

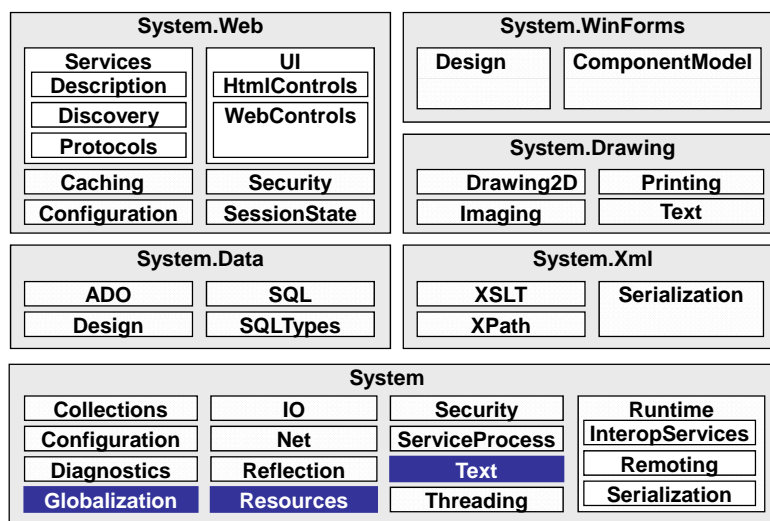
Internacionalización en ASP.NET 2.0

Localización, Internacionalización y Globalización

- Definiciones según la LISA (*Localization Industry Standards Association*, www.lisa.org):
 - **Localización (L10N)**: proceso que consiste en adaptar un producto para adecuarlo a las necesidades lingüísticas y culturales del país, región o idioma donde se pretenderá comercializar y usar
 - Traducción de la IU, mensajes de error, documentación, ayuda, etc.
 - **Internacionalización (I18N)**: es el proceso de diseñar un producto de forma genérica para facilitar el proceso de localización sin necesidad de modificar el componente central
 - Separar desde un principio el código fuente de la información sobre el idioma y las peculiaridades culturales
 - **Globalización (G11N)**: consiste en el estudio de los aspectos de mercado que afectan a la comercialización internacional de un producto. Incluye las fases de internacionalización y localización
 - Marketing, ventas, soporte, etc.

Internacionalización en ASP.NET 2.0

.NET Framework Namespace



Internacionalización en ASP.NET 2.0

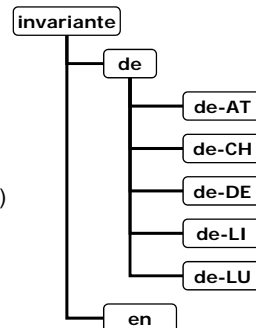
El espacio de nombres `System.Globalization`

- Soporte para Globalización de .NET
- Contiene clases que definen información relativa a:
 - Idioma (e.g. castellano vs inglés)
 - País o región (e.g. España vs México)
 - Calendarios utilizados (e.g. gregoriano)
 - Modelos de formato para las fechas (e.g. yy/mm/dd vs. dd/mm/yy)
 - Formato numérico (e.g. 12,000.00 vs 12.000,00)
 - Moneda (e.g. €)
 - Etc.

Internacionalización en ASP.NET 2.0

Estándares y tipos de culturas

- Los nombres de culturas siguen el estándar RFC 1766
 - Formato: `<codigoidioma2>-<pais/codigoregion2>` (e.g. de-DE)
 - `<codigoidioma2>`: código de 2 letras minúsculas (ISO 639-1)
 - `<pais/codigoregion2>`: código de 2 letras mayúsculas (ISO 3166)
 - La cultura invariante
 - Raíz de todas las demás culturas
 - Útil para casos en los que sea necesario representar información de forma independiente de culturas específicas (e.g. almacenamiento de datos que no se mostrarán directamente al usuario final)
 - La cultura neutral
 - Indica el idioma, pero no el país o región (e.g. en)
 - La cultura específica
 - Indica tanto el idioma como el país o región (e.g. en-US)



Internacionalización en ASP.NET 2.0

La clase `System.Globalization.CultureInfo`

- Proporciona información sobre una cultura concreta
 - sistema de escritura, calendario utilizado, etc.
- Acceso a objetos específicos de la cultura, que proporcionan métodos para operaciones comunes
 - formateo de fechas

