

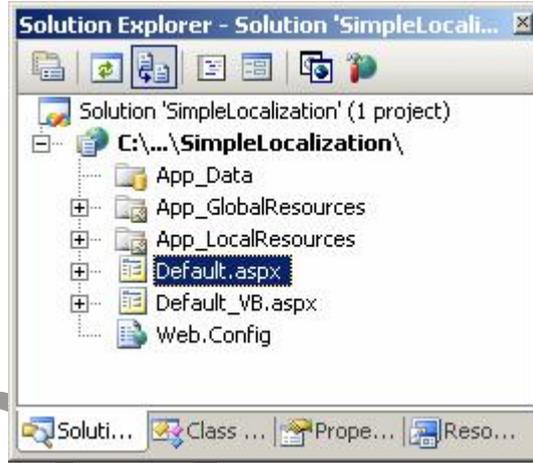
الفصل السابع : تعدد اللغات و تغيير واجه التعامل.

Localization, theme and skinning.

مقدمة

§ من منا لم يتعرض إلى مطلب غاية في البساطة ولكنه يحتاج إلى عمل كثير وهو تعدد اللغات؟؟؟ تعدد اللغات من الممكن أن يكون مهم جدا و لنفرض أن العميل طلب منك أن يكون الموقع يدعم اللهجة الخاصة لكل بلد؟؟ معنى هذا أن المستخدم يريد مثلا عند دخول زائر سعودي يظهر له رسالة ترحيب بلهجته و لتكن "كيف حالك؟" ، و إذا دخل زائر لبناني تظهر رسالة "أش لونك؟" ، و إذا دخل زائر مصري تظهر رسالة "أزيك يا ريس" ، المغزى من الموضوع أن هذا المطلب هو مطلب أساسى و يكون فى بعض الأحيان ضروري جدا للعمل ؛ و لذلك فإن هذا الموضوع لطالما شغل كل الناس ، و قديما كان يوجد طرق كثيرة و التى كانت - فى بعض الأحيان - تسبب ألم للمطور ؛ حيث كان يقوم بإعادة تصميم واجهة التعامل بالإضافة إلى معالجة الرسائل التى تظهر للمستخدم .

§ و بما أننا تعودنا على وجود أمثلة توضح أفكارنا فهيا بنا ننتقل إلى مثال يوضح كيفية التعامل مع موضوع تعدد اللغات فى تطبيق واحد.



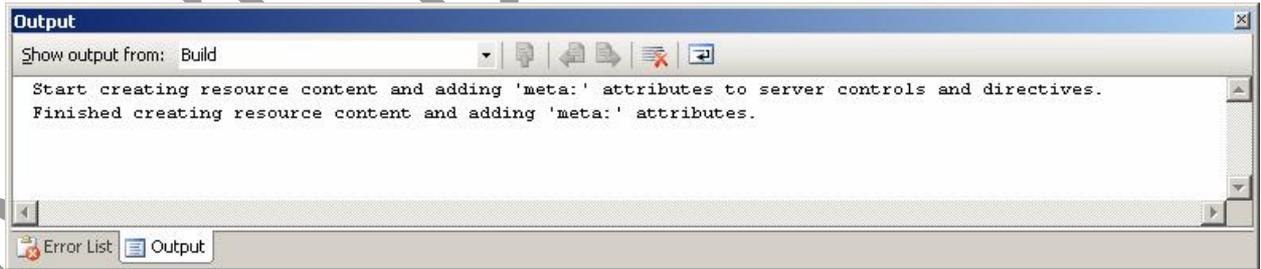
الشكل ١ [نقوم بفتح مشروع جديد ، و لتكن هذه هى مكوناته الأساسية]



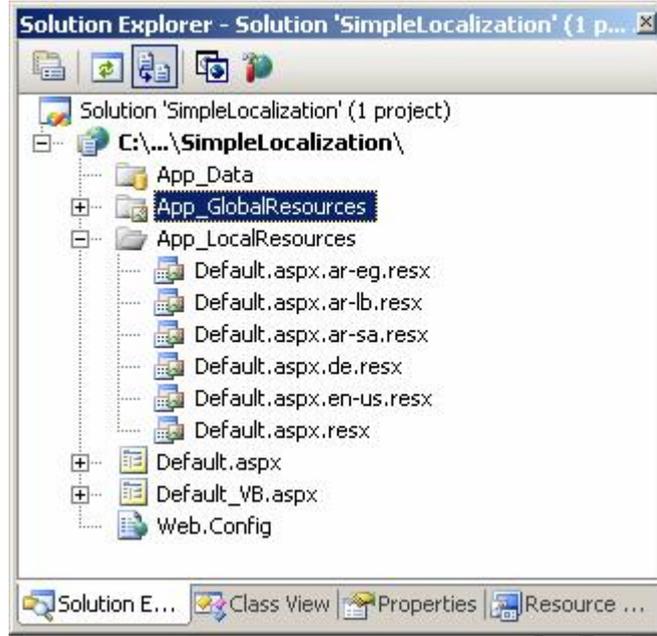
الشكل 2 [نقوم بعمل صفحة تحتوى على مجموعة أدوات - كما نرى - "تقويم و بعض الأزرار و بعض الـ (Label)]



