Module

Internationalization

Localization

Table of contents

Internationalization

- The process of developing an application that supports
  - Localized user interfaces
  - Regional data
- For users in multiple cultures
- Separation of the application executable code from the resources to translate the user interface
- Two conceptual blocks
  - A block that contains all user interface elements
  - A block that contains executable code

Localization

- The process of adapting an application for a specific local market
  - The translation of application into localized versions
  - For each culture that the application will support
- Consists primarily of translating the user interface
- For each localized version of the application,
  - Add a new resource file that contains the localized user interface block translated into the appropriate language for the target culture
- The combination of
  - A localized version of the user interface block
  - With the executable code block

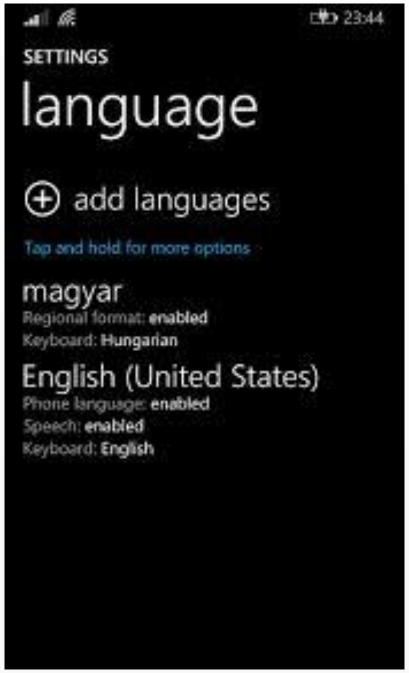
  ↳ produces a localized version of the application

Localization

- Includes
  - Translating the user interface
  - Resizing dialog boxes
  - Customizing features
  - Testing results to ensure that the application works for the target market
- The user interface block contains elements such as
  - Strings
  - Error messages
  - Dialog boxes
  - Menus
  - Embedded object resources

Languages

- The user can specify a language preference list in Phone Settings


23:44

SETTINGS

language

⊕ add languages

Tap and hold for more options

magyar
Regional format: enabled
Keyboard: Hungarian

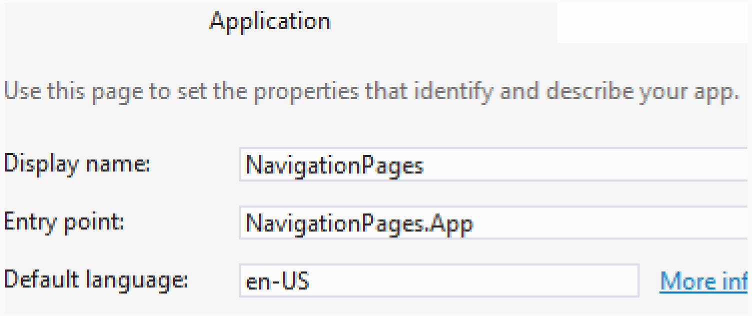
English (United States)
Phone language: enabled
Speech: enabled
Keyboard: English

Languages

- The developer can decide which languages will support its app
- A language is represented as a BCP-47 language tag
  - Can support a regional language
    - E.g. “en” for English
  - Can support regional variants
    - E.g. “en-US”, “en-GB”, …

Languages

- At runtime, Windows handles the matching of the users' language preferences and the language resources packaged in the app
  - If the user preference is "en-US", in priority order : "en-US", "en", "en-GB",…
  - If no resources can be matched, the default language of the application is used



Application

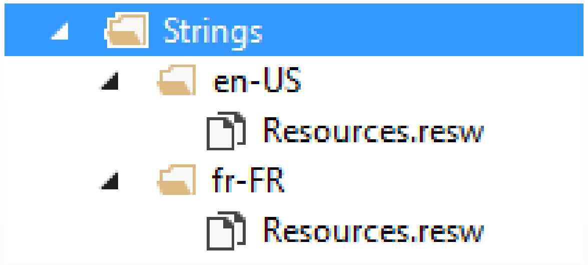
Use this page to set the properties that identify and describe your app.

Display name:	NavigationPages
Entry point:	NavigationPages.App
Default language:	en-US

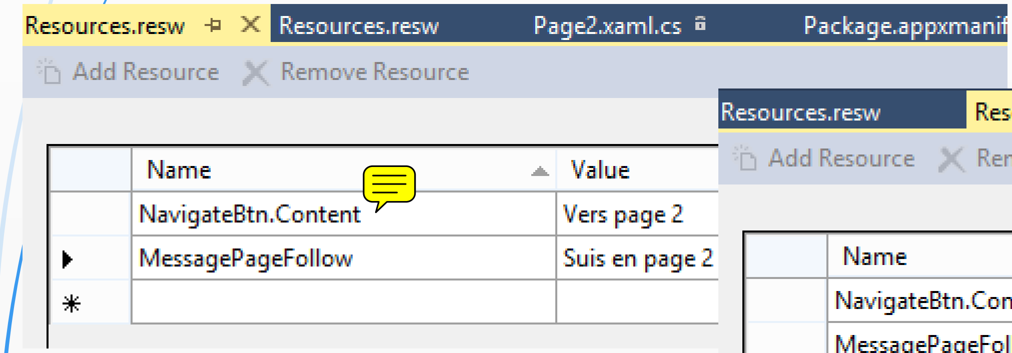
(in the Package.appxmanifest)

