

# Localization Best Practices on tvOS

Session 248

Joaquim Lobo Silva, Internationalization Software Engineer

# Language

English

Español

简体中文

繁體中文

日本語

Français

Deutsch

Русский

Português

Italiano



Text

Layout and images

Exporting and testing

**Text**

Layout and images

Exporting and testing

**Text**

# Text

Strings in Storyboards are localizable by default

# Text

Strings in Storyboards are localizable by default

In code, wrap strings with `NSLocalizedString()` calls

# Text

Strings in Storyboards are localizable by default

In code, wrap strings with `NSLocalizedString()` calls

Comments are extremely useful



# Text

Strings in Storyboards are localizable by default

In code, wrap strings with `NSLocalizedString()` calls

Comments are extremely useful

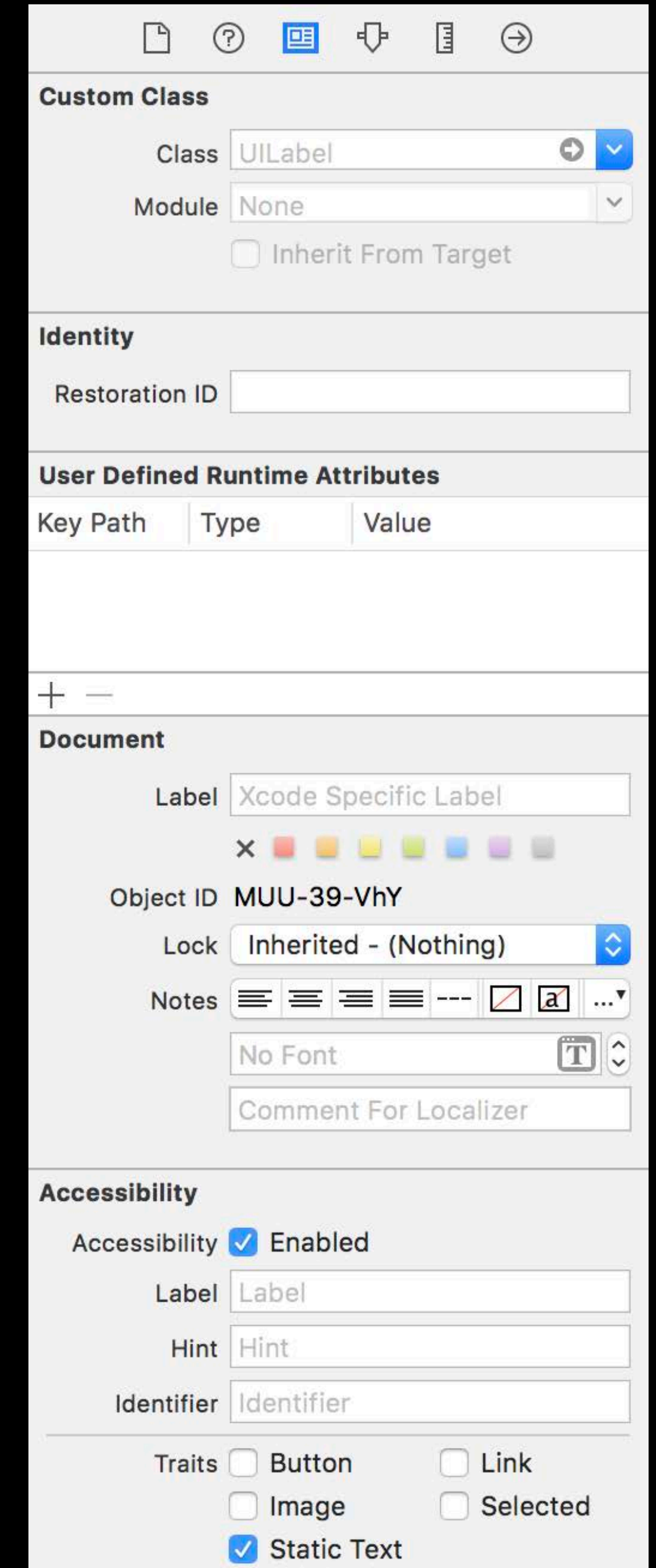
```
let instruction = NSLocalizedString("Subscribe", comment: "Button title, start subscription")
```

# Text

Strings in Storyboards are localizable by default

In code, wrap strings with NSLocalizedString() calls

Comments are extremely useful

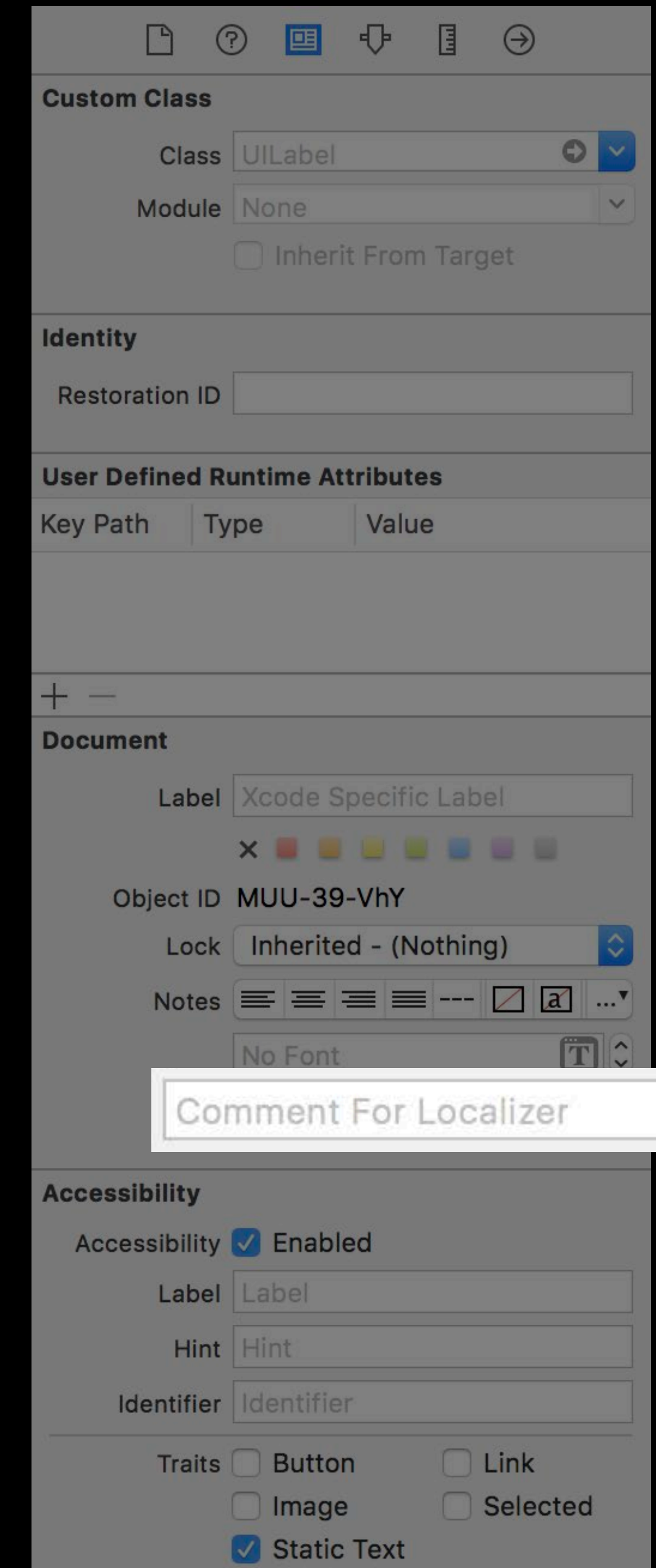


# Text

Strings in Storyboards are localizable by default

In code, wrap strings with NSLocalizedString() calls

Comments are extremely useful



**Text**

# Text

Not everything needs to be a localized string literal

# Text

Not everything needs to be a localized string literal

Formatters for numbers, dates and date intervals, units, etc.

