**Localization / Internationalization in iOS**

**Localization** is merely translating the app’s user interface and resources into different languages

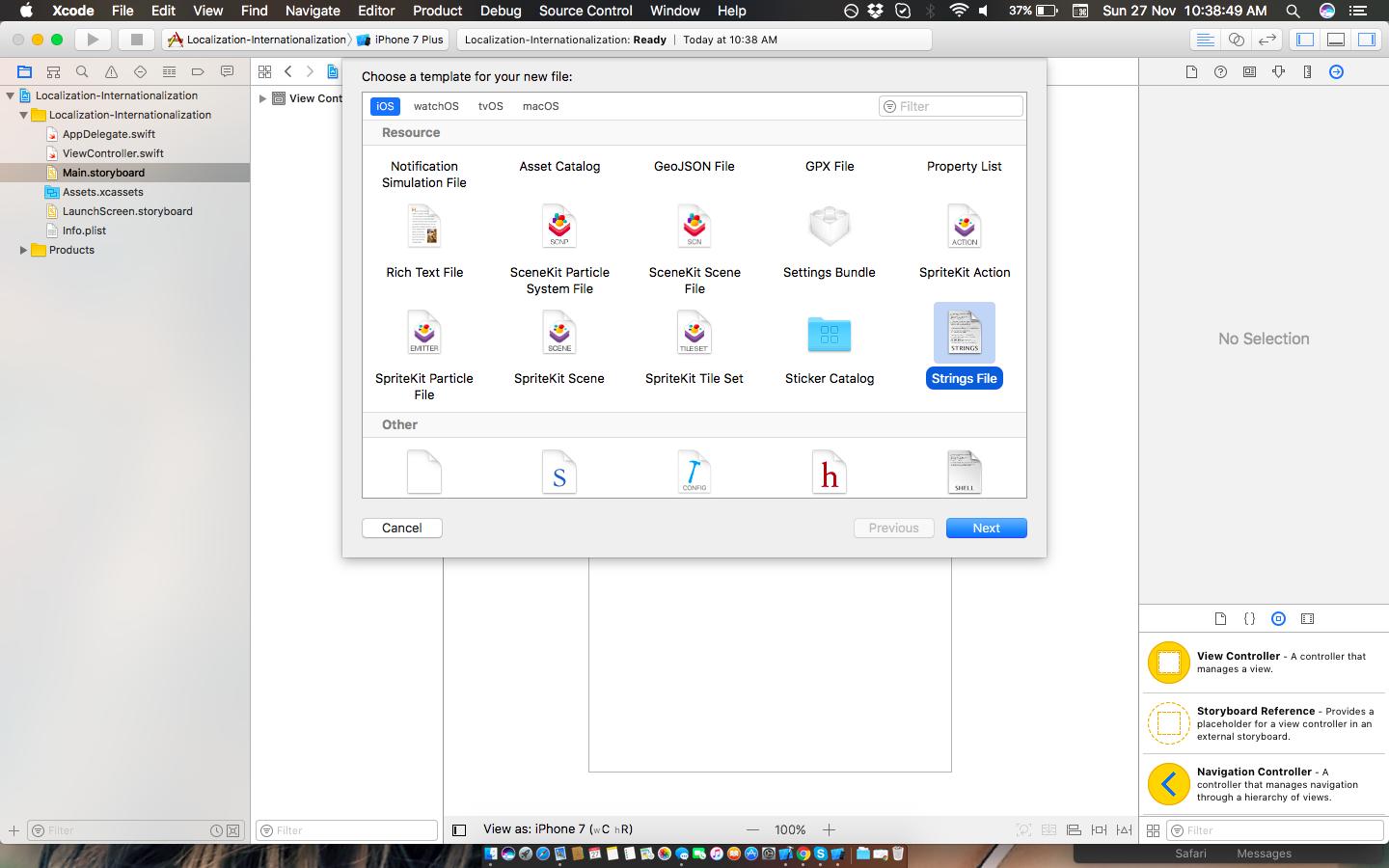
**Internationalization** is the process of designing your app for international compatibility. For example:

* Handle text input, output processing in the user’s native language.
* Handle different date, time and number formats.
* Utilize the appropriate calendar and time zone for processing data.

We can change almost all the UI containing text also images.

