

Improving Existing Apps

Using modern best practices

Session 213

Woody L., 🐙 to the Knowledge

Agenda

Reduce Technical Debt

Asset Catalogs

Dependency Injection

Live Playgrounds

Cycle of Development

You down with 'Dub-DC? Yeah, you know me.

Lots of Requests

Your boss



More Requests

Your customers



Technical Debt

```
//TODO: Write and clean up
```



Customer's Perspective



Actuality



ore **New API** zsh AppKit CF AirPortUtility PreferencesAp

iCal **Foundation** AVFoundation AirPortAssistant AirPortSetting

AirPortAssistant **OpenGL** GameKit Dock **Mail** MapKit

cPlayer xnu **AppKit** AppStore MobileSafari zsh QuickTime

udf **WebKit** BlueToothSettings cups **Messages** Dock

yMonitor **MobileSafari** bash Mail AccessibilitySettings GameKit

GameKitServices **MediaPlayerUI** MediaPlayer MediaStream MobileMail

Swift 3

Source code compatibility



New and Updated Platforms

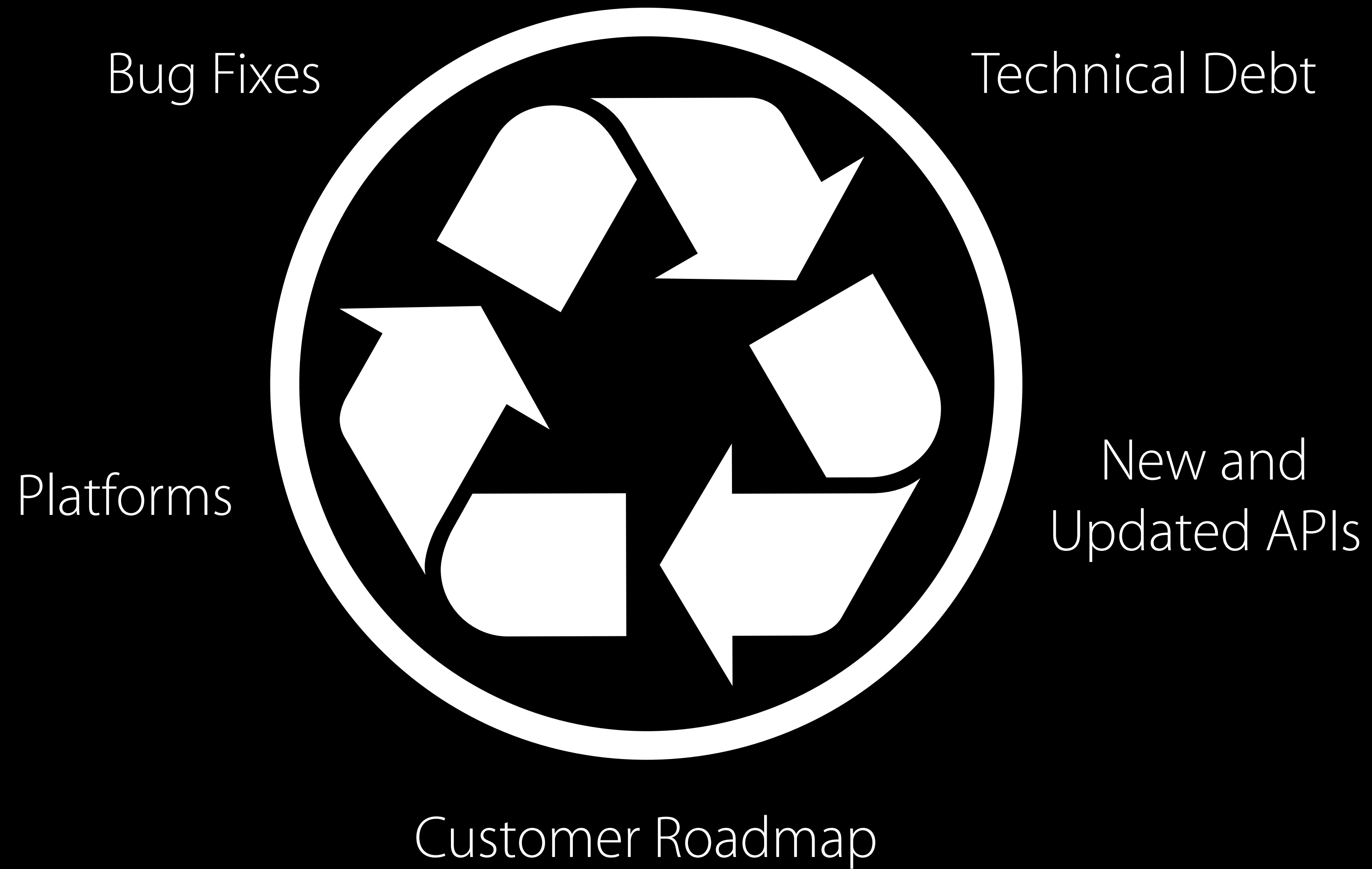
macOS

iOS

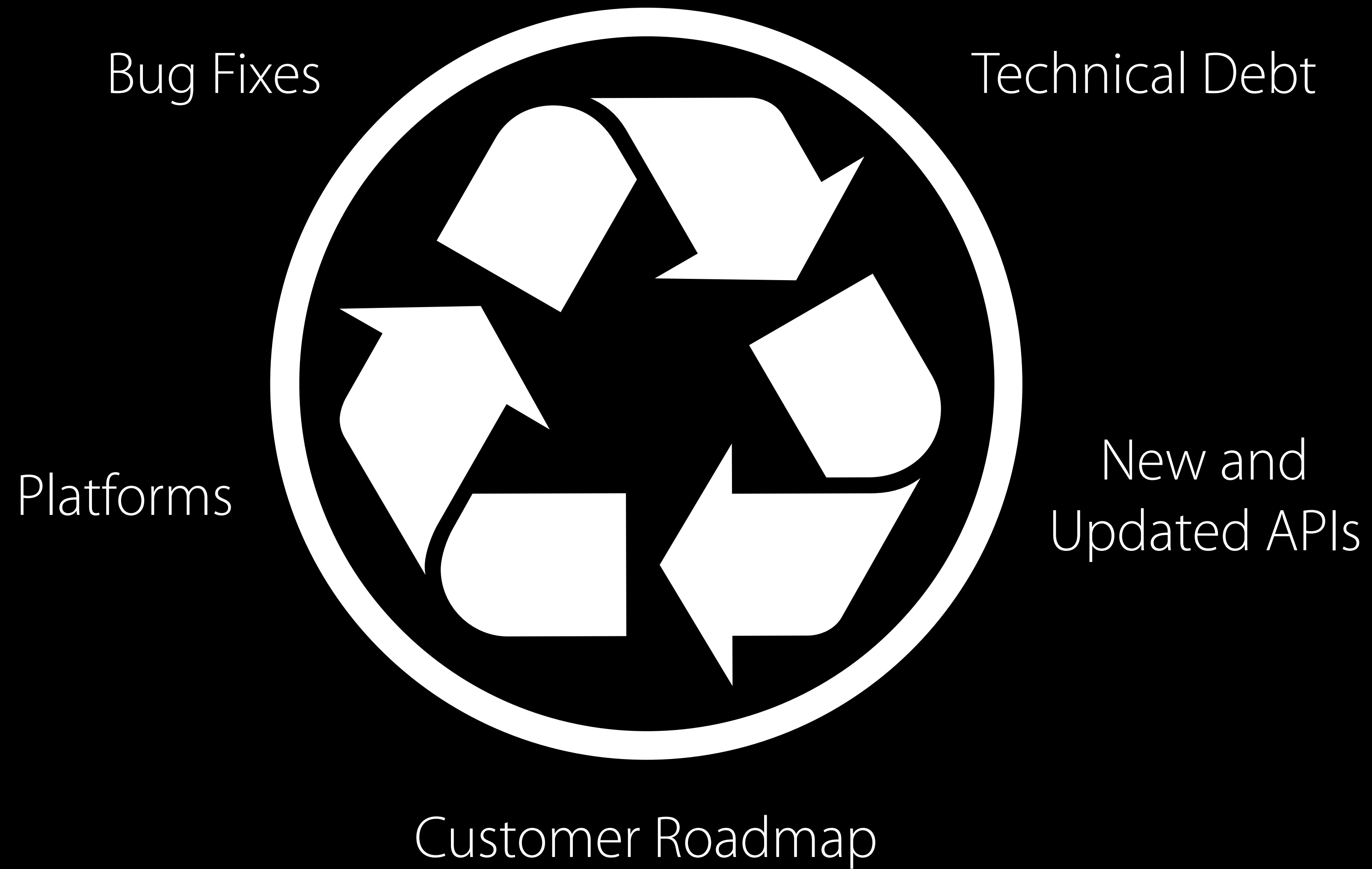
tvOS

watchOS

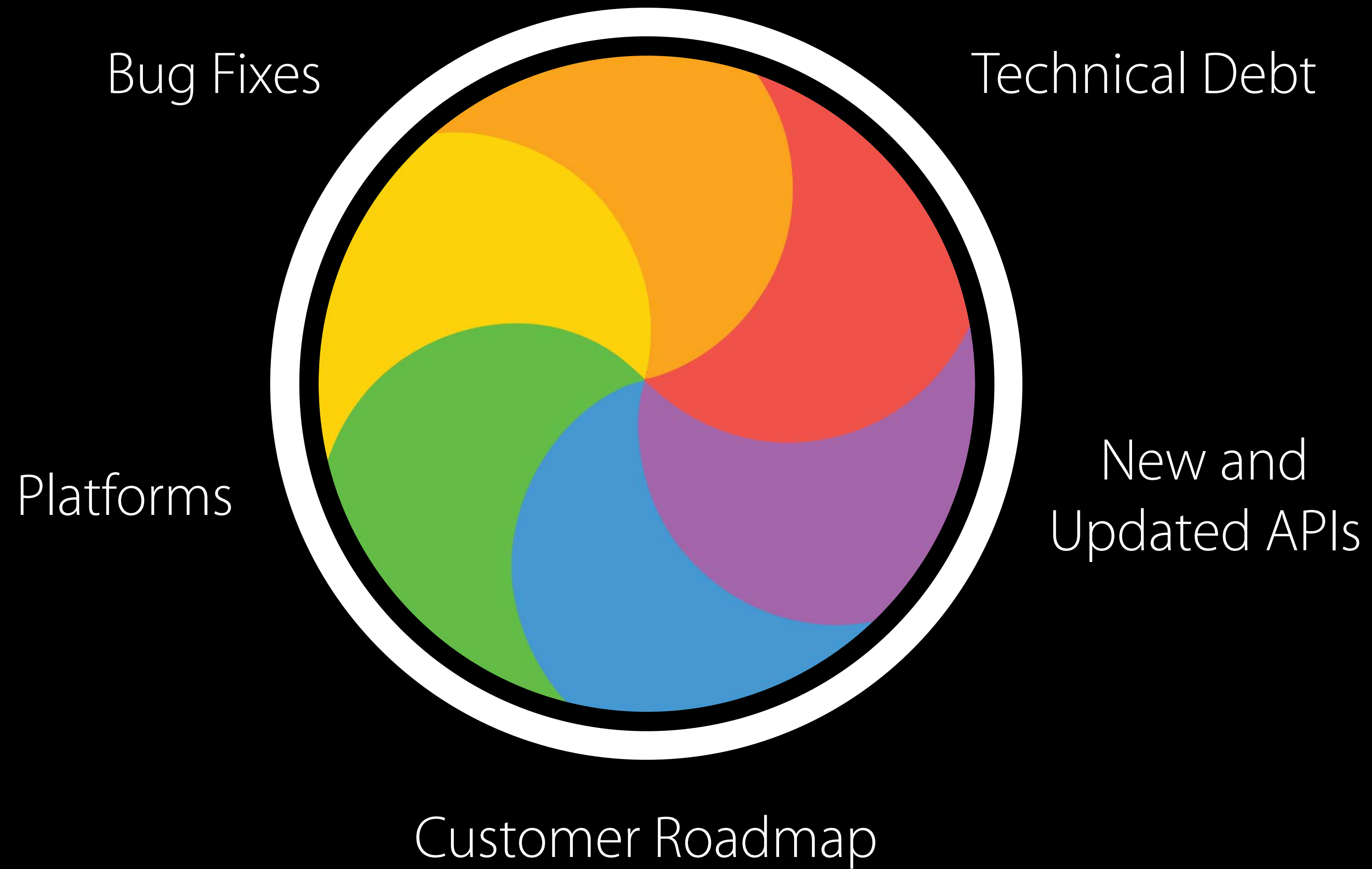
A Dev's Run Loop



A Dev's Run Loop



A Dev's Run Loop

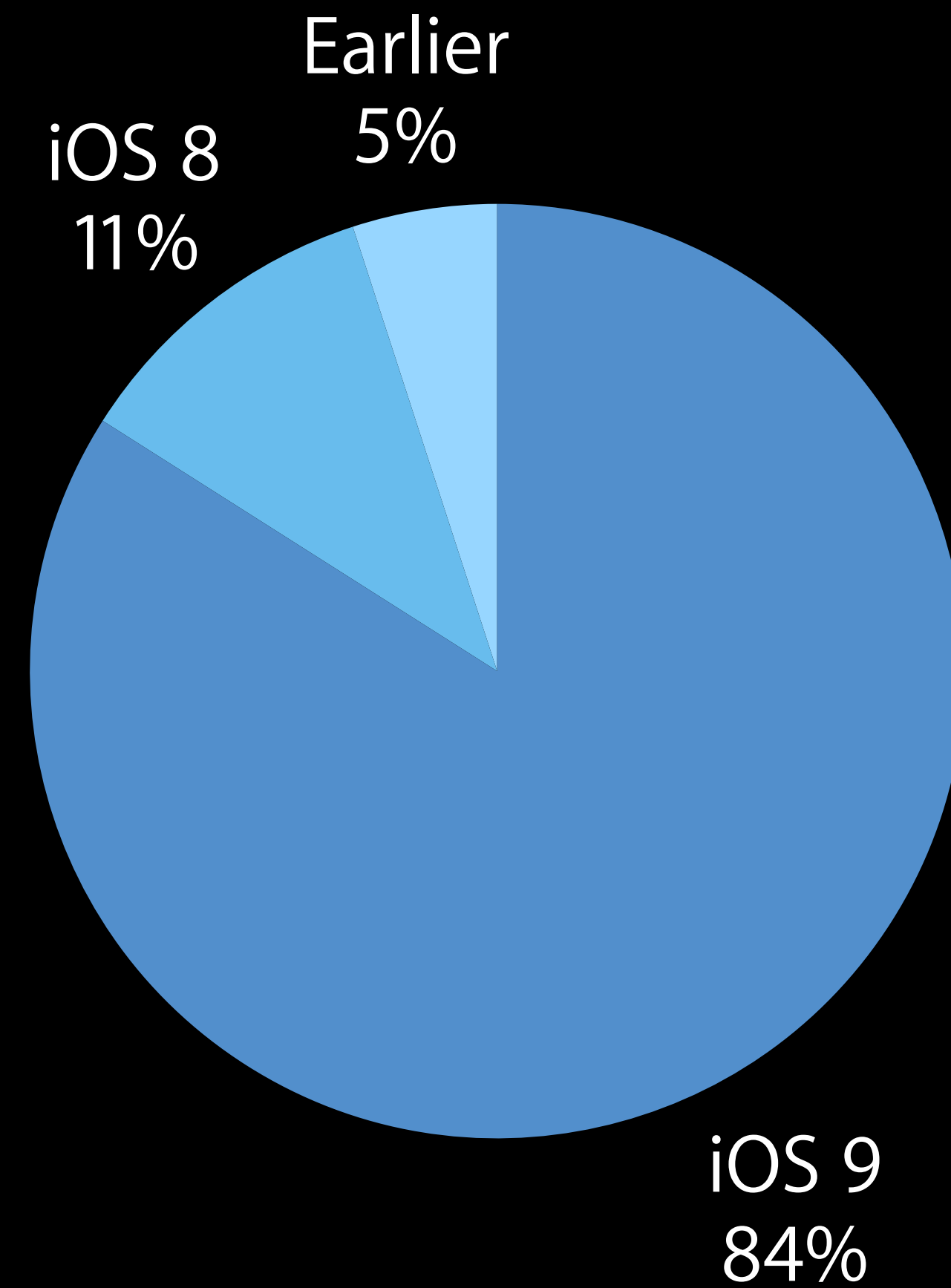


The Essentials

A very good place to start

Minimum Deployment of iOS 8

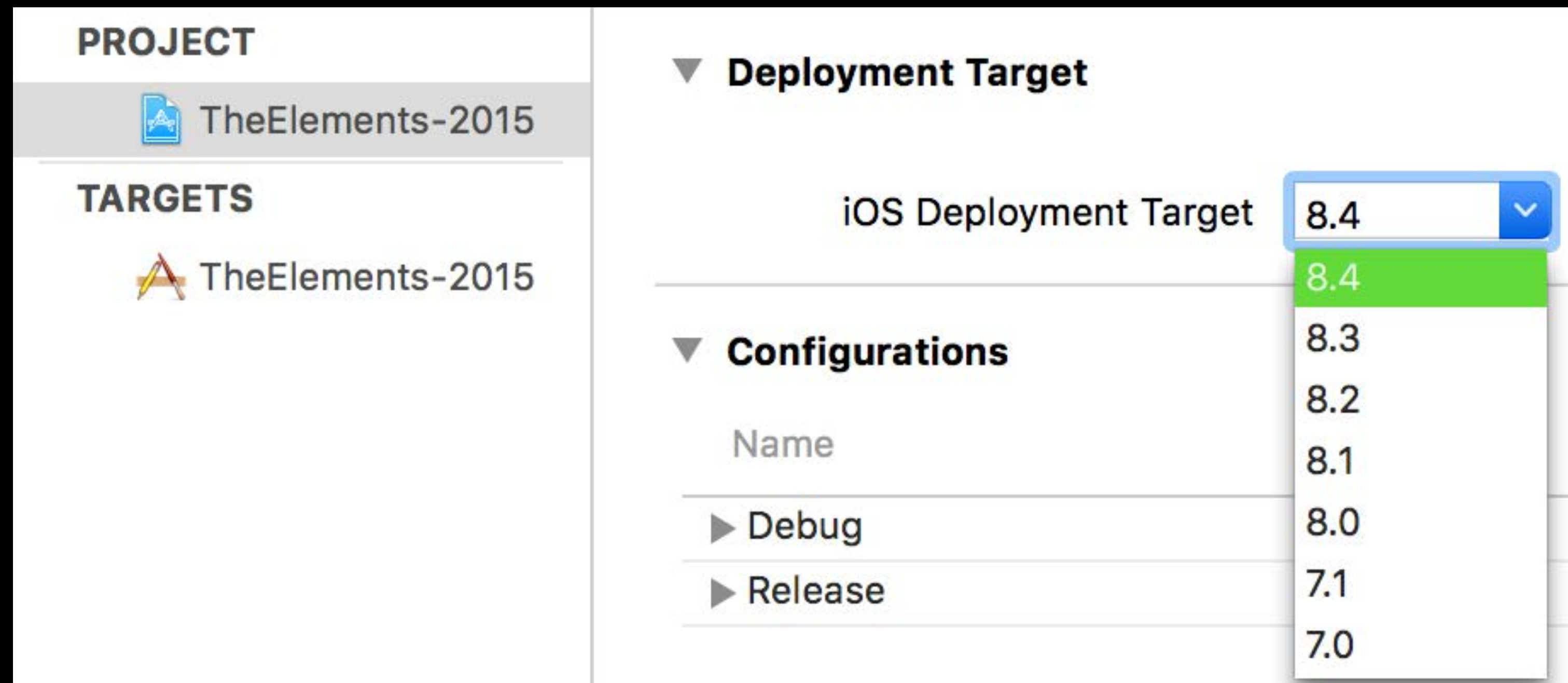
- 95% of Devices



As measured by the App Store on May 9, 2016

Pick a Deployment Target

Latest update of previous release






The screenshot shows the Xcode interface for a project named 'TheElements-2015'. The 'TARGETS' list contains one target, 'TheElements-2015'. The 'Deployment Target' dropdown menu is open, showing a list of iOS versions from 7.0 to 8.4. The version 8.4 is highlighted in green, indicating it is the selected target.


Deployment Target	Selected
8.4	Yes
8.3	No
8.2	No
8.1	No
8.0	No
7.1	No
7.0	No

Deprecated API

▼ AtomicElement.m

- ▶  'drawAtPoint:withFont:' is deprecated: first deprecated in iOS 7.0 - Use -drawAtPoint:withAttributes:
- ▶  'sizeWithFont:' is deprecated: first deprecated in iOS 7.0 - Use -sizeWithAttributes:
- ▶  'drawAtPoint:withFont:' is deprecated: first deprecated in iOS 7.0 - Use -drawAtPoint:withAttributes:

▼ AtomicElementView.m

- ▶  Implicit conversion loses integer precision: 'NSUInteger' (aka 'unsigned long') to 'int'

Deprecated API

▼ AtomicElement.m

- ▶ ⚠ 'drawAtPoint:withFont:' is deprecated: first deprecated in iOS 7.0 - Use -drawAtPoint:withAttributes:
- ▶ ⚠ 'sizeWithFont:' is deprecated: first deprecated in iOS 7.0 - Use -sizeWithAttributes:

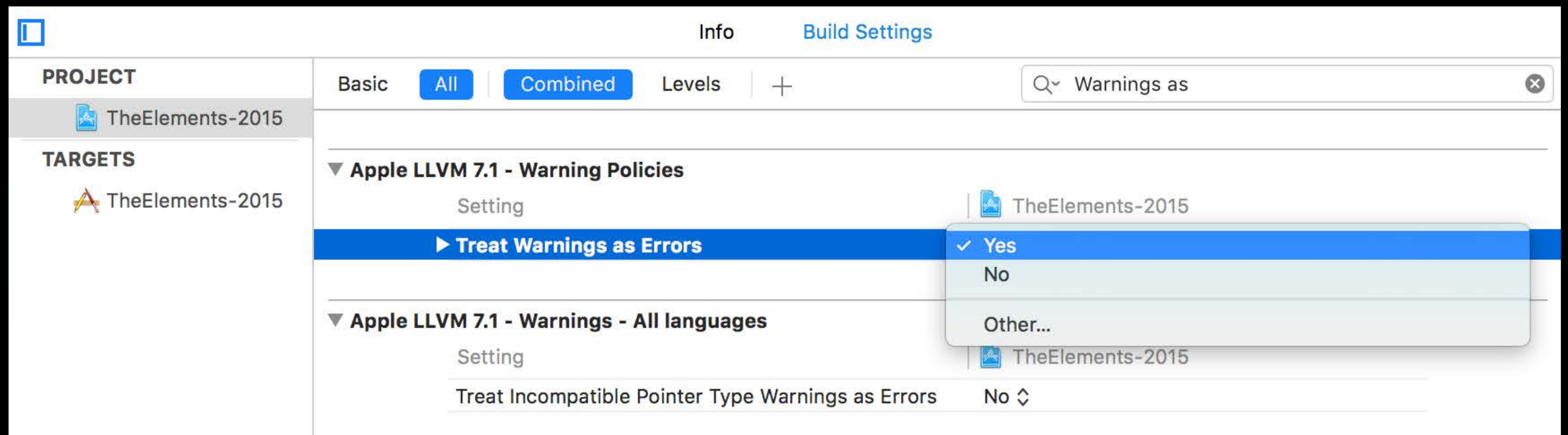
▶ ⚠ 'drawAtPoint:withFont:' is deprecated: first deprecated in iOS 7.0 - Use -drawAtPoint:withAttributes:

▼ AtomicElementView.m

- ▶ ⚠ Implicit conversion loses integer precision: 'NSUInteger' (aka 'unsigned long') to 'int'

Treat Warnings as Errors

LLVM—Warning Policy, Xcode 7.3



Treat Warnings as Errors

Swift Compiler—Warning Policy, Xcode 8

NEW

The screenshot shows the Xcode Build Settings interface for a project named 'DemoClangWarning...'. The 'Build Settings' tab is active, and the 'All' configuration is selected. A search filter 'warnings as' is applied to the settings list. The settings are organized into three sections:

- Apple LLVM 8.0 - Warning Policies**: The setting 'Treat Warnings as Errors' is set to 'Yes'.
- Apple LLVM 8.0 - Warnings - All languages**: The setting 'Treat Incompatible Pointer Type Warnings as Errors' is set to 'No'.
- Swift Compiler - Warnings Policies**: The setting 'Treat Warnings as Errors' is set to 'Yes'.

