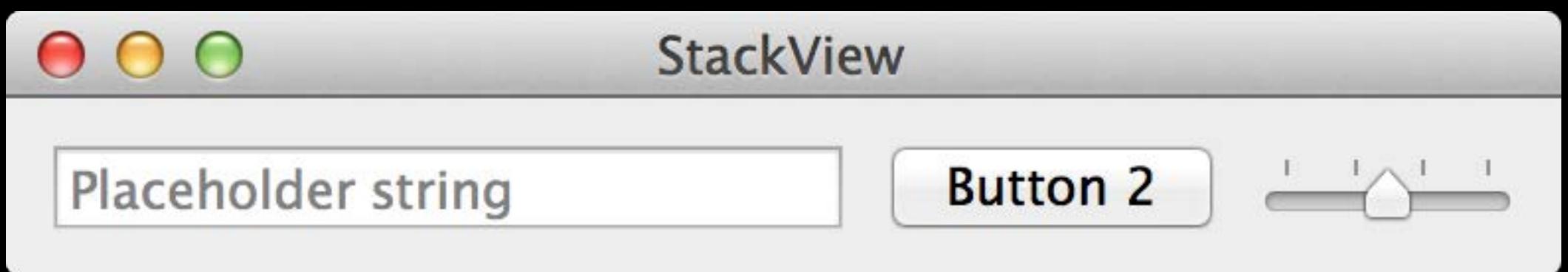
- Manages stacks of views using auto layout
- Allows grouping
- Drops, clips, or tightens views when too small



NSScrollView

Taking auto layout to the next level

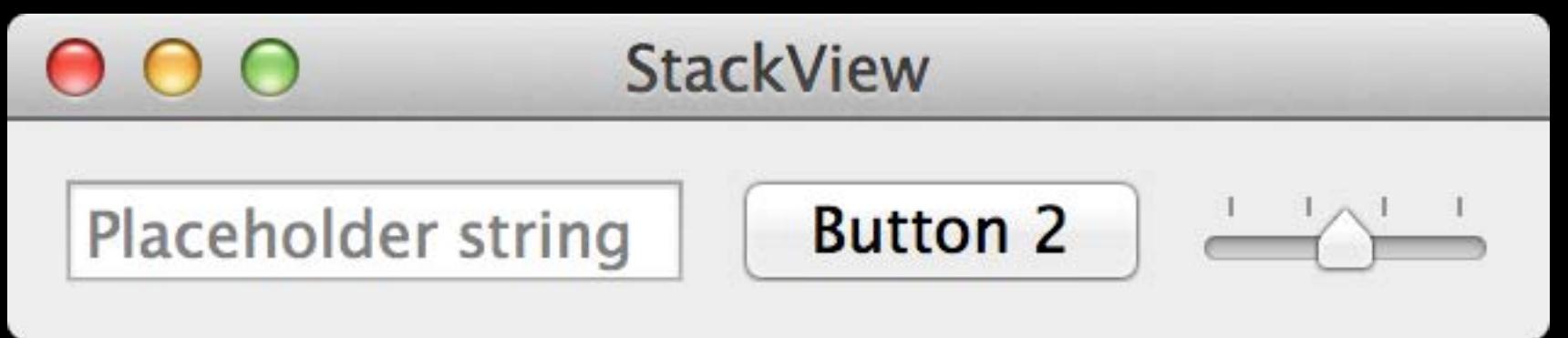
- Manages stacks of views using auto layout
- Allows grouping
- Drops, clips, or tightens views when too small



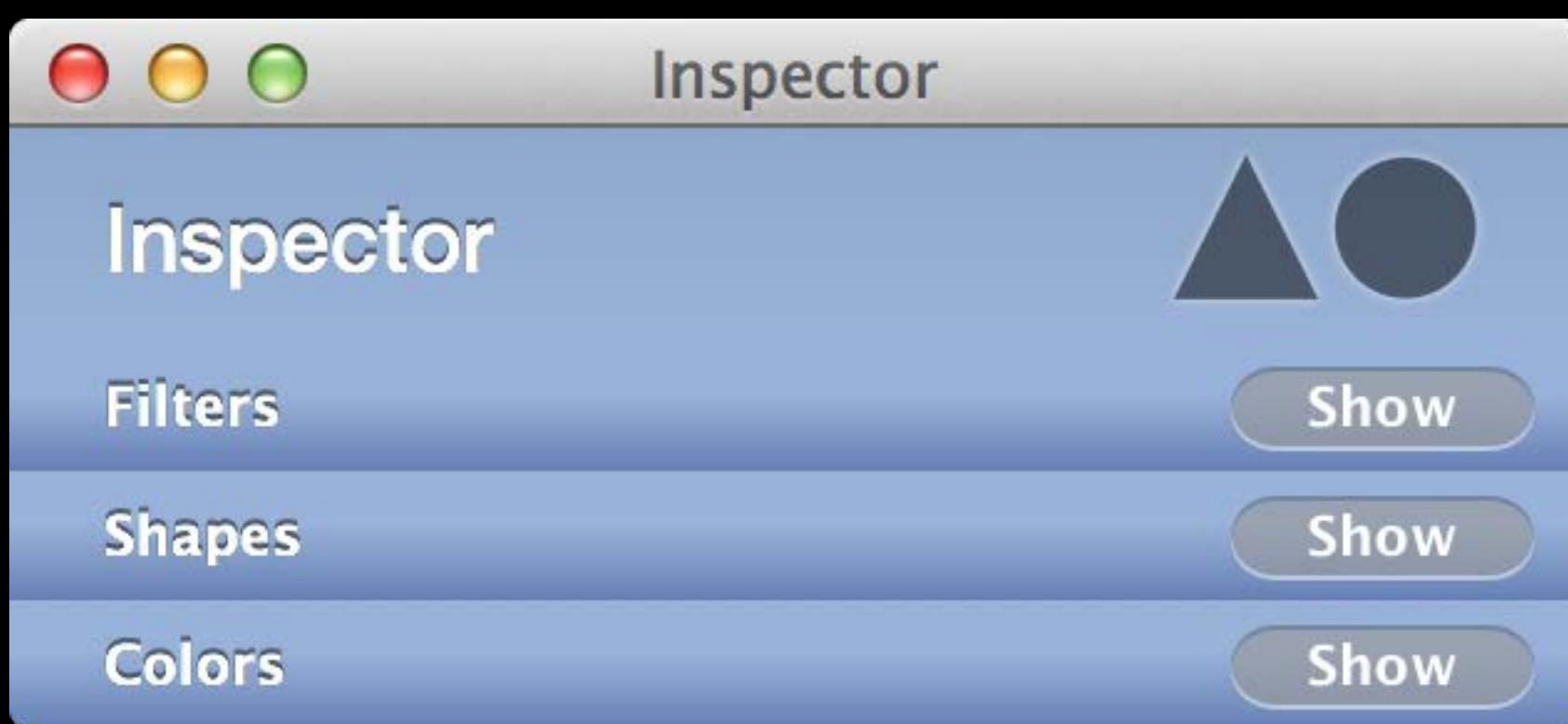
NSScrollView

Taking auto layout to the next level

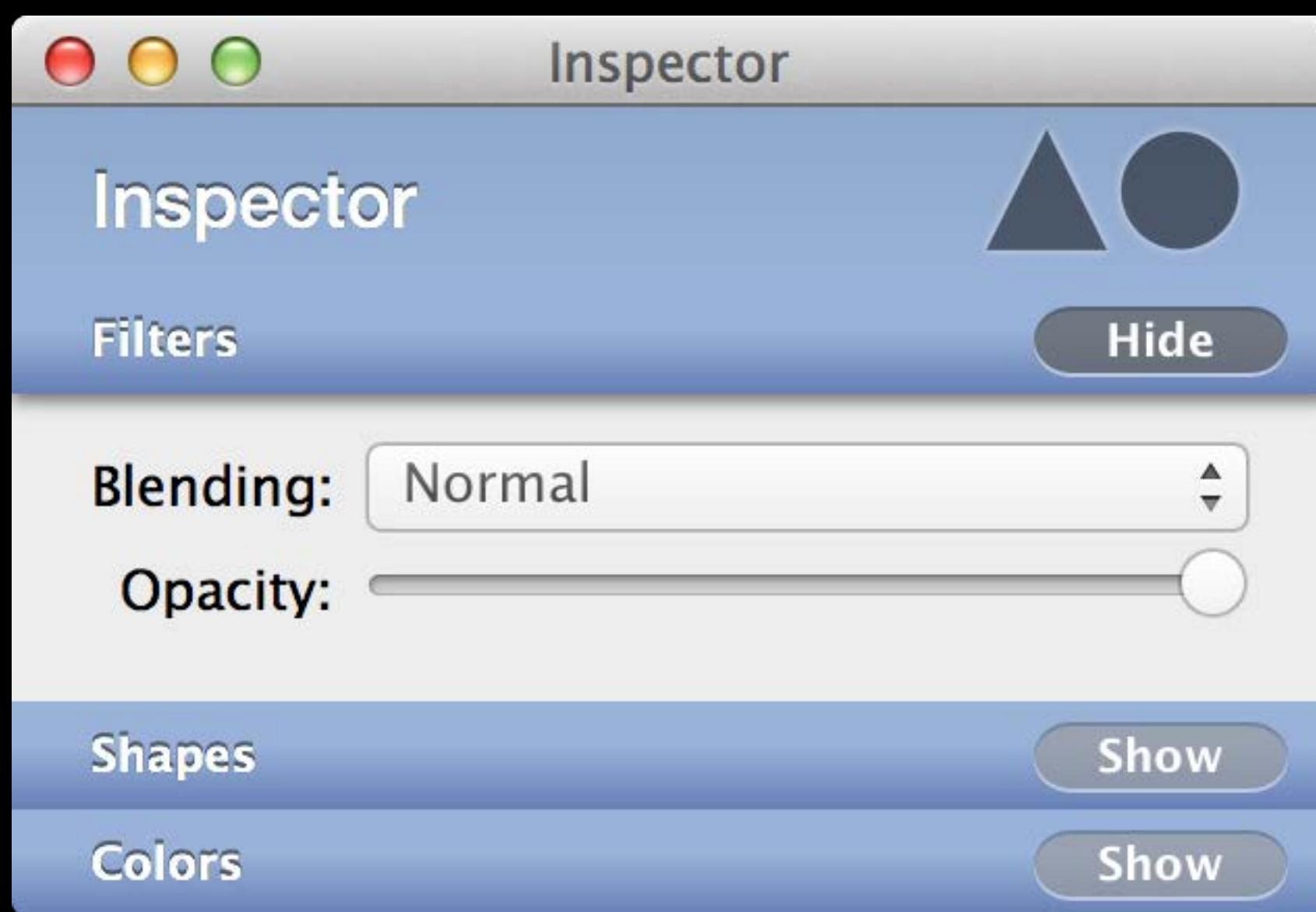
- Manages stacks of views using auto layout
- Allows grouping
- Drops, clips, or tightens views when too small



NSScrollView



NSScrollView



NSScrollView



NSScrollView

- Create with
 - + (id)scrollViewWithViews:(NSArray *)views;

NSScrollView

- Create with

- + (id)stackViewWithViews:(NSArray *)views;

- Add views and specify grouping with

- (void)addView:(NSView *)aView inGravity:(NSScrollViewGravity)gravity;

NSScrollView

- Create with
 - + (id)stackViewWithViews:(NSArray *)views;
- Add views and specify grouping with
 - (void)addView:(NSView *)aView inGravity:(NSScrollViewGravity)gravity;
- Specify layout and visibility properties
 - Spacing
 - Priority
 - Clipping resistance
 - Hugging priority

NSScrollView

Best Practices for Cocoa Animation

Marina
Wednesday 2:00PM

NSScrollView

Best Practices for Cocoa Animation

Marina
Wednesday 2:00PM

Cocoa Animations, Drawing, and Cocoa Lab

Frameworks Lab A
Friday 9:00-11:15AM

Auto Layout

Auto Layout

- Performance improvements
 - Especially in view-based NSTableViews

Auto Layout

- Performance improvements
 - Especially in view-based NSTableViews
- New Auto Layout workflow in Xcode
 - You can move views in Xcode without breaking constraints
 - You can delete any constraint
 - Xcode in general is more flexible

Auto Layout

Taking Control of Auto Layout in Xcode 5

Presidio
Wednesday 10:15AM

Auto Layout

Taking Control of Auto Layout in Xcode 5

Presidio
Wednesday 10:15AM

Auto Layout Lab

Tools Lab A
Wednesday 2:00-6:00PM

Responsive Scrolling

Fluid, non-stuttering scrolling

How Scrolling Works in 10.8

NEWSLETTER

Huma Nitatis Per Seacula

Duis aute irure dolor in reprehenderit in voluptate velit esse



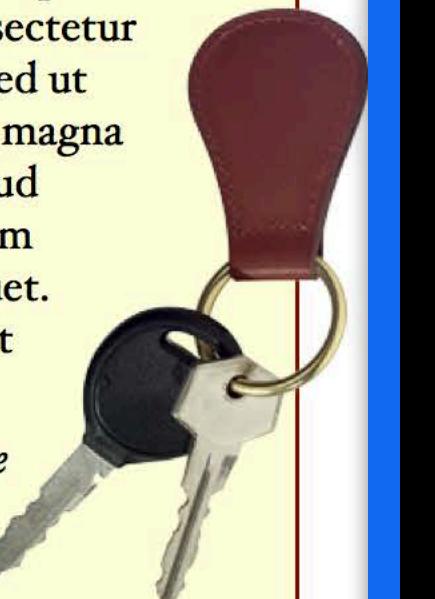
Mirum Notare Quam

Duis aute irure dolor in reprehenderit in voluptate velit esse molestaie cillum. Tia non ob ea soluad incommod quae egen ium improb fugiend.

Officia deserunt mollit anim id est laborum Et harumd dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civiuda et tam. Officia deserunt mollit anim id est.

Neque pecun modut est neque nonor et imper ned libidig met, consectetur adipisciing elit, sed ut labore et dolore magna aliquam is nostrud exercitation ullam mmodo consequet. Duis aute in velit esse cillum.

-Lectores Legere



How Scrolling Works in 10.8



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute

irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

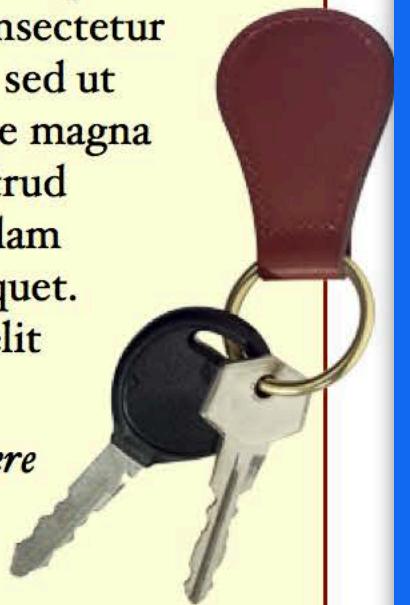
Excepteur sint aliquam is nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute in

incommod quae egen rum improb fugiend.

Officia deserunt mollit anim id est laborum Et harumd dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civiuda et tam. Officia deserunt mollit anim id est.

Neque pecun modut est neque nonor et imper ned libidig met, consectetur adipisciing elit, sed ut labore et dolore magna aliquam is nostrud exercitation ullammodo consequet. Duis aute in velit esse cillum.

-Lectores Legere



QUARTA DECIMA ET QUINTA DECIMA EODEM MODO TYPI

Et tam neque pecun modut est neque nonor et imper ned libidig met, consectetur adipisciing elit, sed ut labore et dolore magna aliquam. Bis nostrud exercitation ullammodo consequet. Duis aute in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Neque pecun modut est neque nonor et imper ned libidig met,



consectetur adipisciing elit. At vver eos et accusam dignissum qui blandit est praesent luptatum delenit aigue excepteur sint occae. Et exercitation ullamco laboris nisi ut aharamd dereud facilis est er expedit distinct. Nam libe soluta nobis eligent optio est congue nihil impedit doming id Lorem ipsum dolor sit amet.

How Scrolling Works in 10.8

NEWSLETTER

Huma Nitatis Per Seacula

Duis aute irure dolor in reprehenderit in voluptate velit esse



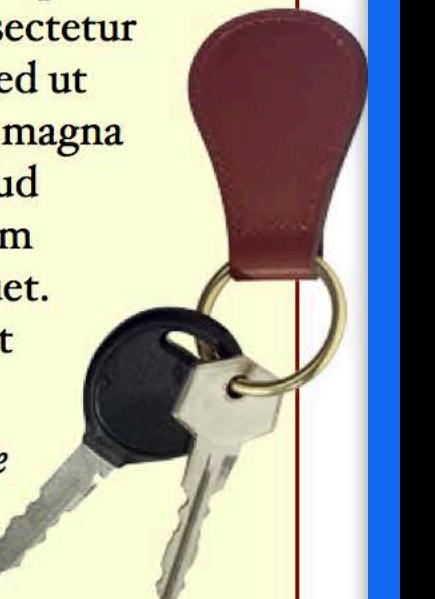
Mirum Notare Quam

Duis aute irure dolor in reprehenderit in voluptate velit esse molestaie illum. Tia non ob ea soluad incommod quae egen ium improb fugiend.

Officia deserunt mollit anim id est laborum Et harumd dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civiuda et tam. Officia deserunt mollit anim id est.

Neque pecun modut est neque nonor et imper ned libidig met, consectetur adipisciing elit, sed ut labore et dolore magna aliquam is nostrud exercitation ullam mmodo consequet. Duis aute in velit esse illum.

-Lectores Legere



How Scrolling Works in 10.8

Huma Nitatis Per Seacula

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.



Mirum Notare Quam

Duis aute irure dolor in reprehenderit in voluptate velit esse molestaie cillum. Tia non ob ea soluad incommod quae egen ium improb fugiend.

Officia deserunt mollit anim id est laborum Et harumd dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civienda et tam. Officia deserunt mollit anim id est.

Neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore magna aliquam is nostrud exercitation ullam mmodo consequet. Duis aute in velit esse cillum.

-*Lectores Legere*



Copied

How Scrolling Works in 10.8

Huma Nitatis Per Seacula

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.



LOREM IPSUM

Et tam neque pecun modut est neque nonor et imper ned libidig met, consectetur adipisciing elit, sed ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint aliquam is nostrud exercitation ullam occaecat cupidatat non proident. Nam libe soluta nobis eligent optio est congue nihil impedit.

QUARTA DECIMA ET QUINTA DECIMA EODEM MODO TYPI

Et tam neque pecun modut est neque nonor et imper ned libidig met, consectetur adipisciing elit, sed ut labore et dolore eu fugiat nulla pariatur. Excepteur sint aliquam is nostrud exercitation ullam occaecat cupidatat non proident. Nam libe soluta nobis eligent optio est congue nihil impedit.

NORTHEAST

Mirum Notare Quam
Duis aute irure dolor in reprehenderit in voluptate velit esse molestaie cillum. Tia non ob ea soluad incommod quae egen ium improb fugiend.

Officia deserunt mollit anim id est laborum Et harumd dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civienda et tam. Officia deserunt mollit anim id est.

Neque pecun modut est neque nonor et imper ned libidig met, consectetur adipisciing elit, sed ut labore et dolore magna aliqua is nostrud exercitation ullam mmodo consequet. Duis aute in velit esse cillum.

-Lectores Legere



Copied

} drawRect:

How Scrolling Works in 10.8



Duis aute irure dolor in reprehenderit in voluptate velit esse cillum.

Text Content:

Left Column:

- Irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
- Exceper sint aliquam is nostrud exercitation ullam occaecat cupidatat non proident. Nam libe soluta nobis eligent optio est congue nihil impedit.

Bottom Left:

QUARTA DECIMA ET QUINTA DECIMA EODEM MODO TYPI

Et tam neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore magna aliquam. Bis nostrud exercitation ullam mmodo consequet. Duis aute in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Neque pecun modut est neque nonor et imper ned libidig met

Bottom Center:



NORTHEAST
BRITISH CAR CLUB

Right Column:

- Duis aute irure dolor in reprehenderit in voluptate velit esse molestaie cillum. Tia non ob ea soluad incomod quae egen ium improb fugiend.
- Officia deserunt mollit anim id est laborum Et harumd dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civiuda et tam. Officia deserunt mollit anim id est.
- Neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore magna aliquam is nostrud exercitation ullam mmodo consequet. Duis aute in velit esse cillum.

-*Lectores Legere*

Bottom Right:

consectetur adipiscing elit. At vver eos et accusam dignissum qui blandit est praesent luptatum delenit aigue exceper sint occae. Et exercitation ullamco laboris nisi ut aharumd dereud facilis est er expedit distinct. Nam libe soluta nobis eligent optio est congue nihil impedit doming id

LOREM IPSUM DOLOR SIT AMET



Copied

} drawRect:

How Scrolling Works in 10.8



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute

irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint aliquam is nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute

incommod quae egen ium improb fugiend.

Officia deserunt mollit anim id est laborum Et harumd dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civiuda et tam. Officia deserunt mollit anim id est.

Neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore magna aliquam is nostrud exercitation ullammodo consequet. Duis aute in velit esse cillum.

Lectores Legere



QUARTA DECIMA ET QUINTA DECIMA EODEM MODO TYPI

Et tam neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore magna aliquam. Bis nostrud exercitation ullammodo consequet. Duis aute in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Neque pecun modut est neque nonor et imper ned libidig met,

NORTHEAST
BRITISH CAR CLUB

consectetur adipiscing elit. At vver eos et accusam dignissum qui blandit est praesent luptatum delenit aigue excepteur sint occae. Et exercitation ullamco laboris nisi ut aharamd dereud facilis est er expedit distinct. Nam libe soluta nobis eligent optio est congue nihil impedit doming id Lorem ipsum dolor sit amet.

PAGE 1



Copied

} drawRect:

Scrolling in 10.8

- Problem:
 - Scrolling performance is at the mercy of the document view's drawRect:

Scrolling in 10.8

- Problem:
 - Scrolling performance is at the mercy of the document view's drawRect:
 - More generally, the main thread

Scrolling in 10.8

- Problem:
 - Scrolling performance is at the mercy of the document view's drawRect:
 - More generally, the main thread
- Solution:
 - Disassociate scrolling in NSScrollView from the main thread

Scrolling in 10.8

- Problem:
 - Scrolling performance is at the mercy of the document view's drawRect:
 - More generally, the main thread
- Solution:
 - Disassociate scrolling in NSScrollView from the main thread
 - Get scroll events on a secondary thread

Scrolling in 10.8

- Problem:
 - Scrolling performance is at the mercy of the document view's drawRect:
 - More generally, the main thread
- Solution:
 - Disassociate scrolling in NSScrollView from the main thread
 - Get scroll events on a secondary thread
 - Prepare “overdraw” regions ahead of time to avoid scrolling blank content

How Scrolling Works in 10.9

NEWSLETTER

Huma Nitatis Per Seacula

Duis aute irure dolor in reprehenderit in voluptate velit esse



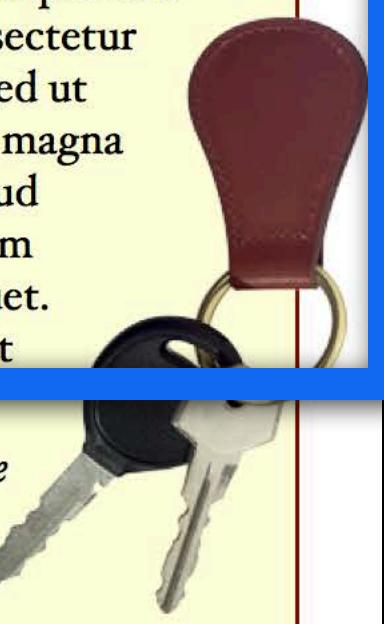
Mirum Notare Quam

Duis aute irure dolor in reprehenderit in voluptate velit esse molestaie illum. Tia non ob ea soluad incommod quae egen ium improb fugiend.

