Xamarin Evolve 2014

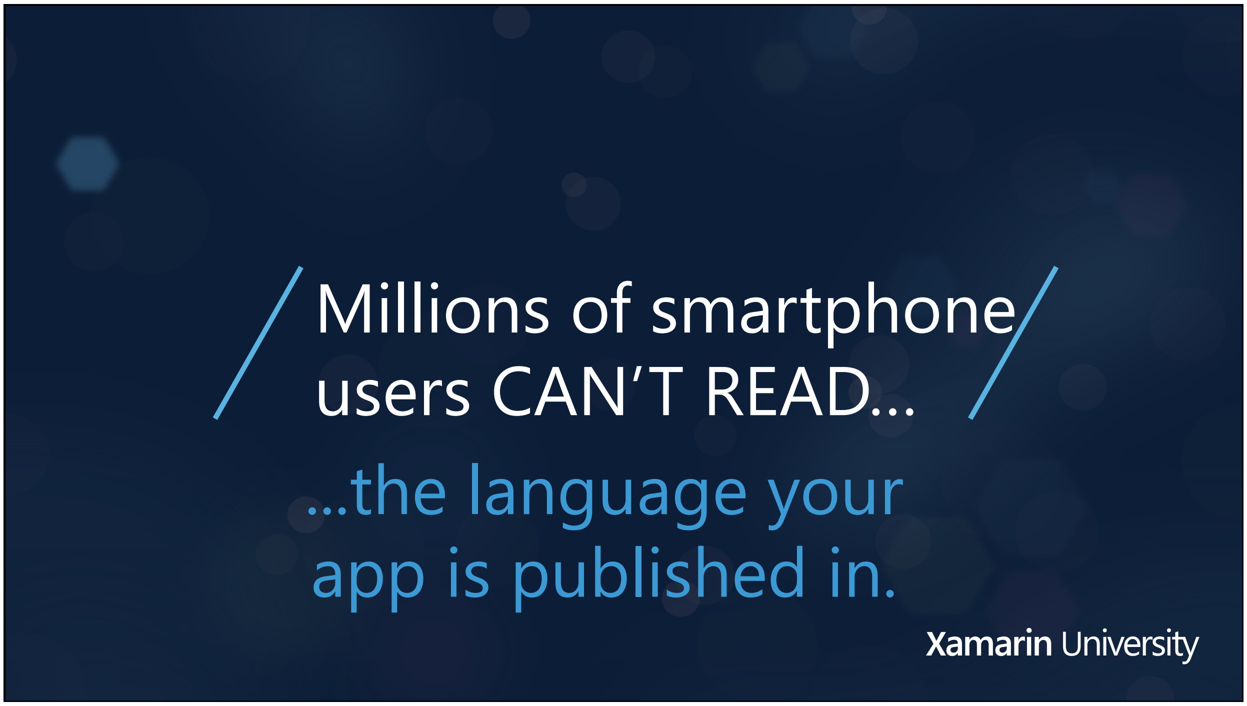
Localization Techniques

Craig Dunn
craig@xamarin.com

Xamarin University

Millions of smartphone users CAN'T READ…

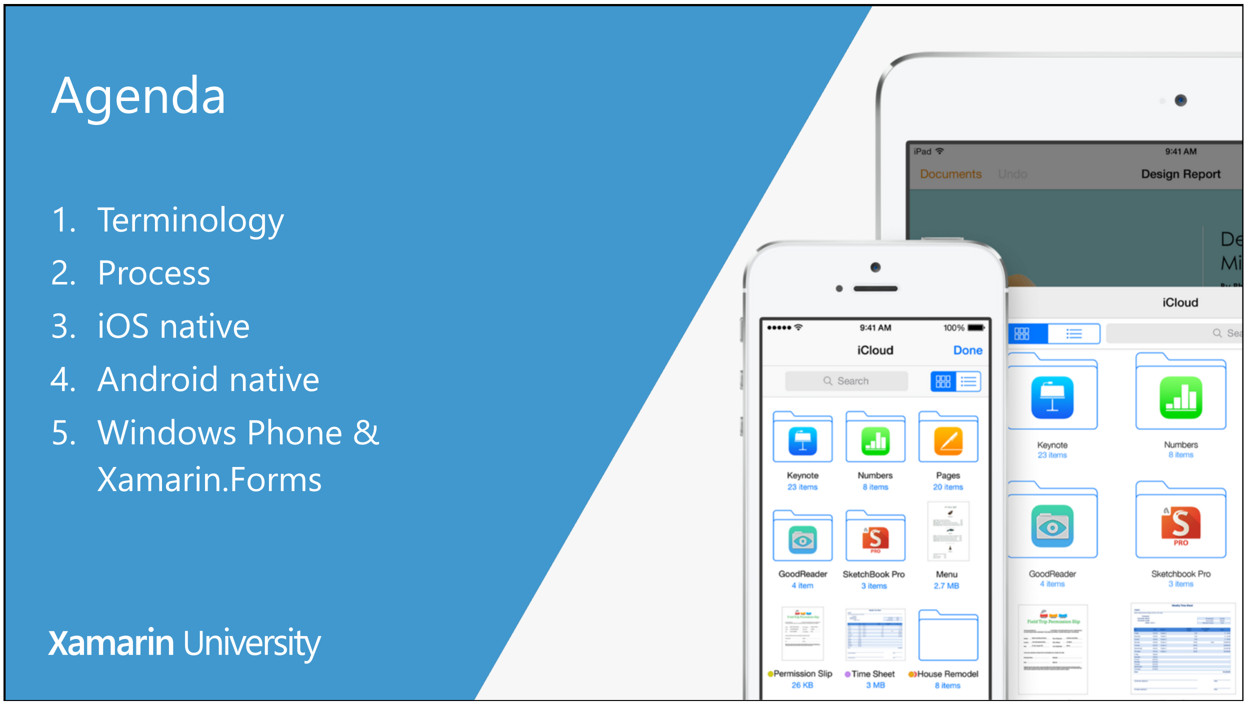
Xamarin University



Millions of smartphone users CAN'T READ...

...the language your app is published in.

Xamarin University



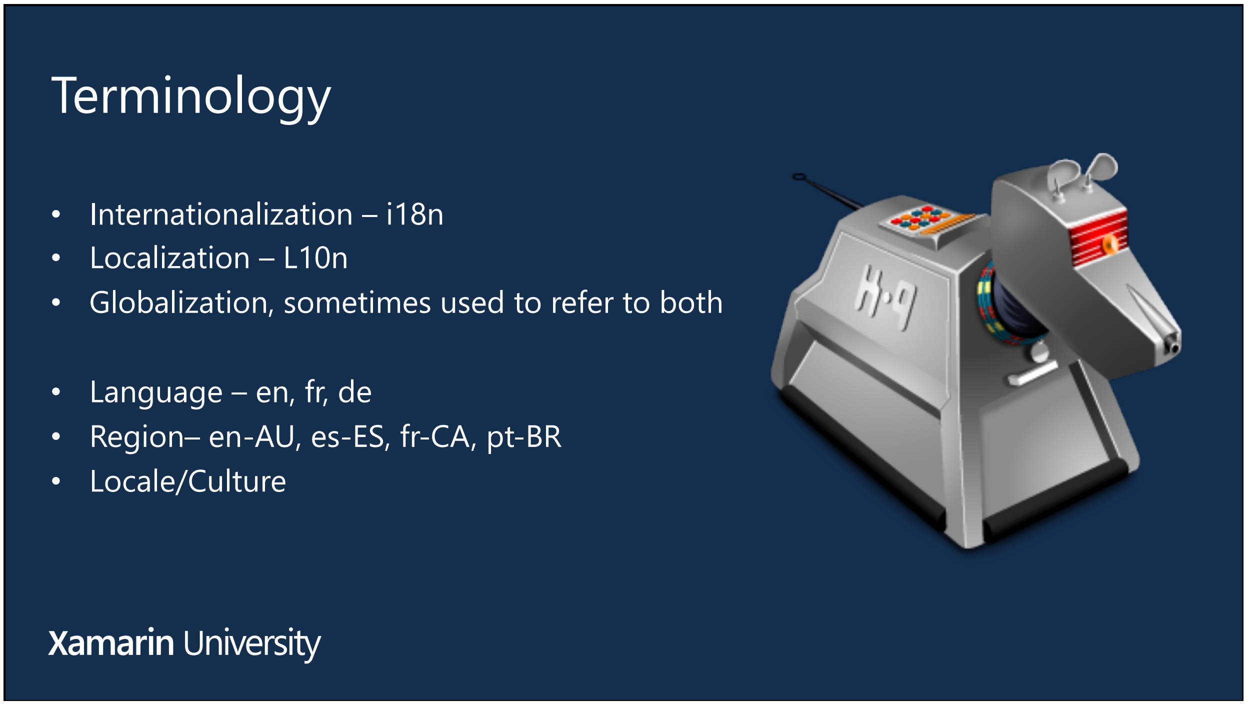
Agenda

1. Terminology
2. Process
3. iOS native
4. Android native
5. Windows Phone & Xamarin.Forms

Xamarin University

iPad
9:41 AM
Documents
Undo
Design Report
iCloud
9:41 AM
100%
iCloud
Done
Search
Keynote
23 items
Numbers
8 items
Pages
20 items
GoodReader
4 item
SketchBook Pro
3 items
Menu
2.7 MB
Permission Slip
26 KB
Time Sheet
3 MB
House Remodel
8 items
Keynote
23 items
Numbers
8 items
GoodReader
4 items
Sketchbook Pro
3 items

