

Introduction to High Resolution on OS X

Session 213

Dan Schimpf
Patrick Heynen

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

Introduction

Introduction

- 2x Retina displays arrived with iPhone 4

Introduction

- 2x Retina displays arrived with iPhone 4
- Macs have only one scale factor

Introduction

- 2x Retina displays arrived with iPhone 4
- Macs have only one scale factor
 - ...until now!

Introduction

- 2x Retina displays arrived with iPhone 4
- Macs have only one scale factor
 - ...until now!
- OS X takes care of a lot of the details

Introduction

- 2x Retina displays arrived with iPhone 4
- Macs have only one scale factor
 - ...until now!
- OS X takes care of a lot of the details
- Full adoption requires some work

Agenda

Agenda

- What is High Resolution for Mac?

Agenda

- What is High Resolution for Mac?
- Checklist for optimizing for High Resolution

Agenda

- What is High Resolution for Mac?
- Checklist for optimizing for High Resolution
- Bitmap images

Agenda

- What is High Resolution for Mac?
- Checklist for optimizing for High Resolution
- Bitmap images
- Icons

Agenda

- What is High Resolution for Mac?
- Checklist for optimizing for High Resolution
- Bitmap images
- Icons
- Final touches

Agenda

- What is High Resolution for Mac?
- Checklist for optimizing for High Resolution
- Bitmap images
- Icons
- Final touches
- Troubleshooting

Technology Overview

Technology Overview

- New high resolution display modes for Retina displays

Technology Overview

- New high resolution display modes for Retina displays
- Screens and windows have a 4:1 pixel per point density

Technology Overview

- New high resolution display modes for Retina displays
- Screens and windows have a 4:1 pixel per point density
- Frameworks provide automatic scaling between 1x and 2x operation

Technology Overview

- New high resolution display modes for Retina displays
- Screens and windows have a 4:1 pixel per point density
- Frameworks provide automatic scaling between 1x and 2x operation
- Quartz Window Manager ensures consistent presentation across multiple displays

What Is High Resolution?

What Is High Resolution?

- With High Resolution, the pixel density **quadruples**

What Is High Resolution?

- With High Resolution, the pixel density **quadruples**
 - Four pixels on screen for each point

What Is High Resolution?

- With High Resolution, the pixel density **quadruples**
 - Four pixels on screen for each point
- A “point” is a unit of measurement

What Is High Resolution?

- With High Resolution, the pixel density **quadruples**
 - Four pixels on screen for each point
- A “point” is a unit of measurement
- A point traditionally corresponds to $1/72$ of an inch (although this varies on actual monitors)

What Is High Resolution?

- With High Resolution, the pixel density **quadruples**
 - Four pixels on screen for each point
- A “point” is a unit of measurement
- A point traditionally corresponds to $1/72$ of an inch (although this varies on actual monitors)
- Sharper text and more detailed graphics

From a Point to a Pixel



1 point

nt	1 point	1 point	1 point	1
nt	1 point	1 point	1 point	1
nt	1 point	1 point	1 point	1

From a Point to a Pixel



1 point

From a Point to a Pixel



1 pixel

1x

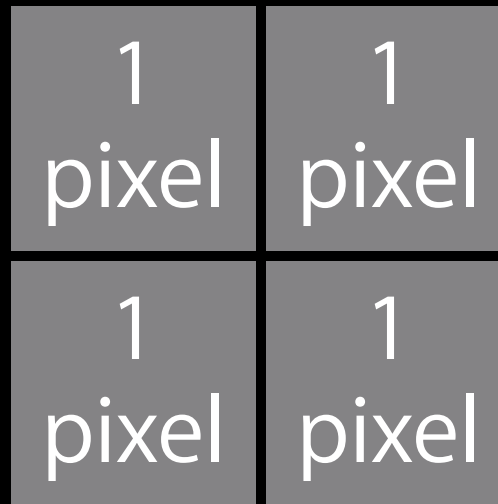
From a Point to a Pixel



1 point

2x

From a Point to a Pixel



2x

What Is High Resolution?

Comparison to iOS

What Is High Resolution?

Comparison to iOS

- Basic tenets of High Resolution are identical to iOS

What Is High Resolution?

Comparison to iOS

- Basic tenets of High Resolution are identical to iOS
 - Integral UI scale factor

What Is High Resolution?

Comparison to iOS

- Basic tenets of High Resolution are identical to iOS
 - Integral UI scale factor
 - Automatic application scaling by frameworks

What Is High Resolution?

Comparison to iOS

- Basic tenets of High Resolution are identical to iOS
 - Integral UI scale factor
 - Automatic application scaling by frameworks
- Enhanced functionality for OS X

What Is High Resolution?

Comparison to iOS

- Basic tenets of High Resolution are identical to iOS
 - Integral UI scale factor
 - Automatic application scaling by frameworks
- Enhanced functionality for OS X
 - Multiple displays

What Is High Resolution?

Comparison to iOS

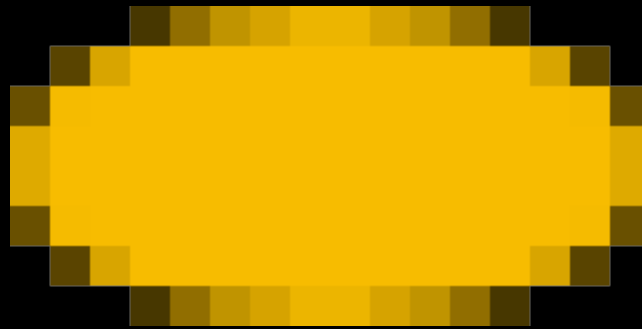
- Basic tenets of High Resolution are identical to iOS
 - Integral UI scale factor
 - Automatic application scaling by frameworks
- Enhanced functionality for OS X
 - Multiple displays
 - Windows vs. screens

What Is High Resolution?