The 'Treat Warnings as Errors' setting in the Swift Compiler section is highlighted in blue.

Accessibility in Xcode

As much a part of your user interface as the artwork

The image shows a screenshot of the Xcode interface. On the left, a menu is open for the 'Demo1' window. The menu items are: New (⌘N), Open... (⌘O), Open Recent (with a submenu arrow), Close (⌘W), Save... (⌘S), Save As... (⇧⌘S), Revert to Saved, Page Setup... (⇧⌘P), and Print... (⌘P). On the right, the Accessibility Inspector is visible, showing the following settings:

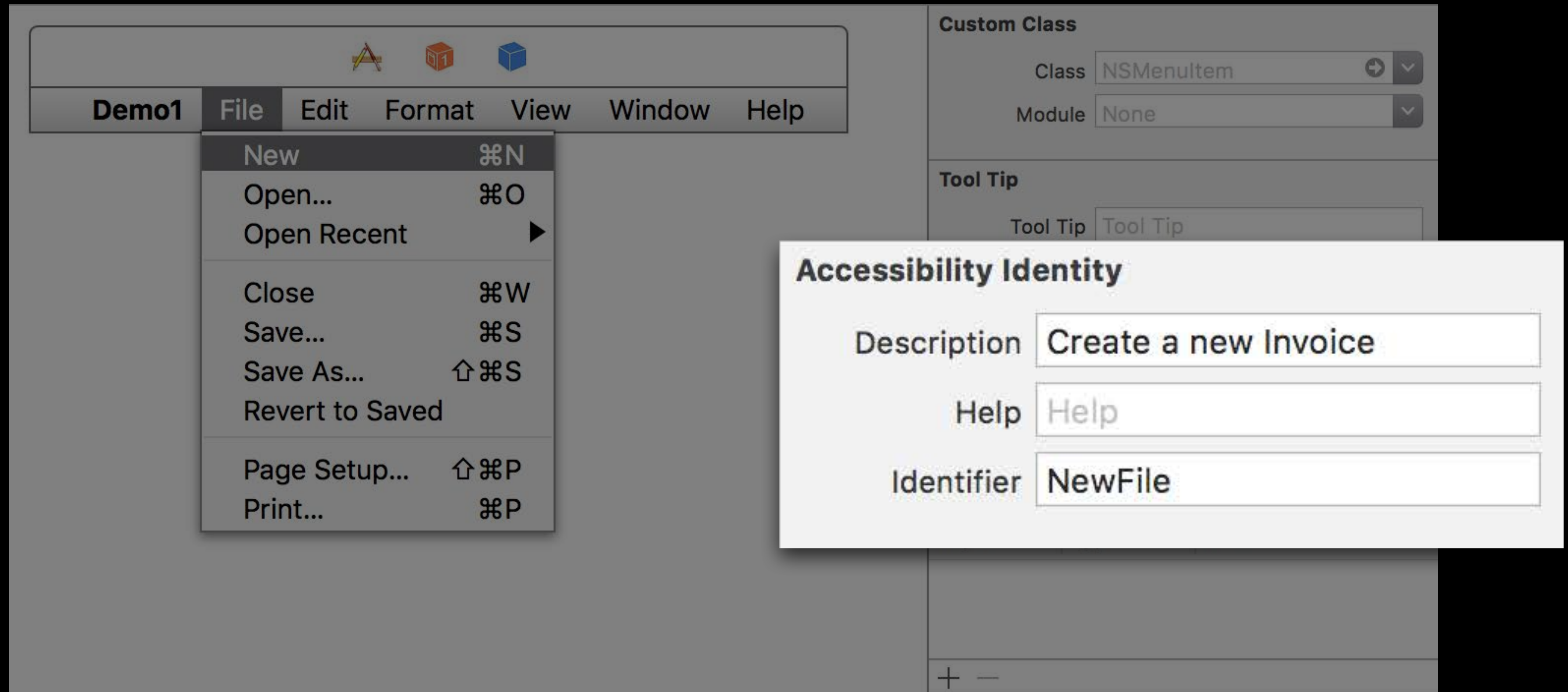
- Custom Class:** Class is set to `NSMenuItem`, Module is `None`.
- Tool Tip:** Tool Tip is `Tool Tip`.
- Accessibility Identity:** Description is `Create a new Invoice`, Help is `Help`, and Identifier is `NewFile`.
- User Defined Runtime Attributes:** A table with columns for Key Path, Type, and Value.

Key Path	Type	Value

At the bottom of the Accessibility Inspector, there are expand/collapse buttons (+ and -).

Accessibility in Xcode

As much a part of your user interface as the artwork



Locale-Aware APIs

Locale

Obtain current region and format

DateFormatter

Format and parse dates and times

NumberFormatter

Format and parse numbers

Calendar and TimeZone

Current calendar and associated operations

Dimension

Represent specific units of measure

MeasurementFormatter

Format and parse units of quantity

PersonNameComponents

Names of people

Locale-Aware APIs

NEW

Locale

Obtain current region and format

DateFormatter

Format and parse dates and times

NumberFormatter

Format and parse numbers

Calendar and TimeZone

Current calendar and associated operations

Dimension

Represent specific units of measure

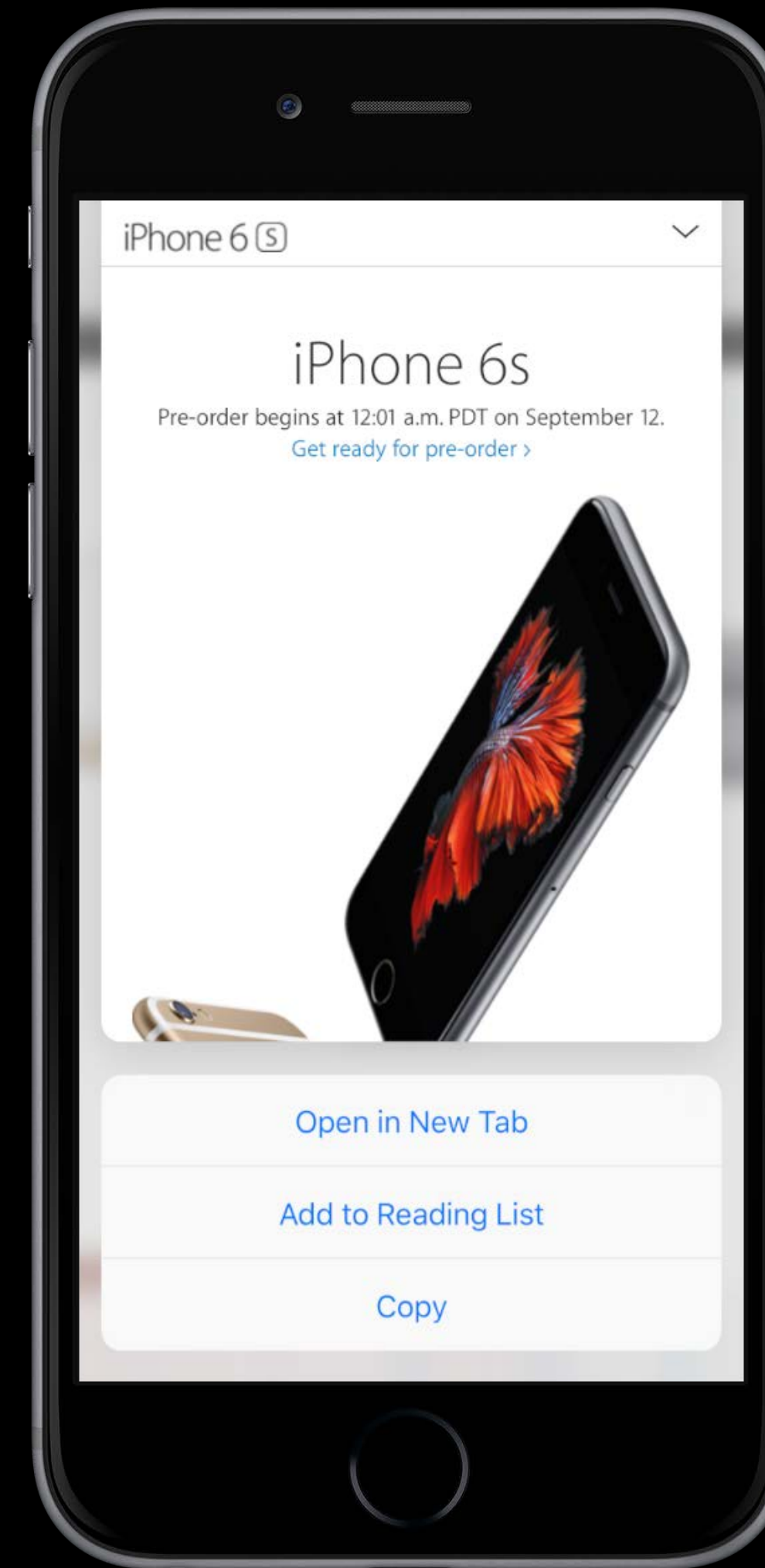
MeasurementFormatter

Format and parse units of quantity

PersonNameComponents

Names of people

Add Peek, Pop, and Quick Actions



Getting Ready for Autumn 2016

Checklist for developer preview seeds

Run the Swift Migrator

Does it work?

- No? Is it Apple's problem or is it your code?
- Are you using APIs in a way we didn't expect?
- Unit tests pass?

File Bug Reports

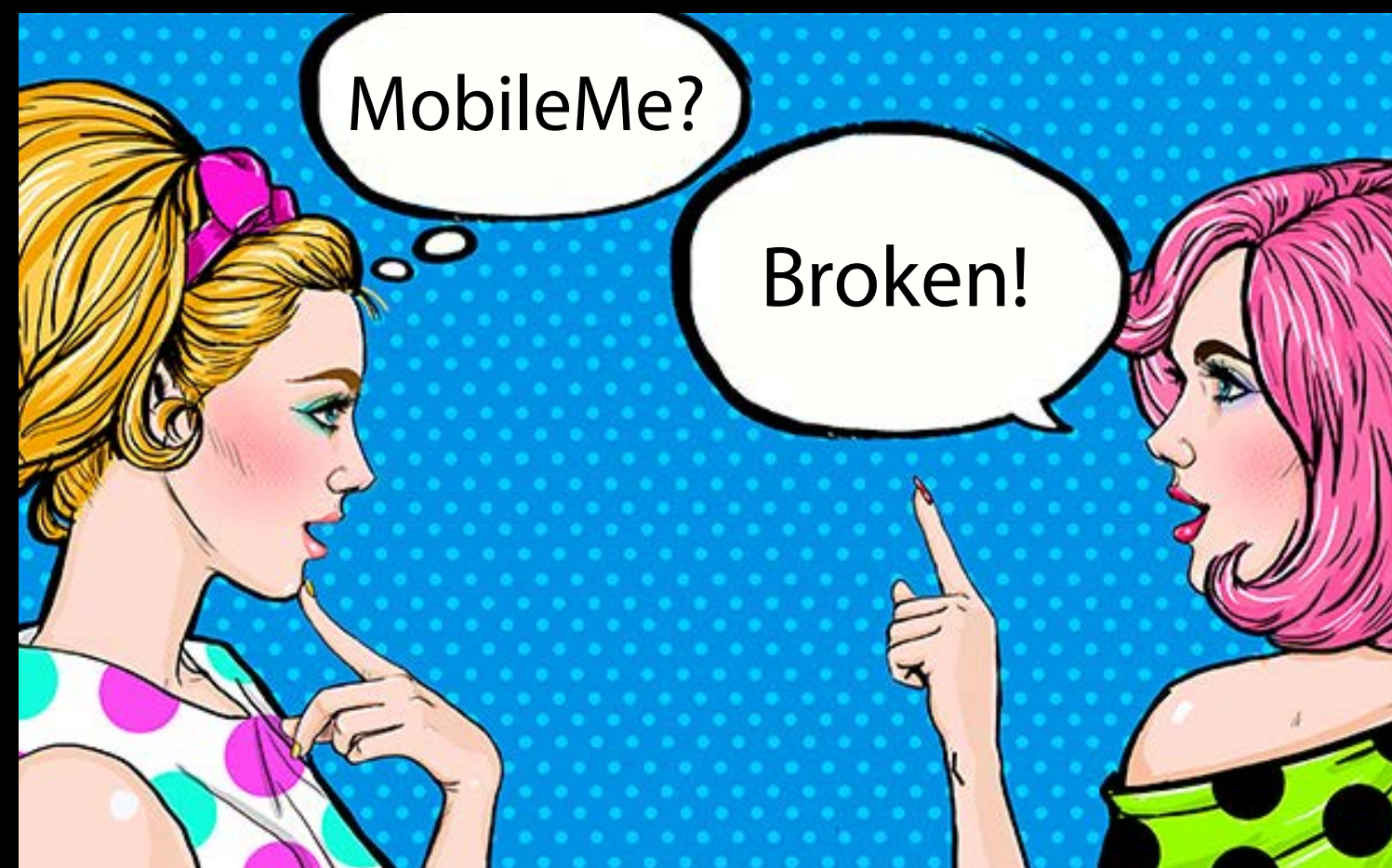
bugreport.apple.com

File Bug Reports

bugreport.apple.com



Banter



Dev Forums



Email



iOS

[Using Apple Bug Reporter](#)

Please include only one issue per report. All fields are **required** except where noted. While we cannot respond directly to every report, all reports are reviewed by the appropriate engineering teams.

Problem will be AutoSaved in 2 minutes.

Classification* Serious Bug

Reproducibility* Always

Please be sure your issue does not better fit under another more specific classification.

[Show instructions for gathering logs](#)

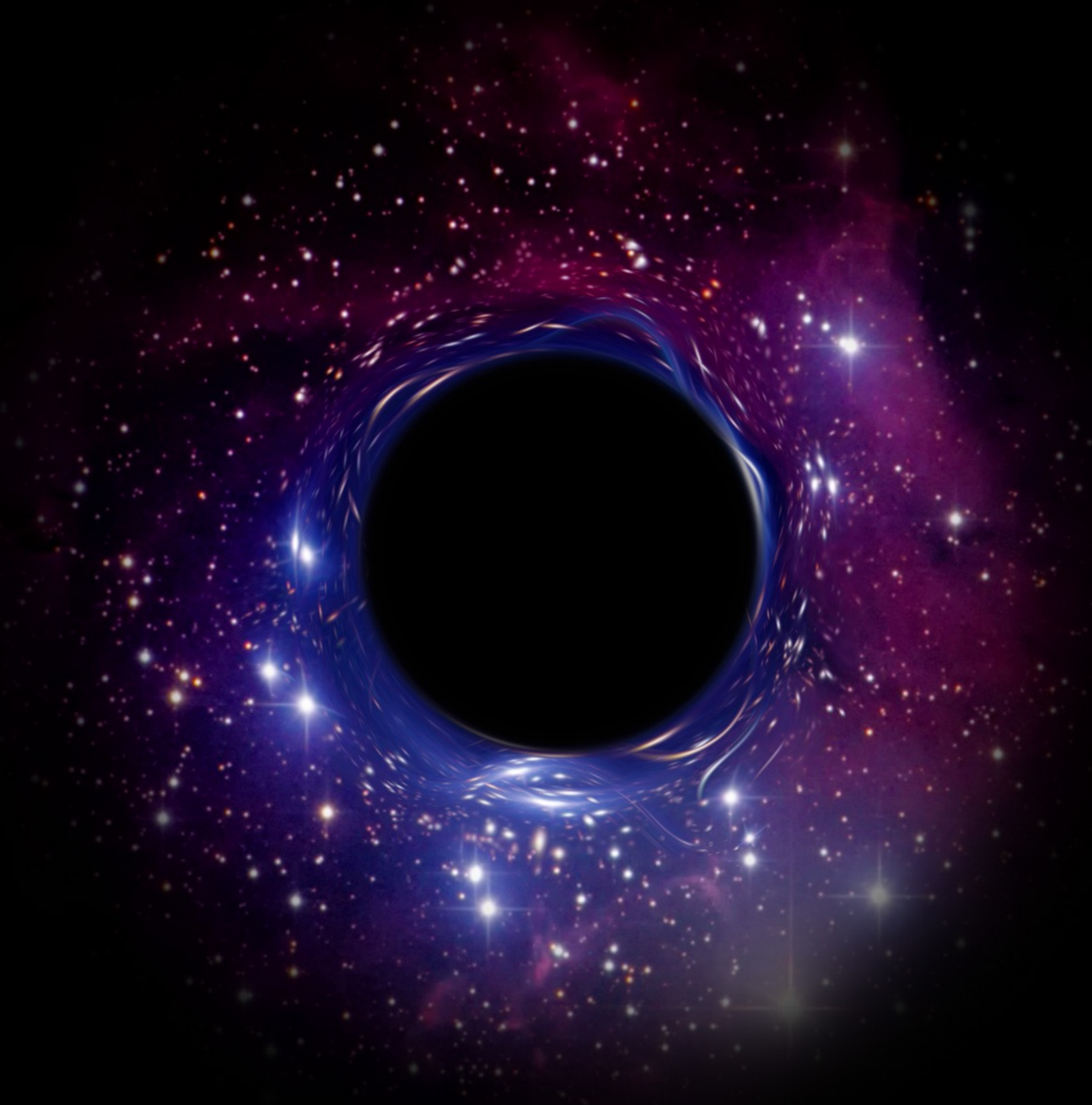
Attach a file

Title: Provide a short but descriptive sentence that summarizes the issue

Example: Compass app shows the wrong direction

BugReport.apple.com

Where Do Lonely Bug Reports Go?



“La-La, Can’t Hear You”



“Each bug report is as unique as a snowflake.”

Paul M., Apple Software Engineer

Get Started Today

Prepare your codebase

iOS 9.3 + Xcode 7

Update project settings

Fix warnings

Replace deprecated API

Localize

Accessibility

iOS 10 + Xcode 8

Use the Swift Migrator

Provide bug reports

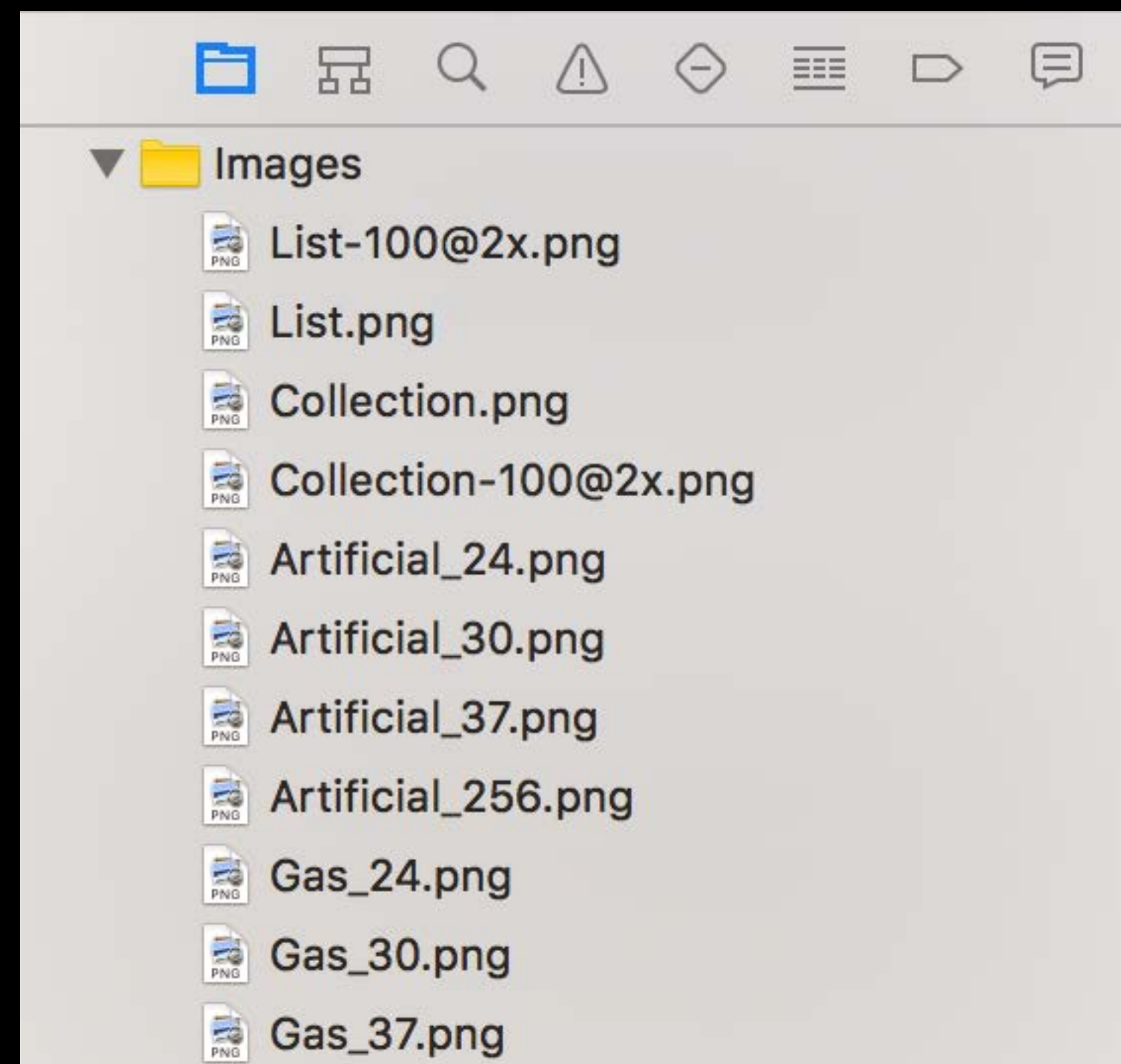
Incorporate new API

Asset Catalogs

Or, as some might write, Catalogues

The Old Way

Files with naming conventions



The Modern Way









Asset catalog

The screenshot displays the Xcode Asset Catalog interface. On the left, a sidebar shows the project structure, with 'Media.xcassets' selected. The main area is divided into three sections:

