Full Screen and Aqua Changes

Session 113

Troy Stephens, Dan Schimpf, and Patrick Heynen
Application Frameworks Engineers

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

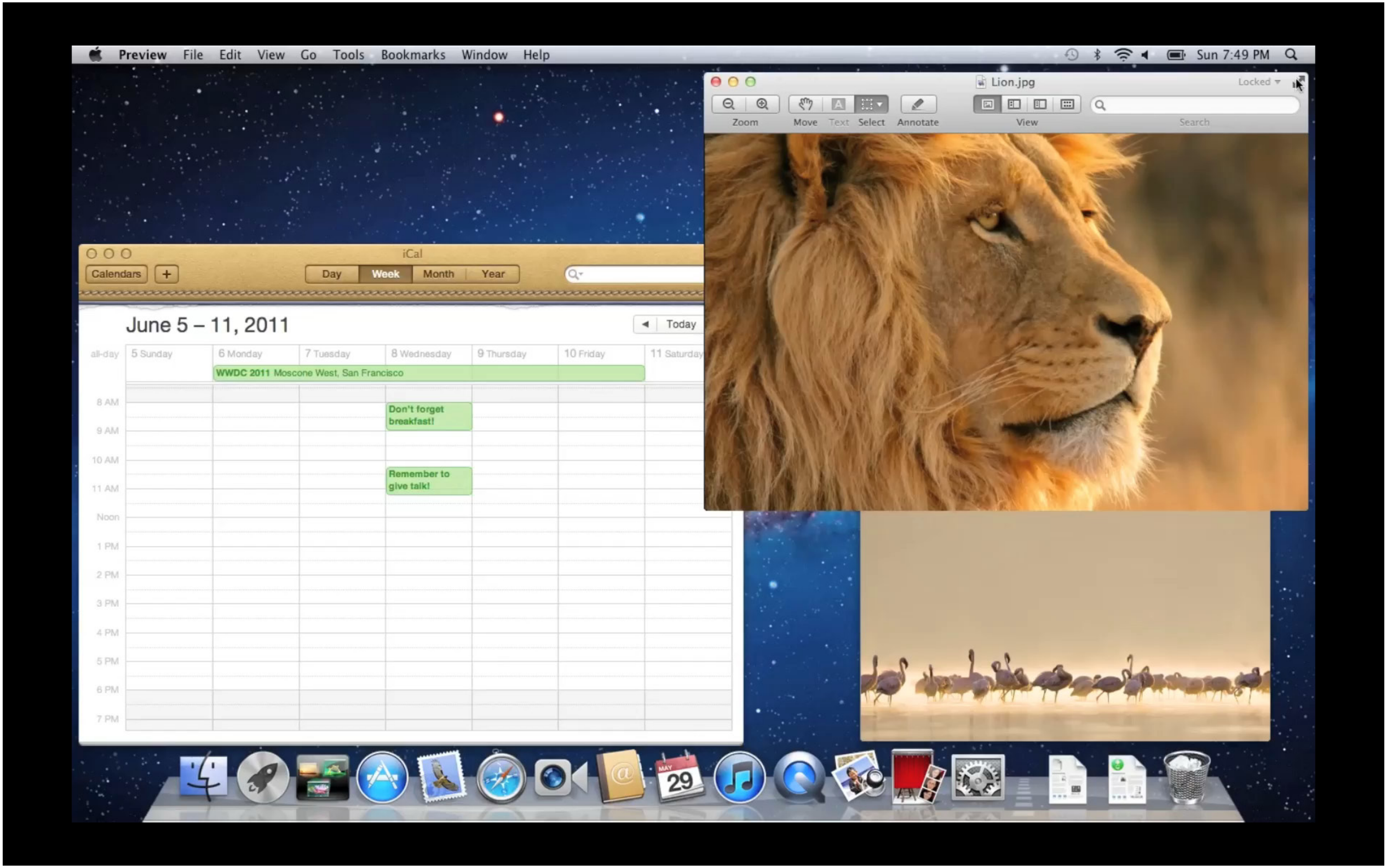
Topics

- Full Screen
- Popovers
- Appearance Changes
- Resolution Independence

Full Screen

Creating immersive user interfaces

Troy Stephens
Application Frameworks Engineer



June 5 – 11, 2011

all-day	5 Sunday	6 Monday	7 Tuesday	8 Wednesday	9 Thursday	10 Friday	11 Saturday
		WWDC 2011 Moscone West, San Francisco					
8 AM				Don't forget breakfast!			
9 AM							
10 AM							
11 AM				Remember to give talk!			
Noon							
1 PM							
2 PM							
3 PM							
4 PM							
5 PM							
6 PM							
7 PM							

Full Screen Mode

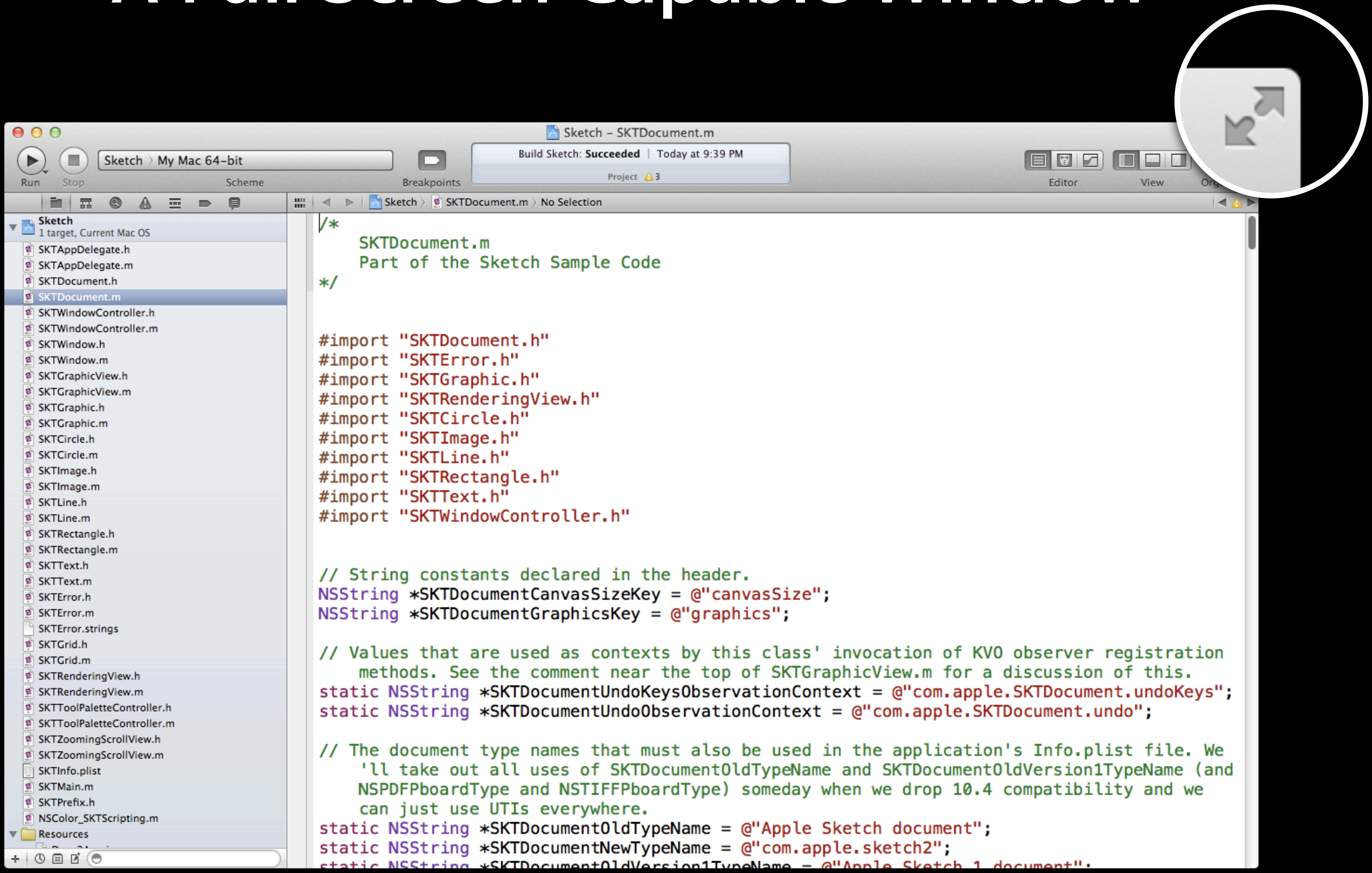
User benefits

- Focus attention on a single task
- Make the most of screen real estate

Why Make “Full Screen” a System Feature?