Comparison to iOS

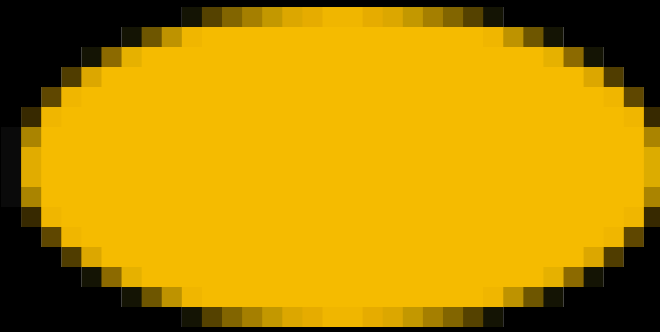
- Basic tenets of High Resolution are identical to iOS
 - Integral UI scale factor
 - Automatic application scaling by frameworks
- Enhanced functionality for OS X
 - Multiple displays
 - Windows vs. screens
 - Dynamic resolution changes

More Pixels Means More Detail



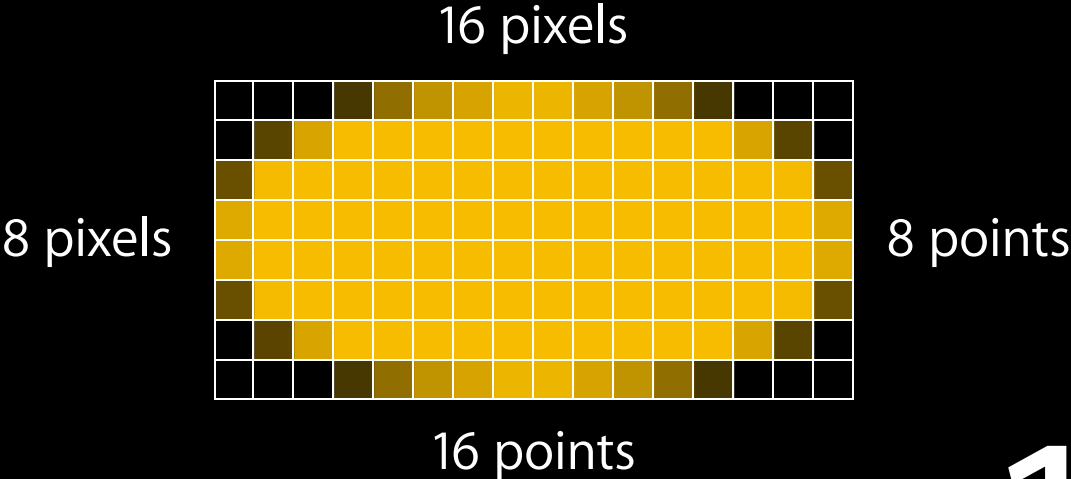
1x

More Pixels Means More Detail



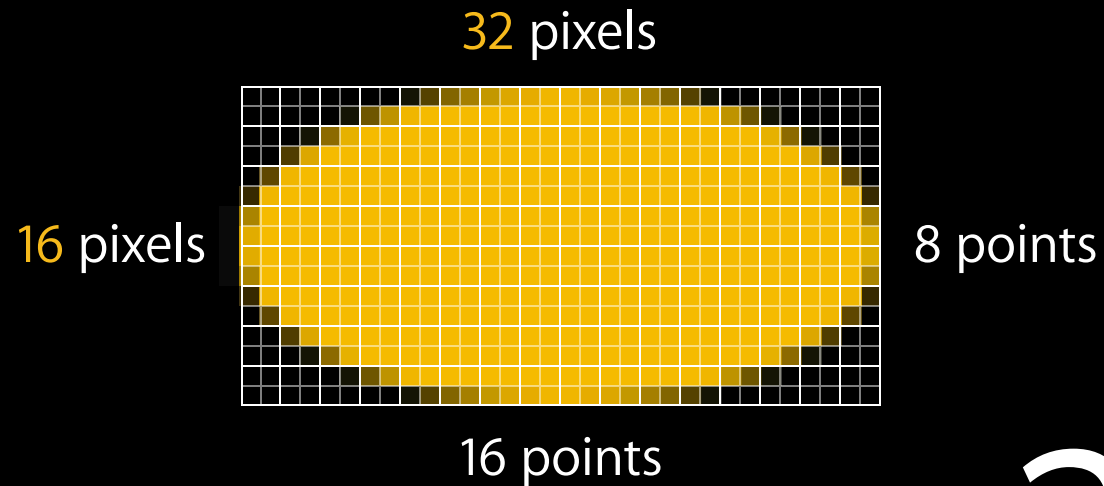
2x

More Pixels Means More Detail



1x

More Pixels Means More Detail



2x

Unified Coordinate System

Unified Coordinate System

- All coordinates are in points

Unified Coordinate System

- All coordinates are in points
 - View frames

Unified Coordinate System

- All coordinates are in points
 - View frames
 - Window locations

Unified Coordinate System

- All coordinates are in points
 - View frames
 - Window locations
 - Screen sizes

Unified Coordinate System

- All coordinates are in points
 - View frames
 - Window locations
 - Screen sizes
- High resolution scale factor can differ by screen and window

Unified Coordinate System

- All coordinates are in points
 - View frames
 - Window locations
 - Screen sizes
- High resolution scale factor can differ by screen and window
- For the most part, apps don't need to know or care about scale factor

Points vs. Pixels

A tale of two screens

1x

1920 × 1080 pixels
1920 × 1080 points

2x

3840 × 2160 pixels
1920 × 1080 points

Compatibility with Current Apps

Compatibility with Current Apps

- By default, Cocoa apps are scaled automatically

Compatibility with Current Apps

- By default, Cocoa apps are scaled automatically
 - Sharp text and Aqua UI widgets

Compatibility with Current Apps

- By default, Cocoa apps are scaled automatically
 - Sharp text and Aqua UI widgets
 - Custom bitmaps won't be optimized

Compatibility with Current Apps

- By default, Cocoa apps are scaled automatically
 - Sharp text and Aqua UI widgets
 - Custom bitmaps won't be optimized
- By default, Carbon apps are magnified

Compatibility with Current Apps

- By default, Cocoa apps are scaled automatically
 - Sharp text and Aqua UI widgets
 - Custom bitmaps won't be optimized
- By default, Carbon apps are magnified
 - Unoptimized text and UI widgets

Checklist for High Resolution

Checklist for High Resolution

✓ Add 2x bitmaps

Checklist for High Resolution

- ✓ Add 2x bitmaps
- ✓ Add high resolution icons for use in Finder

Checklist for High Resolution

- ✓ Add 2x bitmaps
- ✓ Add high resolution icons for use in Finder
- ✓ Replace uses of deprecated API

Checklist for High Resolution

- ✓ Add 2x bitmaps
- ✓ Add high resolution icons for use in Finder
- ✓ Replace uses of deprecated API
- ✓ Examine and replace bad assumptions about pixels

Checklist for High Resolution

- ✓ Add 2x bitmaps
- ✓ Add high resolution icons for use in Finder
- ✓ Replace uses of deprecated API
- ✓ Examine and replace bad assumptions about pixels
- ✓ Give it a try!