- Asset List:** A list of assets including 'ARTIFICIAL_30', 'Artificial_37', 'Artificial_256', 'Collection-100', 'Collection', 'Default', 'Gas_24', 'Gas_30', 'Gas_37', 'Gas_256', 'Insert Table', 'Liquid_24', 'Liquid_30', 'Liquid_37', 'Liquid_256', 'List-100', 'List', 'long_battery_life' (highlighted), 'RoundedRectangle', 'Solid_24', 'Solid_30', 'Solid_37', and 'Solid_256'.
- Preview:** A preview of the 'long_battery_life' asset, showing a clock icon with a battery symbol and the text 'All' on a blue background, labeled 'Universal'.
- Image Set Configuration:** A panel on the right with the following settings:
 - Name:** long_battery_life
 - Render As:** Default
 - Compression:** Automatic
 - Devices:** All Universal, iOS iPhone, iPad, OS X Mac, tvOS Apple TV, watchOS Apple Watch
 - Scale Factors:** Single Vector
 - Width:** Any
 - Height:** Any
 - Direction:** Fixed
 - Color:** Any
 - Memory:** 1 GB, 2 GB, 4 GB

Add a Catalog

Choose a template for your new file:

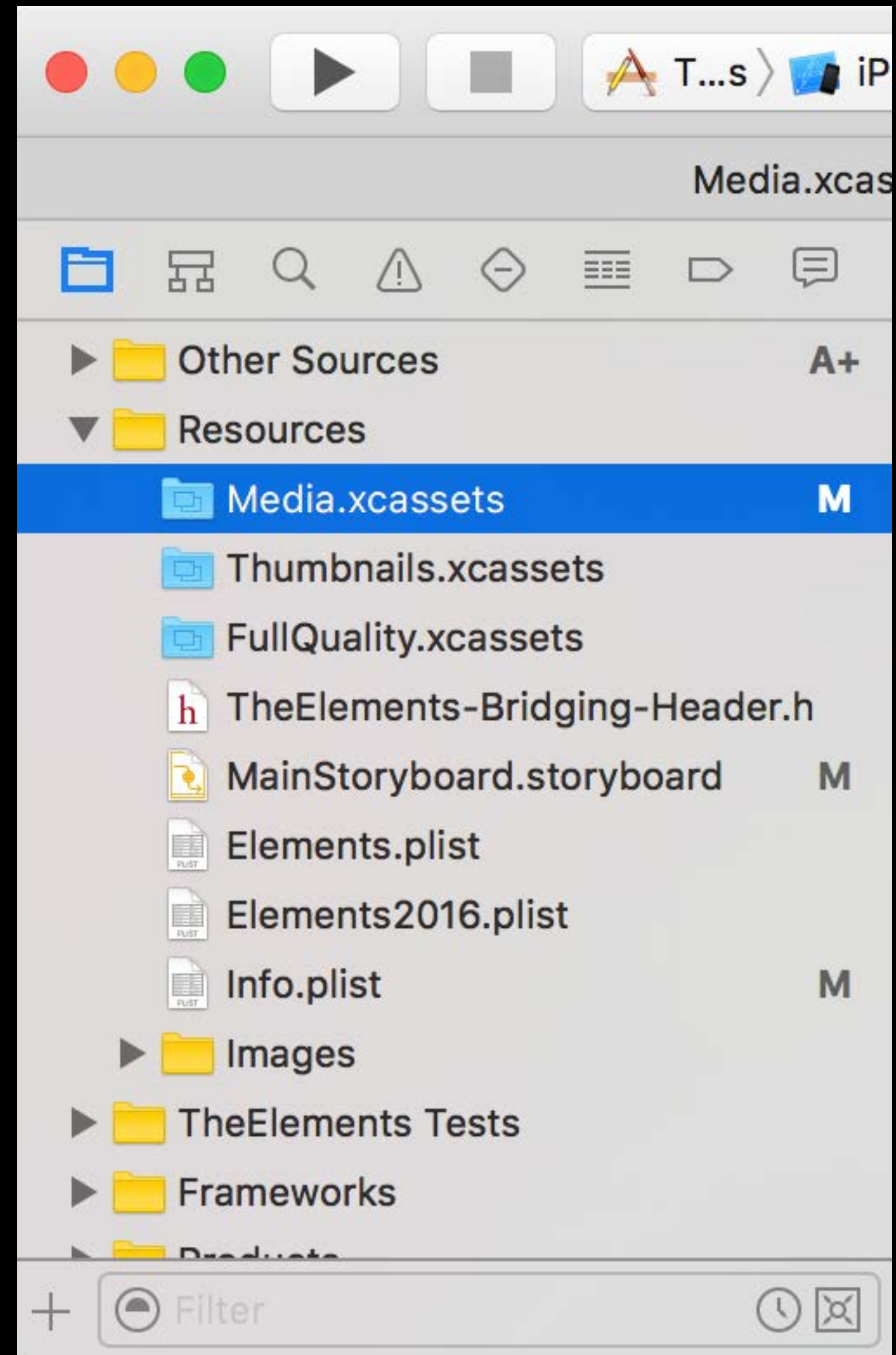
iOS	 GeoJSON File	 GPX File	 Asset Catalog	 Settings Bundle
Source				
User Interface				
Core Data				
Apple Watch				
Resource	 Property List	 Rich Text File	 SceneKit Particle System	 SceneKit Scene File
Other				
watchOS				
Source				
User Interface				
Core Data				
Resource				
Other				
tvOS				
Source				
User Interface				
Core Data				
Resource				

Asset Catalog

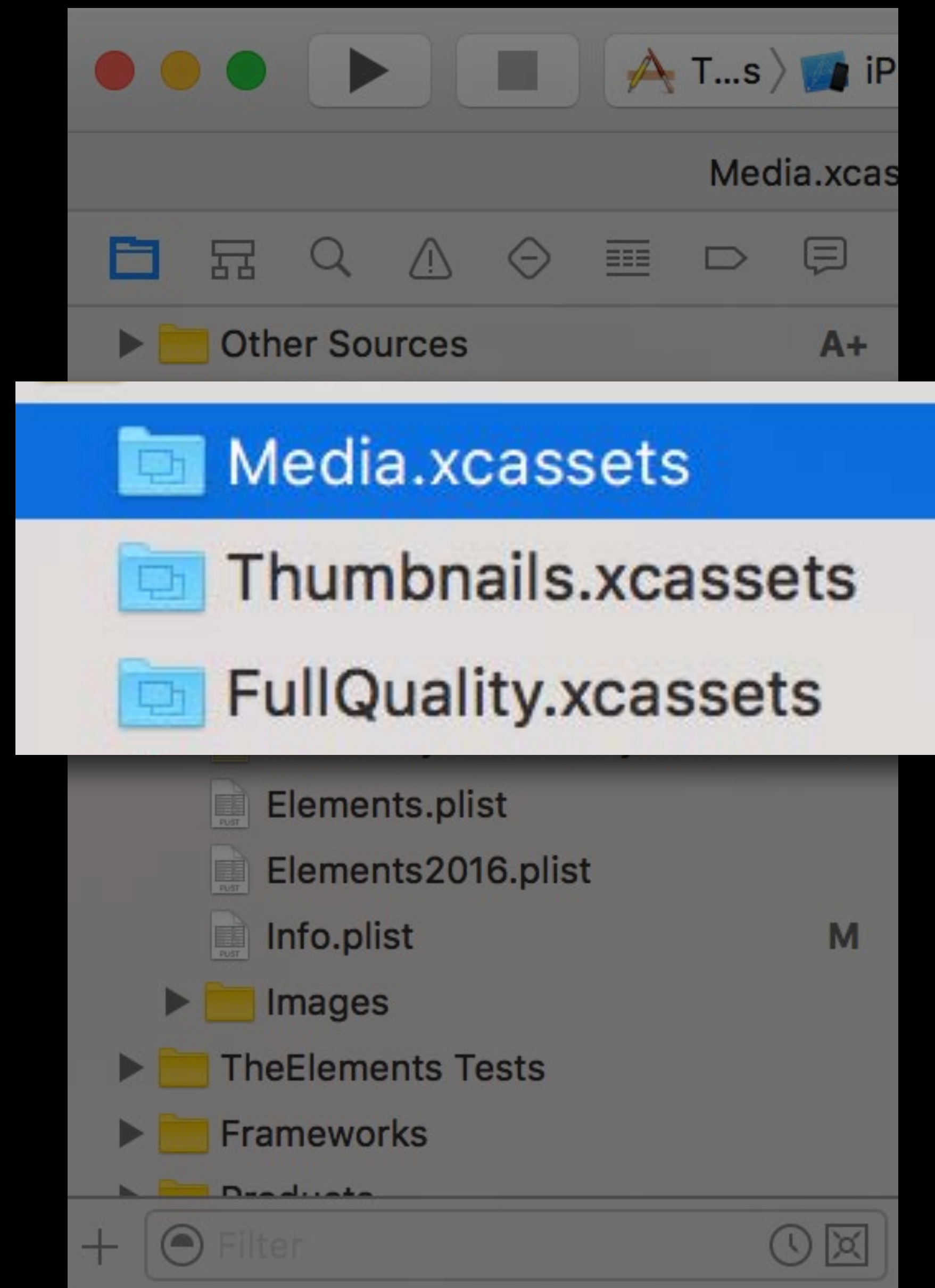
Asset catalogs store and categorize resources for different platforms, devices, and capabilities (such as scale factors). When built, items in asset catalogs are compiled into a unified, efficient runtime format or exported to their expected format (based on use).