- Per-window model
- Consistent user experience
  - Standard enter/exit UI
  - Standard navigation

A Full Screen Capable Window



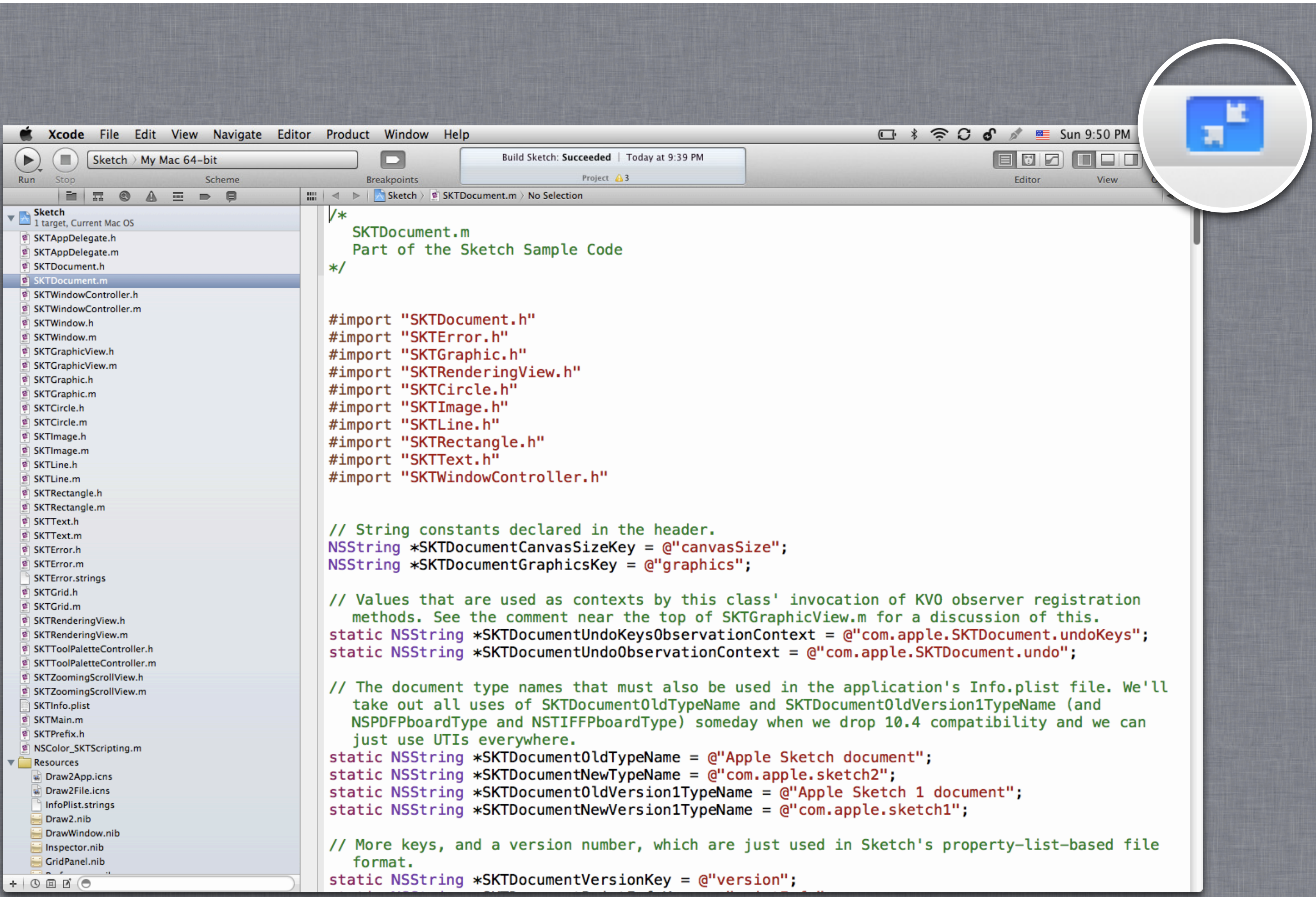
```
/*
    SKTDocument.m
    Part of the Sketch Sample Code
*/


#import "SKTDocument.h"
#import "SKTError.h"
#import "SKTGraphic.h"
#import "SKTRenderingView.h"
#import "SKTCircle.h"
#import "SKTImage.h"
#import "SKTLine.h"
#import "SKTRectangle.h"
#import "SKTText.h"
#import "SKTWindowController.h"


// String constants declared in the header.
NSString *SKTDocumentCanvasSizeKey = @"canvasSize";
NSString *SKTDocumentGraphicsKey = @"graphics";

// Values that are used as contexts by this class' invocation of KVO observer registration
    methods. See the comment near the top of SKTGraphicView.m for a discussion of this.
static NSString *SKTDocumentUndoKeysObservationContext = @"com.apple.SKTDocument.undoKeys";
static NSString *SKTDocumentUndoObservationContext = @"com.apple.SKTDocument.undo";

// The document type names that must also be used in the application's Info.plist file. We
    'll take out all uses of SKTDocumentOldTypeName and SKTDocumentOldVersion1TypeName (and
    NSPDFPboardType and NSTIFFPboardType) someday when we drop 10.4 compatibility and we
    can just use UTIs everywhere.
static NSString *SKTDocumentOldTypeName = @"Apple Sketch document";
static NSString *SKTDocumentNewTypeName = @"com.apple.sketch2";
static NSString *SKTDocumentOldVersion1TypeName = @"Apple Sketch 1 document";
```



```
/*
    SKTDocument.m
    Part of the Sketch Sample Code
*/


#import "SKTDocument.h"
#import "SKTError.h"
#import "SKTGraphic.h"
#import "SKTRenderingView.h"
#import "SKTCircle.h"
#import "SKTImage.h"
#import "SKTLine.h"
#import "SKTRectangle.h"
#import "SKTText.h"
#import "SKTWindowController.h"


// String constants declared in the header.
NSString *SKTDocumentCanvasSizeKey = @"canvasSize";
NSString *SKTDocumentGraphicsKey = @"graphics";

// Values that are used as contexts by this class' invocation of KVO observer registration
    methods. See the comment near the top of SKTGraphicView.m for a discussion of this.
static NSString *SKTDocumentUndoKeysObservationContext = @"com.apple.SKTDocument.undoKeys";
static NSString *SKTDocumentUndoObservationContext = @"com.apple.SKTDocument.undo";

// The document type names that must also be used in the application's Info.plist file. We'll
    take out all uses of SKTDocumentOldTypeName and SKTDocumentOldVersion1TypeName (and
    NSPDFPboardType and NSTIFFPboardType) someday when we drop 10.4 compatibility and we can
    just use UTIs everywhere.
static NSString *SKTDocumentOldTypeName = @"Apple Sketch document";
static NSString *SKTDocumentNewTypeName = @"com.apple.sketch2";
static NSString *SKTDocumentOldVersion1TypeName = @"Apple Sketch 1 document";
static NSString *SKTDocumentNewVersion1TypeName = @"com.apple.sketch1";

// More keys, and a version number, which are just used in Sketch's property-list-based file
    format.
static NSString *SKTDocumentVersionKey = @"version";
```

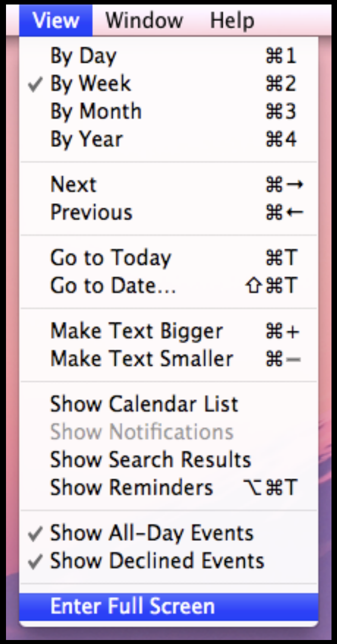
Full Screen Checklist

The Basics

- ✓ Specify which windows can be made full screen
- ✓ Add an “Enter Full Screen” menu item

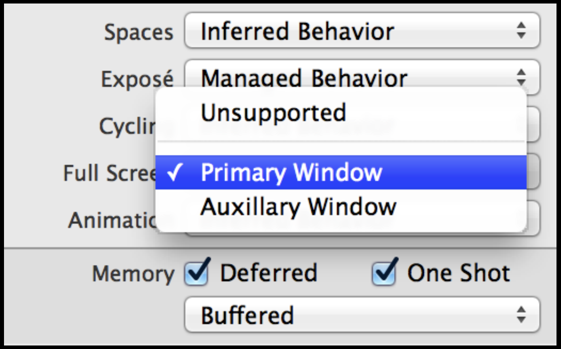
Making Your App Shine

- ★ Consider auto-hiding your window’s toolbar
- ★ Customize the window’s full-screen size?
- ★ Modify the window’s contents or layout for full screen?
- ★ Provide a separate full-screen window?
- ★ Provide a custom enter/exit animation?