Terminology



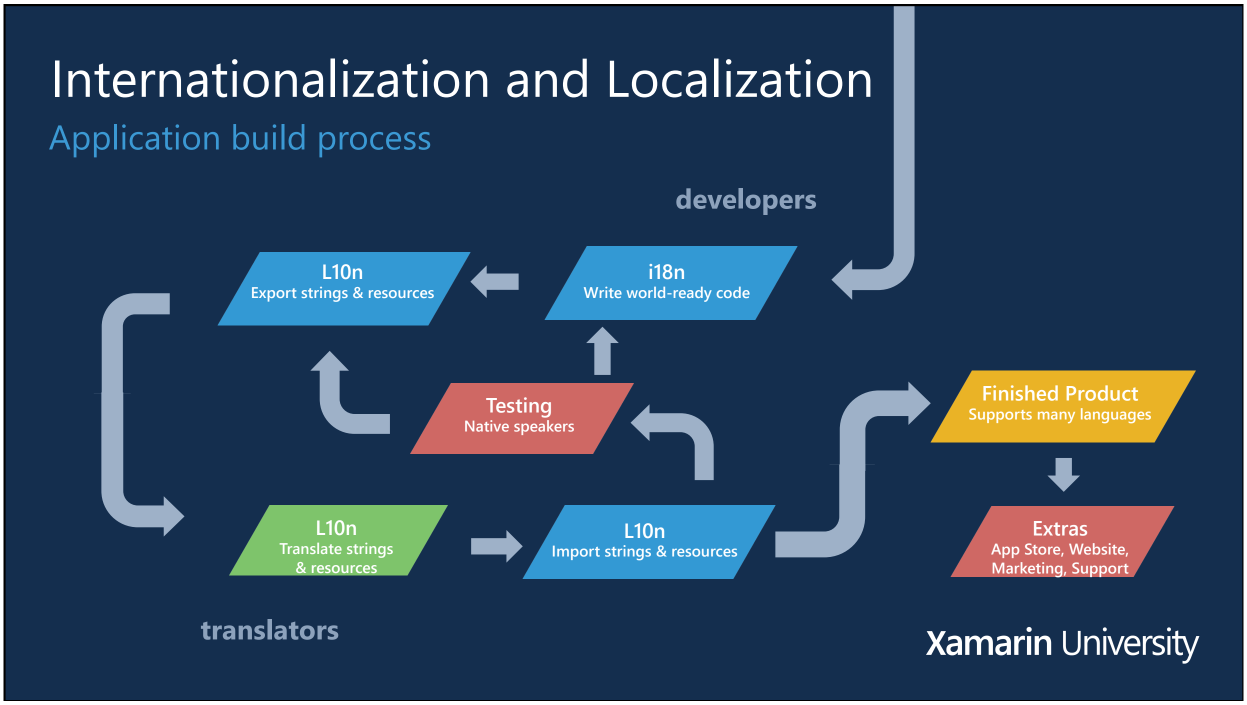
Terminology

- Internationalization – i18n
- Localization – L10n
- Globalization, sometimes used to refer to both

- Language – en, fr, de
- Region– en-AU, es-ES, fr-CA, pt-BR
- Locale/Culture

Process

Xamarin University



Internationalization and Localization

Application build process

developers

i18n
Write world-ready code

L10n
Export strings & resources

L10n
Translate strings & resources

L10n
Import strings & resources

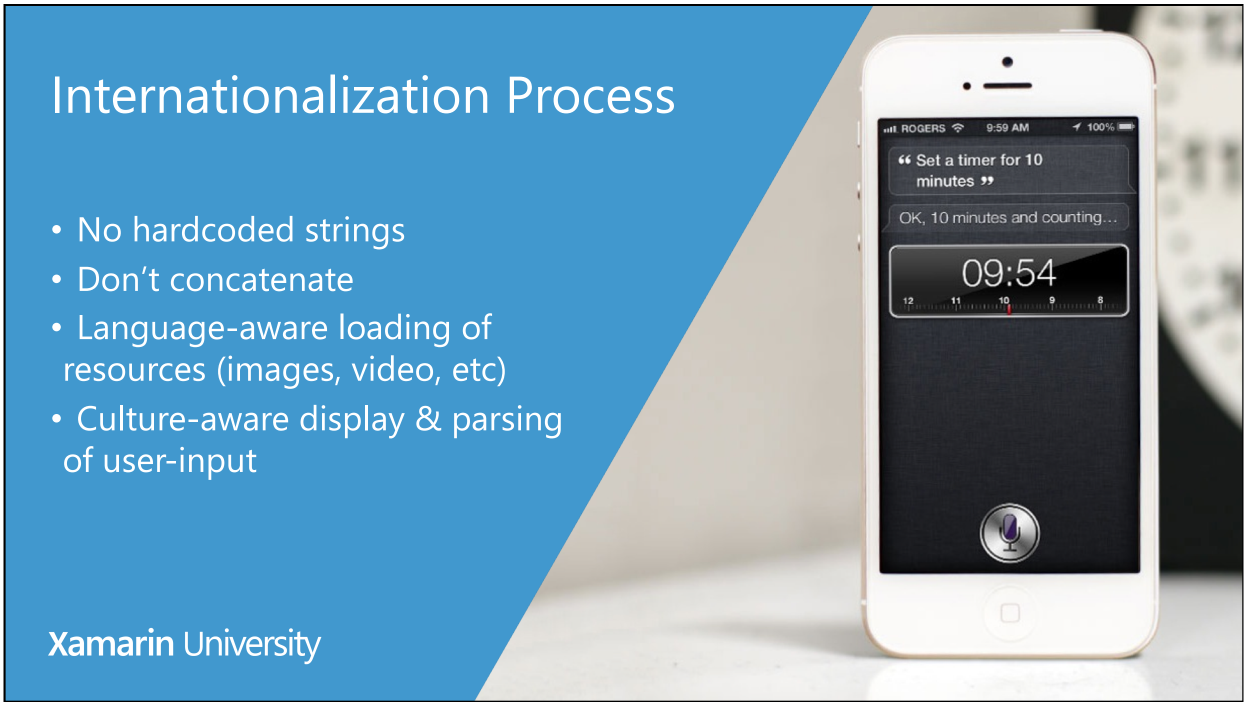
Testing
Native speakers

Finished Product
Supports many languages

Extras
App Store, Website, Marketing, Support

translators

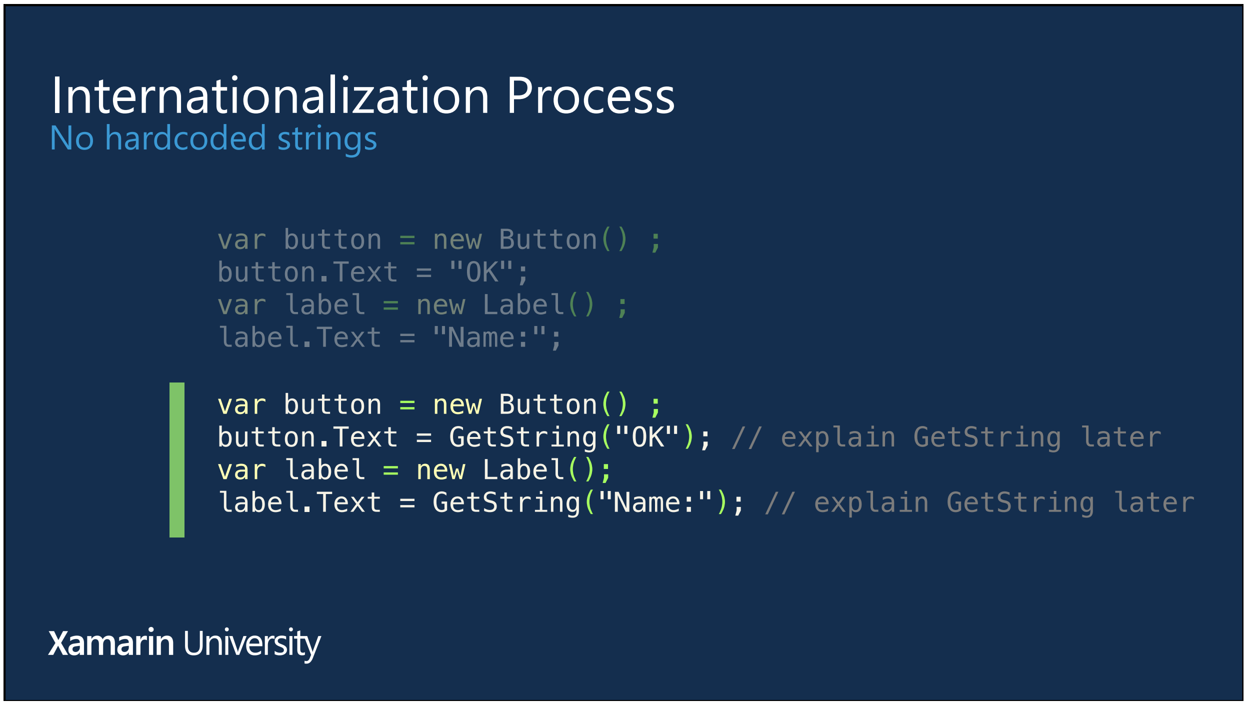
Xamarin University



Internationalization Process

- No hardcoded strings
- Don’t concatenate
- Language-aware loading of resources (images, video, etc)
- Culture-aware display & parsing of user-input