Testing High Resolution

Testing High Resolution

- Install Graphics Tools for Xcode

Testing High Resolution

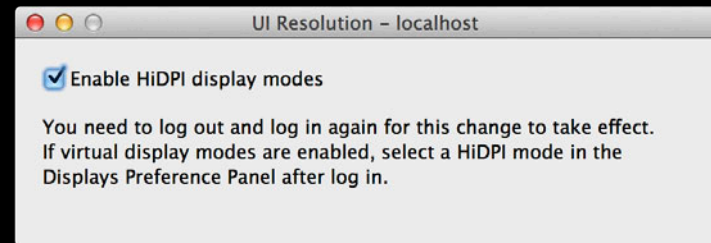
- Install Graphics Tools for Xcode
- Open Quartz Debug

Testing High Resolution

- Install Graphics Tools for Xcode
- Open Quartz Debug
- Open UI Resolution window

Testing High Resolution

- Install Graphics Tools for Xcode
- Open Quartz Debug
- Open UI Resolution window
- Select “Enable HiDPI display modes”



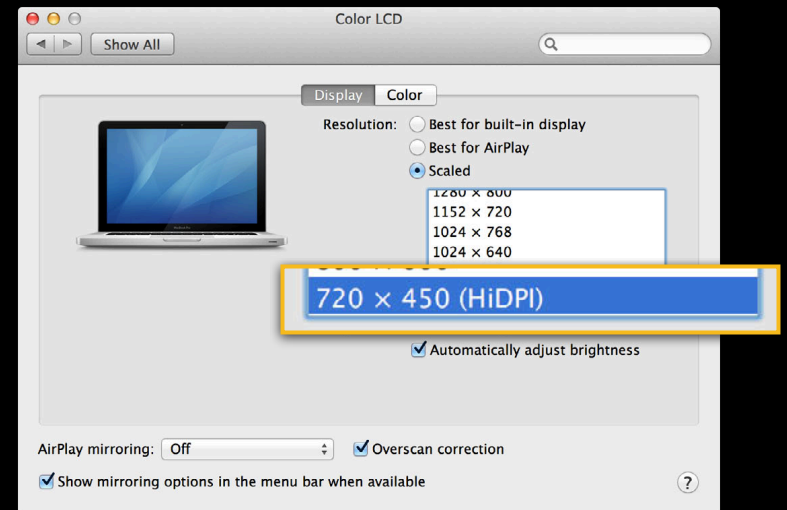
Testing High Resolution

Testing High Resolution

- Log out and back in

Testing High Resolution

- Log out and back in
- Use the Displays System Preferences pane to pick a “HiDPI” mode



High Resolution Artwork

Patrick Heynen

High Resolution Artwork

Optimizing for Retina Display

- It's all about pixels!



High Resolution Artwork

Optimizing for Retina Display



- It's all about pixels!
- Well-executed designs can have a huge impact

High Resolution Artwork

Case Study—Standard Resolution vs. High Resolution





Reminders



Completed

Reminders

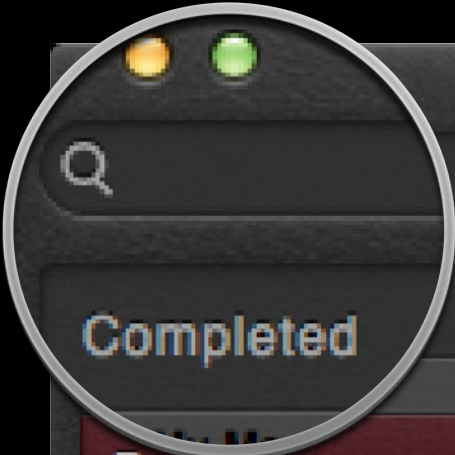
lcalserver.apple.com

tasks

Reminders



1 Completed



Reminders



Completed

Reminders

icalserver.apple.com

tasks

Reminders



1 Completed



High Resolution Artwork

What you need to do

High Resolution Artwork

What you need to do

- Graphics resources need to be created at twice the pixel density

High Resolution Artwork

What you need to do

- Graphics resources need to be created at twice the pixel density
- Just integrate new “@2x” image resources into your project

High Resolution Artwork

What you need to do

- Graphics resources need to be created at twice the pixel density
- Just integrate new “@2x” image resources into your project
- You’re done!

Artwork Production for High Resolution

Things to keep in mind

Artwork Production for High Resolution

Things to keep in mind

- Up-resing large quantities of graphics = challenging design task

Artwork Production for High Resolution

Things to keep in mind

- Up-resing large quantities of graphics = challenging design task
- Artwork tasks typically > 50% of overall effort

Artwork Production for High Resolution

Things to keep in mind

- Up-resing large quantities of graphics = challenging design task
- Artwork tasks typically > 50% of overall effort
- Good communication between designer and developer is essential

High Resolution Artwork

Image resource categories

Image Resource	File Type
Bitmap	PNG, TIFF, JPG
Vector	PDF
Application and Document Icons	ICNS