Cancel Previous Next

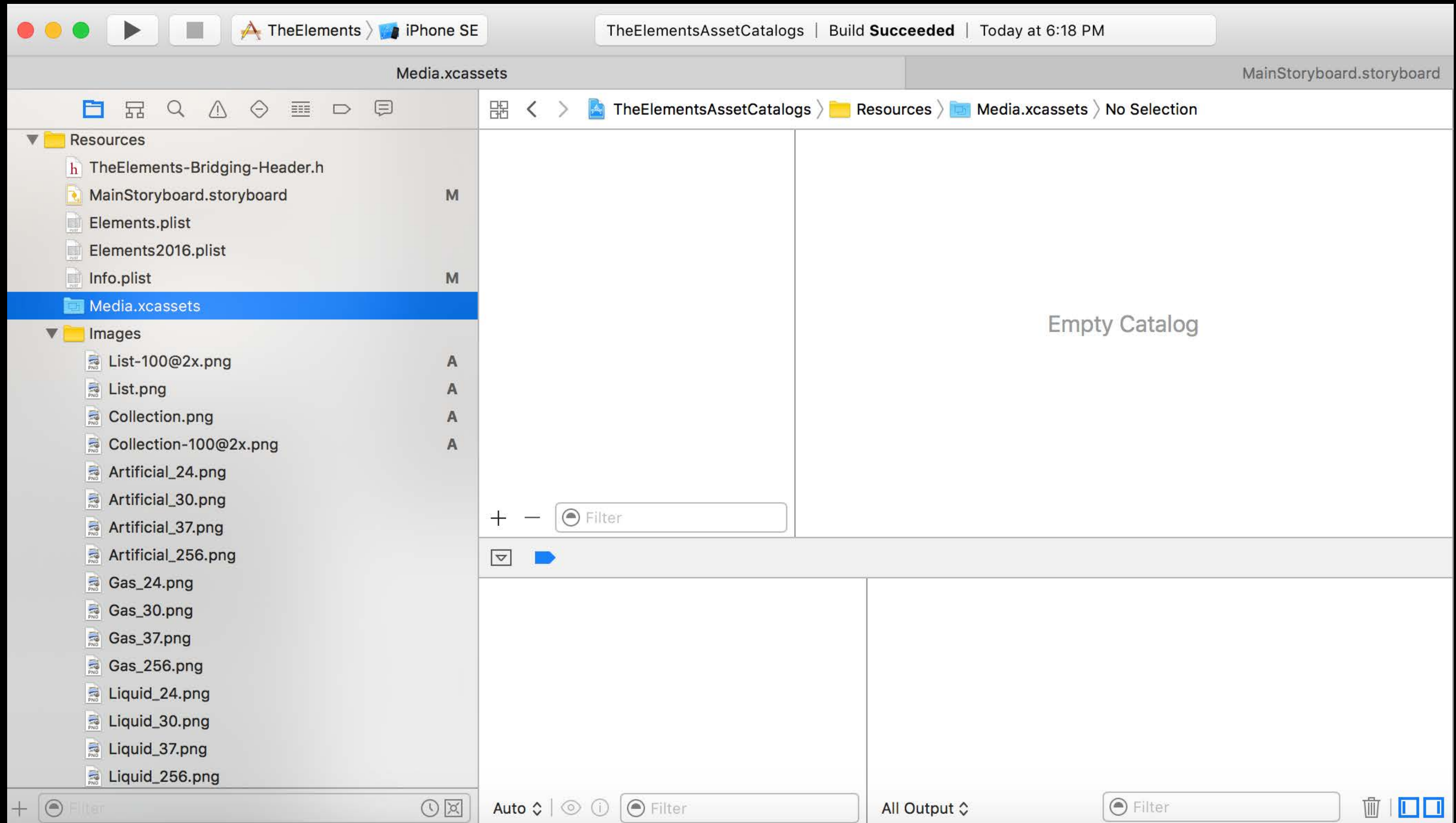
Multiple Asset Catalogs



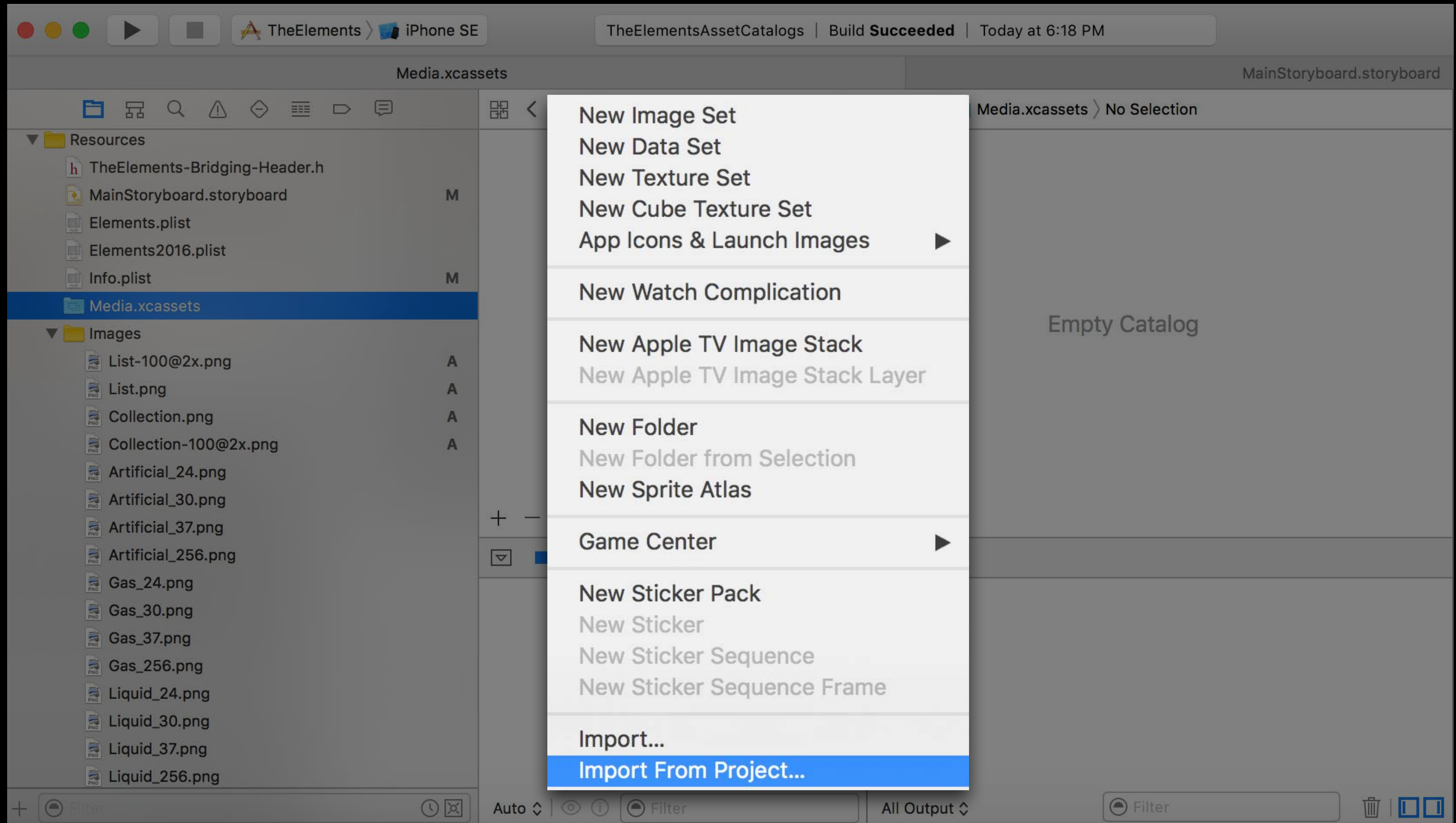
Multiple Asset Catalogs



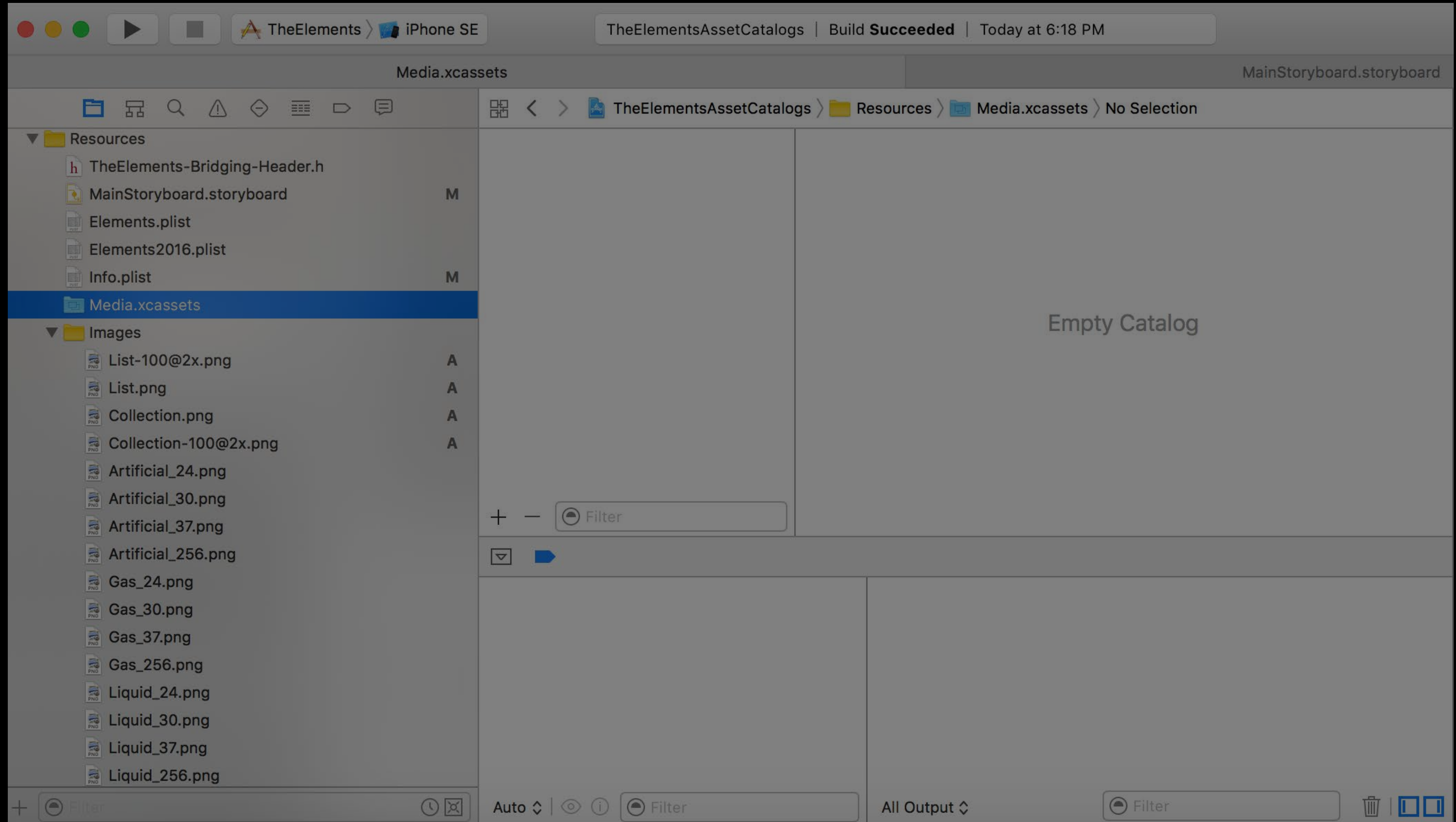
Migrating Project Images



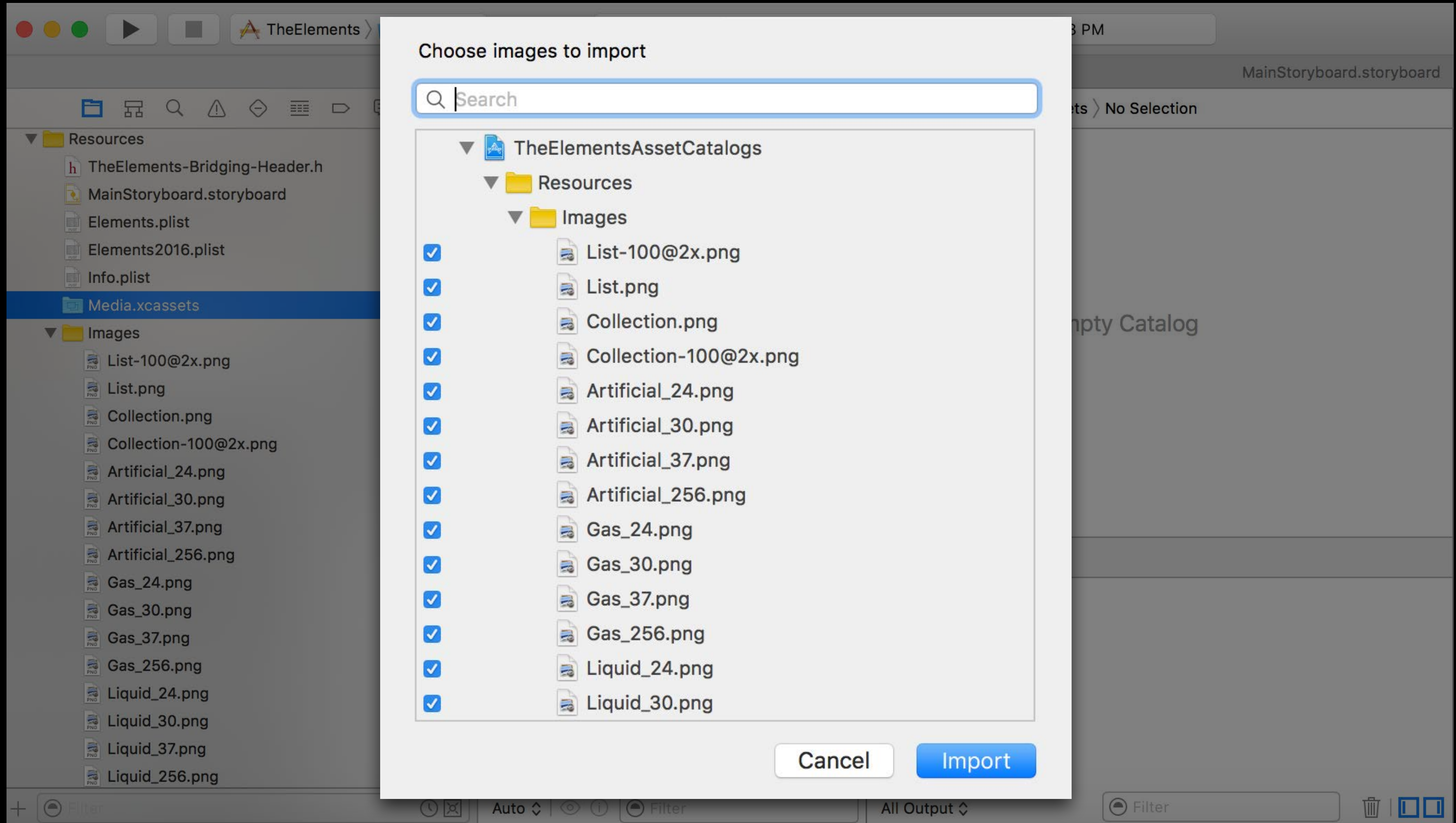
Migrating Project Images



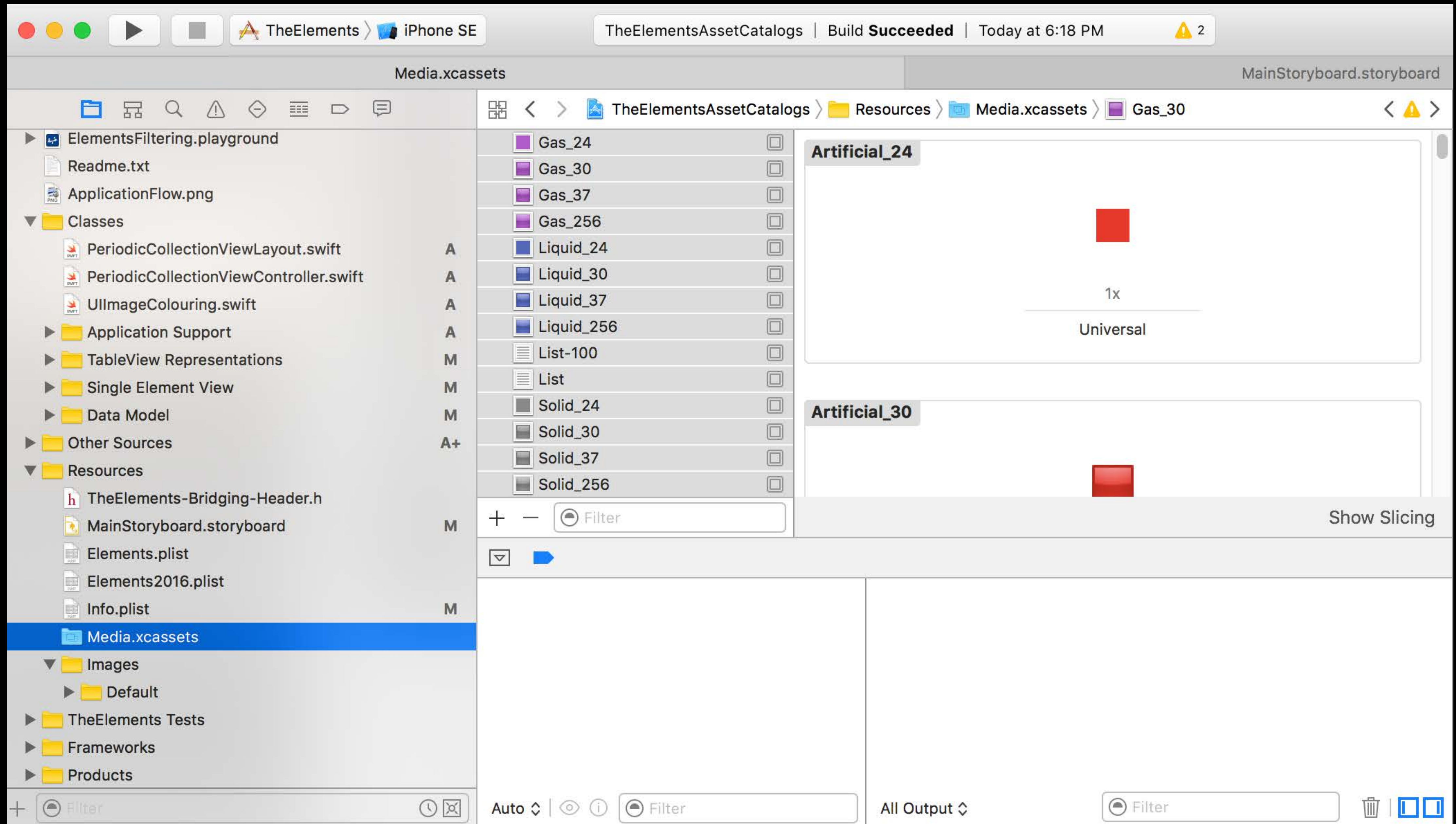
Migrating Project Images



Migrating Project Images



Migration Completed




```
// Loading Image Assets
```

```
// Old Way
```

```
if let logo = UIImage(contentsOfFile: Bundle.main().pathForResource("Logo", ofType: "png")) {  
    imageView.image = logo  
}
```

```
// Modern Way
```

```
if let logo = UIImage(named: "List") {  
    imageView.image = logo  
}
```

```
// Loading Image Assets
```

```
// Old Way
```

```
if let logo = UIImage(contentsOfFile: Bundle.main().pathForResource("Logo", ofType: "png")) {  
    imageView.image = logo  
}
```

```
// Modern Way
```

```
if let logo = UIImage(named: "List") {  
    imageView.image = logo  
}
```



```
// Loading Image Assets
```

```
// Old Way
```

```
if let logo = UIImage(contentsOfFile: Bundle.main().pathForResource("Logo", ofType: "png")) {  
    imageView.image = logo  
}
```

```
// Modern Way
```

```
if let logo = UIImage(named: "List") {  
    imageView.image = logo  
}
```

```
// Loading Image Assets
```

```
// Old Way
```

```
if let logo = UIImage(contentsOfFile: Bundle.main().pathForResource("Logo", ofType: "png")) {  
    imageView.image = logo  
}
```

```
// Modern Way
```

```
if let logo = UIImage(named: "List") {  
    imageView.image = logo  
}
```


Multiple Representations of Assets

The screenshot displays the Xcode Asset Catalog interface. On the left is a list of asset categories, with 'XcodeBeta' selected. The main area shows three representations of the 'XcodeBeta' asset: a 1x version, a 2x version, and a 3x version. Below these is a 'Universal' representation. The right-hand panel, titled 'Image Set', contains configuration options for the selected asset set.

Asset List:

- Artificial_24
- Artificial_30
- Artificial_37
- Artificial_256
- Collection-100
- Collection
- Default
- Gas_24
- Gas_30
- Gas_37
- Gas_256
- Insert Table
- Liquid_24
- Liquid_30
- Liquid_37
- Liquid_256
- List-100
- List
- long_battery_life
- RoundedRectangle
- Solid_24
- Solid_30
- Solid_37
- Solid_256
- XcodeBeta**

XcodeBeta Asset Representations:

- 1x
- 2x
- 3x
- Universal

Image Set Configuration:

- Name: XcodeBeta
- Render As: Default
- Compression: Lossless (Inherited)
- Devices:
 - All Universal
 - iOS iPhone
 - iPad
 - OS X Mac
 - tvOS Apple TV
 - watchOS Apple Watch
- Scale Factors: Individual Scales
- Width: Any
- Height: Any
- Direction: Fixed
- Color: Any
- Memory:
 - 1 GB
 - 2 GB
 - 4 GB
- Graphics:
 - Metal 1v2
 - Metal 2v2
 - Metal 3v1
 - Metal 3v2

Multiple Representations of Assets

The screenshot displays the Xcode Asset Catalog interface. On the left is a list of asset categories, with 'XcodeBeta' selected. The main area shows three representations of the 'XcodeBeta' asset: a 1x version, a 2x version, and a 3x version. Below these is a 'Universal' representation. The right-hand panel, titled 'Image Set', contains configuration options for the asset set.

Asset List:

- Artificial_24
- Artificial_30
- Artificial_37
- Artificial_256
- Collection-100
- Collection
- Default
- Gas_24
- Gas_30
- Gas_37
- Gas_256
- Insert Table
- Liquid_24
- Liquid_30
- Liquid_37
- Liquid_256
- List-100
- List
- long_battery_life
- RoundedRectangle
- Solid_24
- Solid_30
- Solid_37
- Solid_256
- XcodeBeta**

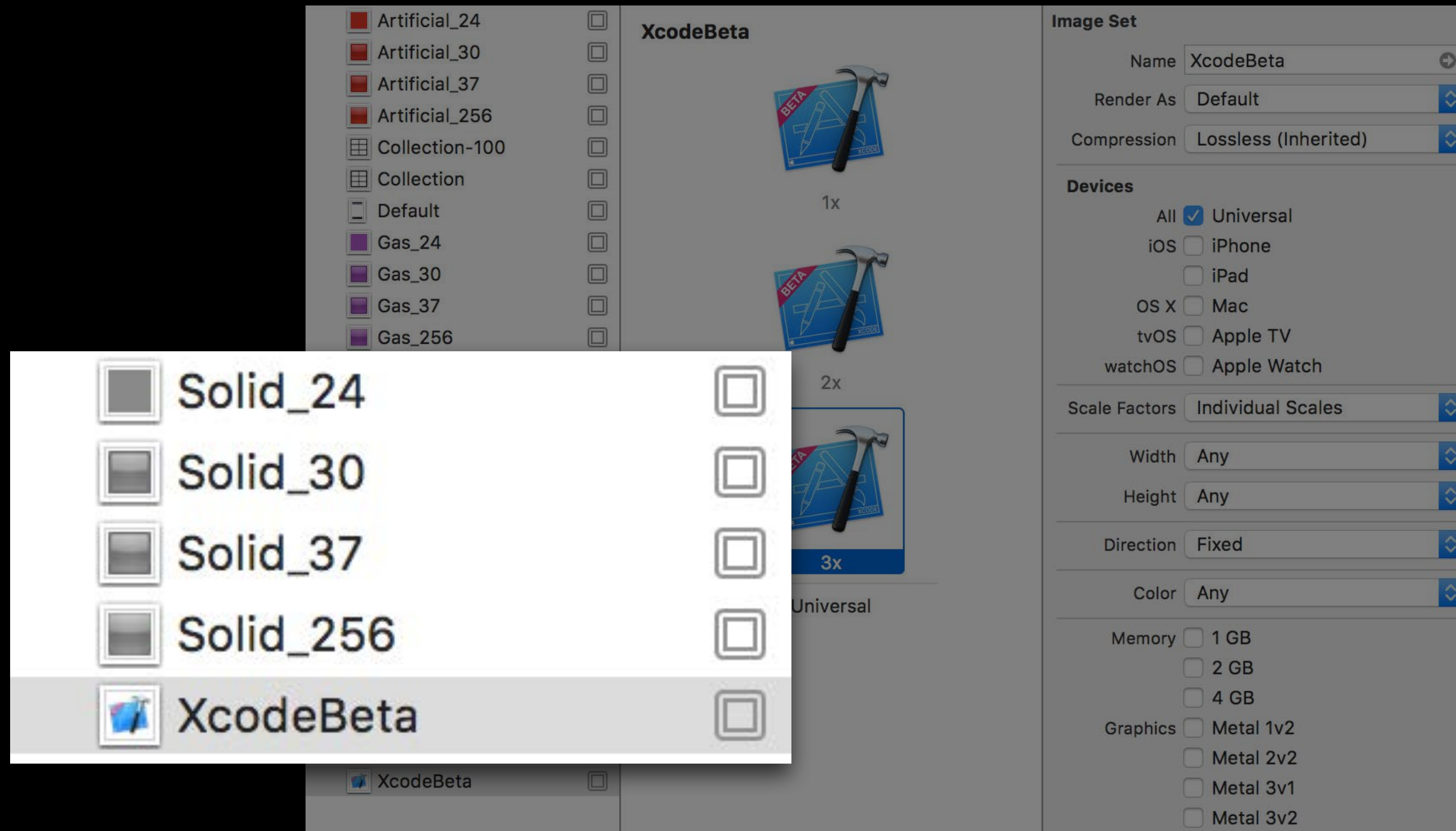
XcodeBeta Asset Representations:

- 1x
- 2x
- 3x
- Universal

Image Set Configuration:

- Name: XcodeBeta
- Render As: Default
- Compression: Lossless (Inherited)
- Devices:
 - All Universal
 - iOS iPhone
 - iPad
 - OS X Mac
 - tvOS Apple TV
 - watchOS Apple Watch
- Scale Factors: Individual Scales
- Width: Any
- Height: Any
- Direction: Fixed
- Color: Any
- Memory:
 - 1 GB
 - 2 GB
 - 4 GB
- Graphics:
 - Metal 1v2
 - Metal 2v2
 - Metal 3v1
 - Metal 3v2

Multiple Representations of Assets



Multiple Representations of Assets

The screenshot displays the Xcode Asset Catalog interface. On the left is a list of asset categories, with 'XcodeBeta' selected at the bottom. The main area shows the 'XcodeBeta' asset with three representations: 1x, 2x, and 3x. The 3x representation is highlighted with a blue bar at the bottom and labeled 'Universal'. On the right, the 'Image Set' configuration panel is visible, showing settings for Name, Render As, Compression, Devices, Scale Factors, Width, Height, Direction, Color, Memory, and Graphics.

Asset List:

- Artificial_24
- Artificial_30
- Artificial_37
- Artificial_256
- Collection-100
- Collection
- Default
- Gas_24
- Gas_30
- Gas_37
- Gas_256
- Insert Table
- Liquid_24
- Liquid_30
- Liquid_37
- Liquid_256
- List-100
- List
- long_battery_life
- RoundedRectangle
- Solid_24
- Solid_30
- Solid_37
- Solid_256
- XcodeBeta**

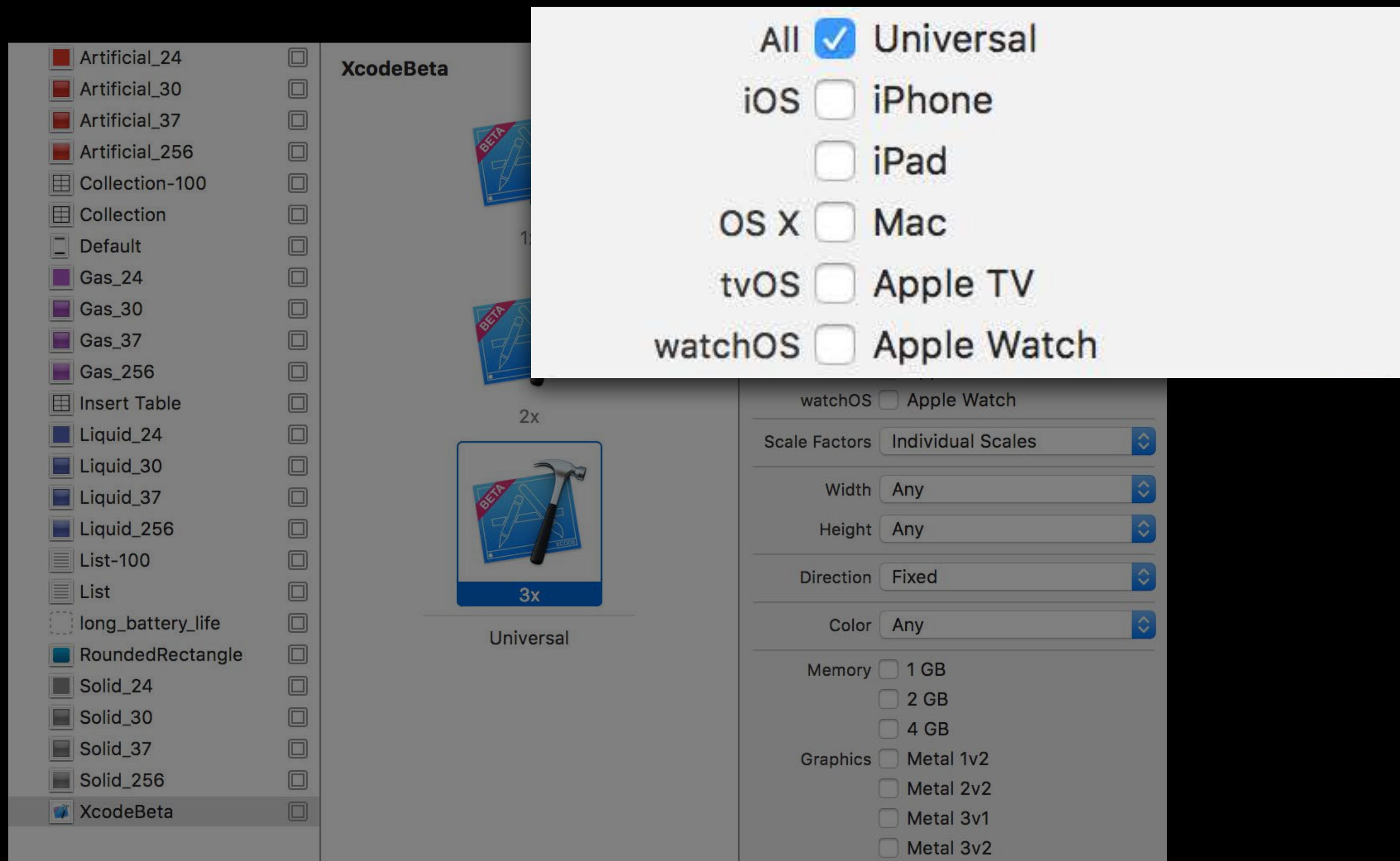
XcodeBeta Asset Representations:

- 1x
- 2x
- 3x (Universal)

Image Set Configuration:

- Name: XcodeBeta
- Render As: Default
- Compression: Lossless (Inherited)
- Devices:
 - All Universal
 - iOS iPhone
 - iPad
 - OS X Mac
 - tvOS Apple TV
 - watchOS Apple Watch
- Scale Factors: Individual Scales
- Width: Any
- Height: Any
- Direction: Fixed
- Color: Any
- Memory:
 - 1 GB
 - 2 GB
 - 4 GB
- Graphics:
 - Metal 1v2
 - Metal 2v2
 - Metal 3v1
 - Metal 3v2

Multiple Representations of Assets



Multiple Representations of Assets

The screenshot displays the Xcode Asset Catalog interface. On the left is a list of asset categories, with 'XcodeBeta' selected at the bottom. The main area shows the 'XcodeBeta' asset with three representations: 1x, 2x, and 3x. The 3x representation is highlighted with a blue border and a '3x' label. Below these is a 'Universal' representation. On the right, the 'Image Set' configuration panel is visible, showing settings for Name, Render As, Compression, Devices, Scale Factors, Width, Height, Direction, Color, Memory, and Graphics.

Asset List:

- Artificial_24
- Artificial_30
- Artificial_37
- Artificial_256
- Collection-100
- Collection
- Default
- Gas_24
- Gas_30
- Gas_37
- Gas_256
- Insert Table
- Liquid_24
- Liquid_30
- Liquid_37
- Liquid_256
- List-100
- List
- long_battery_life
- RoundedRectangle
- Solid_24
- Solid_30
- Solid_37
- Solid_256
- XcodeBeta**

XcodeBeta Asset Representations:

- 1x
- 2x
- 3x
- Universal

Image Set Configuration:

- Name: XcodeBeta
- Render As: Default
- Compression: Lossless (Inherited)
- Devices:
 - All Universal
 - iOS iPhone
 - iPad
 - OS X Mac
 - tvOS Apple TV
 - watchOS Apple Watch
- Scale Factors: Individual Scales
- Width: Any
- Height: Any
- Direction: Fixed
- Color: Any
- Memory:
 - 1 GB
 - 2 GB
 - 4 GB
- Graphics:
 - Metal 1v2
 - Metal 2v2
 - Metal 3v1
 - Metal 3v2

Multiple Representations of Assets

The image displays the Xcode Asset Catalog interface. On the left, the Asset Catalog sidebar lists various asset types such as Artificial, Collection, Default, Gas, Insert Table, Liquid, List, and XcodeBeta. The main area shows three representations of a hammer icon:

- 1x**: A standard resolution icon.
- 2x**: A high-resolution icon.
- 3x**: A high-resolution icon, highlighted with a blue bar and labeled **Universal** below it.

On the right, the target selection panel shows the following settings:

- XcodeBeta (selected)
- Default
- Lossless (Inherited)
- Universal
- iPhone
- iPad
- Mac
- Apple TV
- Apple Watch
- Individual Scales
- Any
- Any
- Fixed
- Any
- 1 GB
- 2 GB
- 4 GB
- Metal 1v2
- Metal 2v2
- Metal 3v1
- Metal 3v2

Multiple Representations of Assets

The screenshot displays the Xcode Asset Catalog interface. On the left is a list of asset categories, with 'XcodeBeta' selected. The main area shows three representations of the 'XcodeBeta' asset: a 1x version, a 2x version, and a 3x version. Below these is a 'Universal' representation. The right-hand panel, titled 'Image Set', contains configuration options for the asset set.