Xamarin University

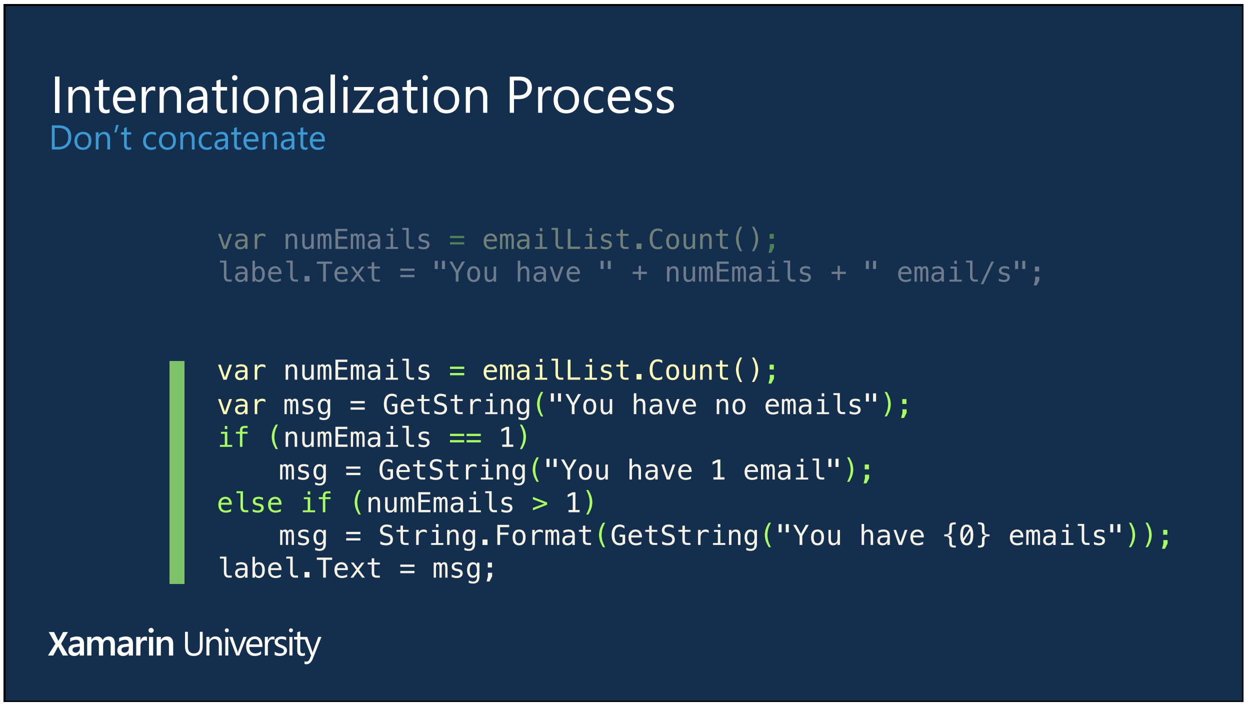


Internationalization Process

No hardcoded strings

```
var button = new Button() ;
button.Text = "OK";
var label = new Label() ;
label.Text = "Name:";

var button = new Button() ;
button.Text = GetString("OK"); // explain GetString later
var label = new Label();
label.Text = GetString("Name:"); // explain GetString later
```

Xamarin University



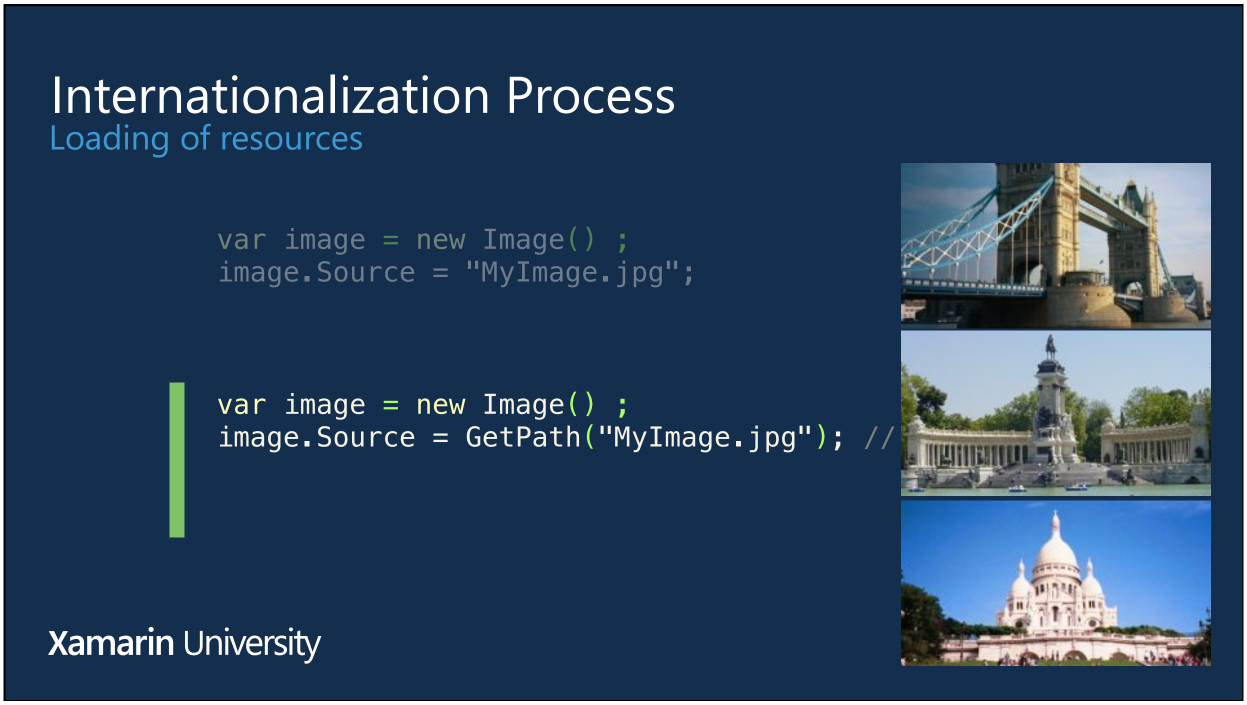
Internationalization Process

Don’t concatenate

```
var numEmails = emailList.Count();
label.Text = "You have " + numEmails + " email/s";
```

```
var numEmails = emailList.Count();
var msg = GetString("You have no emails");
if (numEmails == 1)
    msg = GetString("You have 1 email");
else if (numEmails > 1)
    msg = String.Format(GetString("You have {0} emails"));
label.Text = msg;
```