Bitmap Image Resources

High Resolution file naming

Bitmap Image Resources

High Resolution file naming

- Standard “@2x” file naming convention for image files



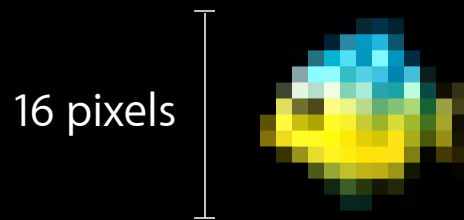
fish.png



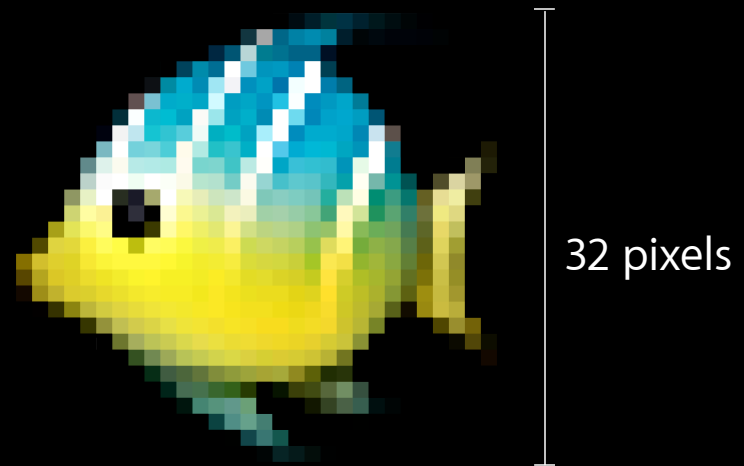
fish@2x.png

Bitmap Image Resources

Construction guidelines for High Resolution



fish.png

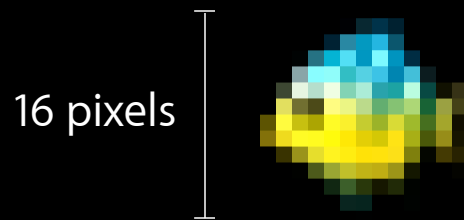


fish@2x.png

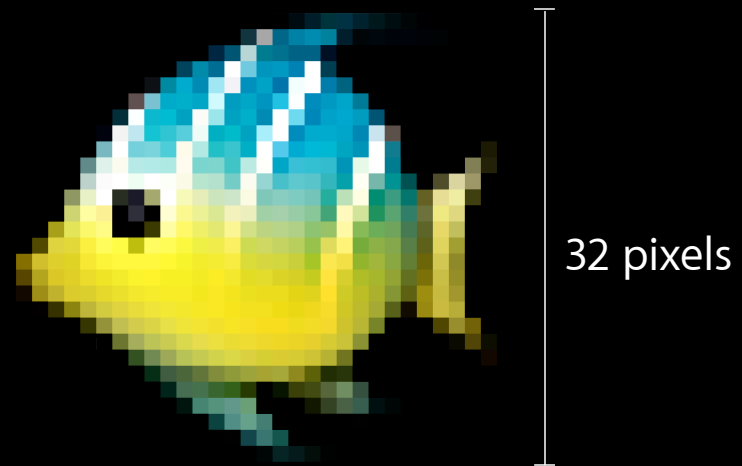
Bitmap Image Resources

Construction guidelines for High Resolution

- Images should have exactly twice the height and width in pixels of standard resolution counterparts



fish.png

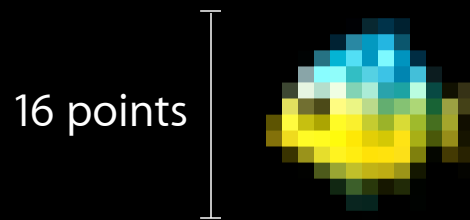


fish@2x.png

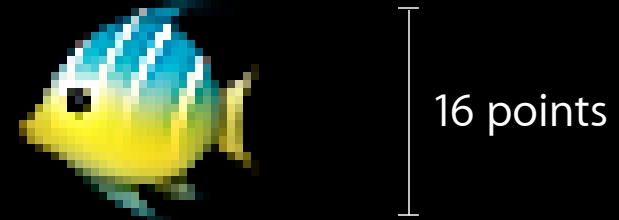
Bitmap Image Resources

Construction guidelines for High Resolution

- Images will be equivalent sizes in points on screen



fish.png



fish@2x.png

Bitmap Image Resources

High Resolution Runtime Support

- NSImage will automatically locate and use @2x image representations under High Resolution

```
[NSImage imageNamed:@"foo"];
```

Bitmap Image Resources

High Resolution Runtime Support

- UIImage will automatically locate and use @2x image representations under High Resolution

```
[UIImage imageNamed:@"foo"];
```

- UIImage also supports multi-resolution TIFF files
 - Use Xcode's "Combine High Resolution Images" project build setting

UIImage

Ready for Duty at any Resolution

NSImage

Ready for Duty at any Resolution

- 1x and 2x images or multiple-image TIFFs are loaded into a single NSImage instance

NSImage

Ready for Duty at any Resolution

- 1x and 2x images or multiple-image TIFFs are loaded into a single NSImage instance
- Use NSImage “by name” lookup methods
 - +`[NSImage imageNamed:]` or `-[NSBundle imageForResource:]`

NSImage

Ready for Duty at any Resolution

- 1x and 2x images or multiple-image TIFFs are loaded into a single NSImage instance
- Use NSImage “by name” lookup methods
 - `+[NSImage imageNamed:]` or `-[NSBundle imageForResource:]`
- Size of image is always in points, but pixel geometry may vary!

NSImage

Ready for Duty at any Resolution

- 1x and 2x images or multiple-image TIFFs are loaded into a single NSImage instance
- Use NSImage “by name” lookup methods
 - `+[NSImage imageNamed:]` or `-[NSBundle imageForResource:]`
- Size of image is always in points, but pixel geometry may vary!
- NSImage automatically chooses best representation to draw
 - 1x: 32 × 32 pixels, 32 × 32 points
 - 2x: 64 × 64 pixels, 32 × 32 points

NSImage

Ready for Duty at any Resolution

- 1x and 2x images or multiple-image TIFFs are loaded into a single NSImage instance
- Use NSImage “by name” lookup methods
 - `+[NSImage imageNamed:]` or `-[NSBundle imageForResource:]`
- Size of image is always in points, but pixel geometry may vary!
- NSImage automatically chooses best representation to draw
 - 1x: 32 × 32 pixels, 32 × 32 points
 - 2x: 64 × 64 pixels, 32 × 32 points
- Contrast: CGImage geometry always in pixels, limited to one resolution

Vector Image Resources

Optimizing for High Resolution

- PDF resources scale automatically to any user size or pixel density

Vector Image Resources

Optimizing for High Resolution

- PDF resources scale automatically to any user size or pixel density
- Recommended for simple graphical elements such as button images



PDF

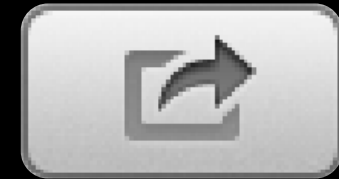
Vector Image Resources

Optimizing for High Resolution

- PDF resources scale automatically to any user size or pixel density
- Recommended for simple graphical elements such as button images
- Best when combined with AppKit template image rendering