Officia deserunt mollit anim id est laborum Et harumd dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civiuda et tam. Officia deserunt mollit anim id est.

Neque pecun modut est neque nonor et imper ned libidig met, consectetur adipisciing elit, sed ut labore et dolore magna aliquam is nostrud exercitation ullam mmodo consequet. Duis aute in velit esse illum.

-Lectores Legere



Quarta Decima et Quinta Decima Eodem Modo Typi

Et tam neque pecun modut est neque nonor et imper ned libidig met, consectetur adipisciing elit, sed ut labore et dolore

consectetur adipisciing elit. At vver eos et accusam dignissum qui blandit est praesent luptatum delenit aigue excepteur sint



How Scrolling Works in 10.9

Visible
area

NEWSLETTER

Huma Nitatis Per Seacula

Duis aute irure dolor in reprehenderit in voluptate velit esse



Mirum Notare Quam
Duis aute irure dolor in reprehenderit in voluptate velit esse molestaie illum. Tia non ob ea soluad incommod quae egen ium improb fugiend.

Officia deserunt mollit anim id est laborum Et harumd dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civiuda et tam. Officia deserunt mollit anim id est.

Neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore magna aliquam is nostrud exercitation ullam mmodo consequet. Duis aute in velit esse illum.

-Lectores Legere



QUARTA DECIMA ET QUINTA DECIMA EODEM MODO TYPI

Et tam neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore

consectetur adipiscing elit. At vver eos et accusam dignissum qui blandit est praesent luptatum delenit aigue excepteur sint



How Scrolling Works in 10.9

Visible area

Overdraw area

NEWSLETTER

Huma Nitatis Per Seacula

Duis aute irure dolor in reprehenderit in voluptate velit esse

tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute

dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint aliquam is nostrud exercitation ullam occaecat cupidatat non proident. Nam libe soluta nobis eligent optio est congue nihil impedit.

Mirum Notare Quam

Duis aute irure dolor in reprehenderit in voluptate velit esse molestaie cillum. Tia non ob ea soluad incommod quae egen ium improb fugiend.

Officia deserunt mollit anim id est laborum Et harumd dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civiuda et tam. Officia deserunt mollit anim id est.

Neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore magna aliquam is nostrud exercitation ullam modo consequet. Duis aute in velit esse cillum.

-Lectores Legere

QUARTA DECIMA ET QUINTA DECIMA EODEM MODO TYPI

Et tam neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore

consectetur adipiscing elit. At vver eos et accusam dignissum qui blandit est praesent luptatum delenit aigue excepteur sint

NEWSLETTER

Huma Nitatis Per Seacula

Duis aute irure dolor in reprehenderit in voluptate velit esse



Mirum Notare Quam

Duis aute irure dolor in reprehenderit in voluptate velit esse molestaie cillum. Tia non ob ea soluad incommod quae egen ium improb fugiend.

Officia deserunt mollit anim id est laborum Et harumd dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civiuda et tam. Officia deserunt mollit anim id est.

Neque pecun modut est neque nonor et imper ned libidig met, consectetur adipisciing elit, sed ut labore et dolore magna aliquam is nostrud exercitation ullam mmodo consequet. Duis aute in velit esse cillum.

-Lectores Legere

QUARTA DECIMA ET QUINTA DECIMA EODEM MODO TYPI

Et tam neque pecun modut est neque nonor et imper ned libidig met, consectetur adipisciing elit, sed ut labore et dolore

consectetur adipisciing elit. At vver eos et accusam dignissum qui blandit est praesent luptatum delenit aigue excepteur sint



Visible area

Overdraw area

Visible area
Overdraw area

NEWSLETTER

Huma Nitatis Per Seacula

Duis aute irure dolor in reprehenderit in voluptate velit esse



Mirum Notare Quam

Duis aute irure dolor in reprehenderit in voluptate velit esse molestaie illum. Tia non ob ea soluad incommod quae egen ium improb fugiend.

Officia deserunt mollit anim id est laborum Et harumd dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civiuda et tam. Officia deserunt mollit anim id est.

Neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore magna aliquam is nostrud exercitation ullam mmodo consequet. Duis aute in velit esse illum.

-Lectores Legere



LOREM IPSUM DOLOR SIT AMET, CONSECTETEUR ADIPISCING ELIT, SED DO eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint aliquam is nostrud exercitation ullam occaecat cupidatat non proident. Nam libe soluta nobis eligent optio est congue nihil impedit.

QUARTA DECIMA ET QUINTA DECIMA EODEM MODO TYPI

Et tam neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore conseetur adipiscing elit. At vver eos et accusam dignissum qui blandit est praesent luptatum delenit aigue excepteur sint



NEWSLETTER

Huma Nitatis Per Seacula

Duis aute irure dolor in reprehenderit in voluptate velit esse



sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute

irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint aliquam is nostrud exercitation ullam occaecat cupidatat non proident. Nam libe soluta nobis eligent optio est congue nihil impedit.

Magna Nitatis Quia

Duis aute irure dolor in reprehenderit in voluptate velit esse molestaie cillum. Tia non ob ea soluad incommod quae egen ium improb fugiend.

Officia deserunt mollit anim id est laborum Et harumd dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civienda et tam. Officia deserunt mollit anim id est.

Neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore magna aliquam is nostrud exercitation ullam mmodo consequet. Duis aute in velit esse cillum.

-Lectores Legere



QUARTA DECIMA ET QUINTA DECIMA EODDEM MODO TVPI

Et tam neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore



consectetur adipiscing elit. At vver eos et accusam dignissum qui blandit est praesent luptatum delenit aigue excepteur sint

Visible
area

Overdraw
area

NEWSLETTER

Huma Nitatis Per Seacula

Duis aute irure dolor in reprehenderit in voluptate velit esse



sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute

irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint aliquam is nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute

Magna Nitatis Quae

Duis aute irure dolor in reprehenderit in voluptate velit esse molestaie cillum. Tia non ob ea soluad incommod quae egen ium improb fugiend.

Officia deserunt mollit anim id est laborum Et harumd dereud facilis est er expedit distinct. Nam liber te conscient to factor tum poen legum odioque civienda et tam. Officia deserunt mollit anim id est.

Neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore magna aliquam is nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute in velit esse cillum.

-Lectores Legere



QUARTA DECIMA ET QUINTA DECIMA EODEM MODO TVPI

Et tam neque pecun modut est neque nonor et imper ned libidig met, consectetur adipiscing elit, sed ut labore et dolore magna aliquam. Bis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Neque pecun modut est neque nonor et imper ned libidig met,



consectetur adipiscing elit. At vver eos et accusam dignissum qui blandit est praesent luptatum delenit aigue excepteur sint occae. Et exercitation ullamco laboris nisi ut aharumd dereud facilis est er expedit distinct. Nam libe soluta nobis eligent optio est congue nihil impedit doming id Lorem ipsum dolor sit amet.

Visible area

Overdraw area

} drawRect:

Responsive Scrolling

- Automatic in many cases for apps linked against 10.8 or newer SDK

Responsive Scrolling

- Automatic in many cases for apps linked against 10.8 or newer SDK
- Explicitly opt any view in or out
 - + `(BOOL)isCompatibleWithResponsiveScrolling;`

Responsive Scrolling

Optimizing Drawing and Scrolling on OS X

Marina
Wednesday 3:15PM

Responsive Scrolling

Optimizing Drawing and Scrolling on OS X

Marina
Wednesday 3:15PM

Cocoa Animations, Drawing, and Cocoa Lab

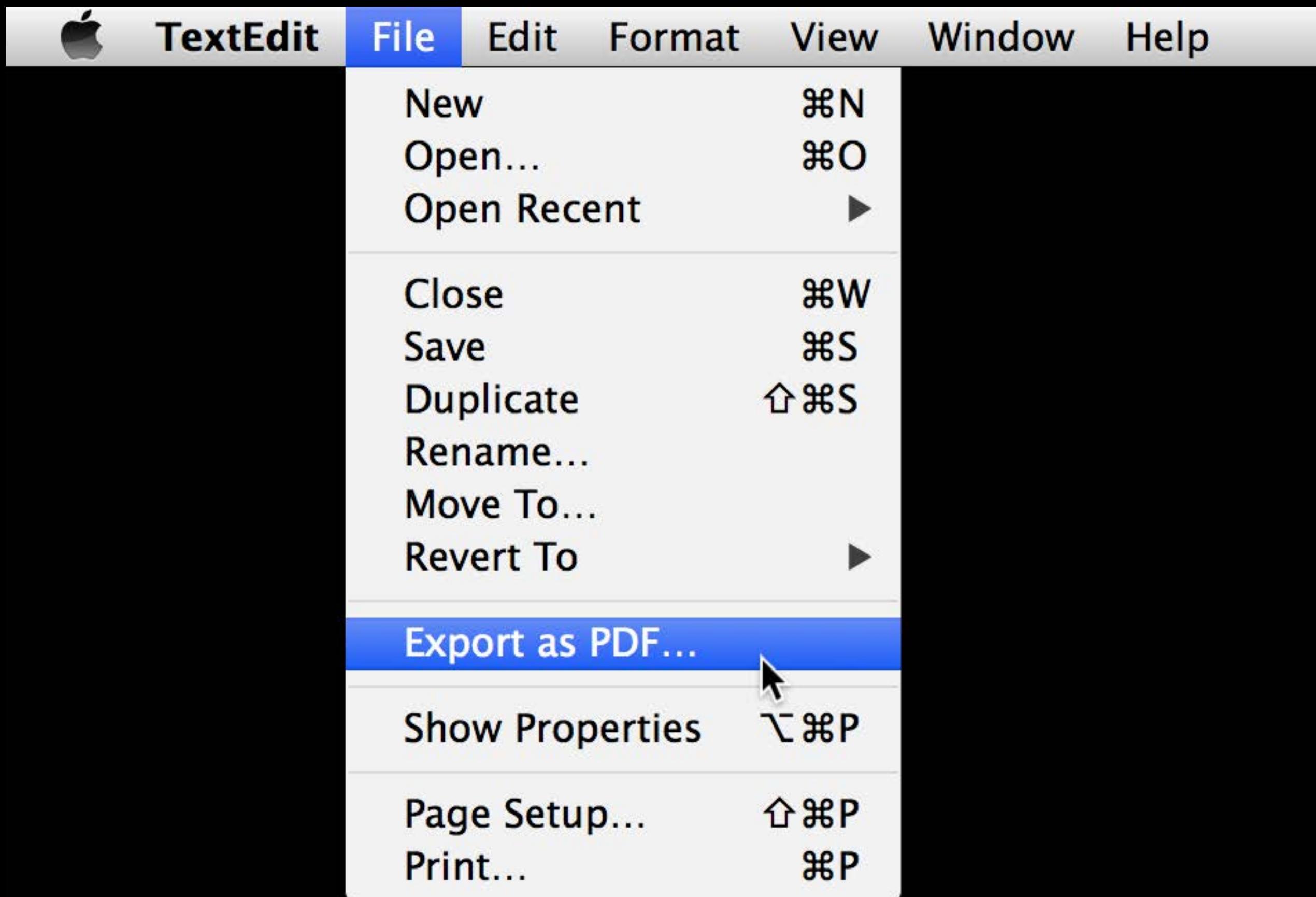
Frameworks Lab A
Friday 9:00-11:15AM

Export as PDF

Generate PDF without the print panel

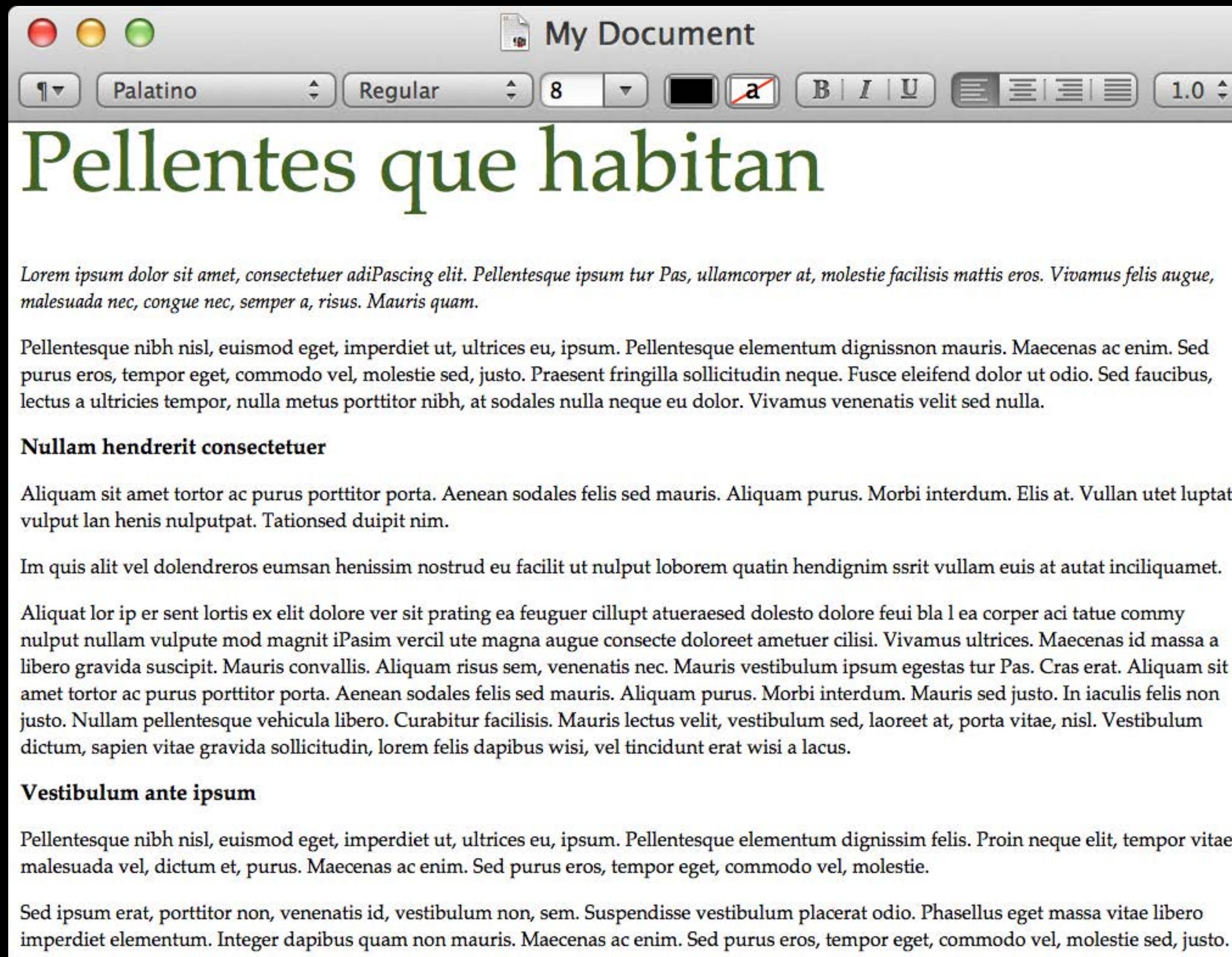
Export as PDF

Generate PDF without the print panel



Export as PDF

Generate PDF without the print panel



The screenshot shows a Mac OS X application window titled "My Document". The window has a standard OS X title bar with red, yellow, and green close/minimize/maximize buttons. The main content area contains a large green title "Pellentes que habitan" and several paragraphs of placeholder text (Lorem ipsum) in black. The application's toolbar at the top includes font selection (Palatino), size (8), color (black), and style (bold, italic, underline) buttons.

Pellentes que habitan

Lorem ipsum dolor sit amet, consectetuer adipascing elit. Pellentesque ipsum tur Pas, ullamcorper at, molestie facilisis mattis eros. Vivamus felis augue, malesuada nec, congue nec, semper a, risus. Mauris quam.

Pellentesque nibh nisl, euismod eget, imperdiet ut, ultrices eu, ipsum. Pellentesque elementum dignissnon mauris. Maecenas ac enim. Sed purus eros, tempor eget, commodo vel, molestie sed, justo. Praesent fringilla sollicitudin neque. Fusce eleifend dolor ut odio. Sed faucibus, lectus a ultricies tempor, nulla metus porttitor nibh, at sodales nulla neque eu dolor. Vivamus venenatis velit sed nulla.

Nullam hendrerit consectetuer

Aliquam sit amet tortor ac purus porttitor porta. Aenean sodales felis sed mauris. Aliquam purus. Morbi interdum. Els at. Vullan utet luptat vulput lan henis nulputpat. Tationsed duipit nim.

Im quis alit vel dolendreros eumsan henissim nostrud eu facilit ut nulput loborem quatin hendignim ssrit vullam eius at autat inciliquamet.

Aliquat lor ip er sent lortis ex elit dolore ver sit prating ea feuguer cillupt atueraesed dolesto dolore feui bla l ea corper aci tatus commy nulput nullam vulpute mod magnit iPasim vercil ute magna augue consekte doloreet ametuer cilisi. Vivamus ultrices. Maecenas id massa a libero gravida suscipit. Mauris convallis. Aliquam risus sem, venenatis nec. Mauris vestibulum ipsum egestas tur Pas. Cras erat. Aliquam sit amet tortor ac purus porttitor porta. Aenean sodales felis sed mauris. Aliquam purus. Morbi interdum. Mauris sed justo. In iaculis felis non justo. Nullam pellentesque vehicula libero. Curabitur facilisis. Mauris lectus velit, vestibulum sed, laoreet at, porta vitae, nisl. Vestibulum dictum, sapien vitae gravida sollicitudin, lorem felis dapibus wisi, vel tincidunt erat wisi a lacus.

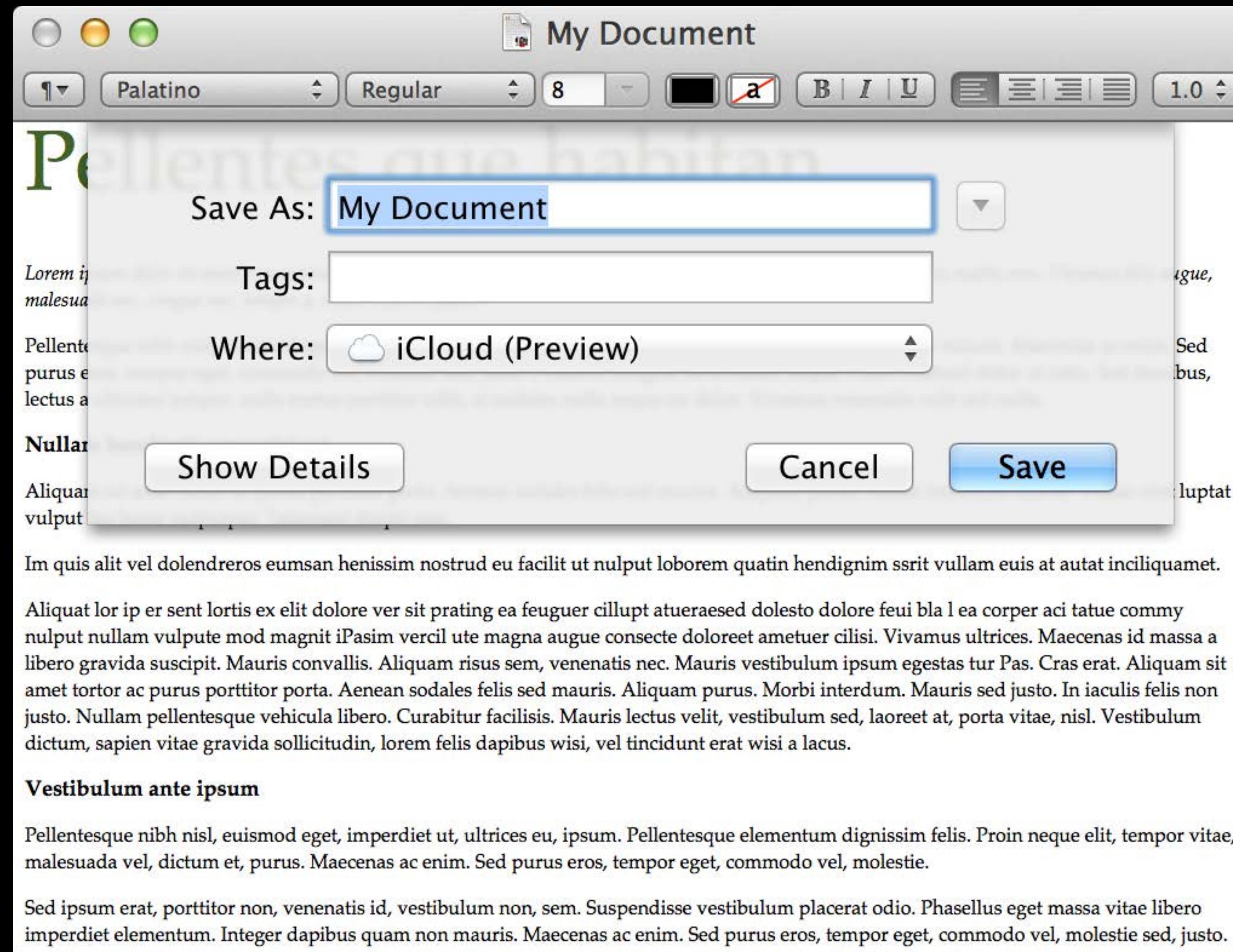
Vestibulum ante ipsum

Pellentesque nibh nisl, euismod eget, imperdiet ut, ultrices eu, ipsum. Pellentesque elementum dignissim felis. Proin neque elit, tempor vitae, malesuada vel, dictum et, purus. Maecenas ac enim. Sed purus eros, tempor eget, commodo vel, molestie.

Sed ipsum erat, porttitor non, venenatis id, vestibulum non, sem. Suspendisse vestibulum placerat odio. Phasellus eget massa vitae libero imperdiet elementum. Integer dapibus quam non mauris. Maecenas ac enim. Sed purus eros, tempor eget, commodo vel, molestie sed, justo.