Adding Full Screen Support

Specify which windows can be made full screen



Spaces: Inferred Behavior

Exposé: Managed Behavior

Cyclir

Full Scree

Animatic

Unsupported

✓ Primary Window

Auxillary Window

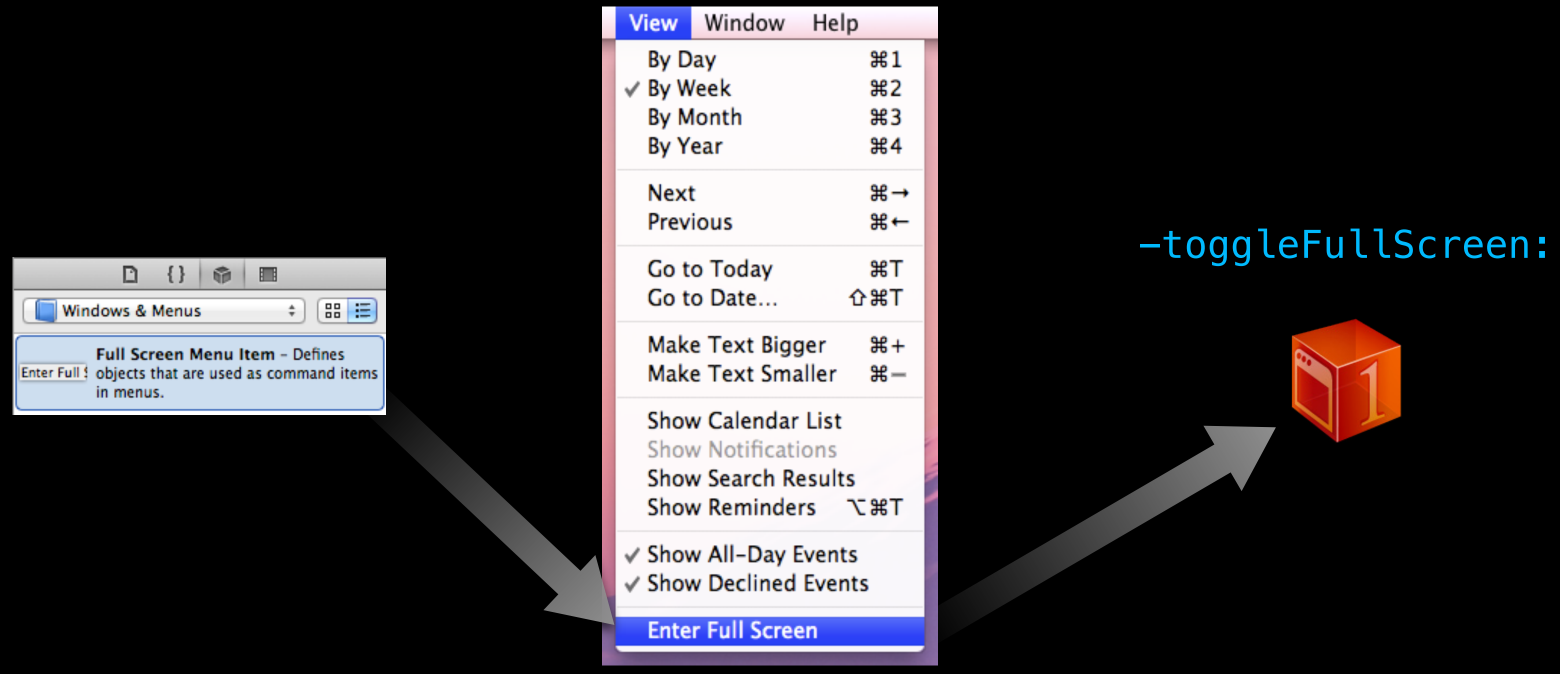
Memory: ☑ Deferred ☑ One Shot

Buffered

```
[window setCollectionBehavior:NSWindowCollectionBehaviorFullScreenPrimary
                              | ...];
```

Adding Full Screen Support

Add an “Enter Full Screen” menu item



Windows & Menus

Full Screen Menu Item – Defines objects that are used as command items in menus.

`-toggleFullScreen:`

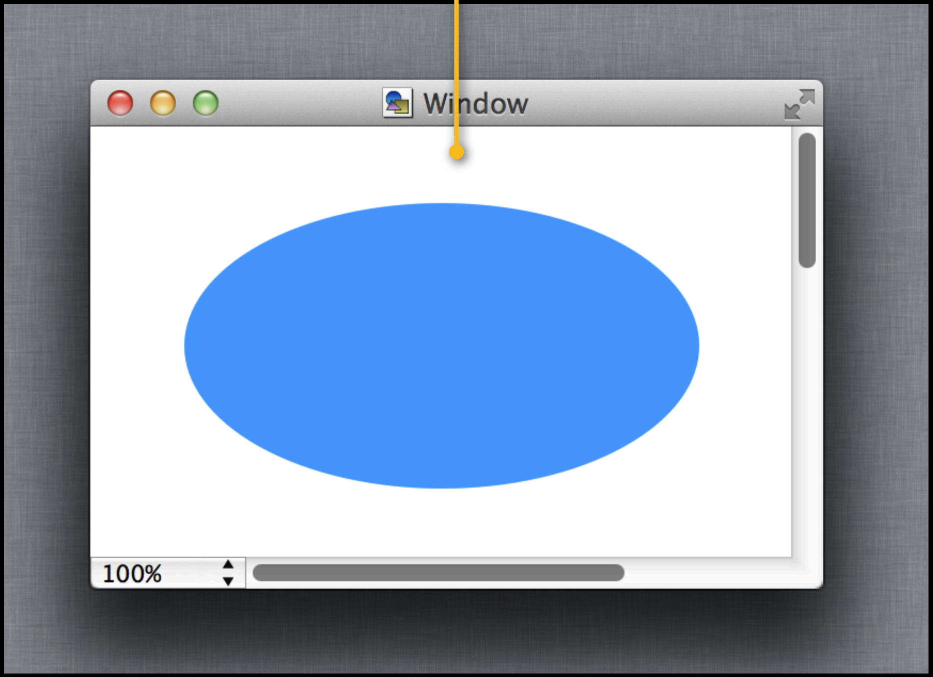
Demo

Adding full screen support to “Sketch”

Troy Stephens
Application Frameworks Engineer

Full Screen Window Style Mask

NSFullScreenWindowMask



Full Screen Presentation Options Bit

```
if ([NSApp presentationOptions] & NSApplicationPresentationFullScreen) {
    // System is in a Full Screen space for one of my app’s windows.
}
```

Automatically Hiding the Toolbar

```
- (NSApplicationPresentationOptions)window:(NSWindow *)window
willUseFullScreenPresentationOptions:(NSApplicationPresentationOptions)options
{
    // Tell AppKit to show/hide Toolbar together with Menu Bar.
    return (options | NSApplicationPresentationAutoHideToolbar);
}
```

Full Screen Notifications

Notification	NSWindow Delegate Method
`NSWindowWillEnterFullScreenNotification`	`– windowWillEnterFullScreen:`
`NSWindowDidEnterFullScreenNotification`	`– windowDidEnterFullScreen:`
`NSWindowWillExitFullScreenNotification`	`– windowWillExitFullScreen:`
`NSWindowDidExitFullScreenNotification`	`– windowDidExitFullScreen:`

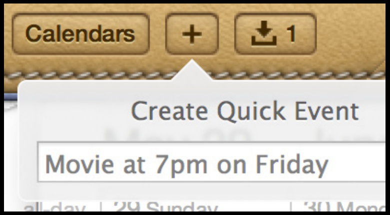
Full Screen Notifications

Notification	NSWindow Delegate Method
`NSWindowWillEnterFullScreenNotification`	`– windowWillEnterFullScreen:`
`NSWindowDidEnterFullScreenNotification`	`– windowDidEnterFullScreen:`
`NSWindowWillExitFullScreenNotification`	`– windowWillExitFullScreen:`
`NSWindowDidExitFullScreenNotification`	`– windowDidExitFullScreen:`

Popovers

Dan Schimpf
Application Frameworks Engineer

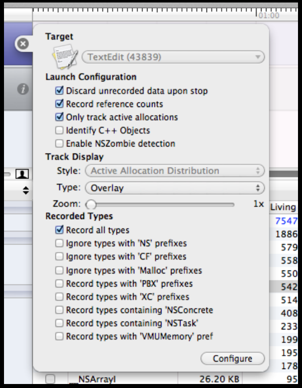
Existing Popovers



Create Quick Event

Movie at 7pm on Friday

iCal


Target

TextEdit (43839)

Launch Configuration

- ☑ Discard unrecorded data upon stop
- ☑ Record reference counts
- ☑ Only track active allocations
- □ Identify C++ Objects
- □ Enable NSZombie detection

Track Display

Style: Active Allocation Distribution

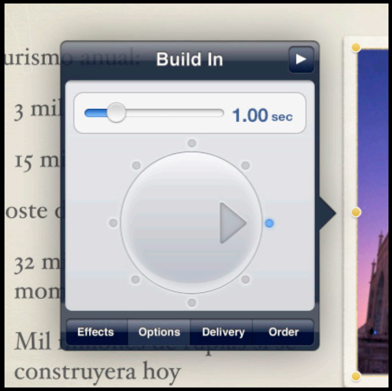
Type: Overlay

Zoom: 1x

Recorded Types

- ☑ Record all types
- □ Ignore types with 'NS' prefixes
- □ Ignore types with 'CF' prefixes
- □ Ignore types with 'Malloc' prefixes
- □ Record types with 'PBX' prefixes
- □ Record types with 'XC' prefixes
- □ Record types containing 'NSConcrete
- □ Record types containing 'NSTask'
- □ Record types with 'VMUMemory' pref

Instruments



UIPopoverController

What Is a Popover?