Xamarin University



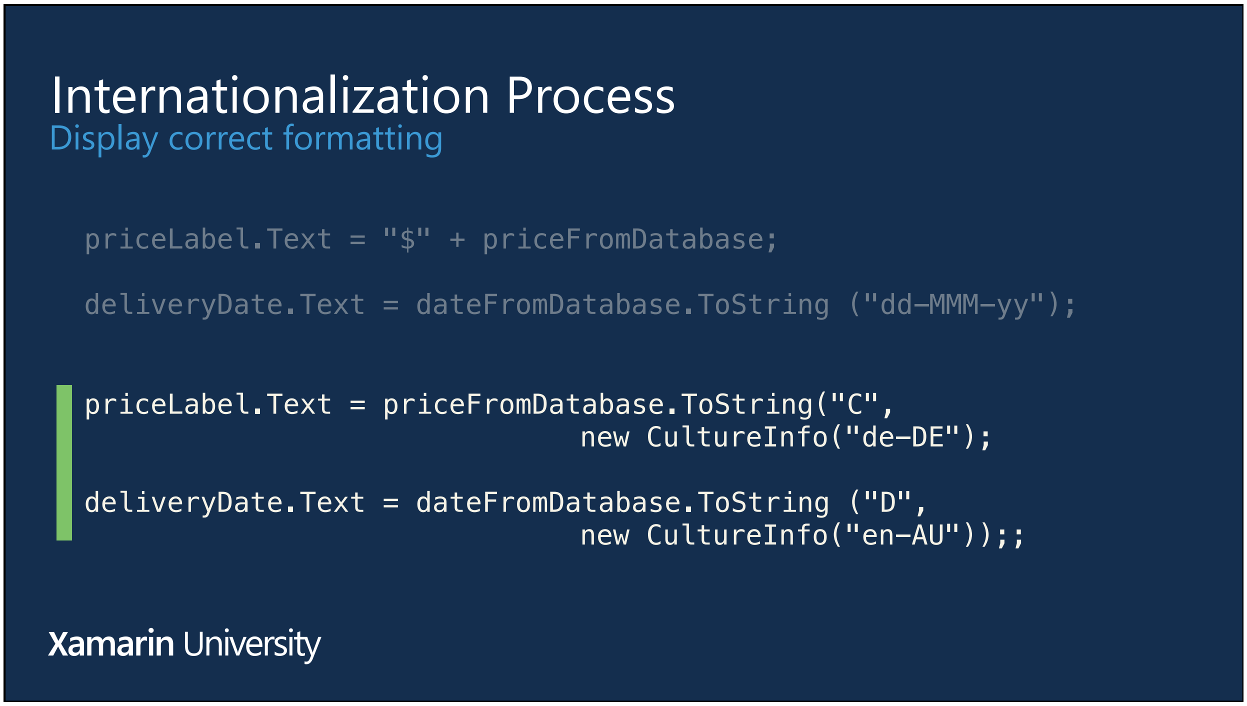
Internationalization Process

Loading of resources

```
var image = new Image() ;
image.Source = "MyImage.jpg";
```

```
var image = new Image() ;
image.Source = GetPath("MyImage.jpg"); //
```

Xamarin University



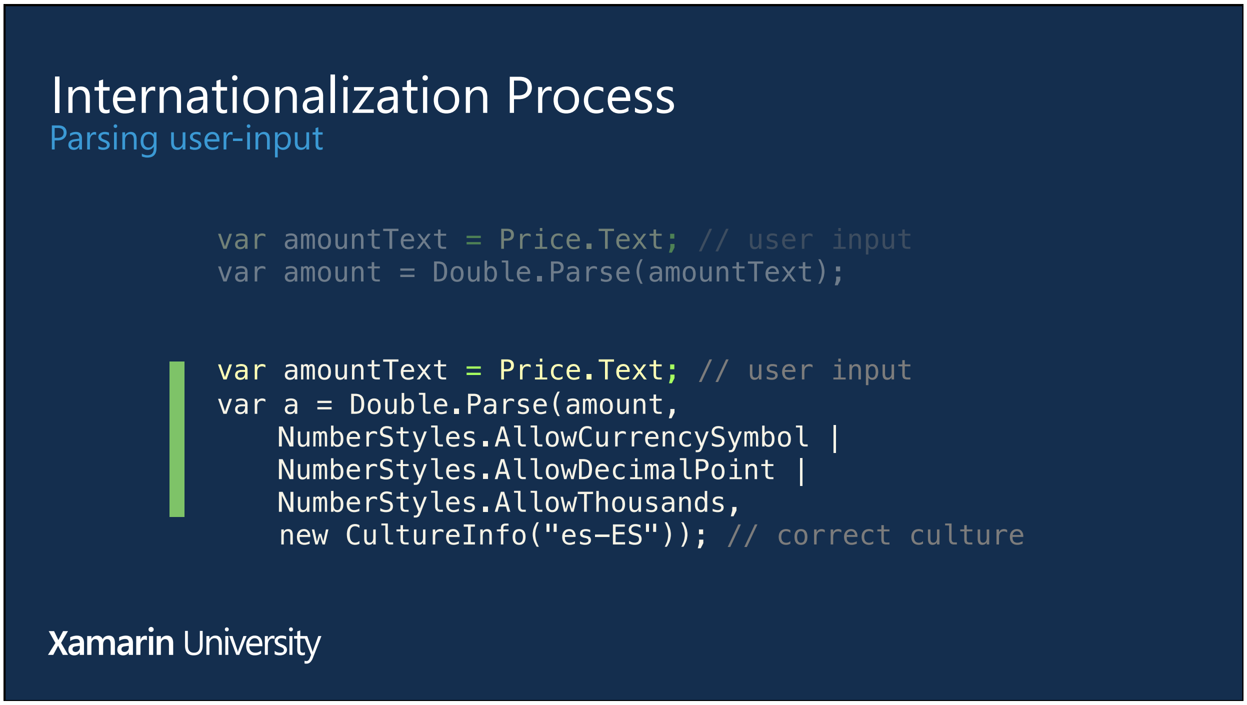
Internationalization Process

Display correct formatting

```
priceLabel.Text = "$" + priceFromDatabase;

deliveryDate.Text = dateFromDatabase.ToString ("dd-MMM-yy");
```

```
priceLabel.Text = priceFromDatabase.ToString("C",
                              new CultureInfo("de-DE");

deliveryDate.Text = dateFromDatabase.ToString ("D",
                              new CultureInfo("en-AU"));;
```

Xamarin University



Internationalization Process

Parsing user-input

```
var amountText = Price.Text; // user input
var amount = Double.Parse(amountText);
```

```
var amountText = Price.Text; // user input
var a = Double.Parse(amount,
    NumberStyles.AllowCurrencySymbol |
    NumberStyles.AllowDecimalPoint |
    NumberStyles.AllowThousands,
    new CultureInfo("es-ES")); // correct culture
```

Xamarin University

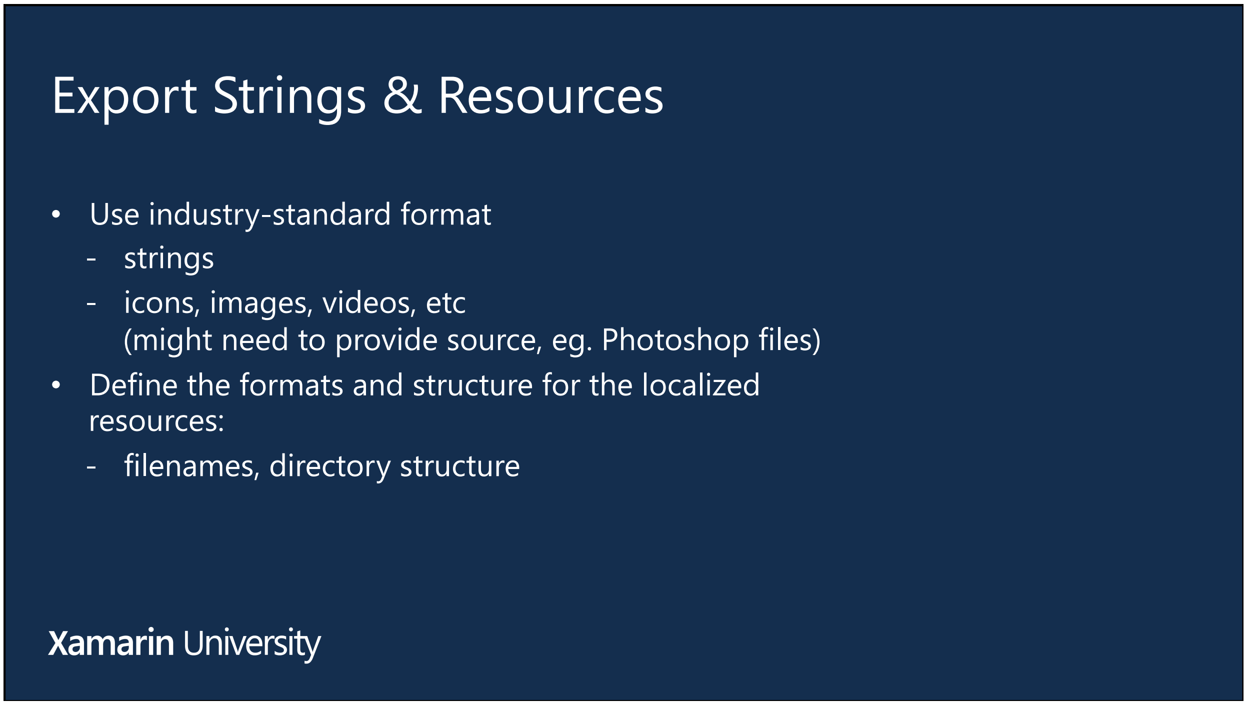
ROGERS
9:59 AM
100%
“ Set a timer for 10 minutes ”
OK, 10 minutes and counting...
09:54
12
11
10
9
8

Localization Process

- Export strings
- Export resources (images, videos, etc)
- Engage translators/designers