Asset List:

- Artificial_24
- Artificial_30
- Artificial_37
- Artificial_256
- Collection-100
- Collection
- Default
- Gas_24
- Gas_30
- Gas_37
- Gas_256
- Insert Table
- Liquid_24
- Liquid_30
- Liquid_37
- Liquid_256
- List-100
- List
- long_battery_life
- RoundedRectangle
- Solid_24
- Solid_30
- Solid_37
- Solid_256
- XcodeBeta**

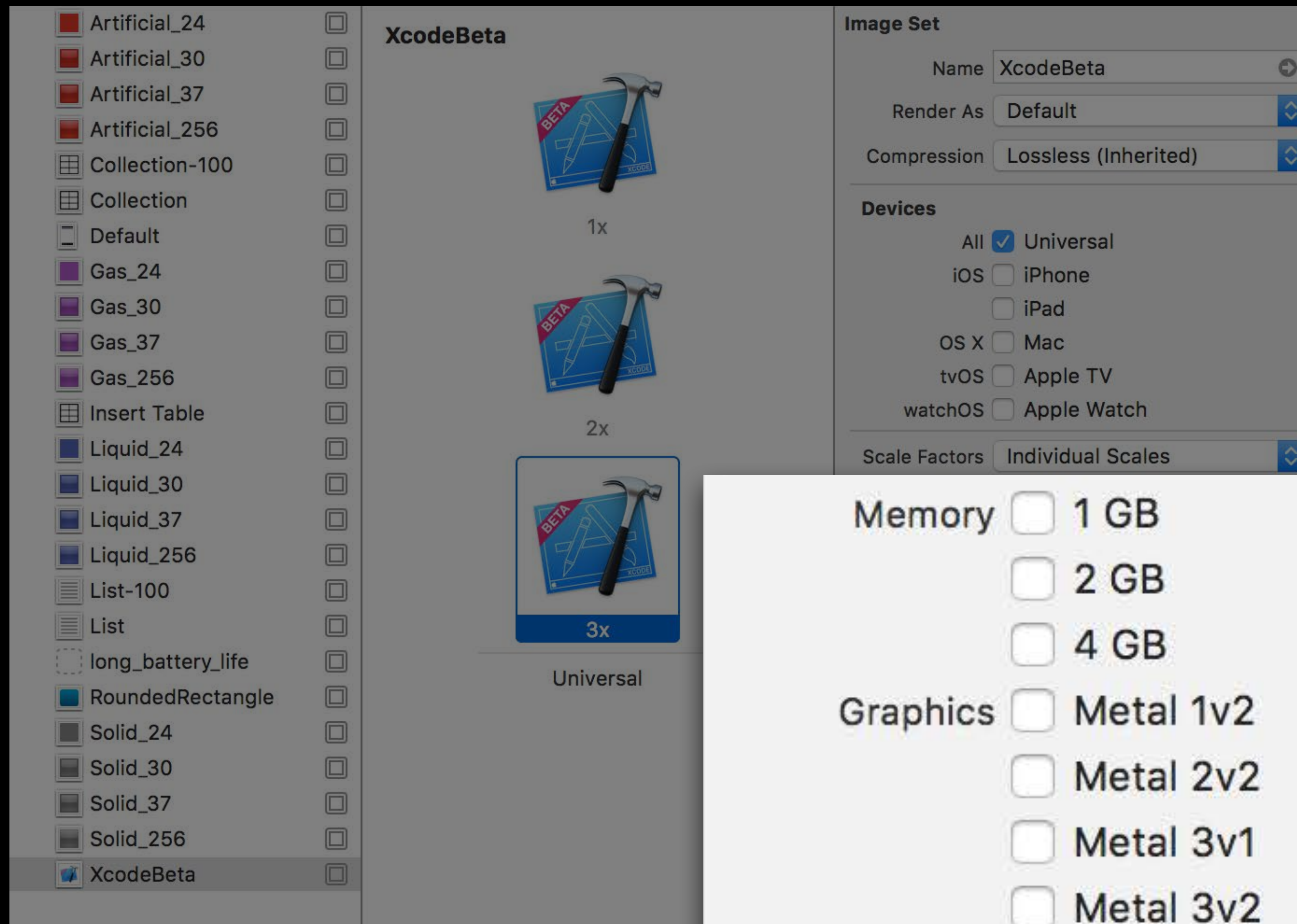
XcodeBeta Asset Representations:

- 1x
- 2x
- 3x
- Universal

Image Set Configuration:

- Name: XcodeBeta
- Render As: Default
- Compression: Lossless (Inherited)
- Devices:
 - All Universal
 - iOS iPhone
 - iPad
 - OS X Mac
 - tvOS Apple TV
 - watchOS Apple Watch
- Scale Factors: Individual Scales
- Width: Any
- Height: Any
- Direction: Fixed
- Color: Any
- Memory:
 - 1 GB
 - 2 GB
 - 4 GB
- Graphics:
 - Metal 1v2
 - Metal 2v2
 - Metal 3v1
 - Metal 3v2

Multiple Representations of Assets



Asset Types

XcodeBeta



1x

2x

3x

Universal

Individual Scales

PNGs

long_battery_life



All

Universal

Single Vector

PDFs

Selecting the Scale Factor

The screenshot shows the Xcode interface for managing an image set named "long_battery_life". On the left, the Asset Catalog displays a clock and battery icon, with "All" selected under the "Universal" device group. On the right, the "Image Set" settings panel is visible, showing the name "long_battery_life", "Render As" set to "Default", and "Compression" set to "Automatic". Under the "Devices" section, "All" is checked, and "Universal" is selected. The "Scale Factor" dropdown menu is open, showing three options: "Individual Scales", "Single Vector" (which is selected), and "Vector and Scales". The "Width" is set to "Any".

long_battery_life

All

Universal

Image Set

Name: long_battery_life

Render As: Default

Compression: Automatic

Devices

- All Universal
- iOS iPhone
- iPad
- OS X Mac
- tvOS Apple TV
- watchOS Apple Watch

Scale Factor: Single Vector

- Individual Scales
- Single Vector
- Vector and Scales

Width: Any

Missing Scaled Asset Representations

No 2x and 3x artwork



Artificial_256

1x 2x 3x

Universal

Image Set

Name

Render As

Compression

Devices

All Universal

iOS iPhone
 iPad

OS X Mac

tvOS Apple TV

watchOS Apple Watch

Missing Scaled Asset Representations



No 2x and 3x artwork

Missing Retina artwork

The screenshot shows the configuration for an image set named 'Artificial_256'. On the left, there are three asset slots: '1x' containing a hammer icon, '2x' which is empty, and '3x' which is empty. A 'Universal' label is positioned below the 2x and 3x slots. On the right, the 'Image Set' panel shows settings for Name, Render As, and Compression. Below that, the 'Devices' section has checkboxes for 'All', 'Universal', 'iOS', 'iPhone', 'iPad', 'OS X', 'Mac', 'tvOS', 'Apple TV', and 'watchOS', 'Apple Watch'.

Scaled up for Retina devices

Missing PNG Asset Representations

Only 3x media



Missing 1x and 2x artwork

The screenshot shows the Asset Catalog interface for an asset named 'Artificial_256'. It displays three asset slots: '1x', '2x', and '3x'. The '1x' and '2x' slots are empty, indicated by dashed red boxes and arrows pointing to them from the text 'Missing 1x and 2x artwork'. The '3x' slot contains a blue icon with a hammer and a pencil, labeled '3x'. Below the '1x' and '2x' slots, the word 'Universal' is written. To the right, the 'Image Set' panel shows the name 'Artificial_256', 'Render As' set to 'Default', and 'Compression' set to 'Lossless (Inherited)'. The 'Devices' section has 'All' checked, 'Universal' selected, and other device options (iPhone, iPad, Mac, Apple TV, Apple Watch) unchecked.

Scaled down for 1x and 2x displays

Asset Scaling and Memory



1x
9Kb



3x
943Kb

Memory Pressure

Temporary Memory Spike

Memory Utilization

Start 3x Image Load

Image Opened and Scaling to 2x

Scaled, Original Deallocated

Memory Pressure

Temporary Memory Spike

Memory Utilization



Start 3x Image Load

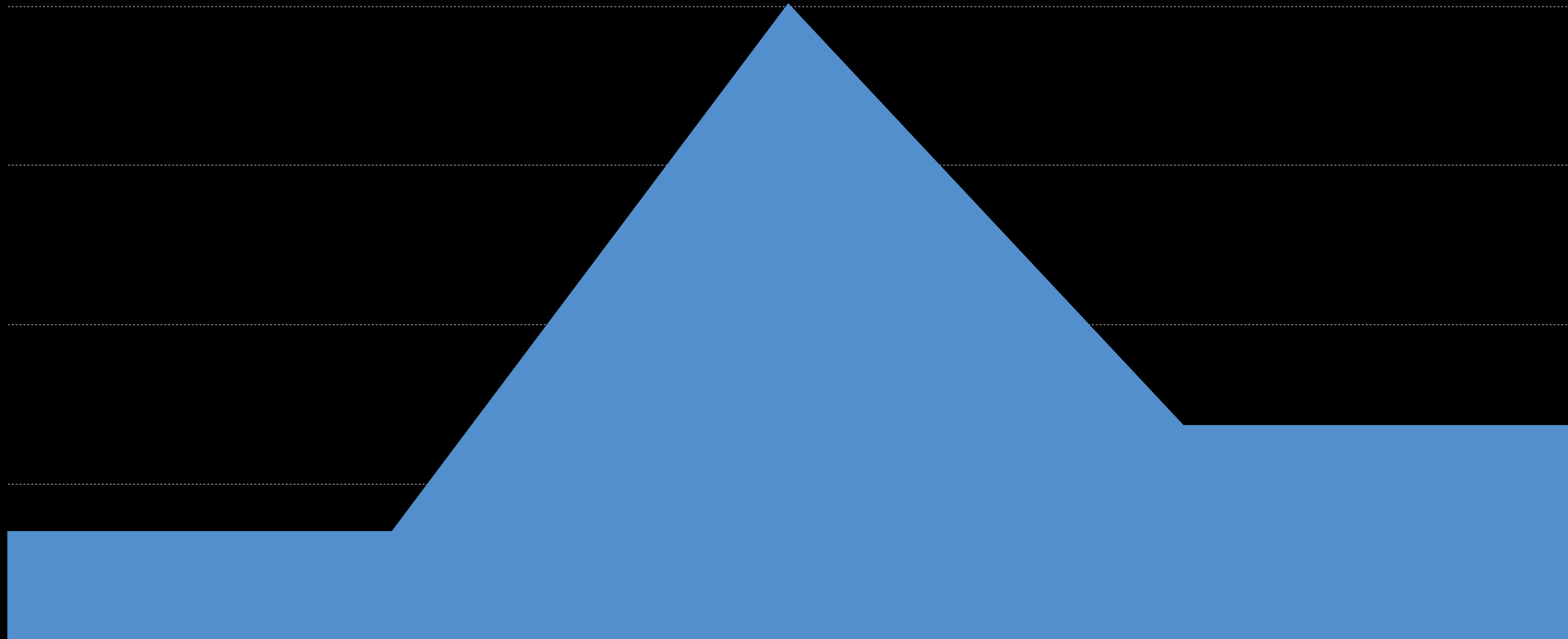
Image Opened and Scaling to 2x

Scaled, Original Deallocated

Memory Pressure

Temporary Memory Spike

Memory Utilization



Start 3x Image Load

Image Opened and Scaling to 2x

Scaled, Original Deallocated

Automator

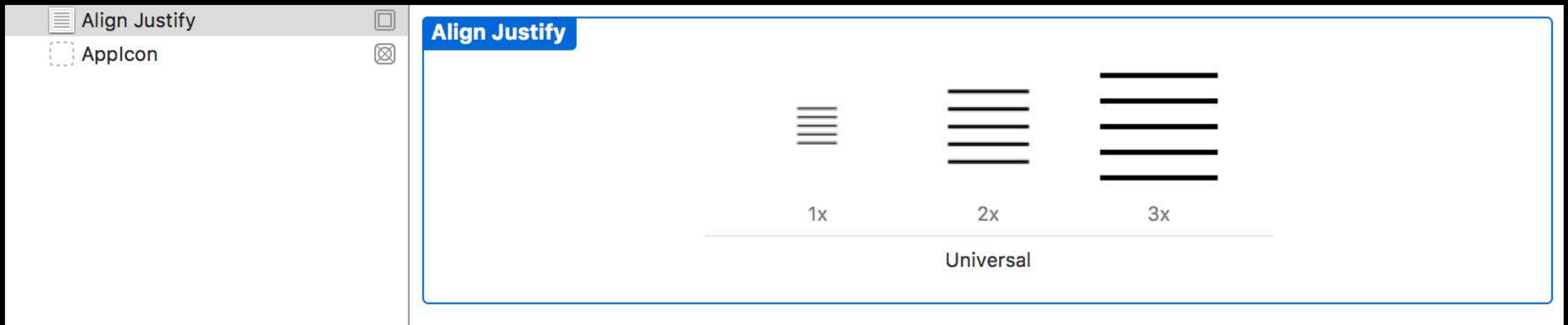
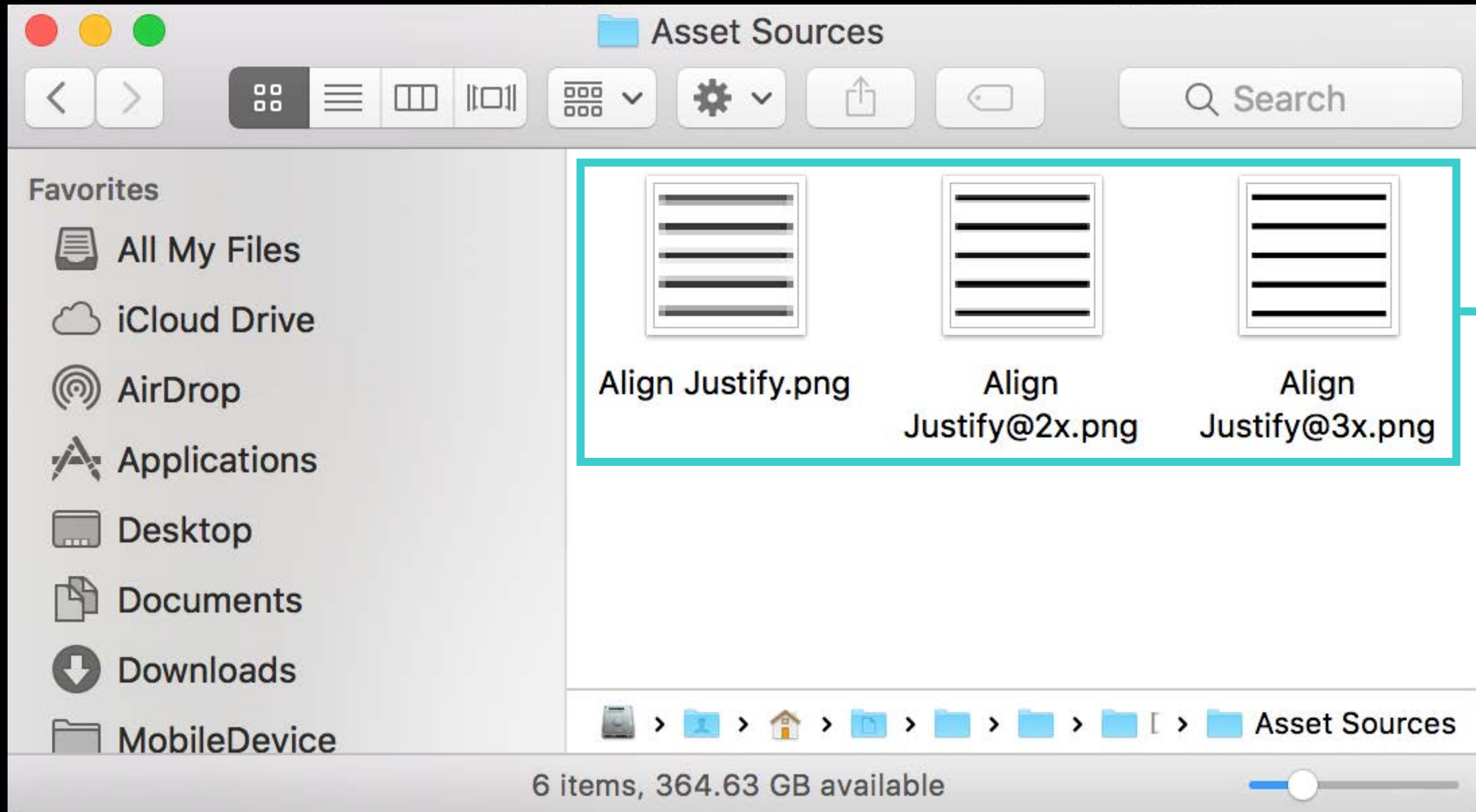
This screenshot shows a vertical stack of five Automator actions:

- Ask for Finder Items**: The first action in the workflow.
- Rename Finder Items: Add Text**: Configured with "Add Text" selected, "Add:" set to "@3x", and "after name" selected. Example: "Item Name@3x.xxx".
- Duplicate Finder Items**: A standard duplicate action.
- Rename Finder Items: Replace Text**: Configured with "Replace Text" selected, "Find:" set to "@3x", "in" set to "basename only", and "Ignore Case" checked. "Replace:" is set to "@2x".
- Rename Finder Items: Replace Text**: Configured with "Replace Text" selected, "Find:" set to "copy", "in" set to "basename only", and "Ignore Case" checked. "Replace:" is set to "with this text".

This screenshot shows a vertical stack of five Automator actions:

- Scale Images**: Configured with "By Percentage" selected and a value of "50".
- Duplicate Finder Items**: A standard duplicate action.
- Rename Finder Items: Replace Text**: Configured with "Replace Text" selected, "Find:" set to "@2x", "in" set to "basename only", and "Ignore Case" checked. "Replace:" is set to "with this text".
- Rename Finder Items: Replace Text**: Configured with "Replace Text" selected, "Find:" set to "copy", "in" set to "basename only", and "Ignore Case" checked. "Replace:" is set to "with this text".
- Scale Images**: Configured with "By Percentage" selected and a value of "50".

A red arrow originates from the top of the left workflow and points to the top of the right workflow, indicating a transition or continuation of the process.



Vector Assets

Scalable to any size



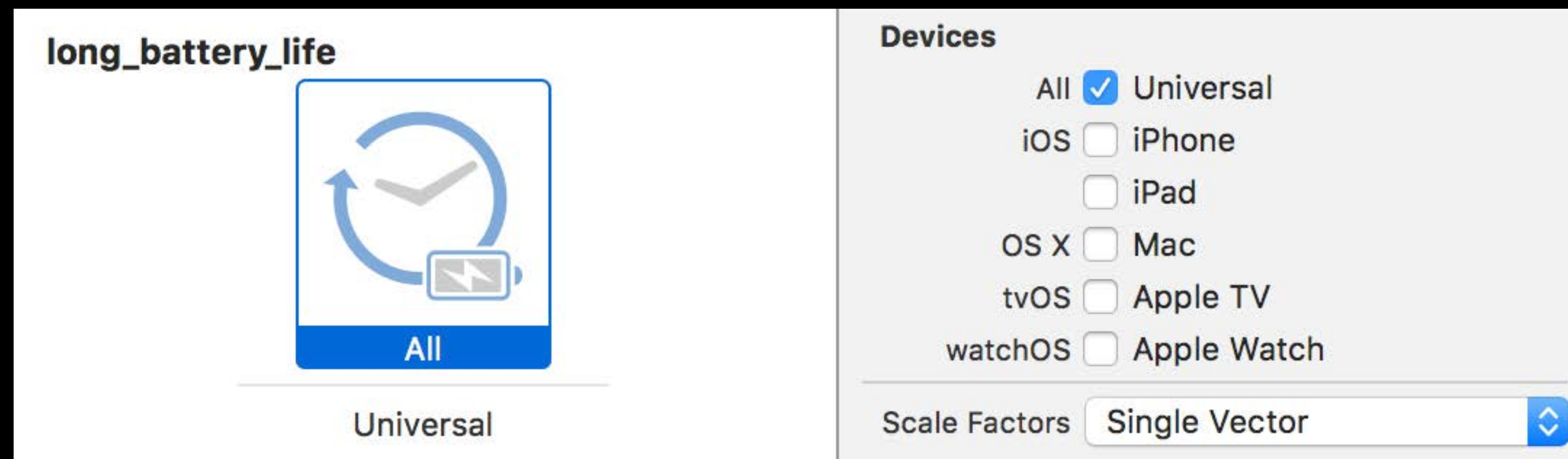
Vector Assets

Scalable to any size



Vector Assets

Scaled and rasterized at build time



Vector Assets

Scaled and rasterized at build time



Override Universal Vector Assets

Improving asset image quality



Override Universal Vector Assets

Improving asset image quality

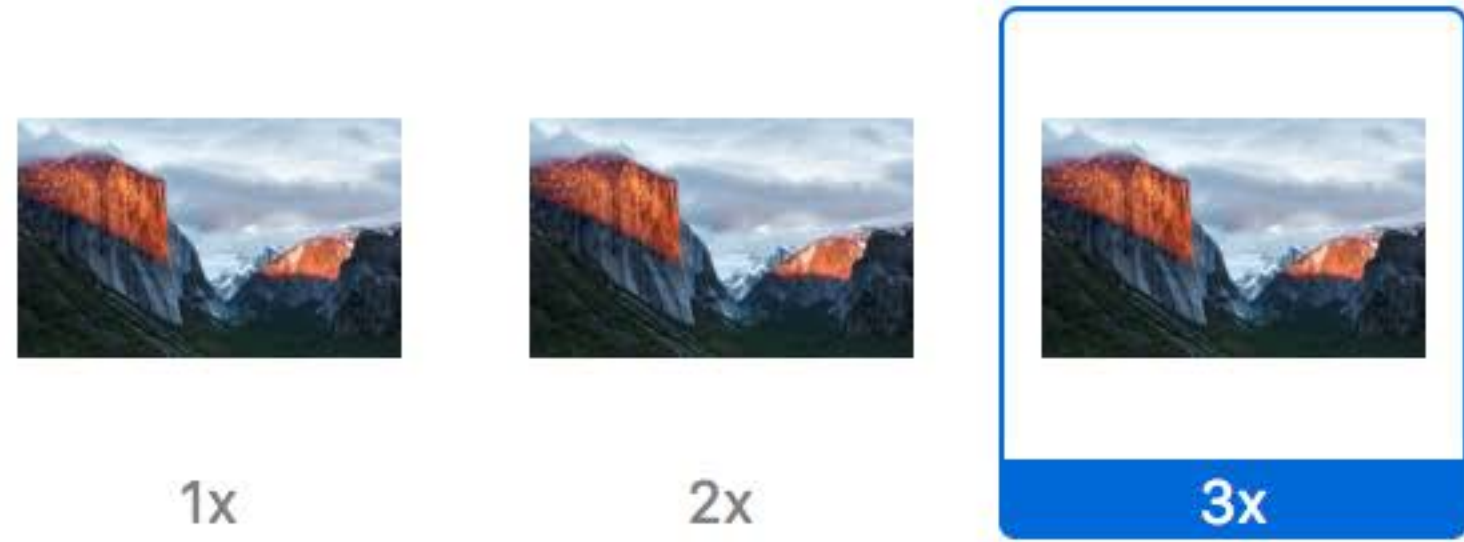


Image Compression

Lossy and lossless

NEW

El Capitan



1x 2x 3x

Universal

Image Set

Lossless (Inherited)

Lossless

Automatic

Lossy

✓ Automatic

Basic


GPU Best Quality

GPU Smallest Size

Devices

iOS iPhone

long_battery_life



All

Universal

Image Set

Lossless (Inherited)