الشكل 3 [هنا أهم خطوة فى الموضوع كله ، و هى أننا سوف نقوم بنقل محتويات واجهة من الصفحة إلى مصدر قابل للتغيير - سوف يتم توضيح كل شئ فى المثال -]

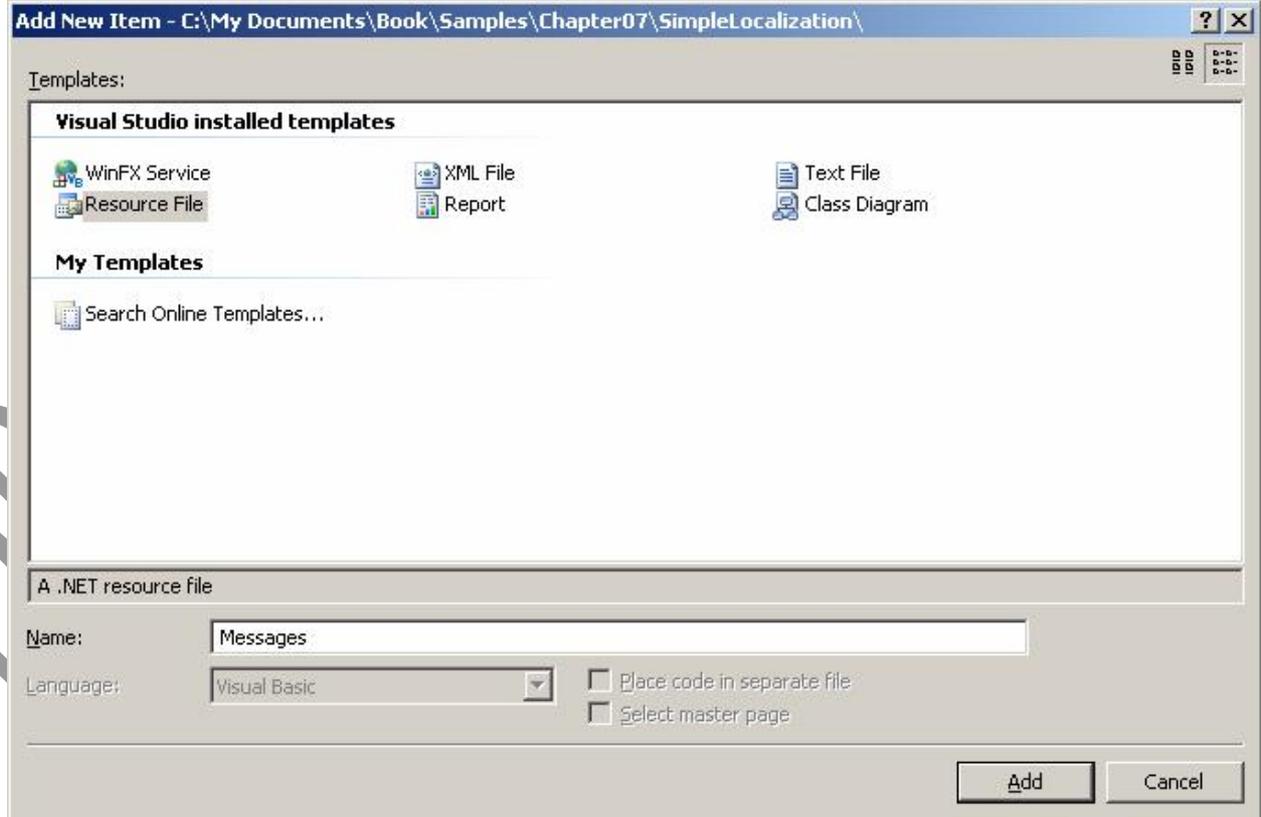


الشكل 4 [هذه النافذة تخبرنا أنه تم نقل المحتويات إلى مصدر بيانات و أنه جاهز للعمل]