# Text

Not everything needs to be a localized string literal

Formatters for numbers, dates and date intervals, units, etc.

```
let formatter = DateFormatter()
```

# Text

Not everything needs to be a localized string literal

Formatters for numbers, dates and date intervals, units, etc.

```
let formatter = DateFormatter()  
  
formatter.timeStyle = .short  
let string = formatter.string(from: date) // 9:41 AM
```



# Text

Not everything needs to be a localized string literal

Formatters for numbers, dates and date intervals, units, etc.

```
let formatter = DateFormatter()  
  
formatter.timeStyle = .short  
let string = formatter.string(from: date) // 9:41 AM
```

---

Measurements and Units

WWDC 2016

---

Internationalization Best Practices

WWDC 2016

---

# Remote Content

# Remote Content

Remote text should match your app's language

# Remote Content

Remote text should match your app's language

- e.g. media content and descriptions

# Remote Content

Remote text should match your app's language

- e.g. media content and descriptions

APIs to determine best language to display

# Remote Content

Remote text should match your app's language

- e.g. media content and descriptions

APIs to determine best language to display

Considers regional variants and appropriate fallbacks

# Remote Content

Remote text should match your app's language

- e.g. media content and descriptions

APIs to determine best language to display

Considers regional variants and appropriate fallbacks

`Locale.preferredLanguages`

`bundle.localizations`

`bundle.preferredLocalizations`

---

es-MX

en, en-GB, es-ES, es-419, zh-Hans

es-419

---

zh-CN

en, es, fr, ja, ko, zh-Hans, zh-Hant

zh-Hans

---

```
// Get the localization the app is currently launched in
let currentLocalization = Bundle.main.preferredLocalizations.first

// Get best localization match from a list of available localizations
let available = myServerSupportedLanguages()
let matchedLocalization = Bundle.preferredLocalizations(from: available).first
```



```
// Get the localization the app is currently launched in
```

```
let currentLocalization = Bundle.main.preferredLocalizations.first
```

```
// Get best localization match from a list of available localizations
```

```
let available = myServerSupportedLanguages()
```

```
let matchedLocalization = Bundle.preferredLocalizations(from: available).first
```

```
// Get the localization the app is currently launched in
let currentLocalization = Bundle.main.preferredLocalizations.first

// Get best localization match from a list of available localizations
let available = myServerSupportedLanguages()
let matchedLocalization = Bundle.preferredLocalizations(from: available).first
```

# Text Summary

Use strings in Storyboards

Use NSLocalizedString

Use formatters

Use Bundle APIs to determine language for remote content

Text

Layout and images

Exporting and testing

# Layout

# Layout

Adapt for...

# Layout

Adapt for...

- Different translation lengths

# Layout

Adapt for...

- Different translation lengths

# Backup

English



# Layout

Adapt for...

- Different translation lengths

**Backup**

English

**Varmuuskopiointi**

Finnish

# Layout

Adapt for...

- Different translation lengths

**Backup**

English

- Script directionality

**Varmuuskopiointi**

Finnish

# Layout

Adapt for...

- Different translation lengths

**Backup**

English

**Varmuuskopointi**

Finnish

- Script directionality

**Hello**



English

# Layout

Adapt for...