Internacionalización en ASP.NET 2.0

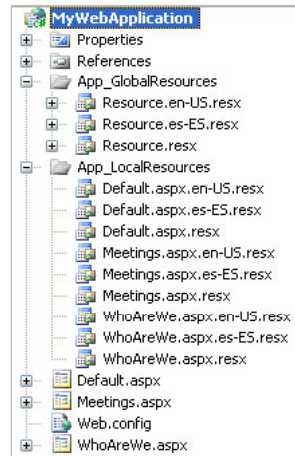
La clase `System.Globalization.CultureInfo`

- Propiedades `CurrentCulture` y `CurrentUICulture`
 - Identifican el idioma y la referencia cultural (país o región) de la aplicación (e.g. en-US)
 - `CurrentCulture`
 - Determina los resultados de las funciones que dependen del país o región, como el formato de fecha, número y moneda, etc.
 - Su valor deben ser referencias culturales concretas (e.g. en-US)
 - `CurrentUICulture`
 - Determina el idioma de las cadenas de texto de la IU
 - Su valor puede ser únicamente el código de idioma (e.g. en)
 - Pueden adoptar valores diferentes

Internacionalización en ASP.NET 2.0

Archivos de recursos

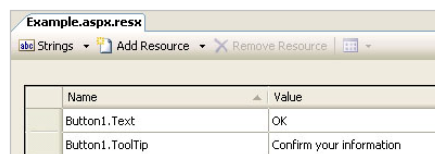
- Almacenan los diferentes recursos (cadenas de texto, rutas a imágenes, etc.) que se utilizarán en la aplicación según las preferencias de idioma y referencia cultural
- Archivos XML con extensión .resx
- Pares clave-valor (se puede añadir un comentario)
- En los nombres de clave no se distingue entre mayúsculas y minúsculas
- Un archivo diferente por idioma (e.g. en) o por idioma y referencia cultural (e.g. en-GB, en-US)
- En ASP.NET es posible crear archivos de recursos de diferente ámbito: **locales** y **globales**



Internacionalización en ASP.NET 2.0

Archivos de recursos locales

- Archivos de recursos locales
 - Orientados a almacenar recursos de páginas específicas
 - Se almacenan en la(s) carpeta(s) **App_LocalResources**, que puede(n) existir en cualquier carpeta de la aplicación
 - Formato: **<NombreWebForm>.aspx.<codigoCultura>.resx**
 - Si no se especifica **<codigoCultura>**, archivo de recursos por defecto
 - Ejemplo: el nombre del archivo de recursos locales asociado a una página Page.aspx para la cultura inglés-EE.UU. sería **Page.aspx.en-US.resx**



Internacionalización en ASP.NET 2.0

Creación de archivos de recursos locales

- Dos formas de crear archivos de recursos locales
 - Automática: desde la vista de diseño de un WebForm
 - Menú *Tools* > *Generate Local Resource* (Crea la carpeta *App_LocalResources* con el archivo de recursos por defecto para el WebForm)
 - Manual: desde el Solution Explorer, en cualquier carpeta de la aplicación
 - Creación de la carpeta *App_LocalResources*: Clic en botón derecho del ratón > *Add* > *Add ASP.Net Folder* > *App_LocalResources*
 - Creación de archivos de recursos locales en la carpeta *App_LocalResources*: *Add* > *New Item* > *Resources File*

Internacionalización en ASP.NET 2.0

Acceso a archivos de recursos locales

- Tres formas de acceso a archivos de recursos locales: localización implícita, localización explícita y mediante programación
1. Localización implícita
 - Atributo especial meta (e.g. `meta:resourcekey="Label1"`)
 - Nomenclatura obligatoria en fichero de recursos: Clave.Propiedad (e.g. `Label1.Text`)
 - Ejemplo:
 - En un archivo de recursos locales (Example.aspx.resx)

```
Button1.Text: "OK"
Button1.ToolTip: "Confirm your information"
```
 - En un WebForm (Example.aspx)

```
<asp:Button ID="Button1" runat="server"
Text="DefaultText" meta:resourcekey="Button1" />
```

Internacionalización en ASP.NET 2.0

Acceso a archivos de recursos locales

2. Localización explícita

- Se utiliza una *expresión de recurso* para establecer una propiedad de un control en un recurso concreto
- Formato de las expresiones de recurso:
`<%$ Resources: Class , ResourceID %>`
donde Class se puede omitir (sólo si el recurso es local)
- No es obligatorio utilizar la nomenclatura Clave.Propiedad en el fichero de recursos
- Ejemplo:
 - En un archivo de recursos locales (Example.aspx.resx)
`Button1Text: "OK"`
`Button1Tooltip: "Confirm your information"`
 - En un WebForm (Example.aspx)
`<asp:Button ID="Button1" runat="server"`
`Text="<%$ Resources: , Button1Text %>"`
`Tooltip="<%$ Resources: , Button1Tooltip %>" />`