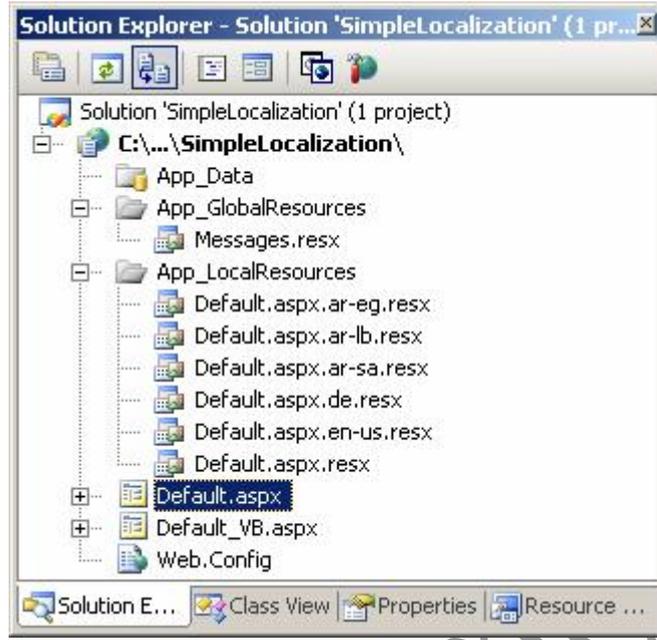
الشكل 5 [في الواقع يقوم الفيچوال ستوديو ينتج الصفحة (Default.aspx.resx)]

و هنا أود أن أشرح شئ بسيط جدا وهو ان لكل دولة لغة ، و هنا - إذا لاحظت - سوف تجد ملفات تحتوى على (ar-eg) و (ar-sa) ؛ و هذا يرمز إلى اللغة و الدولة و هنا تقوم أنت بتعريف المفردات التي تحتويها الصفحة و لا بد من معرفة أنه طالما انك قمت بعمل ملف (Default.aspx.resx) مع اعتبار أن الـ (*) تعبر عن اسم الصفحة و الـ (??) تعبر عن الدولة و اللغة ؛ فإن الصفحة سوف تقوم تلقائيا بالتعرف على لغة المستخدم و عرض الملف المناسب و في حالة عدم وجود ملف للغة يتم أخذ المفردات من الصفحة التي لاتحتوى على تعريف للغة (***.aspx.resx) - و سوف نرى بشكل أوضح في المثال-