- ~~Light muffin from a thin batter~~
- Small chunks of user interface
- Relevant to a specific screen location
- Contextual to what is clicked

NSPopover

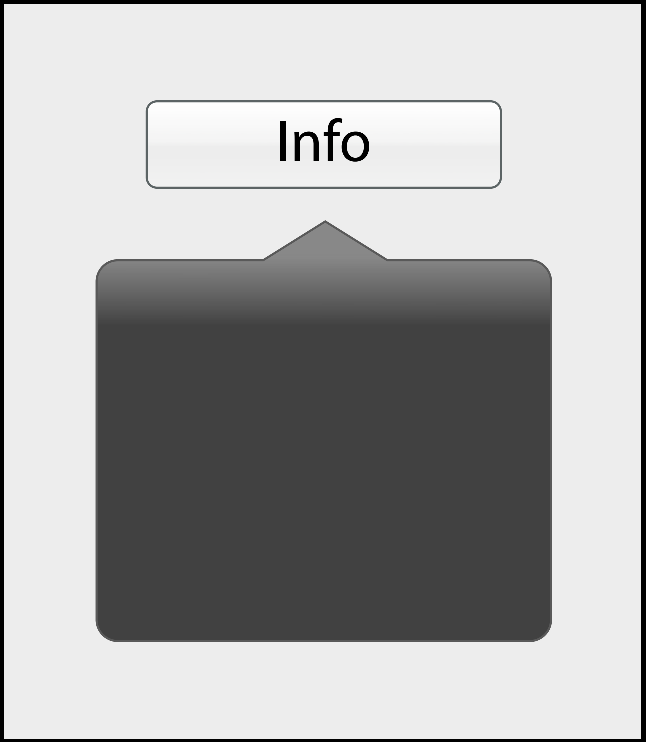
- New class in AppKit in Lion
- Hosts any content view relative to any another view

Creating Popovers

- NSPopover hosts an NSViewController
- NSViewController controls the view and can load a nib for the popover
- Use a new view controller for each popover instance

Popover Appearance

- Minimal (default)
- HUD



Showing Popovers

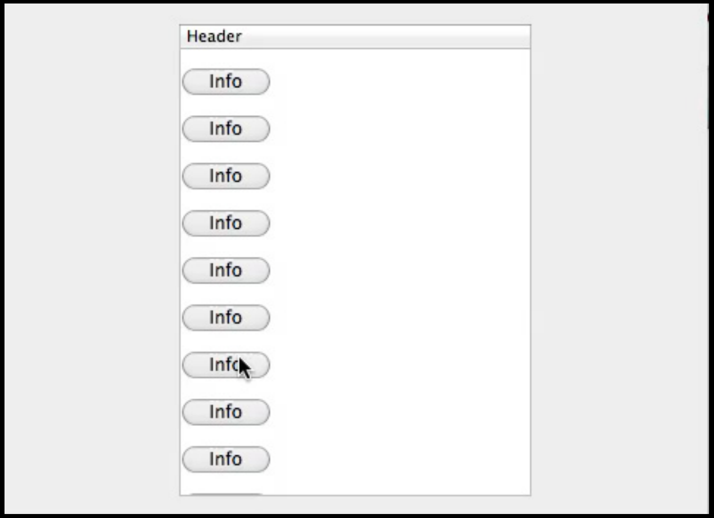
```
-[NSPopover showRelativeToRect:ofView:preferredEdge:]
```

- Relative to a rect inside an existing view onscreen
- Can hint which side is best for the popover

Moving Popovers

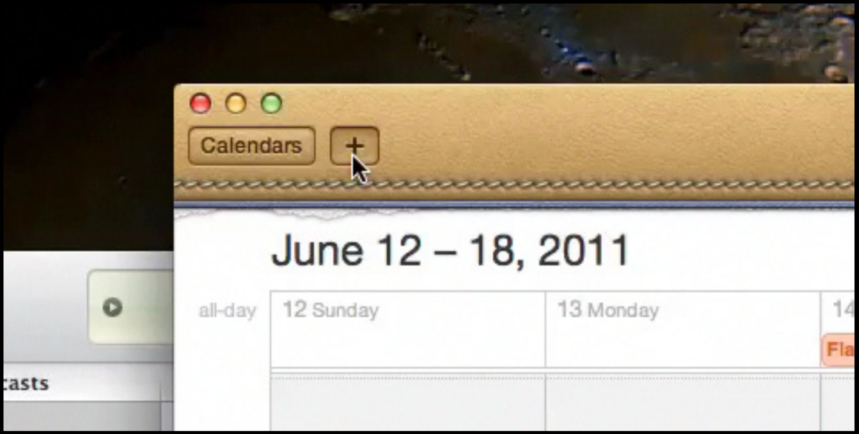
- Popovers will automatically track the positioning rect of the original view

Use `-setPositioningRect:` to update this rect if internal view drawing moves



Popover Animation

- Animates onscreen
- Animates movement
- On by default



Dismissing Popovers

- Application defined (default)
- Transient
- Semi-transient

Detaching Popovers

- User can drag out popover to create a standalone window
- Popover delegate has control over this
    - `-detachableWindowForPopover:`
- Create a new instance of your view controller to populate the window

Demo

Popovers

When to Use Popovers

- Inline inspectors
- Single metadatum interface
- Expand UI in main window
- Not as a replacement for NSPopUpButton!

Popover UI Tips

- No border around content
- Keep it simple
- No close button unless you need it

Popovers and Toolbars

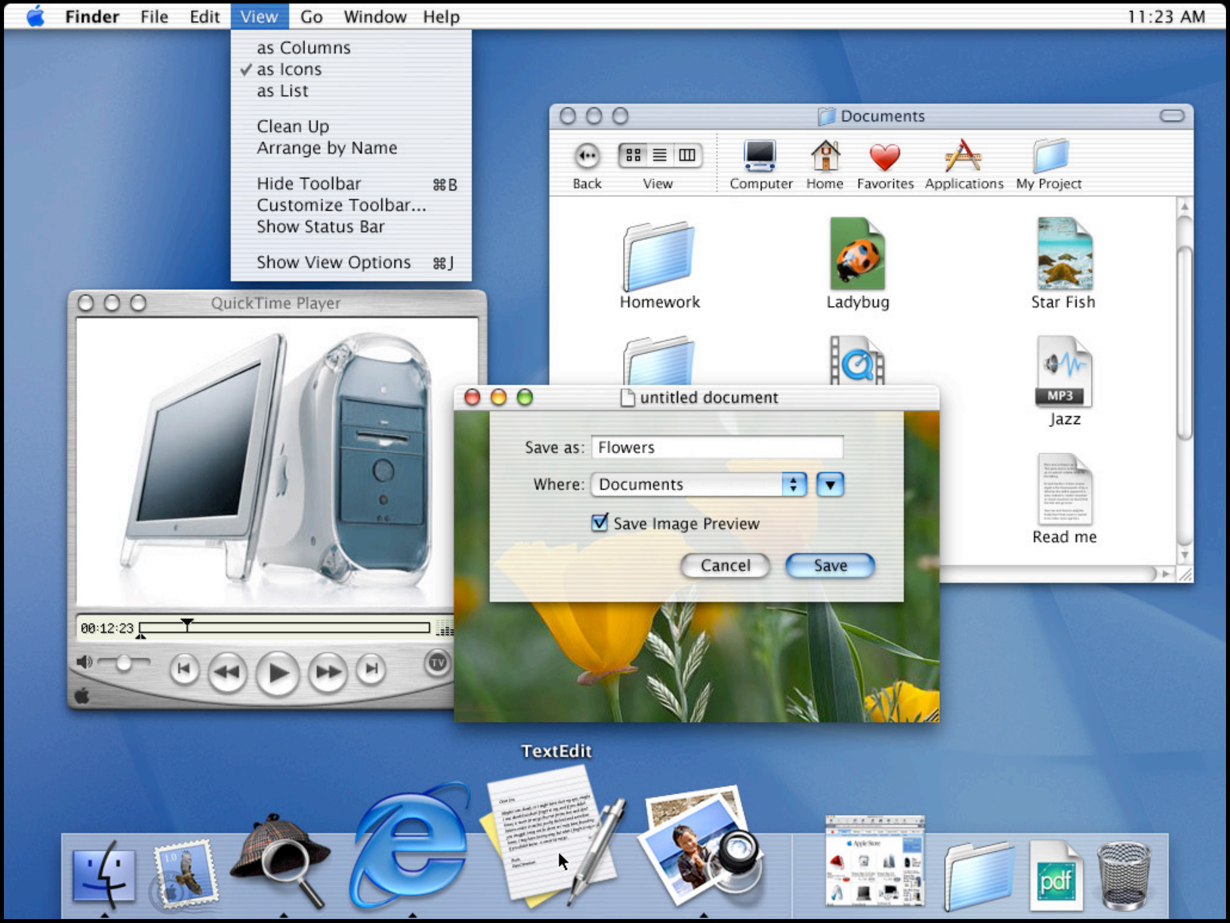
- Popovers require a rect in a view to show
- NSToolbarItem gives you neither
- Solution: use a custom view toolbar item with an NSButton to show the popover

Aqua Changes in Lion

What Is Aqua?

- The “Look and Feel” of Mac OS X
- Appearance and behavior of user interface controls

Mac OS X 10.0 “Cheetah”