Internacionalización en ASP.NET 2.0

Acceso a archivos de recursos locales

3. Mediante programación (Code Behind)

- Sintaxis:
`GetLocalResourceObject("<clave>")`
- Ejemplo:
 - En un archivo de recursos locales (Example.aspx.resx)
`Button1.Text: "OK"`
 - En el CodeBehind de un WebForm
`Button1.Text =`
`GetLocalResourceObject("Button1.Text").ToString();`

Internacionalización en ASP.NET 2.0

Archivos de recursos globales

■ Archivos de recursos globales

- Orientados a almacenar recursos presentes en varias páginas del sitio Web (e.g. las cadenas de texto: "Anterior", "Siguiete", "Aceptar", "Cancelar", etc.)
- Se almacenan en la carpeta **App_GlobalResources**, que debe ubicarse en el raíz de la aplicación
- Formato: <Nombre>.<codigoCultura>.resx
- Ejemplos:
 - Resource.en-US.resx
 - Resource.es.resx
 - Resource.resx (por defecto)

Internacionalización en ASP.NET 2.0

Creación de archivos de recursos globales

■ Creación de archivos de recursos globales

- No soportan la creación automática
- Como la creación manual de archivos de recursos locales, pero sólo desde la carpeta raíz de la aplicación
- Desde el Solution Explorer, en la carpeta raíz de la aplicación
 - Creación de la carpeta App_GlobalResources: Clic en botón derecho del ratón > *Add* > *Add ASP.Net Folder* > *App_GlobalResources*
 - Creación de archivos de recursos globales en la carpeta App_GlobalResources: *Add* > *New Item* > *Resources File*

Internacionalización en ASP.NET 2.0

Acceso a archivos de recursos globales

- Dos formas de acceso a archivos de recursos globales: localización explícita y mediante programación (no accesibles mediante localización implícita)

1. Localización explícita

```
<%$ Resources: Class , ResourceID %>
```

- A diferencia del acceso a rec. locales, Class NO se puede omitir

□ Ejemplo:

- En un archivo de recursos globales (Resource.resx)

```
Button1Text: "OK"
```

```
Button1Tooltip: "Confirm your information"
```

- En un WebForm (Example.aspx)

```
<asp:Button ID="Button1" runat="server"
```

```
Text="<%$ Resources:Resource, Button1Text %>"
```

```
Tooltip="<%$ Resources:Resource, Button1ToolTip %>" />
```

Internacionalización en ASP.NET 2.0

Acceso a archivos de recursos globales

2. Mediante programación (Code Behind)

- Sintaxis:

```
Resources.<nombreArchivoRecGlob>.<clave>
```

- Ejemplo:

- En un archivo de recursos globales (Resource.resx)

```
Button1Text: "OK"
```

- En el CodeBehind de un WebForm

```
Button1.Text = Resources.Resource.Button1Text;
```


Internacionalización en ASP.NET 2.0

Texto estático

■ El control `<asp:Localize>`

- Adecuado para incluir bloques de texto
- Editable en la vista de diseño
- No genera código de marcado
- Ejemplo (mediante localización implícita)

```
<h3><asp:Localize runat=server ID="LocWelcMess"  
  Text="Welcome"  
  meta:resourcekey="WelcomeMessage"/></h3>
```

Código HTML generado

```
<h3>Welcome</h3>
```

Internacionalización en ASP.NET 2.0

Selección de la cultura utilizada

■ Selección de la cultura utilizada por la aplicación. 3 opciones:

1. En el Web.config

```
<globalization uiCulture="es" culture="es-MX" />
```

- Afecta a todas las páginas del sitio Web
- Posibles valores de `uiCulture` y `culture`
 - `auto` : selección automática según las preferencias establecidas en el navegador (e.g. `uiCulture="auto"`)
 - `auto:<cultura>` : donde "cultura" indica la preferencia cultural que ASP.NET usará si no puede determinar la cultura preferida por el usuario (e.g. `uiCulture="auto:es-ES"`)

Internacionalización en ASP.NET 2.0

Selección de la cultura utilizada

2. En el WebForm

- En la directiva Page

```
<%@Page UICulture="es" Culture="es-MX" %>
```

- Afecta a una página concreta
- Posibles valores de uiCulture y culture: los mismos que en (1)