**Initiating the process : separating texts from the code**

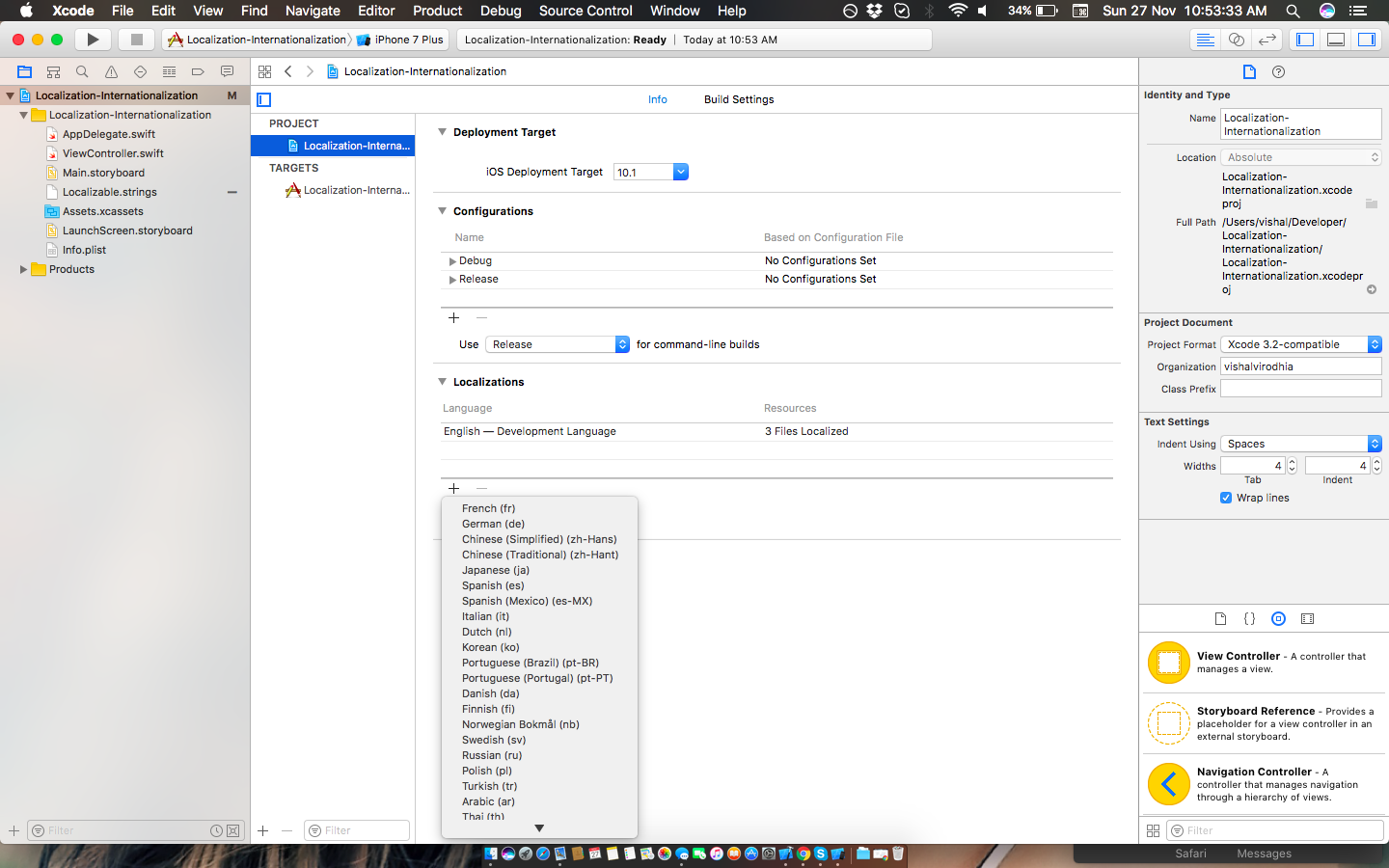
1. Create the String file Go to **File > New > File**. Choose **Strings File** unders the Resource subsection.

