

# Localizing with Xcode 6

Best practices and new workflows

Session 412


Zoltan Foley-Fisher


Xcode Software Engineer

Chris Hanson

Xcode Software Engineer



[← Mapa](#) Localização 



Rio De Janeiro - RJ  
10.601 quilômetros, Rio De Janeiro

[Endereço](#)  
Rio De Janeiro - RJ  
Brasil

[Itinerários até Aqui](#) [Itinerários Daqui](#)

[Itinerários por Transporte Público](#)

[Criar Novo Contato](#)

[Adicionar a Contato](#)

[Novo Favorito](#)

[Indicar Problema](#)



تم [إضافة حدث](#) [إلغاء](#)

[العنوان](#)

[الموقع](#)

اليوم كله

يبدأ ٢٠١٤/٠٣/١٤ م ٢:٠٠

ينتهي م ٣:٠٠

ج ح خ ه ع غ ف ق ث ص ض  
ة ك م ن ت ا ل ب ي س ش  
ى و ر ز د ن ط ظ ء

الرجوع ٠ مسافة ٣٢١



台北市



台北市

台灣 台北市 >

3D  

# Agenda

# Agenda

Localization Frameworks on OS X and iOS

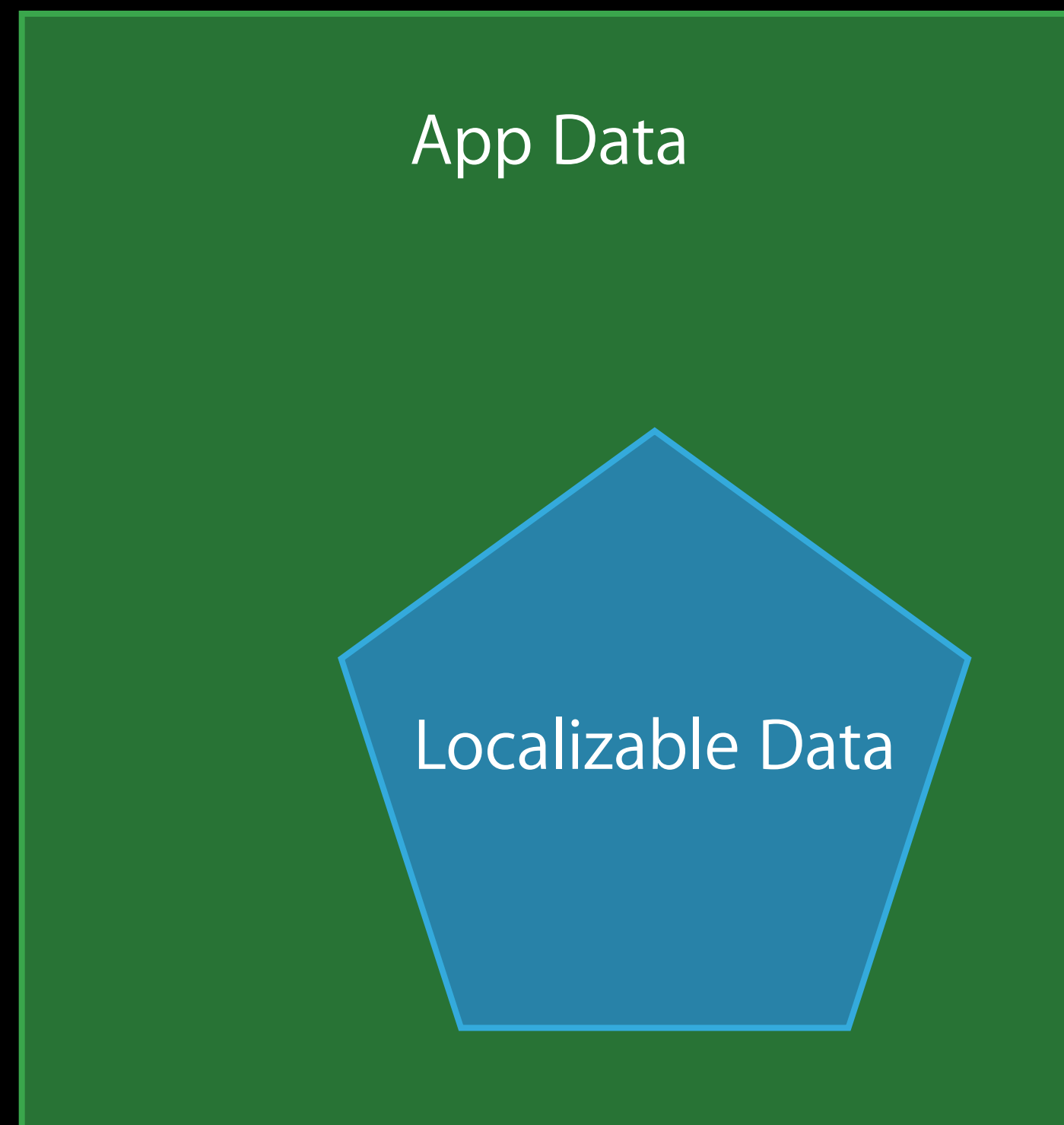
# Agenda

Localization Frameworks on OS X and iOS  
Xcode Workflows for Localization

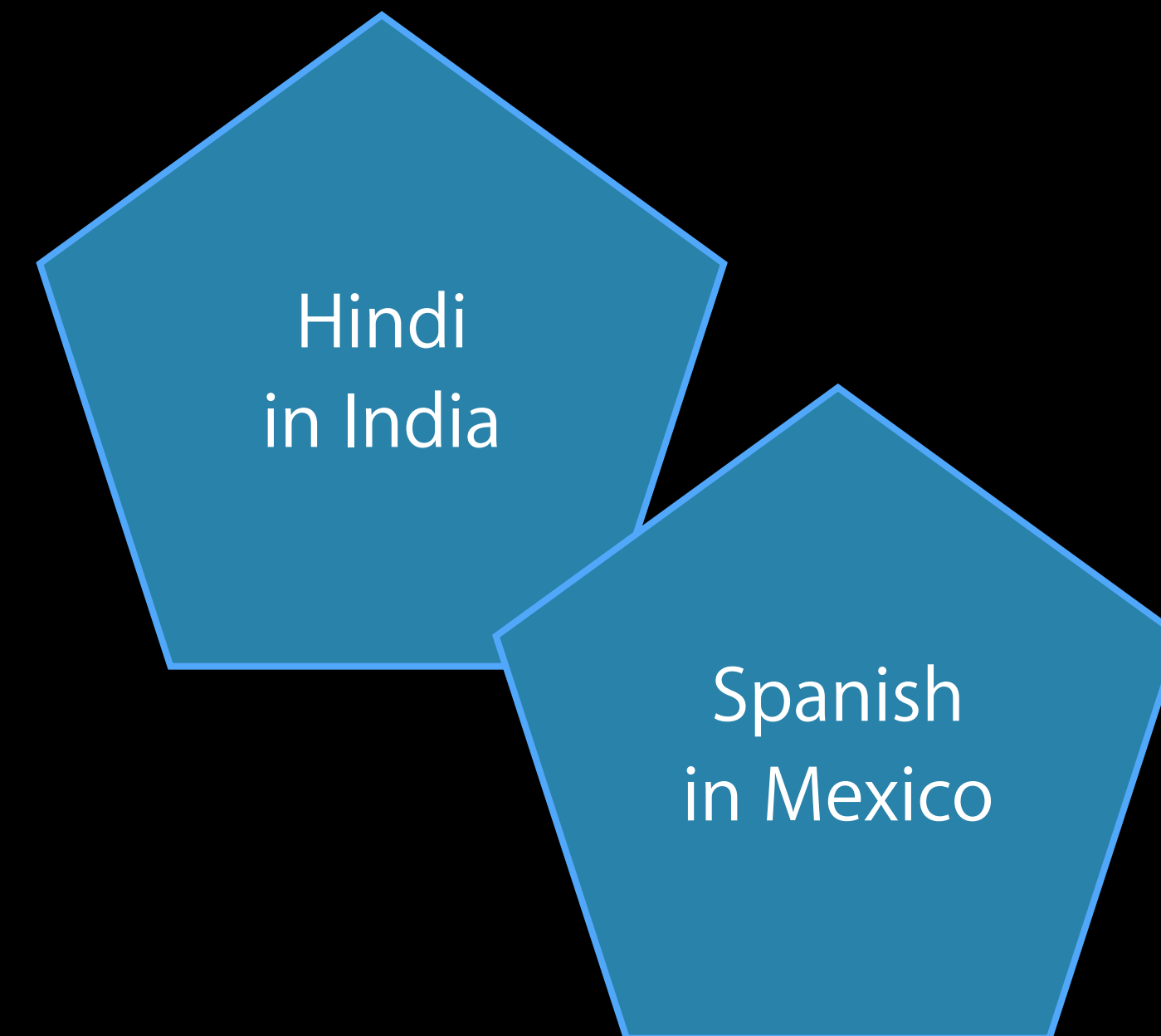
# Internationalization and Localization

# Internationalization and Localization

Step 1



Step 2





# Languages and Regions



iOS Simulator - iPhone 4s - iPhone 4s / iOS 8.0 (12A4265u)

Carrier 11:55 AM

### Earthquakes

California		3.5
May 20, 2014, 12:47 PM		
Virgin Islands region		3.3
May 20, 2014, 5:09 PM		
Virgin Islands region		2.7
May 20, 2014, 4:59 PM		
Puerto Rico region		2.8
May 20, 2014, 10:16 AM		
Puerto Rico region		2.8
May 20, 2014, 6:24 AM		

# Languages and Regions



iOS Simulator - iPhone 4s - iPhone 4s / iOS 8.0 (12A4265u)

Carrier 11:55 AM

### Earthquakes

California		3.5
May 20, 2014, 12:47 PM		
Virgin Islands region		3.3
May 20, 2014, 5:09 PM		
Virgin Islands region		2.7
May 20, 2014, 4:59 PM		
Puerto Rico region		2.8
May 20, 2014, 10:16 AM		
Puerto Rico region		2.8
May 20, 2014, 6:24 AM		



iOS Simulator - iPhone 4s - iPhone 4s / iOS 8.0 (12A4265u)

Carrier 11:54 AM

### Terremotos

California		3.5
20/05/2014 12:47		
Virgin Islands region		3.3
20/05/2014 17:09		
Virgin Islands region		2.7
20/05/2014 16:59		
Puerto Rico region		2.8
20/05/2014 10:16		
Puerto Rico region		2.8
20/05/2014 06:24		

# Languages and Regions



iOS Simulator - iPhone 4s - iPhone 4s / iOS 8.0 (12A4265u)

Carrier 11:55 AM

### Earthquakes

California		3.5
May 20, 2014, 12:47 PM		
Virgin Islands region		3.3
May 20, 2014, 5:09 PM		
Virgin Islands region		2.7
May 20, 2014, 4:59 PM		
Puerto Rico region		2.8
May 20, 2014, 10:16 AM		
Puerto Rico region		2.8
May 20, 2014, 6:24 AM		



iOS Simulator - iPhone 4s - iPhone 4s / iOS 8.0 (12A4265u)

Carrier 11:54 AM

### Terremotos

California		3.5
20/05/2014 12:47		
Virgin Islands region		3.3
20/05/2014 17:09		
Virgin Islands region		2.7
20/05/2014 16:59		
Puerto Rico region		2.8
20/05/2014 10:16		
Puerto Rico region		2.8
20/05/2014 06:24		

# Languages and Regions



Carrier 11:55 AM

**Earthquakes**

California		3.5
May 20, 2014, 12:47 PM		
Virgin Islands region		3.3
May 20, 2014, 5:09 PM		
Virgin Islands region		2.7
May 20, 2014, 4:59 PM		
Puerto Rico region		2.8
May 20, 2014, 10:16 AM		
Puerto Rico region		2.8
May 20, 2014, 6:24 AM		



Carrier 11:54 AM

**Terremotos**

California		3.5
20/05/2014 12:47		
Virgin Islands region		3.3
20/05/2014 17:09		
Virgin Islands region		2.7
20/05/2014 16:59		
Puerto Rico region		2.8
20/05/2014 10:16		
Puerto Rico region		2.8
20/05/2014 06:24		



Carrier 11:55 AM

**Terremotos**

California		3,5
20/5/2014 12:47		
Virgin Islands region		3,3
20/5/2014 17:09		
Virgin Islands region		2,7
20/5/2014 16:59		
Puerto Rico region		2,8
20/5/2014 10:16		
Puerto Rico region		2,8
20/5/2014 6:24		

# Languages and Regions



iOS Simulator - iPhone 4s - iPhone 4s / iOS 8.0 (12A4265u)

Carrier 11:55 AM

**Earthquakes**

California		3.5
May 20, 2014, 12:47 PM		
Virgin Islands region		3.3
May 20, 2014, 5:09 PM		
Virgin Islands region		2.7
May 20, 2014, 4:59 PM		
Puerto Rico region		2.8
May 20, 2014, 10:16 AM		
Puerto Rico region		2.8
May 20, 2014, 6:24 AM		



iOS Simulator - iPhone 4s - iPhone 4s / iOS 8.0 (12A4265u)

Carrier 11:54 AM

**Terremotos**

California		3.5
20/05/2014 12:47		
Virgin Islands region		3.3
20/05/2014 17:09		
Virgin Islands region		2.7
20/05/2014 16:59		
Puerto Rico region		2.8
20/05/2014 10:16		
Puerto Rico region		2.8
20/05/2014 06:24		



iOS Simulator - iPhone 4s - iPhone 4s / iOS 8.0 (12A4265u)

