

ASP .NET Web Page Globalization

Context

Set the Culture and UI Culture for ASP.NET Web Page Globalization.

Challenges

In order to reach international markets through the Internet, supporting different cultures through our applications is essential for being successful. The .NET Framework comes with an integrated infrastructure for creating international applications.

Tips

This article will show how to build a Web Site that supports several languages, with Microsoft Visual Studio .Net 2008.

The .NET Framework comes with an integrated infrastructure that allows building Multi-Language Web Sites simply and easily. Basically, the CLR supports a mechanism for packaging and deploying resources with any type of application. The CLR and the base class library of the .NET Framework come with several classes for managing and accessing resources in applications. These classes are located in the System.Resources and System.Globalization namespaces.

In these Web Sites, each Web Form should have a resource for every culture (language) it should support. For example:

If you have a web form with name default.aspx and your web site support English and Portuguese, then you should have 2 resources files (1 for each culture):

- default.aspx.en-US.resx
- default.aspx.pt-PT.resx

Note: For list of supported cultures in .Net Framework, please visit [http://msdn.microsoft.com/en-us/library/kx54z3k7\(v=VS.80\).aspx](http://msdn.microsoft.com/en-us/library/kx54z3k7(v=VS.80).aspx)

How to build Multi-Language Web Sites with ASP.NET 2.0 and Visual Studio.Net 2008

To set the culture and UI culture for an ASP.NET Web page programmatically

- Open an *.aspx file in your Web Site project, in Design mode.
- Then select Tools à Generate Local Resources. A folder called App_LocalResources is generated.
- A new folder has been added in the web application's root folder called App_LocalResources. This folder holds all the resource files for all the ASP.NET Web Forms.
- Set the values for the default language of the Web Form. Visual Studio generates resources for several properties of each control. The resources are always prefixed with the resource key (definition below) of the control and postfixed with the name of the property. Visual Studio automatically generates the default resources for the controls of the page only. The resource generation tool creates an entry for every property that is marked with the Localizable attribute in the control. Therefore, if you want to create a custom, localizable control, you have to mark all Localizable properties with this attribute.
- In addition to generating the resource file, Visual Studio has changed the page's source code. For every Localizable property of each control placed on the page, it has added a localization expression, as shown in the following code snippet:

```
<asp:Button ID="ButtonEN" runat="server" Text="English" onclick="EnglishClick"
meta:resourcekey="Button1Resource1" />
```

- Localization expressions are identified by the meta:resourceKey attribute of the tag.
- If you want to add another language, you will have to do it manually.
- Note that the new resource file should be suffixed with the Culture Value. For example, if you want to add a Portuguese Culture to the Default.aspx Web Form, then you should name the resource file Default.aspx.pt-PT.resx.
- Then you only have to change the values of the objects to the new language.
- Now, to change the language dynamically you must do the following steps:
- Override the InitializeCulture method for the page.
- In the overridden method, determine which language and culture to set the page to. Note: the InitializeCulture method is called very early in the page life cycle, before controls are created or properties are set for the page. Therefore, to read values that are passed to the page from controls, you must get them directly from the request using the Form collection.
- Set the UI culture and culture in one of the following ways:
- Set the Culture and UICulture properties of the page to the language and culture string (for example, pt-PT). These properties are internal to the page, and can only be used in a page.

```
UICulture = "pt-PT";
```

```
Culture = "pt-PT";
```

- Set the `CurrentUICulture` and `CurrentCulture` properties of the current thread to the UI culture and culture, respectively. The `CurrentUICulture` property takes a language and culture information string. To set the `CurrentCulture` property, you create an instance of the `CultureInfo` class and call its `CreateSpecificCulture` method.
- `Thread.CurrentThread.CurrentCulture=CultureInfo.CreateSpecificCulture("pt-PT");`
- `Thread.CurrentThread.CurrentUICulture = new CultureInfo("pt-PT");`
- You can also generate resources that are common and reusable to several pages. Instead of writing the same text to every pages, you can generate a single resource file and place it on the `App_GlobalResources` folder. This is useful when you want to write validation expressions and you want to use them in several controls. To create a new global resource, right click your Web Application root, add an ASP.NET Folder and choose `App_GlobalResources`. Then right click this folder and add a new Resource file. The structure of this resource file is the same as the structure of the resource files placed inside the `App_LocalResources`. However, the name of this file can be any meaningful name and is not limited to the name of the ASP.NET web form. This file, as mentioned above, will hold shared resource keys for several ASP.NET web forms.

To set the culture and UI culture for an ASP.NET Web page declaratively

- To set the UI culture and culture for all pages, add a globalization section to the `Web.config` file, and then set the `uiculture` and `culture` attributes, as shown in the following example:

```
globalization uiCulture="pt" culture="pt-PT" />
```

- To set the UI culture and culture for an individual page, set the `Culture` and `UICulture` attributes of the `@Page` directive, as shown in the following example:

```
<%@ Page="" UICulture="pt" Culture="pt-PT" %>
```

More information:

- <http://msdn.microsoft.com/en-us/library/bz9tc508.aspx>
- Microsoft Net Framework 2.0 Web-Based Client Development, Glenn Johnson, Tony Northrup

Pedro Fernandes

Software Developer