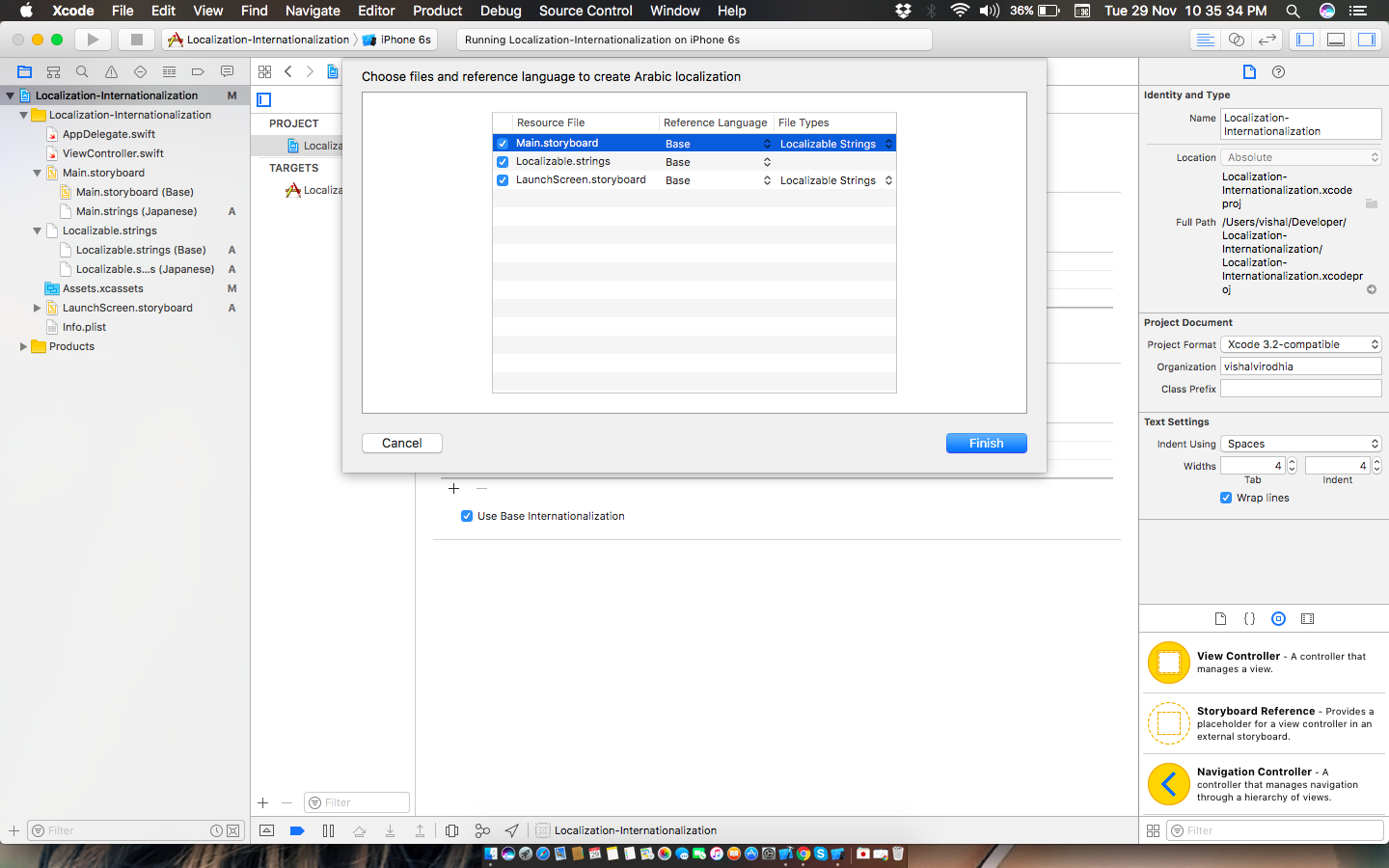


2. Name the file **Localizable.strings**, its standard. In this file you have to define the text with respect to key like key content pair, you have to hardcode it all for all languages :

|  |
| --- |
| "KEY" = "CONTENT";  "labelText" = "Localization text";  "buttonTitle" = "Working ?";  "imageName" = "India"; |
| *:- Define all the text used in your app like above.* |
|  |
|  |
|  |

3. Add different language support by selecting clicking on your project Name the click on **Project** in the next pane not Target. Click in the + icon below Localization section and select your required language.