Internacionalización en ASP.NET 2.0

Selección de la cultura utilizada

3. En el Code Behind

- Sobreescribiendo (*override*) el método **InitializeCulture()** de la página
- Estableciendo las propiedades **CurrentCulture** y **CurrentUICulture** del thread actual a los valores deseados

```
using System.Globalization;
using System.Threading;
<<...>>
public partial class MainPage : System.Web.UI.Page
{
    <<...>>
    protected override void InitializeCulture()
    {
        CultureInfo cultureInfo = new CultureInfo("en-US");
        Thread.CurrentThread.CurrentCulture = cultureInfo;
        Thread.CurrentThread.CurrentUICulture = cultureInfo;
    }
}
<<...>>
```

Internacionalización en ASP.NET 2.0

Selección de la cultura utilizada

3. En el Code Behind (cont.)

- Repetir este código en todas las páginas no es una buena solución
- Recomendable utilizar Master Pages
 - Problema: la página maestra no hereda de `System.Web.UI.Page`, por lo que no puede sobrescribir su método `InitializeCulture()`
 - Solución: crear una clase hija de `System.Web.UI.Page`, que sobrescriba el método `InitializeCulture()` y hacer que todas las páginas del sitio Web hereden de ella

Internacionalización en ASP.NET 2.0

Selección de la cultura utilizada

```
using System.Globalization;
using System.Threading;
<<...>>
namespace MyWebApplication.Code
{
    public class SpecificPage : Page {
        <<...>>
        protected override void InitializeCulture()
        {
            CultureInfo cultureInfo = new CultureInfo("en-US");
            Thread.CurrentThread.CurrentCulture = cultureInfo;
            Thread.CurrentThread.CurrentUICulture = cultureInfo;
        }
        <<...>>
    }
}

-----

namespace MyWebApplication
{
    public partial class MainPage : Code.SpecificPage
    {
        <<...>>
    }
}
```

Internacionalización en ASP.NET 2.0

Selección de la cultura utilizada

- De las 3 opciones comentadas, la opción más específica tiene prioridad
 - Code Behind más prioridad que directiva Page
 - Directiva Page más prioridad que Web.config

Internacionalización en ASP.NET 2.0

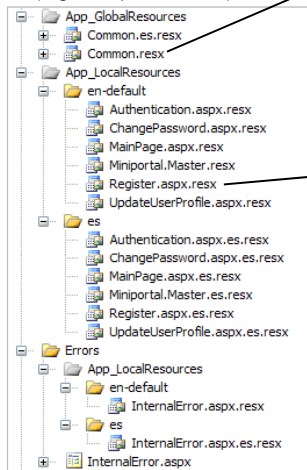
Ejemplo - Miniportal

- Miniportal
 - Localizado al inglés y al español (archivos de recursos locales y globales en inglés – por defecto – y en español)
 - Comportamiento
 - Si el usuario no está autenticado, se utilizan las preferencias culturales establecidas en el navegador
 - Si el usuario está autenticado, la aplicación debe utilizar las preferencias culturales que el usuario seleccionó en el momento de registrarse (idioma – país)

Internacionalización en ASP.NET 2.0

Ejemplo - Miniportal

- Archivos de recursos (inglés – por defecto)

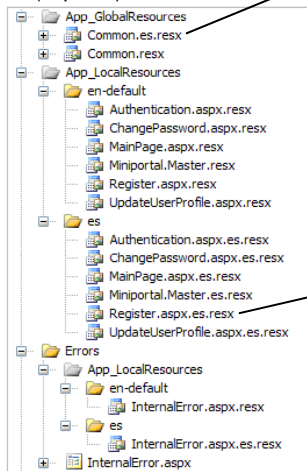


Name	Value
btrRegister.Text	Register
cvPasswordCheck.Text	Passwords do not match
lblLoginError.Text	User already exists
IdCountry.Text	Country
IdEmail.Text	Email
IdFirstName.Text	First name
IdLanguage.Text	Language
IdMenuExplanation.Text	User registration
IdPassword.Text	Password
IdRetypePassword.Text	Retype password
IdSurname.Text	Surname
IdUserName.Text	User name
Page.Title	Miniportal - User Registration
revEmail.ErrorMessage	Incorrect email syntax

Internacionalización en ASP.NET 2.0

Ejemplo - Miniportal

- Archivos de recursos (español)