Lossless

✓ Automatic

Lossy

Automatic

Basic

GPU Best Quality

GPU Smallest Size

Devices

iOS iPhone

Capped Images

Challenge of rounded corners



Continue

Capped Images

Challenge of rounded corners



Capped Images

Challenge of rounded corners



Continue

Capped Images

Challenge of rounded corners

Continue

Capped Images

Challenge of rounded corners



Continue

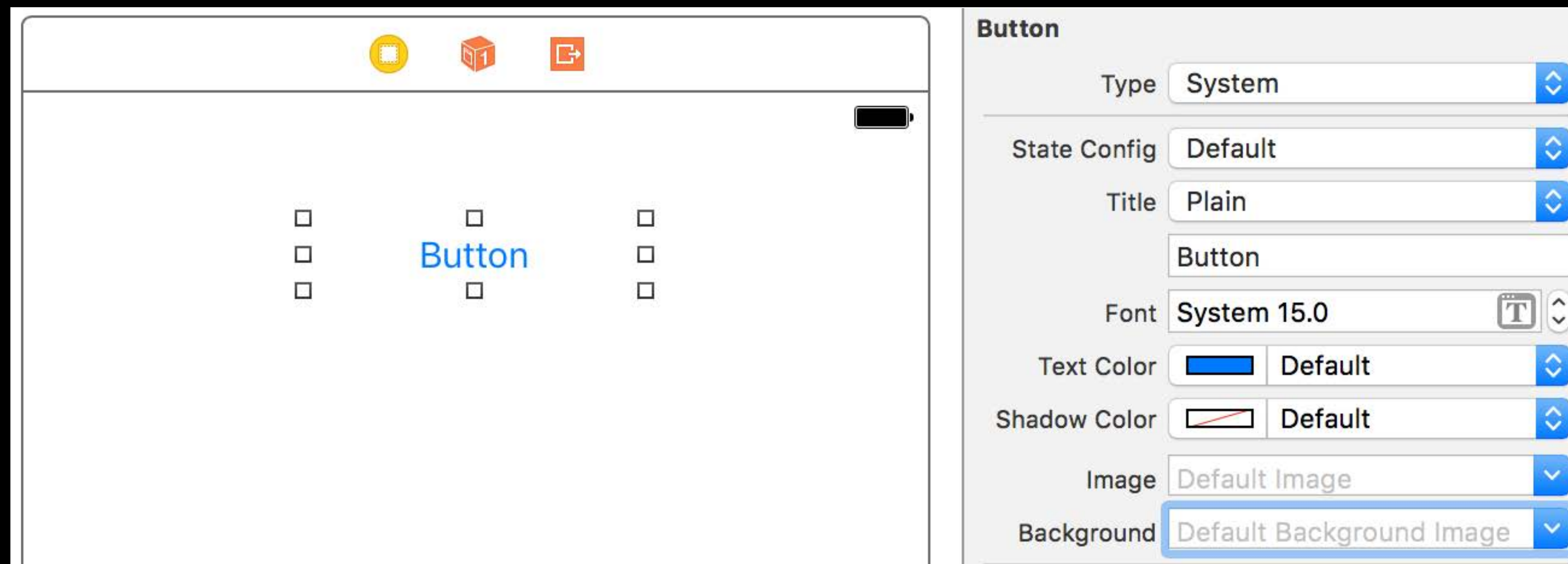
Capped Images

Challenge of rounded corners

Continue

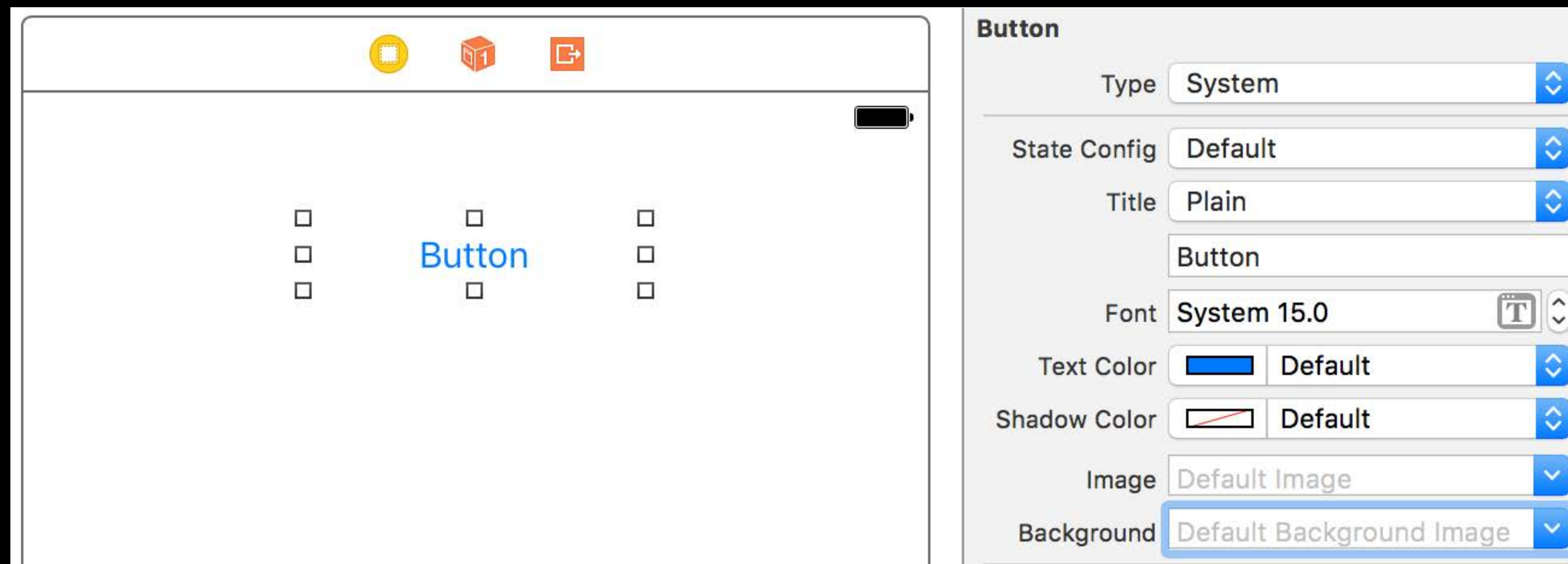
Button Background Image

Interface Builder's Attributes Inspector



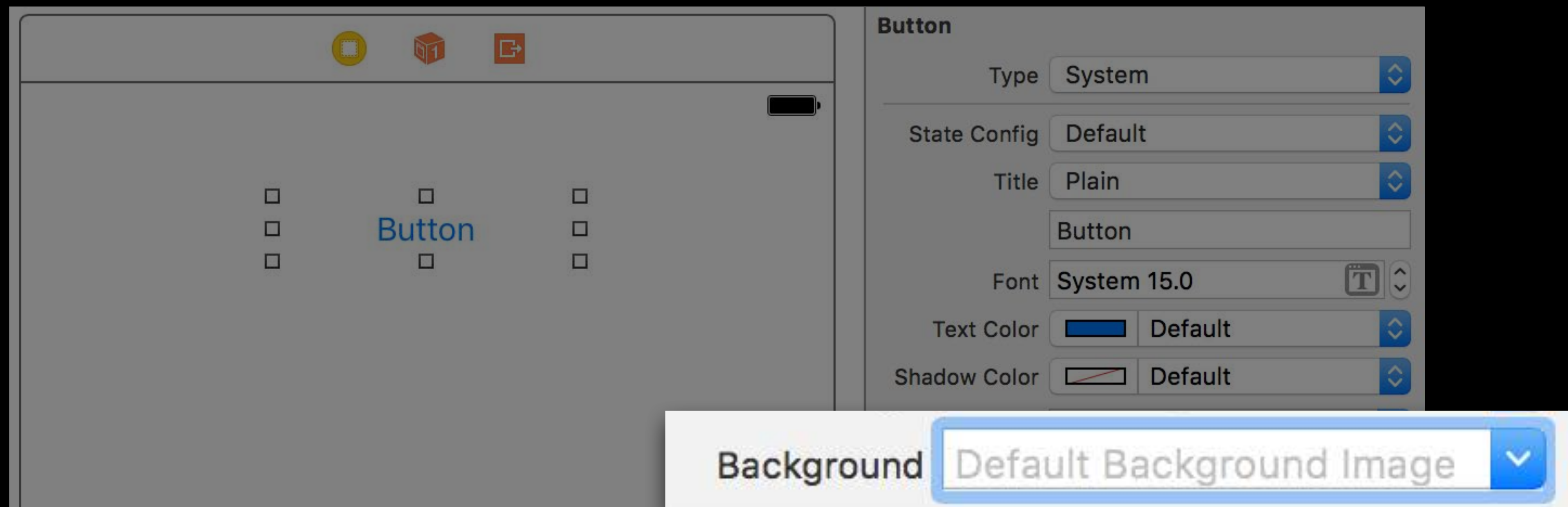
Button Background Image

Interface Builder's Attributes Inspector



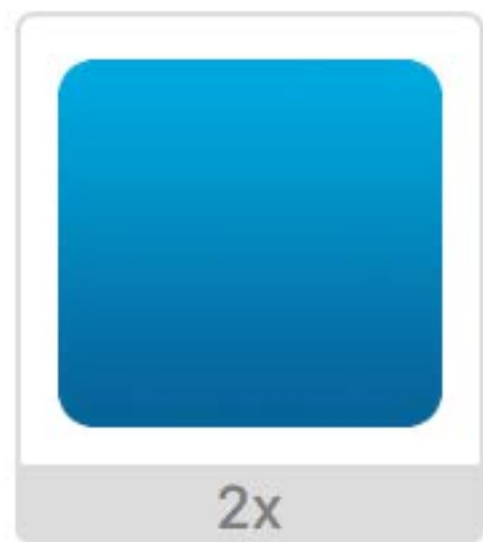
Button Background Image

Interface Builder's Attributes Inspector

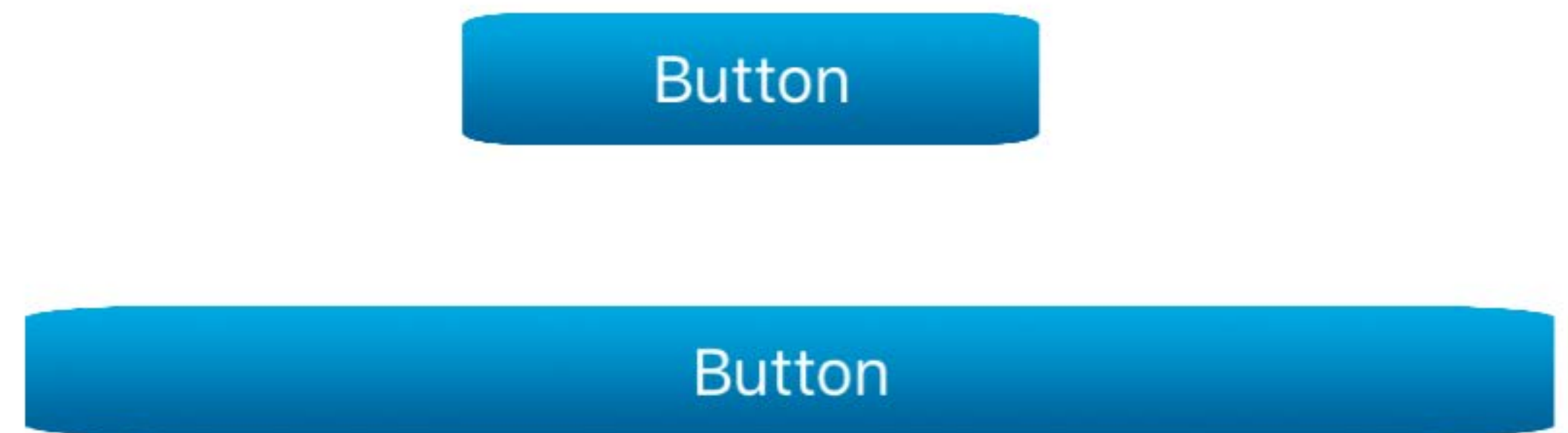
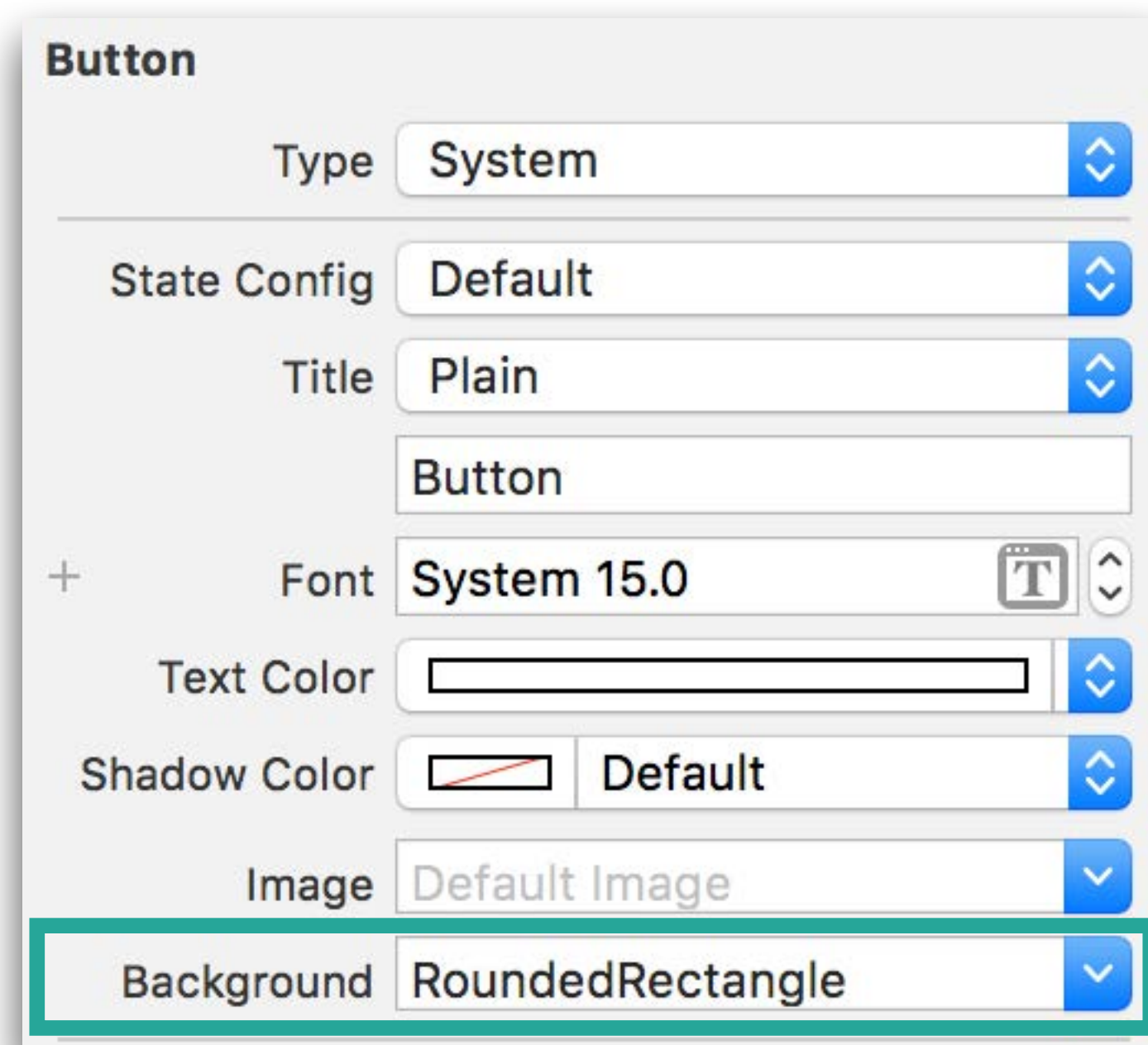


Capped Images

Challenge of rounded corners

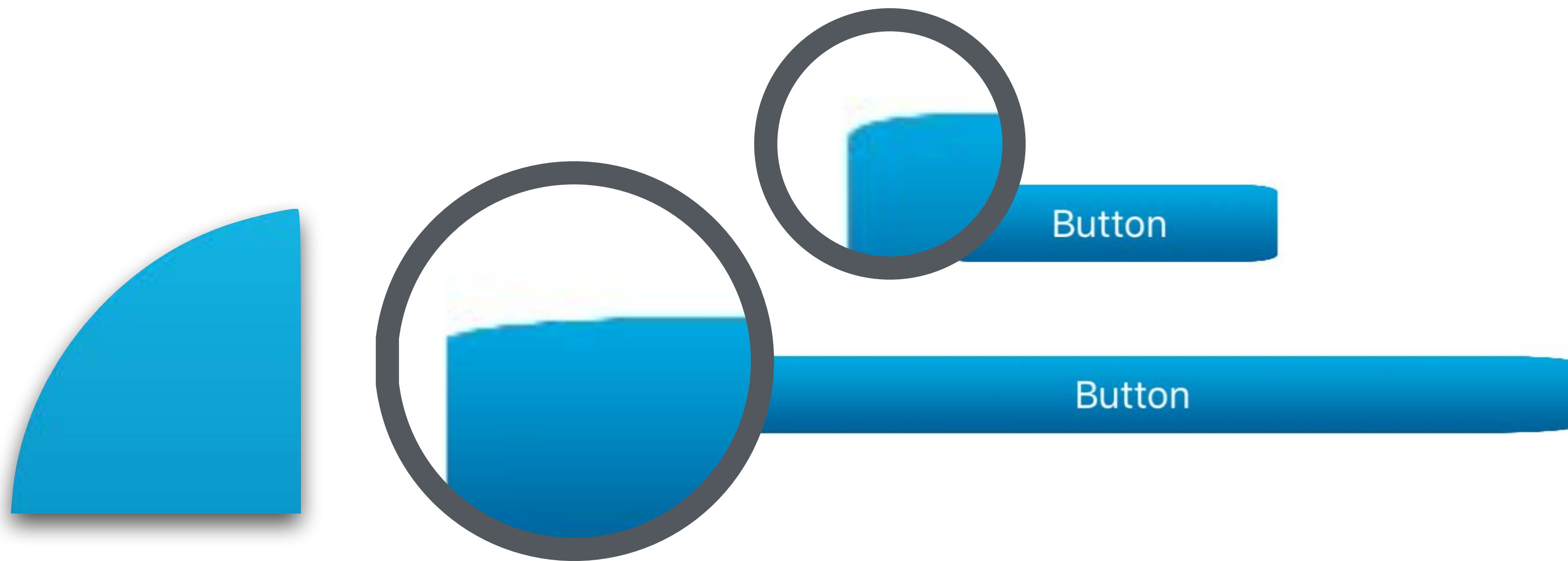


Rounded
Rectangle



Capped Images

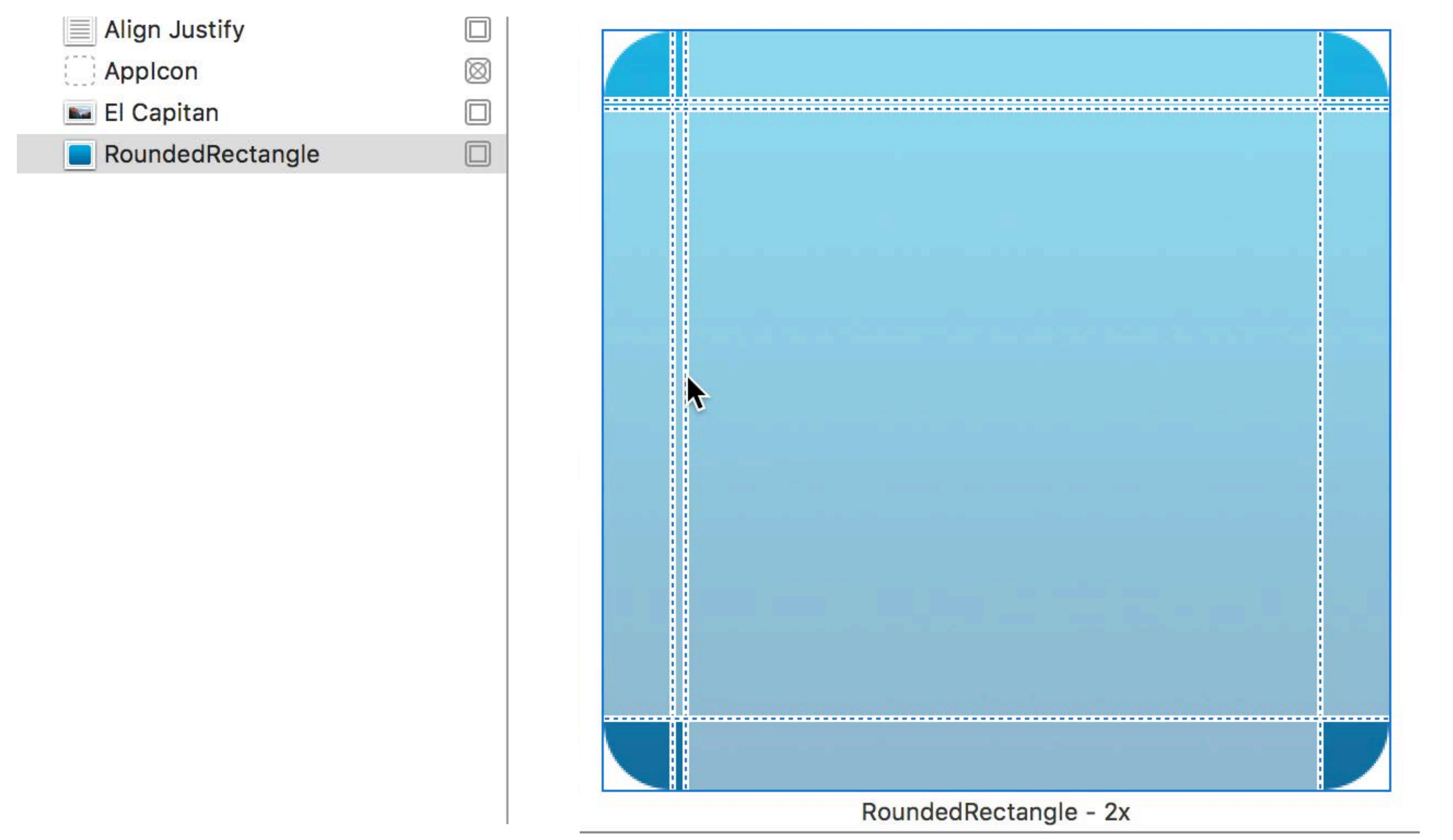
Challenge of rounded corners



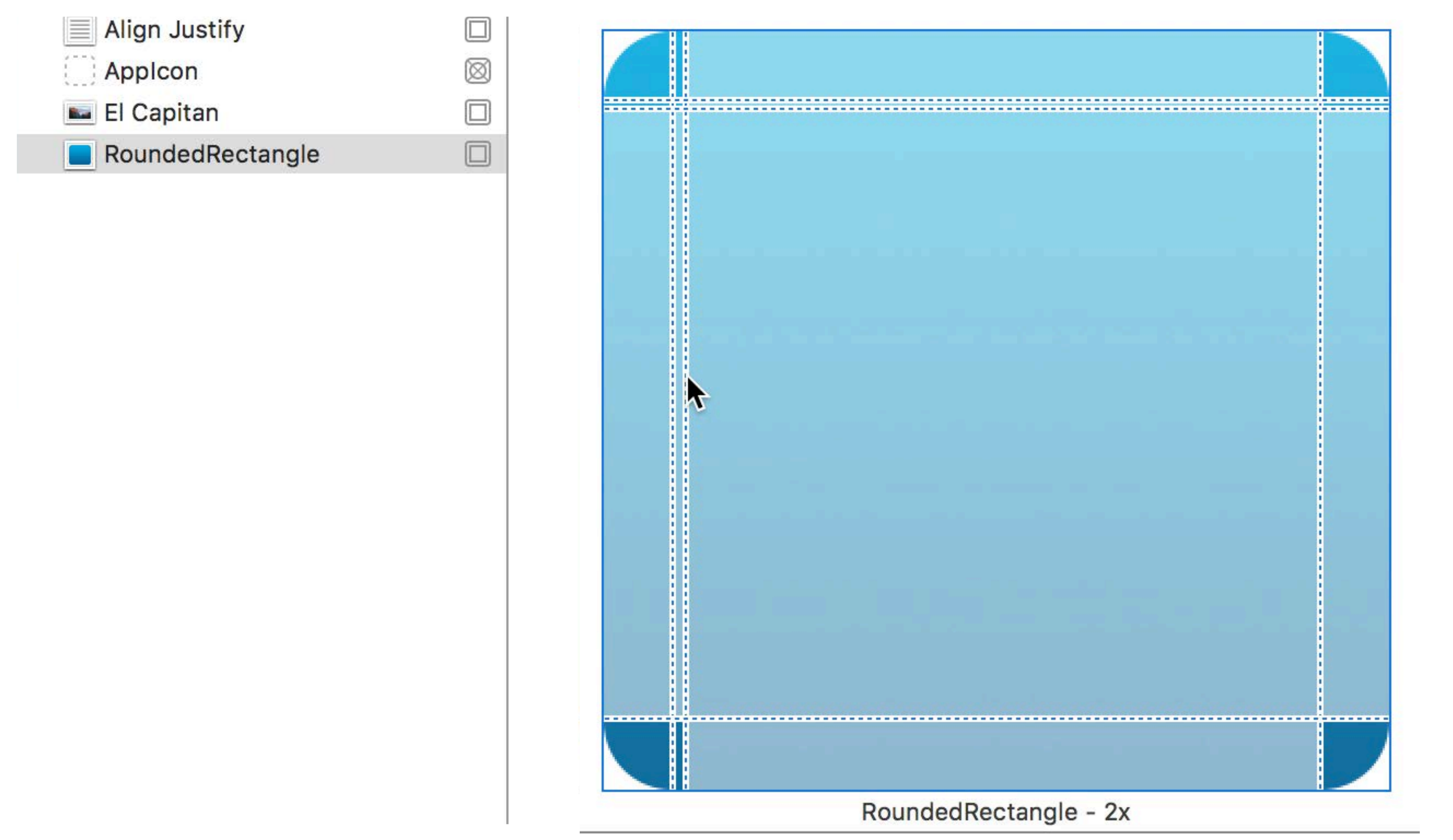
```
// Stretchable Images
// Programmatic, Traditional Way:

if let image = UIImage(named: "RoundedRectangle")
    let background = image.stretchableImage(withLeftCapWidth: 10, topCapHeight: 10) {
    button.setBackgroundImage(background, for: [])
}
}
```