- Different translation lengths

**Backup**

English

- Script directionality

**Hello**



English

**Varmuuskopiointi**

Finnish

**مرحباً**



Arabic

# Layout in UIKit

# Layout in UIKit

UIStackView

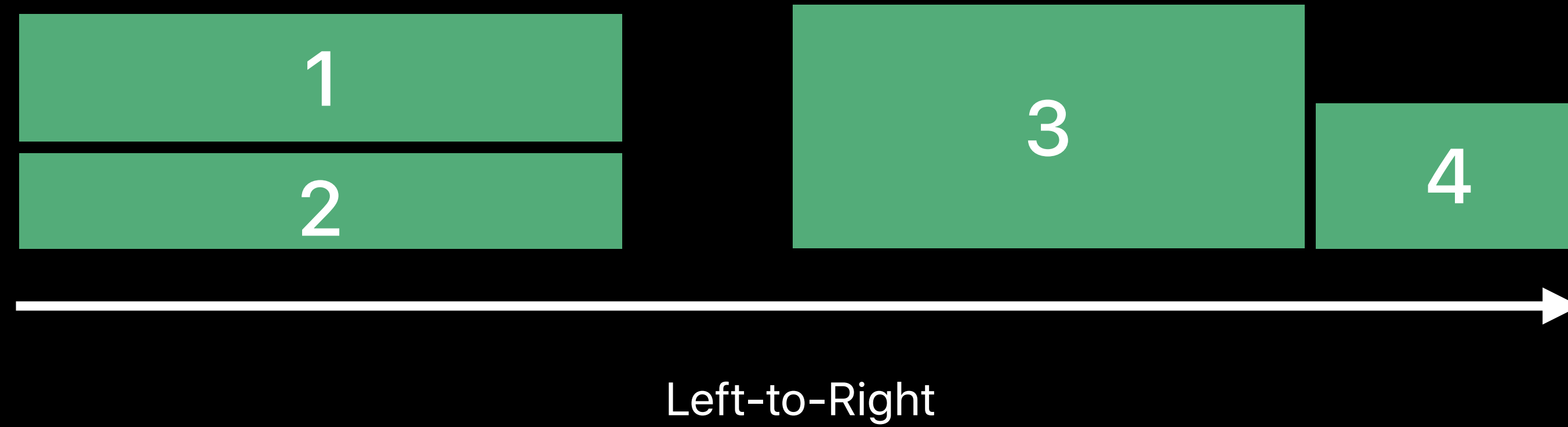
# Layout in UIKit

UIStackView



# Layout in UIKit

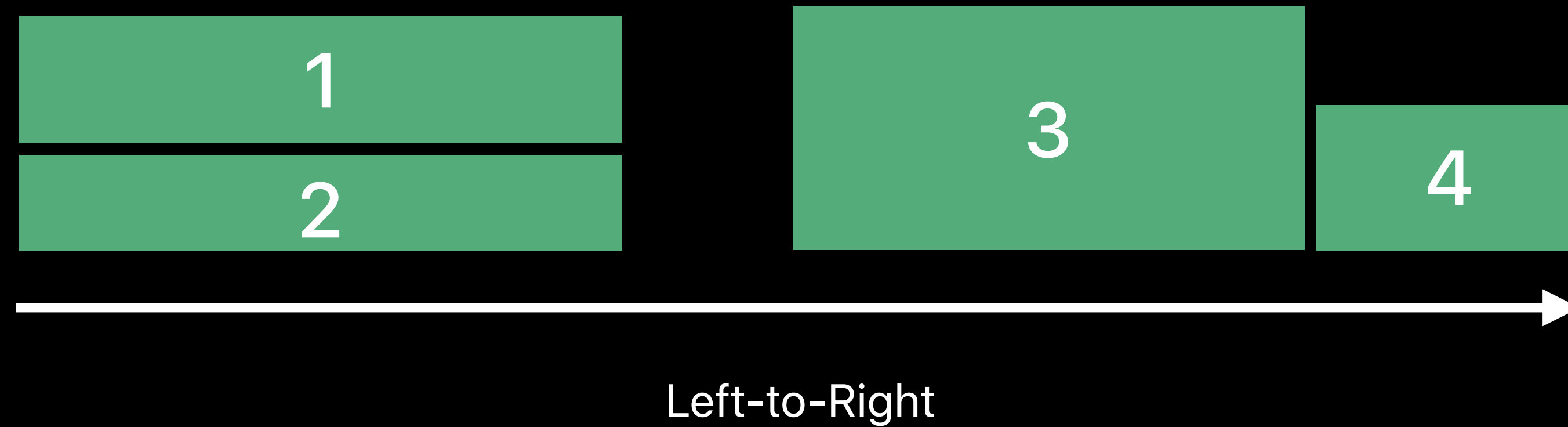
UIStackView





# Layout in UIKit

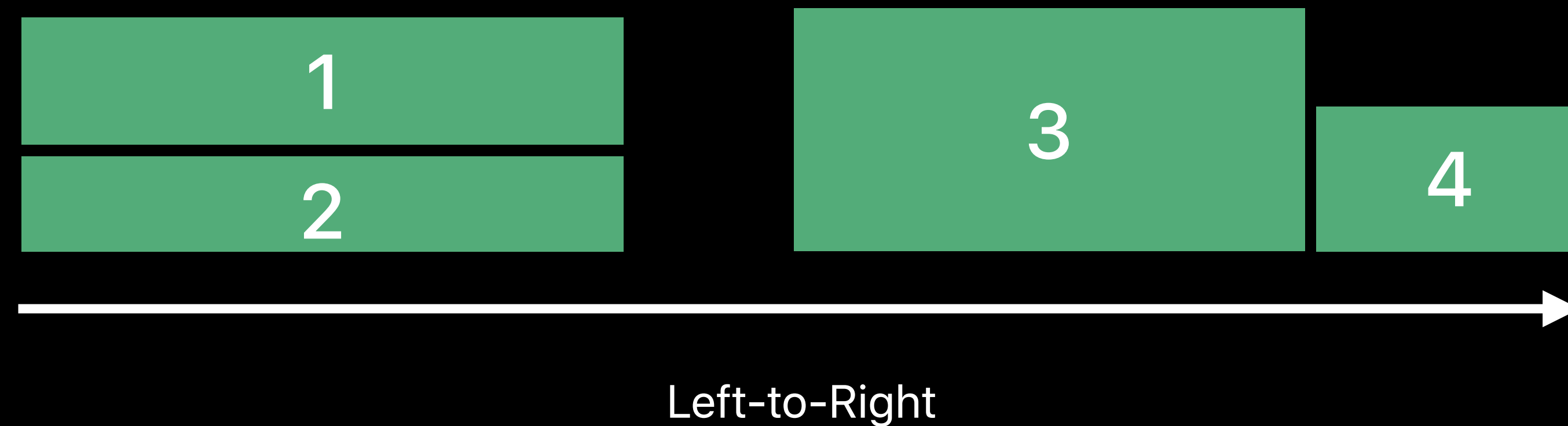
UIStackView



Auto Layout: Leading/trailing constraints

# Layout in UIKit

## UIStackView



Auto Layout: Leading/trailing constraints

- Leading resolves to left in left-to-right, right in right-to-left

# Layout in UIKit

## UIStackView



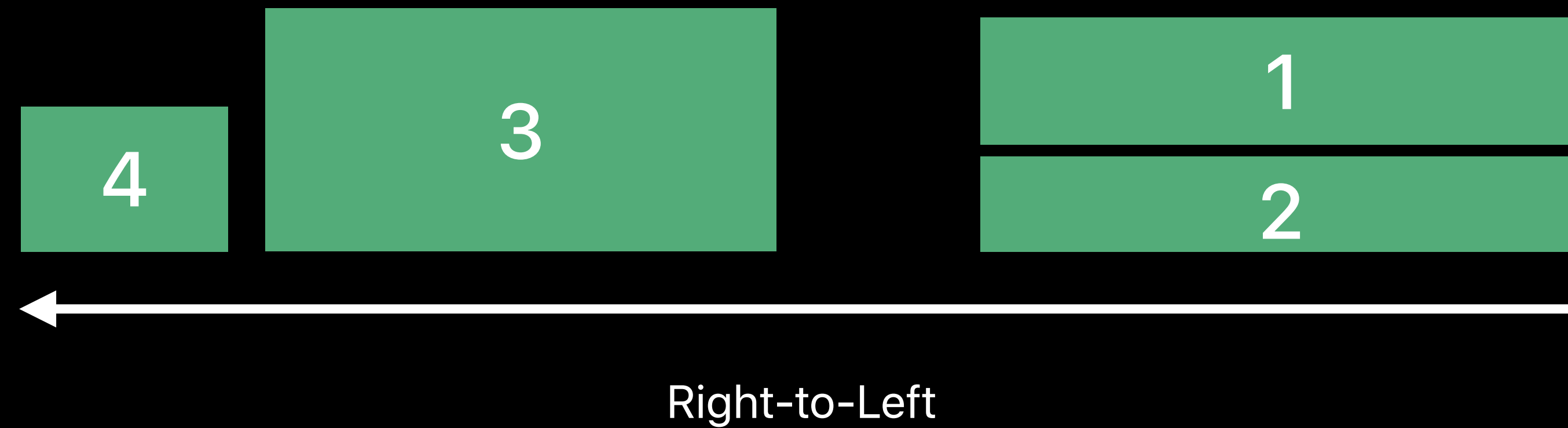
Right-to-Left

Auto Layout: Leading/trailing constraints

- Leading resolves to left in left-to-right, right in right-to-left

# Layout in UIKit

## UIStackView



Auto Layout: Leading/trailing constraints

- Leading resolves to left in left-to-right, right in right-to-left

# Layout in TVMLKit

NEW

# Layout in TVMLKit



NEW

Templates do all the work

# Layout in TVMLKit



NEW

Templates do all the work

For custom positioning:

# Layout in TVMLKit



NEW

Templates do all the work

For custom positioning:

- `leading` and `trailing` `tv-position` and `tv-align`



# Layout in TVMLKit



NEW

Templates do all the work

For custom positioning:

- `leading` and `trailing` `tv-position` and `tv-align`
  - `leading` resolves to `left` in left-to-right, `right` in right-to-left

# Layout in TVMLKit

NEW

Templates do all the work

For custom positioning:

- `leading` and `trailing` `tv-position` and `tv-align`
  - `leading` resolves to `left` in left-to-right, `right` in right-to-left
- `@(media layout-direction: rtl)` media queries for margins and padding

# Layout in TVMLKit

NEW

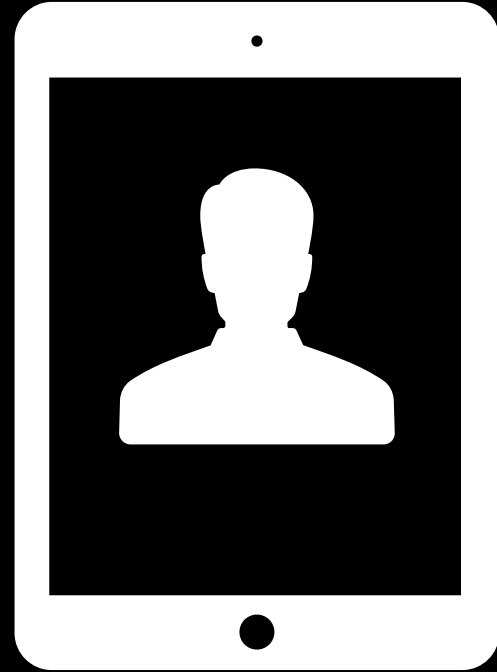
Templates do all the work

For custom positioning:

- `leading` and `trailing` `tv-position` and `tv-align`
  - `leading` resolves to `left` in left-to-right, `right` in right-to-left
- `@(media layout-direction: rtl)` media queries for margins and padding