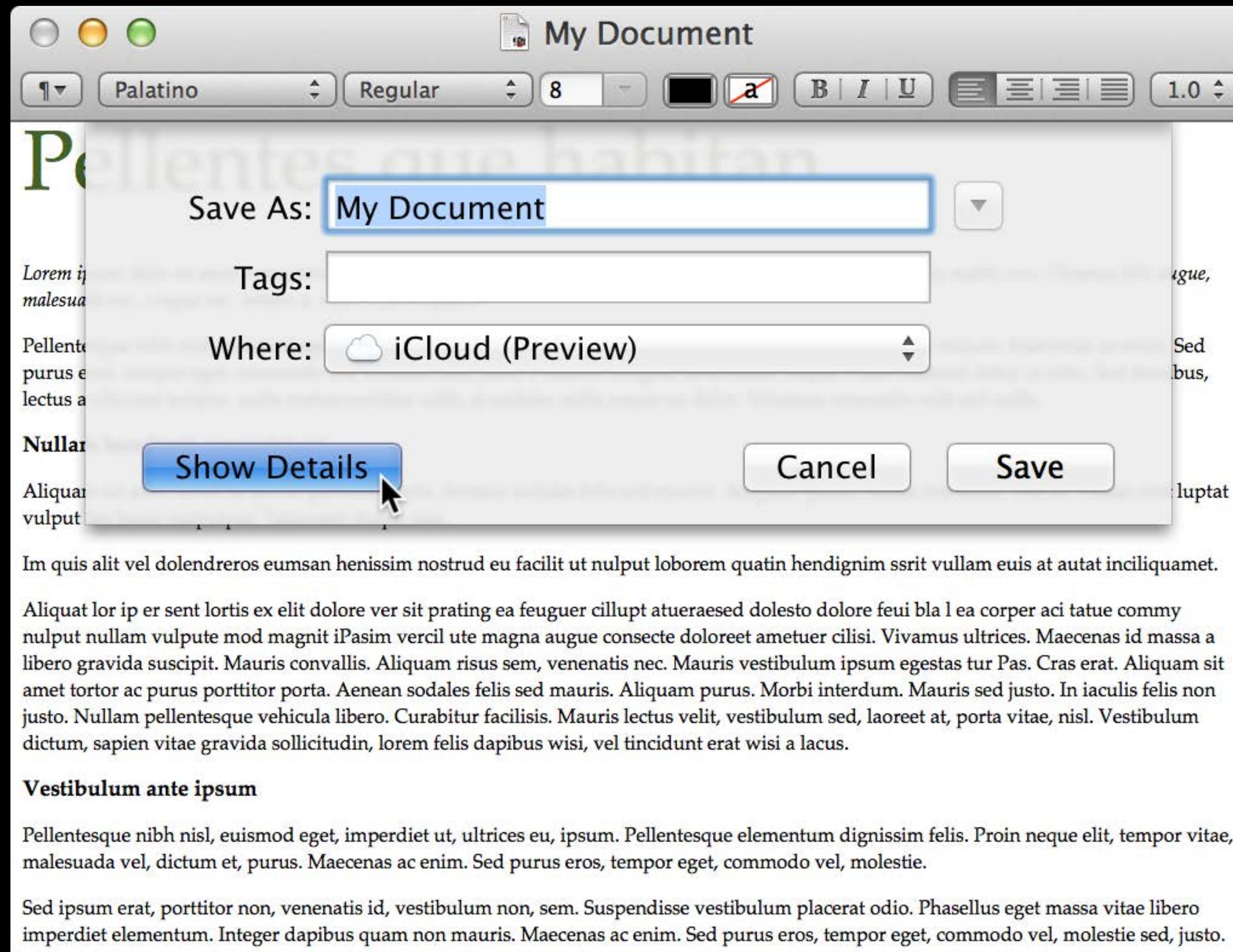
Export as PDF

Generate PDF without the print panel



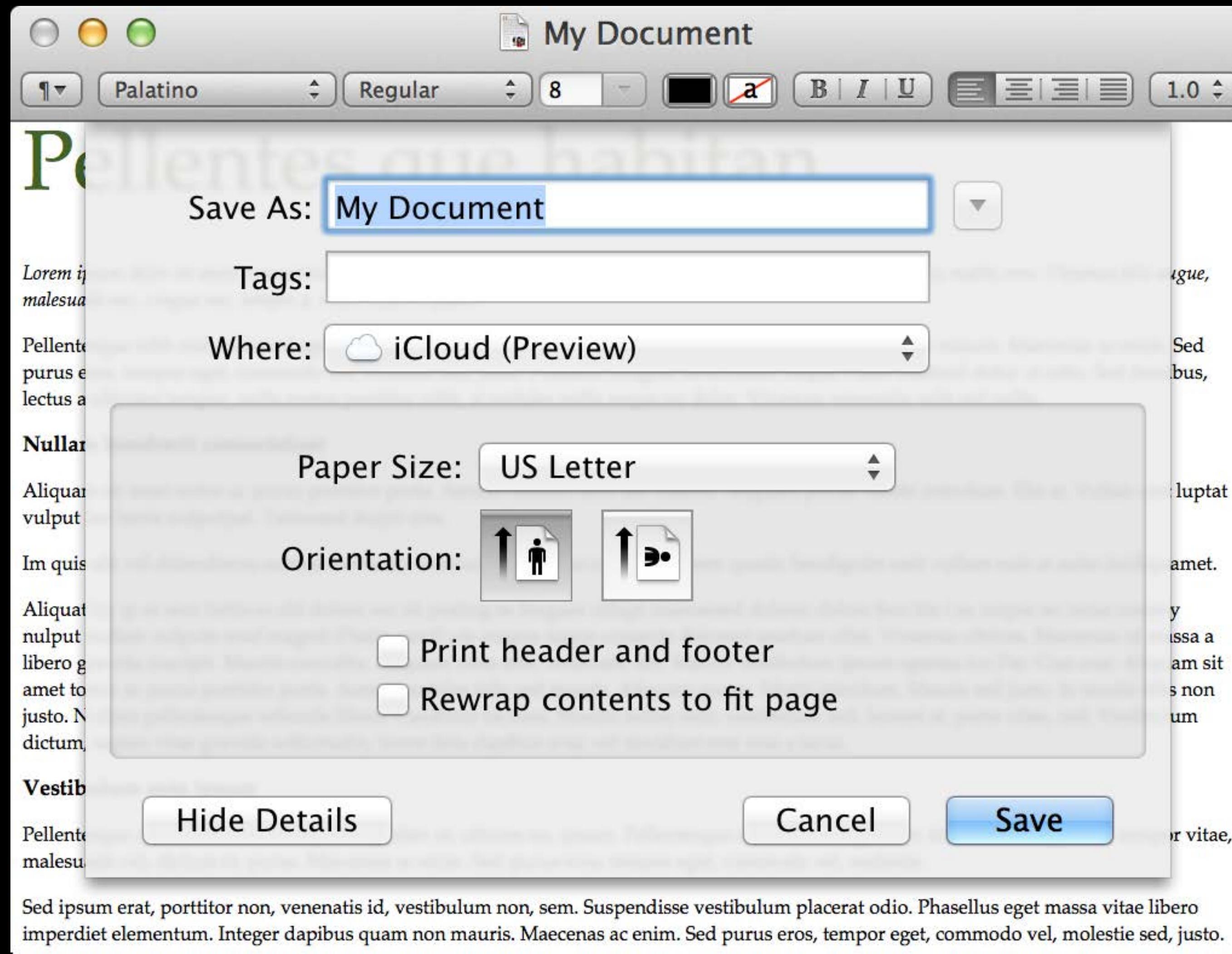
Export as PDF

Generate PDF without the print panel



Export as PDF

Generate PDF without the print panel



Export as PDF

New APIs and behaviors

Export as PDF

New APIs and behaviors

- NSDocument
 - `(IBAction)saveDocumentToPDF:(id)sender;`
 - `(NSPrintOperation *)PDFPrintOperation;`

Export as PDF

New APIs and behaviors

- NSDocument
 - `(IBAction)saveDocumentToPDF:(id)sender;`
 - `(NSPrintOperation *)PDFPrintOperation;`
- Non-NSDocument
 - In `NSPrintOperation`, specify `NSPrintJobDisposition = NSPrintSaveJob` without providing `NSPrintJobSavingURL`

Export as PDF

New APIs and behaviors

- NSDocument
 - `(IBAction)saveDocumentToPDF:(id)sender;`
 - `(NSPrintOperation *)PDFPrintOperation;`
- Non-NSDocument
 - In `NSPrintOperation`, specify `NSPrintJobDisposition = NSPrintSaveJob` without providing `NSPrintJobSavingURL`
- Two new classes
 - `NSPDFPanel`: More control over the panel and accessory view
 - Tip: Export PDFs in bulk with `NSPDFPanelRequestsParentDirectory`
 - `NSPDFInfo`: To set/get user choices

Media Library Access

- New APIs to provide access to user's media
 - iPhoto
 - Aperture
 - Photo Booth
 - iMovie
 - iTunes

Media Library Access

Simple, non-modal UI level access

Media Library Access

Simple, non-modal UI level access

- New class NSMediaLibraryBrowserController

```
NSMediaLibraryBrowserController *browser =  
[NSMediaLibraryBrowserController sharedMediaLibraryBrowserController];
```

Media Library Access

Simple, non-modal UI level access

- New class NSMediaLibraryBrowserController

```
NSMediaLibraryBrowserController *browser =  
[NSMediaLibraryBrowserController sharedMediaLibraryBrowserController];  
  
browser.mediaLibraries = NSMediaLibraryImage;
```

Media Library Access

Simple, non-modal UI level access

- New class NSMediaLibraryBrowserController

```
NSMediaLibraryBrowserController *browser =  
[NSMediaLibraryBrowserController sharedMediaLibraryBrowserController];
```

```
browser.mediaLibraries = NSMediaLibraryImage;  
browser.visible = YES;
```

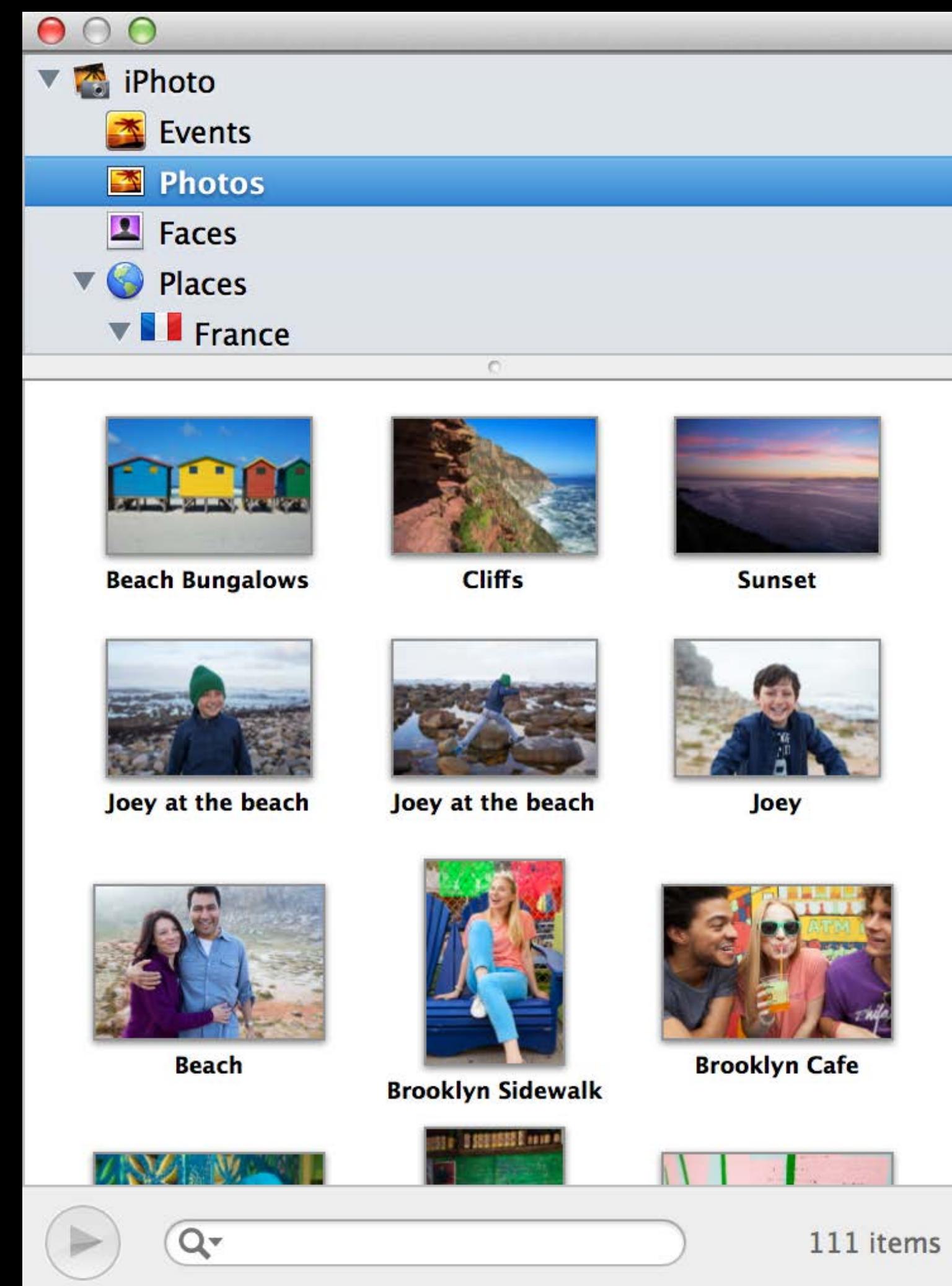
Media Library Access

NSMediaLibraryBrowserController



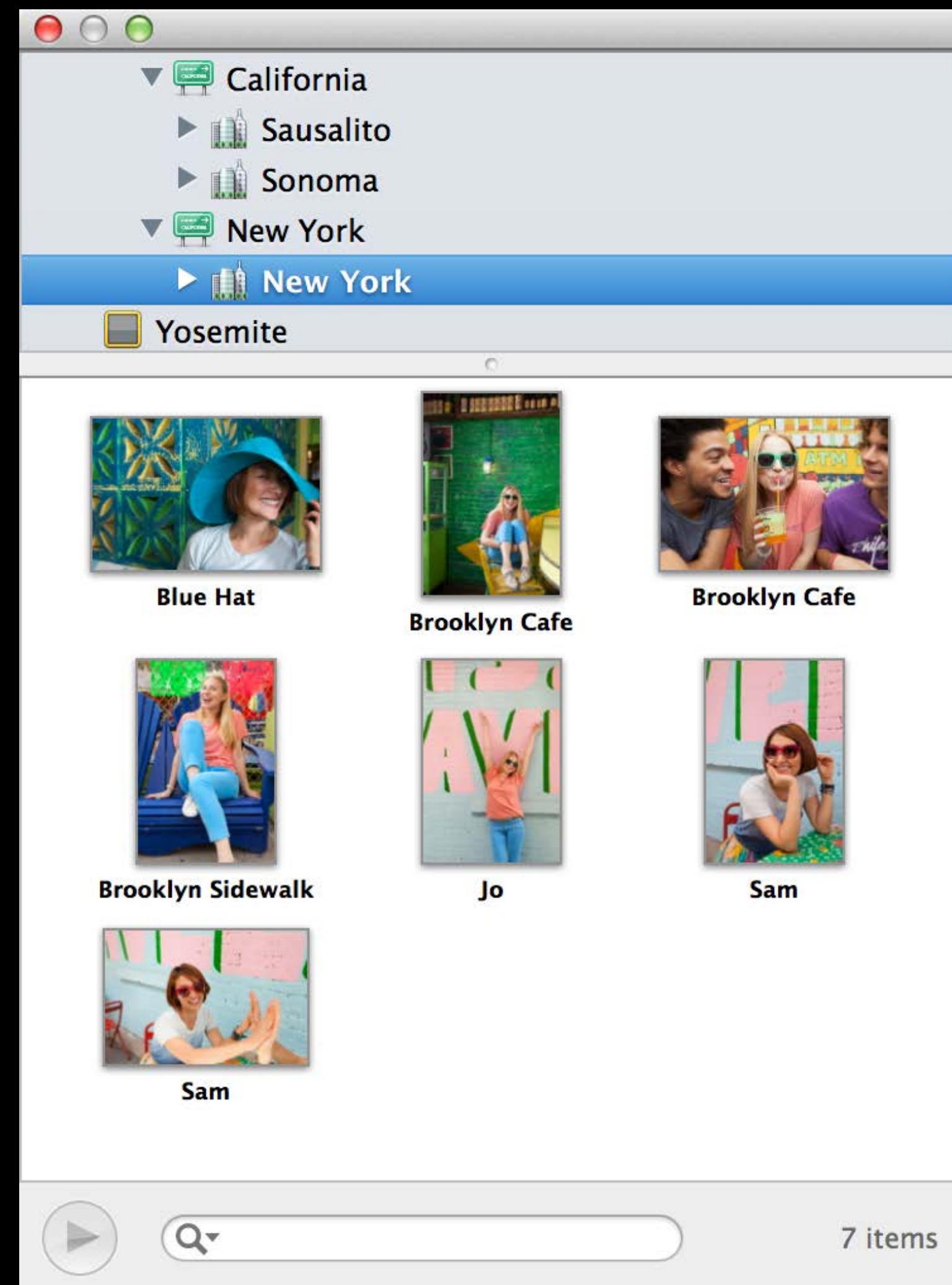
Media Library Access

NSMediaLibraryBrowserController



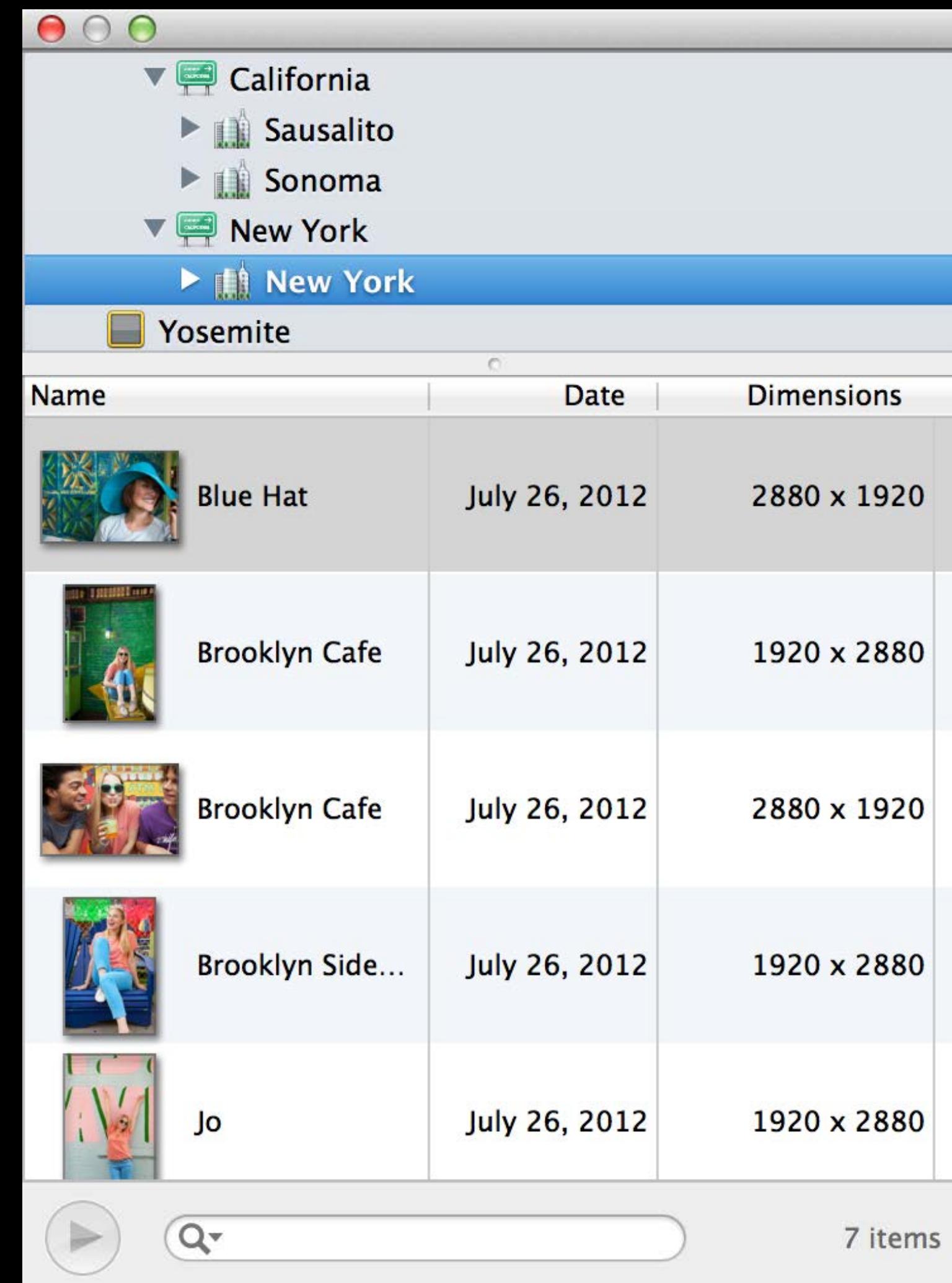
Media Library Access

NSMediaLibraryBrowserController



Media Library Access

NSMediaLibraryBrowserController



Media Library Access

Low-level access

Media Library Access

Low-level access

- New framework MediaLibrary
 - Read-only Objective-C data model representing the user's media
 - Asynchronous

Media Library Access

Low-level access

- New framework MediaLibrary
 - Read-only Objective-C data model representing the user's media
 - Asynchronous
- Primary classes
 - MLMediaLibrary
 - MLMediaSource
 - MLMediaGroup
 - MLMediaObject

Block-Based Sheet Presentation

In 10.8 and earlier

Block-Based Sheet Presentation

In 10.8 and earlier

```
{  
    ...  
    [NSApp beginSheet:sheet  
        modalForWindow:parentWindow  
        modalDelegate:self  
        didEndSelector:@selector(sheetDidEnd:returnCode:contextInfo:)  
        contextInfo:NULL];  
    ...  
}
```

Block-Based Sheet Presentation

In 10.8 and earlier

```
{  
    ...  
    [NSApp beginSheet:sheet  
        modalForWindow:parentWindow  
        modalDelegate:self  
        didEndSelector:@selector(sheetDidEnd:returnCode:contextInfo:)  
        contextInfo:NULL];  
    ...  
}
```

Block-Based Sheet Presentation

In 10.8 and earlier

```
{  
    ...  
    [NSApp beginSheet:sheet  
        modalForWindow:parentWindow  
        modalDelegate:self  
        didEndSelector:@selector(sheetDidEnd:returnCode:contextInfo:)  
        contextInfo:NULL];  
    ...  
}  
  
- (void)sheetDidEnd:(NSWindow *)sheet  
    returnCode:(NSInteger)code  
    contextInfo:(void *)context {  
    ... process user response ...  
}
```

Block-Based Sheet Presentation

In 10.8 and earlier

```
{  
    ...  
    [NSApp beginSheet:sheet  
        modalForWindow:parentWindow  
        modalDelegate:self  
        didEndSelector:@selector(sheetDidEnd:returnCode:contextInfo:)  
        contextInfo:NULL];  
    ...  
}  
  
- (void)sheetDidEnd:(NSWindow *)sheet  
    returnCode:(NSInteger)code  
    contextInfo:(void *)context {  
    ... process user response ...  
}
```

Block-Based Sheet Presentation

10.9

Block-Based Sheet Presentation

10.9

```
[parentWindow beginSheet:sheet  
completionHandler:^(NSModalResponse result)handler {  
    ... process user response ...  
}];
```

Block-Based Sheet Presentation

10.9

```
[parentWindow beginSheet:sheet  
completionHandler:^(NSModalResponse result)handler {  
    ... process user response ...  
}];
```

Block-Based Sheet Presentation

10.9

```
[parentWindow beginSheet:sheet  
completionHandler:^(NSModalResponse result)handler {  
    ... process user response ...  
}];
```

- This version will also

Block-Based Sheet Presentation

10.9

```
[parentWindow beginSheet:sheet  
completionHandler:^(NSModalResponse result)handler {  
    ... process user response ...  
}];
```