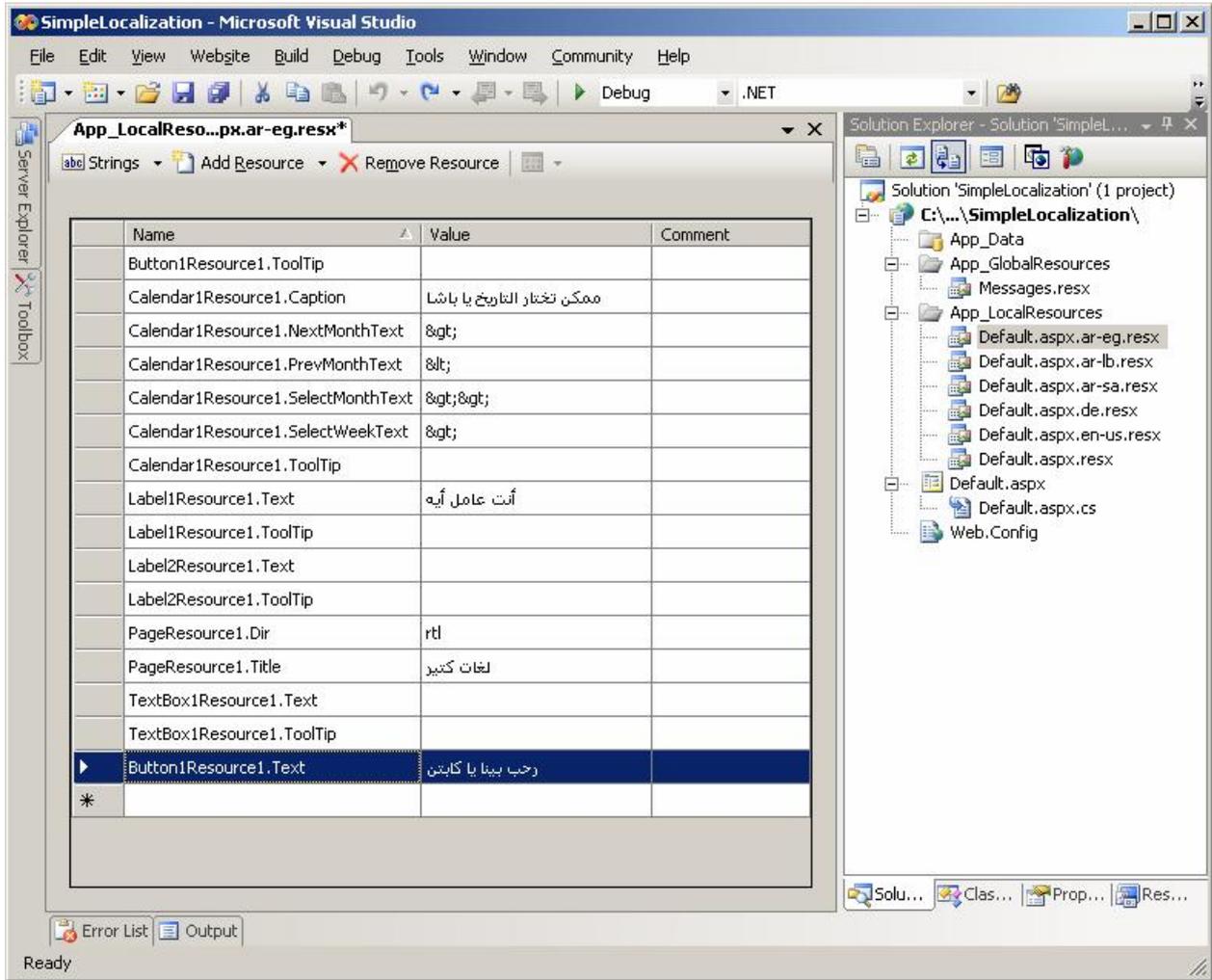


الشكل 6 [في بعض الأحيان نحتاج إلى وجود إلى معرفات عامة داخل التطبيق كله بشكل عام ، نقوم بالنقر بالزر الأيمن على

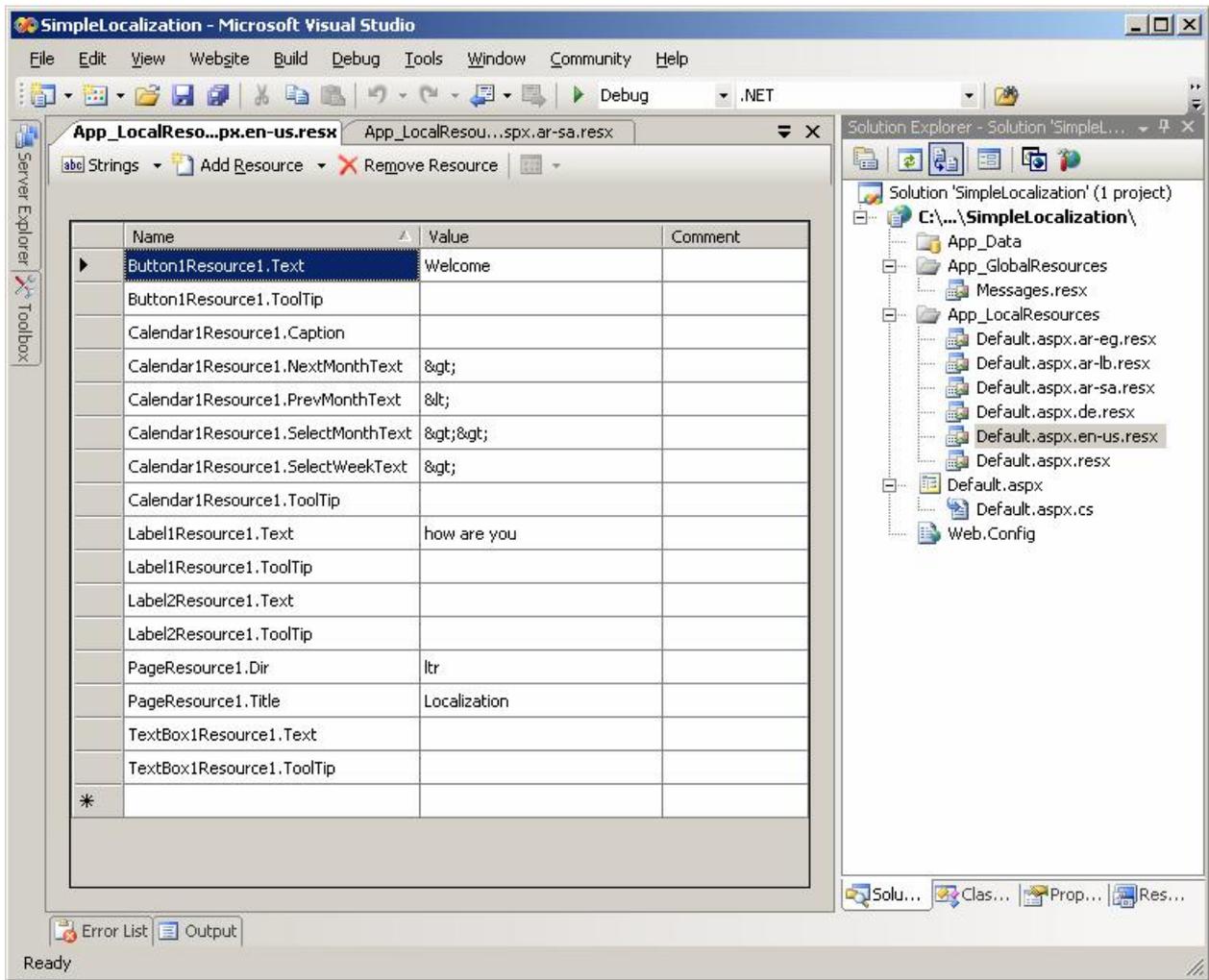
(Global Resources) و من ثم نقوم باختيار (Resource File) ، و نقوم بإدخال الاسم [