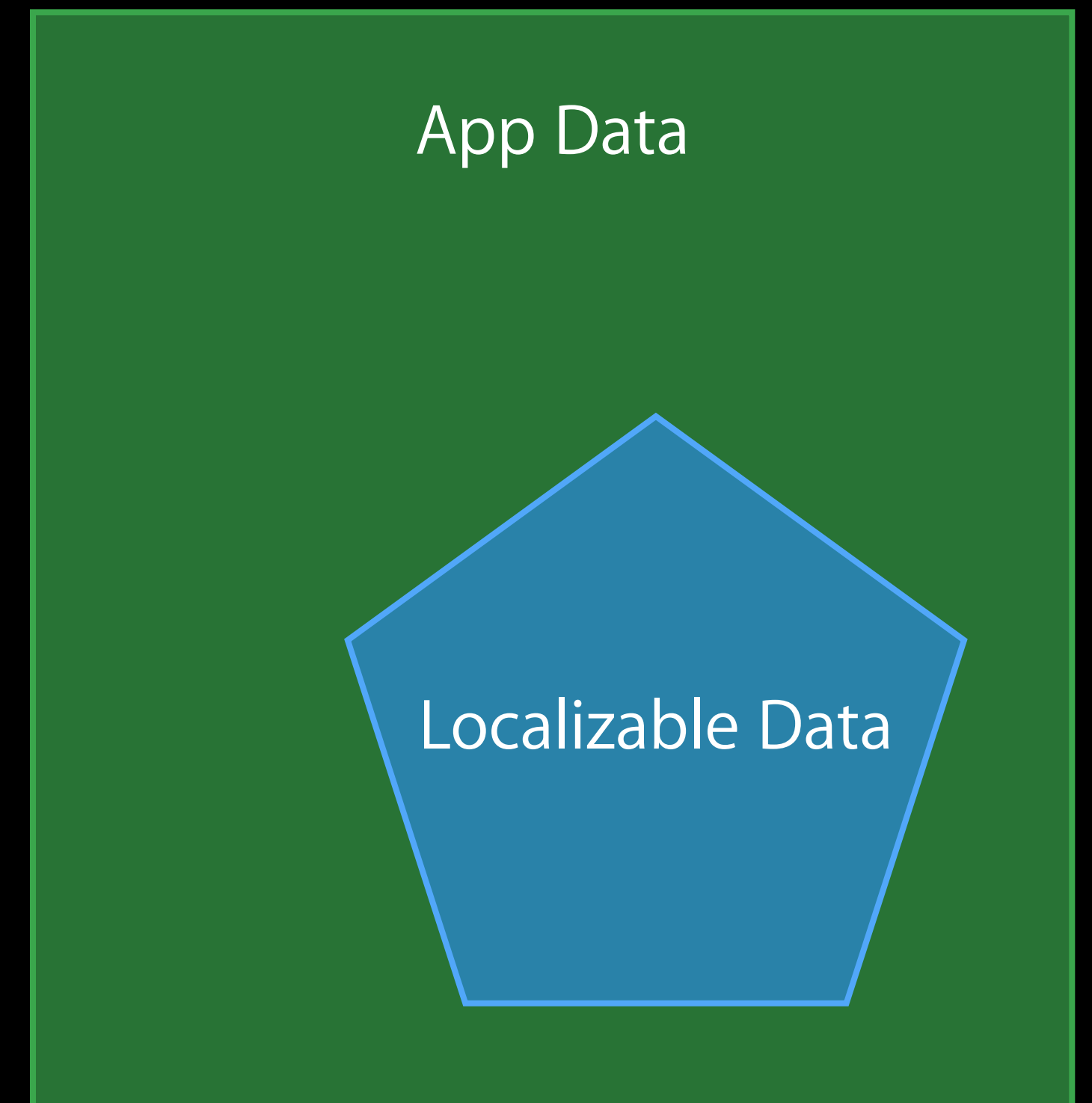
Carrier 11:55 AM

**Terremotos**

California		3,5
20/5/2014 12:47		
Virgin Islands region		3,3
20/5/2014 17:09		
Virgin Islands region		2,7
20/5/2014 16:59		
Puerto Rico region		2,8
20/5/2014 10:16		
Puerto Rico region		2,8
20/5/2014 6:24		

# Internationalization

Preparing your App



# Framework Support

# Framework Support

Localization support is pervasive in the system frameworks



# Framework Support

Localization support is pervasive in the system frameworks

Separates localizable data from the rest of your app

# Framework Support

Localization support is pervasive in the system frameworks

Separates localizable data from the rest of your app

- Images, sounds, movies, documentation

# Framework Support

Localization support is pervasive in the system frameworks

Separates localizable data from the rest of your app

- Images, sounds, movies, documentation
- User-facing text

# Framework Support

Localization support is pervasive in the system frameworks

Separates localizable data from the rest of your app

- Images, sounds, movies, documentation
- User-facing text
- ...even in your source code!

# Framework Support

Localization support is pervasive in the system frameworks

Separates localizable data from the rest of your app

- Images, sounds, movies, documentation
- User-facing text
- ...even in your source code!

Extremely simple to use

NSString()

# NSStringLocalizedString()

Internationalize strings for your interface in your source code

# NSStringLocalizedString()

Internationalize strings for your interface in your source code

Lets you keep “nice” strings in your code, including format strings



# NSStringLocalizedString()

Internationalize strings for your interface in your source code

Lets you keep “nice” strings in your code, including format strings

No explicit loading/management of localized strings needed at runtime

# NSStringLocalizedString()

Internationalize strings for your interface in your source code

Lets you keep “nice” strings in your code, including format strings

No explicit loading/management of localized strings needed at runtime

Easily keep separate string tables with NSStringLocalizedStringFromTable()

# NSLocalizedString()

```
label.text = [NSString stringWithFormat:  
    @"%@ is %d meters tall",  
    mountain.name, mountain.height];
```

# NSLocalizedString()

```
label.text = [NSString stringWithFormat:  
    NSLocalizedString(@"%@ is %d meters tall",  
    @"Name and height (in meters) of a mountain"),  
    mountain.name, mountain.height];
```

# NSLocalizedString()

```
label.text = [NSString localizedStringWithFormat:  
    NSLocalizedString(@"%@ is %d meters tall",  
    @"Name and height (in meters) of a mountain"),  
    mountain.name, mountain.height];
```

# NSStringLocalizedString()

```
label.text = String.localizedStringWithFormat(
    NSStringLocalizedString("%@ is %d meters tall",
        comment:"Name and height (in meters) of a mountain"),
    mountain.name, mountain.height)
```

# Localizable.strings

```
/* Name and height (in meters) of a mountain */  
"%@ is %d meters tall" = "%1$@ is %2$d meters tall";
```

# Localizable.strings

```
/* Name and height (in meters) of a mountain */  
"%@ is %d meters tall" = "%1$@ is %2$d meters tall";
```



# Localizable.strings

```
/* Name and height (in meters) of a mountain */  
"@ is %d meters tall" = "%1$@ is %2$d meters tall";
```

# Localizable.strings

```
/* Name and height (in meters) of a mountain */  
"%@ is %d meters tall" = "%1$@ is %2$d meters tall";
```

# Localizable.strings

```
/* Name and height (in meters) of a mountain */  
"%@" is %d meters tall" = "%2$d meters is the height of %1$@";
```

# NSFormatter

# NSNumberFormatter

Converts between objects and human-readable text

# NSNumberFormatter

Converts between objects and human-readable text

For both presenting and interpreting text

# NSNumberFormatter

Converts between objects and human-readable text

For both presenting and interpreting text

Uses the current locale by default

# NSDateFormatter

```
NSDateFormatter *formatter = [[NSDateFormatter alloc] init];  
formatter.dateFormat = NSDateFormatterMediumStyle;  
formatter.timeStyle = NSDateFormatterNoStyle;
```



# NSDateFormatter

```
NSDateFormatter *formatter = [[NSDateFormatter alloc] init];  
formatter.dateFormat = NSDateFormatterMediumStyle;  
formatter.timeStyle = NSDateFormatterNoStyle;
```

# NSDateFormatter

```
NSDateFormatter *formatter = [[NSDateFormatter alloc] init];  
formatter.dateFormat = NSDateFormatterMediumStyle;  
formatter.timeStyle = NSDateFormatterNoStyle;
```

# NSDateFormatter

```
NSDateFormatter *formatter = [[NSDateFormatter alloc] init];  
formatter.dateFormat = NSDateFormatterMediumStyle;  
formatter.timeStyle = NSDateFormatterNoStyle;  
  
NSDate *date = mountain.latestClimb.date;  
label.text = [formatter stringFromDate:date];
```

# NSDateFormatter

```
let formatter = NSDateFormatter()
formatter.dateStyle = .MediumStyle
formatter.timeStyle = .NoStyle

let date = mountain.latestClimb.date
label.text = formatter.stringFromDate(date)
```

# NSFormatter

# NSDateFormatter

Dates, date components, and date/time intervals

# NSDateFormatter

Dates, date components, and date/time intervals

Numbers, currency, and byte counts

# NSNumberFormatter

Dates, date components, and date/time intervals

Numbers, currency, and byte counts

Health-related quantities



# NSNumberFormatter

Dates, date components, and date/time intervals

Numbers, currency, and byte counts

Health-related quantities

- Energy

# NSNumberFormatter

Dates, date components, and date/time intervals

Numbers, currency, and byte counts

Health-related quantities

- Energy
- Length

# NSNumberFormatter

Dates, date components, and date/time intervals

Numbers, currency, and byte counts

Health-related quantities

- Energy
- Length
- Mass

# NSBundle

# NSBundle

Standard way to access resources in your app

# NSBundle

Standard way to access resources in your app

Uses the most appropriate resource for current language and region

# NSBundle

Standard way to access resources in your app

Uses the most appropriate resource for current language and region

Directly supported by Xcode

# NSBundle

```
NSURL *imageUrl =  
    [[NSBundle mainBundle] URLForResource:@"GameMenu" withExtension:@"png"];  
  
UIImage *image = [UIImage imageWithContentsOfFile:imageURL.path];  
  
attentionIcon.image = image;
```



# NSBundle

```
attentionIcon.image = [UIImage imageNamed:@"GameMenu"];
```

# NSBundle

```
NSURL *welcomeURL =  
    [[NSBundle mainBundle] URLForResource:@"Welcome" withExtension:@"m4a"];  
  
AVAudioPlayer *player =  
    [[AVAudioPlayer alloc] initWithContentsOfURL:welcomeURL error:&error];
```

# NSBundle

```
let welcomeURL =  
    NSBundle.mainBundle().URLForResource("Welcome", withExtension:"m4a")  
  
let player =  
    AVAudioPlayer(contentsOfURL:welcomeURL, error:&error)
```