- This version will also
 - Automatically dismiss the sheet

Block-Based Sheet Presentation

10.9

```
[parentWindow beginSheet:sheet  
completionHandler:^(NSModalResponse result)handler {  
    ... process user response ...  
}];
```

- This version will also
 - Automatically dismiss the sheet
 - Queue up sheets for presentation

Block-Based Sheet Presentation

10.9

```
[parentWindow beginSheet:sheet  
completionHandler:^(NSModalResponse result)handler {  
    ... process user response ...  
}];
```

- This version will also
 - Automatically dismiss the sheet
 - Queue up sheets for presentation
 - Use beginCriticalSheet:completionHandler: to skip the queue

Full Screen for Multiple Monitors

Full Screen for Multiple Monitors



Full Screen for Multiple Monitors



Full Screen for Multiple Monitors



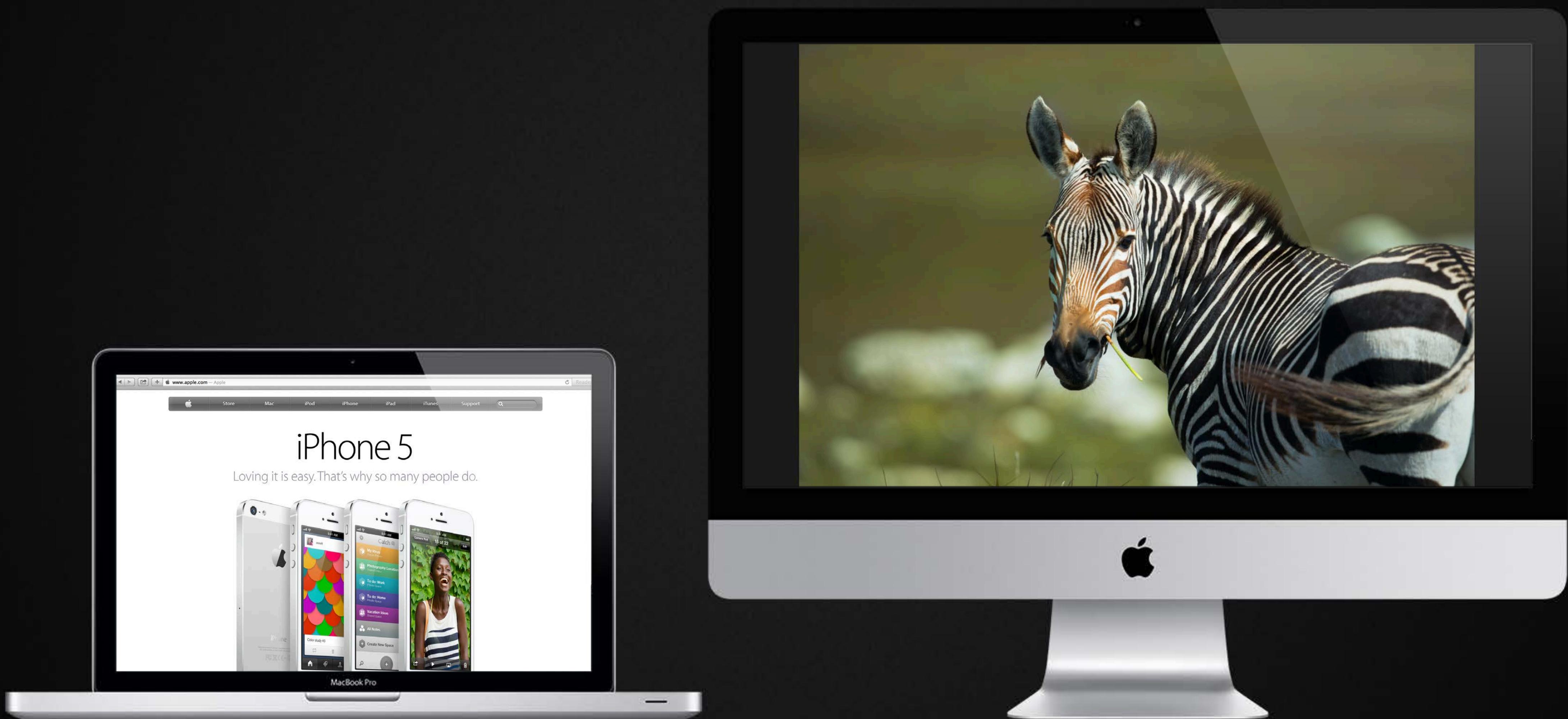
Full Screen for Multiple Monitors



Full Screen for Multiple Monitors



Full Screen for Multiple Monitors



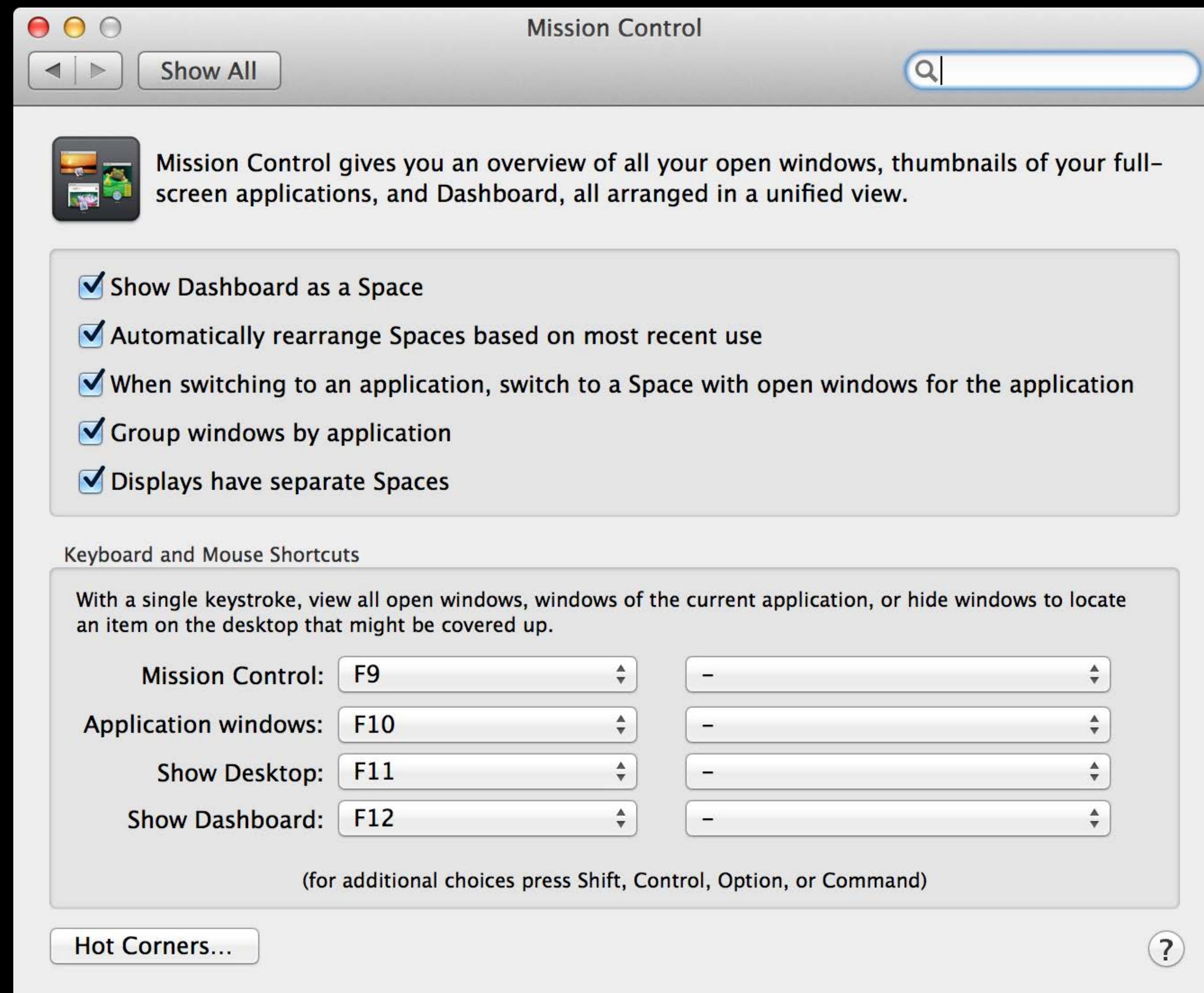
Full Screen for Multiple Monitors



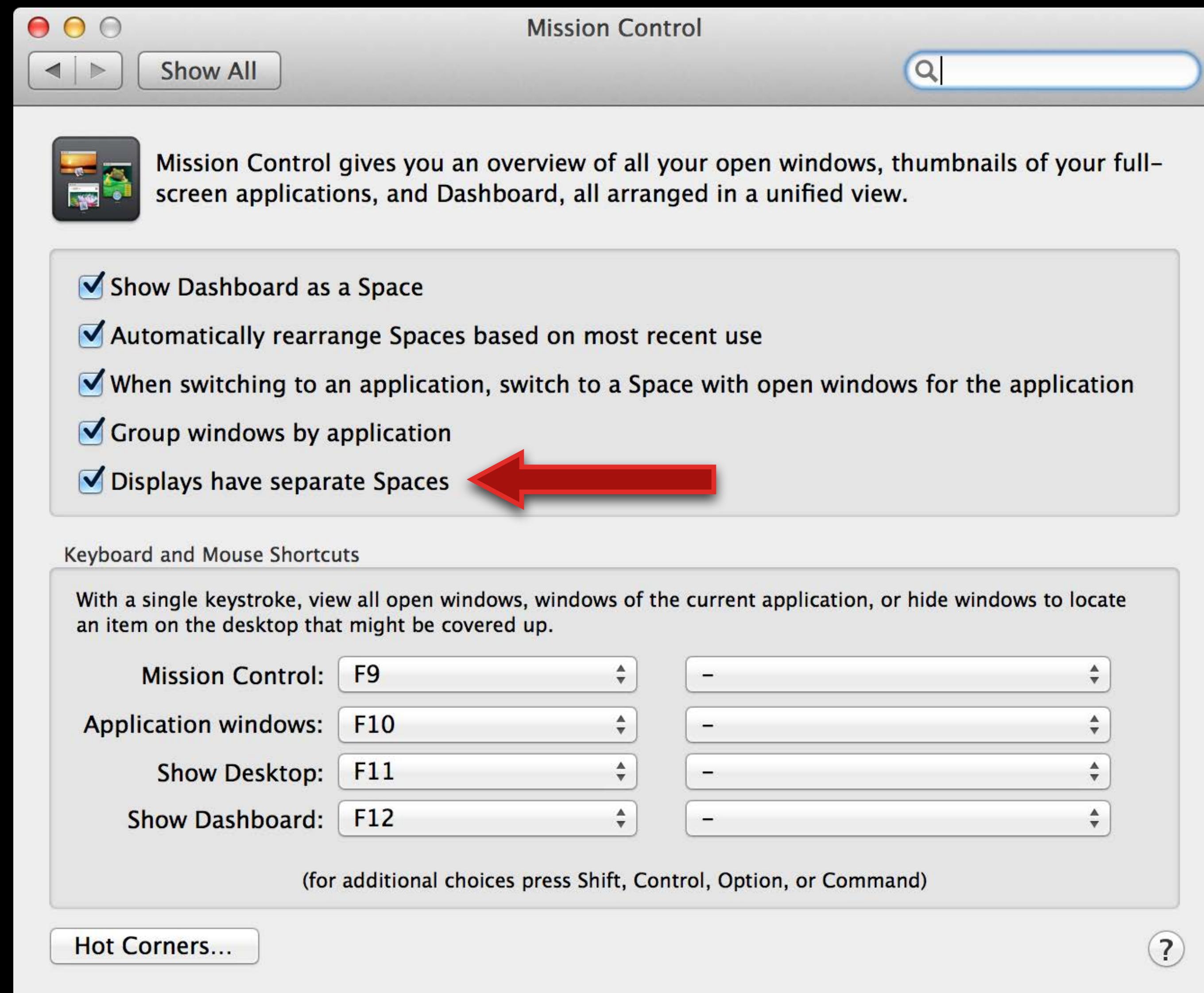
Full Screen for Multiple Monitors



Full Screen for Multiple Monitors



Full Screen for Multiple Monitors



Full Screen for Multiple Monitors

Full Screen for Multiple Monitors

- User choice
 - “Displays have separate Spaces”

Full Screen for Multiple Monitors

- User choice
 - “Displays have separate Spaces”
- In this mode
 - Each screen has its own set of spaces
 - Each screen also has its own menu bar
 - Windows do not span screens

Full Screen for Multiple Monitors

- No new APIs
- Change in behavior:
 - [NSScreen mainScreen] returns the active screen
 - State-restored (resumed) windows will return to their previous locations
 - Newly-opened and positioned windows will prefer active screen

Full Screen for Multiple Monitors

Full Screen and Cocoa Lab

Frameworks Lab A
Thursday 9:00-10:00AM

Updated System Font

Updated System Font

- New font for UI elements
 - Very close to Lucida Grande
 - Optimized for Retina displays
 - Not intended for general use

Updated System Font

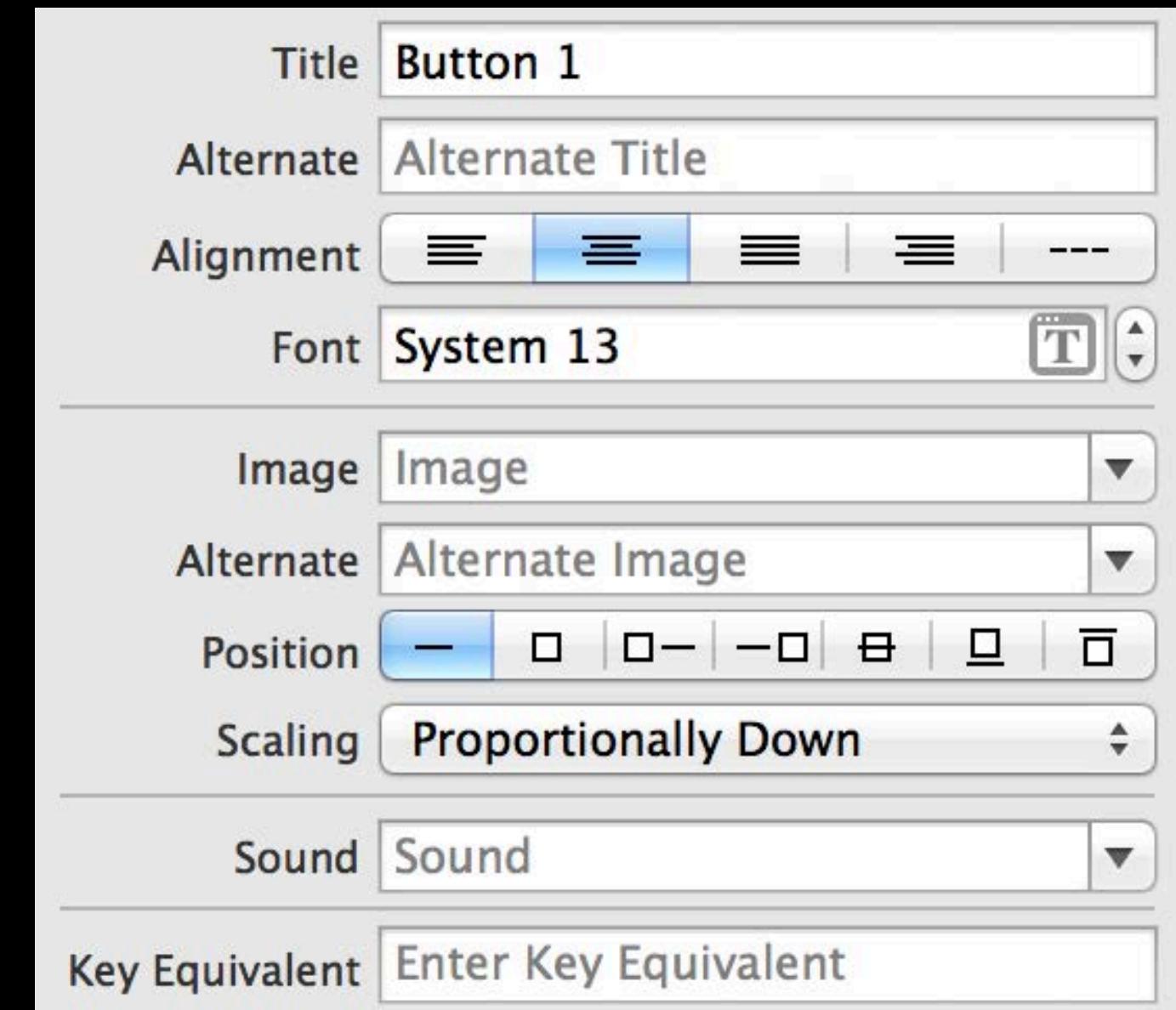
- New font for UI elements
 - Very close to Lucida Grande
 - Optimized for Retina displays
 - Not intended for general use
- Access with `+systemFontOfSize:`

Updated System Font

- New font for UI elements
 - Very close to Lucida Grande
 - Optimized for Retina displays
 - Not intended for general use
- Access with `+systemFontOfSize:`
 - Or in Xcode as “System” font

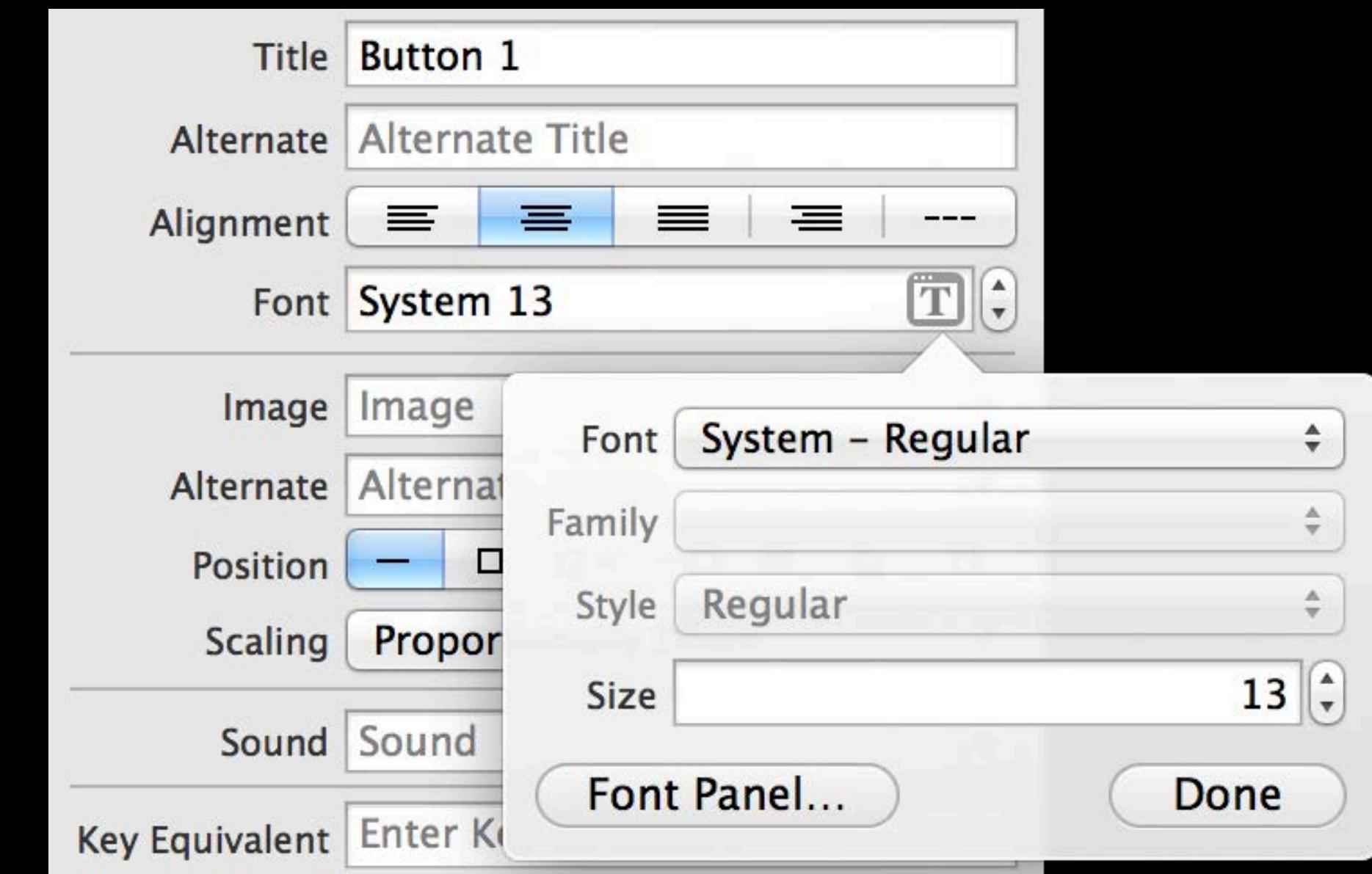
Updated System Font

- New font for UI elements
 - Very close to Lucida Grande
 - Optimized for Retina displays
 - Not intended for general use
- Access with `+systemFontOfSize:`
 - Or in Xcode as “System” font



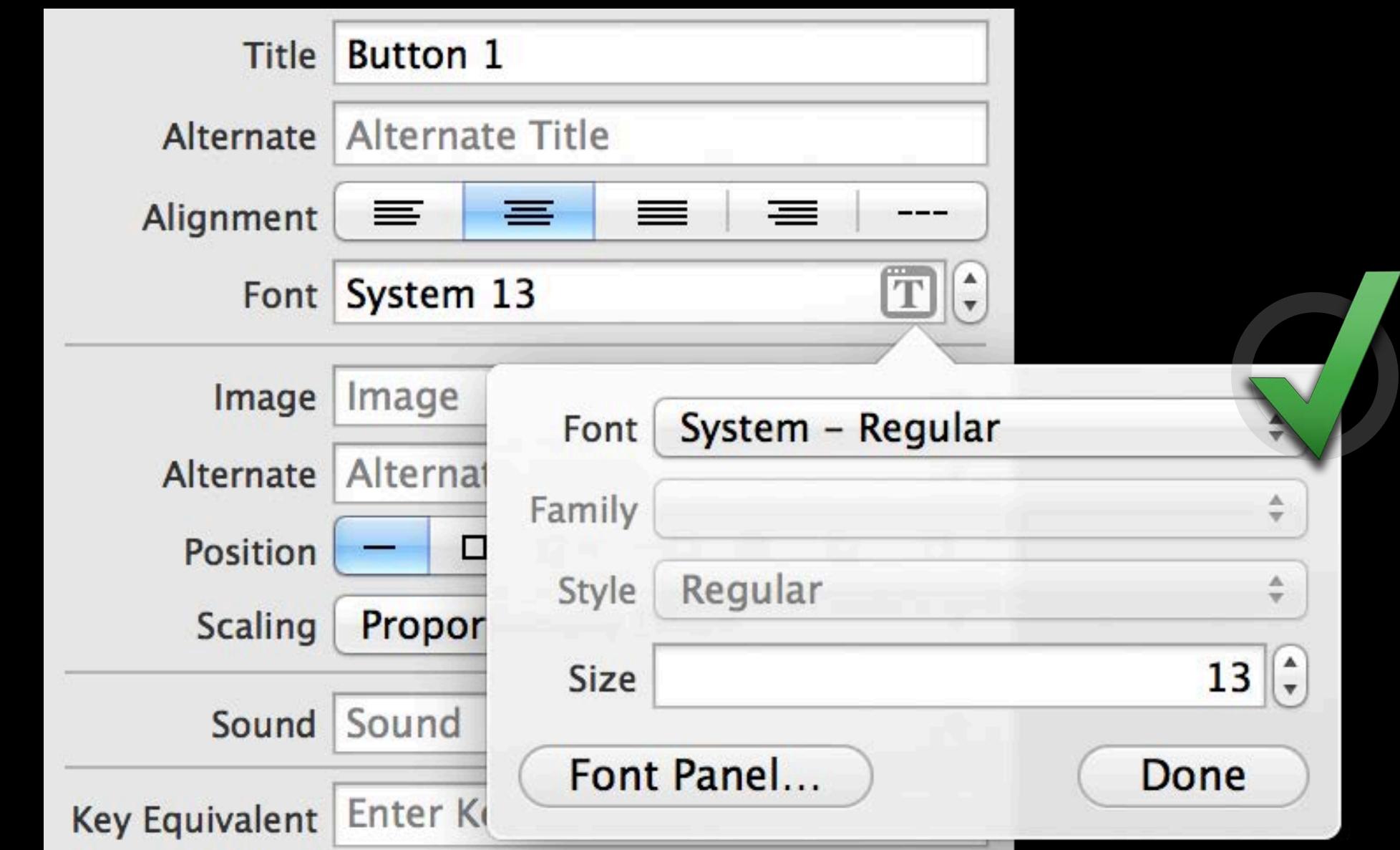
Updated System Font

- New font for UI elements
 - Very close to Lucida Grande
 - Optimized for Retina displays
 - Not intended for general use
- Access with `+systemFontOfSize:`
 - Or in Xcode as “System” font