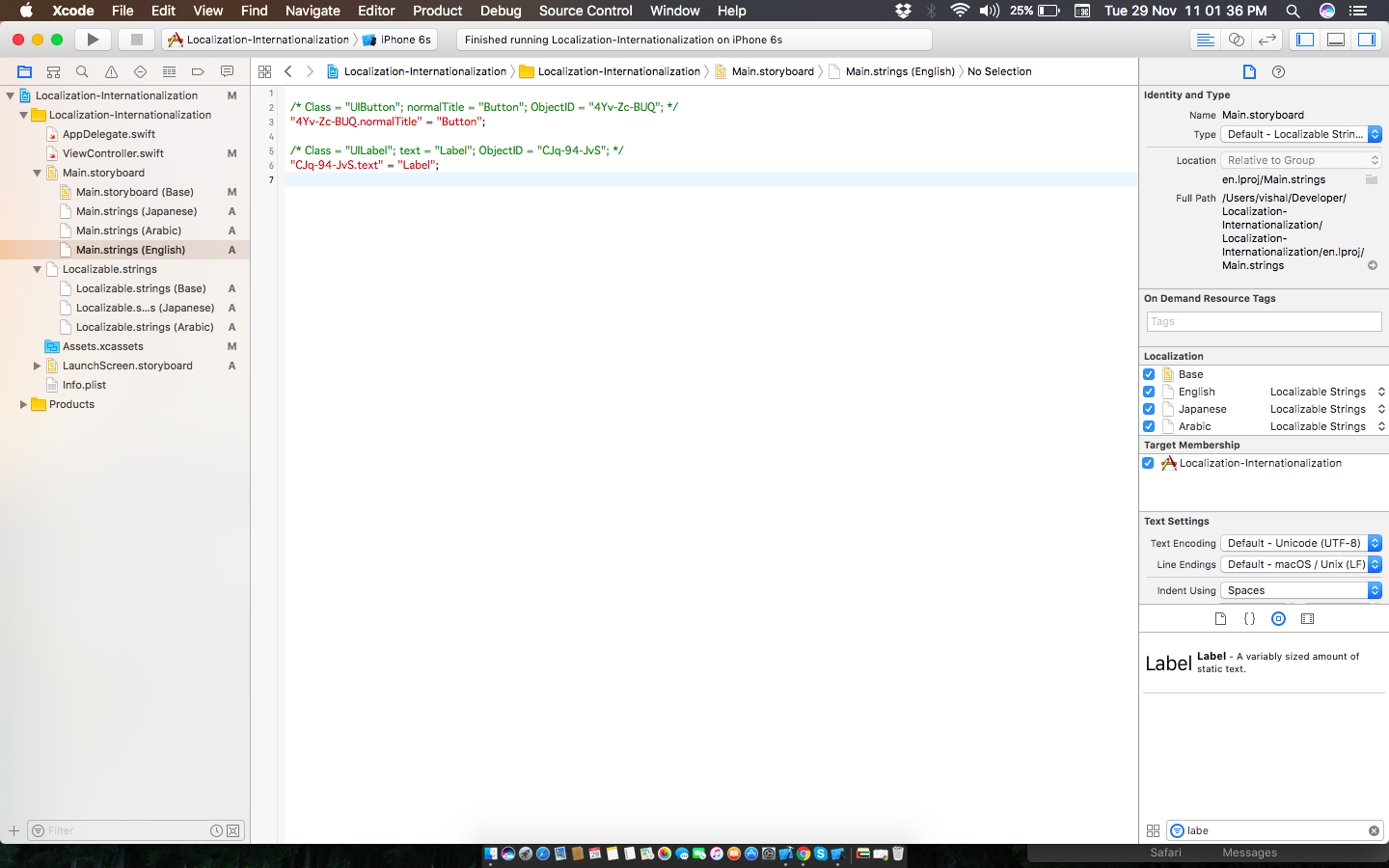
**Showing the text on UI elements**

4. Showing the text on UI elements is just a line code.

For example :

* For label : label.text = NSLocalizedString("labelText", comment: "label")
* For Button : anyButton.setTitle(NSLocalizedString("buttonTitle", comment: "title"), for: .normal)
* For image : flagImage.image = UIImage(named: NSLocalizedString("imageName", comment: "image"))

5. For your static elements of storyboards you can assign them text by using their ID in Main.String(Language) individually. Like below :