Xcode

# Xcode

Use base internationalization and Auto Layout to internationalize your interface

# Xcode

Use base internationalization and Auto Layout to internationalize your interface

Localize other project resources as needed

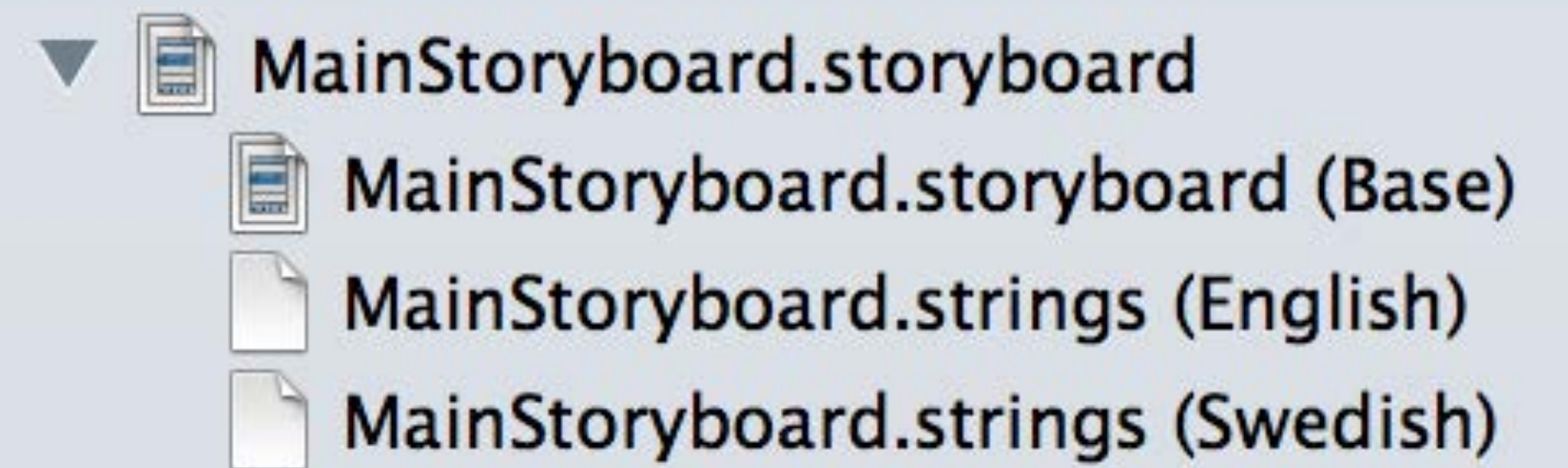
# Xcode

Use base internationalization and Auto Layout to internationalize your interface

Localize other project resources as needed

Preview your app when running and within Xcode

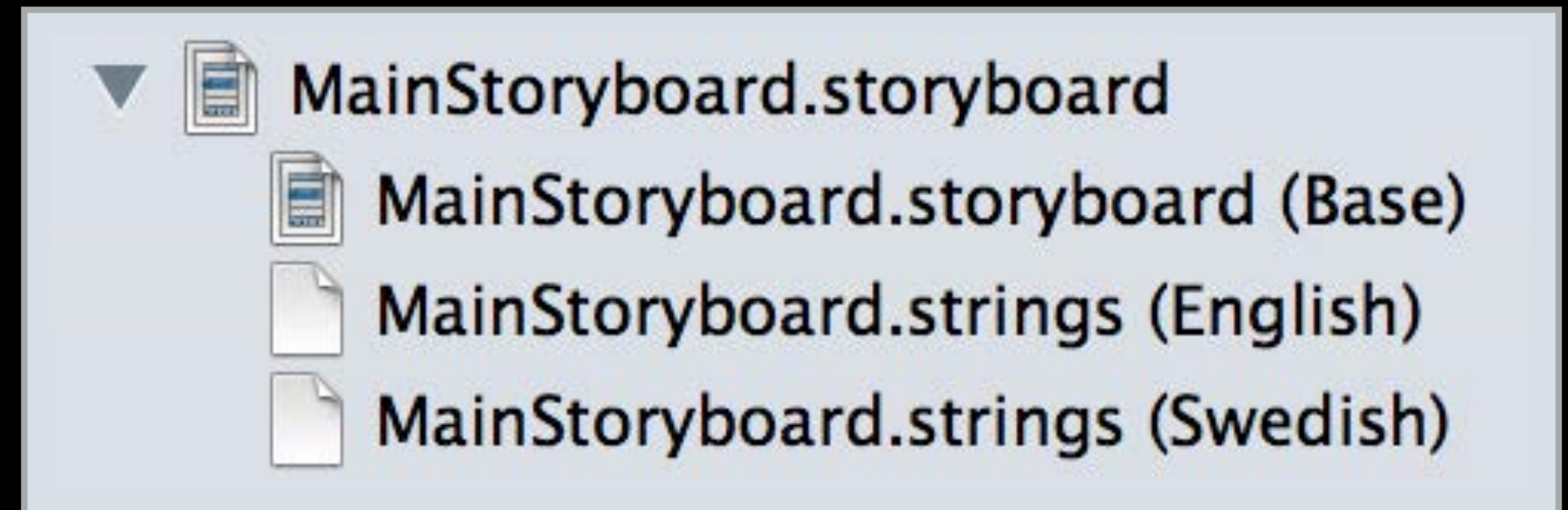
# Base Internationalization





# Base Internationalization

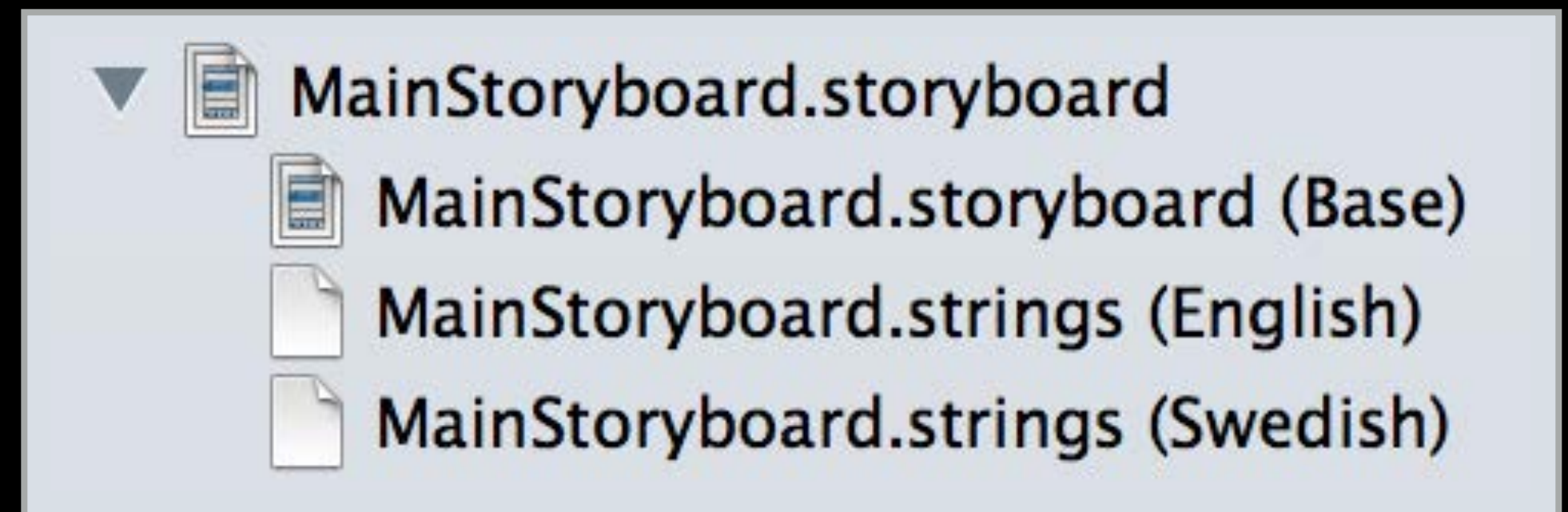
Works with Auto Layout to keep user-facing text distinct from the interface objects themselves



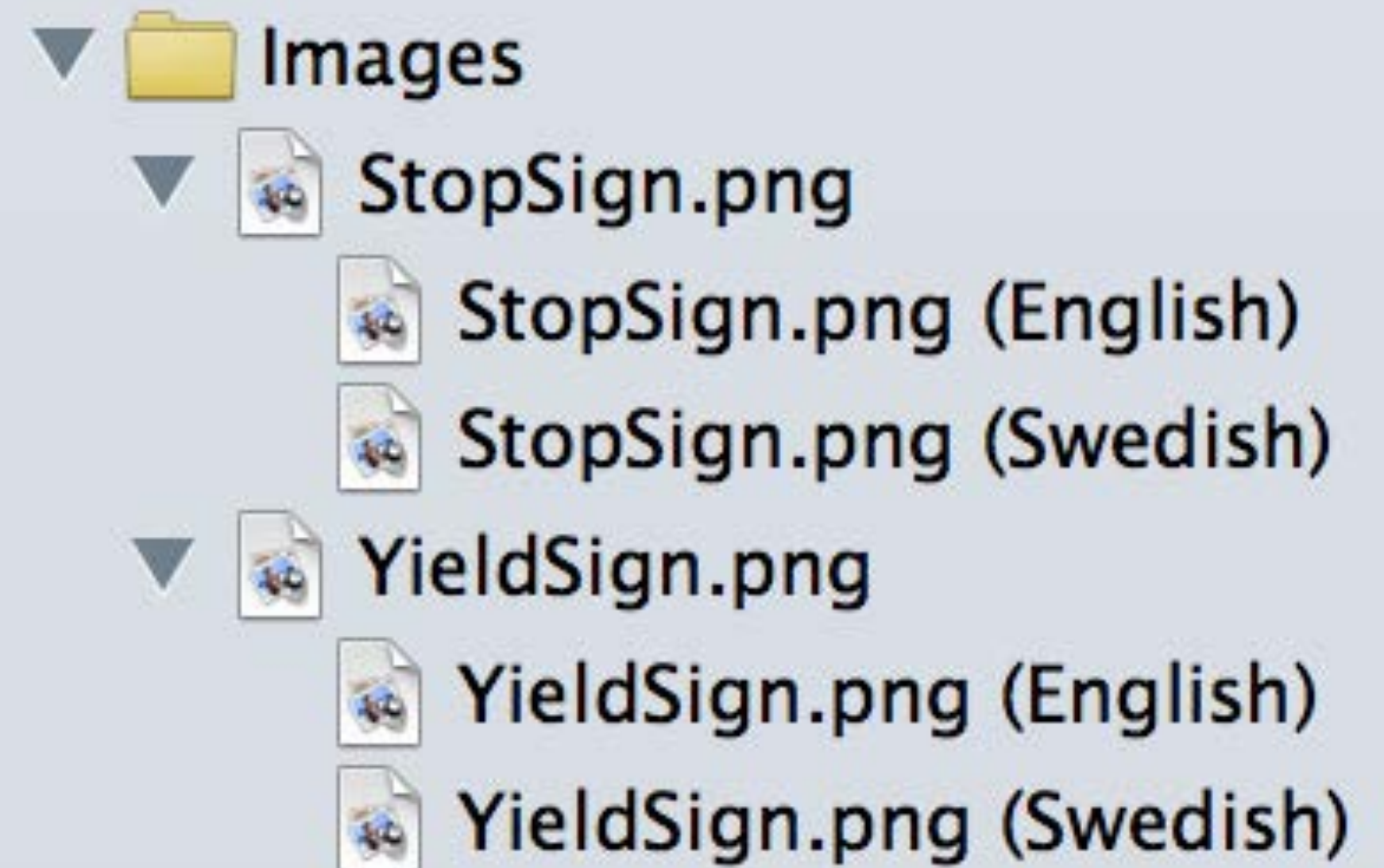
# Base Internationalization

Works with Auto Layout to keep user-facing text distinct from the interface objects themselves