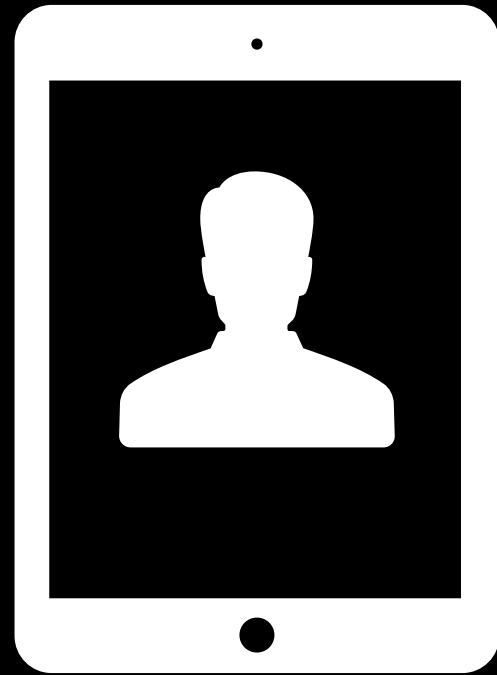
# Images

# Images

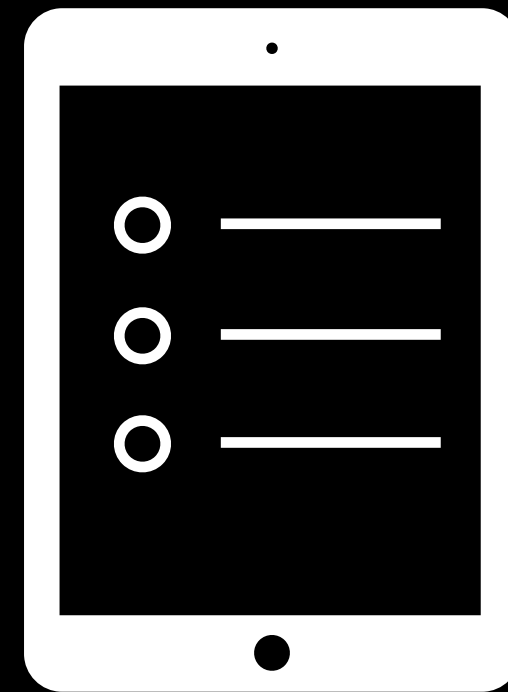


Universal

# Images

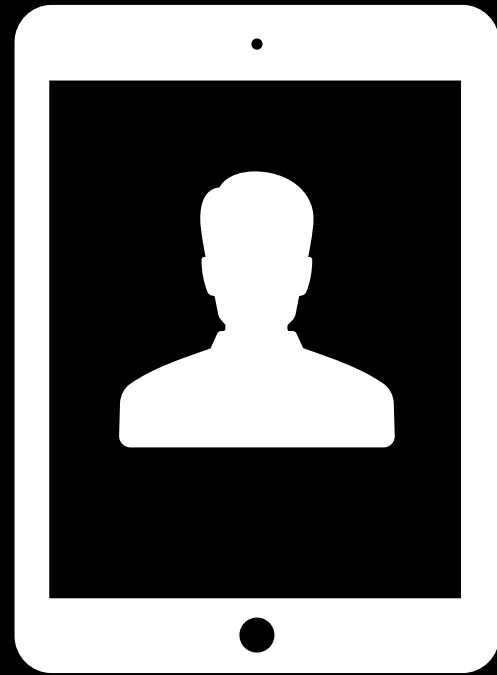


Universal

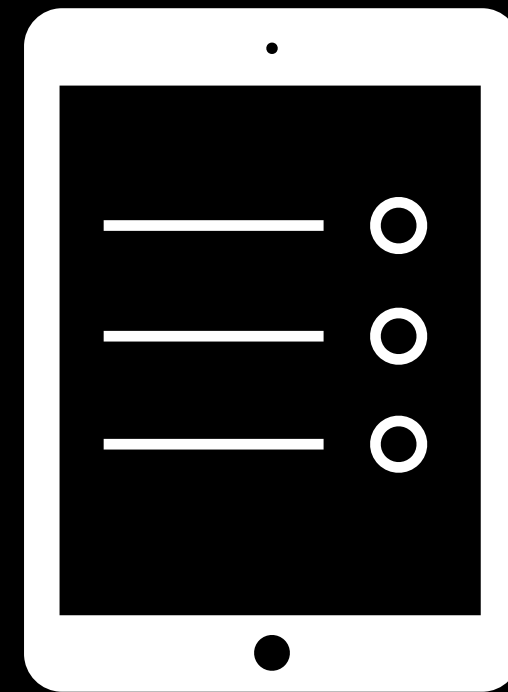


Mirrored

# Images

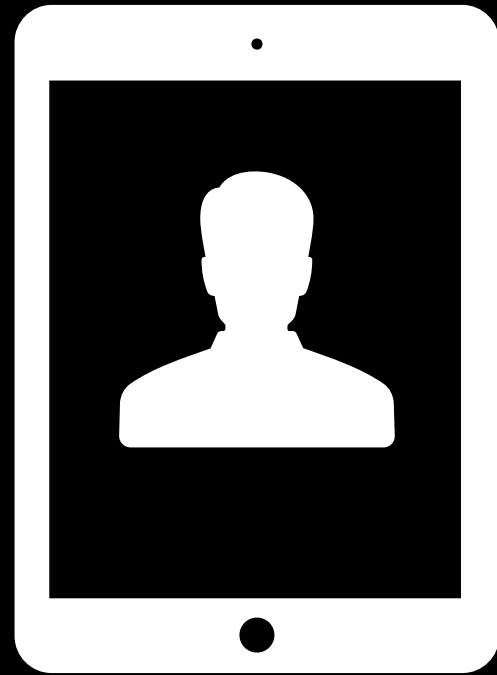


Universal

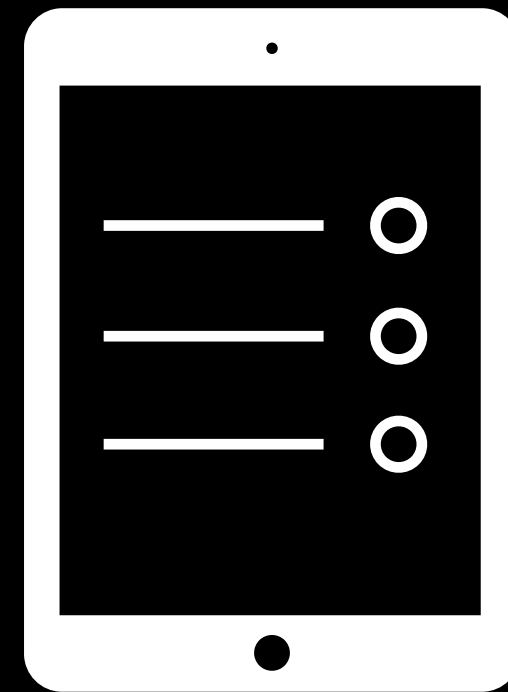


Mirrored

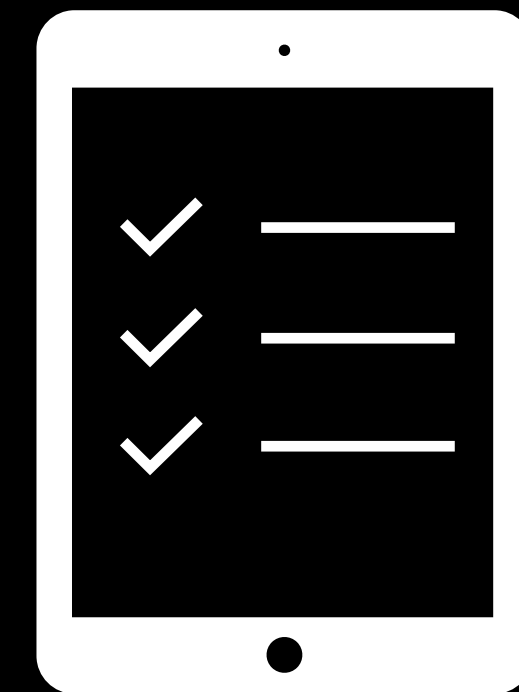
# Images



Universal



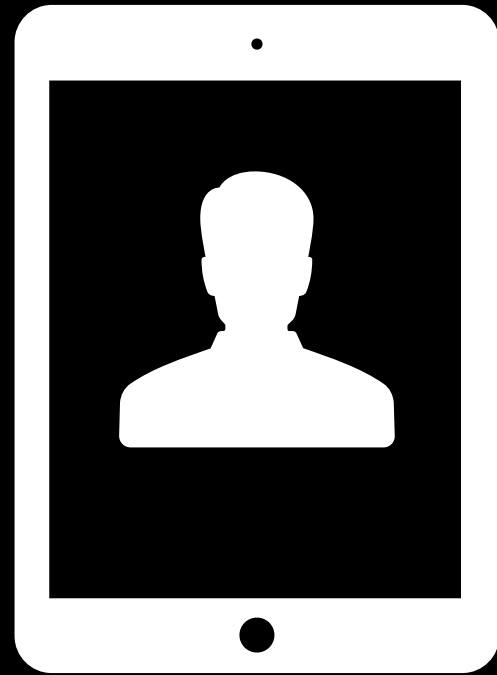
Mirrored



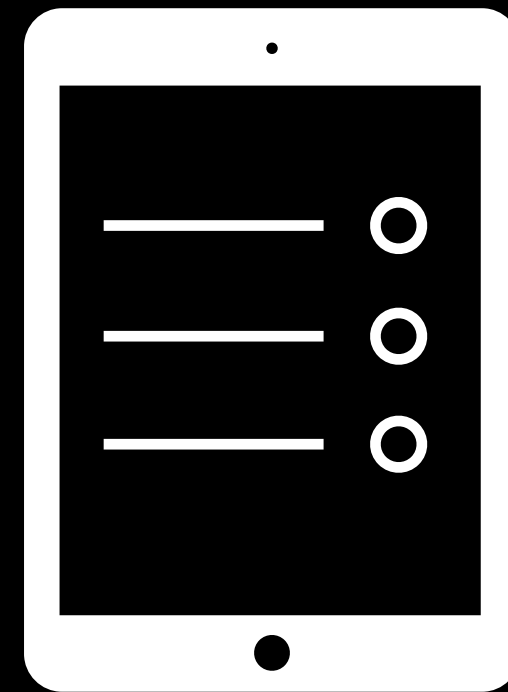
Dedicated



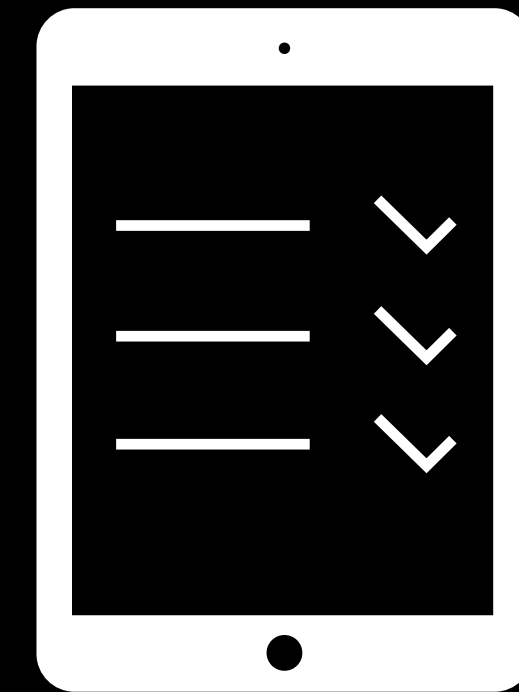
# Images



Universal

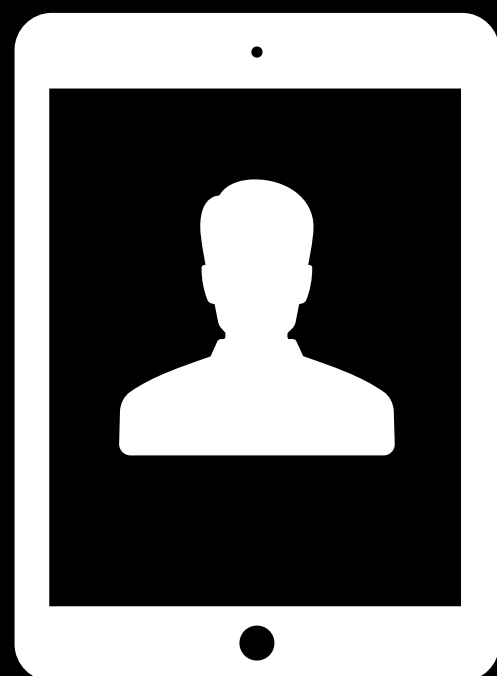


Mirrored

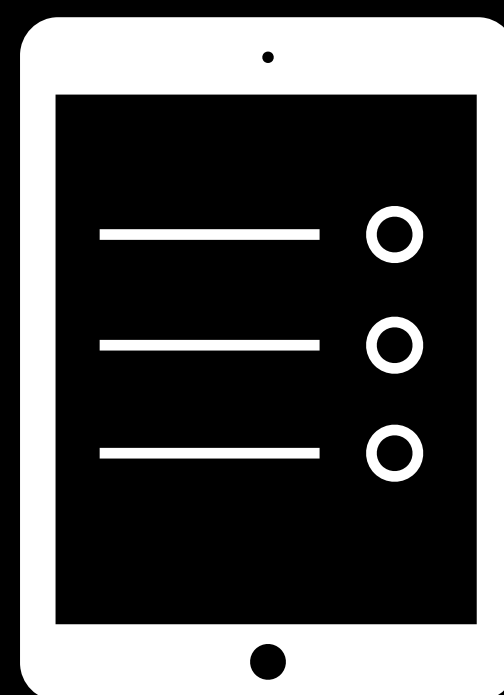


Dedicated

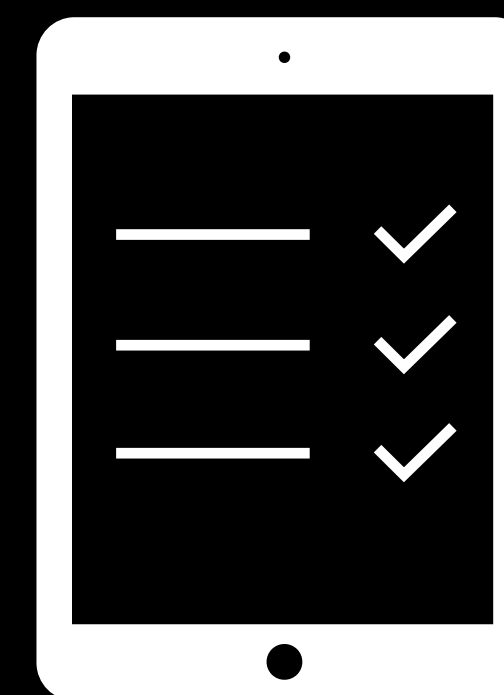
# Images



Universal

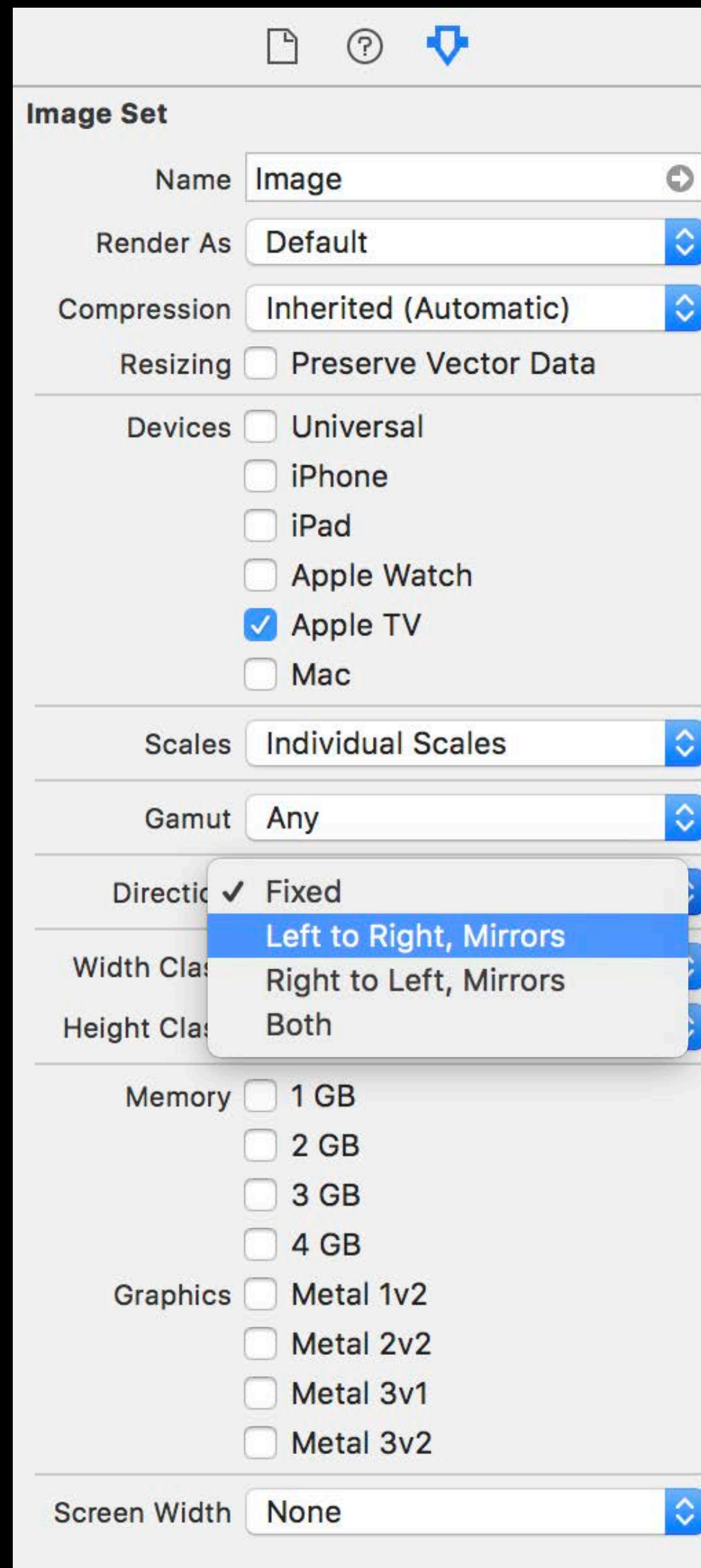


Mirrored



Dedicated

# Images

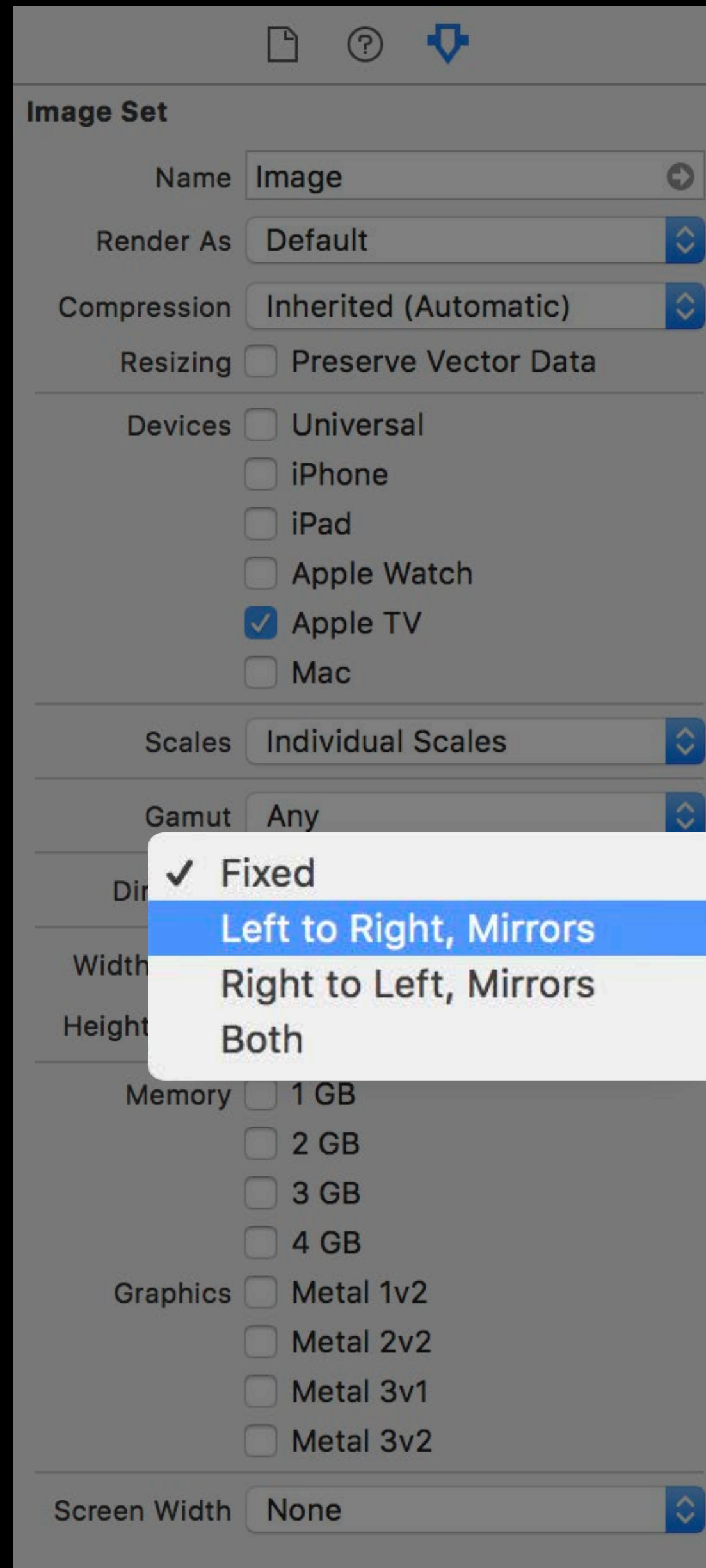