PDF



Template Images

More bang for your artwork buck

- Images treated as shapes

Template Images

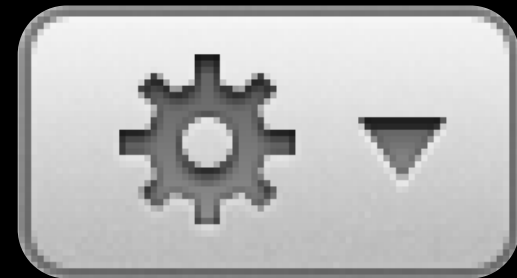
More bang for your artwork buck

- Images treated as shapes
- Loaded from file ending in "Template" or marked with `[image setTemplate:YES]`

Template Images

More bang for your artwork buck

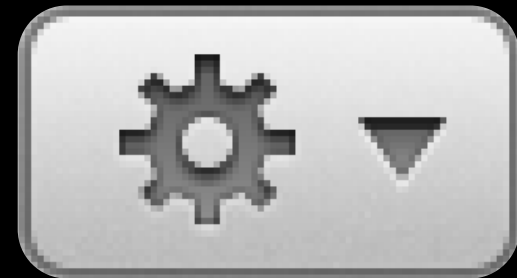
- Images treated as shapes
- Loaded from file ending in "Template" or marked with `[image setTemplate:YES]`
- Transformed by AppKit with context-specific appearance



Template Images

More bang for your artwork buck

- Images treated as shapes
- Loaded from file ending in "Template" or marked with `[image setTemplate:YES]`
- Transformed by AppKit with context-specific appearance
- Effects rendering is done at full backing resolution



Application and Document Icons

Optimizing for High Resolution

Application and Document Icons

Optimizing for High Resolution

- Loaded from ICNS files



Application and Document Icons

Optimizing for High Resolution

- Loaded from ICNS files
- Used by Finder, Open/Save Panel, Spotlight Menu...

Application Icons

Multiple user sizes

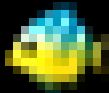
- ICNS provides five separate user size slots for providing artwork

Application Icons

Multiple user sizes

- ICNS provides five separate user size slots for providing artwork
- By default, the next highest slot is used to draw for a given size
 - For example, the 256 slot is used to draw into 128×128 points @2x

16×16



32×32



128×128

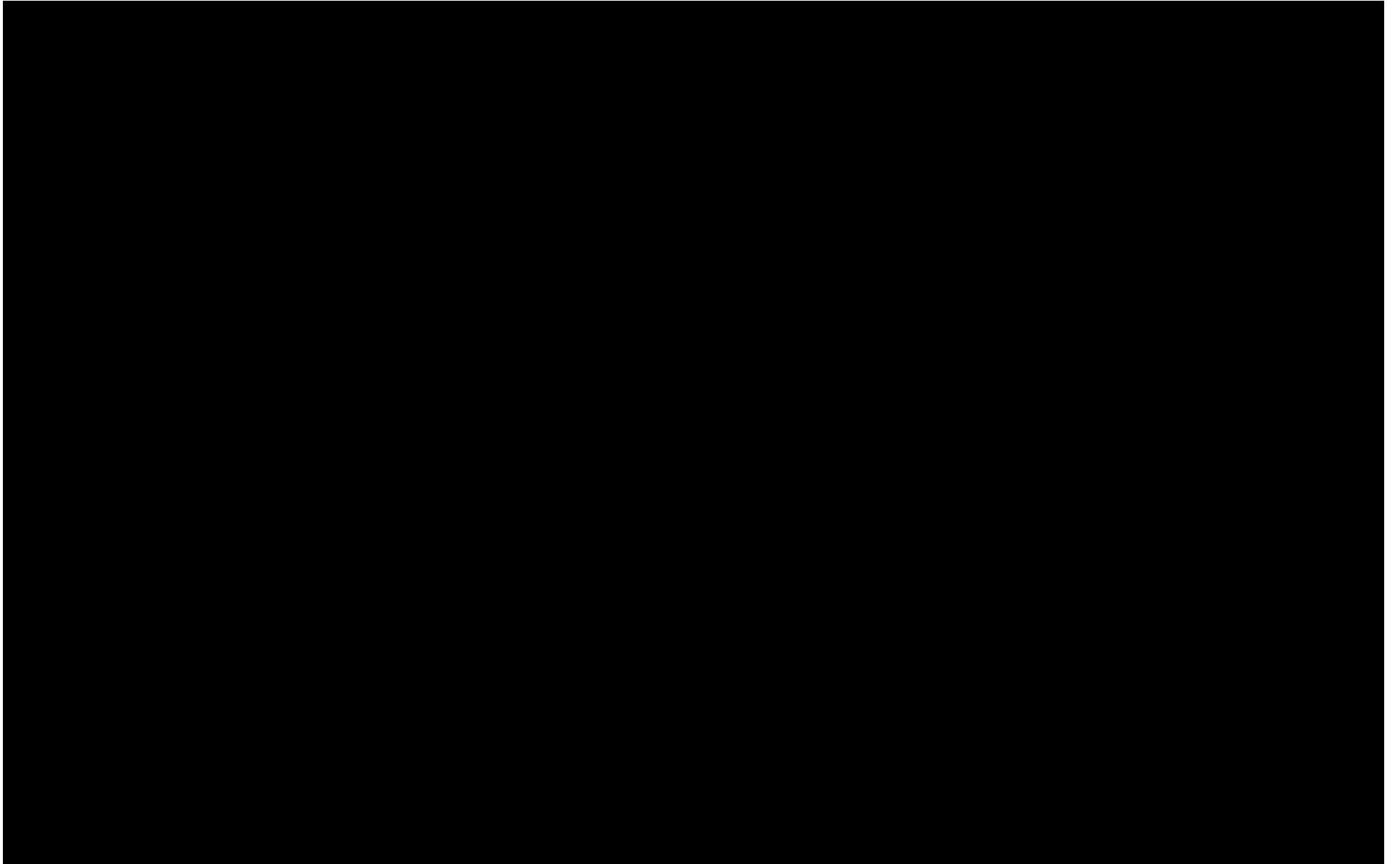


256×256



512×512



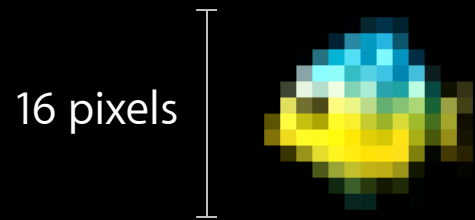


What's the Problem?