No need to adjust every xib or storyboard file for every supported localization

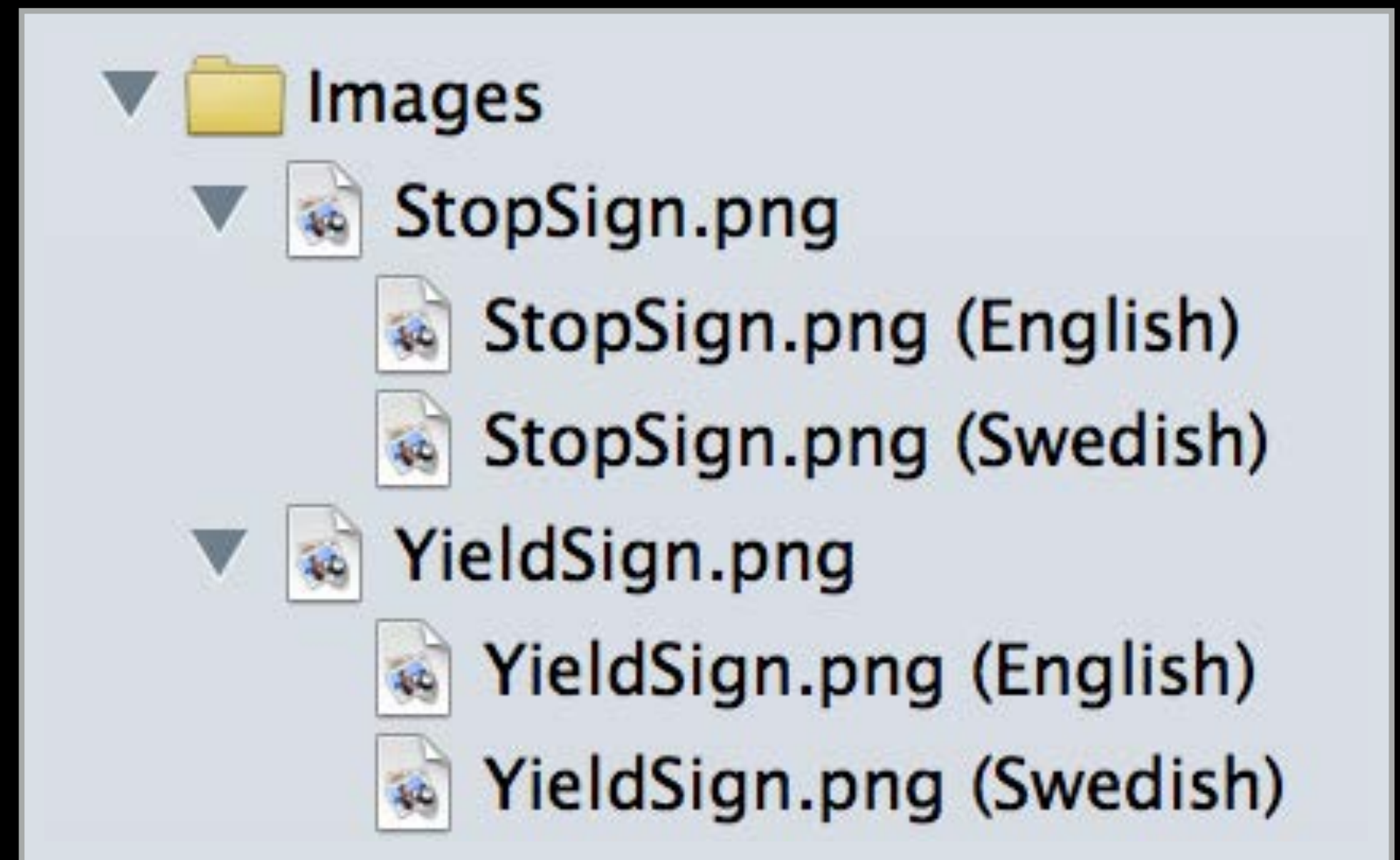


# Other Resources



# Other Resources

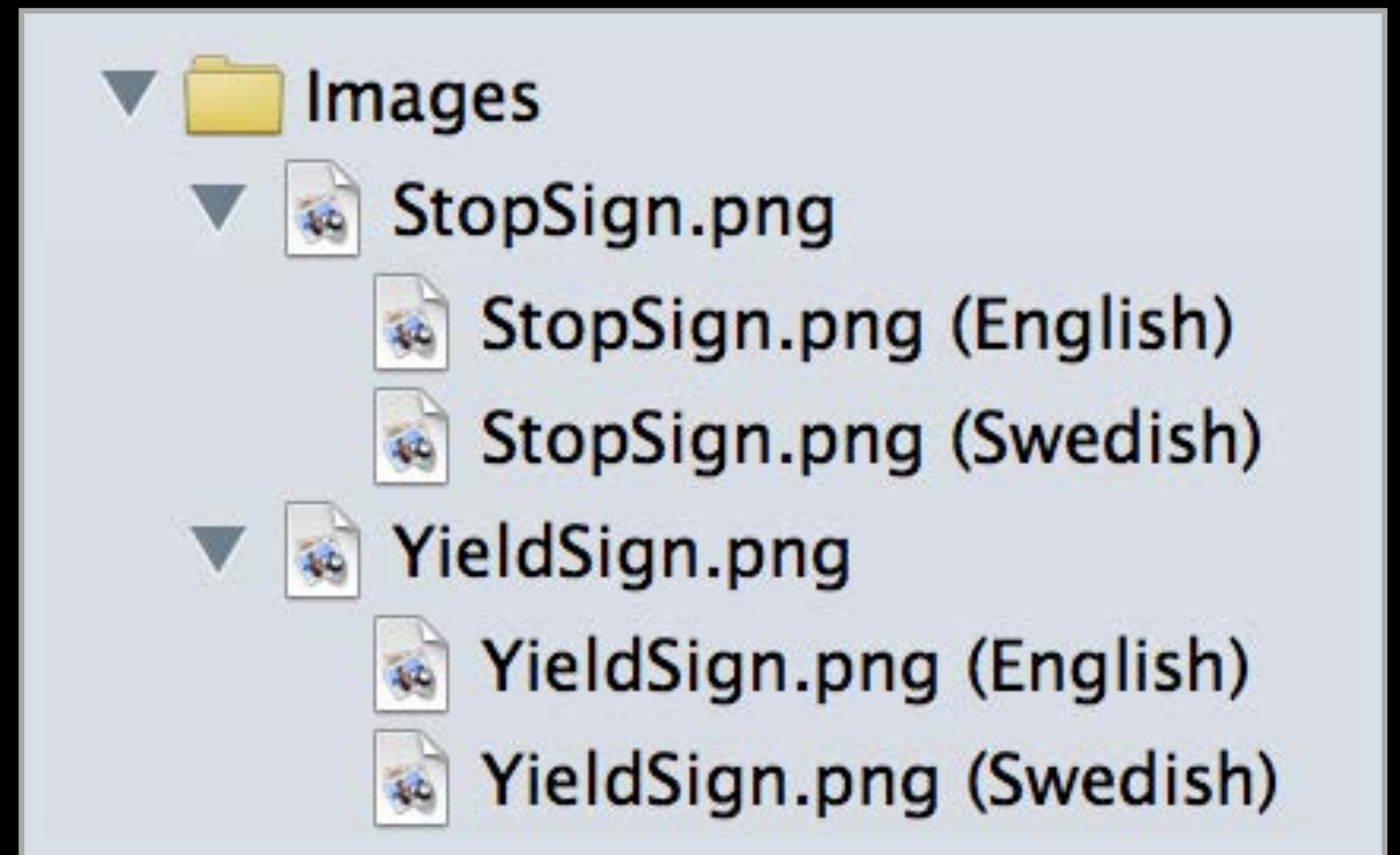
## Images



# Other Resources

Images

Sounds

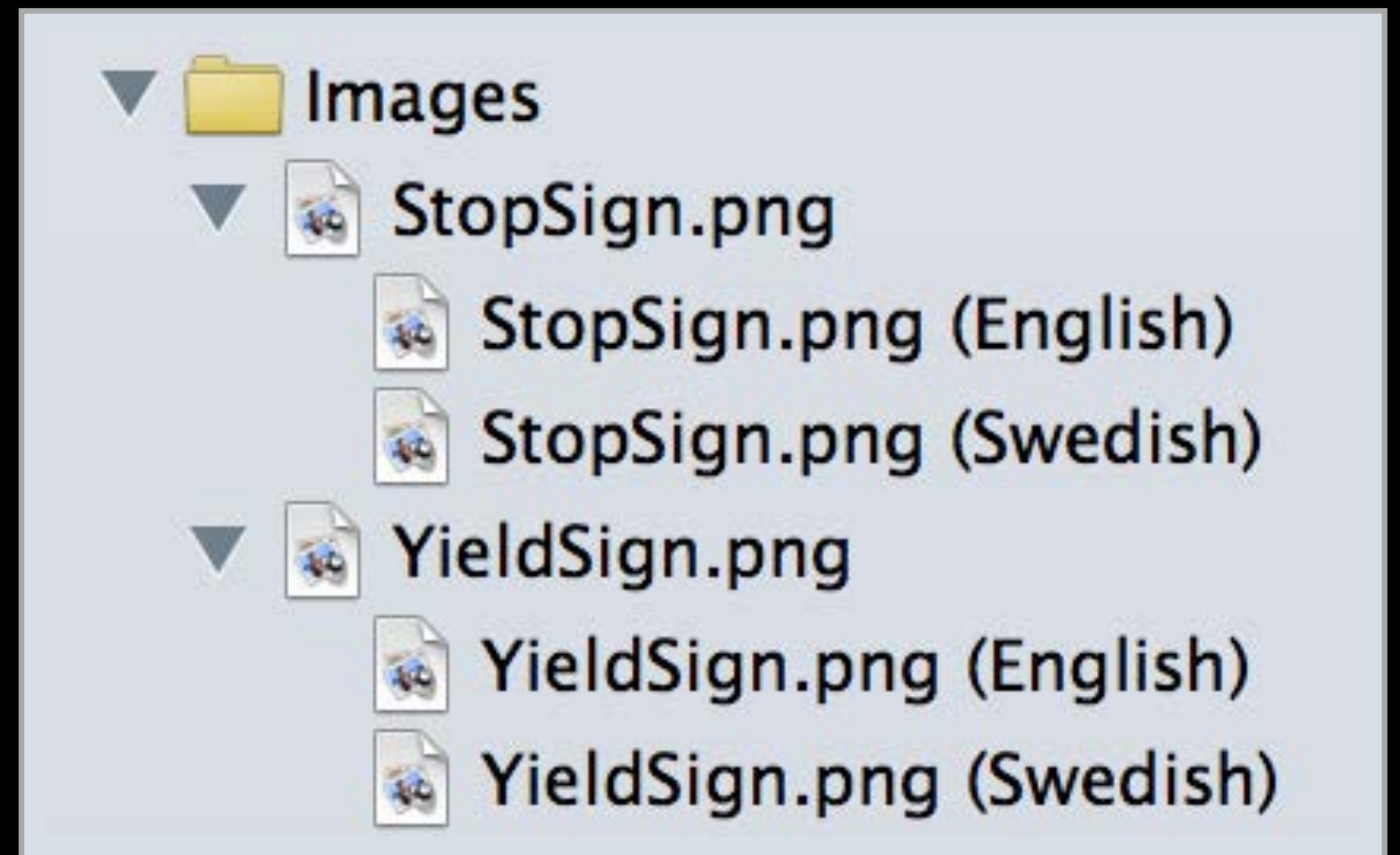


# Other Resources

Images

Sounds

Movies



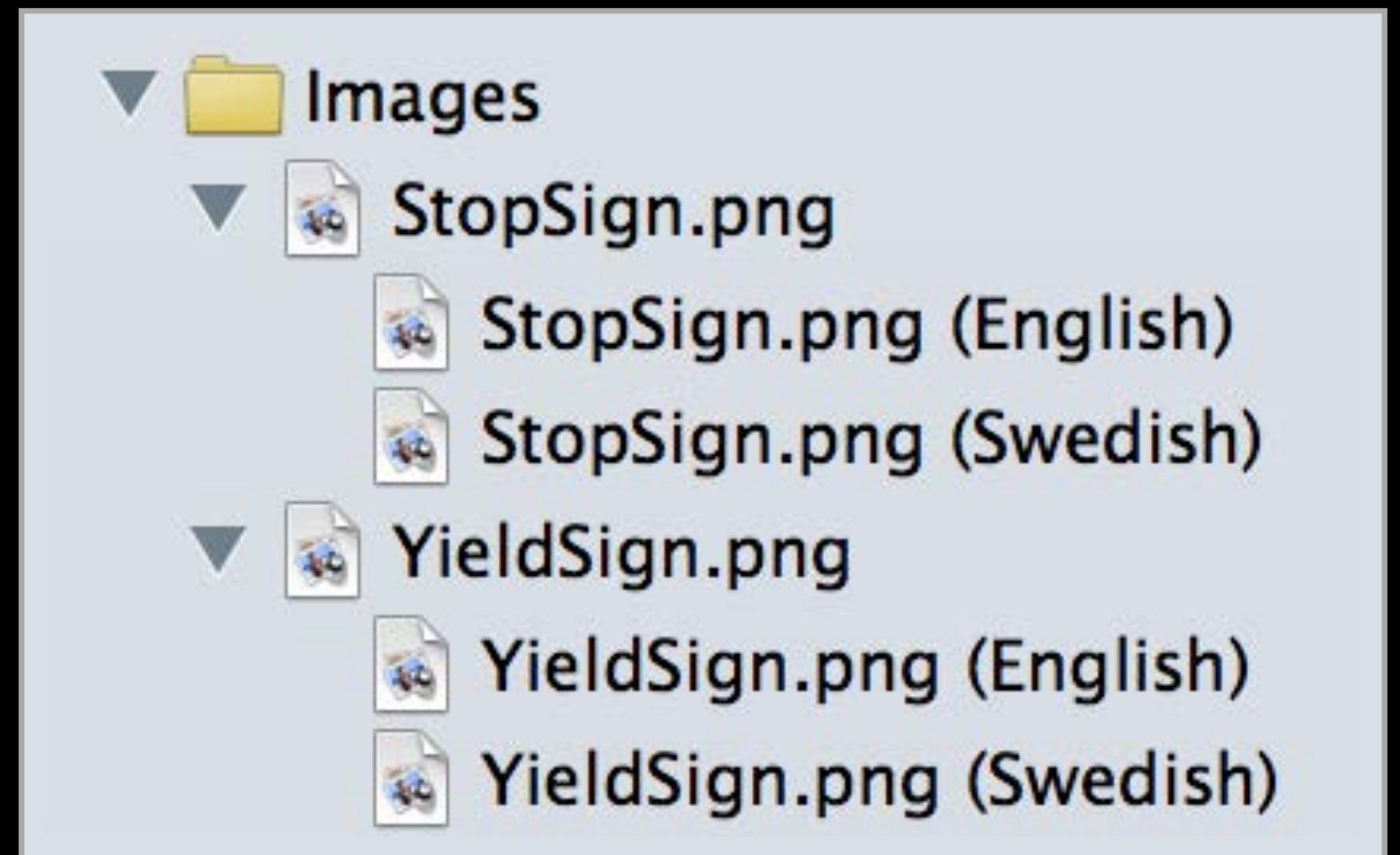
# Other Resources

Images

Sounds

Movies

Documentation



# Other Resources

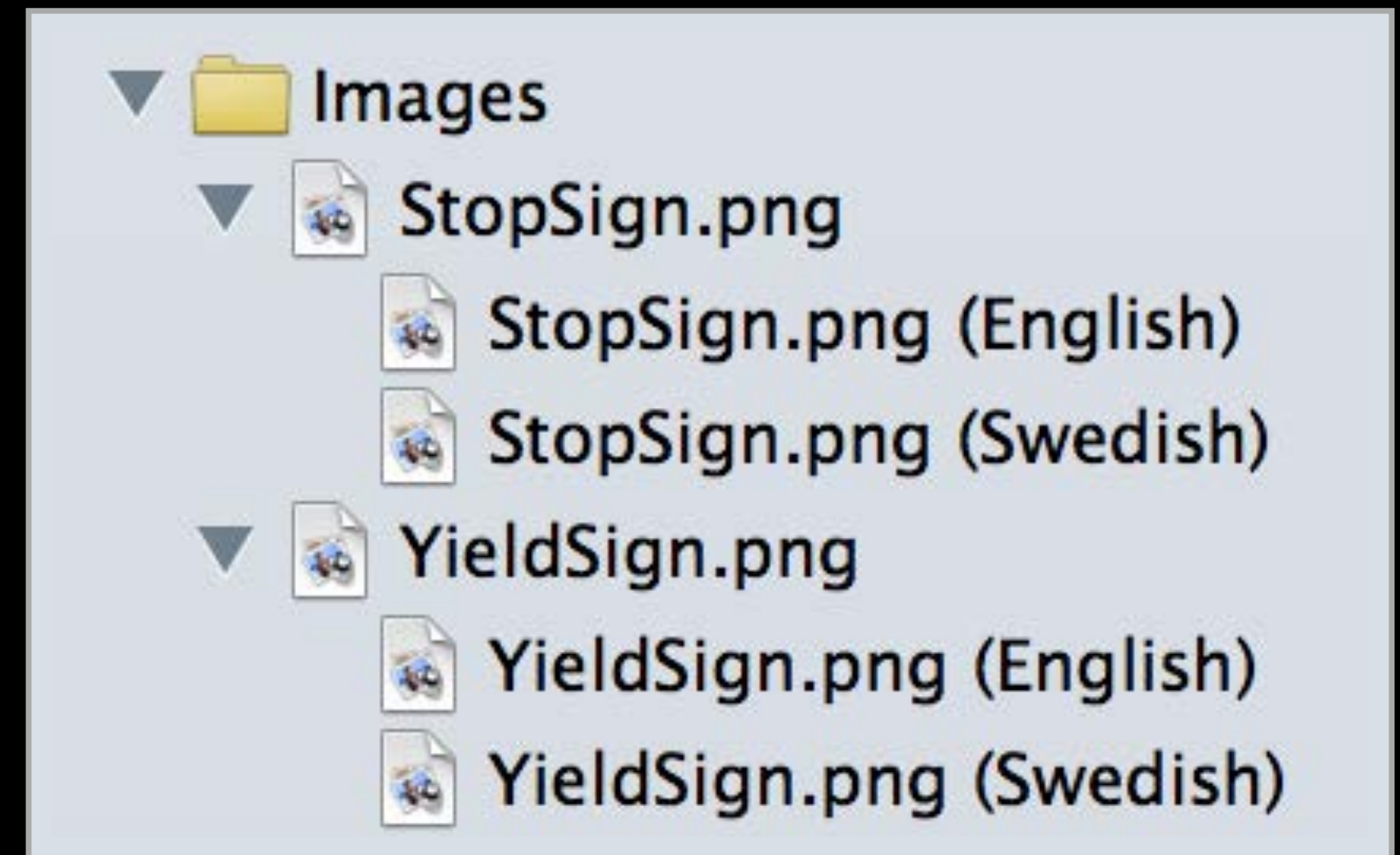
Images

Sounds

Movies

Documentation

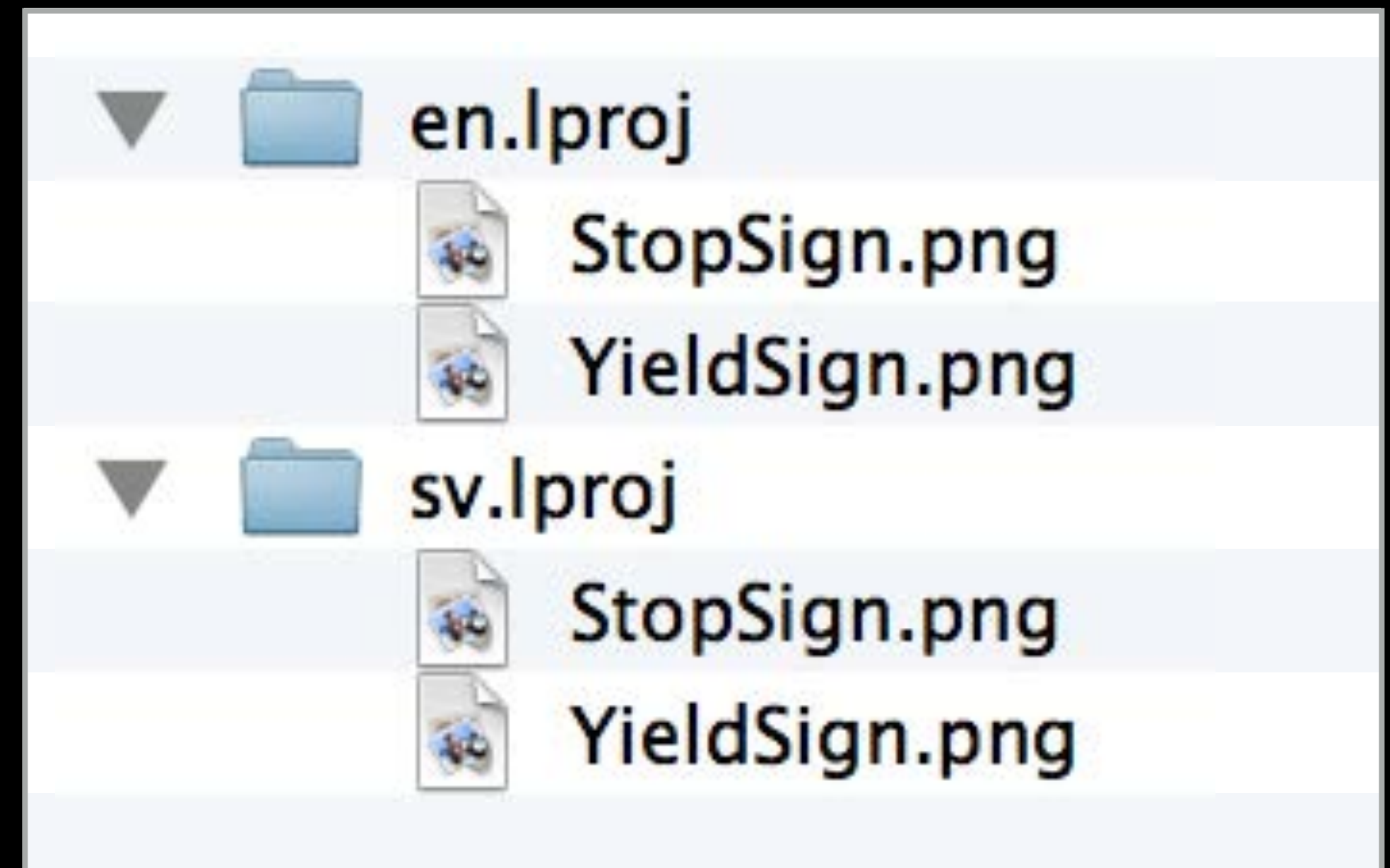
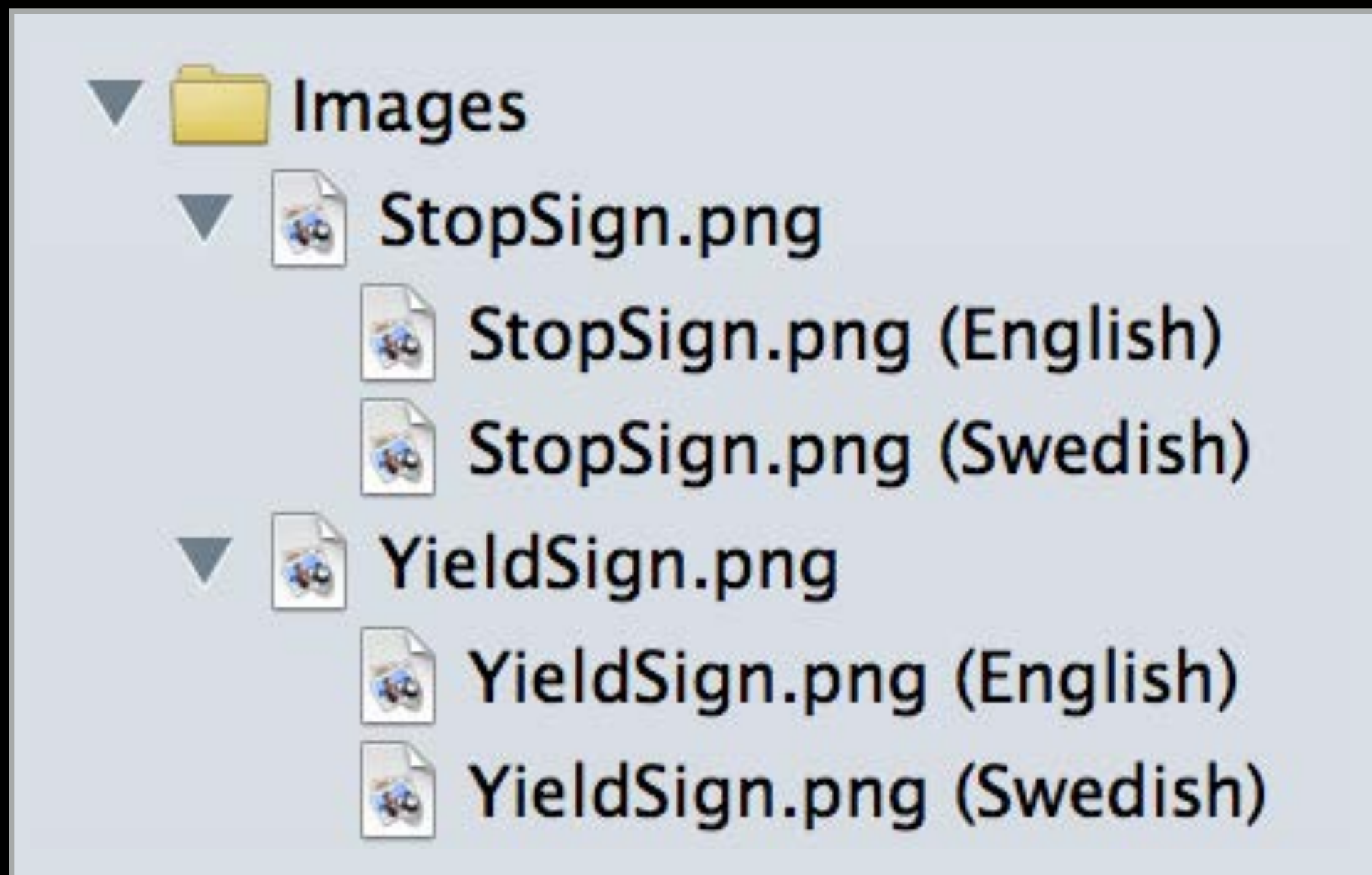
...any other kind of resource, really!