- Add translated strings and resources to the project

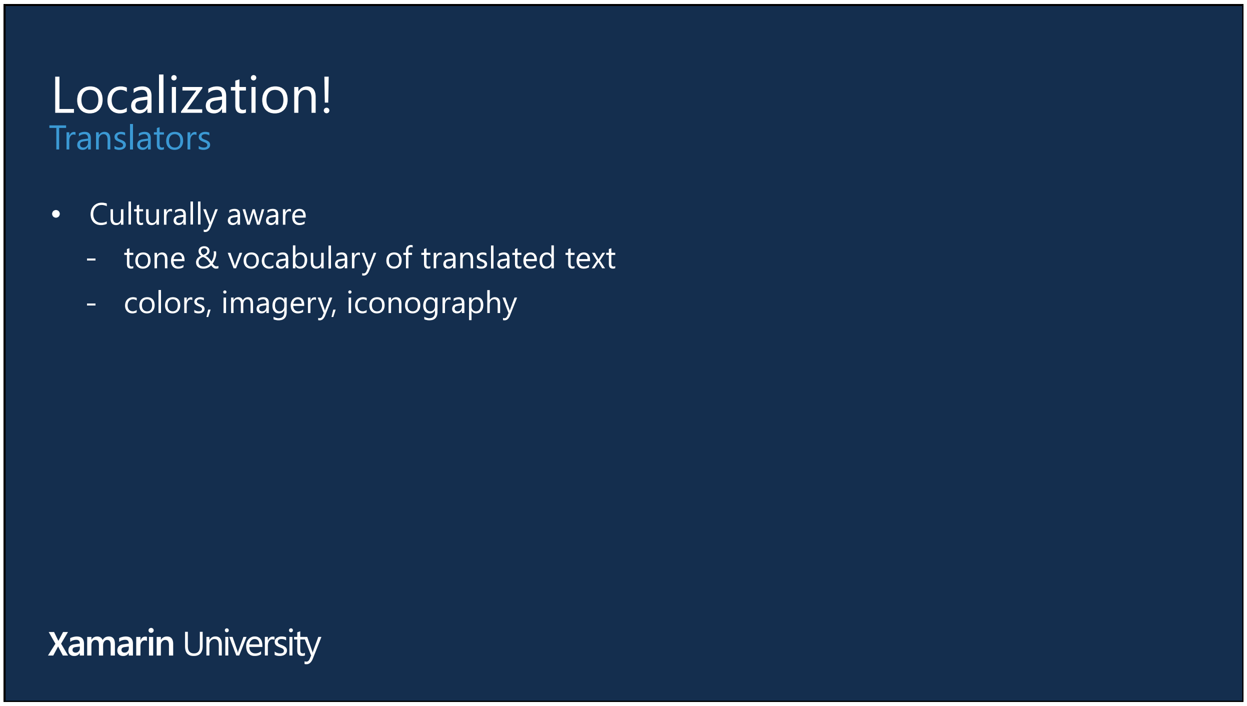
Xamarin University



Export Strings & Resources

- Use industry-standard format
  - strings
  - icons, images, videos, etc
    (might need to provide source, eg. Photoshop files)
- Define the formats and structure for the localized resources:
  - filenames, directory structure

Xamarin University

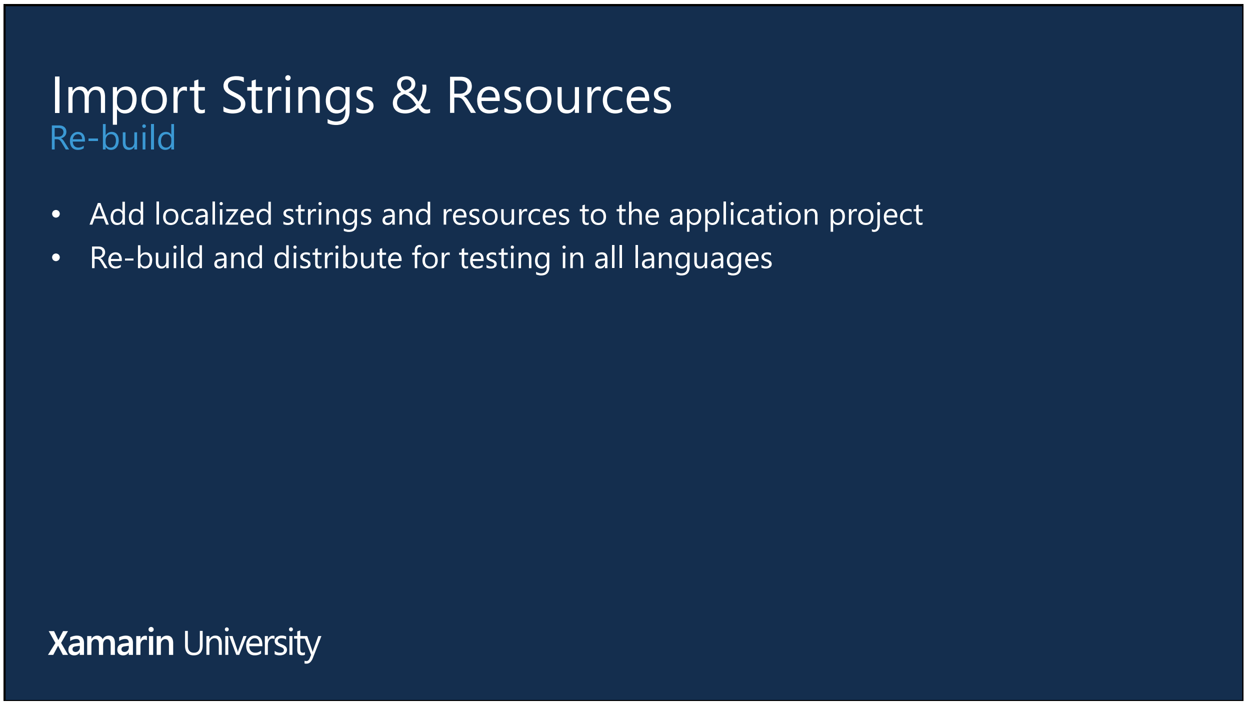


Localization!

Translators

- Culturally aware
  - tone & vocabulary of translated text
  - colors, imagery, iconography

Xamarin University



Import Strings & Resources

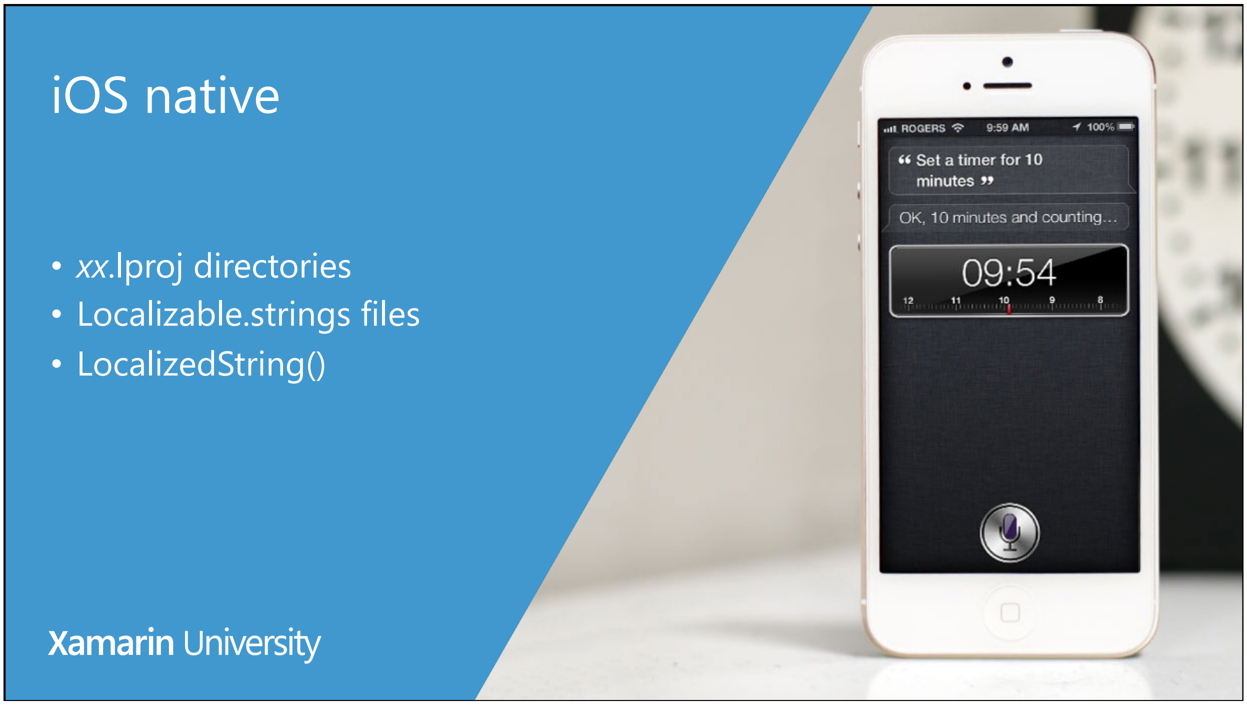
Re-build

- Add localized strings and resources to the application project
- Re-build and distribute for testing in all languages

