

Resources and Internationalization



Outline

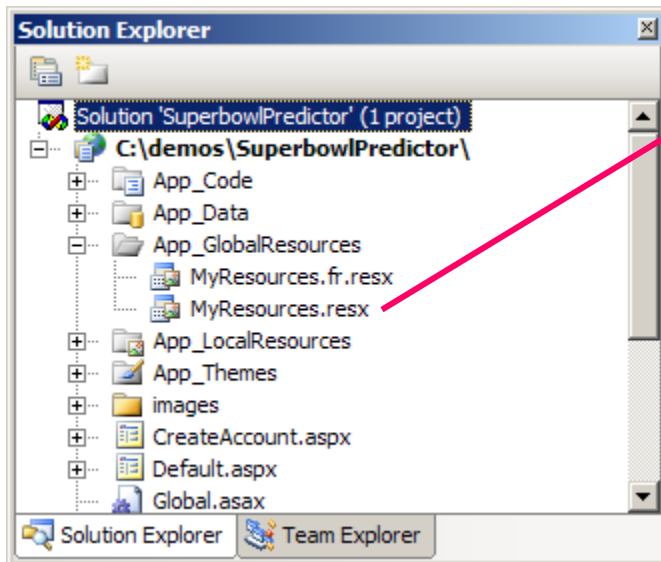
- **Localization / Internationalization**
- **App_GlobalResources**
- **App_LocalResources & Local Resource Generator**
- **Programmatic access**
- **Declarative access and resource expressions**

Localization

- **ASP.NET 2.0 simplifies the process of using resources for localizing your site**
 - Strongly-typed named resource access
 - Special resource compilation directories
 - New [Localizable] attribute for controls indicating which properties are candidates for localization
 - New tool for generating resource file for a page with all localizable attributes represented
 - New UICulture and Culture properties in @Page directive
 - Can be set to "auto" to set based on user language preference

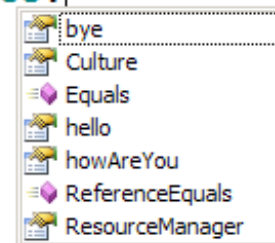
/App_GlobalResources

- **Place .resx files in /App_GlobalResources**
 - Generates an assembly with embedded resources
 - Creates a class with the same name as the resource file in the 'Resources' namespace (strongly typed)



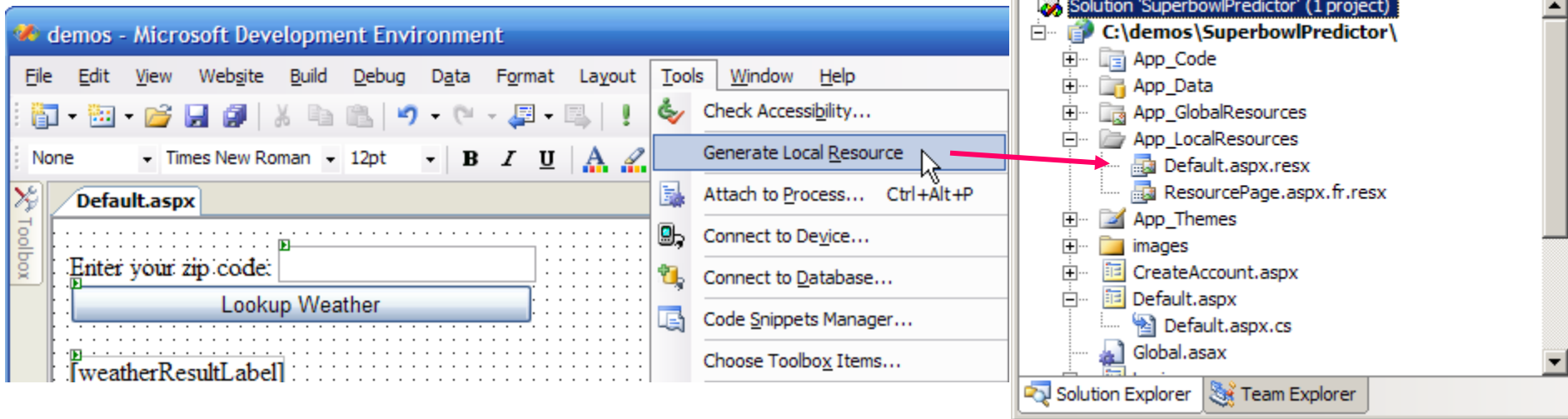
	Name	Value	Comment
▶	bye	Goodbye!	
	hello	Hello!	
	howAreYou	How are you?	
*			

```
protected void Page_Load(object sender, EventArgs e)
{
    Resources.MyResources.
```



Localization with Page Resources

- **Technique #1 for localizing resources**
 - Generate a local resource file for a page
 - This creates resource keys for each localizable property and marks these properties with meta:resourcekey=...
 - Add `UICulture="auto"` to your `@Page` directive
 - Create additional .resx files with culture name
 - Like `Default.aspx.fr.resx`



Resource Expressions

- **ASP.NET 2.0 supports both explicit and implicit declarative resource expressions**
 - Explicit syntax (for accessing global resources)
`<%$ Resources: [filename prefix,]resource-key %>`
 - Implicit syntax (for accessing local resources)
`<asp:Label ID="_label1" runat="server"
meta:resourcekey="resource-key-prefix" />`

Localization with Resources

■ Technique #2 for localizing with resources

- Create your own .resx files (either under /App_LocalResources or /App_GlobalResources)
- Add `UICulture="auto"` to your `@Page` directive
- Use the new explicit resource evaluation syntax in your pages
 - `<%$ Resources:[namespace, classname.]ResourceKey[, default] %>`
- Use the explicit class created when /App_GlobalResources is compiled
 - Eg. `goodDayLabel.Text = Resources.MyResources.good_day;`

Localizing Static Content

- Can use new Localize control to localize blocks of static HTML
 - Supports PassThrough or Encode mode (to html encode content if needed)

```
<asp:Localize runat="server" ID="_myLocalizableHtml" Mode="PassThrough"  
  meta:resourcekey="_myLocalizableHtmlResource1"  
  Text="Good <b>morning</b>, how <em>are</em> you today?<br />">  
</asp:Localize>
```


Changing Culture Dynamically

- **May decide to let user select language preference**
 - User Language may not be accurate
- **New InitializeCulture method available in Page class**
 - Called at the appropriate time during initialization to set culture

```
protected override void InitializeCulture()
{
    // Assuming we have stored culture preference in Profile
    string uc = Profile.PreferredCulture;
    Thread.CurrentThread.CurrentUICulture =
        new CultureInfo(uc);
    Thread.CurrentThread.CurrentCulture =
        CultureInfo.CreateSpecificCulture(uc);
}
```

Building Culture-Aware Custom Controls

- Use the new Localizable attribute for any culture-specific string properties

```
public class InternationalControl : Control {  
    → [Localizable(true)]  
    public string Text {  
        get { return (string)ViewState["text"] ?? ""; }  
        set { ViewState["text"] = value; }  
    }  
  
    protected override void Render(HtmlTextWriter writer) {  
        writer.RenderBeginTag(HtmlTextWriterTag.H2);  
        writer.Write(Text);  
        writer.RenderEndTag();  
        base.Render(writer);  
    }  
}
```

Summary

- **Localization / Internationalization**
 - Many new features to simplify the process of localizing/internationalizing your sites
- **App_GlobalResources**
 - Store application global resources with strongly typed access
- **App_LocalResources**
 - Store per-page resources
- **Declarative access and resource expressions**
 - Implicit or explicit expressions