# Other Resources

## Xcode and your App



# Debug-Time Preview

NEW

SeismicXML > iPhone 5s

Build  
2 targets

Run SeismicXML.app  
Debug

Test  
Debug

Profile SeismicXML.app  
Release

Analyze  
Debug

Archive  
Release

Info Arguments **Options** Diagnostics

Core Location  Allow Location Simulation  
Default Location None

Application Data None

Routing App Coverage File None

GPU Frame Capture Automatically Enabled

Background Fetch

Localization Debugging

Application Language  Swedish

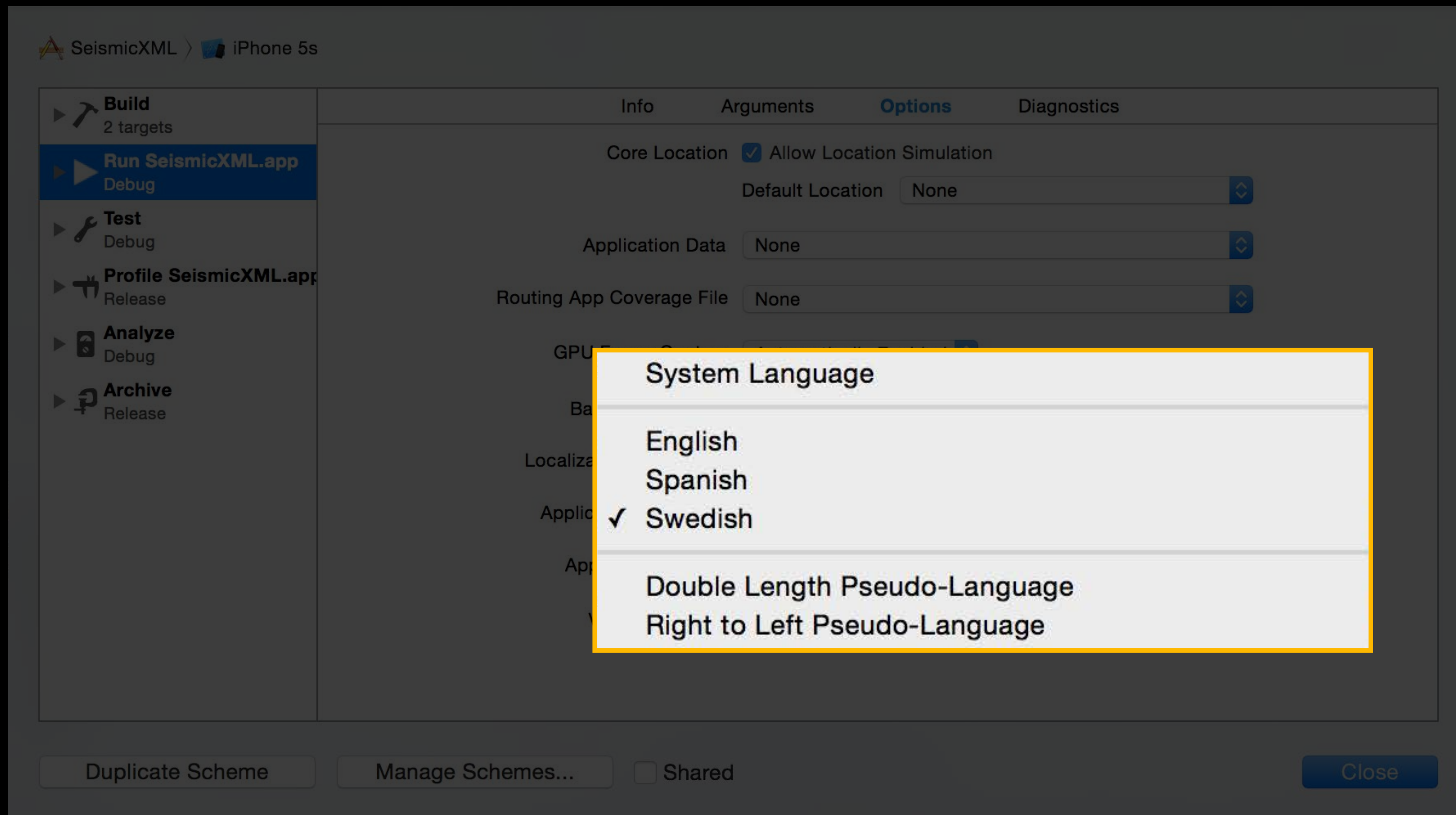
Application Region

View Debugging  Enable user interface debugging

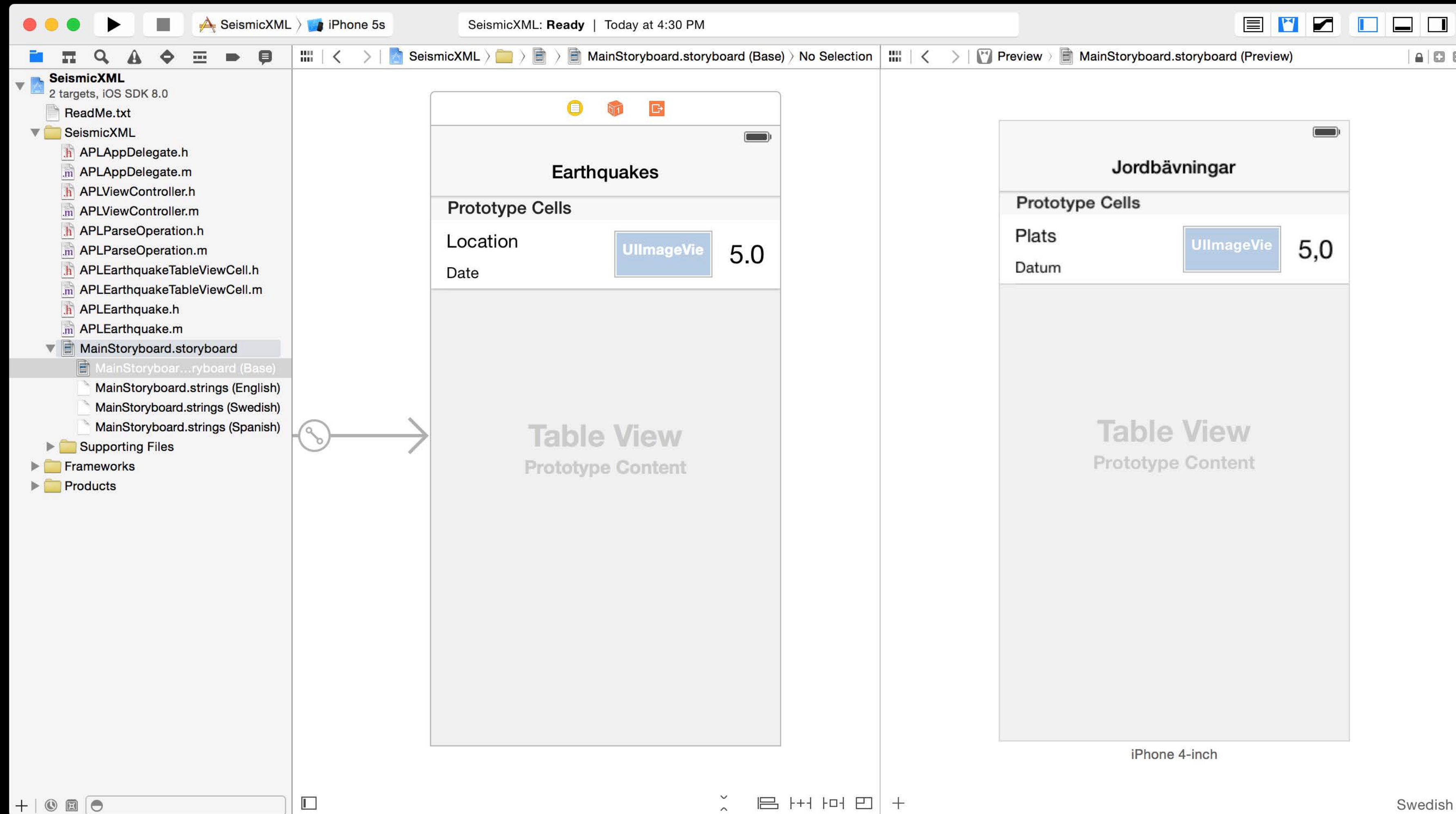
Duplicate Scheme Manage Schemes...  Shared Close