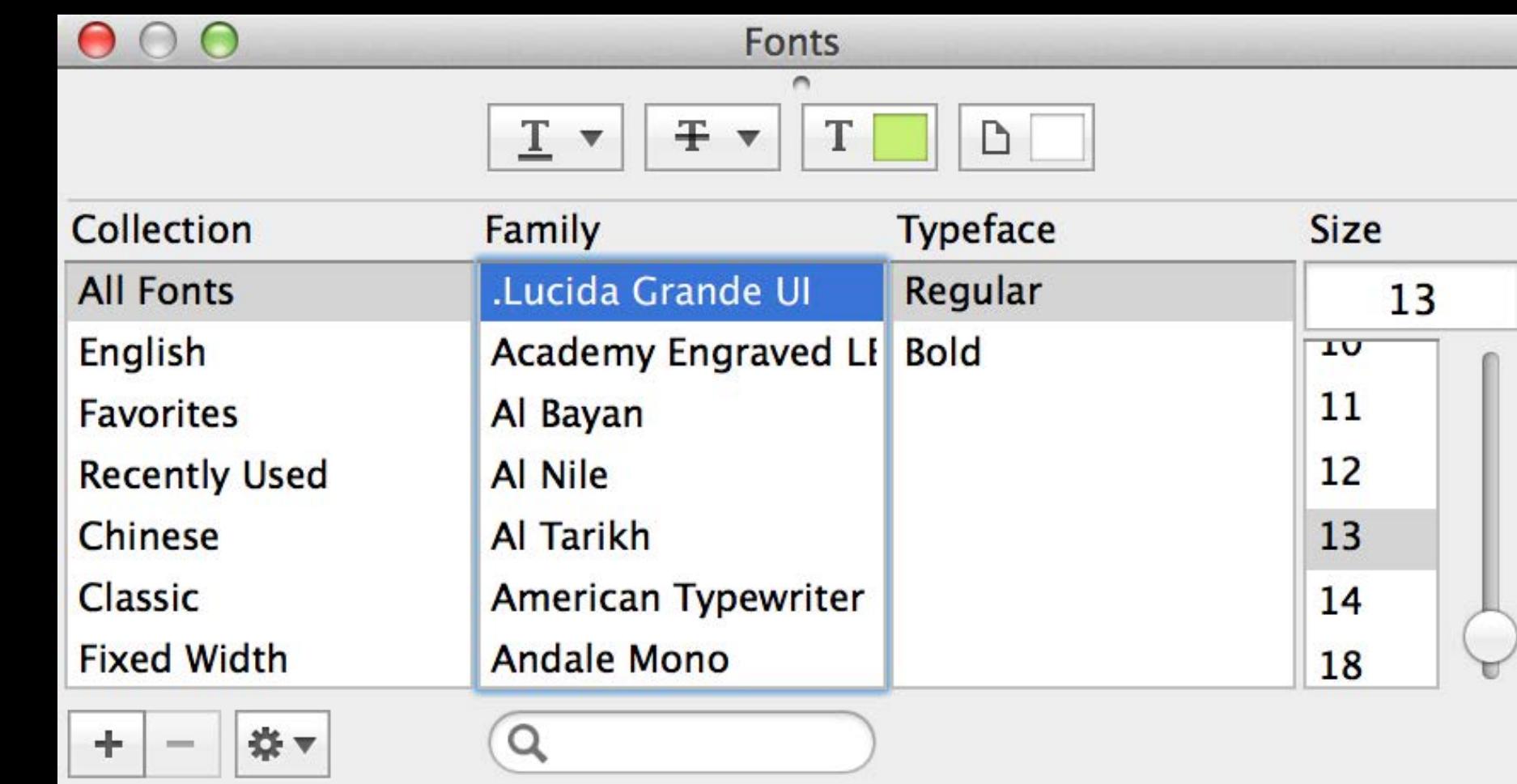
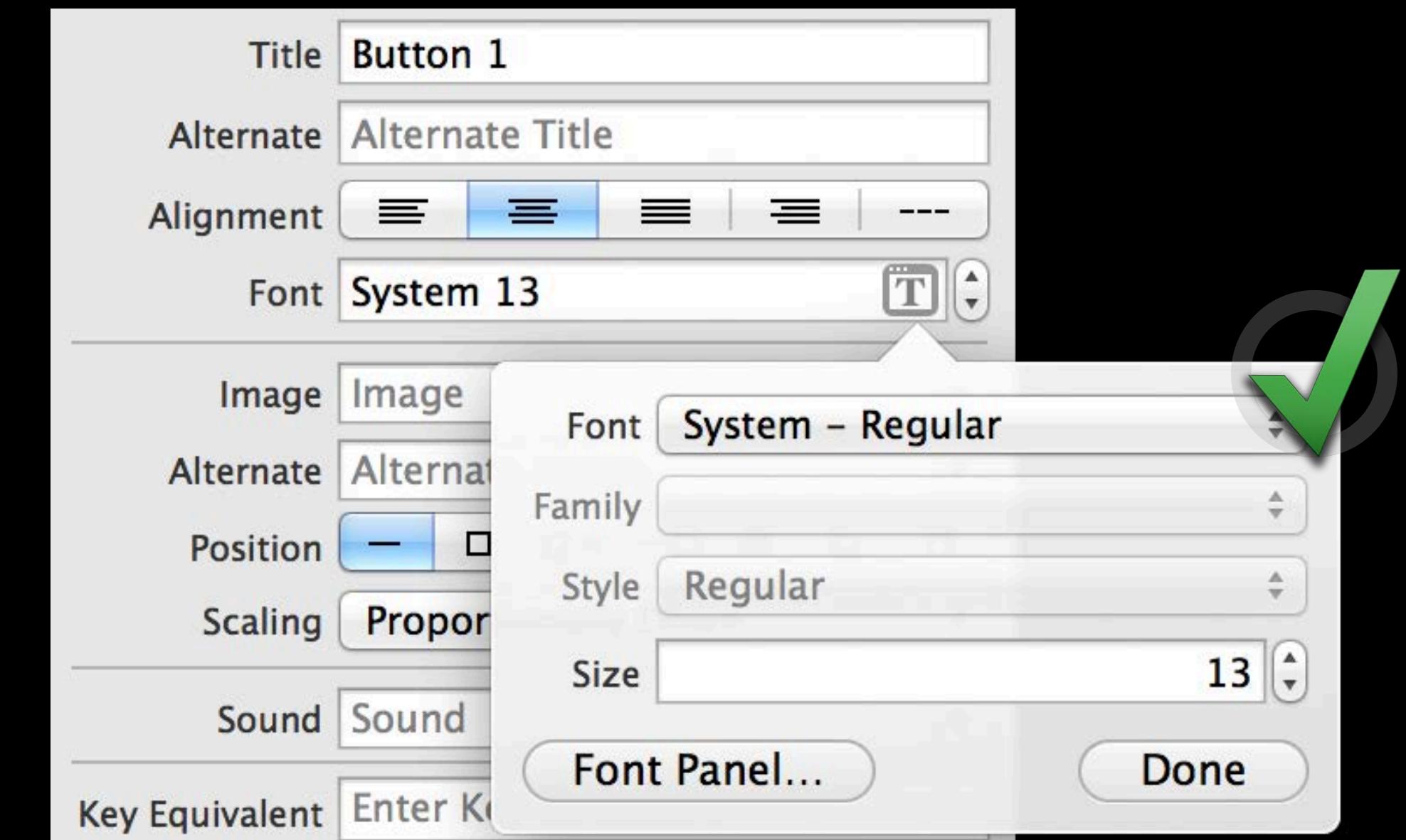
Updated System Font

- New font for UI elements
 - Very close to Lucida Grande
 - Optimized for Retina displays
 - Not intended for general use
- Access with `+systemFontOfSize:`
 - Or in Xcode as “System” font



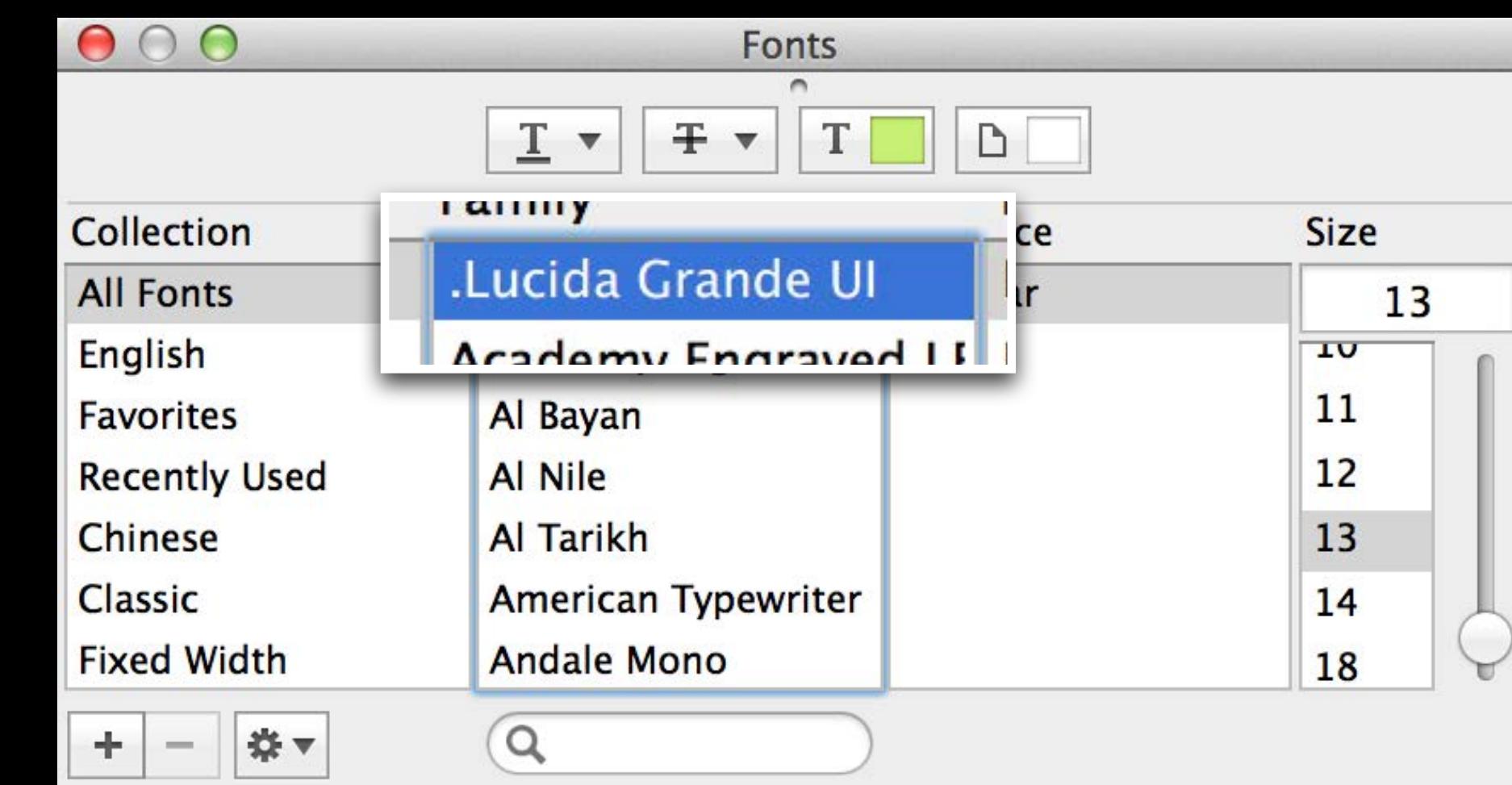
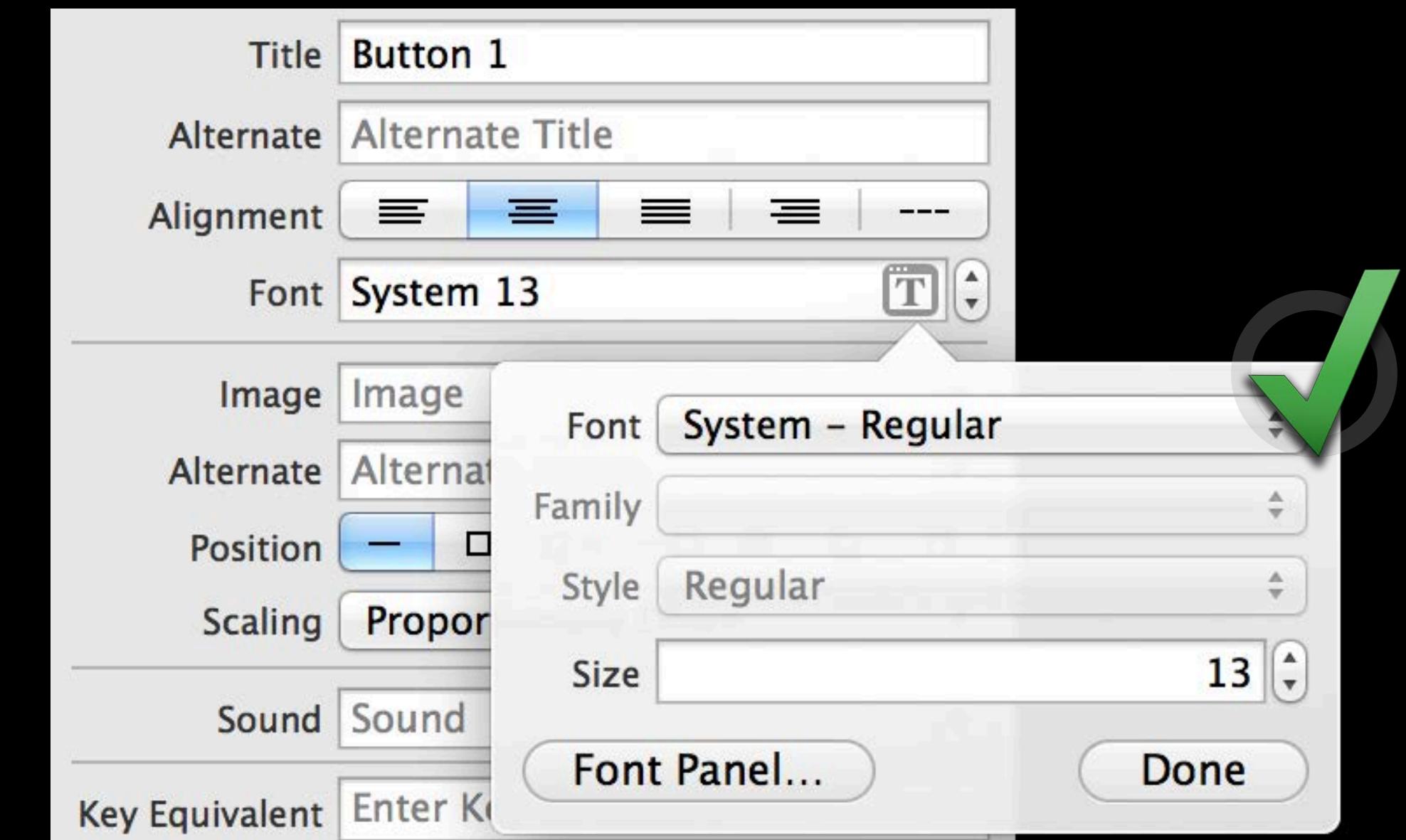
Updated System Font

- New font for UI elements
 - Very close to Lucida Grande
 - Optimized for Retina displays
 - Not intended for general use
- Access with `+systemFontOfSize:`
 - Or in Xcode as “System” font



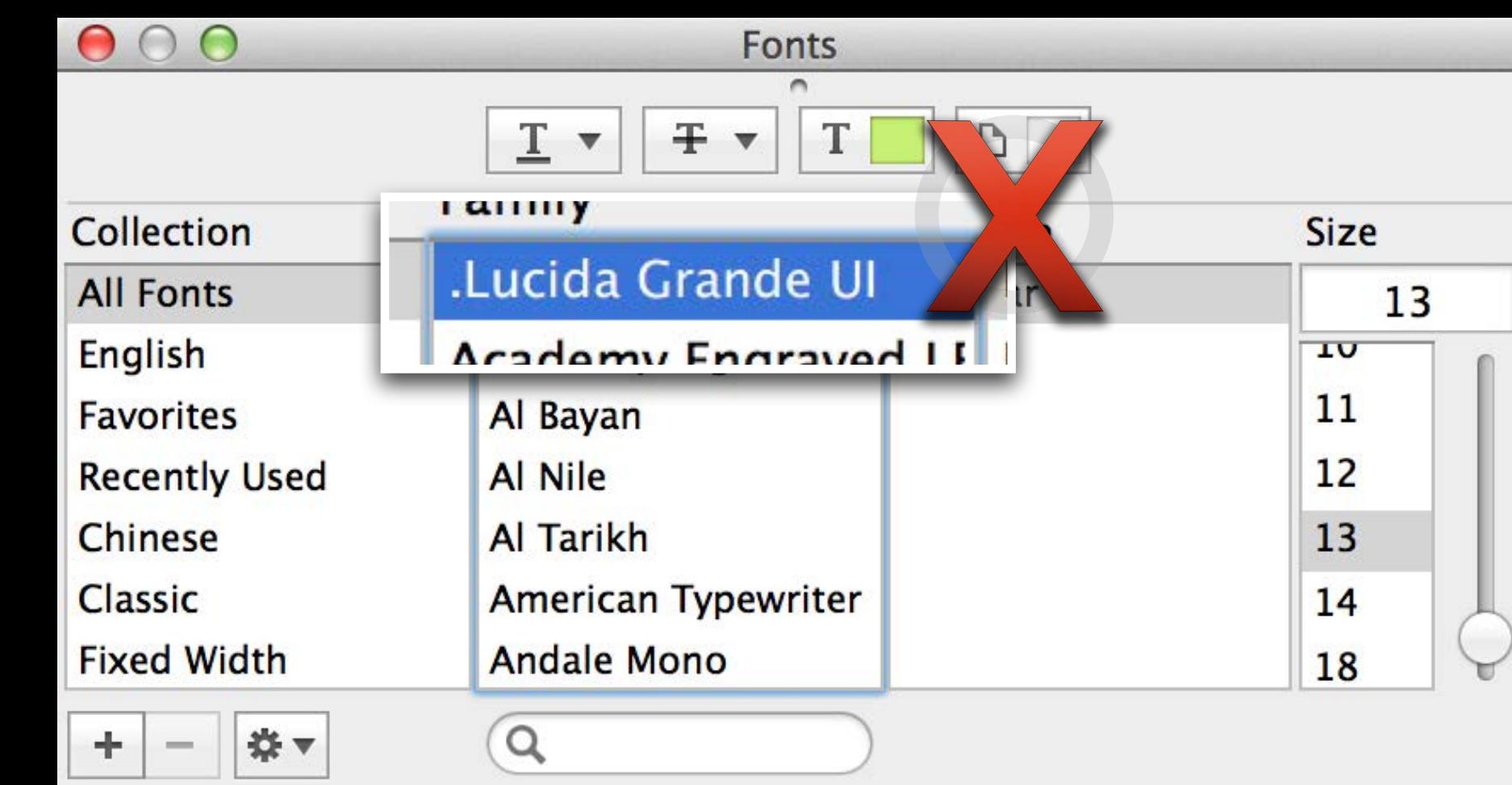
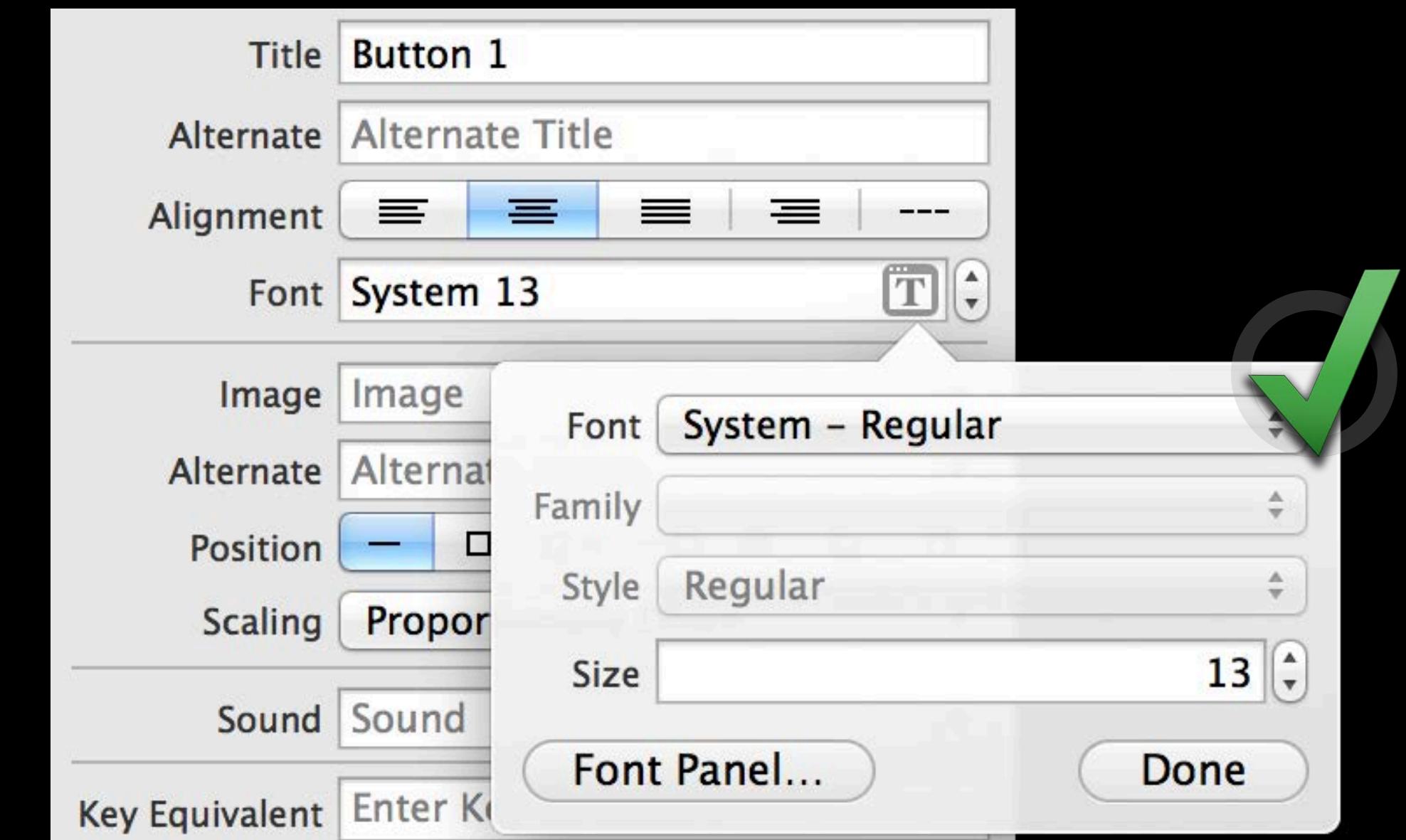
Updated System Font