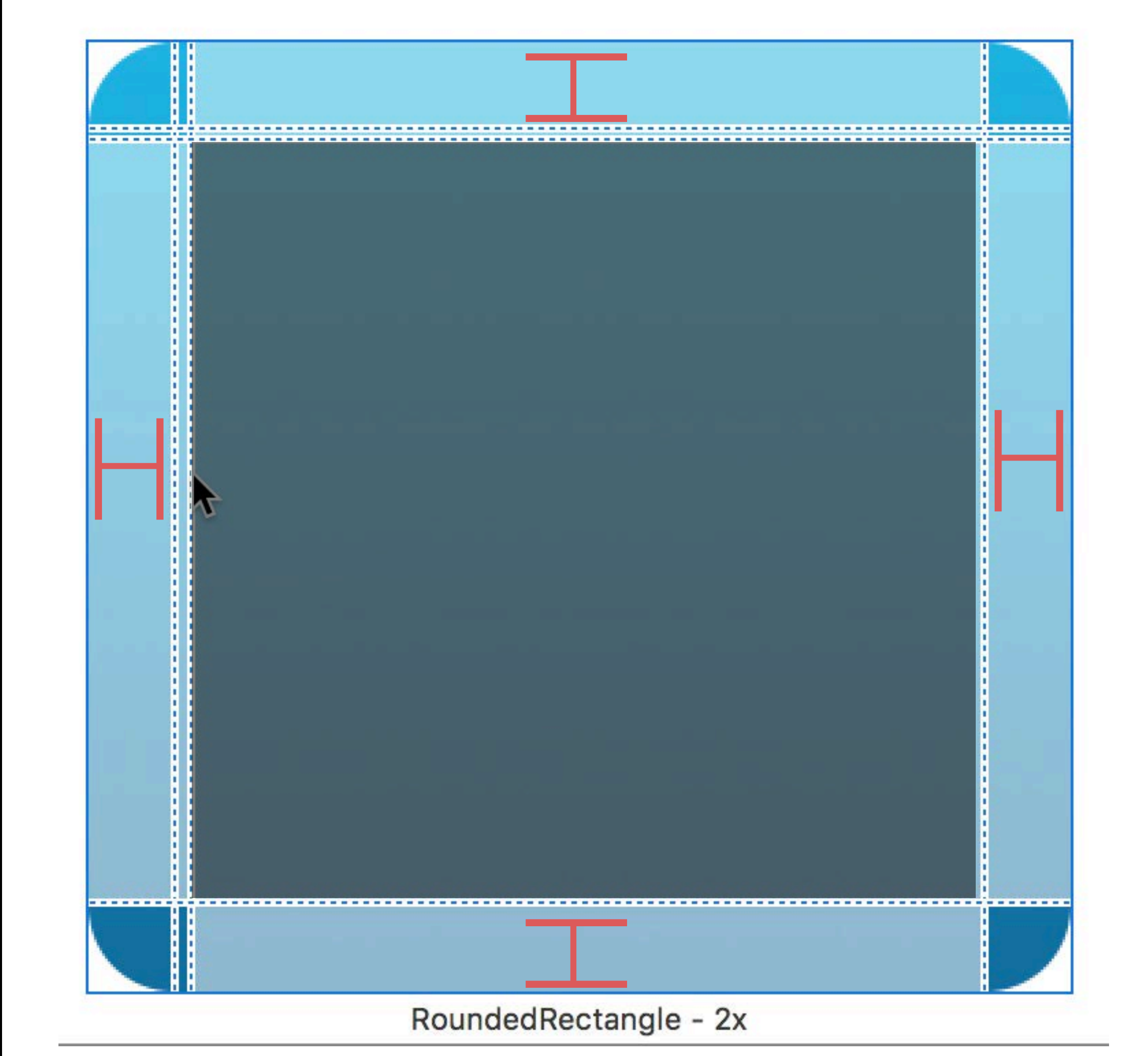

Asset Slicing



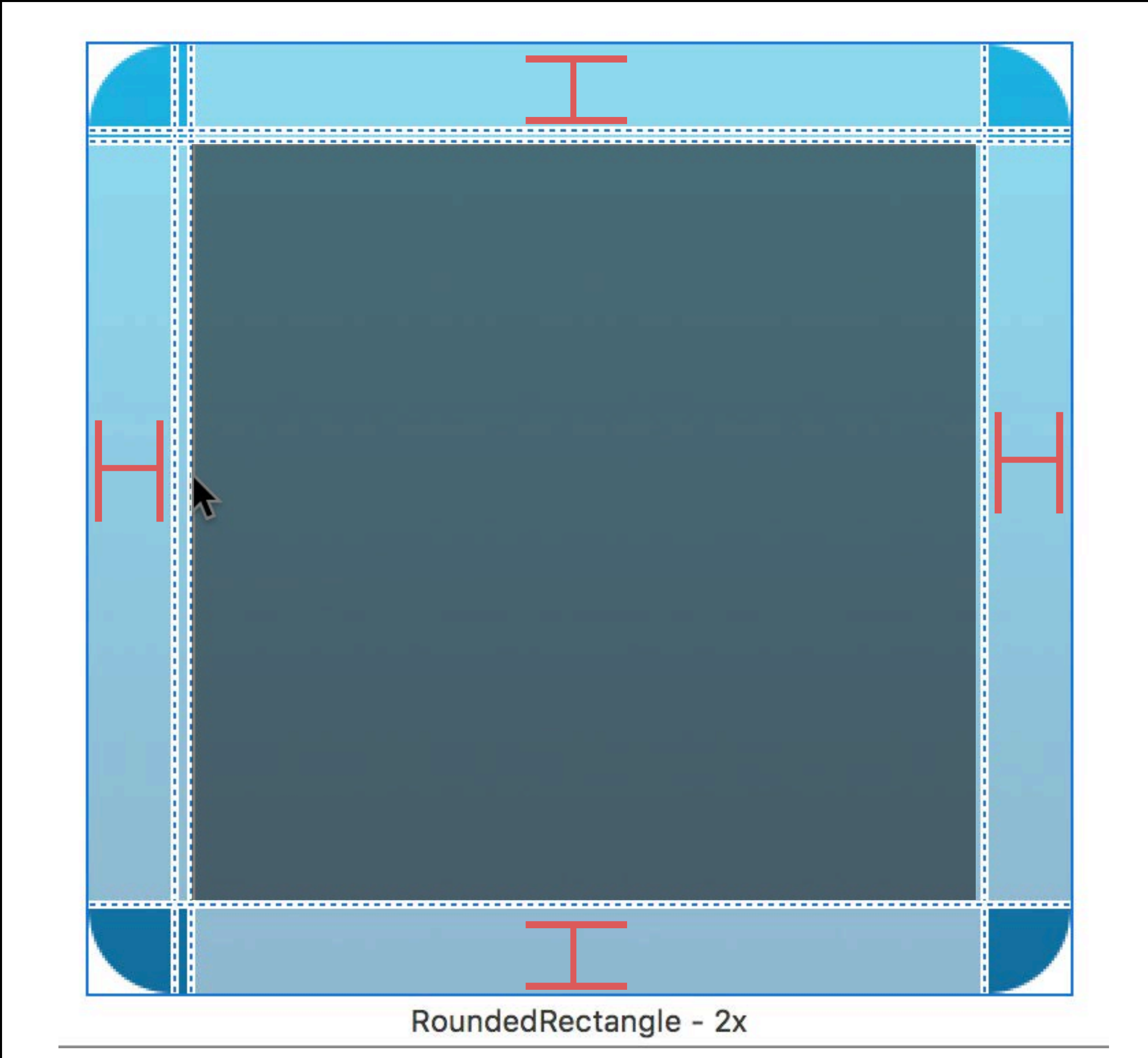
Asset Slicing



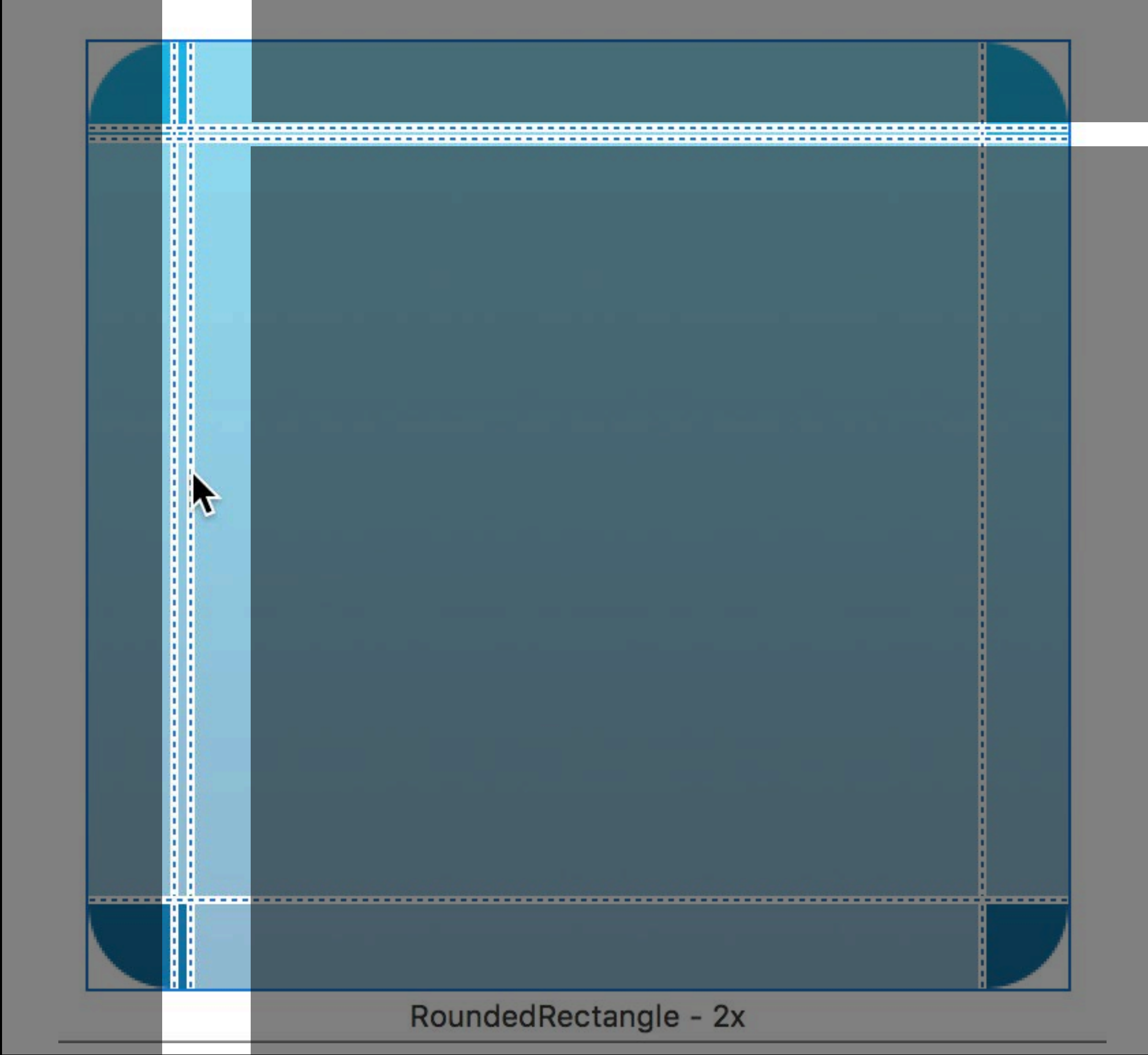
Don't stretch these edges



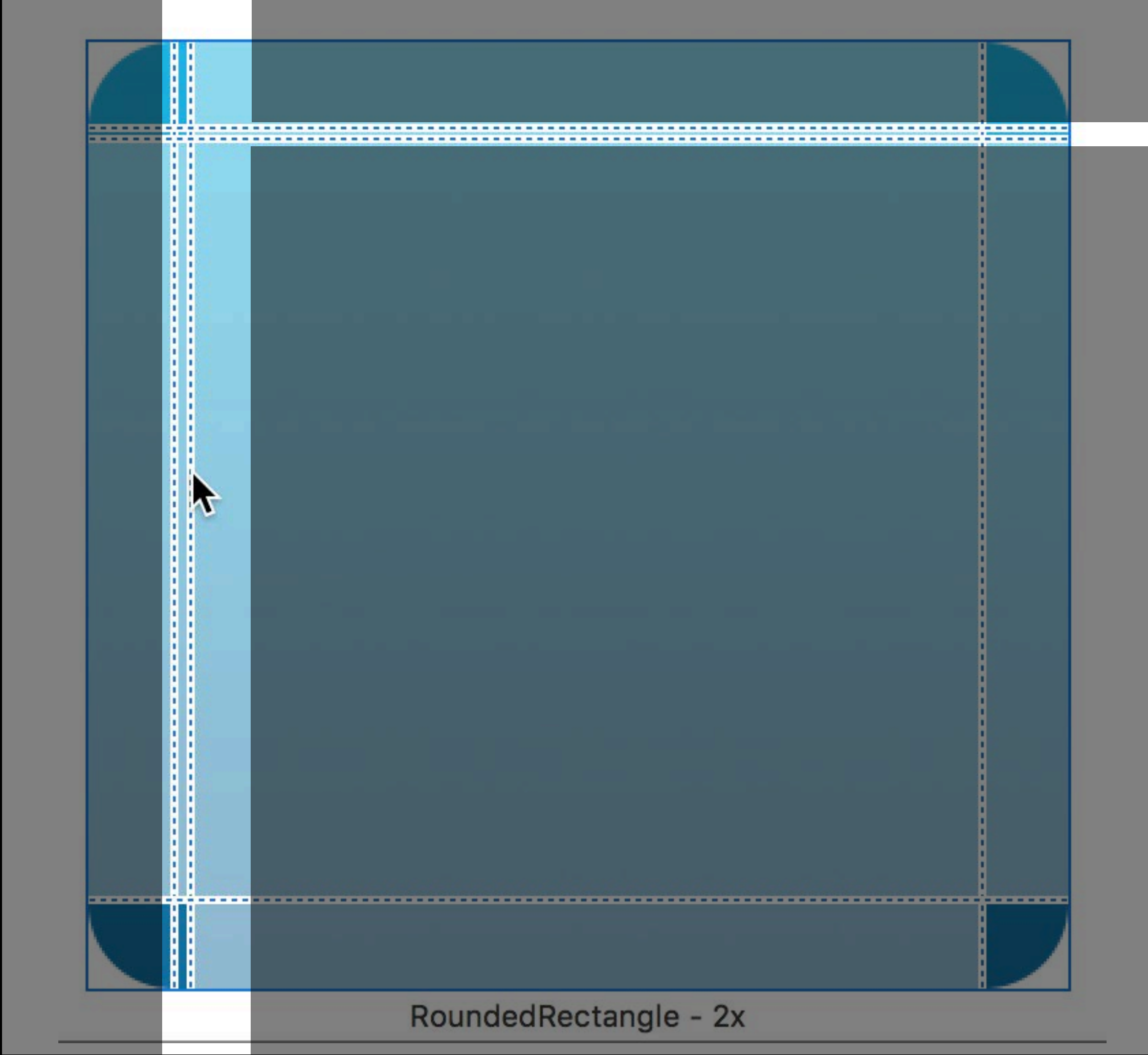
Don't stretch these edges



Repeat these pixels

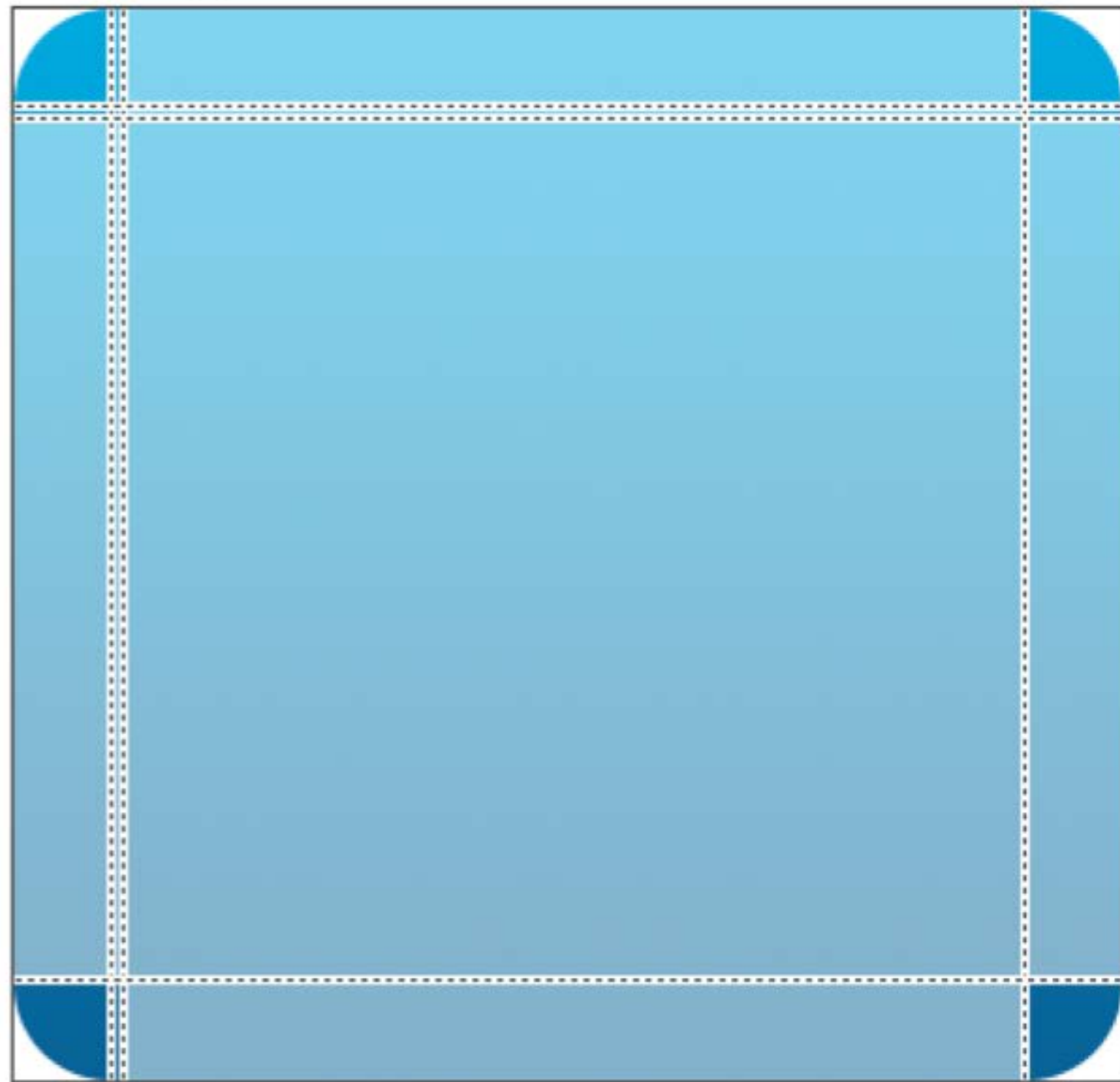


Repeat these pixels



Capped Images

Preserving of rounded corners



RoundedRectangle - 2x



Summary

Asset Catalogs

Use Asset Catalogs

Vector images preserve image quality

Scaled images should be provided for each asset representation (1x, 2x, 3x)

Perform slicing within the asset catalog

Dependency Injection

Pay it forward

UITextFieldDelegate

Methods relating to UITextField

UITextField

UITextFieldDelegate

```
func textFieldShouldBeginEditing(_ textField:  
    UITextField) -> Bool
```

```
func textFieldDidBeginEditing(_ textField: UITextField)
```

WCSessionDelegate

Methods relating to WCSession

WCSession

WCSessionDelegate

```
func session(_ session: WCSession,  
            activationDidCompleteWith activationState:  
            WCSessionActivationState, error: NSError?)
```

App Delegate

Everything else



NS/UIApplication

NS/ApplicationDelegate



App Delegate

Everything else



NS/UIApplication

NS/ApplicationDelegate

```
func applicationWillResignActive(_ application: UIApplication)
```

App Delegate

Everything else



NS/UIApplication

NS/ApplicationDelegate

```
func applicationWillResignActive(_ application: UIApplication)
var sqlite3DB: SQLite3Db?
```

App Delegate

Everything else



NS/UIApplication

NS/ApplicationDelegate

```
func applicationWillResignActive(_ application: UIApplication)
var sqlite3DB: SQLite3Db?
var sink = Kitchen(garbageDisposal: true)
```


Accessing the App Delegate

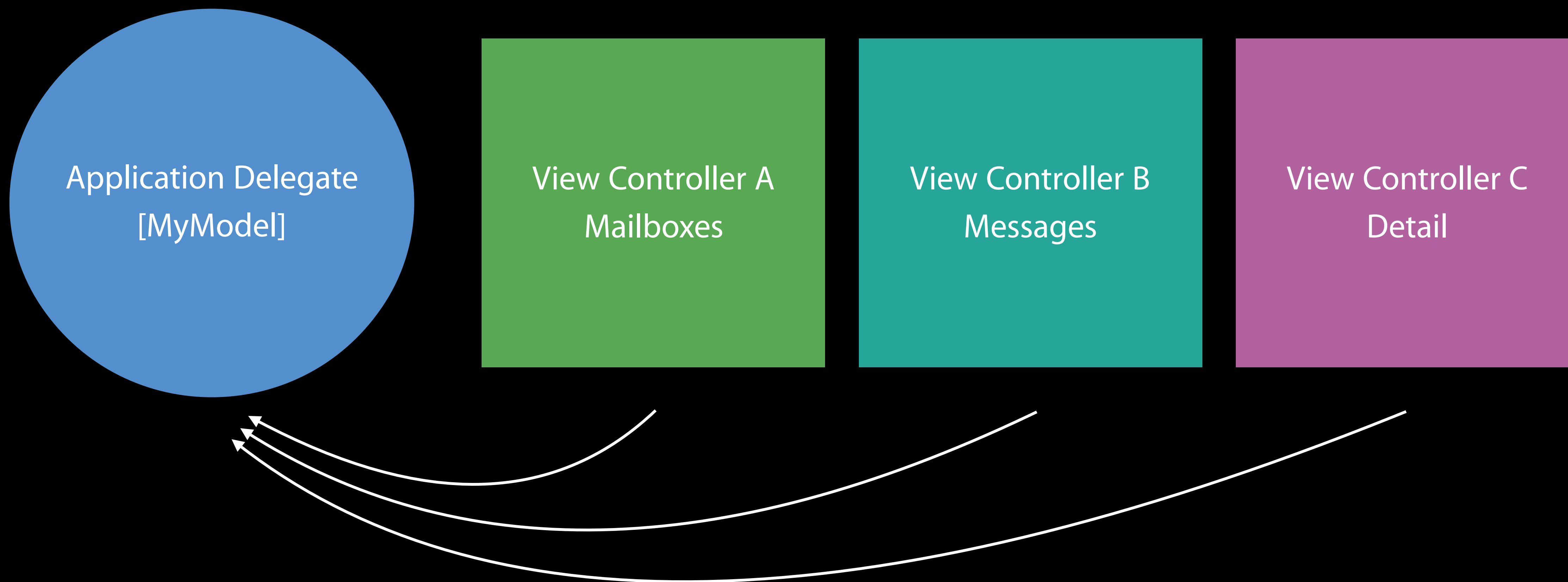
Convenient, but strongly coupled



```
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {  
    return (UIApplication.shared().delegate as? AppDelegate)?.content?.count ?? 0  
}
```