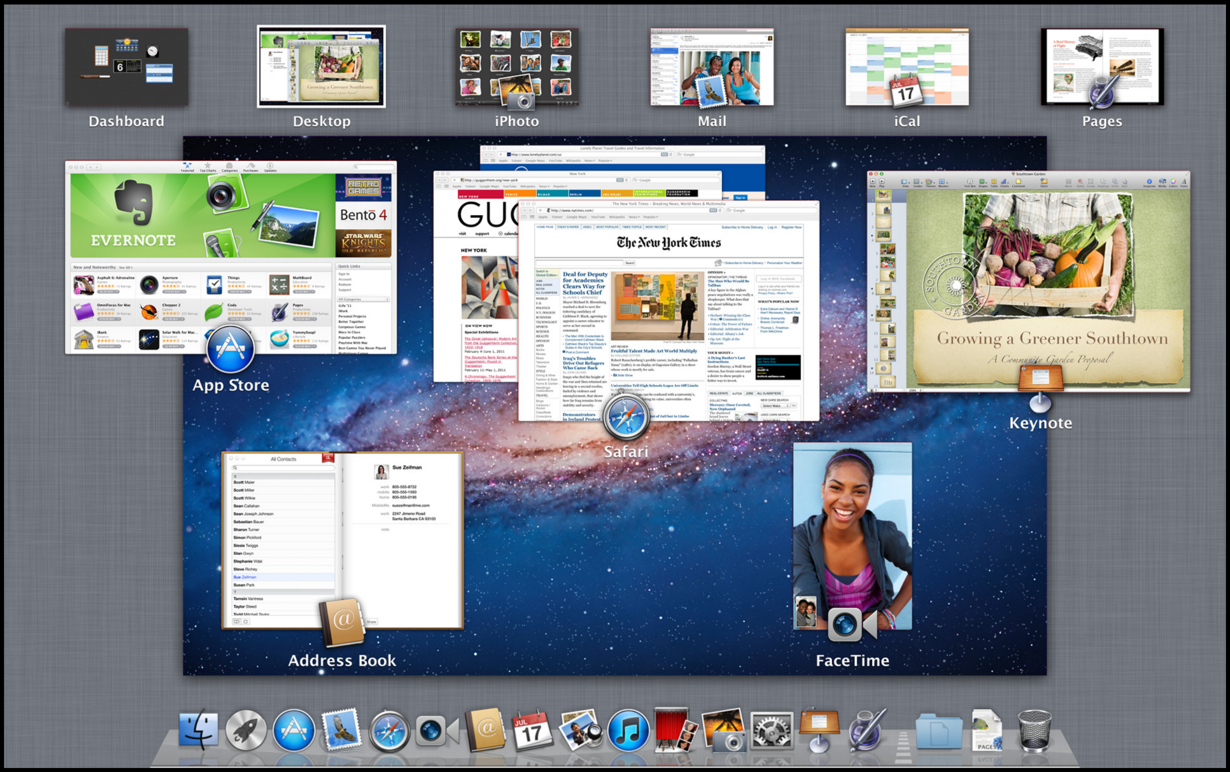


Save as: Flowers

Where: Documents

☑ Save Image Preview

Mac OS X 10.7 “Lion”



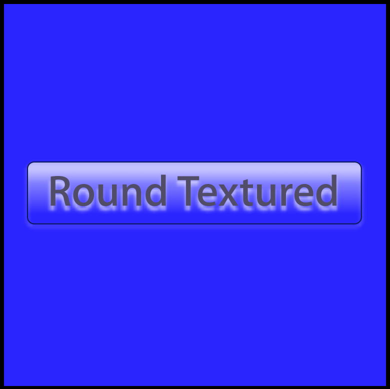
Aqua Changes in Lion

- Refreshed many controls
- More transparency
- More context-specific appearances
- Focus user attention on active window content

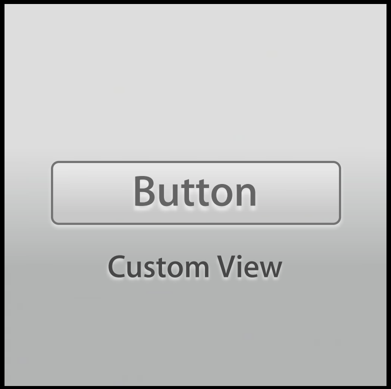
Textured Controls

- Some controls have “textured” appearances
- Intended for use on a textured background
- Transparent to show new window background

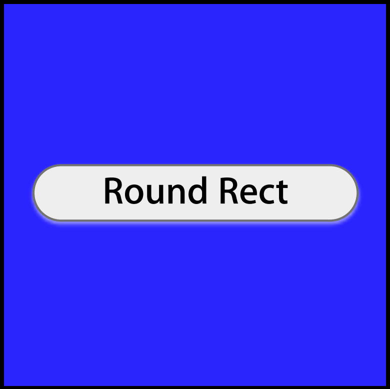
Button Usage



Textured buttons are
now transparent



Best for use
in toolbars



Consider using Round
Rect button instead

Inactive Controls

- Many controls now have unique appearances for inactive states
- Check to see if window is key
- Buttons are still usable with first click

Rollovers

- Indicates when the mouse is overhead
- Use rollovers to highlight an interactive control in an inactive window
- Highlight glyph but not bezel
- Use NSTrackingArea to watch for mouse entering

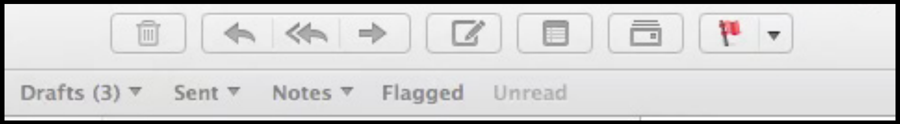


Image Templates

- Grayscale images
- AppKit will process the image for contextual effect
- Existing apps updated automatically

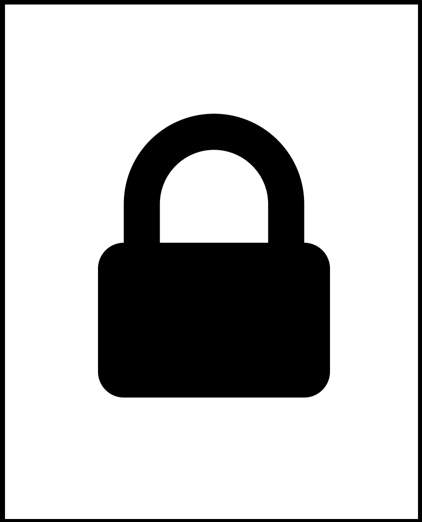
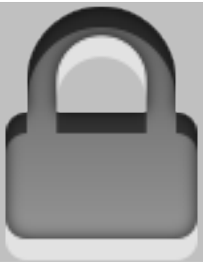
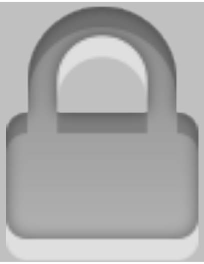
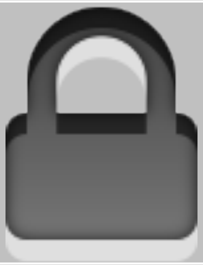
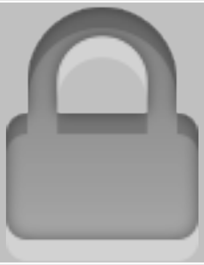
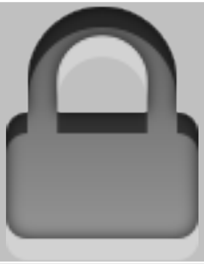
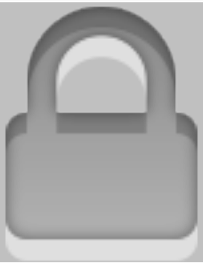
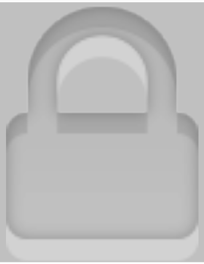


Image Templates

- Grayscale image, PDFs are best
- Think about the “shadow” a normal icon would cast
- Make sure it looks good when inverted as well
- Also consider different source list icon sizes
- File name ends in “Template”
- Or call -[NSImage setTemplate:]

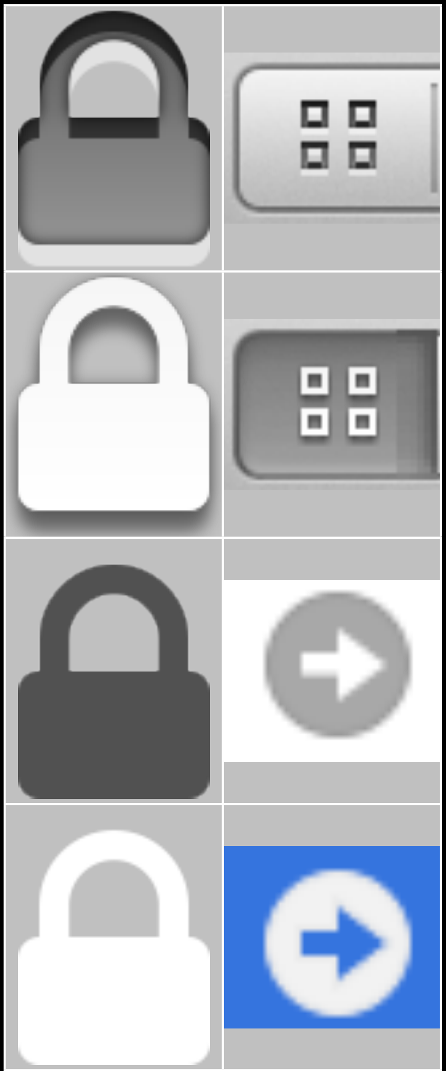
Drawing Templates

- Use NSButtonCell for interactive elements
- Use NSImageCell for static elements
- Remember to set the background style

	Active	Inactive
Normal		
Pressed		
Rollover	NA	
Disabled		

Background Styles

- Hints how your custom content should draw, including images and text
- Existing clients are updated automatically



Raised
Segmented control in toolbar

Lowered
Selected segmented control in toolbar

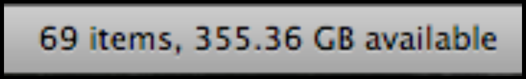
Light
Table view row

Dark
Selected table view row

Engraved Text

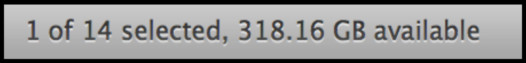
- New in Lion
- Use NSTextField or NSTextFieldCell to draw control text
- Redraw when window becomes inactive
- Existing clients updated automatically

Engraved Text



69 items, 355.36 GB available

Snow Leopard



1 of 14 selected, 318.16 GB available

Lion

Demo

Adapting to the new Aqua

Resolution Independence

Patrick Heynen
Cocoa Senior Software Engineer

Resolution Independence

State of the Pixel Union

- LCD display technology is pervasive
- Benefiting from Moore’s Law applied to displays
- The future of the pixel is small
- Challenge for software compatibility
- Back to the Mac!