Strings Resources

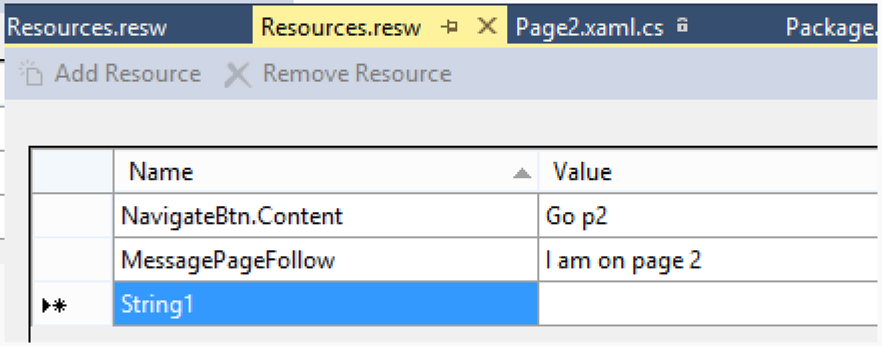
- Strings folder
  - One sub-folder per language
    - Resources.resw file



```
◢ Strings
   ◢ en-US
        Resources.resw
   ◢ fr-FR
        Resources.resw
```



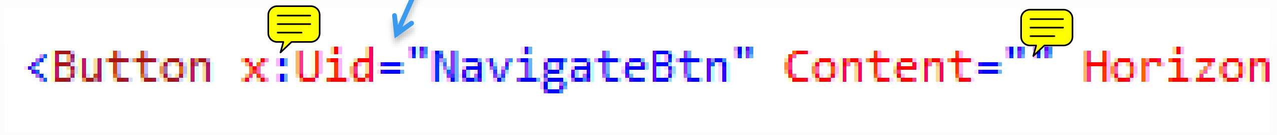
Name	Value
NavigateBtn.Content	Vers page 2
MessagePageFollow	Suis en page 2



Name	Value
NavigateBtn.Content	Go p2
MessagePageFollow	I am on page 2
String1	

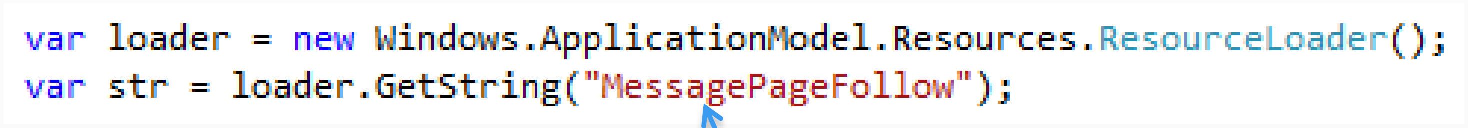
Strings Resources

- In the XAML file,



```
<Button x:Uid="NavigateBtn" Content="" Horizon
```

- In the code,

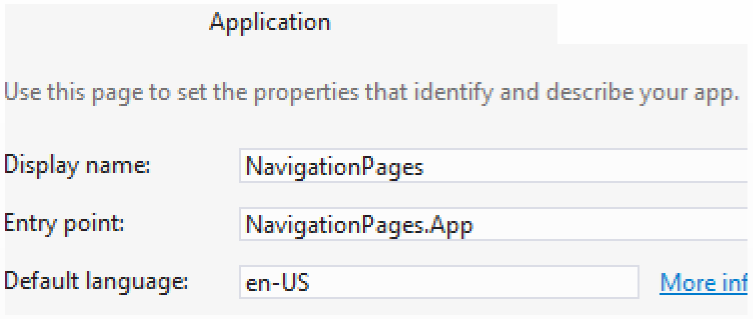


```
var loader = new Windows.ApplicationModel.Resources.ResourceLoader();
var str = loader.GetString("MessagePageFollow");
```

- FlowDirection property

Strings Resources

- Fields of the manifest as
  - Display name
  - Description
  - ….
    Can be localized
- Their values are in *`ms-resource:TokenName`*
  where *`TokenName`* is a resource name in the app resource files