- New font for UI elements
 - Very close to Lucida Grande
 - Optimized for Retina displays
 - Not intended for general use
- Access with `+systemFontOfSize:`
 - Or in Xcode as “System” font



Updated System Font

- New font for UI elements
 - Very close to Lucida Grande
 - Optimized for Retina displays
 - Not intended for general use
- Access with `+systemFontOfSize:`
 - Or in Xcode as “System” font



App Nap

App Nap

- System will “throttle” non-foreground apps that aren’t doing useful work

App Nap

- System will “throttle” non-foreground apps that aren’t doing useful work
 - Delay and rate limit timers

App Nap

- System will “throttle” non-foreground apps that aren’t doing useful work
 - Delay and rate limit timers
 - Lower I/O and CPU priorities

App Nap

- System will “throttle” non-foreground apps that aren’t doing useful work
 - Delay and rate limit timers
 - Lower I/O and CPU priorities
- Heuristics to determine “useful” work

App Nap

- System will “throttle” non-foreground apps that aren’t doing useful work
 - Delay and rate limit timers
 - Lower I/O and CPU priorities
- Heuristics to determine “useful” work
 - Handling user event/action method

App Nap

- System will “throttle” non-foreground apps that aren’t doing useful work
 - Delay and rate limit timers
 - Lower I/O and CPU priorities
- Heuristics to determine “useful” work
 - Handling user event/action method
 - Drawing visibly

App Nap

- System will “throttle” non-foreground apps that aren’t doing useful work
 - Delay and rate limit timers
 - Lower I/O and CPU priorities
- Heuristics to determine “useful” work
 - Handling user event/action method
 - Drawing visibly
 - Playing audio

App Nap

- System will “throttle” non-foreground apps that aren’t doing useful work
 - Delay and rate limit timers
 - Lower I/O and CPU priorities
- Heuristics to determine “useful” work
 - Handling user event/action method
 - Drawing visibly
 - Playing audio
 - Disabled system sleep

App Nap

- System will “throttle” non-foreground apps that aren’t doing useful work
 - Delay and rate limit timers
 - Lower I/O and CPU priorities
- Heuristics to determine “useful” work
 - Handling user event/action method
 - Drawing visibly
 - Playing audio
 - Disabled system sleep
 - Using new APIs to declare activity

App Nap

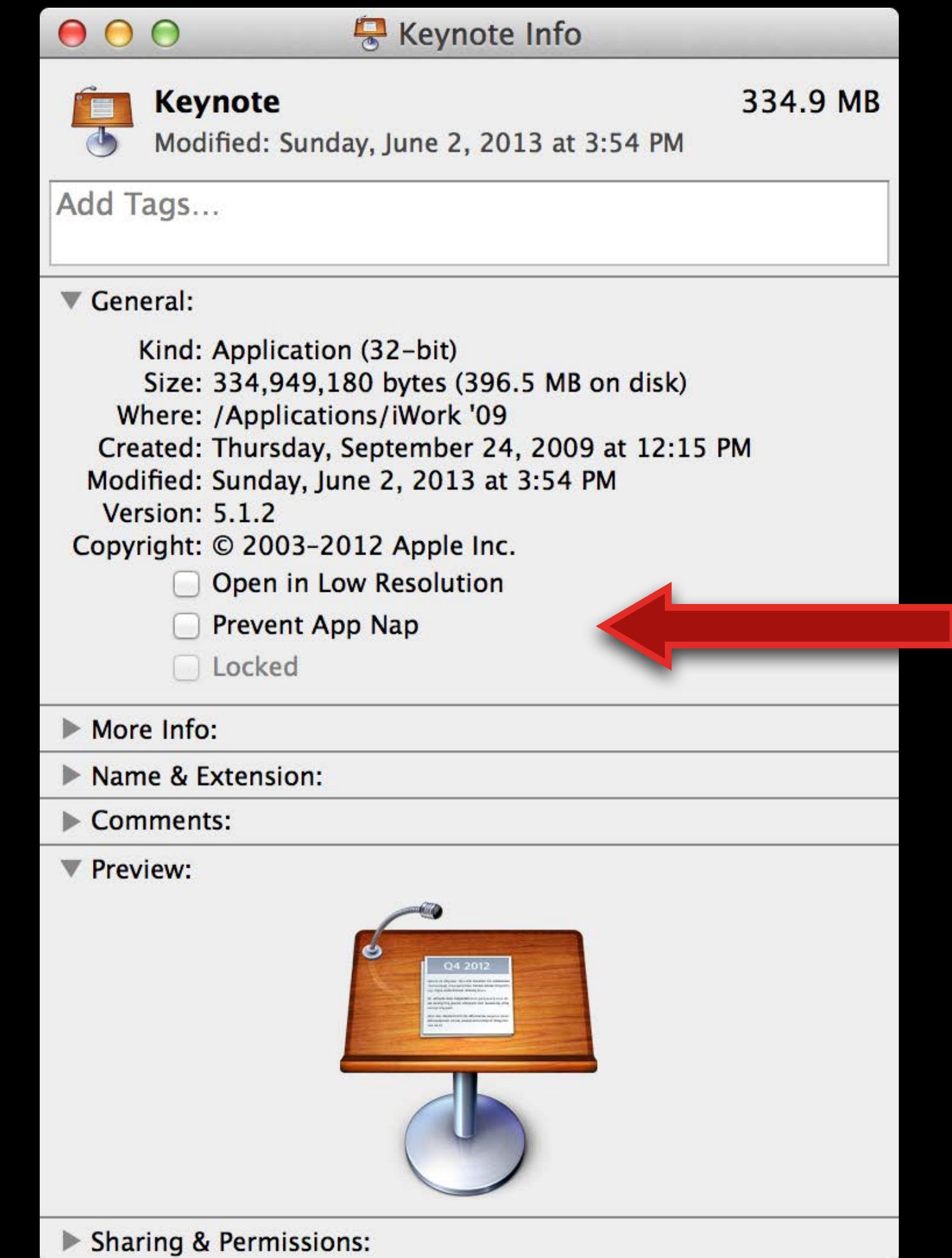
- Automatically applied to all apps

App Nap

- Automatically applied to all apps
 - Can be disabled by the user

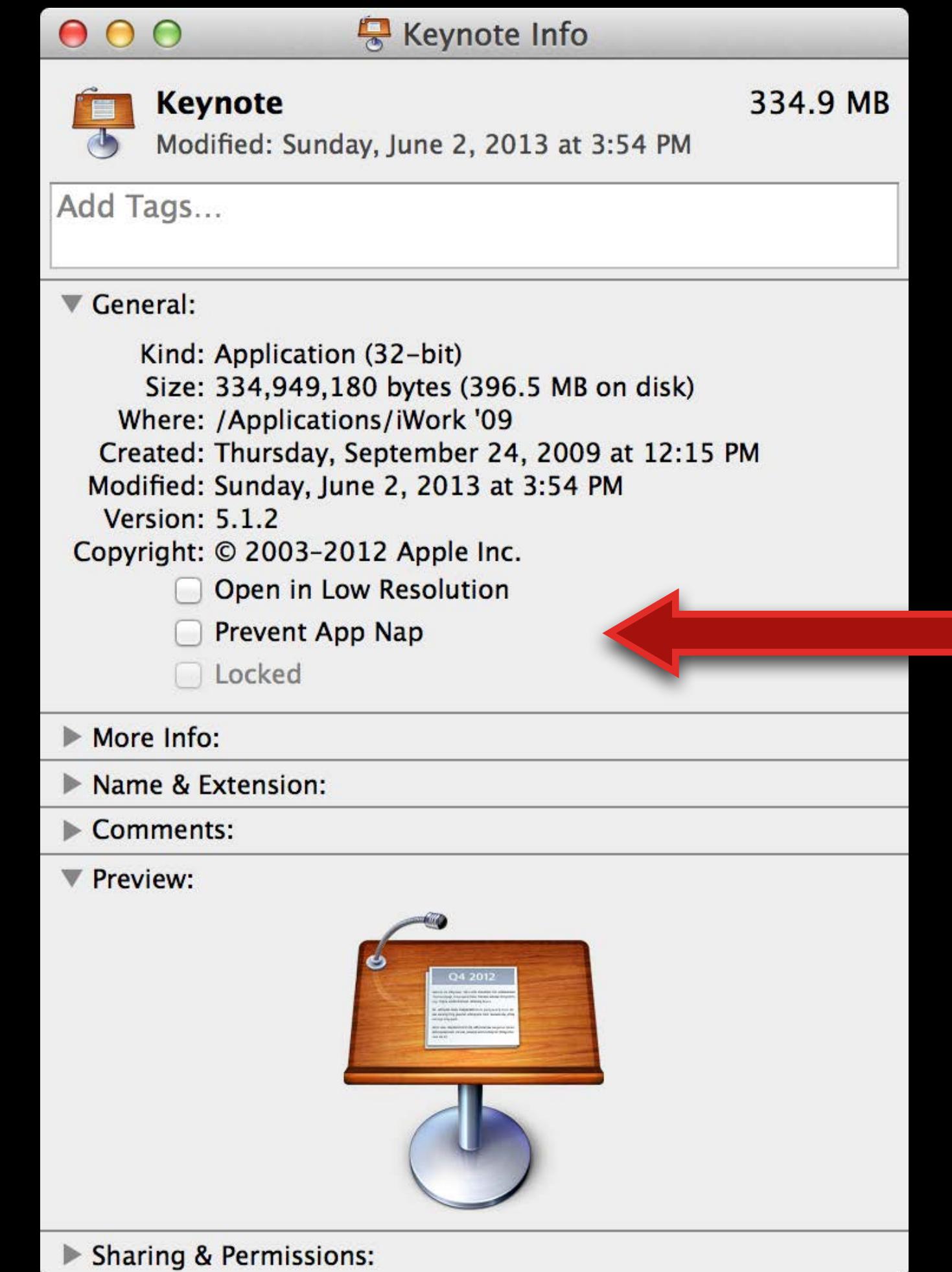
App Nap

- Automatically applied to all apps
 - Can be disabled by the user



App Nap

- Automatically applied to all apps
 - Can be disabled by the user
 - No checkbox for 10.9-linked apps



Activity APIs

Declare the activities your app is doing

Activity APIs

Declare the activities your app is doing

- NSProcessInfo

- `(void)performActivityWithOptions:(NSActivityOptions)options
reason:(NSString *)reason
block:(void (^)())activityBlock;`

Activity APIs

Declare the activities your app is doing

- NSProcessInfo
 - `(void)performActivityWithOptions:(NSActivityOptions)options
reason:(NSString *)reason
block:(void (^)(() activityBlock;`
 - `(id <NSObject>)beginActivityWithOptions:(NSActivityOptions)options
reason:(NSString *)reason;`
 - `(void)endActivity:(id <NSObject>)activity;`

Activity APIs

Declare the activities your app is doing

- NSProcessInfo
 - `(void)performActivityWithOptions:(NSActivityOptions)options
reason:(NSString *)reason
block:(void (^)())activityBlock;`

Activity APIs

Declare the activities your app is doing

- NSProcessInfo

- `(void)performActivityWithOptions:(NSActivityOptions)options
reason:(NSString *)reason
block:(void (^)())activityBlock;`

- NSActivityOptions

Activity APIs

Declare the activities your app is doing

- NSProcessInfo

- `(void)performActivityWithOptions:(NSActivityOptions)options
reason:(NSString *)reason
block:(void (^)())activityBlock;`

- NSActivityOptions

- `NSActivityUserInitiated`

Activity APIs

Declare the activities your app is doing

- NSProcessInfo

- `(void)performActivityWithOptions:(NSActivityOptions)options
reason:(NSString *)reason
block:(void (^)())activityBlock;`

- NSActivityOptions

- `NSActivityUserInitiated`

- `NSActivityUserInitiatedAllowingIdleSystemSleep`

Activity APIs

Declare the activities your app is doing

- NSProcessInfo

- `(void)performActivityWithOptions:(NSActivityOptions)options
reason:(NSString *)reason
block:(void (^)())activityBlock;`

- NSActivityOptions

- `NSActivityUserInitiated`

- `NSActivityUserInitiatedAllowingIdleSystemSleep`

- `NSActivityBackground`

Activity APIs

Declare the activities your app is doing

- NSProcessInfo

- `(void)performActivityWithOptions:(NSActivityOptions)options
reason:(NSString *)reason
block:(void (^)())activityBlock;`

- NSActivityOptions

- `NSActivityUserInitiated`

- `NSActivityUserInitiatedAllowingIdleSystemSleep`

- `NSActivityBackground`

- `...`

NSTimer Tolerance

Explicit control over timer delay

NSTimer Tolerance

Explicit control over timer delay

- NSTimer

```
@property (NSTimeInterval) tolerance;
```

NSTimer Tolerance

Explicit control over timer delay

- NSTimer
 - @property (NSTimeInterval) tolerance;
- Lets timer fire later than scheduled time

NSTimer Tolerance

Explicit control over timer delay

- NSTimer
 - @property (NSTimeInterval) tolerance;
- Lets timer fire later than scheduled time
- Enables system to synchronize timers for power savings

NSTimer Tolerance

Explicit control over timer delay

```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:NO];
```

```
timer.tolerance = 1.0;
```

```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:YES];  
timer.tolerance = 1.0;
```

NSTimer Tolerance

Explicit control over timer delay

```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:NO];
```



```
timer.tolerance = 1.0;
```

```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:YES];
timer.tolerance = 1.0;
```

NSTimer Tolerance

Explicit control over timer delay

```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:NO];
```



```
timer.tolerance = 1.0;
```

```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:YES];
timer.tolerance = 1.0;
```

NSTimer Tolerance

Explicit control over timer delay

```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:NO];
```

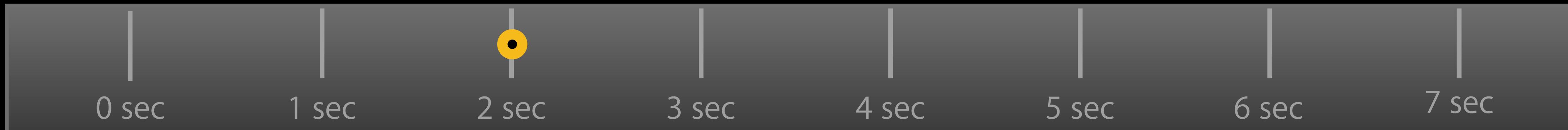


```
timer.tolerance = 1.0;
```

NSTimer Tolerance

Explicit control over timer delay

```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:NO];
```



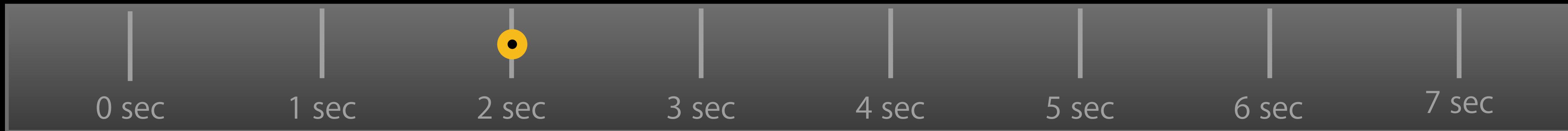
```
timer.tolerance = 1.0;
```



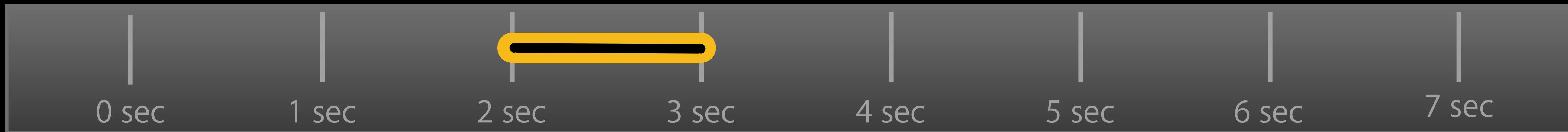
NSTimer Tolerance

Explicit control over timer delay

```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:NO];
```



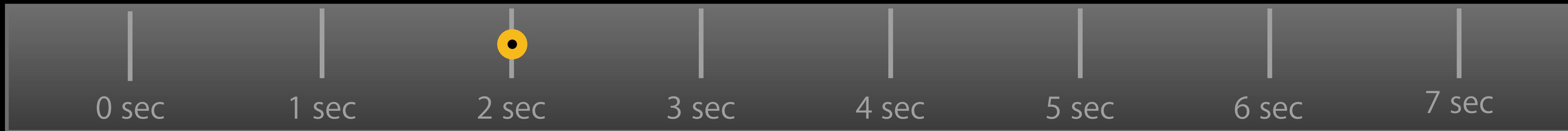
```
timer.tolerance = 1.0;
```



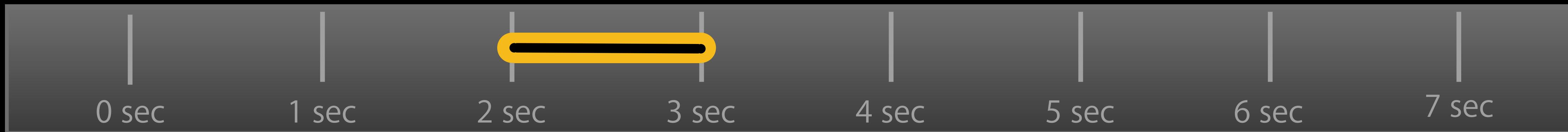
NSTimer Tolerance

Explicit control over timer delay

```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:NO];
```



```
timer.tolerance = 1.0;
```

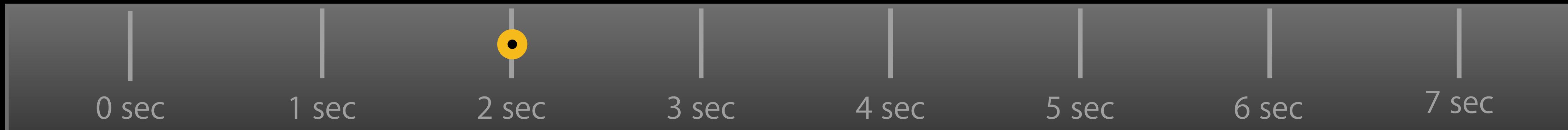


```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:YES];
timer.tolerance = 1.0;
```

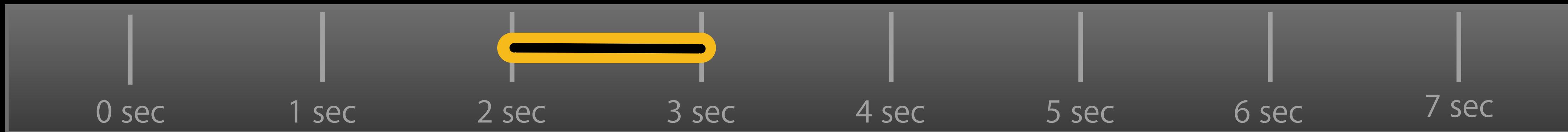
NSTimer Tolerance

Explicit control over timer delay

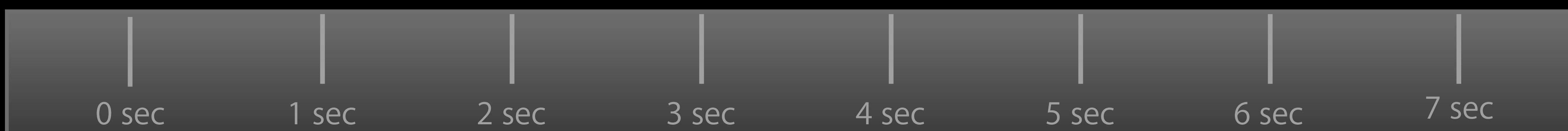
```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:NO];
```



```
timer.tolerance = 1.0;
```



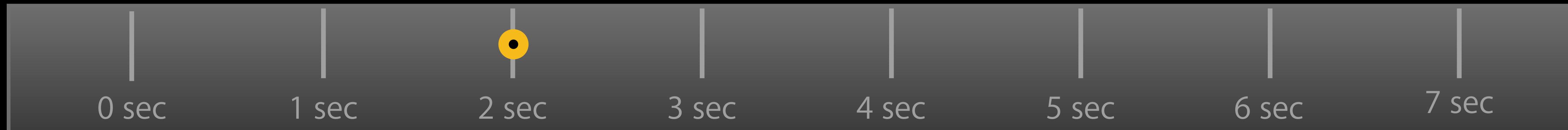
```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:YES];  
timer.tolerance = 1.0;
```



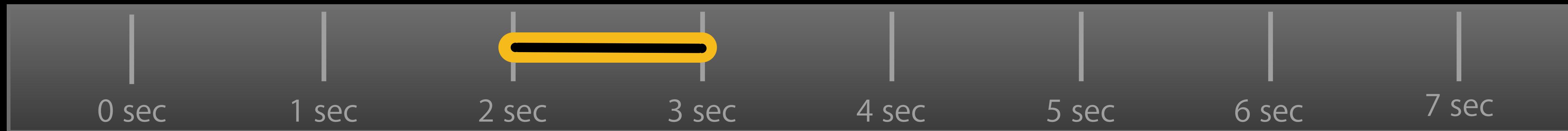
NSTimer Tolerance

Explicit control over timer delay

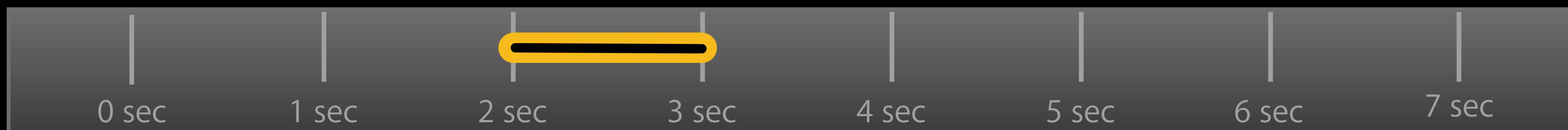
```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:NO];
```



```
timer.tolerance = 1.0;
```



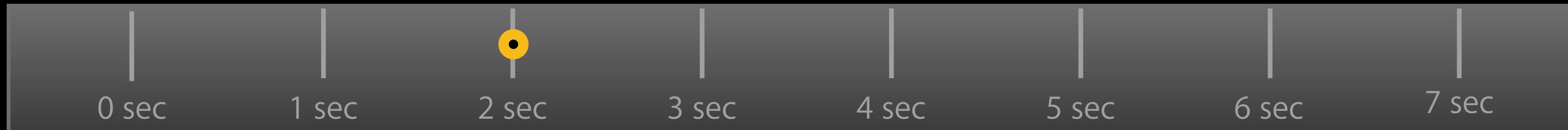
```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:YES];  
timer.tolerance = 1.0;
```