The image shows a settings panel for an image set. At the top, there are icons for a document, a question mark, and a download arrow. The panel is titled "Image Set" and contains the following settings:

- Name: Image
- Render As: Default
- Compression: Inherited (Automatic)
- Resizing:  Preserve Vector Data
- Devices:  Universal,  iPhone,  iPad,  Apple Watch,  Apple TV,  Mac
- Scales: Individual Scales
- Gamut: Any
- Direction:  Fixed,  Left to Right, Mirrors,  Right to Left, Mirrors,  Both
- Width Class: (partially visible)
- Height Class: (partially visible)
- Memory:  1 GB,  2 GB,  3 GB,  4 GB
- Graphics:  Metal 1v2,  Metal 2v2,  Metal 3v1,  Metal 3v2
- Screen Width: None

A dropdown menu is open over the "Direction" setting, showing the options: Fixed (checked), Left to Right, Mirrors (highlighted), Right to Left, Mirrors, and Both.

# Images



# Images

---

What's New in International User Interfaces

WWDC 2016

---

# Layout and Images Summary

## UIKit

- UINavigationController
- Auto Layout

## TVMLKit

- Templates
- New direction APIs

Text

Layout and images

Exporting and testing

# Adding Languages to Your Project

The screenshot shows the Xcode interface for a project named "My tvOS App". The left sidebar displays the project's file structure, including source files like AppDelegate.swift, ViewController.swift, and Main.storyboard, as well as test targets. The right pane is set to the "Info" tab, showing the "Deployment Target" section with the message "No macOS or iOS Targets Found". Below this, the "Configurations" section lists "Debug" and "Release" configurations, both with "No Configurations Set". The "Localizations" section shows "English — Development Language" with "1 File Localized" and a checked option for "Use Base Internationalization".