Xamarin University

iOS Native

Xamarin University

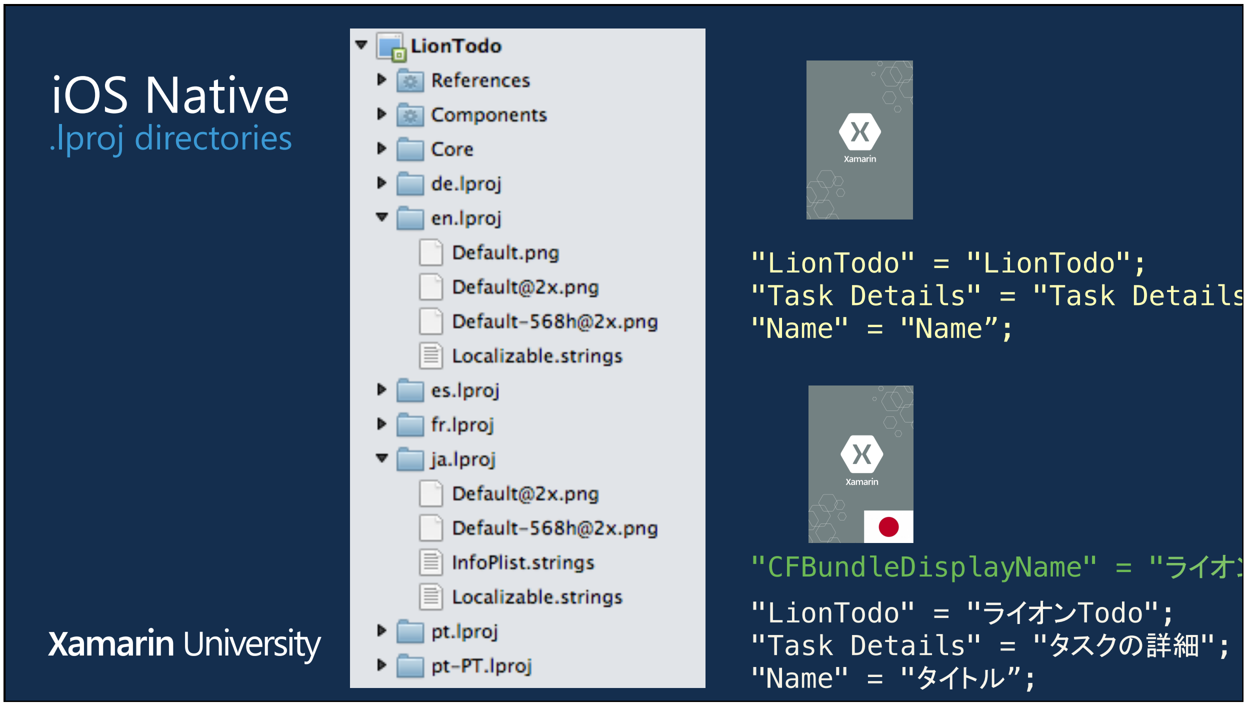


iOS native

- *xx*.lproj directories
- Localizable.strings files
- LocalizedString()

Xamarin University

ROGERS 9:59 AM 100%
“ Set a timer for 10 minutes ”
OK, 10 minutes and counting…
09:54
12 11 10 9 8



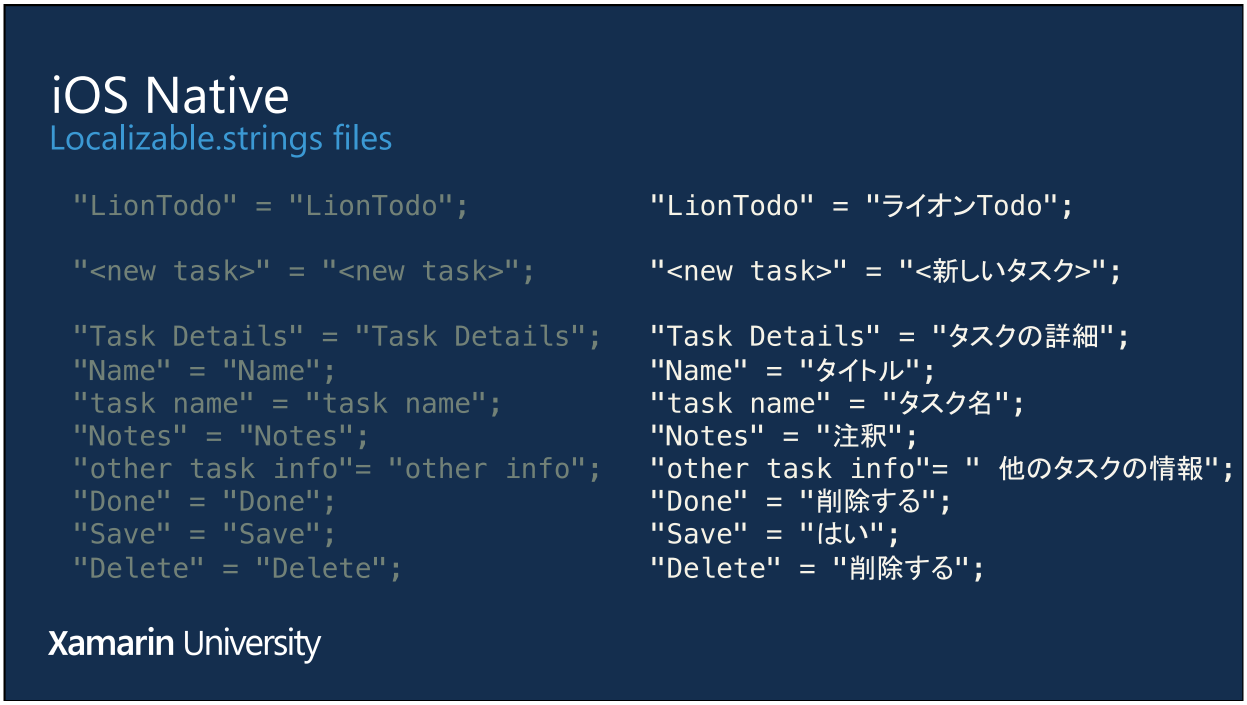
iOS Native
.lproj directories

```
▼ LionTodo
  ▶ References
  ▶ Components
  ▶ Core
  ▶ de.lproj
  ▼ en.lproj
      Default.png
      Default@2x.png
      Default-568h@2x.png
      Localizable.strings
  ▶ es.lproj
  ▶ fr.lproj
  ▼ ja.lproj
      Default@2x.png
      Default-568h@2x.png
      InfoPlist.strings
      Localizable.strings
  ▶ pt.lproj
  ▶ pt-PT.lproj
```

```
"LionTodo" = "LionTodo";
"Task Details" = "Task Details
"Name" = "Name”;
```

```
"CFBundleDisplayName" = "ライオ
"LionTodo" = "ライオンTodo";
"Task Details" = "タスクの詳細";
"Name" = "タイトル”;
```

Xamarin University



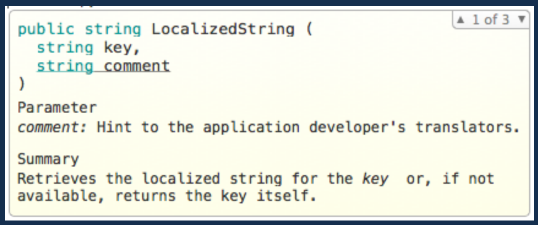
iOS Native
Localizable.strings files

```
"LionTodo" = "LionTodo";                    "LionTodo" = "ライオンTodo";

"<new task>" = "<new task>";                "<new task>" = "<新しいタスク>";

"Task Details" = "Task Details";            "Task Details" = "タスクの詳細";
"Name" = "Name";                            "Name" = "タイトル";
"task name" = "task name";                  "task name" = "タスク名";
"Notes" = "Notes";                          "Notes" = "注釈";
"other task info"= "other info";            "other task info"= " 他のタスクの情報";
"Done" = "Done";                            "Done" = "削除する";
"Save" = "Save";                            "Save" = "はい";
"Delete" = "Delete";                        "Delete" = "削除する";
```

Xamarin University

iOS Native

LocalizedString()



```
public string LocalizedString (
  string key,
  string comment
)
Parameter
comment: Hint to the application developer's translators.

Summary
Retrieves the localized string for the key  or, if not
available, returns the key itself.
```

```
var bundle = NSBundle.MainBundle;
// labels
nameLabel.Text  = bundle.LocalizedString("Name", "Title of a todo task");
notesLabel.Text = bundle.LocalizedString("Notes", "Extra info about task'
// button
saveButton.SetTitle(bundle.LocalizedString("Save", "Save task info"),
                    UIControlState.Normal);
```