Name	Value
btrRegister.Text	Aceptar
cvPasswordCheck.Text	Las contraseñas no coinciden
lblLoginError.Text	Usuario ya existente
IdCountry.Text	País
IdEmail.Text	Email
IdFirstName.Text	Nombre
IdLanguage.Text	Idioma
IdMenuExplanation.Text	Registro de usuario
IdPassword.Text	Contraseña
IdRetypePassword.Text	Repita la contraseña
IdSurname.Text	Apellidos
IdUserName.Text	Nombre de usuario
Page.Title	Miniportal - Registro de Usuario
revEmail.ErrorMessage	Sintaxis de email incorrecta

Internacionalización en ASP.NET 2.0

Ejemplo - Miniportal

■ Register.aspx

```
<%@ Page Language="C#" MasterPageFile="~/Miniportal.Master"
    AutoEventWireup="true" CodeBehind="Register.aspx.cs"
    Inherits="Es.UDC.DotNet.MiniPortal.Register" meta:resourcekey="Page" %>

<asp:Content ID="Content1"
    ContentPlaceHolderID="ContentPlaceHolder_MenuExplanation"
    runat="server">
    -
    <asp:Localize ID="lclMenuExplanation" runat="server"
        meta:resourcekey="lclMenuExplanation" />
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder_MenuLinks"
    runat="server">
</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder_BodyContent"
    runat="server">
    <div id="form">
        <form id="RegisterForm" method="post" runat="server">
            <div class="field">
                <span class="label">
```

Internacionalización en ASP.NET 2.0

Ejemplo - Miniportal

■ Register.aspx

```
        <asp:Localize ID="lclUserName" runat="server"
            meta:resourcekey="lclUserName" />
    </span>

    <span class="entry">
        <asp:TextBox ID="txtLogin" runat="server"></asp:TextBox>

        <asp:RequiredFieldValidator ID="rfvUserName" runat="server"
            ControlToValidate="txtLogin" Display="Dynamic"
            Text="<%$ Resources: Common, mandatoryField %>"
        </asp:RequiredFieldValidator>

        <asp:Label ID="lblLoginError" runat="server"
            ForeColor="Red" Style="position: relative" Visible="False"
            meta:resourcekey="lblLoginError">
        </asp:Label>

    </span>
</div>

<<..>>
```

Internacionalización en ASP.NET 2.0

Ejemplo - Miniportal

■ Web.config

- Por defecto, selección automática de preferencias culturales, según las preferencias establecidas en el navegador Web

```
<<...>>
  <system.web>
    <<...>>
    <globalization culture="auto" uiCulture="auto" />
    <<...>>
  </system.web>
<<...>>
```

Internacionalización en ASP.NET 2.0

Ejemplo - Miniportal

- **SpecificCulturePage.cs**: clase hija de de **System.Web.UI.Page**, que sobrescribe el método **InitializeCulture()**

```
<<...>>
using System.Threading;
using System.Globalization;

namespace Es.UDC.DotNet.MiniPortal.HTTP.Session
{
    public class SpecificCulturePage : Page
    {
        protected override void InitializeCulture()
        {
            /* If the user is authenticated, the Web application will use the
            * culture(language-country) which the user selected when he
            * registered into the application. Otherwise, the cultural
            * preferences established in the Web browser will be used.
            */
            if (SessionManager.IsUserAuthenticated(Context))
            {
                String language = SessionManager.GetUserLanguage(Context);
                String country = SessionManager.GetUserCountry(Context);
                String culture = language + "-" + country;
                CultureInfo cultureInfo;
            }
        }
    }
}
```

Internacionalización en ASP.NET 2.0

Ejemplo - Miniportal

```
try
{
    cultureInfo = new CultureInfo(culture);
    Thread.CurrentThread.CurrentCulture = cultureInfo;

    // If the string language-country is not a valid culture
}
catch (ArgumentException)
{
    cultureInfo = new CultureInfo(language);
    Thread.CurrentThread.CurrentCulture =
        CultureInfo.InvariantCulture;
}
Thread.CurrentThread.CurrentUICulture = cultureInfo;
}
}
}
```

Internacionalización en ASP.NET 2.0

Ejemplo - Miniportal

- **Register.aspx.cs:** hereda de **SpecificCulturePage.cs**
 - Y por lo tanto, hereda el método **InitializeCulture()**

```
<<...>>
namespace Es.UDC.DotNet.MiniPortal
{
    public partial class Register : SpecificCulturePage
    {
        <<...>>
    }
}
```