**HOW DO I …**

Localize an Application?

We’re going to answer this question by accomplishing the following steps:

* Localizing the UI elements of an Application
* Localizing the Application Title and Tile

**Localizing the UI elements of an Application**

* In Visual Studio, create a new Project based on the “Windows Phone Application” project template.
* Add a New Item using the Resources template and name the file “AppResources.resx”
* In AppResources.resx, use the editor to add two strings
* Name: “Greeting”, Value: “Hello”
* Name: “Farewell”, Value: “Goodbye”
* Using the editor, set the Access Modifier on the file to Public
* Add a New Item using the Resources template and name the file “AppResources.es-ES.resx”.
* In AppResources.es-ES.resx, use the editor to add two strings
* Name: “Greeting”, Value: “Hola”
* Name: “Farewell ”, Value: “Adios”
* Using the editor, set the Access Modifier on the file to Public
* Open the Solution Properties, on the Applications tap select Assembly Information
* Set Neutral Language to English
* Right click the project and select the “Unload Project” option
* Edit the Project file and add “es-ES” to the “SupportedCultures” XML element
* Reload the project
* Add a New Item using the Class template and name the files “LocalizedStrings”
* Add a AppResources property named *LocalizedResources*

private AppResources localizedresources = new AppResources();

public AppResources LocalizedResources { get { return localizedresources; } }

* In App.xaml add an xml namespace named “local” referencing your project assembly
* Add a resource instantiating the new class named “LocalizedStrings”

xmlns:local="clr-namespace:LocalizationFeatures"

<Application.Resources>

<local:LocalizedStrings x:Key="LocalizedStrings" />

</Application.Resources>

* Change ContentPanel from a Grid to a StackPanel
* Add TextBlocks bound to each localized property

<StackPanel x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">

<TextBlock Text="{Binding Path=LocalizedResources.Greeting, Source={StaticResource LocalizedStrings}}" />

<TextBlock Text="{Binding Path=LocalizedResources.Farewell, Source={StaticResource LocalizedStrings}}" />

</StackPanel>

* Run Project
* Notice application uses default values from AppResources.resx
* Close application and navigate to Settings
* Change Regional Settings to Spanish
* Tap to Restart
* Run Project
* Notice application now using default values from AppResources.es-ES.resx

**Localizing the Application Title and Tile**

* Add a New Project using the “Win32 Project” template named “AppResLib”
* In the wizard set ApplicationSettings type equal to “DLL” and options to “empty project”
* In Project Properties > Linker > Advanced set No Entry Point equal to “Yes”
* Add a new Resource > String Table > New
* Using the editor, add two strings:
* ID: “AppTitle”, Value: “100”, Caption: “TitleNeutral”
* ID: “AppTileString”, Value: “200”, Caption: “TileTitleNeutral”
* Build Project
* Add AppResLib.dll to the Windows Phone project
* Using the editor, update the two strings:
* ID: “AppTitle”, Value: “100”, Caption: “TitleSpanish”
* ID: “AppTileString”, Value: “200”, Caption: “TileTitleSpanish”
* Rebuild Project
* Rename AppResLib.dll to “AppResLib.dll.0c0a.mui”and add to Windows Phone project
* Select “AppResLib.dll.0c0a.mui” and set Build Action to Content
* Open WMAppManifest.xml found in the Properties folder
* Change the Title attribute of the App element to “@AppResLib.dll, -100”
* Change the Title, under Token, element value to “@AppResLib.dll, -200”
* Run Project and notice title update based on Regional Settings

In conclusion, localization is an important step when developing Windows Phone applications, especially with the quickly expanding number of supported regions and languages.