**PROJECT**

- My tvOS App

**TARGETS**

- My tvOS App
- My tvOS AppTests
- My tvOS AppUITests

**Deployment Target**

No macOS or iOS Targets Found

**Configurations**

Name	Based on Configuration File
▶ Debug	No Configurations Set
▶ Release	No Configurations Set

Use  for command-line builds

**Localizations**

Language	Resources
English — Development Language	1 File Localized

Use Base Internationalization



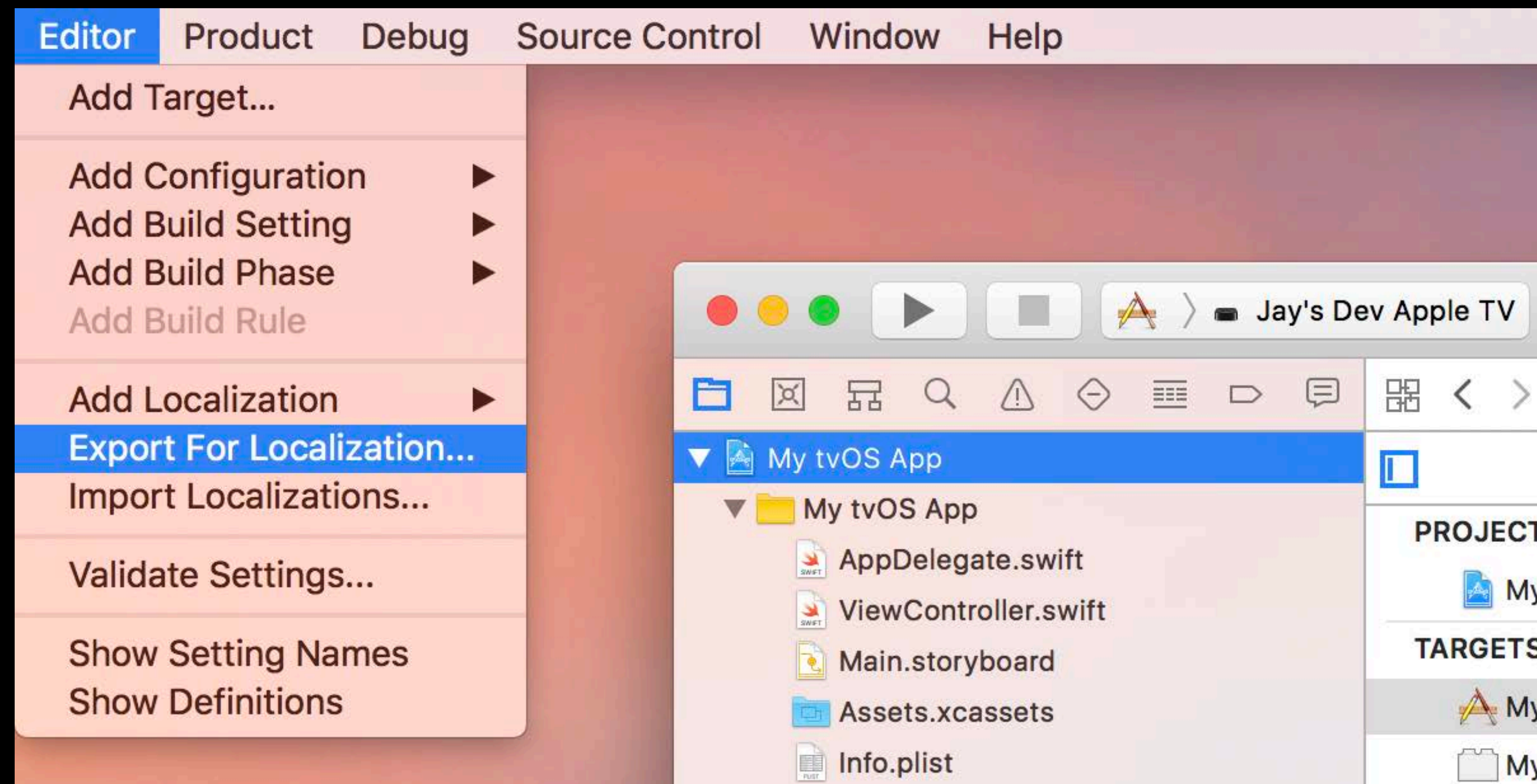
# Adding Languages to Your Project

The screenshot shows the Xcode interface for a project named "My tvOS App". The left sidebar displays the project's file structure, including source files like AppDelegate.swift, ViewController.swift, Main.storyboard, and Info.plist, as well as test targets. The main area shows the "Build Settings" tab, with the "Deployment Target" section indicating "No macOS or iOS Targets Found" and the "Configurations" section listing "Debug" and "Release" configurations. A "Localizations" dialog box is overlaid on the bottom right, showing a table of localized resources.