Application

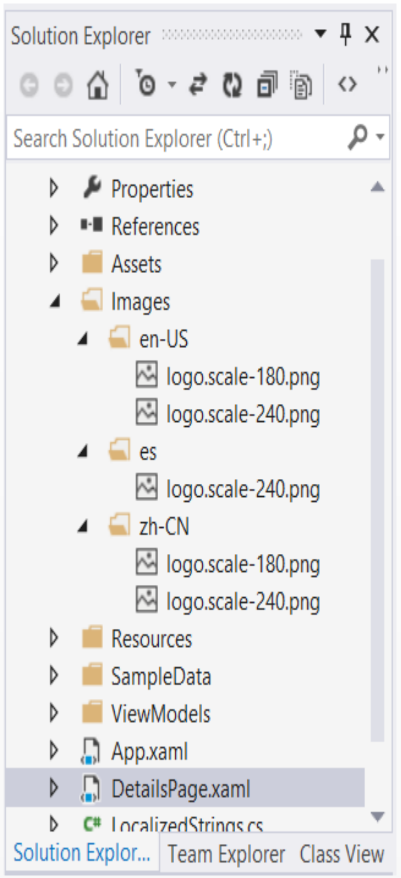
Use this page to set the properties that identify and describe your app.

Display name:	NavigationPages
Entry point:	NavigationPages.App
Default language:	en-US

More inf

Other Resources

- E.g.,
  - In the xaml or in the code
    - Images/nameFile
    - Not Images/en-US/nameFile



Globalization

- First step in internationalization
- The application executable code is written
- A truly global application should be culture-neutral and language-neutral
  - No translation of the user interface
- The executable code block contains only the application code to be used by all supported cultures

Globalization

- Elements susceptible to display differently according to the culture or the language
  - Dates
  - Hours
  - Numbers
  - Calendars
  - Currencies
  - …
- The process of adaptation of an app for new markets will be less complicated if the programmer takes precautions from the creation of the app

Globalization

- Namespace Windows.Globalization

Class	Description
ApplicationLanguages	Language-related preferences that the app can use and maintain.
Calendar	Date and time within a given calendar and clock.
CalendarIdentifiers	Calendar identifiers for the supported calendars
ClockIdentifiers	Clock identifiers for the supported clocks
CurrencyIdentifiers	Currency identifiers for the supported currencies
GeographicRegion	Region (usually a country, but may be a macroregion).
Language	Information related to BCP-47 language tags such as the language name and the script.
NumeralSystemIdentifiers	Numeral system identifiers for the supported numeral systems.

http://msdn.microsoft.com/en-us/library/windows/apps/windows.globalization.aspx

Globalization

- Date/Time
  - Standard date and time picker controls are conform to users' selected region and language
  - If the developper will program,

```
// To display dates and times using basic formatters
var sdatefmt = new Windows.Globalization.DateTimeFormatting.DateTimeFormatter("shortdate");
var stimefmt = new Windows.Globalization.DateTimeFormatting.DateTimeFormatter("shorttime");

// Obtain the date
var dateToFormat = DateTime.Now;
// Perform the actual formatting
var sdate = sdatefmt.Format(dateToFormat);
var stime = stimefmt.Format(dateToFormat);


var results = "Short Date: " + sdate + "\n" + "Short Time: " + stime;
```

Globalization

- *Windows. System. UserProfile. GlobalizationPreferences*
    - Static Class
    - To obtain the preferences defined by the user
    - E.g.,

```
var userRegion = Windows.System.UserProfile.GlobalizationPreferences.HomeGeographicRegion;
var userCalendars = Windows.System.UserProfile.GlobalizationPreferences.Calendars;
var userClocks  = Windows.System.UserProfile.GlobalizationPreferences.Clocks;
var userCurrencies =  Windows.System.UserProfile.GlobalizationPreferences.Currencies;
var userLanguages = Windows.System.UserProfile.GlobalizationPreferences.Languages;
var userWeekStartsOn = Windows.System.UserProfile.GlobalizationPreferences.WeekStartsOn;
```

Globalization

- To format numbers and currencies appropriately
  - Use NumberFormatting to display decimal, percent/permille numbers, currencies

```
// Determine the current users default currency
var userCurrency = userCurrencies.Currencies[0];

var fractionalNumber = 12345.67;
// Currency formatter using the current users preference settings for number formatting
var userCurrencyFormat = new
Windows.Globalization.NumberFormatting.CurrencyFormatter(userCurrency);
var currencyDefault = userCurrencyFormat.Format(fractionalNumber);

// Create a formatter initialized to a specific currency
var currencyFormatEuroFR =
    new Windows.Globalization.NumberFormatting.CurrencyFormatter("EUR", new[] {"fr-FR"}, "FR");
var currencyEuroFR = currencyFormatEuroFR.Format(fractionalNumber);

var results = "Fixed number (" + fractionalNumber + ")\n" + "With user's default currency: " + currencyDefault
+ "\n" + "Formatted Euro (fr-FR defaults): " + currencyEuroFR;
```