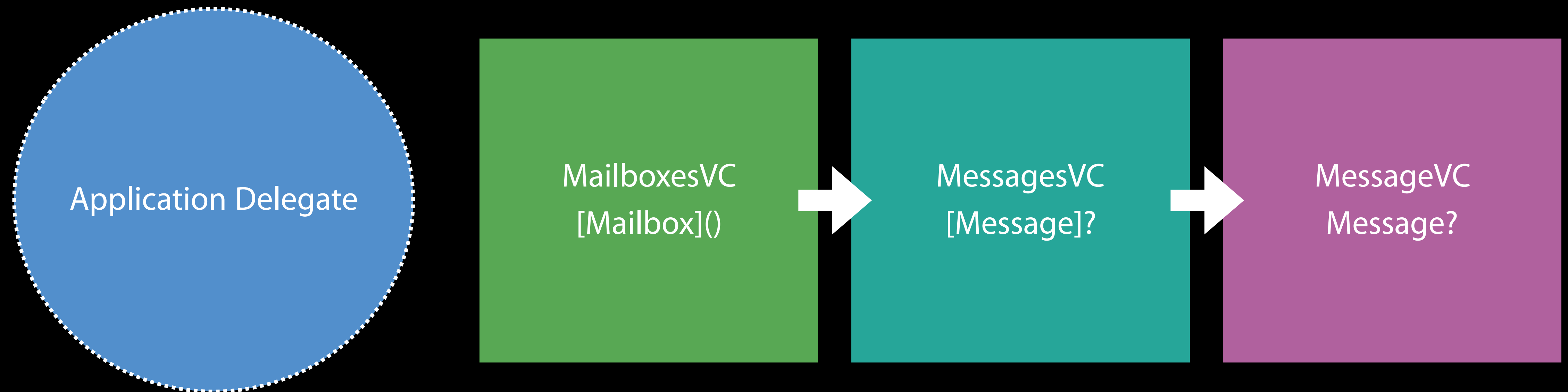
Reach-Back to the App Delegate

Tight coupling



Dependency Injection

Pay it forward



Dependency Injection

Storyboard Segues



```
override func prepare(for segue: UIStoryboardSegue, sender: AnyObject?) {  
    if let messagesVC = segue.destinationViewController as? MessagesViewController {  
        messagesVC.content = model  
    }  
}
```

Dependency Injection

Programatic presentation



```
@IBAction func showDetailVC(sender: AnyObject?) {  
    let detailVC = DetailVC()  
    let indexOfSelectedMessage = 0 // ...  
    let message = messages[indexOfSelectedMessage]  
    detailVC.message = message  
}
```

Returning from the View Controller

Coming back from Dependency Injection

Create and implement a Protocol

- UIImagePickerControllerControllerDelegate
- MFMailComposeViewControllerDelegate
- Mark the delegate property as weak to prevent memory leaks to prevent circular references

Pass a Closure to the Destination View Controller

Pass Model Objects by Reference

Unwind/Exit Segues, with `prepareForSegue:`

Previously at WWDC

Flashback to 2015

Once upon a time, in a session room not that far away...

WWDC2015

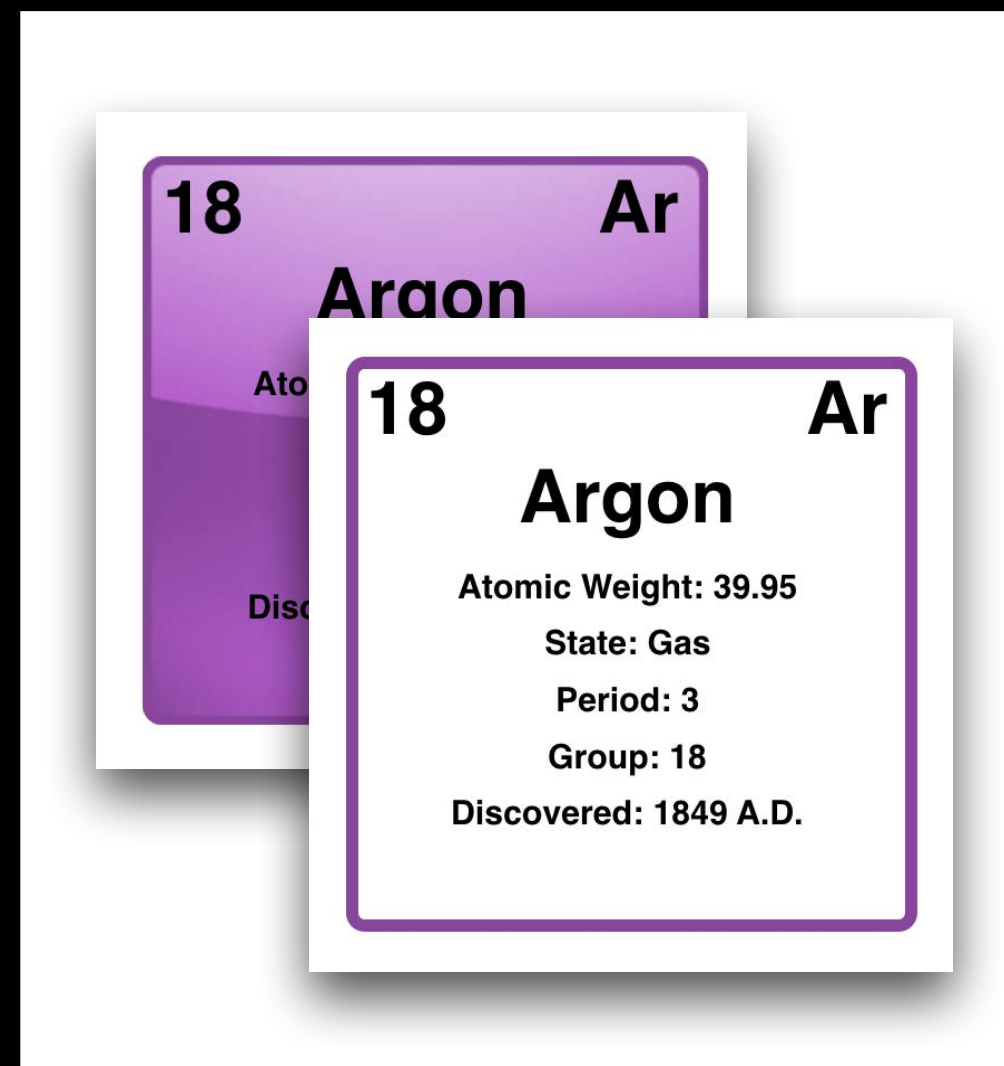
Modernizing existing apps with Swift

WWDC2015

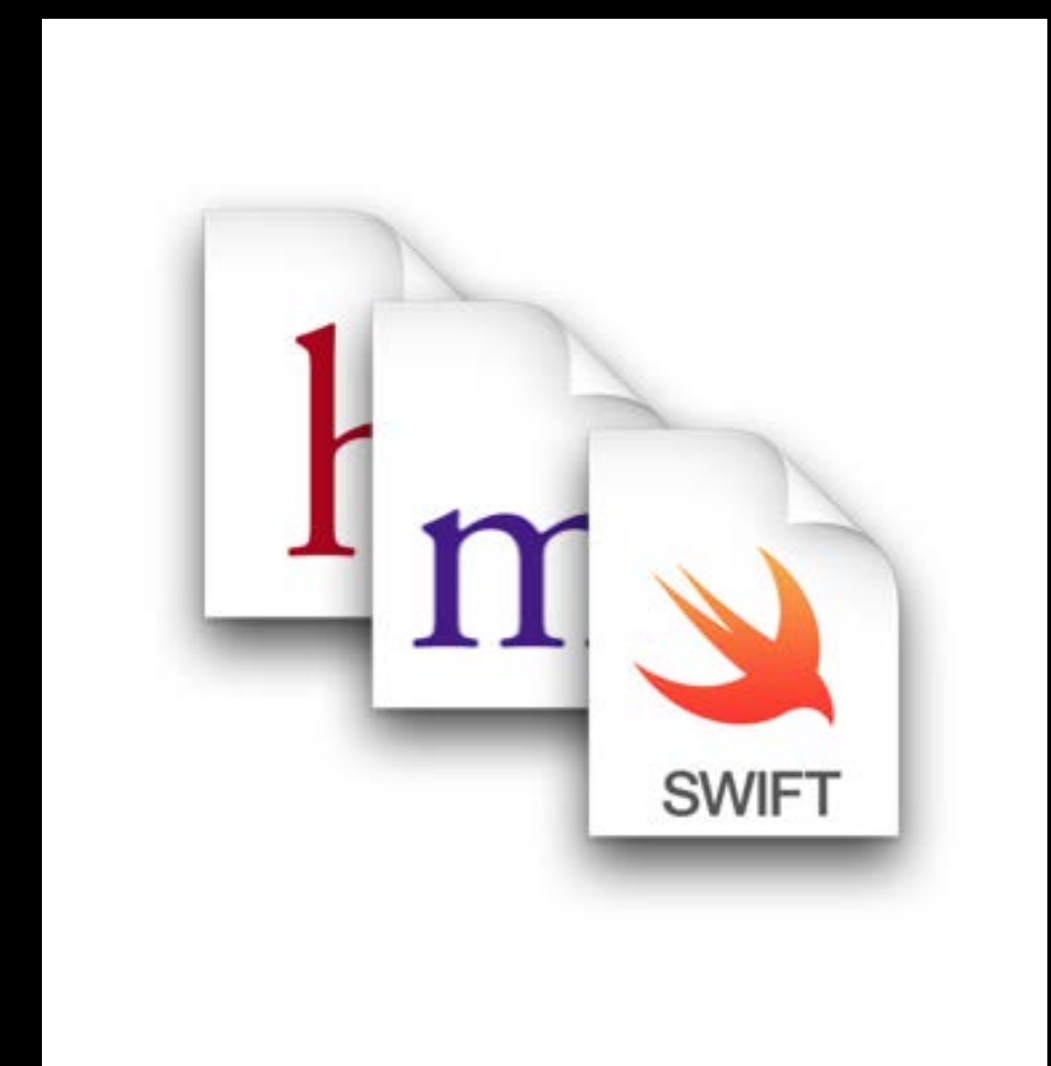
Modernizing existing apps with Swift



Playgrounds



Modern UI



Interoperability

Live Playgrounds!

Better than `!(Live Playgrounds)`

Demo

TableViews in Live Playgrounds

The Needle? We like to move it, move it.

Demo Summary

Live Playground supporting API

Import the Playground Module

```
import PlaygroundSupport
```

Indefinite/Asynchronous Execution

```
PlaygroundPage.current.needsIndefiniteExecution = true
```

Assign a Live View

```
PlaygroundPage.current.liveView = customTableViewController.tableView
```


Demo Summary

Tips

Playground Sources and Resources

- Files dragged to these folders are copied, not referenced
- Methods, properties, and data types in the Sources folder must use the Public specifier

Demo Summary

Caches folder

Temporary data that can be recreated if it's missing

Thumbnails

Downloaded resources

Purged when device under storage pressure

```
let fm = FileManager.default()
let cacheUrl = fm.urlsForDirectory(.cachesDirectory, inDomains: .userDomainMask).last
```

NSURLSession

Powerful and performant

Out-of-process networking

Delegation and completion handlers

Supported by tvOS, iOS, macOS, and watchOS platforms

Backed by the expertise of our core networking engineers

NSURLSession: New Features and Best Practices

Pacific Heights

Thursday 10:00AM

Networking for the Modern Internet

Pacific Heights

Thursday 3:00PM

Must See  16

It's time to play the music, it's time to light the lights

```
// tvOS Conditional Compilation
```

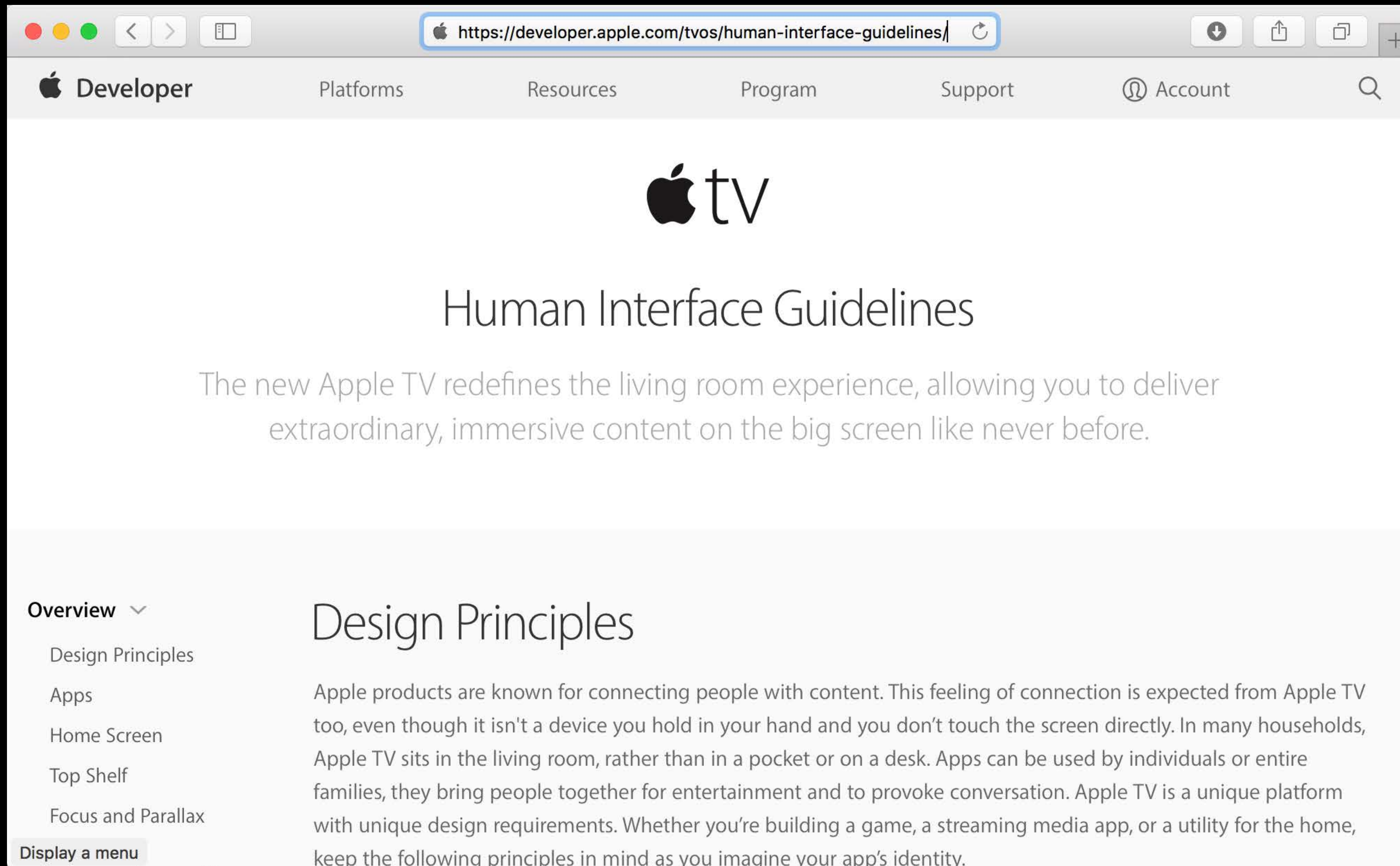
```
#if os(tvOS)
```

```
    backgroundImageView.adjustsImageWhenAncestorFocused = true
```

```
#endif
```

Adopting Additional Platforms

Human Interface Guidelines



The screenshot shows a web browser window with the URL `https://developer.apple.com/tvos/human-interface-guidelines/`. The page features the Apple logo and the text "Developer" in the top left, and navigation links for "Platforms", "Resources", "Program", "Support", and "Account" in the top right. The main content area displays the "Apple tv" logo, followed by the title "Human Interface Guidelines" and a descriptive paragraph: "The new Apple TV redefines the living room experience, allowing you to deliver extraordinary, immersive content on the big screen like never before." A left-hand navigation menu is visible, with "Overview" expanded to show "Design Principles", "Apps", "Home Screen", "Top Shelf", and "Focus and Parallax". The "Design Principles" section is currently selected and expanded, showing the text: "Apple products are known for connecting people with content. This feeling of connection is expected from Apple TV too, even though it isn't a device you hold in your hand and you don't touch the screen directly. In many households, Apple TV sits in the living room, rather than in a pocket or on a desk. Apps can be used by individuals or entire families, they bring people together for entertainment and to provoke conversation. Apple TV is a unique platform with unique design requirements. Whether you're building a game, a streaming media app, or a utility for the home, keep the following principles in mind as you imagine your app's identity."

Developer

Platforms Resources Program Support Account

Apple tv

Human Interface Guidelines

The new Apple TV redefines the living room experience, allowing you to deliver extraordinary, immersive content on the big screen like never before.

Overview ▾

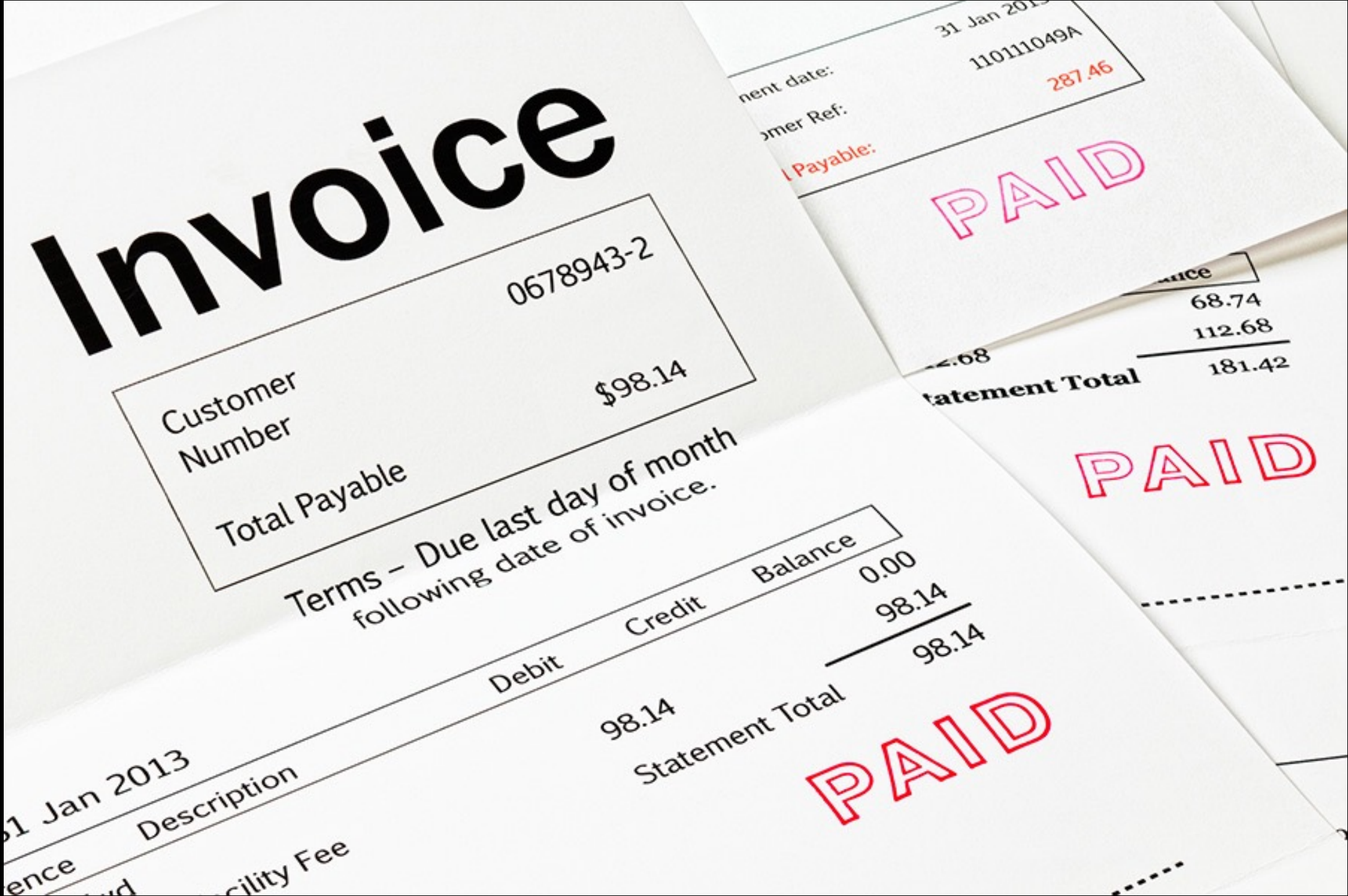
- Design Principles
- Apps
- Home Screen
- Top Shelf
- Focus and Parallax

Display a menu

Design Principles

Apple products are known for connecting people with content. This feeling of connection is expected from Apple TV too, even though it isn't a device you hold in your hand and you don't touch the screen directly. In many households, Apple TV sits in the living room, rather than in a pocket or on a desk. Apps can be used by individuals or entire families, they bring people together for entertainment and to provoke conversation. Apple TV is a unique platform with unique design requirements. Whether you're building a game, a streaming media app, or a utility for the home, keep the following principles in mind as you imagine your app's identity.

Invoicing



Financial Charts



Adding Platforms

Read and use our Human Interface Guidelines

Design to the platform

Pivot your data model

Summary

Modernizing your app is an on-going process

Rely on the frameworks

Start today—No need to wait for Xcode 8

Live Playgrounds allow for more experimentation

Architect your app with few inter-object dependencies

Consider bringing your model layer to our other platforms, with platform-specific UI

Related Sessions

What's New in Cocoa Touch	Presidio	Tuesday 1:40PM
What's New in tvOS	Presidio	Tuesday 3:00PM
What's New in watchOS 3	Presidio	Tuesday 5:00PM
What's New in Swift	Presidio	Tuesday 9:00AM
Crafting Modern Cocoa Apps	Pacific Heights	Friday 5:00PM
Increase Usage of Your App with Proactive Suggestions	Mission	Friday 1:40PM

Labs

Cocoa Touch Lab	Frameworks Lab A	Wednesday 3:00PM
Cocoa Lab	Frameworks Lab D	Thursday 2:00PM
Interface Builder and Auto Layout Lab	Developer Tools Lab C	Friday 9:00AM
UIKit and UIKit Animations Lab	Frameworks Lab C	Thursday 1:00PM
Swift Open Hours	Developer Tools Lab A	Daily

More Information

<https://developer.apple.com/wwdc16/213>



W

W

D

C

1

6