Localizing with Xcode 6
Best practices and new workflows

Session 412
Zoltan Foley-Fisher
Xcode Software Engineer

Chris Hanson
Xcode Software Engineer
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

App Data

Localizable Data

Step 2

Hindi in India

Spanish in Mexico
Languages and Regions

Earthquakes

California
May 20, 2014, 12:47 PM
3.5

Virgin Islands region
May 20, 2014, 5:09 PM
3.3

Virgin Islands region
May 20, 2014, 4:59 PM
2.7

Puerto Rico region
May 20, 2014, 10:16 AM
2.8

Puerto Rico region
May 20, 2014, 6:24 AM
2.8
Languages and Regions

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

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

Earthquakes

California
May 20, 2014, 12:47 PM
3.5

Virgin Islands region
May 20, 2014, 5:09 PM
3.3

Virgin Islands region
May 20, 2014, 4:59 PM
2.7

Puerto Rico region
May 20, 2014, 10:16 AM
2.8

Puerto Rico region
May 20, 2014, 6:24 AM
2.8

Terremotos

California
20/05/2014 12:47
3.5

Virgin Islands region
20/05/2014 17:09
3.3

Virgin Islands region
20/05/2014 16:59
2.7

Puerto Rico region
20/05/2014 10:16
2.8

Puerto Rico region
20/05/2014 06:24
2.8
Languages and Regions

Earthquakes

California
May 20, 2014, 12:47 PM
3.5

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

Terremotos

California
20/05/2014 12:47
3.5

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

Terremotos

California
20/5/2014 12:47
3,5

Virgin Islands region
20/5/2014 17:09
3,3
Virgin Islands region
20/5/2014 16:59
2,7
Puerto Rico region
20/5/2014 10:16
2,8
Puerto Rico region
20/5/2014 06:24
2,8
Languages and Regions

<table>
<thead>
<tr>
<th>Location</th>
<th>Date/Time</th>
<th>Magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>California</td>
<td>May 20, 2014, 12:47 PM</td>
<td>3.5</td>
</tr>
<tr>
<td>Virgin Islands</td>
<td>May 20, 2014, 5:09 PM</td>
<td>3.3</td>
</tr>
<tr>
<td>Virgin Islands</td>
<td>May 20, 2014, 4:59 PM</td>
<td>2.7</td>
</tr>
<tr>
<td>Puerto Rico</td>
<td>May 20, 2014, 10:16 AM</td>
<td>2.8</td>
</tr>
<tr>
<td>Puerto Rico</td>
<td>May 20, 2014, 6:24 AM</td>
<td>2.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>Date/Time</th>
<th>Magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>California</td>
<td>20/05/2014 12:47</td>
<td>3.5</td>
</tr>
<tr>
<td>Virgin Islands</td>
<td>20/05/2014 17:09</td>
<td>3.3</td>
</tr>
<tr>
<td>Virgin Islands</td>
<td>20/05/2014 16:59</td>
<td>2.7</td>
</tr>
<tr>
<td>Puerto Rico</td>
<td>20/05/2014 10:16</td>
<td>2.8</td>
</tr>
<tr>
<td>Puerto Rico</td>
<td>20/05/2014 06:24</td>
<td>2.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>Date/Time</th>
<th>Magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>California</td>
<td>20/5/2014 12:47</td>
<td>3.5</td>
</tr>
<tr>
<td>Virgin Islands</td>
<td>20/5/2014 17:09</td>
<td>3.3</td>
</tr>
<tr>
<td>Virgin Islands</td>
<td>20/5/2014 16:59</td>
<td>2.7</td>
</tr>
<tr>
<td>Puerto Rico</td>
<td>20/5/2014 10:16</td>
<td>2.8</td>
</tr>
<tr>
<td>Puerto Rico</td>
<td>20/5/2014 06:24</td>
<td>2.8</td>
</tr>
</tbody>
</table>
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
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
NSLocalizedString()
Apple developers should use `NSLocalizedString()` to internationalize strings for their interface in their source code.
NSLocalizedString()

Internationalize strings for your interface in your source code
Lets you keep “nice” strings in your code, including format strings
NSLocalizedString()

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
IntlNSLocalizedStringString()

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 NSLocalizedStringFromTable()
NSLocalizedString()

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

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

Converts between objects and human-readable text
NSFormatter

Converts between objects and human-readable text
For both presenting and interpreting text
NSFormatter

