SwiftUI / Text input and output / Preparing views for localization

Article

Preparing views for localization

Specify hints and add strings to localize your SwiftUI views.

Overview

Localize SwiftUI views so users experience your app in their own native language, region, and culture. Xcode parses SwiftUI views for strings to localize when exporting a localization catalog. You can add hints so that Xcode generates correct, hinted strings to localize for your app.

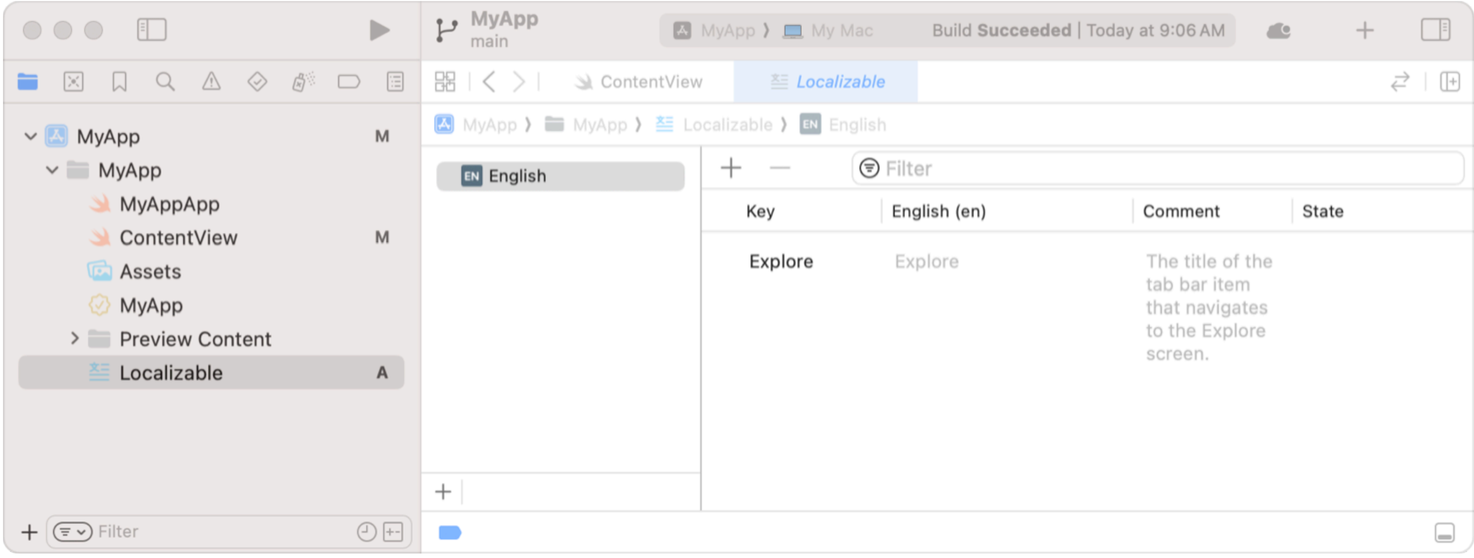
For information about string catalogs, see Localizing and varying text with a string catalog.

Add comments to text views

To ease the translation process, provide hints to translators that share how and where your app displays the strings of a `Text` view. To add a hint, use the optional `comment` parameter to the text view initializer `init(_:tableName:bundle:comment:)`. When you localize your app with Xcode, it includes the comment string along with the string. For example, the following text view includes a comment:

```
Text("Explore",
     comment: "The title of the tab bar item that navigates to the Explore screen.")
```

Xcode creates the following entry in your string catalog file for this view:



Key	English (en)	Comment	State
Explore	Explore	The title of the tab bar item that navigates to the Explore screen.	

Provide additional information with text views

You can localize many SwiftUI views that have a string label by providing a string that SwiftUI interprets as a `LocalizedStringKey`. The system uses the key to retrieve a localized value from your string catalog at runtime, or uses the string directly if it can't find the key in the catalog. For example, SwiftUI uses the string input to the following `Label` initializer as a localized string key:

```
Label("Message", image: "msgSymbol")
```

If you additionally want to provide a comment for localization, you can use an explicit `Text` view instead:

```
Label {
    Text("Message",
         comment: "A label that displays 'Message' and a corresponding image.")
} icon: {
    Image("msgSymbol")
}
```

Many SwiftUI controls have view builder initializers that enable you to follow this pattern. For more information on how to make your app's text translatable, see Preparing your app's text for translation.

See Also

Localizing text

`struct LocalizedStringKey`

The key used to look up an entry in a strings file or strings dictionary file.

`var locale: Locale`

The current locale that views should use.

`func typesettingLanguage(_:isEnabled:)`

Specifies the language for typesetting.

`struct TypesettingLanguage`

Defines how typesetting language is determined for text.