Xamarin University

Android Native

Xamarin University

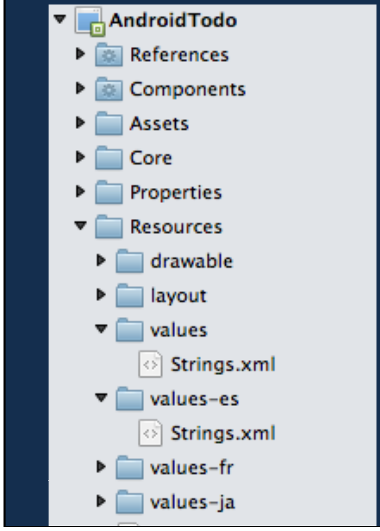
Android native

- values-*xx* directories
- Strings.xml files

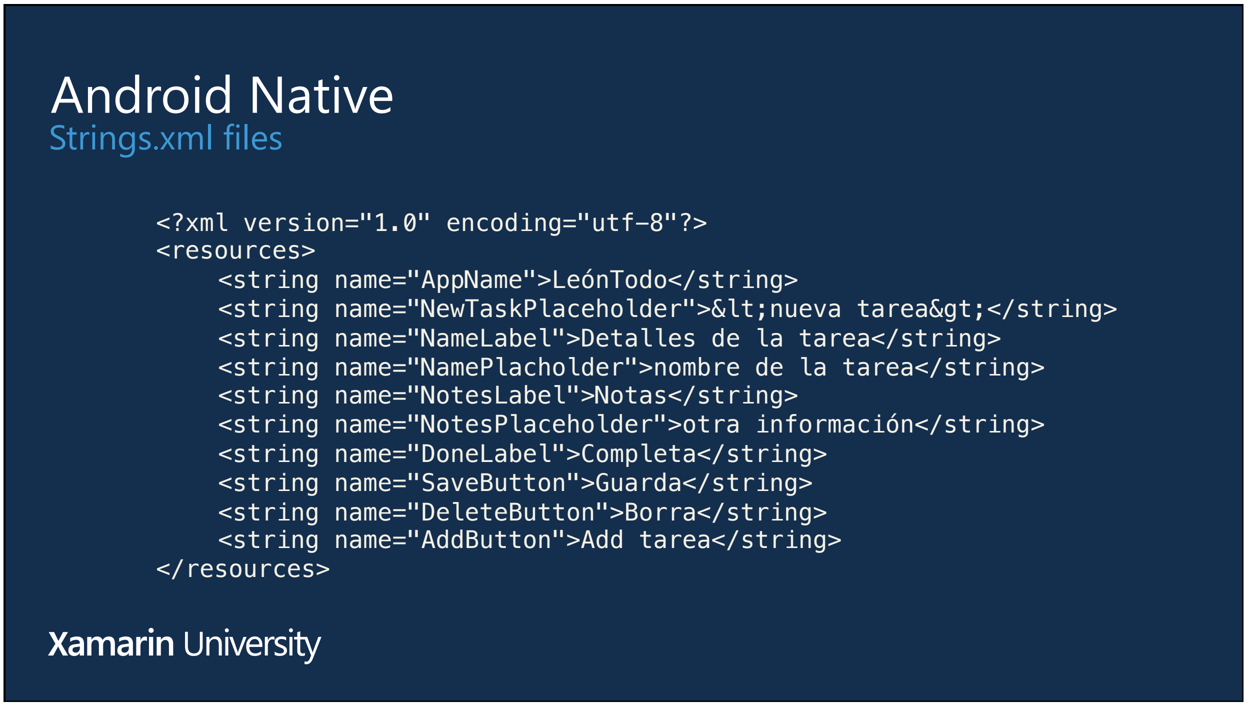
Xamarin University

Android Native

values- directories



```
<string name="AppName">LionTodo</string>
<string name="NameLabel">Name</string>
<string name="NamePlacholder">task name</string>

<string name="AppName">LeónTodo</string>
<string name="NameLabel">Detalles de la tarea</st
<string name="NamePlacholder">nombre de la tarea<
```



Android Native

Strings.xml files

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="AppName">LeónTodo</string>
    <string name="NewTaskPlaceholder">&lt;nueva tarea&gt;</string>
    <string name="NameLabel">Detalles de la tarea</string>
    <string name="NamePlacholder">nombre de la tarea</string>
    <string name="NotesLabel">Notas</string>
    <string name="NotesPlaceholder">otra información</string>
    <string name="DoneLabel">Completa</string>
    <string name="SaveButton">Guarda</string>
    <string name="DeleteButton">Borra</string>
    <string name="AddButton">Add tarea</string>
</resources>
```

Android Native

Code

```
d
```

Windows Phone & Xamarin.Forms

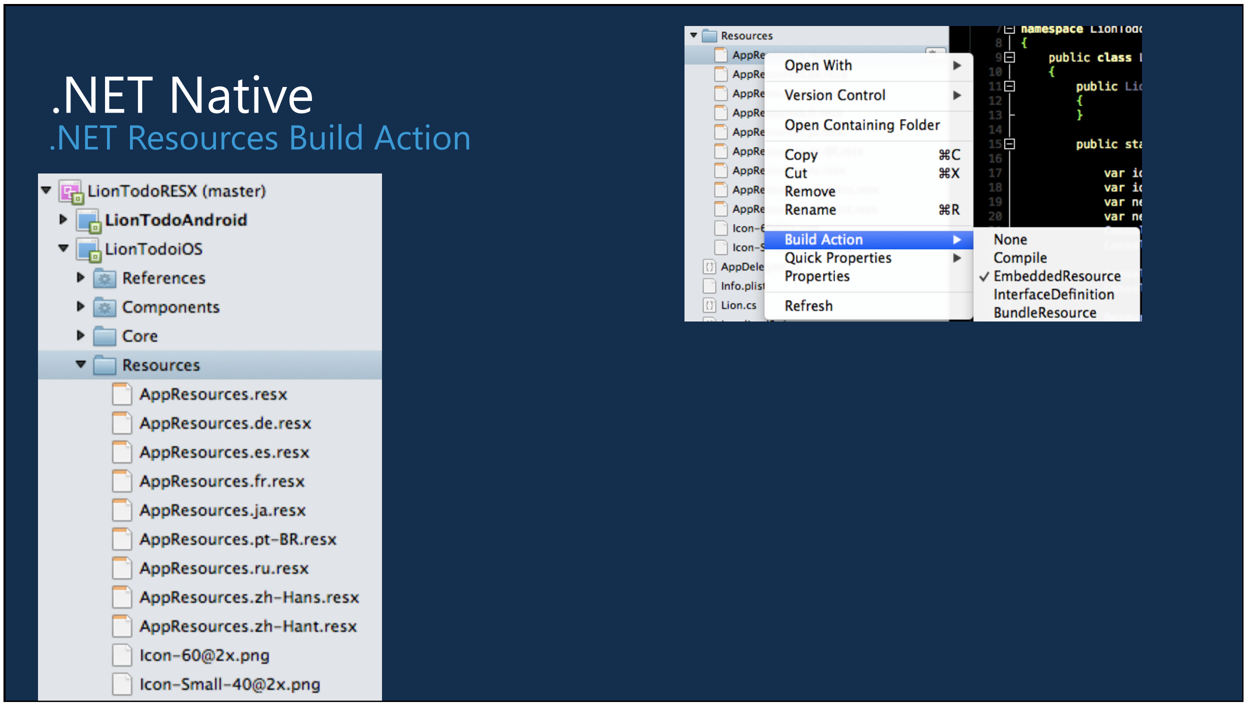
Xamarin University

Windows Phone & Xamarin.Forms

- RESX
- Strings.xml files

Xamarin University

ROGERS
9:59 AM
100%
“ Set a timer for 10 minutes ”
OK, 10 minutes and counting…
09:54
12
11
10
9
8

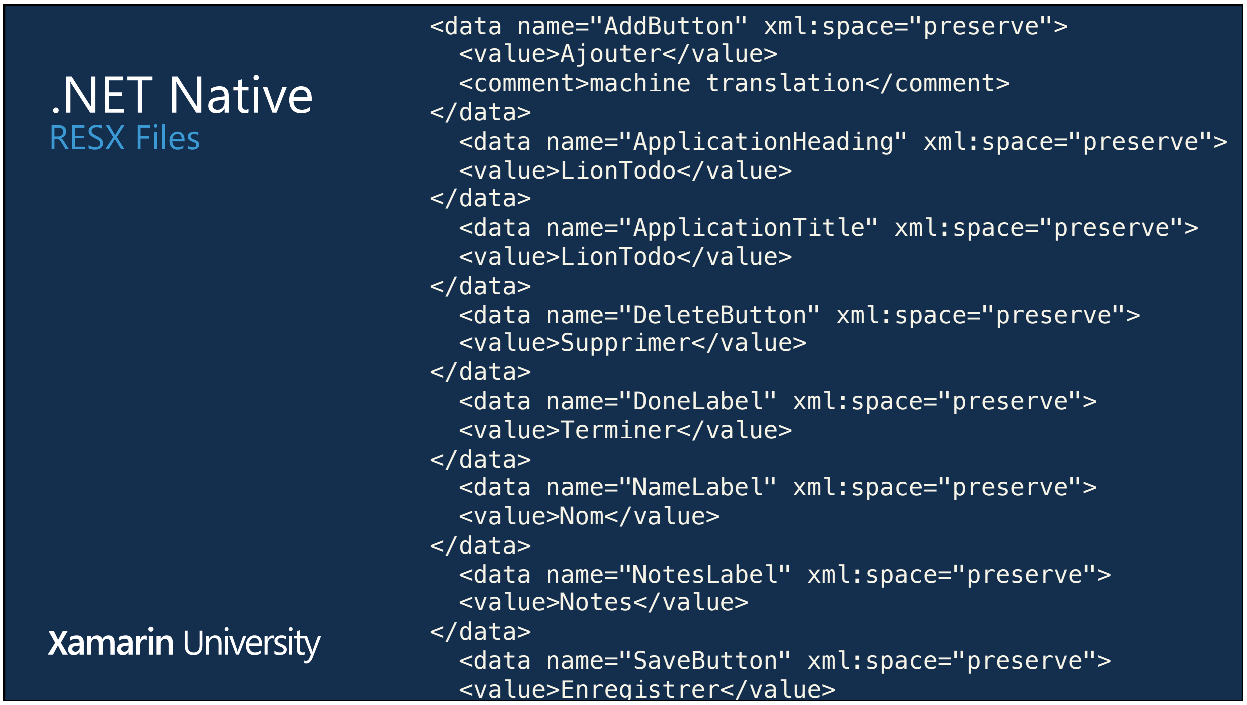


.NET Native

.NET Resources Build Action

- LionTodoRESX (master)
  - LionTodoAndroid
  - LionTodoiOS
    - References
    - Components
    - Core
    - Resources
      - AppResources.resx
      - AppResources.de.resx
      - AppResources.es.resx
      - AppResources.fr.resx
      - AppResources.ja.resx
      - AppResources.pt-BR.resx
      - AppResources.ru.resx
      - AppResources.zh-Hans.resx
      - AppResources.zh-Hant.resx
      - Icon-60@2x.png
      - Icon-Small-40@2x.png

Resources
Open With
Version Control
Open Containing Folder
Copy ⌘C
Cut ⌘X
Remove
Rename ⌘R
Build Action
Quick Properties
Properties
Refresh
None
Compile
✓ EmbeddedResource
InterfaceDefinition
BundleResource
Info.plist
Lion.cs
namespace LionTodo
public class
public sta

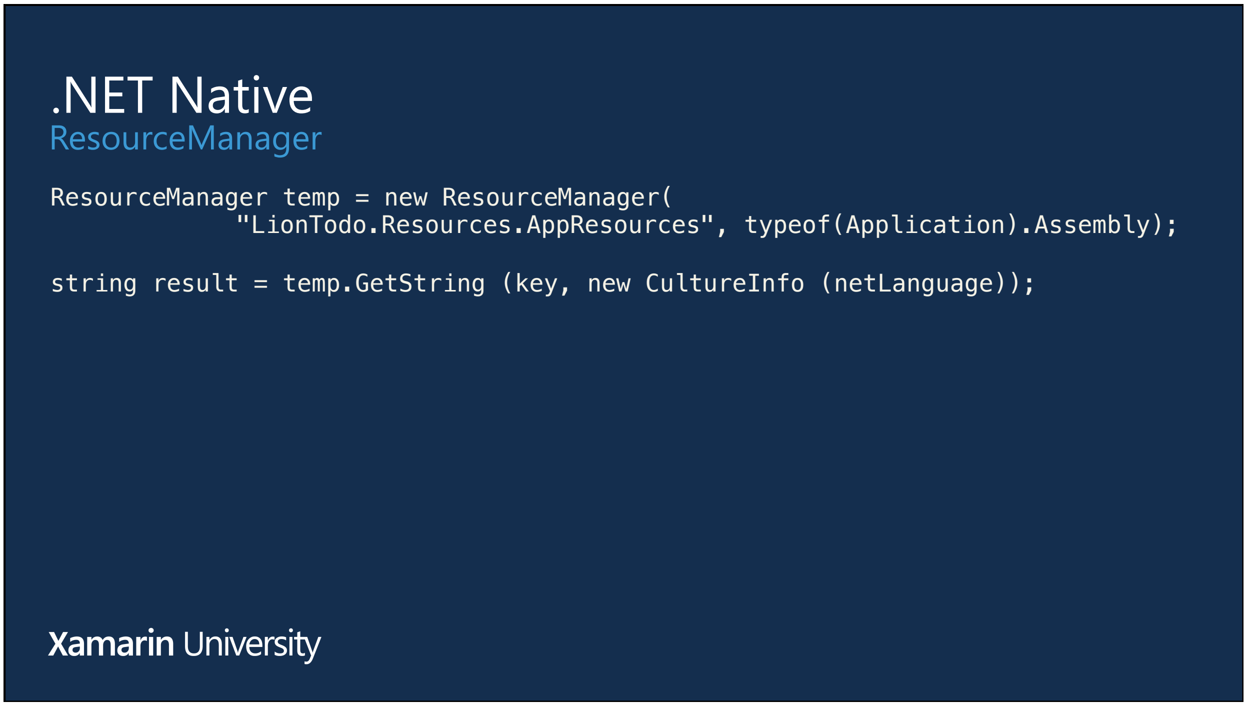


.NET Native

RESX Files