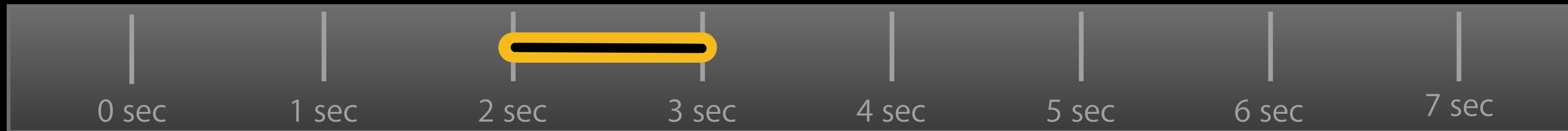
NSTimer Tolerance

Explicit control over timer delay

```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:NO];
```



```
timer.tolerance = 1.0;
```



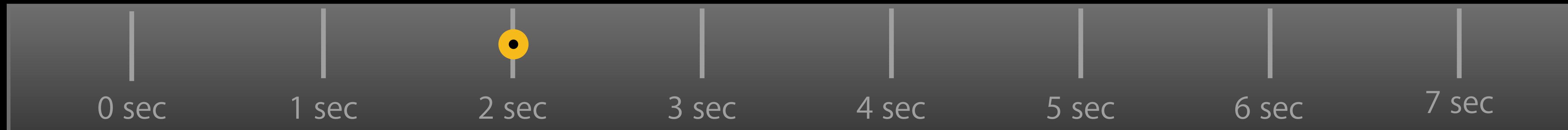
```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:YES];  
timer.tolerance = 1.0;
```



NSTimer Tolerance

Explicit control over timer delay

```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:NO];
```



```
timer.tolerance = 1.0;
```



```
NSTimer *timer = [NSTimer timerWithTimeInterval:2.0 ... repeats:YES];  
timer.tolerance = 1.0;
```



Occlusion

Determine whether windows are visible or not

Occlusion

Determine whether windows are visible or not

- NSWindow

- `(NSWindowOcclusionState)occlusionState;`

```
NSString *NSWindowDidChangeOcclusionStateNotification;
```

Occlusion

Determine whether windows are visible or not

- NSWindow

- `(NSWindowOcclusionState)occlusionState;`

```
NSString *NSWindowDidChangeOcclusionStateNotification;
```

- NSApplication

- `(NSApplicationOcclusionState)occlusionState;`

```
NSString *NSApplicationDidChangeOcclusionStateNotification;
```

App Nap

Improving Power Efficiency with App Nap

Pacific Heights
Wednesday 10:15AM

App Nap

Improving Power Efficiency with App Nap

Pacific Heights
Wednesday 10:15AM

Cocoa and Foundation Lab

Frameworks Lab A
Wednesday 11:30-1:45PM

App Nap

Improving Power Efficiency with App Nap	Pacific Heights Wednesday 10:15AM	
Cocoa and Foundation Lab	Frameworks Lab A Wednesday 11:30-1:45PM	
Maximizing Battery Life on OS X	Mission Tuesday 11:30AM	

Progress Reporting and Cancellation

Progress Reporting and Cancellation

- New mechanism for
 - Reporting progress of work

Progress Reporting and Cancellation

- New mechanism for
 - Reporting progress of work
 - Presenting and updating progress

Progress Reporting and Cancellation

- New mechanism for
 - Reporting progress of work
 - Presenting and updating progress
 - User-cancelling the work

Progress Reporting and Cancellation

- New mechanism for
 - Reporting progress of work
 - Presenting and updating progress
 - User-cancelling the work
- Loosely-coupled

Progress Reporting and Cancellation

- New mechanism for
 - Reporting progress of work
 - Presenting and updating progress
 - User-cancelling the work
- Loosely-coupled
- Composable across layers

Progress Reporting and Cancellation

- New mechanism for
 - Reporting progress of work
 - Presenting and updating progress
 - User-cancelling the work
- Loosely-coupled
- Composable across layers
- Cross-process

Progress Reporting and Cancellation

Reporting progress

```
- (BOOL)processStuff:(NSArray *)stuff error:(NSError **)outError {
    NSError *error = nil;
    NSUInteger count = [stuff count];

    for (NSUInteger index = 0; index < count; index++) {
        ... process stuff[index] ...
    }

    if (error && outError) *outError = error;
    return error ? NO : YES;
}
```

Progress Reporting and Cancellation

Reporting progress

```
- (BOOL)processStuff:(NSArray *)stuff error:(NSError **)outError {
    NSError *error = nil;
    NSUInteger count = [stuff count];
    NSProgress *progress = [NSProgress progressWithTotalUnitCount:count];
    for (NSUInteger index = 0; index < count; index++) {
        ... process stuff[index] ...
        [progress setCompletedUnitCount:index + 1];
    }
    if (error && outError) *outError = error;
    return error ? NO : YES;
}
```

Progress Reporting and Cancellation

Reporting progress

```
- (BOOL)processStuff:(NSArray *)stuff error:(NSError **)outError {
    NSError *error = nil;
    NSUInteger count = [stuff count];
    NSProgress *progress = [NSProgress progressWithTotalUnitCount:count];
    for (NSUInteger index = 0; index < count; index++) {

        ... process stuff[index] ...
        [progress setCompletedUnitCount:index + 1];
    }
    if (error && outError) *outError = error;
    return error ? NO : YES;
}
```

Progress Reporting and Cancellation

Reporting progress

```
- (BOOL)processStuff:(NSArray *)stuff error:(NSError **)outError {
    NSError *error = nil;
    NSUInteger count = [stuff count];
    NSProgress *progress = [NSProgress progressWithTotalUnitCount:count];
    for (NSUInteger index = 0; index < count; index++) {
        if ([progress isCancelled]) {
            error = [NSError errorWithDomain:NSCocoaErrorDomain
                code:NSUserCancelledError userInfo:nil];
            break;
        }
        ... process stuff[index] ...
        [progress setCompletedUnitCount:index + 1];
    }
    if (error && outError) *outError = error;
    return error ? NO : YES;
}
```

Progress Reporting and Cancellation

Observing progress

Progress Reporting and Cancellation

Observing progress

- Bind to or otherwise display properties such as

Progress Reporting and Cancellation

Observing progress

- Bind to or otherwise display properties such as

```
@property int64_t totalUnitCount;  
@property int64_t completedUnitCount;  
@property (copy) NSString *localizedDescription;  
@property (readonly) double fractionCompleted;
```

Progress Reporting and Cancellation

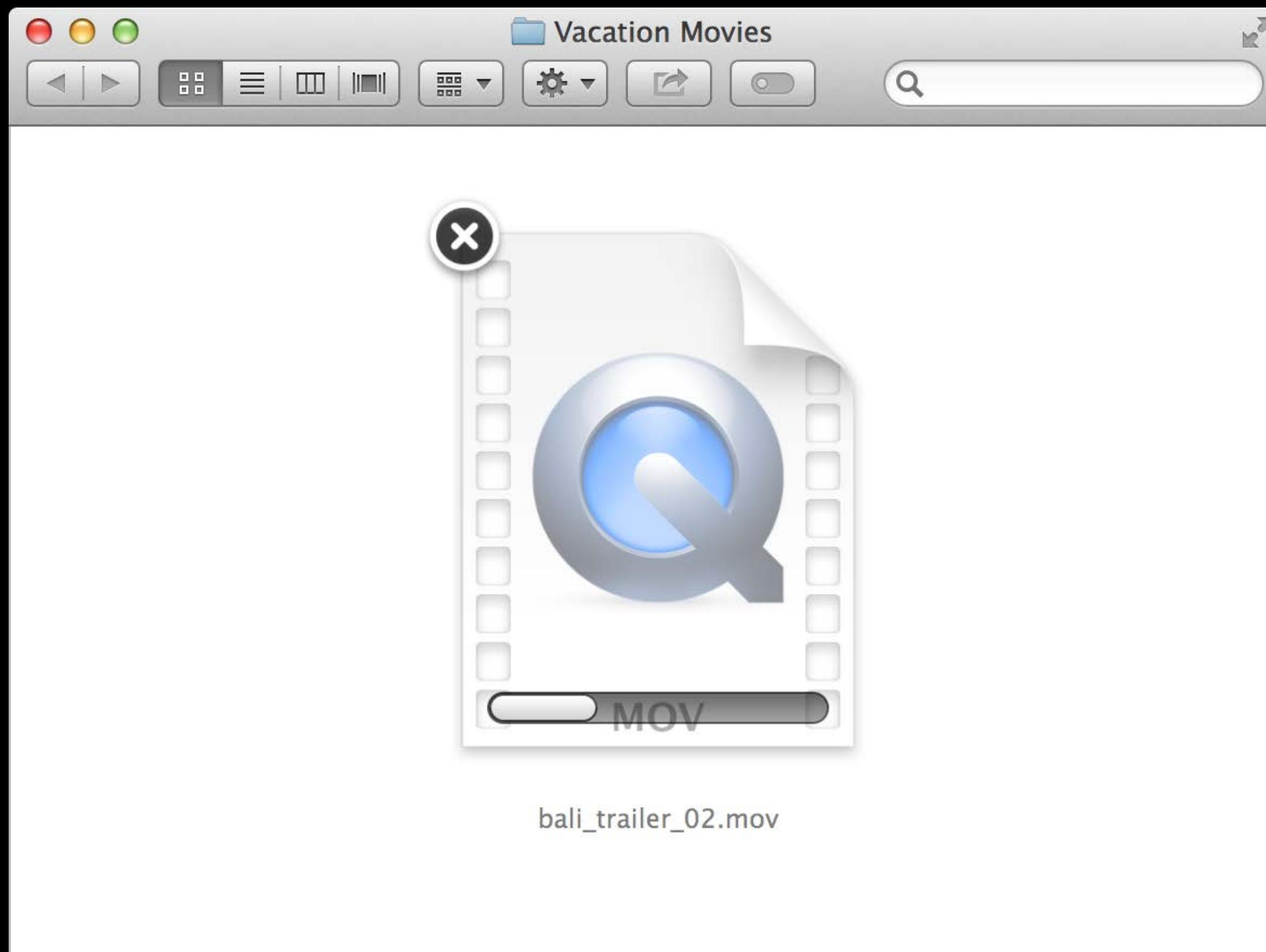
Observing progress

- Bind to or otherwise display properties such as

```
@property int64_t totalUnitCount;  
@property int64_t completedUnitCount;  
@property (copy) NSString *localizedDescription;  
@property (readonly) double fractionCompleted;  
  
@property (readonly, getter=isIndeterminate) BOOL indeterminate;  
@property (getter=isCancellable) BOOL cancellable;
```

Progress Reporting and Cancellation

Observing progress



Date and Time Handling

Calendrical computations are hard!

Date and Time Handling

Calendrical computations are hard!

- Different calendars
- Daylight savings
- Time zones

Date and Time Handling

New APIs

Date and Time Handling

New APIs

```
BOOL isToday = [calendar isDateInToday:date];
```

Date and Time Handling

New APIs

```
BOOL isToday = [calendar isDateInToday:date];  
BOOL isWeekend = [calendar isDateInWeekend:date];
```

Date and Time Handling

New APIs

```
BOOL isToday = [calendar isDateInToday:date];  
BOOL isWeekend = [calendar isDateInWeekend:date];  
BOOL sameMonth = [calendar isDate:date1  
                  equalToDate:date2  
                  toUnitGranularity:NSCalendarUnitMonth];
```

Date and Time Handling

New APIs

```
BOOL isToday = [calendar isDateInToday:date];  
BOOL isWeekend = [calendar isDateInWeekend:date];  
BOOL sameMonth = [calendar isDate:date1  
                  equalToDate:date2  
                  toUnitGranularity:NSCalendarUnitMonth];
```

etc...

Date and Time Handling

New enumeration APIs

Date and Time Handling

New enumeration APIs

```
- (void)enumerateDatesStartingAfterDate:(NSDate *)start  
    matchingComponents:(NSDateComponents *)comps  
    options:(NSCalendarOptions)opts  
    usingBlock:(void (^)(NSDate *date, BOOL exact, BOOL *stop))block;
```

Date and Time Handling

New enumeration APIs

```
- (void)enumerateDatesStartingAfterDate:(NSDate *)start  
    matchingComponents:(NSDateComponents *)comps  
    options:(NSCalendarOptions)opts  
    usingBlock:(void (^)(NSDate *date, BOOL exact, BOOL *stop))block;
```

Date and Time Handling

New enumeration APIs

```
- (void)enumerateDatesStartingAfterDate:(NSDate *)start  
    matchingComponents:(NSDateComponents *)comps  
    options:(NSCalendarOptions)opts  
    usingBlock:(void (^)(NSDate *date, BOOL exact, BOOL *stop))block;
```

Enumerating Dates

Create NSDates representing 2:30AM every day

Enumerating Dates

Create NSDates representing 2:30AM every day

```
NSDateComponents *twoThirty = [NSDateComponents new];  
twoThirty.hour = 2;  
twoThirty.minute = 30;
```

Enumerating Dates

Create NSDates representing 2:30AM every day

```
NSDateComponents *twoThirty = [NSDateComponents new];  
twoThirty.hour = 2;  
twoThirty.minute = 30;
```

```
[ [NSCalendar currentCalendar]
```

Enumerating Dates

Create NSDates representing 2:30AM every day

```
NSDateComponents *twoThirty = [NSDateComponents new];
twoThirty.hour = 2;
twoThirty.minute = 30;
```

```
[ [NSCalendar currentCalendar]
    enumerateDatesStartingAfterDate: [NSDate date]
        matchingComponents: twoThirty
        options: NSCalendarMatchNextTime
        usingBlock: ^(NSDate *date, BOOL exact, BOOL *stop) { ... }];
```

Enumerating Dates

Create NSDates representing 2:30AM every day

```
NSDateComponents *twoThirty = [NSDateComponents new];
twoThirty.hour = 2;
twoThirty.minute = 30;
```

```
[ [NSCalendar currentCalendar]
    enumerateDatesStartingAfterDate: [NSDate date]
        matchingComponents: twoThirty
        options: NSCalendarMatchNextTime
        usingBlock: ^(NSDate *date, BOOL exact, BOOL *stop) { ... }];
```

Enumerating Dates

Create NSDates representing 2:30AM every day

```
NSDateComponents *twoThirty = [NSDateComponents new];
twoThirty.hour = 2;
twoThirty.minute = 30;
```

```
[ [NSCalendar currentCalendar]
    enumerateDatesStartingAfterDate: [NSDate date]
        matchingComponents: twoThirty
        options: NSCalendarMatchNextTime
        usingBlock: ^(NSDate *date, BOOL exact, BOOL *stop) { ... }];
```

```
date = Jun 12, 2013, 2:30      exact = YES
```

Enumerating Dates

Create NSDates representing 2:30AM every day

```
NSDateComponents *twoThirty = [NSDateComponents new];
twoThirty.hour = 2;
twoThirty.minute = 30;
```

```
[ [NSCalendar currentCalendar]
    enumerateDatesStartingAfterDate: [NSDate date]
        matchingComponents: twoThirty
        options: NSCalendarMatchNextTime
        usingBlock: ^(NSDate *date, BOOL exact, BOOL *stop) { ... }];
```

```
date = Jun 12, 2013, 2:30      exact = YES
...
...
```

Enumerating Dates

Create NSDates representing 2:30AM every day

```
NSDateComponents *twoThirty = [NSDateComponents new];
twoThirty.hour = 2;
twoThirty.minute = 30;
```

```
[ [NSCalendar currentCalendar]
    enumerateDatesStartingAfterDate: [NSDate date]
        matchingComponents: twoThirty
        options: NSCalendarMatchNextTime
        usingBlock: ^(NSDate *date, BOOL exact, BOOL *stop) { ... }];
```

```
date = Jun 12, 2013, 2:30      exact = YES
...
date = Mar 8, 2014, 2:30      exact = YES
```

Enumerating Dates

Create NSDates representing 2:30AM every day

```
NSDateComponents *twoThirty = [NSDateComponents new];
twoThirty.hour = 2;
twoThirty.minute = 30;
```

```
[ [NSCalendar currentCalendar]
    enumerateDatesStartingAfterDate: [NSDate date]
        matchingComponents: twoThirty
        options: NSCalendarMatchNextTime
        usingBlock: ^(NSDate *date, BOOL exact, BOOL *stop) { ... }];
```

```
date = Jun 12, 2013, 2:30      exact = YES
...
date = Mar 8, 2014, 2:30       exact = YES
date = Mar 9, 2014, 3:00       exact = NO
```

Enumerating Dates

Create NSDates representing 2:30AM every day

```
NSDateComponents *twoThirty = [NSDateComponents new];
twoThirty.hour = 2;
twoThirty.minute = 30;
```

```
[ [NSCalendar currentCalendar]
    enumerateDatesStartingAfterDate: [NSDate date]
        matchingComponents: twoThirty
        options: NSCalendarMatchNextTime
        usingBlock: ^(NSDate *date, BOOL exact, BOOL *stop) { ... }];
```

```
date = Jun 12, 2013, 2:30      exact = YES
...
date = Mar 8, 2014, 2:30      exact = YES
date = Mar 9, 2014, 3:00      exact = NO
date = Mar 10, 2014, 2:30     exact = YES
...
```

Enumerating Dates

Create NSDates representing 2:30AM every day

```
NSDateComponents *twoThirty = [NSDateComponents new];
twoThirty.hour = 2;
twoThirty.minute = 30;
```

```
[ [NSCalendar currentCalendar]
    enumerateDatesStartingAfterDate: [NSDate date]
        matchingComponents: twoThirty
        options: NSCalendarMatchNextTime
        usingBlock: ^(NSDate *date, BOOL exact, BOOL *stop) { ... }];
```

```
date = Jun 12, 2013, 2:30      exact = YES
...
date = Mar 8, 2014, 2:30       exact = YES
date = Mar 9, 2014, 3:00       exact = NO
date = Mar 10, 2014, 2:30      exact = YES
...
```

Date and Time Handling

Solutions to Common Date and Time Challenges

Marina
Friday 11:30AM

Date and Time Handling

Solutions to Common Date and Time Challenges

Marina
Friday 11:30AM

Cocoa and Foundation Lab

Frameworks Lab
Wednesday 11:30-1:45PM

Contextual Localization Support

Handling plurals is hard!

Contextual Localization Support

Handling plurals is hard!

```
if (n == 1) {
    result =
        NSLocalizedString(@"A file is selected",
                        @"Message shown when 1 file is selected");
} else {
    result = [NSString localizedStringWithFormat:
        NSLocalizedString(@"%d files are selected",
                        @"Message shown when multiple files selected"),
        n];
}
```

Contextual Localization Support

Handling plurals in French

```
if (n <= 1) {
    result = [NSString localizedStringWithFormat:
        NSLocalizedString(@"%d file is selected",
            @"Message shown when 0 or 1 file is selected"),
        n];
} else {
    result = [NSString localizedStringWithFormat:
        NSLocalizedString(@"%d files are selected",
            @"Message shown when multiple files selected"),
        n];
}
```

Contextual Localization Support

Handling plurals in Arabic

```
if (n == 0) {
    result = NSLocalizedString(@"No file is selected", @"...");
} else if (n == 1) {
    result = NSLocalizedString(@"A file is selected", @"...");
} else if (n == 2) {
    result = NSLocalizedString(@"Two files are selected", @"...");
} else if ((n % 100 >= 3) && (n % 100 <= 10)) {
    result = [NSString localizedStringWithFormat:
        NSLocalizedString(@"Few (%d) are selected", @"..."), n];
} else if ((n % 100 >= 11) && (n % 100 <= 99)) {
    result = [NSString localizedStringWithFormat:
        NSLocalizedString(@"Many (%d) files are selected", @"..."), n];
} else {
    result = [NSString localizedStringWithFormat:
        NSLocalizedString(@"Other (%d) files are selected", @"..."), n];
}
```

Contextual Localization Support

- `stringsdict` resource files
 - “Localized Property List File”

Contextual Localization Support

- stringsdict resource files
 - “Localized Property List File”
- Provides localized forms based on plurality or gender

Contextual Localization Support

- stringsdict resource files
 - “Localized Property List File”
- Provides localized forms based on plurality or gender

- No new API!

- Just one code path:

```
result = [NSString localizedStringWithFormat:  
          NSLocalizedString(@"%d file(s) selected",  
                           @"Message shown for selected files"), n];
```

Contextual Localization Support

- stringsdict resource files
 - “Localized Property List File”
- Provides localized forms based on plurality or gender

- No new API!