# Debug-Time Preview

NEW

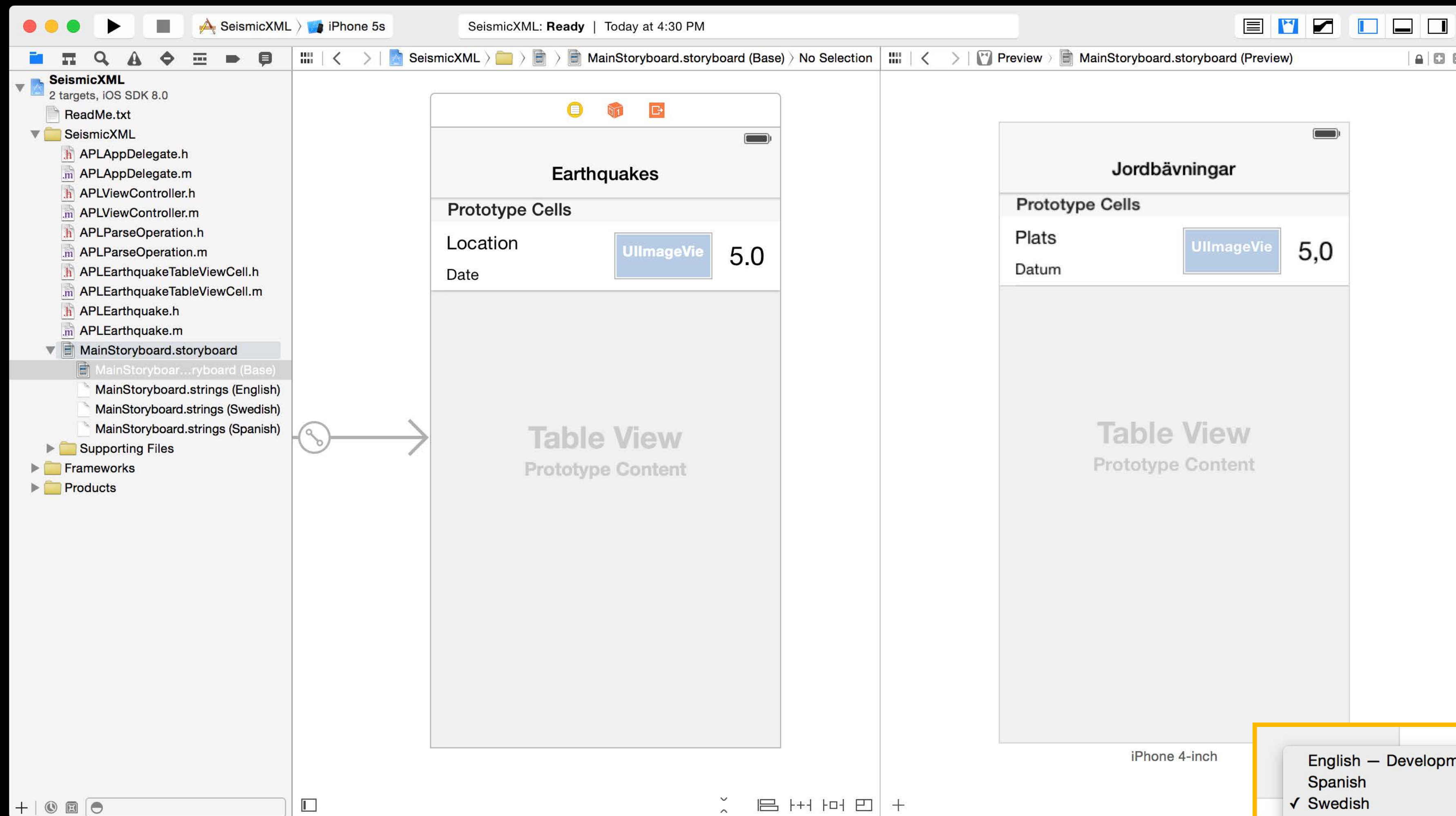


# Design-Time Preview



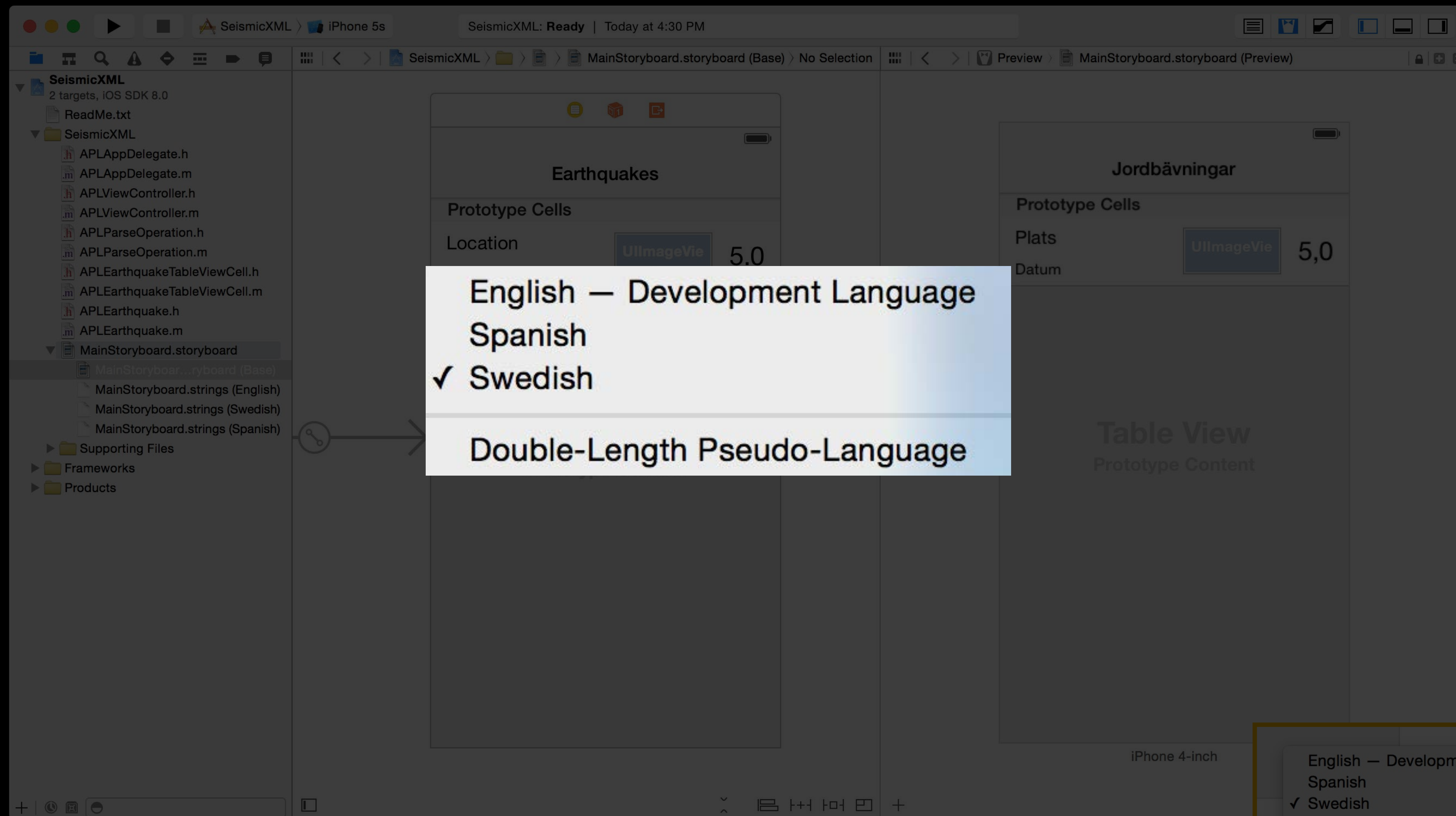
# Design-Time Preview

NEW



# Design-Time Preview

NEW



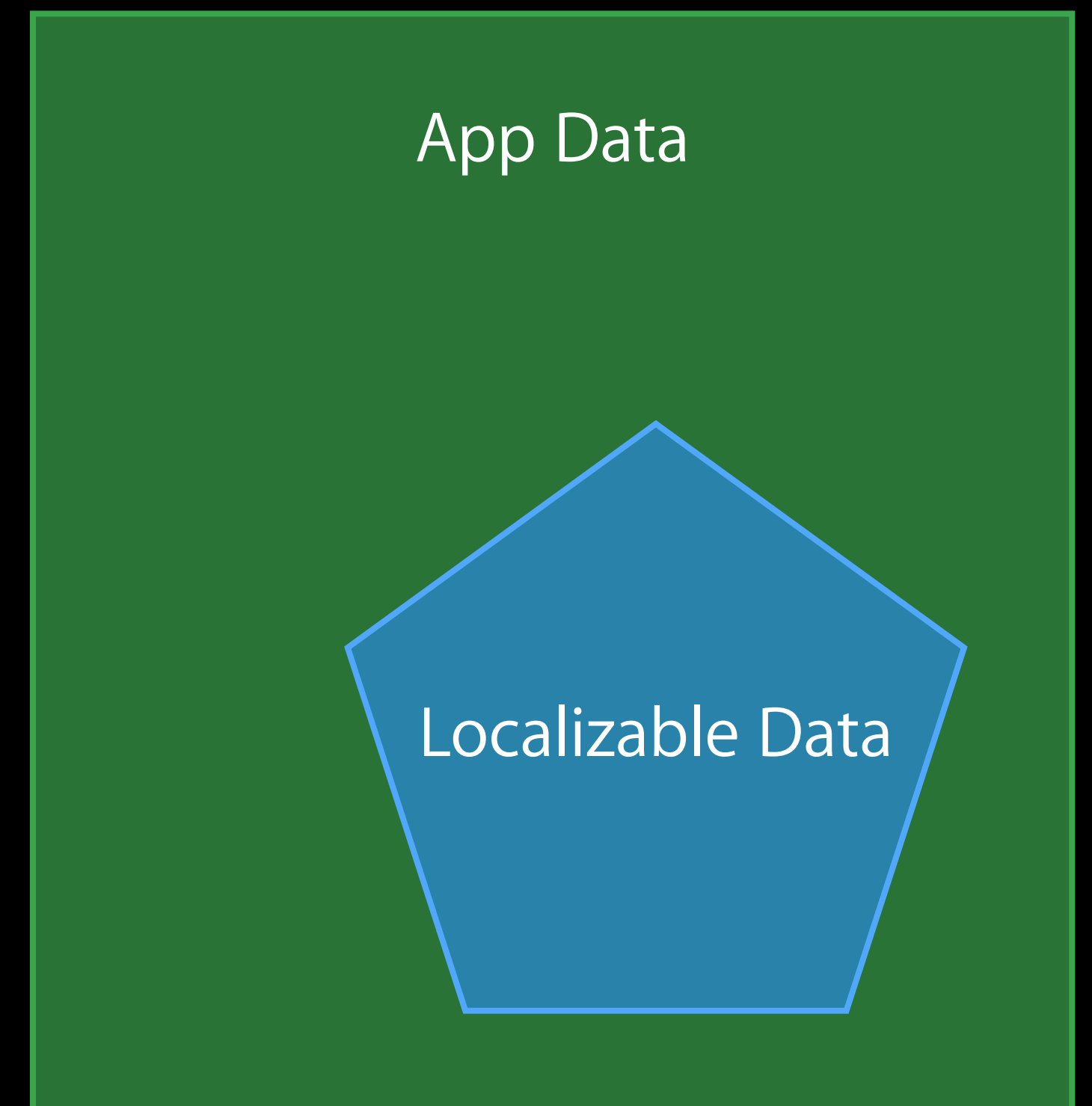
# Internationalization

Identify strings for translations

Use base internationalization and Auto Layout

Localize other project resources

Step 1



*Demo*

Internationalizing your project

Chris Hanson

Xcode Software Engineer



# Internationalization Summary

# Internationalization Summary

Let the frameworks do the heavy lifting in your code

# Internationalization Summary

Let the frameworks do the heavy lifting in your code

Use the debug-time preview to check region support

# Internationalization Summary

Let the frameworks do the heavy lifting in your code

Use the debug-time preview to check region support

Take advantage of design-time preview via base internationalization

# Internationalization Summary

Let the frameworks do the heavy lifting in your code

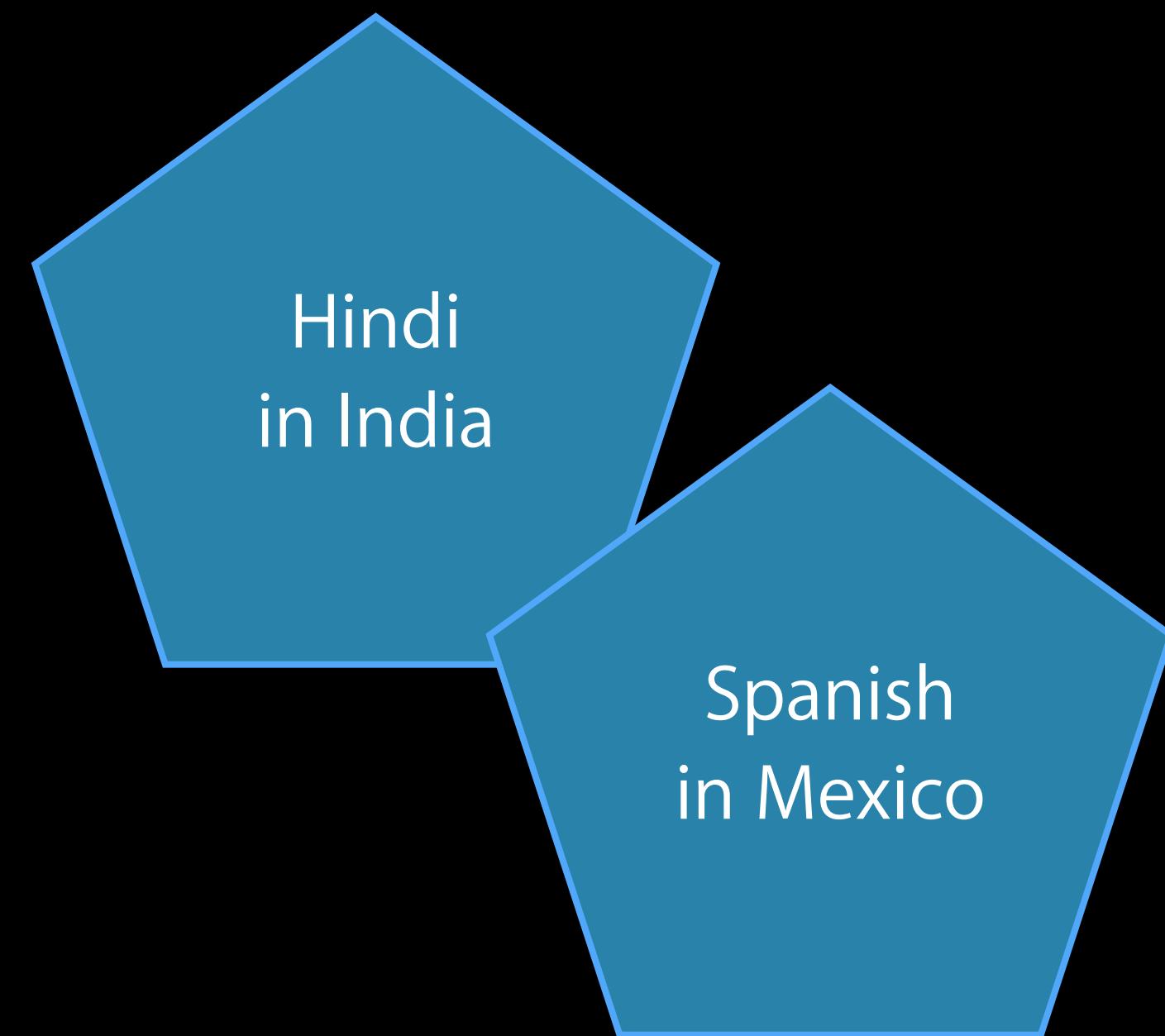
Use the debug-time preview to check region support