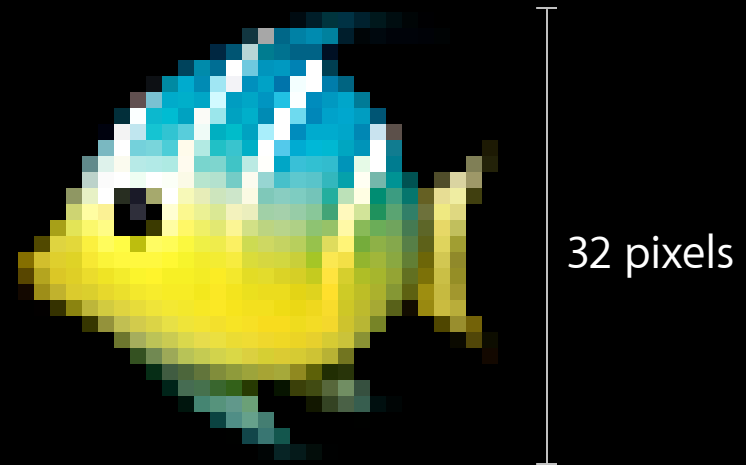
What's the Problem?

Retina displays are different

16x16 Icons Shown Pixel-for-Pixel

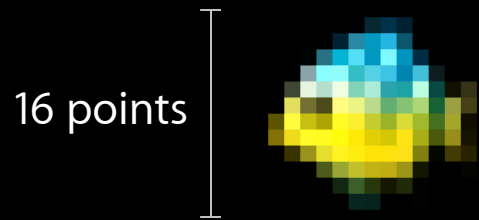


16x16 @1x

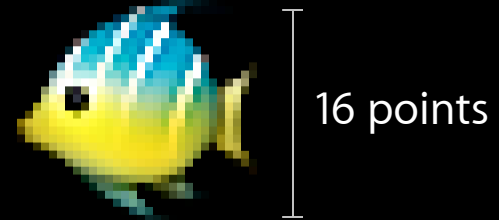


16x16 @2x

16x16 Icons Shown Point-for-Point

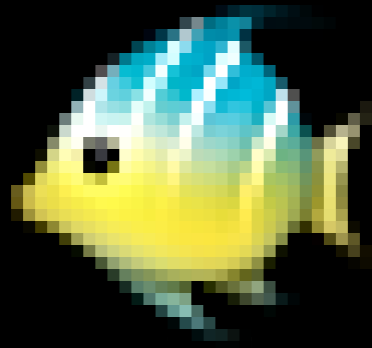


16x16 @1x



16x16 @2x

Icons with the same pixel count but different target resolutions may need different visual treatment because of the display size

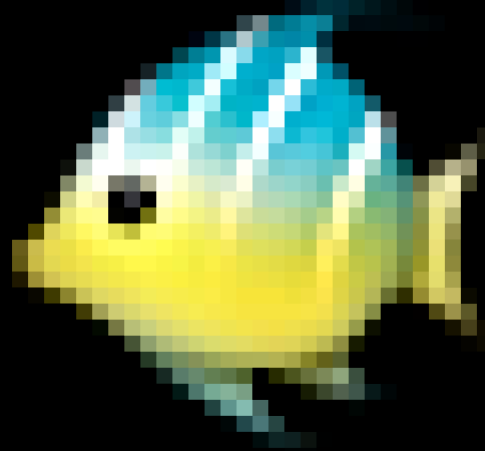


32x32 @1x



16x16 @2x

Icons with the same pixel count but different target resolutions may need different visual treatment because of the display size



32x32 @1x



16x16 @2x

A standard resolution icon at a larger size will not always serve as a high resolution icon at a smaller size because the image content may be different



32x32 @2x



128x128 @1x

Table of Icon Images (not to scale)

	16x16	32x32	128x128	256x256	512x512
@1x					
@2x					

Table of Icon Images (pixel-for-pixel)

16x16

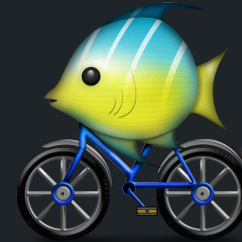
32x32

128x128

256x256

512x512

@1x



@2x

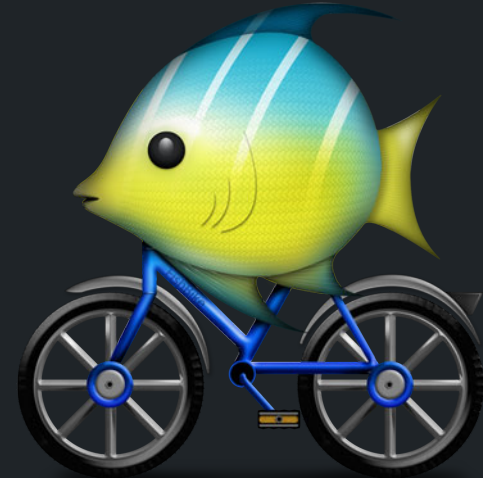
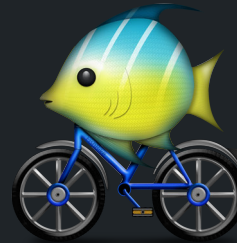












Table of Icon Images (point-for-point)

	16x16	32x32	128x128	256x256	512x512
@1x					
@2x					

Application and Document Icons

ICNS enhancements for High Resolution



- ICNS format has been extended to support @2x variants for every user size slot

Application and Document Icons

ICNS enhancements for High Resolution



- ICNS format has been extended to support @2x variants for every user size slot
- Under High Resolution, @2x variants are prioritized when drawn to screen

Application and Document Icons

ICNS enhancements for High Resolution



- ICNS format has been extended to support @2x variants for every user size slot
- Under High Resolution, @2x variants are prioritized when drawn to screen
- 1024×1024 is now 512×512@2x

Application and Document Icons

ICNS enhancements for High Resolution



- ICNS format has been extended to support @2x variants for every user size slot
- Under High Resolution, @2x variants are prioritized when drawn to screen
- 1024×1024 is now 512×512@2x
- Not all slots need to be populated, but having @2x counterparts is recommended

Application and Document Icons

Creating High Resolution ICNS files

- Existing Icon Composer workflow—one image well at a time



Application and Document Icons

Creating High Resolution ICNS files

- Existing Icon Composer workflow—one image well at a time
- Now deprecated!