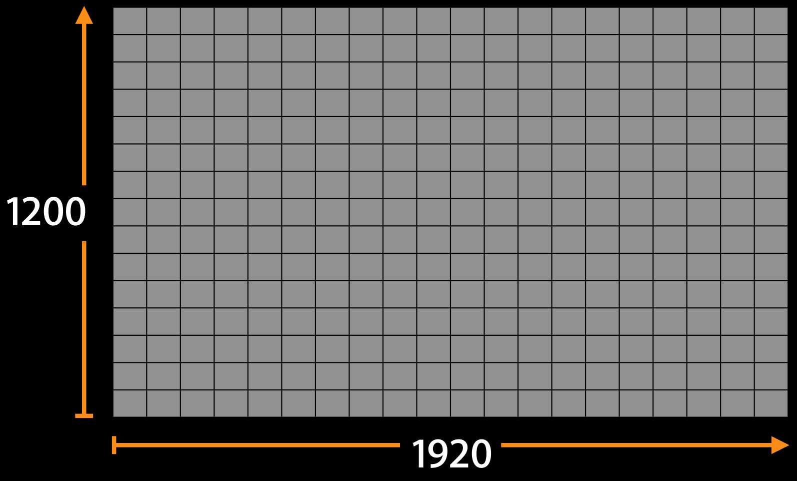
Resolution Independence

Architecture Update for Mac OS X Lion

Quartz Screen Scaling

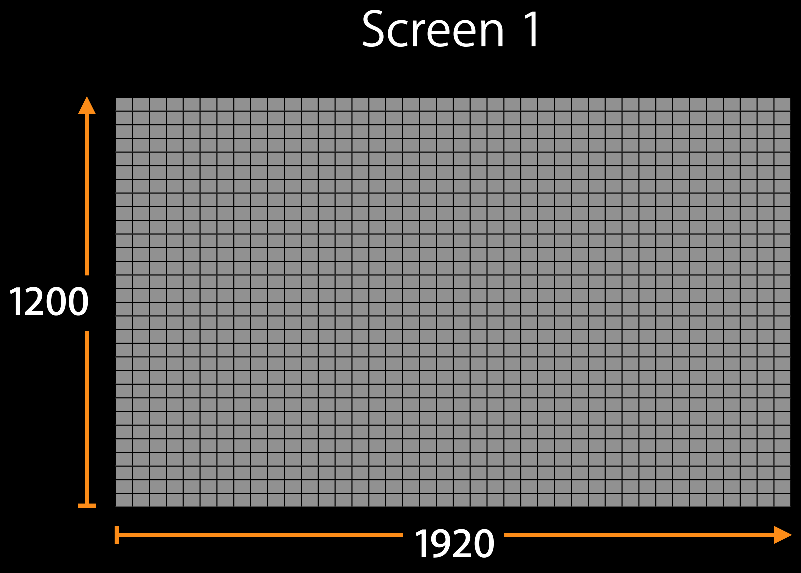
Device-independent drawing model extended to screens

Screen 0



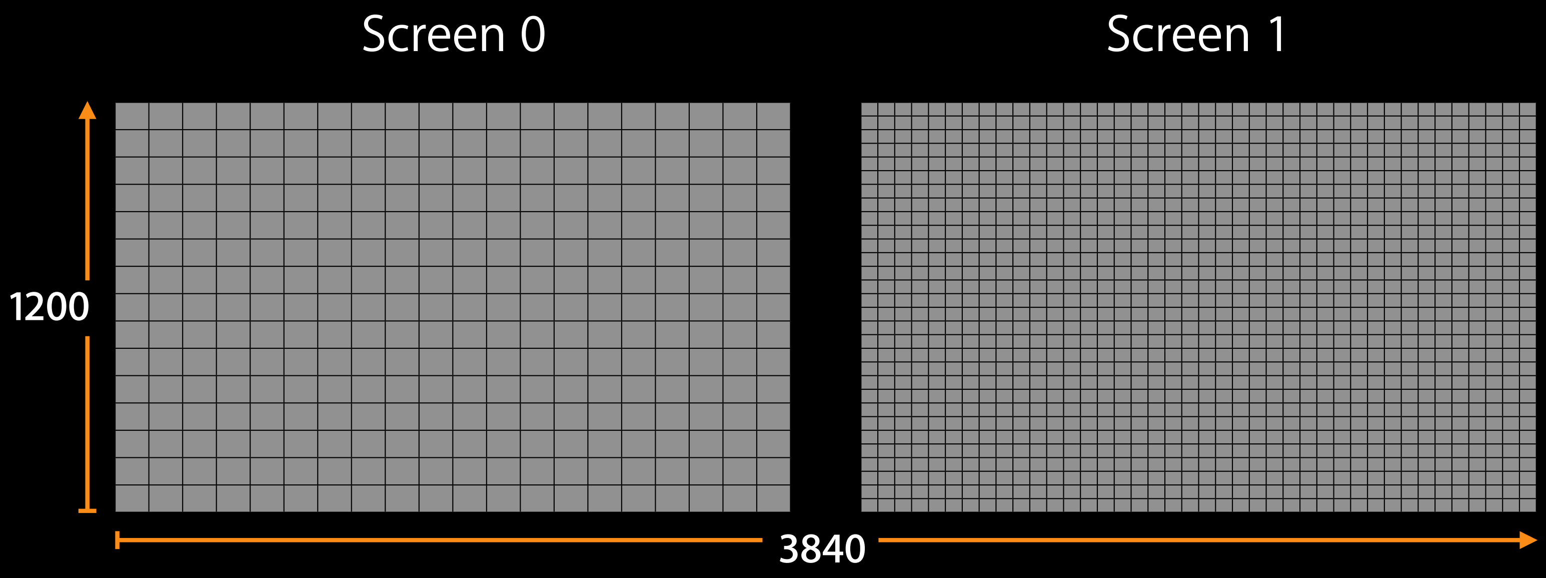
Quartz Screen Scaling

Device-independent drawing model extended to screens



Virtual Workspace Across Screens

All dimensions are in points, not pixels!