Converts between objects and human-readable text
For both presenting and interpreting text
Uses the current locale by default
NSDateFormatter *formatter = [[NSDateFormatter alloc] init];
formatter.dateStyle = NSDateFormatterMediumStyle;
formatter.timeStyle = NSDateFormatterNoStyle;
NSDateFormatter *formatter = [[NSDateFormatter alloc] init];
formatter.dateStyle = NSDateFormatterMediumStyle;
formatter.timeStyle = NSDateFormatterNoStyle;
NSDateFormatter *formatter = [[NSDateFormatter alloc] init];
formatter.dateStyle = NSDateFormatterMediumStyle;
formatter.timeStyle = NSDateFormatterNoStyle;
NSDateFormatter *formatter = [[NSDateFormatter alloc] init];
formatter.dateFormat = NSDateFormatterMediumStyle;
formatter.timeStyle = NSDateFormatterNoStyle;

NSDate *date = mountain.latestClimb.date;
label.text = [formatter stringFromDate:date];
let formatter = NSDateFormatter()
formatter.dateStyle = .MediumStyle
formatter.timeStyle = .NoStyle

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

Dates, date components, and date/time intervals
NSFormatter

Dates, date components, and date/time intervals
Numbers, currency, and byte counts
NSFormatter

Dates, date components, and date/time intervals
Numbers, currency, and byte counts
Health-related quantities
NSFormatter

Dates, date components, and date/time intervals
Numbers, currency, and byte counts
Health-related quantities
  • Energy
NSFormatter

Dates, date components, and date/time intervals
Numbers, currency, and byte counts
Health-related quantities
  • Energy
  • Length
NSFormatter

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

```objective-c
NSURL *imageURL =
    [[NSBundle mainBundle] URLForResource:@"GameMenu" withExtension:@"png"];

UIImage *image = [UIImage imageWithContentsOfFile:imageURL.path];

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

NSURL *welcomeURL =
    [[NSBundle mainBundle] URLForResource:@"Welcome" withExtension:@"m4a"];

AVAudioPlayer *player =
    [[AVAudioPlayer alloc] initWithContentsOfURL:welcomeURL error:&error];
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

- Images
  - StopSign.png
    - StopSign.png (English)
    - StopSign.png (Swedish)
  - YieldSign.png
    - YieldSign.png (English)
    - YieldSign.png (Swedish)
Other Resources

Images

- **StopSign.png**
  - StopSign.png (English)
  - StopSign.png (Swedish)
- **YieldSign.png**
  - YieldSign.png (English)
  - YieldSign.png (Swedish)
Other Resources

Images

Sounds

- StopSign.png
  - StopSign.png (English)
  - StopSign.png (Swedish)

- YieldSign.png
  - YieldSign.png (English)
  - YieldSign.png (Swedish)
Other Resources

Images
Sounds
Movies
Other Resources

Images
Sounds
Movies
Documentation

Images
- StopSign.png
  - StopSign.png (English)
  - StopSign.png (Swedish)
- YieldSign.png
  - YieldSign.png (English)
  - YieldSign.png (Swedish)
Other Resources

Images
Sounds
Movies
Documentation

...any other kind of resource, really!
Other Resources

Xcode and your App
Debug-Time Preview

System Language

- English
- Spanish
- Swedish

Double Length Pseudo-Language
Right to Left Pseudo-Language
Design-Time Preview
Design-Time Preview
Design-Time Preview
Internationalization

Identify strings for translations
Use base internationalization and Auto Layout
Localize other project resources

Step 1
App Data
Localizable Data
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
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

- Hindi in India
- Spanish in Mexico
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

XLIFF
Export and Import
Exchange user facing strings easily

Project Resources

Export

XLIFF
Export and Import
Exchange user facing strings easily

Project Resources
Import
Export
Transport Format

Export
Project Resources
Import
Demo
Localizing your project

Zoltan Foley-Fisher
Xcode Software Engineer
Command Line

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

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

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Topics in Internationalization</td>
<td>Russian Hill</td>
<td>Tuesday 9:00AM</td>
</tr>
<tr>
<td>Apps for China Get Together</td>
<td>Folsom</td>
<td>Wednesday 4:30PM</td>
</tr>
<tr>
<td>Labs</td>
<td>Tools Lab C</td>
<td>Tuesday 2:00PM</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Xcode and Localization Lab</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internationalization Lab</td>
<td>Frameworks Lab B</td>
<td>Tuesday 3:15PM</td>
</tr>
</tbody>
</table>