Introducing Icon Sets

Icon art delivery format for High Resolution



Introducing Icon Sets

Icon art delivery format for High Resolution



- Folder and file naming convention



FishBike.iconset

Icon Sets

New icon art delivery format



- Keep separate PNG images for each ICNS representation
- Name files by the ICNS size slot
 - icon_<UserSize>x<UserSize> [@2x].png
 - icon_16x16.png
 - icon_32x32.png
 - icon_16x16@2x.png
 - icon_32x32@2x.png
 - ...

Icon Sets

Features and benefits



- Better fit with existing design workflows
- More reliable color management
- Automatic transformation and validation by Xcode at build time
- QuickLook Plugin for verification
- iconutil command line tool
- Icon Composer “Export to Iconset”

Demo

Adding high resolution artwork and application icons

Final Touches

Dan Schimpf

Displays Everywhere!

- Display scale is dynamic. The user may...

Displays Everywhere!

- Display scale is dynamic. The user may...
 - Change it at any time

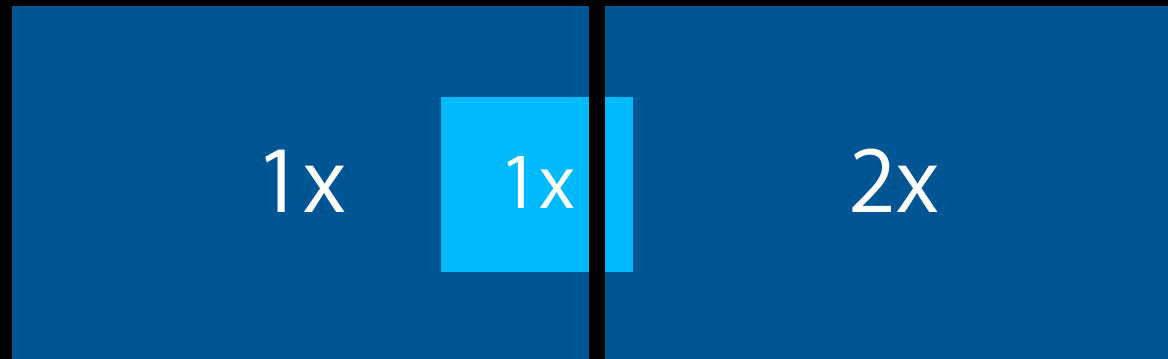
Displays Everywhere!

- Display scale is dynamic. The user may...
 - Change it at any time
 - Attach a display or projector of a different scale

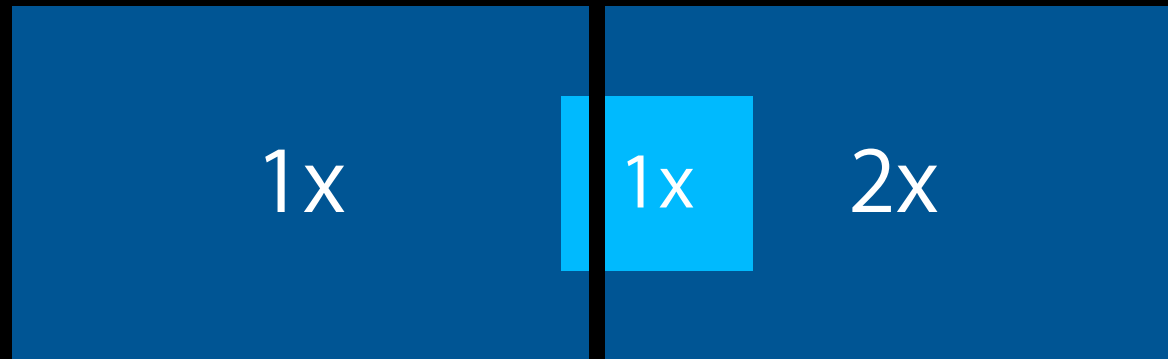
Displays Everywhere!

- Display scale is dynamic. The user may...
 - Change it at any time
 - Attach a display or projector of a different scale
 - Drag the window to a different display at any time

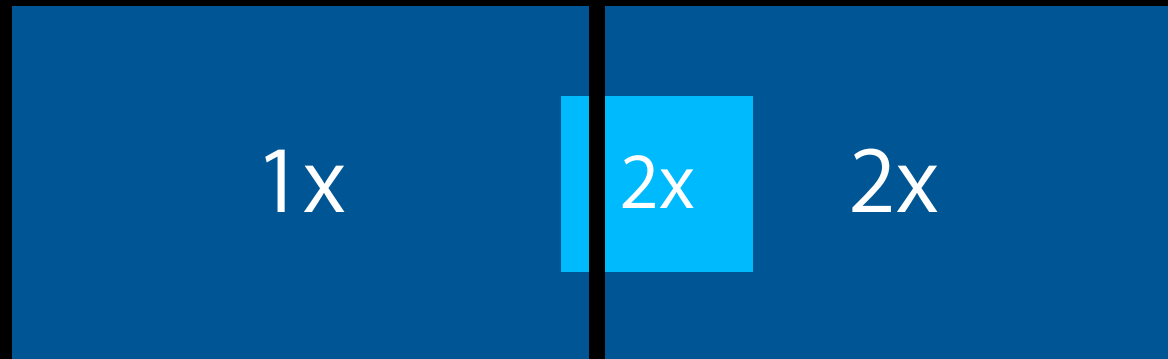
Multiple Displays



Multiple Displays



Multiple Displays



Reacting to Display Changes

Reacting to Display Changes

- NSWindow redraws when it switches scale factor

Reacting to Display Changes

- NSWindow redraws when it switches scale factor
- Don't cache drawing at a certain scale factor

Reacting to Display Changes

- NSWindow redraws when it switches scale factor
- Don't cache drawing at a certain scale factor
 - If you do, invalidate the cache when appropriate

Deprecated API

Deprecated API

- `convertRectToBase:` (and friends)
 - Use `convertRectToLayer:` for dealing with layer backing
 - Use `convertRect:toView:` with nil view to get window coordinates

Deprecated API

- `convertRectToBase:` (and friends)
 - Use `convertRectToLayer:` for dealing with layer backing
 - Use `convertRect:toView:` with nil view to get window coordinates
- `-[NSImage compositeToPoint:]`
 - Use methods beginning with “draw”

Deprecated API

- `convertRectToBase:` (and friends)
 - Use `convertRectToLayer:` for dealing with layer backing
 - Use `convertRect:toView:` with nil view to get window coordinates
- `-[NSImage compositeToPoint:]`
 - Use methods beginning with “draw”
- `-[NSScreen userSpaceScaleFactor]`
 - Tied to pre-10.7 model of high resolution interface