```
<data name="AddButton" xml:space="preserve">
  <value>Ajouter</value>
  <comment>machine translation</comment>
</data>
  <data name="ApplicationHeading" xml:space="preserve">
  <value>LionTodo</value>
</data>
  <data name="ApplicationTitle" xml:space="preserve">
  <value>LionTodo</value>
</data>
  <data name="DeleteButton" xml:space="preserve">
  <value>Supprimer</value>
</data>
  <data name="DoneLabel" xml:space="preserve">
  <value>Terminer</value>
</data>
  <data name="NameLabel" xml:space="preserve">
  <value>Nom</value>
</data>
  <data name="NotesLabel" xml:space="preserve">
  <value>Notes</value>
</data>
  <data name="SaveButton" xml:space="preserve">
  <value>Enregistrer</value>
```

Xamarin University

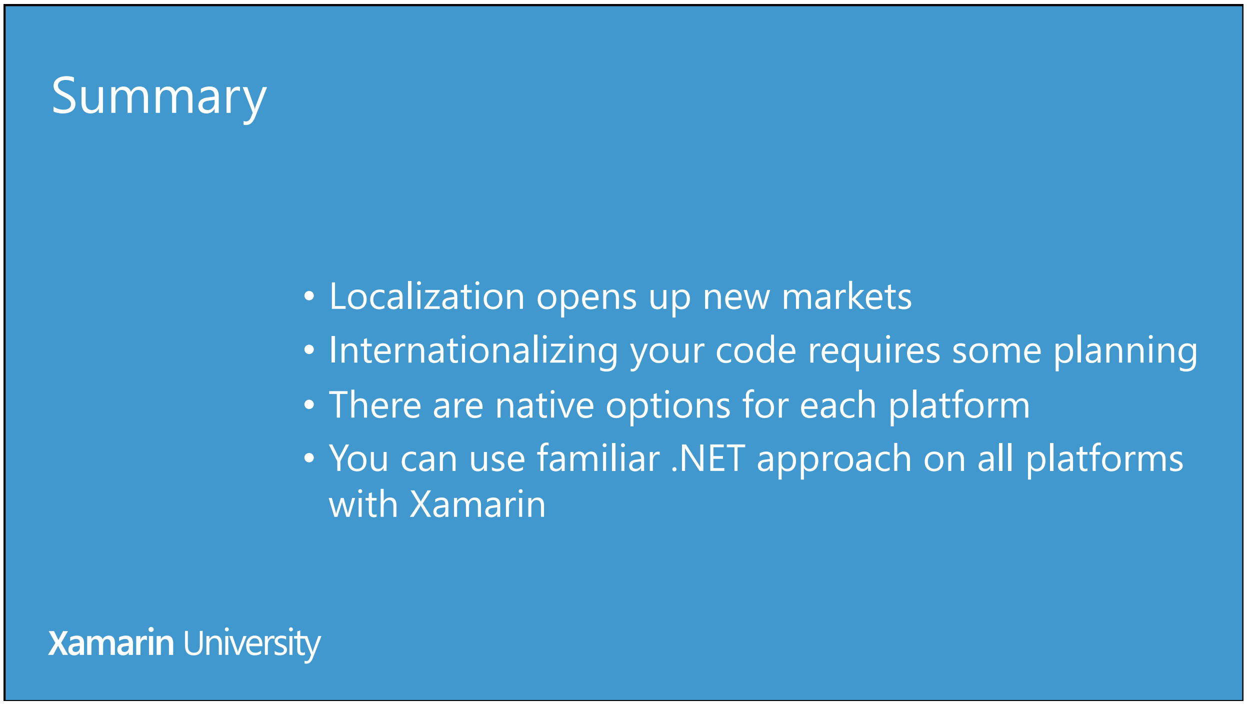


.NET Native

ResourceManager

```
ResourceManager temp = new ResourceManager(
             "LionTodo.Resources.AppResources", typeof(Application).Assembly);

string result = temp.GetString (key, new CultureInfo (netLanguage));
```

Xamarin University



Summary

- Localization opens up new markets
- Internationalizing your code requires some planning
- There are native options for each platform
- You can use familiar .NET approach on all platforms with Xamarin

Xamarin University