Language	Resources
English — Development Language	1 File Localized

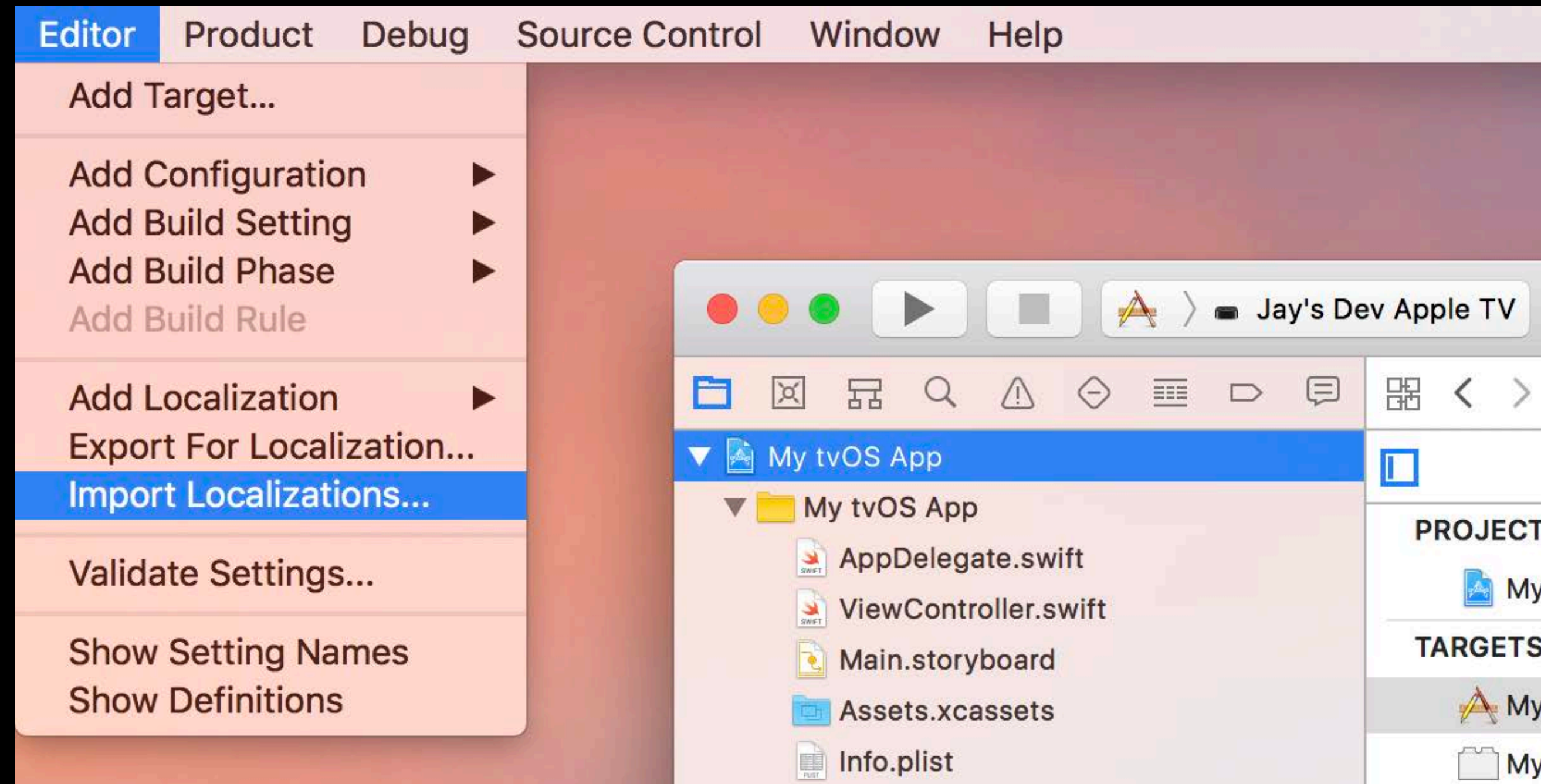
Use Base Internationalization

# Exporting Your Localizable Content





# Importing Translated Content



# Testing

# Testing

Xcode has several features for testing and verification

# Testing

Xcode has several features for testing and verification

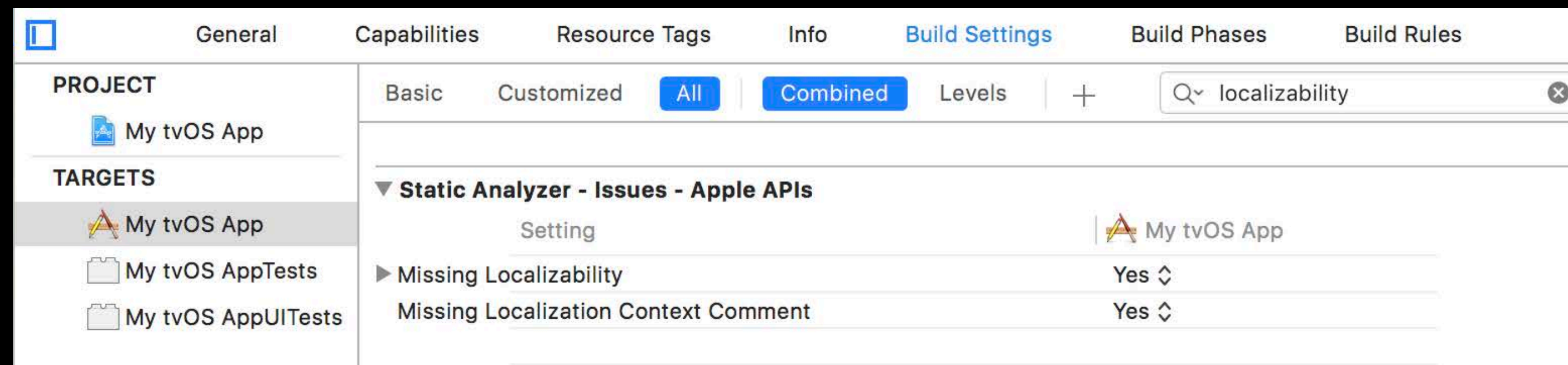
Even if you don't speak the language!

# Testing

Xcode has several features for testing and verification

Even if you don't speak the language!

- Localizability static analyzer



# Testing

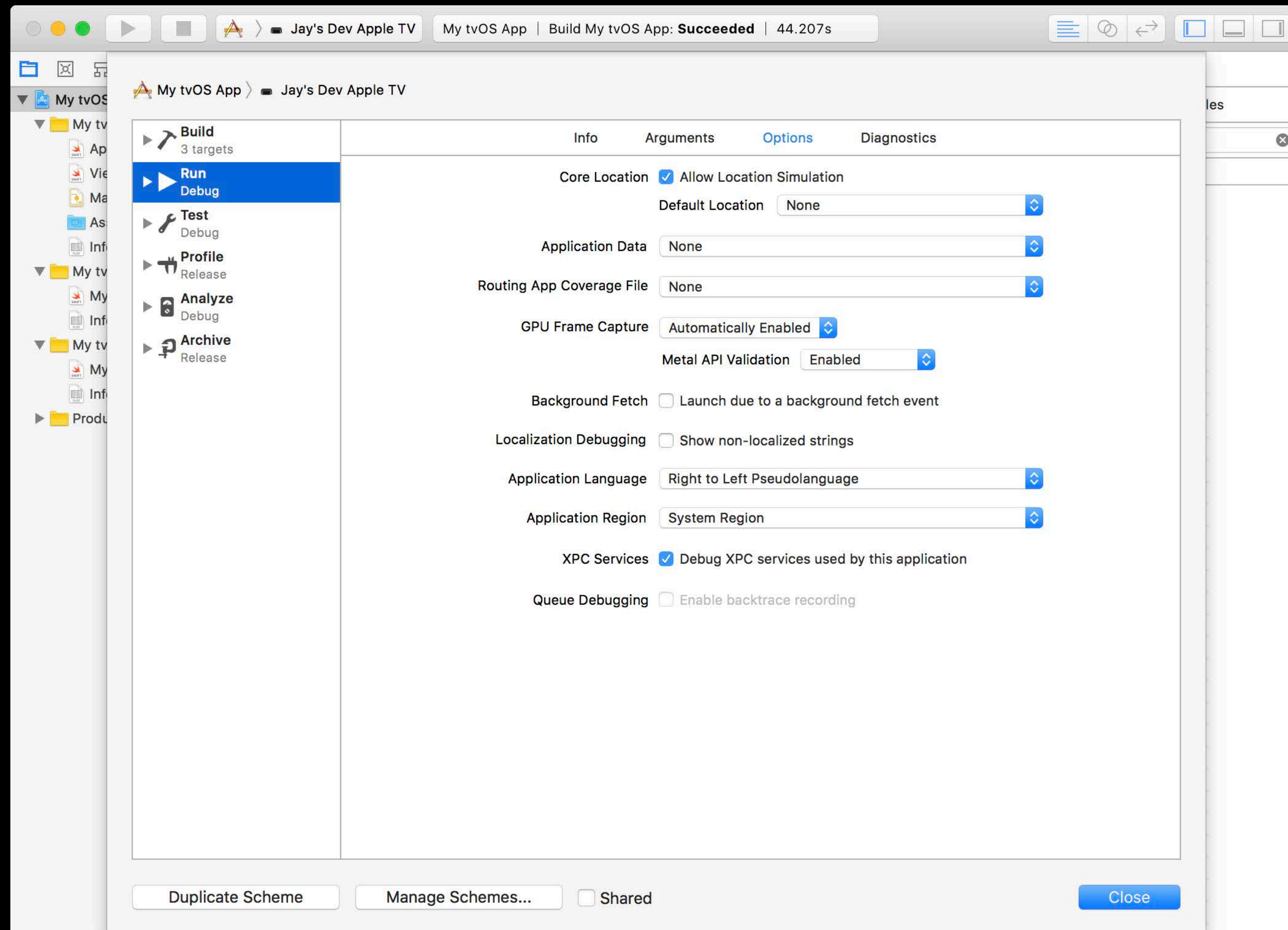
Xcode has several features for testing and verification

Even if you don't speak the language!

- Localizability static analyzer
- Pseudolanguages



# Testing



# Testing

Xcode has several features for testing and verification

Even if you don't speak the language!

- Localizability static analyzer
- Pseudolanguages

# Project Setup Summary

Add supported languages to your project

Export and import localizable content

Static analyzer and runtime pseudolanguages

# More Information

<https://developer.apple.com/wwdc17/248>

# Related Sessions

---

[Localizing with Xcode 9](#)

Grand Ballroom B

Tuesday 10:20AM

---

[Advances in TVMLKit](#)

Grand Ballroom B

Tuesday 11:20AM

---

[Internationalization Best Practices](#)

WWDC 2016

---

[What's New in International User Interfaces](#)

WWDC 2016

---

[Measurements and Units](#)

---

WWDC 2016