Deprecated API

- `convertRectToBase:` (and friends)
 - Use `convertRectToLayer:` for dealing with layer backing
 - Use `convertRect:toView:` with nil view to get window coordinates
- `-[NSImage compositeToPoint:]`
 - Use methods beginning with “draw”
- `-[NSScreen userSpaceScaleFactor]`
 - Tied to pre-10.7 model of high resolution interface
- QuickDraw
 - Also related technologies, such as `NSMovieView`

Bitmap Drawing

Bitmap Drawing

- Preferred—draw into `NSBitmapImageRep` created with graphics context

Bitmap Drawing

- Preferred—draw into NSBitmapImageRep created with graphics context
 - Automatically scaled

Bitmap Drawing

- Preferred—draw into `NSBitmapImageRep` created with graphics context
 - Automatically scaled
- `CGBitmapContext`, like `CGImage`, is measured in pixels

Bitmap Drawing

- Preferred—draw into NSBitmapImageRep created with graphics context
 - Automatically scaled
- CGContext, like CGImage, is measured in pixels
 - Scale the context by the scale of the final drawing destination

Bitmap Drawing

- Preferred—draw into NSBitmapImageRep created with graphics context
 - Automatically scaled
- CGContext, like CGImage, is measured in pixels
 - Scale the context by the scale of the final drawing destination
- Re-draw the bitmap if the destination scale factor changes

Troubleshooting

Troubleshooting

Troubleshooting

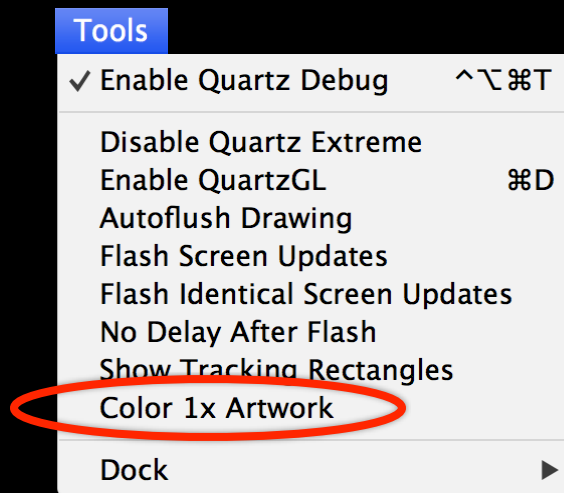
- My 2x images aren't drawing!

Troubleshooting

- My 2x images aren't drawing!
 - Look at the build product to make sure the 2x images are there, either on its own or in a TIFF

Troubleshooting

- My 2x images aren't drawing!
 - Look at the build product to make sure the 2x images are there, either on its own or in a TIFF
 - Use Quartz Debug to enable scaled image highlighting



Troubleshooting

Troubleshooting

- My image is drawing 4x!

Troubleshooting

- My image is drawing 4x!
 - Solution—make sure you aren't scaling it twice when drawing it; The more you can ignore 2x and let AppKit handle it, the better

Troubleshooting

- My image is drawing 4x!
 - Solution—make sure you aren't scaling it twice when drawing it; The more you can ignore 2x and let AppKit handle it, the better
 - Make sure you are drawing the image via modern API; `compositeToPoint:` can lead to bad behavior in certain cases

Troubleshooting

Troubleshooting

- My views aren't aligned!

Troubleshooting

- My views aren't aligned!

Solution—use `-[NSView centerScanRect:]` to align the view frame on a good grid boundary

Troubleshooting

Troubleshooting

- My custom drawing looks too high or low at 2x!

Troubleshooting

- My custom drawing looks too high or low at 2x!
 - Solution—look for rounding differences between 1x and 2x due to pixel sizes doubled at 2x (and therefore always even amounts of pixels)

Troubleshooting

Troubleshooting

- My windows are in the wrong place!

Troubleshooting

- My windows are in the wrong place!
 - Solution—make sure you're not scaling window coordinates anymore; Window coordinates are in points

Troubleshooting

Troubleshooting

- My custom CoreAnimation layer isn't sharp!

Troubleshooting

- My custom CoreAnimation layer isn't sharp!
 - Solution—set the `contentsScale` on layers that you manage yourself outside of an `NSView`

Troubleshooting

- My custom CoreAnimation layer isn't sharp!
 - Solution—set the `contentsScale` on layers that you manage yourself outside of an `NSView`
 - Can also use a new delegate method to manage it easily

Troubleshooting

Troubleshooting

- My OpenGL content isn't sharp!

Troubleshooting

- My OpenGL content isn't sharp!
 - Solution—opt into sharp OpenGL drawing

Demo

Troubleshooting common issues

More Information

Jake Behrens

UI Frameworks Evangelist

behrens@apple.com

Documentation

High Resolution Guidelines for OS X

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

Apple Developer Forums

<http://devforums.apple.com>

More Sessions

Advanced Tips and Tricks for High Resolution on OS X

Mission
Friday 10:15AM

Layer-Backed Views: AppKit + Core Animation

Nob Hill
Wednesday 10:15AM

Delivering Web Content on High Resolution Displays

Nob Hill
Wednesday 11:30AM

Advances in OpenGL and OpenGL ES

Pacific Heights
Wednesday 2:00PM

Labs

High Resolution on OS X Lab

Essentials Lab B
Wednesday 11:30AM

Web Content Optimization Lab

Safari & Web Lab
Wednesday 3:15PM

OpenGL Lab

Graphics, Media & Games Lab
Thursday 9:00AM

Cocoa and XPC Lab

Essentials Lab A
Friday 10:15AM

Summary

Summary

- New High Resolution display modes

Summary

- New High Resolution display modes
- Four pixels on screen for each point

Summary

- New High Resolution display modes
- Four pixels on screen for each point
- Add 2x bitmaps and icons

Summary

- New High Resolution display modes
- Four pixels on screen for each point
- Add 2x bitmaps and icons
- Clean up code for 2x

Summary

- New High Resolution display modes
- Four pixels on screen for each point
- Add 2x bitmaps and icons
- Clean up code for 2x
- Handle display changes in your app

Summary

- New High Resolution display modes
- Four pixels on screen for each point
- Add 2x bitmaps and icons
- Clean up code for 2x
- Handle display changes in your app
- Read the documentation

 WWDC2012