Take advantage of design-time preview via base internationalization

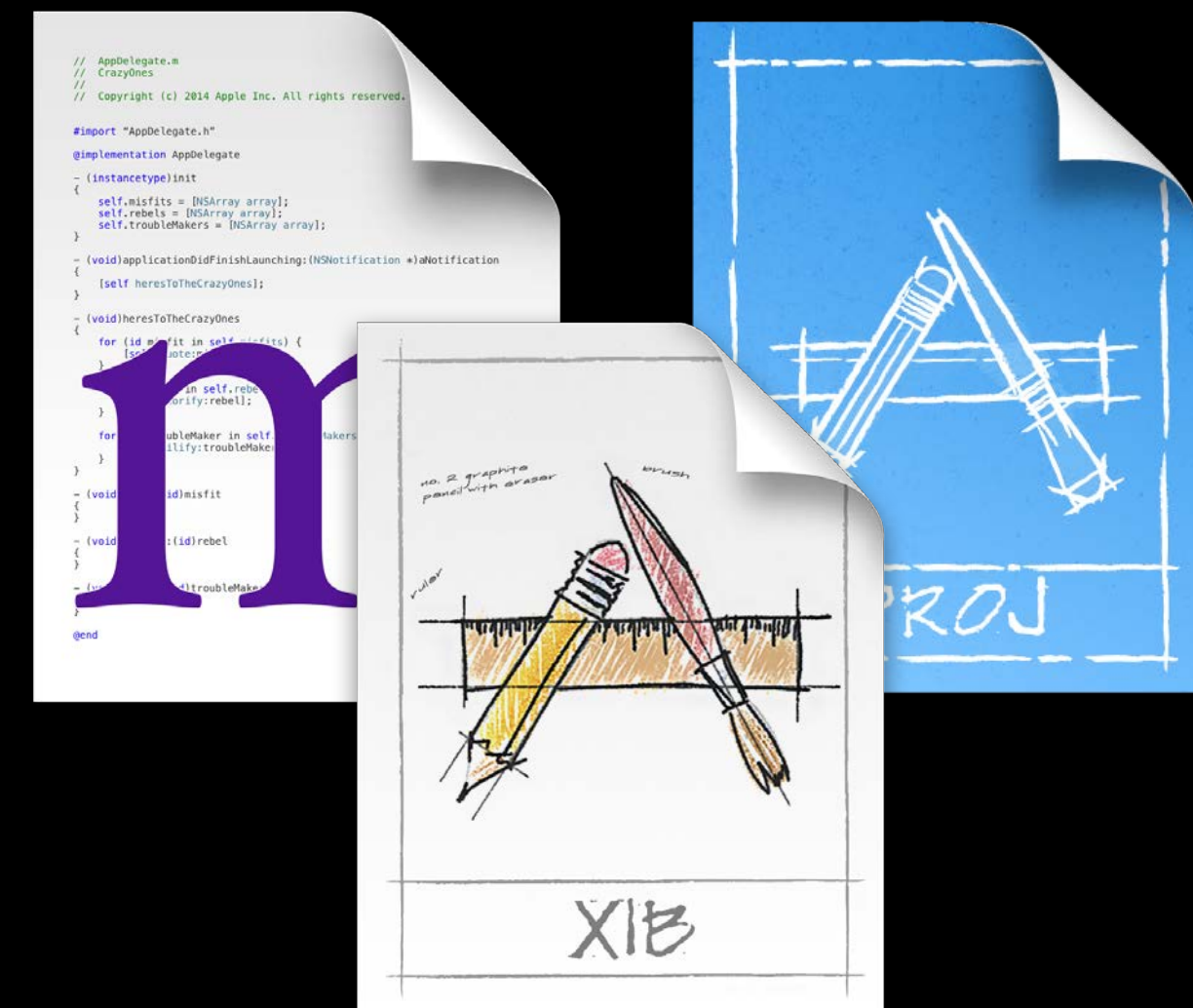
Localize other project resources as necessary

# Localization

Translating Your App

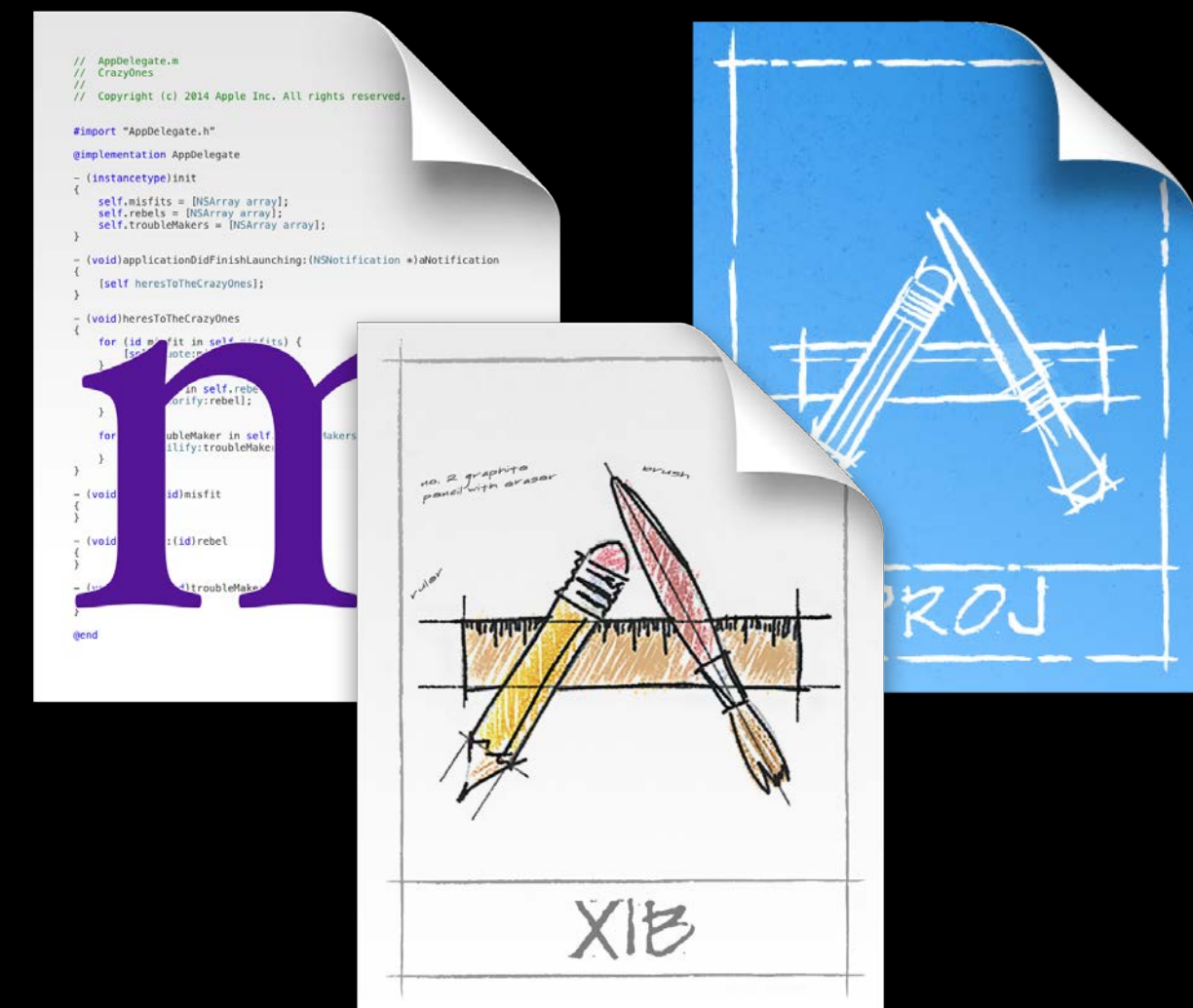


# User Facing Strings



# User Facing Strings

How to gather from your project?

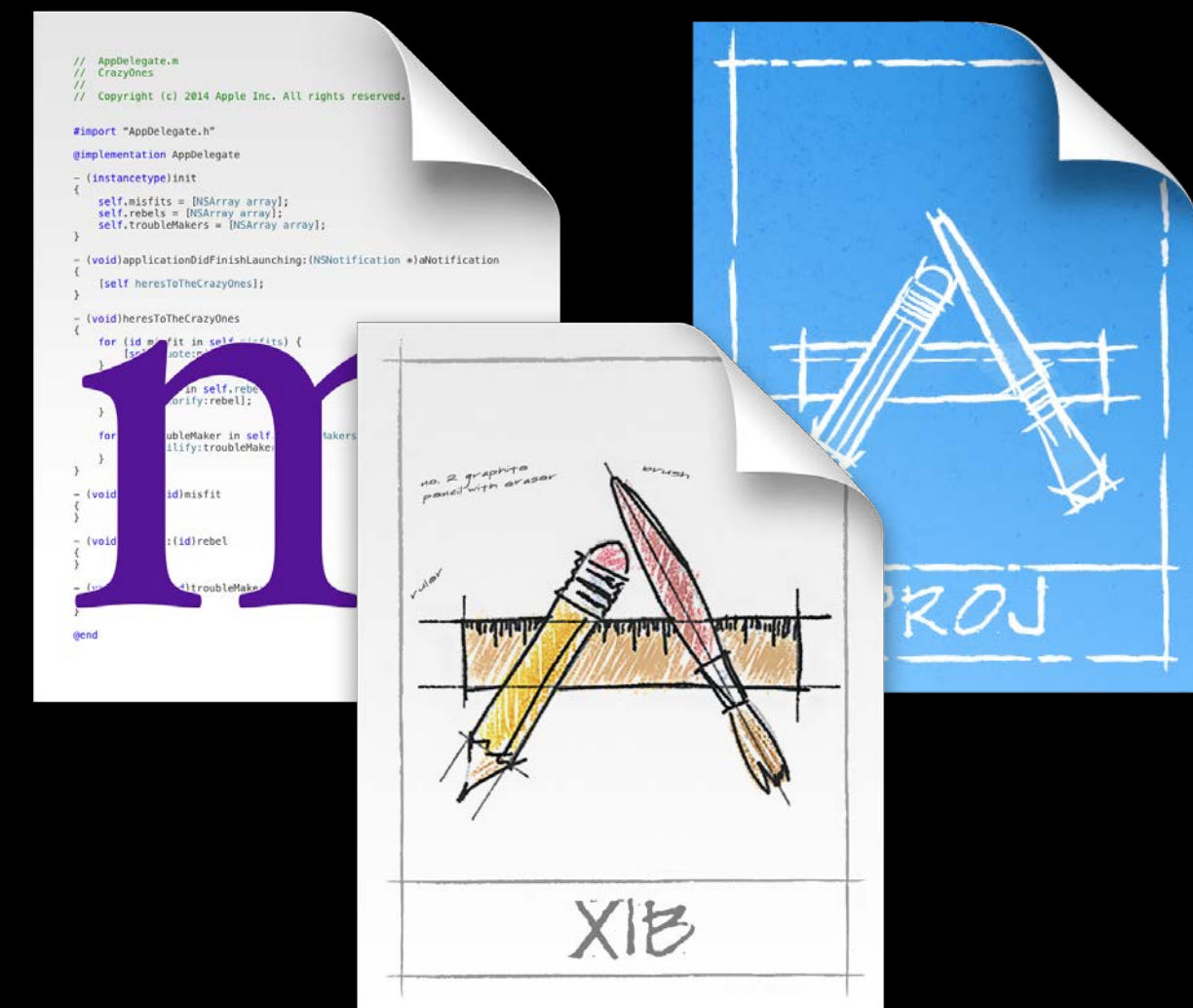




# User Facing Strings

How to gather from your project?

How to insert translations?