Screen 0

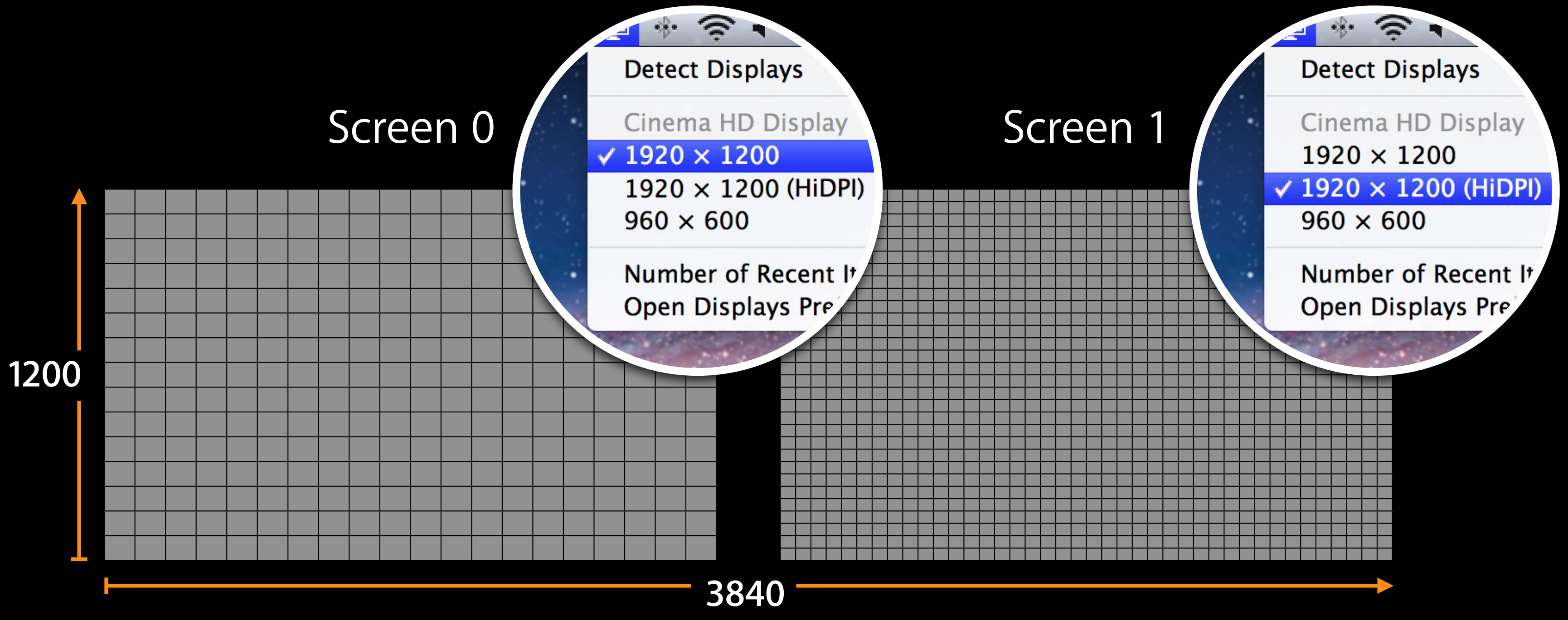
Screen 1

1200

3840

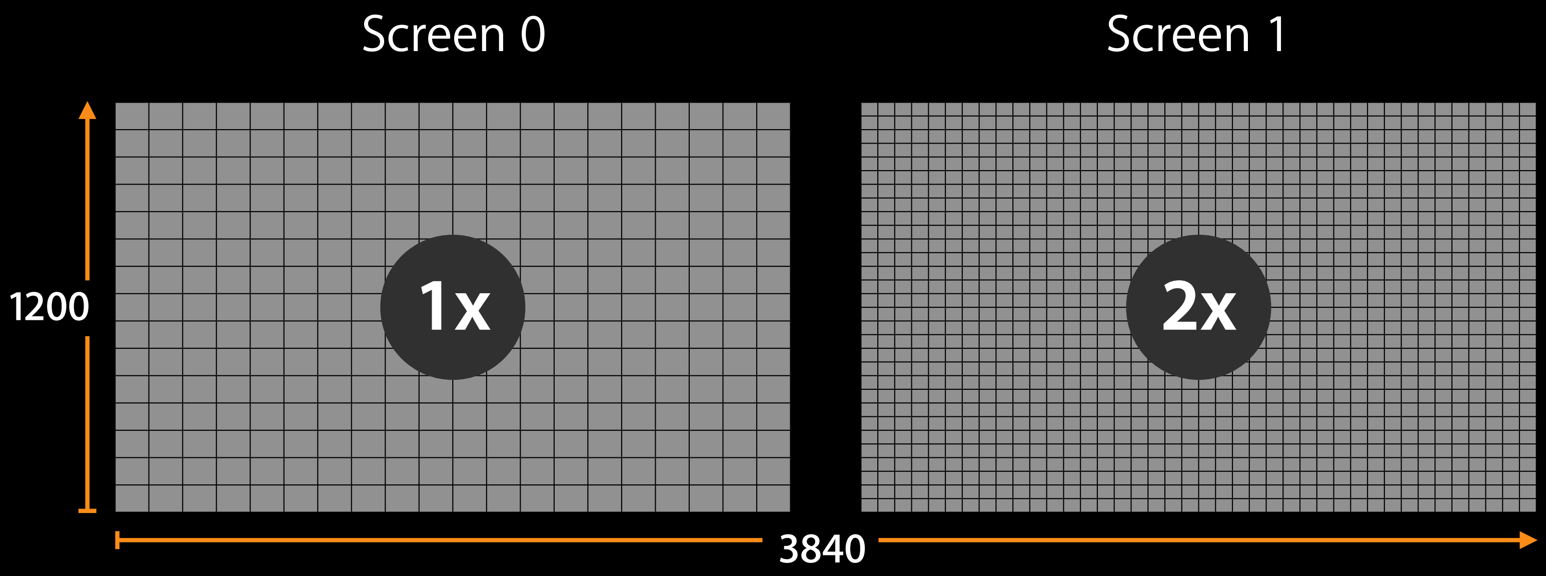
Quartz Screen Scaling

New High Resolution Quartz Display Modes



Integral Scaling per Screen

HiDPI modes configured with 2:1 pixels per point



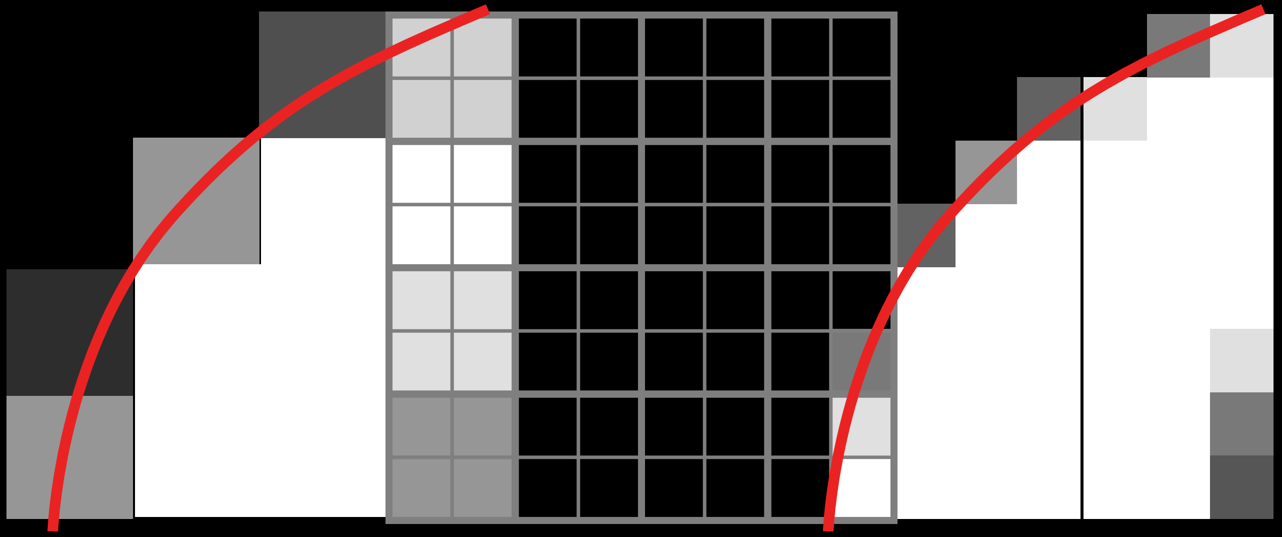
High Resolution Operation

User interface scaling

- Automatic for most Cocoa Applications

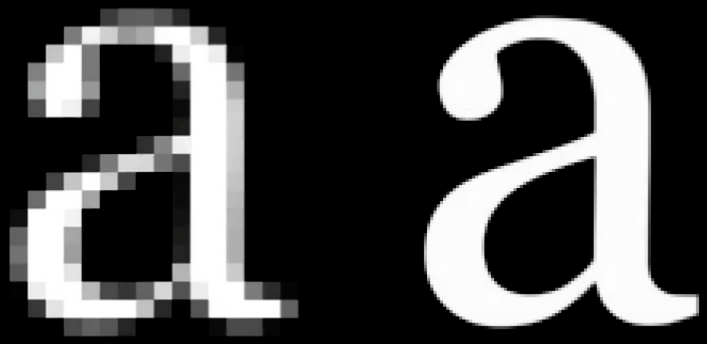
Windows Gain High Resolution Backing

Sharp Text and Quartz Path Drawing



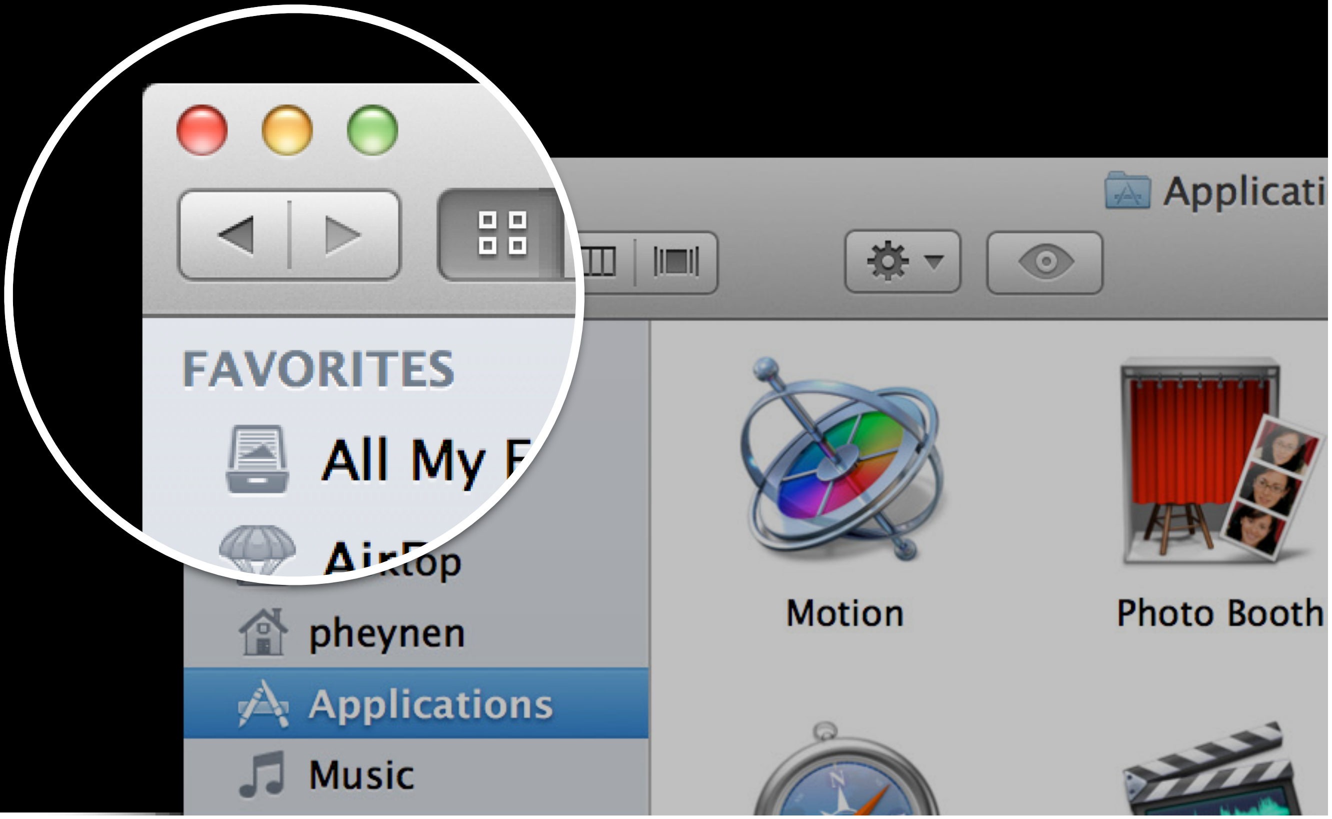
Windows Gain High Resolution Backing

Sharp Text and Quartz Path Drawing



New Aqua

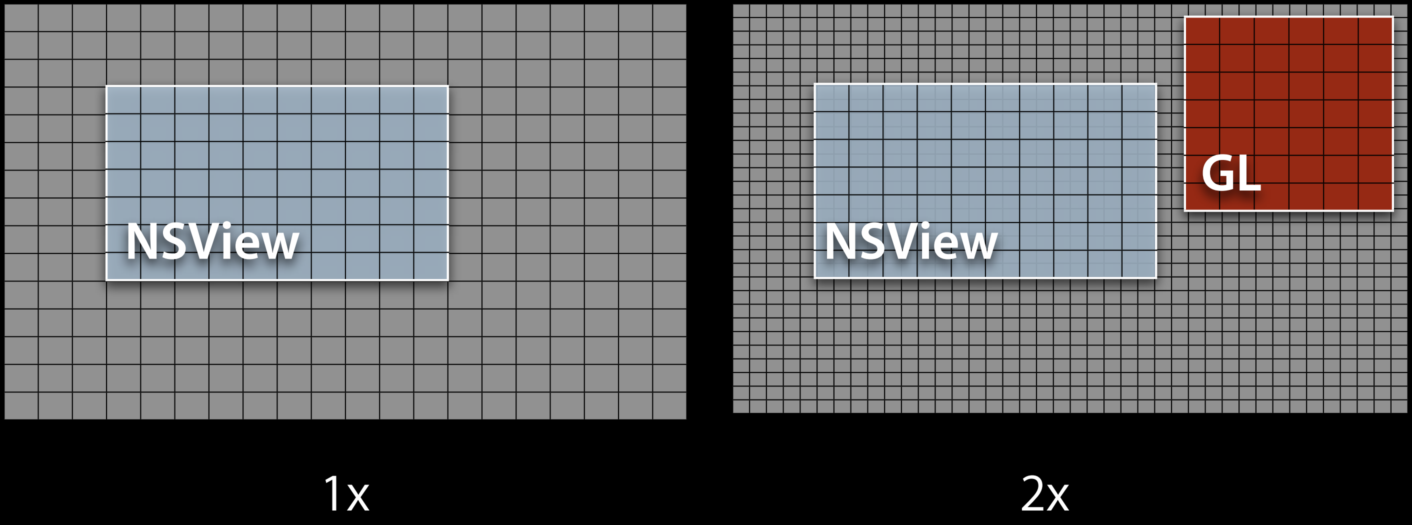
Automatically adapts to changes in resolution



High Resolution Operation

Application compatibility

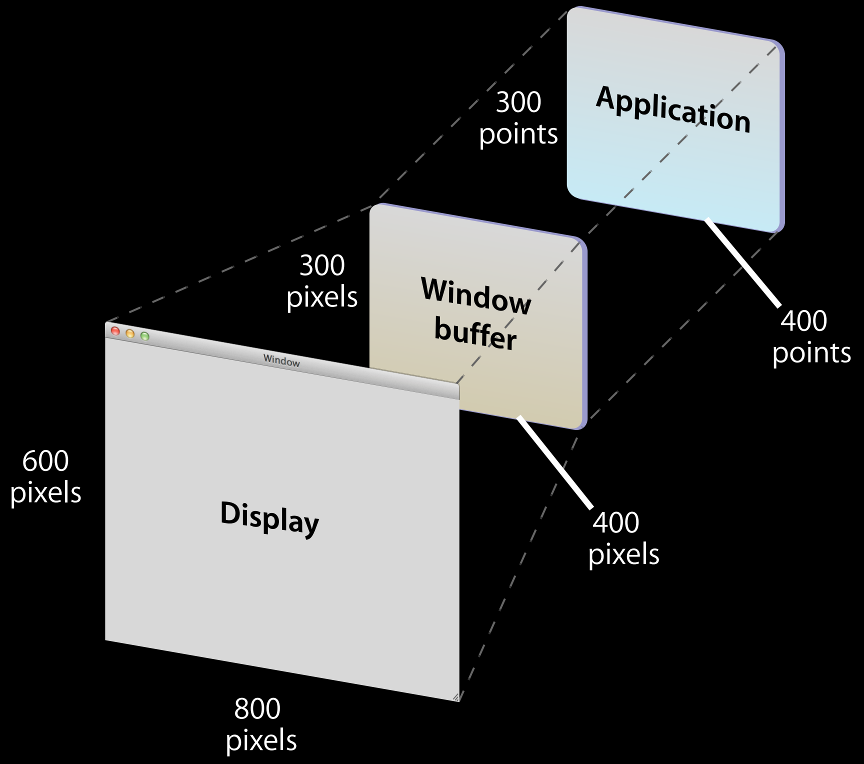
- Image resources are interpolated to user presentation size
- View frame coordinates and layout remain consistent with 1x operation
- OpenGL framebuffer remains consistent with 1x operation



Magnified Mode

Application compatibility

- Window backing remains low resolution for compatibility
- Window contents scaled to fit display
- Backstop for applications unable to provide a good user experience with default scaling



Optimizing for High Resolution

How to provide a stunning user experience

- Provide high resolution image resources

```
        foo.png, foo@2x.png  => foo.tiff
```

  - Use Xcode to combine image resources for deployment
- Use PDF template images for button icons
- Adapt OpenGL and CoreAnimation code for high resolution

Optimizing for High Resolution

OpenGL and CoreAnimation

- OpenGL and CoreAnimation are inherently pixel based technologies
- Applications need to request high resolution backing surface

```
-[NSView setWantsBestResolutionOpenGLSurface:YES]