- Just one code path:

```
result = [NSString localizedStringWithFormat:  
          NSLocalizedString(@"%d file(s) selected",  
                           @"Message shown for selected files"), n];
```

Contextual Localization Support

Sample stringsdict file

Contextual Localization Support

Sample stringsdict file

```
<dict>
    <key>%d file(s) selected</key>
    <dict>
        <key>NSStringLocalizedFormatKey</key>
        <string>%#@filecount@ selected</string>
        <key>filecount</key>
        <dict>
            <key>NSStringFormatSpecTypeKey</key>
            <string>NSStringPluralRuleType</string>
            <key>NSStringFormatValueTypeKey</key>
            <string>d</string>
            <key>one</key>
            <string>A file is</string>
            <key>other</key>
            <string>%d files are</string>
        </dict>
    </dict>
</dict>
```

Contextual Localization Support

Sample stringsdict file

```
<dict>
  <key>%d file(s) selected</key>
  <dict>
    <key>NSStringLocalizedFormatKey</key>
    <string>%#@filecount@ selected</string>
    <key>filecount</key>
    <dict>
      <key>NSStringFormatSpecTypeKey</key>
      <string>NSStringPluralRuleType</string>
      <key>NSStringFormatValueTypeKey</key>
      <string>d</string>
      <key>one</key>
      <string>A file is</string>
      <key>other</key>
      <string>%d files are</string>
    </dict>
  </dict>
</dict>
```

Contextual Localization Support

Sample stringsdict file

```
<dict>
  <key>%d file(s) selected</key>
  <dict>
    <key>NSStringLocalizedFormatKey</key>
    <string>%#@filecount@ selected</string>
    <key>filecount</key>
    <dict>
      <key>NSStringFormatSpecTypeKey</key>
      <string>NSStringPluralRuleType</string>
      <key>NSStringFormatValueTypeKey</key>
      <string>d</string>
      <key>one</key>
      <string>A file is</string>
      <key>other</key>
      <string>%d files are</string>
    </dict>
  </dict>
</dict>
```

Contextual Localization Support

Sample stringsdict file

```
<dict>
  <key>%d file(s) selected</key>
  <dict>
    <key>NSStringLocalizedFormatKey</key>
    <string>%#@filecount@ selected</string>
    <key>filecount</key>
    <dict>
      <key>NSStringFormatSpecTypeKey</key>
      <string>NSStringPluralRuleType</string>
      <key>NSStringFormatValueTypeKey</key>
      <string>d</string>
      <key>one</key>
      <string>A file is</string>
      <key>other</key>
      <string>%d files are</string>
    </dict>
  </dict>
</dict>
```

Contextual Localization Support

Sample stringsdict file

```
<dict>
  <key>%d file(s) selected</key>
  <dict>
    <key>NSStringLocalizedFormatKey</key>
    <string>%#@filecount@ selected</string>
    <key>filecount</key>
    <dict>
      <key>NSStringFormatSpecTypeKey</key>
      <string>NSStringPluralRuleType</string>
      <key>NSStringFormatValueTypeKey</key>
      <string>d</string>
      <key>one</key>
      <string>A file is</string>
      <key>other</key>
      <string>%d files are</string>
    </dict>
  </dict>
</dict>
```

Contextual Localization Support

Sample stringsdict file

```
<dict>
  <key>%d file(s) selected</key>
  <dict>
    <key>NSStringLocalizedFormatKey</key>
    <string>%#@filecount@ selected</string>
    <key>filecount</key>
    <dict>
      <key>NSStringFormatSpecTypeKey</key>
      <string>NSStringPluralRuleType</string>
      <key>NSStringFormatValueTypeKey</key>
      <string>d</string>
      <key>one</key>
      <string>A file is</string>
      <key>other</key>
      <string>%d files are</string>
    </dict>
  </dict>
</dict>
```

Contextual Localization Support

Sample stringsdict file

```
<dict>
  <key>%d file(s) selected</key>
  <dict>
    <key>NSStringLocalizedFormatKey</key>
    <string>%#@filecount@ selected</string>
    <key>filecount</key>
    <dict>
      <key>NSStringFormatSpecTypeKey</key>
      <string>NSStringPluralRuleType</string>
      <key>NSStringFormatValueTypeKey</key>
      <string>d</string>
      <key>one</key>
      <string>A file is</string>
      <key>other</key>
      <string>%d files are</string>
    </dict>
  </dict>
</dict>
```

Contextual Localization Support

Sample stringsdict file

```
<dict>
  <key>%d file(s) selected</key>
  <dict>
    <key>NSStringLocalizedFormatKey</key>
    <string>%#@filecount@ selected</string>
    <key>filecount</key>
    <dict>
      <key>NSStringFormatSpecTypeKey</key>
      <string>NSStringPluralRuleType</string>
      <key>NSStringFormatValueTypeKey</key>
      <string>d</string>
      <key>one</key>
      <string>A file is</string>
      <key>other</key>
      <string>%d files are</string>
    </dict>
  </dict>
</dict>
```

instancetype

instancetype

- `instancetype` indicates that the method returns an instance of its receiver

instancetype

- `instancetype` indicates that the method returns an instance of its receiver
- Already applied implicitly to `+alloc`, `+new`, `-init`

```
@interface NSObject
+ (instancetype)alloc;
+ (instancetype)new;
- (instancetype)init;
@end
```

instancetype

- `instancetype` indicates that the method returns an instance of its receiver
- Already applied implicitly to `+alloc`, `+new`, `-init`

```
@interface NSObject
+ (instancetype)alloc;
+ (instancetype)new;
- (instancetype)init;
@end
```

- In 10.9 declared on appropriate methods in a number of Foundation classes

instancetype

instancetype

```
NSMutableSet *mySet = [NSMutableArray array];
```

instancetype

```
NSMutableSet *mySet = [NSMutableArray array];
```



instancetype

```
NSMutableSet *mySet = [NSMutableArray array];  
[mySet addObject: @"Hello World"];
```

instancetype

```
NSMutableSet *mySet = [NSMutableArray array];  
[mySet addObject: @"Hello World"];
```

- On 10.8, no complaints, app “works”

instancetype

```
NSMutableSet *mySet = [NSMutableArray array];  
[mySet addObject: @"Hello World"];
```

- On 10.8, no complaints, app “works”
- On 10.9

```
prog.m:6:17: warning: incompatible pointer types initializing  
  'NSMutableSet *' with an expression of type 'NSMutableArray *'  
  [-Wincompatible-pointer-types]  
 NSMutableSet *mySet = [NSMutableArray array];  
 ^ ~~~~~
```

NSArray

NSArray

- NSArray has always had
 - `(id)lastObject;`

NSArray

- NSArray has always had
 - `(id)lastObject;`
- And now we've added

NSArray

- NSArray has always had
 - `(id)lastObject;`
- And now we've added
 - `(id)firstObject;`

NSArray

- NSArray has always had
 - `(id)lastObject;`
- And now we've added
 - `(id)firstObject;`
- Works back to OS X 10.6 and iOS 4

NSData

NSData

- Base64 encoding and decoding support
 - To/from NSString or NSData

NSData

- Base64 encoding and decoding support
 - To/from NSString or NSData
- dispatch_data_t bridging
 - Can cast dispatch_data_t to NSData *
 - But not the other way round

NSData

- Base64 encoding and decoding support
 - To/from NSString or NSData
- dispatch_data_t bridging
 - Can cast dispatch_data_t to NSData *
 - But not the other way round
- Efficient enumeration of NSData with discontiguous memory
 - `(void)enumerateByteRangesUsingBlock:`
`(void (^)(const void *bytes, NSRange range, BOOL *stop))block;`

NSURLComponents

NSURLComponents

- New class to parse and create NSURLs

NSURLComponents

- New class to parse and create NSURLs
 - Compliant with RFC3986/STD66

NSURLComponents

- New class to parse and create NSURLs
 - Compliant with RFC3986/STD66
- API
 - `(id)initWithURL:(NSURL *)url resolvingAgainstBaseURL:(BOOL)resolve;`

NSURLComponents

- New class to parse and create NSURLs
 - Compliant with RFC3986/STD66
- API
 - `(id)initWithURL:(NSURL *)url resolvingAgainstBaseURL:(BOOL)resolve;`
 - `(id)init;`

NSURLComponents

- New class to parse and create NSURLs
 - Compliant with RFC3986/STD66
- API
 - `(id)initWithURL:(NSURL *)url resolvingAgainstBaseURL:(BOOL)resolve;`
 - `(id)init;`

```
@property (copy) NSString *scheme;  
@property (copy) NSString *user;  
@property (copy) NSString *path;  
@property (copy) NSString *query;  
...
```

NSURLComponents

- New class to parse and create NSURLs
 - Compliant with RFC3986/STD66
- API
 - `(id)initWithURL:(NSURL *)url resolvingAgainstBaseURL:(BOOL)resolve;`
 - `(id)init;`
 -
 - `@property (copy) NSString *scheme;`
 - `@property (copy) NSString *user;`
 - `@property (copy) NSString *path;`
 - `@property (copy) NSString *query;`
 -
 - `...`
 -
 - `(NSURL *)URL;`

NSURLComponents

Creating an NSURL from NSURLComponents

NSURLComponents

Creating an NSURL from NSURLComponents

```
NSURLComponents *comps = [NSURLComponents alloc] init];
```

NSURLComponents

Creating an NSURL from NSURLComponents

```
NSURLComponents *comps = [NSURLComponents alloc] init];  
  
comps.scheme = @"http";
```

NSURLComponents

Creating an NSURL from NSURLComponents

```
NSURLComponents *comps = [NSURLComponents alloc] init];
```

```
comps.scheme = @"http";
```

```
comps.host = @"www.apple.com";
```

NSURLComponents

Creating an NSURL from NSURLComponents

```
NSURLComponents *comps = [NSURLComponents alloc] init];

comps.scheme = @"http";
comps.host   = @"www.apple.com";
comps.path   = @"/hello world.html";
```

NSURLComponents

Creating an NSURL from NSURLComponents

```
NSURLComponents *comps = [NSURLComponents alloc] init];

comps.scheme = @"http";
comps.host   = @"www.apple.com";
comps.path   = @"/hello world.html";

NSURL *url = [comps URL];
```

NSURLComponents

Creating an NSURL from NSURLComponents

```
NSURLComponents *comps = [NSURLComponents alloc] init];

comps.scheme = @"http";
comps.host   = @"www.apple.com";
comps.path   = @"/hello world.html";

NSURL *url = [comps URL];

NSLog(@"%@", [url absoluteString]);
```

NSURLComponents

Creating an NSURL from NSURLComponents

```
NSURLComponents *comps = [NSURLComponents alloc] init];
```

```
comps.scheme = @"http";  
comps.host = @"www.apple.com";  
comps.path = @"/hello world.html";
```

```
NSURL *url = [comps URL];
```

```
NSLog(@"%@", [url absoluteString]);
```

Output: <http://www.apple.com/hello%20world.html>

NSURLSession

NSURLSession

- Replacement APIs for NSURLConnection

NSURLSession

- Replacement APIs for NSURLConnection
- Improved control over configuration options

NSURLSession

- Replacement APIs for NSURLConnection
- Improved control over configuration options
- Ability to perform out-of-process downloads and uploads

NSURLSession

- Replacement APIs for NSURLConnection
- Improved control over configuration options
- Ability to perform out-of-process downloads and uploads
- Convenience APIs for simple, cancelable asynchronous operation

NSURLSession

What's New in Foundation Networking

Mission
Wednesday 9:00AM

NSURLSession

What's New in Foundation Networking

Mission
Wednesday 9:00AM

Foundation Networking Lab

Core OS Lab B
Wednesday 10:15-1:45PM

Core Data Improvements

Core Data Improvements

- iCloud Support

Core Data Improvements

- iCloud Support
 - Core Data managed fallback store

Core Data Improvements

- iCloud Support
 - Core Data managed fallback store
 - Asynchronous iCloud initialization

Core Data Improvements

- iCloud Support
 - Core Data managed fallback store
 - Asynchronous iCloud initialization
 - Automatic handling of account changes

Core Data Improvements

- iCloud Support
 - Core Data managed fallback store
 - Asynchronous iCloud initialization
 - Automatic handling of account changes
 - Many other fixes and enhancements

Core Data Improvements

- iCloud Support
 - Core Data managed fallback store
 - Asynchronous iCloud initialization
 - Automatic handling of account changes
 - Many other fixes and enhancements
- Write ahead logging (WAL) journaling on by default
 - Improved reliability
 - Improved performance
 - Improved concurrency

Core Data

What's New in Core Data and iCloud

Pacific Heights
Wednesday 9:00AM

Core Data Performance Optimization and Debugging

Nob Hill
Wednesday 2:00PM

Core Data

What's New in Core Data and iCloud	Pacific Heights Wednesday 9:00AM	
Core Data Performance Optimization and Debugging	Nob Hill Wednesday 2:00PM	
Core Data Lab	Services Lab B Wednesday 3:15-6:00PM	
Core Data Lab	Frameworks Lab A Thursday 2:00-4:15PM	
Core Data Lab	Services Lab A Friday 9:00-11:15AM	

Core Data

What's New in Core Data and iCloud	Pacific Heights Wednesday 9:00AM	
Core Data Performance Optimization and Debugging	Nob Hill Wednesday 2:00PM	
Core Data Lab	Services Lab B Wednesday 3:15-6:00PM	
Core Data Lab	Frameworks Lab A Thursday 2:00-4:15PM	
Core Data Lab	Services Lab A Friday 9:00-11:15AM	
iCloud Lab	Tools Lab A Wednesday 9:00-12:30PM	
iCloud Lab	Tools Lab C Thursday 9:00-12:30PM	

Modules

Better way to import a framework's interface

Modules

Better way to import a framework's interface

- Instead of

```
#import <UIKit/UIKit.h>
```

Modules

Better way to import a framework's interface

- Instead of

```
#import <UIKit/UIKit.h>
```

```
do
```

```
@import UIKit;
```

Modules

Better way to import a framework's interface

- Instead of

```
#import <UIKit/UIKit.h>
```

```
do
```

```
@import UIKit;
```

- Loads binary representation

Modules

Better way to import a framework's interface

- Instead of

```
#import <UIKit/UIKit.h>
```

```
do
```

```
@import UIKit;
```

- Loads binary representation
- More flexible than precompiled headers

Modules

Better way to import a framework's interface

- Instead of

```
#import <Cocoa/Cocoa.h>
```

```
do
```

```
@import Cocoa;
```

- Loads binary representation
- More flexible than precompiled headers
- Immune to effects of local macro definitions

Modules

Better way to import a framework's interface

- Instead of

```
#import <Cocoa/Cocoa.h>
```

```
do
```

```
@import Cocoa;
```

- Loads binary representation
- More flexible than precompiled headers
- Immune to effects of local macro definitions
- Enabled for new projects by default

Objective-C

Advances in Objective-C

Mission
Tuesday 4:30PM

Objective-C

Advances in Objective-C	Mission Tuesday 4:30PM	
Objective-C and LLVM Lab	Tools Lab B Wednesday 9:00-12:30PM	
Objective-C and LLVM Lab	Tools Lab C Thursday 2:00-4:15PM	

New Frameworks

New Frameworks

- MapKit

New Frameworks

- MapKit
- SpriteKit

New Frameworks

- MapKit
- SpriteKit
- AVKit

New Frameworks

Moving to AVKit and AV Foundation	Pacific Heights Tuesday 4:30PM	
Introduction to SpriteKit	Presidio Wednesday 11:30AM	
What's New in MapKit	Presidio Thursday 9:00AM	

Text Kit

Text Kit

- OS X Cocoa Text

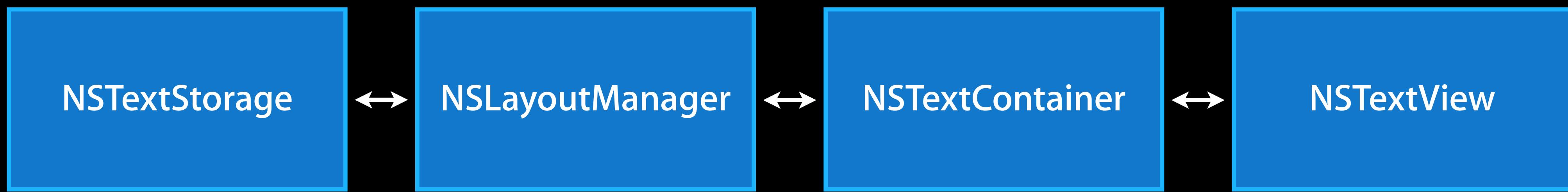
Text Kit

- OS X Cocoa Text

NSTextView

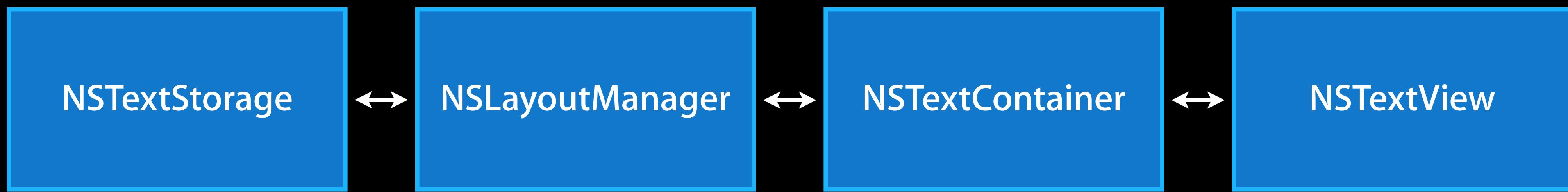
Text Kit

- OS X Cocoa Text



Text Kit

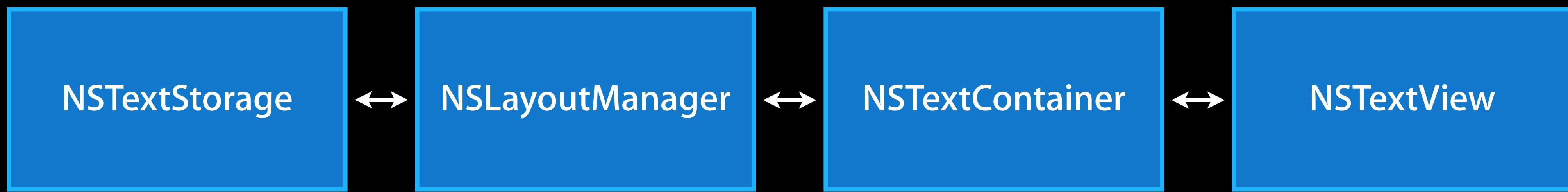
- OS X Cocoa Text



- iOS Text Kit

Text Kit

- OS X Cocoa Text

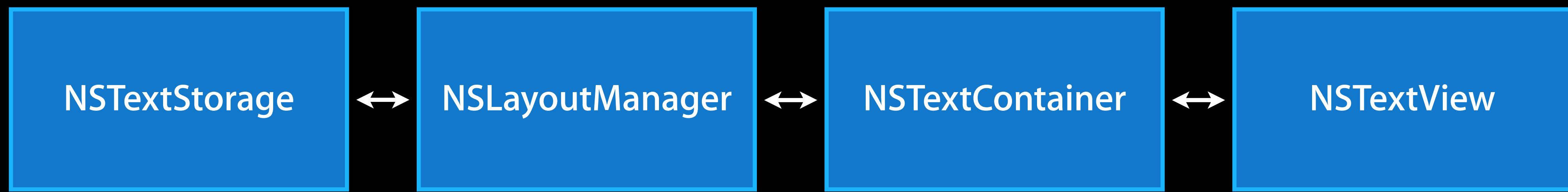


- iOS Text Kit

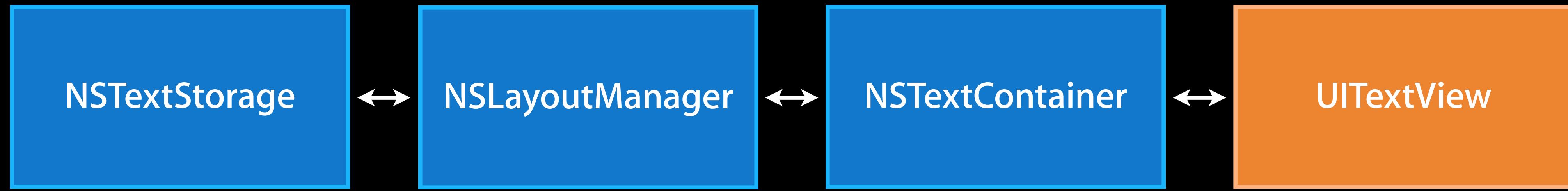


Text Kit

- OS X Cocoa Text



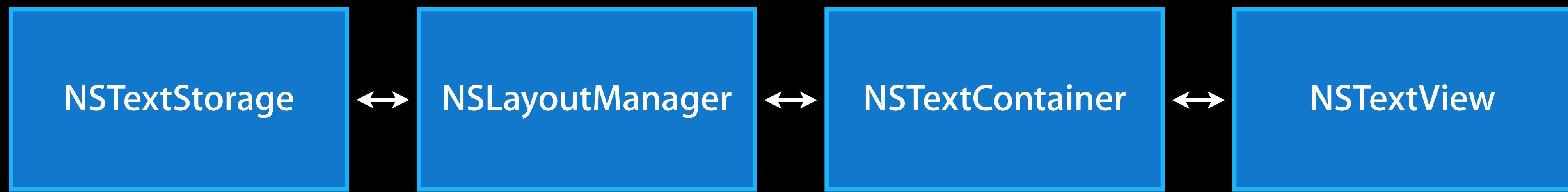
- iOS Text Kit



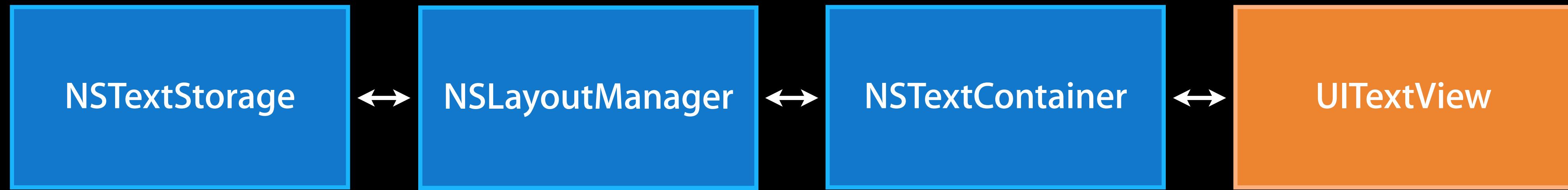
Text Kit



- OS X Cocoa Text



- iOS Text Kit



Text Kit



Introducing Text Kit

Presidio
Wednesday 2:00PM

Advanced Text Layouts and Effects with Text Kit

Mission
Thursday 2:00PM

Some Other Improvements and Changes

- NSTableView/NSScrollView improvements
- Layer-backed view improvements
- NSSlider new APIs and deprecations
- Additional built-in services and properties in NSSharingService
- NSColor convenience APIs for increased UIColor parity
- NSNib/NSBundle nib loading deprecations

Some Other Improvements and Changes

- NSURL improvements
- Built-in NSCharacterSets for URL handling
- NSMetadata APIs and Foundation-level attribute and scope names
- UserDefaults suite APIs
- NSScanner unsignedLongLong support

Some Other Improvements and Changes

- NSURL improvements
- Built-in NSCharacterSets for URL handling
- NSMetadata APIs and Foundation-level attribute and scope names
- UserDefaults suite APIs
- NSScanner unsignedLongLong support

Please refer to the 10.9 release notes!

10.8 and Earlier Features to Adopt

iCloud Support

Cocoa XPC

Auto Save

ARC

Auto Localization

Lookup

64 Bit

Retina Support

File Coordination

Full Screen

Gestures

Overlay Scrollers

*Application
Sandboxing*

Versions

Sharing Service

Resume

Auto Layout

Modern Obj-C Syntax

*User Notification
Center*

*Automatic
Termination*



Related Sessions

Related Sessions

Advances in Objective-C	Mission Tuesday 4:30PM	
What's New in Core Data and iCloud	Pacific Heights Wednesday 9:00AM	
What's New in Foundation Networking	Mission Wednesday 9:00AM	
Improving Power Efficiency with App Nap	Pacific Heights Wednesday 10:15AM	
Best Practices for Cocoa Animation	Marina Wednesday 2:00PM	
Optimizing Drawing and Scrolling on OS X	Marina Wednesday 3:15PM	
Bringing Your iOS Apps to OS X	Nob Hill Thursday 9:00AM	
Solutions to Common Date and Time Challenges	Marina Friday 11:30AM	

Labs

Cocoa Lab	Frameworks Lab B Tuesday 4:30-6:00PM	
Foundation Networking Lab	Core OS Lab B Wednesday 10:15-1:45PM	
Cocoa and Foundation Lab	Frameworks Lab A Wednesday 11:30-1:45PM	
Auto Layout Lab	Tools Lab A Wednesday 2:00-6:00PM	
Full Screen and Cocoa Lab	Frameworks Lab A Thursday 9:00-10:00AM	
NSTableView, NSView, and Cocoa Lab	Frameworks Lab A Thursday 10:15-11:15AM	
iOS to OS X Conversion Lab	Frameworks Lab A Thursday 11:30AM-1:45PM	
Cocoa Animations, Drawing, and Cocoa Lab	Frameworks Lab A Friday 9:00-11:15AM	

Labs

Core Data Lab	Services Lab B Wednesday 3:15-6:00PM	
Core Data Lab	Frameworks Lab A Thursday 2:00-4:15PM	
Core Data Lab	Services Lab A Friday 9:00-11:15AM	
iCloud Lab	Tools Lab A Wednesday 9:00-12:30PM	
iCloud Lab	Tools Lab C Thursday 9:00-12:30PM	

More Information

Jake Behrens

UI Frameworks Evangelist

behrens@apple.com

Documentation

Mac Dev Center

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

Application Kit Release Notes for 10.9

Foundation Kit Release Notes for 10.9

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

Apple Developer Forums

<http://devforums.apple.com>