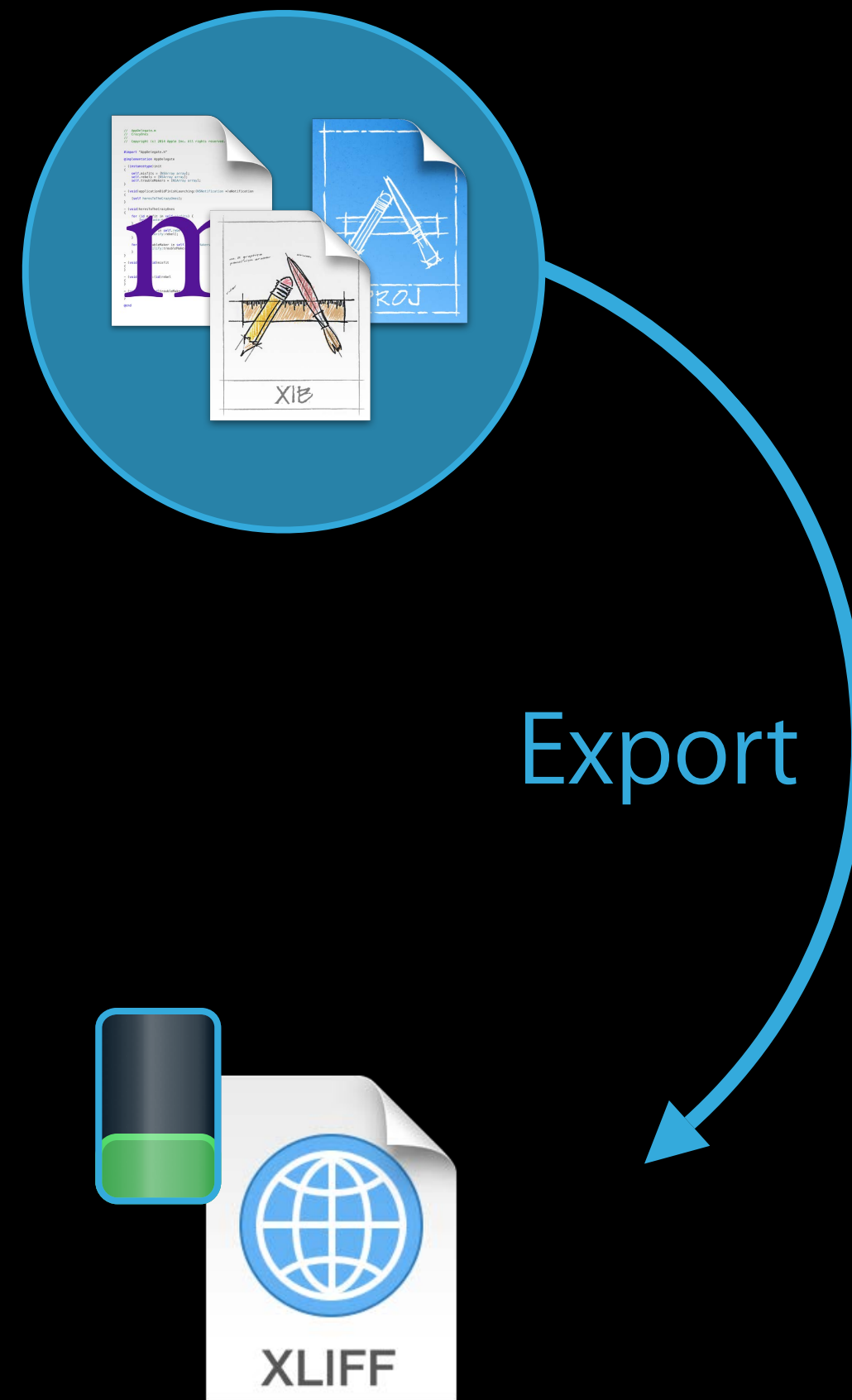


# Export and Import

Exchange user facing strings easily



Project Resources

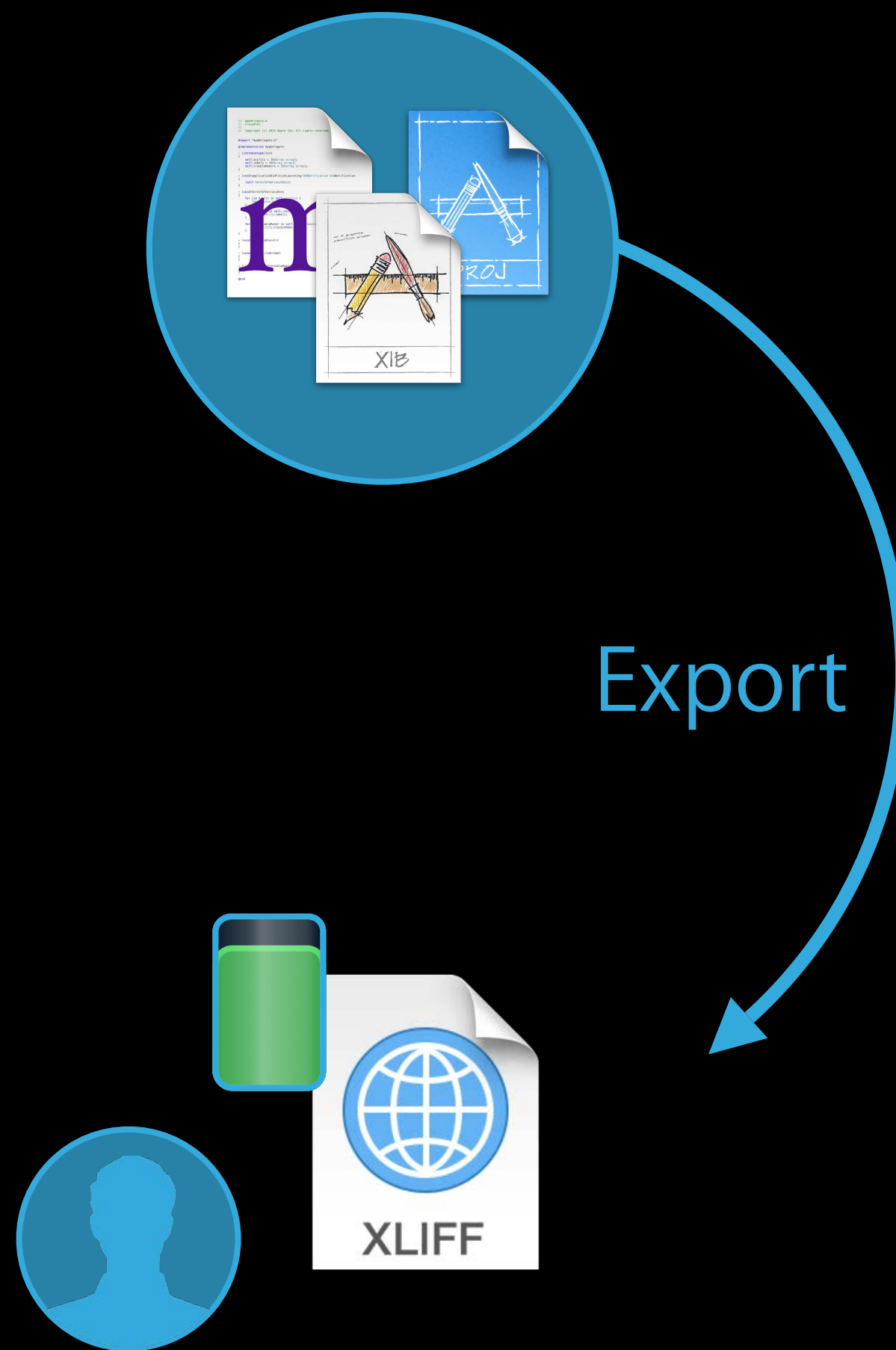


# Export and Import

Exchange user facing strings easily



Project Resources

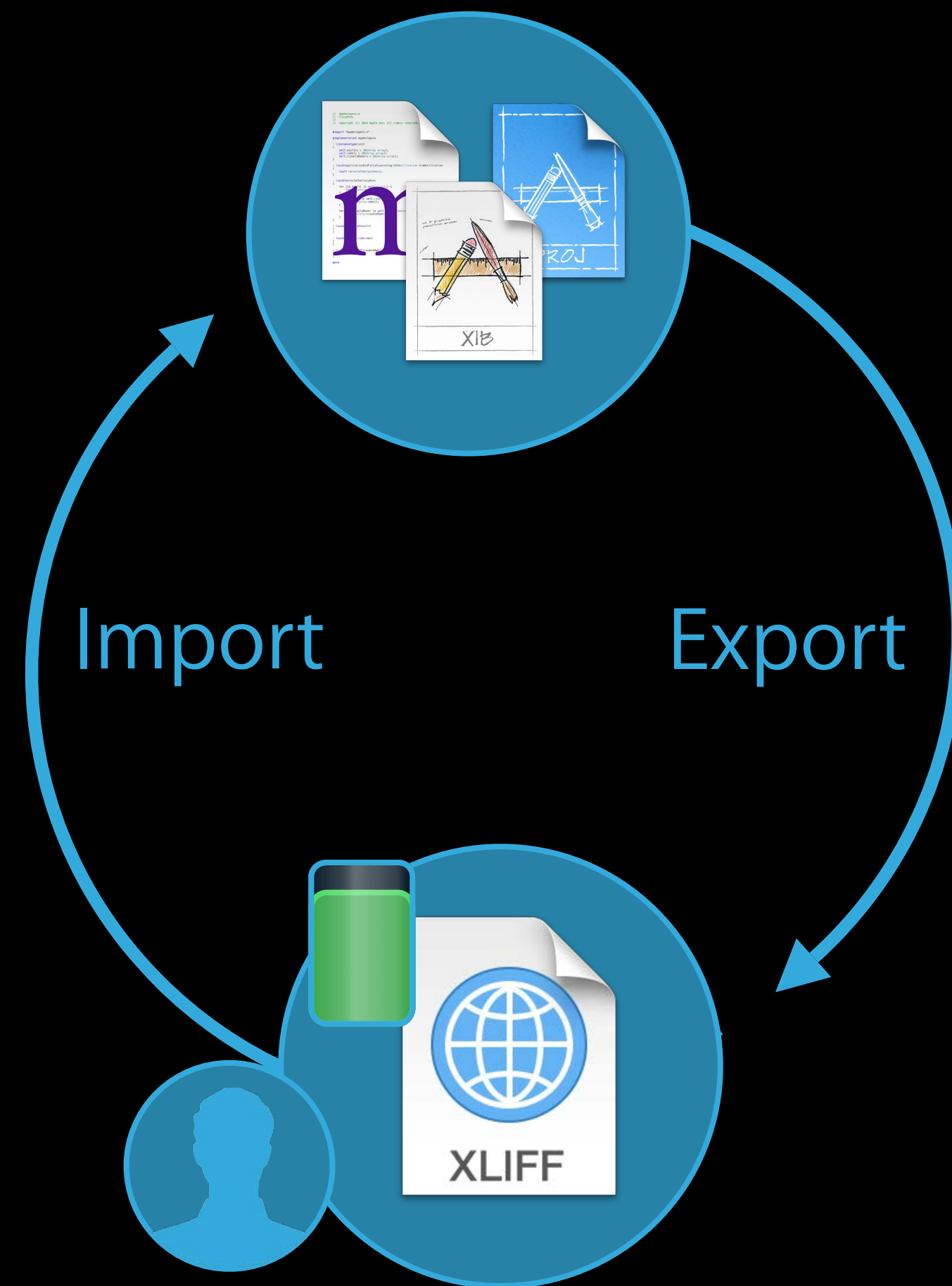


# Export and Import

Exchange user facing strings easily



Project Resources



Transport Format

# *Demo*

Localizing your project

Zoltan Foley-Fisher

Xcode Software Engineer

# Command Line

```
xcodebuild -exportLocalizations -project <project> -localizationPath <path>
```

```
xcodebuild -importLocalizations -project <project> -localizationPath <path>
```

# Summary

# Summary

Export and import Interface Builder and source strings



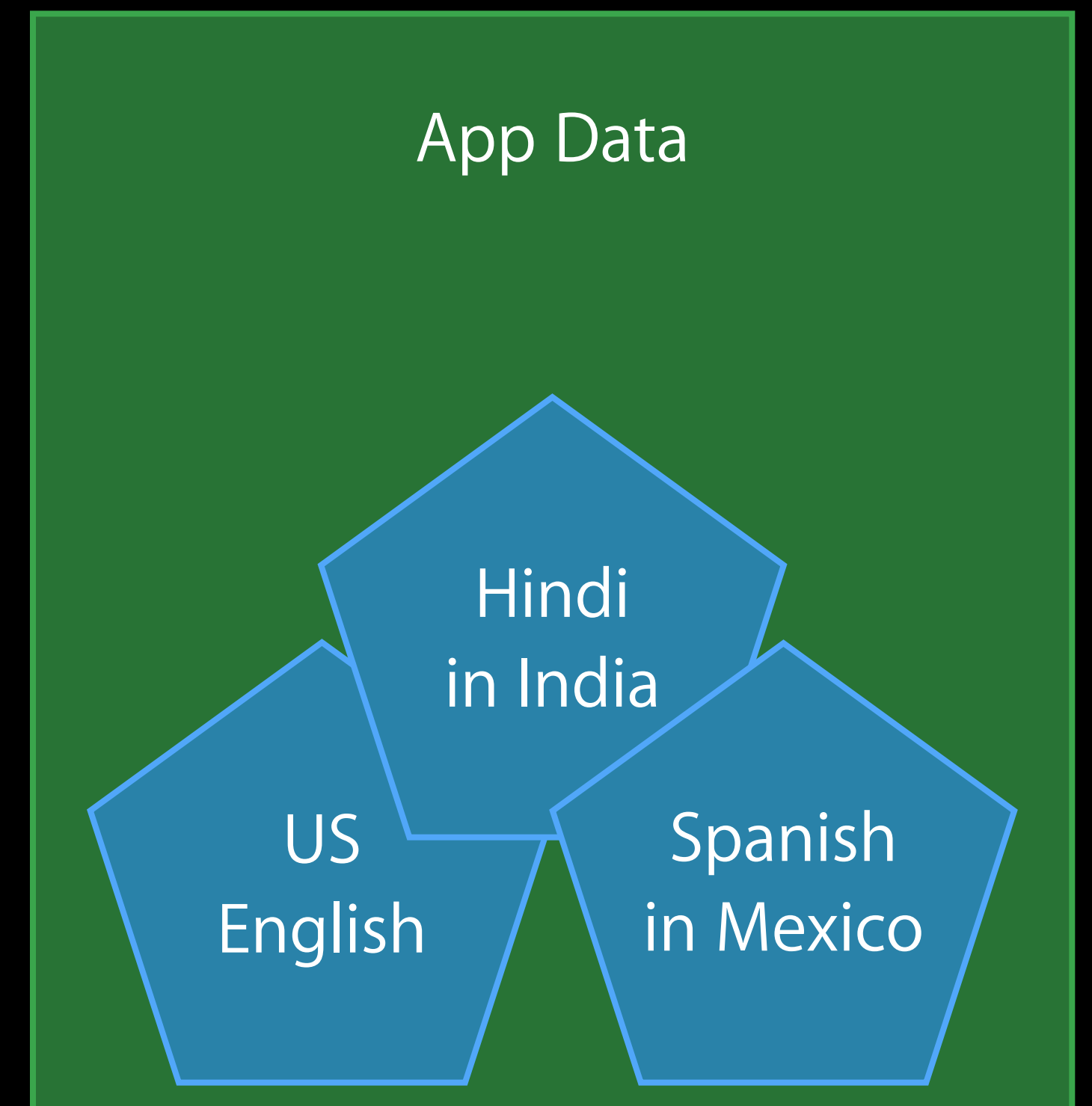
# Summary

Export and import Interface Builder and source strings

Exchange strings in the widely accepted XLIFF format

# Iteration

Updating your app



# Iterate on Your App

# Iterate on Your App

No need to put off localization to the end—Export and import at will

# Iterate on Your App

No need to put off localization to the end—Export and import at will

Stay localized as you work, using previews to catch missing translations easily

# Iterate on Your App

No need to put off localization to the end—Export and import at will

Stay localized as you work, using previews to catch missing translations easily

Prepare your app for right-to-left languages independently of translation

# *Demo*

Updating a localized project

Chris Hanson

Xcode Software Engineer

# Summary



# Summary

Update translations as you go to keep your app current

# Summary

Update translations as you go to keep your app current

Catch missing translations before your users

# Summary

Update translations as you go to keep your app current

Catch missing translations before your users

Simulate many different behaviors to ensure your apps work for everyone

Reach a Global Audience

# Reach a Global Audience

Apple's frameworks and tools make it easy!

# Reach a Global Audience

Apple's frameworks and tools make it easy!

Take advantage of Xcode's new workflows

# Reach a Global Audience

Apple's frameworks and tools make it easy!

Take advantage of Xcode's new workflows

- XLIFF export and import

# Reach a Global Audience

Apple's frameworks and tools make it easy!

Take advantage of Xcode's new workflows

- XLIFF export and import
- Localization previews



Try It Out This Week!

# Related Sessions

- 
- Advanced Topics in Internationalization Russian Hill Tuesday 9:00AM
  - Apps for China Get Together Folsom Wednesday 4:30PM
-

# Labs

- 
- Xcode and Localization Lab Tools Lab C      Tuesday 2:00PM
  - Internationalization Lab Frameworks Lab B    Tuesday 3:15PM
-

 WWDC14