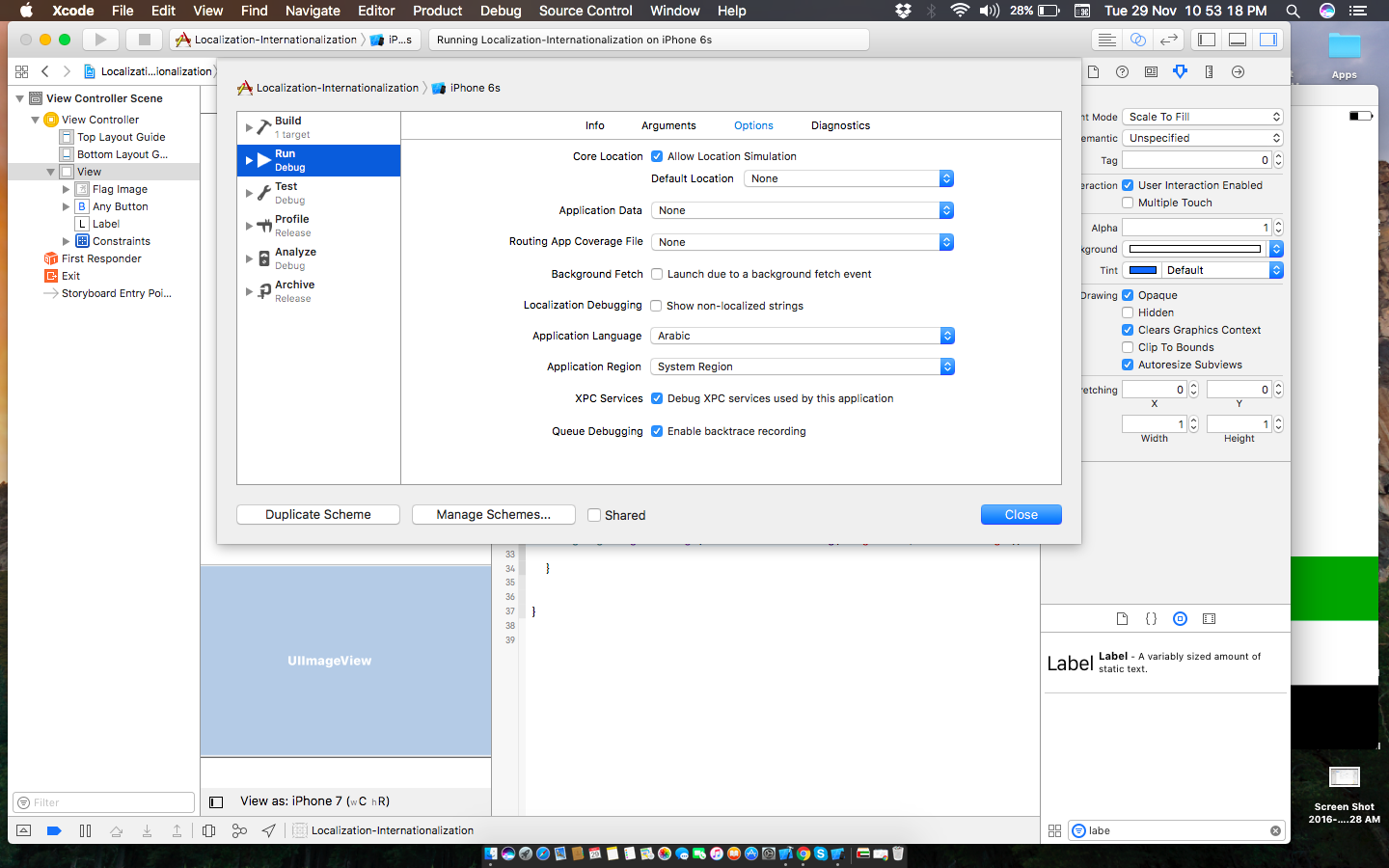


6. Change the language of your phone/simulator by going to

**Settings app > General > Language & Region > iPhone Language.**

Or

Select **Project > Edit Schema > Run > Option > Application Language.**



**Manually Changing language within app :**

Sometimes we need or required to change language within app so we have to override the method.

7. Manually selecting language : Save your selected language to *UserDefaults*

Create an override method like below :

func GetTextForKey(key:String)->String{

// For arabic

let path = Bundle.main.path(forResource: "ar", ofType: "lproj")

let bundle = Bundle(path: path!)

return (bundle?.localizedString(forKey: key, value: nil, table: nil))!

}

You can download sample project from [here](https://github.com/vishalvirodhia/Localization-Internationalization-iOS).