```

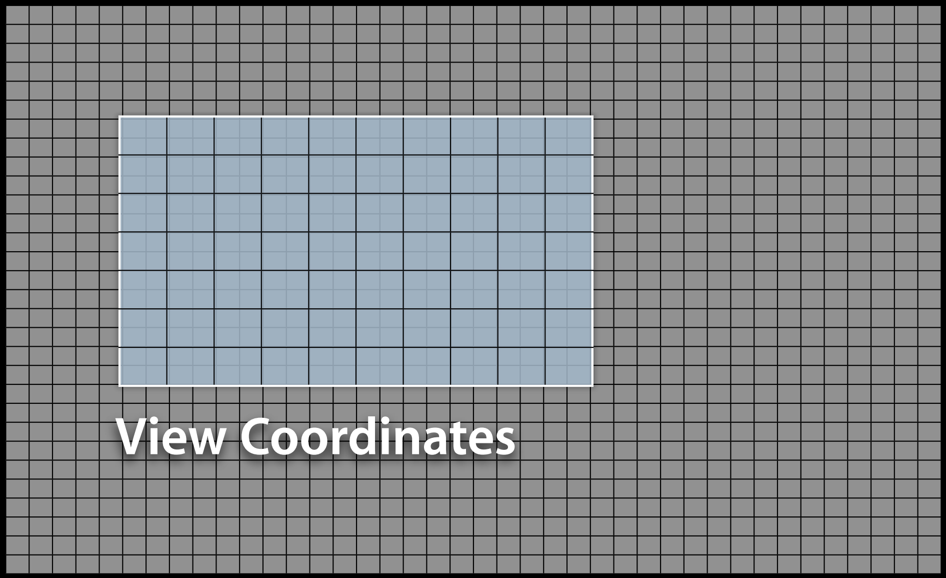
- Custom CALayers need explicit scale factor

```
myLayer.contentsScale=2
```

- Layer-backed NSViews are managed automatically

Optimizing for High Resolution

Coordinate space conversion



Window Backing Coordinates

Optimizing for High Resolution

Achieving pixel precision

Aligning drawing to device pixel boundaries	
`-centerScanRect:`	NSView
`-backingAlignedRect:options:`	NSView, NSWindow, NSScreen

Coordinate System Conversion	
`-convertRectToBacking:`	NSView, NSWindow, NSScreen
`-convertRectFromBacking:`	NSView, NSWindow, NSScreen

Optimizing for High Resolution

Realities of a virtual world

- Be prepared for fractional mouse locations and view coordinates
- Don’t make assumptions about pixels
  - NSView, NSWindow and NSScreen may all have independently varying backing resolutions
- NSImage size is in points, CGImage size is in pixels!
- Most of the time, calling -centerScanRect: is enough

Wrap Up

This concludes our regularly scheduled programming

More Information

Bill Dudney
Application Frameworks Evangelist
dudney@apple.com

Documentation
Application Kit Release Notes
Resolution Independence Guidelines
http://developer.apple.com/devcenter/mac

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

Related Sessions

Scrolling, Swiping, Dragging: Now with more animation!	Russian Hill Wednesday 3:15PM
View Based NSTableView Basic to Advanced	Nob Hill Thursday 10:15AM
Design Patterns to Simplify Mac Accessibility	Pacific Heights Thursday 3:15PM

Labs

Cocoa, Full Screen, and Aqua Lab	Application Frameworks Lab A Wednesday 2:00-6:00PM
Interface Builder Lab	Developer Tools Lab B Thursday 11:30AM-1:30PM
Cocoa, Auto Save, File Coordination, and Resume Lab	Application Frameworks Lab A Thursday 2:00-4:00PM

WWDC2011