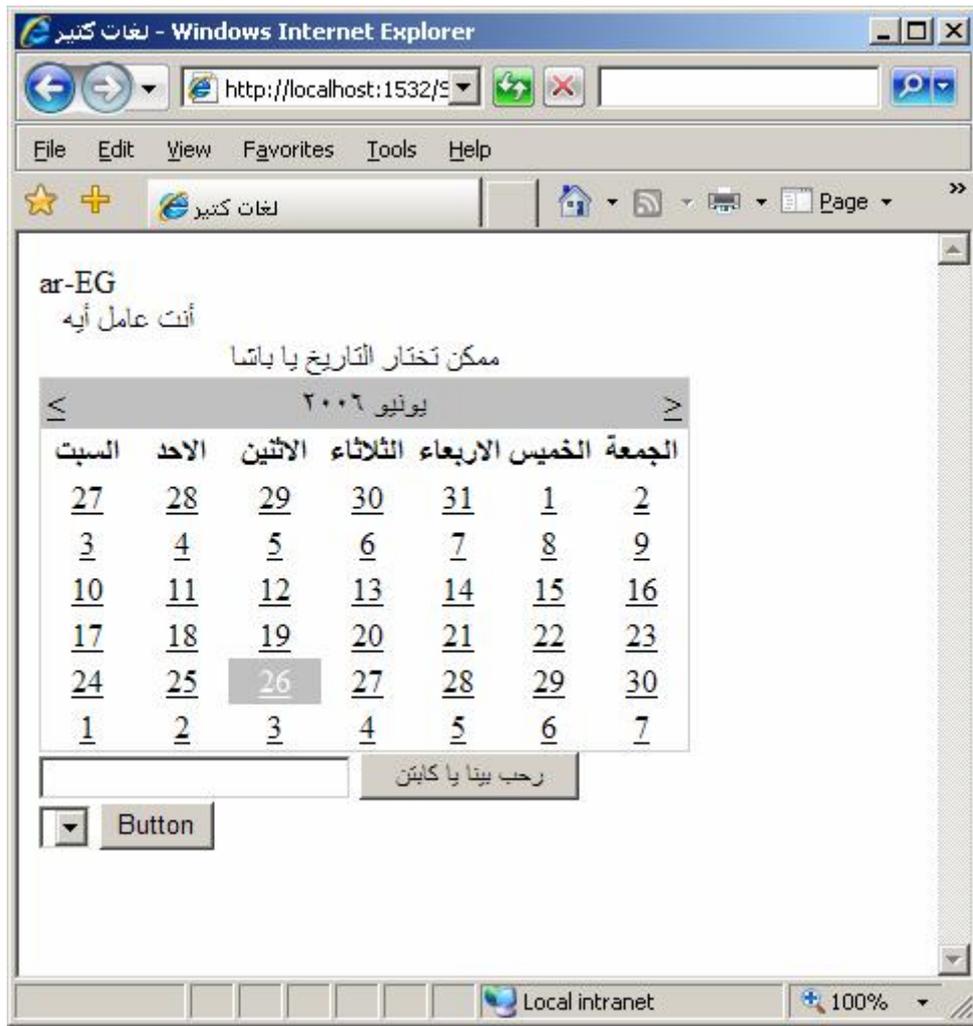
الشكل 7 [هذا هو شكل المشروع بعد أن قمنا بإضافة ملفات تعدد اللغات، حتى الآن لا يوجد أي نوع من التعقيدات]



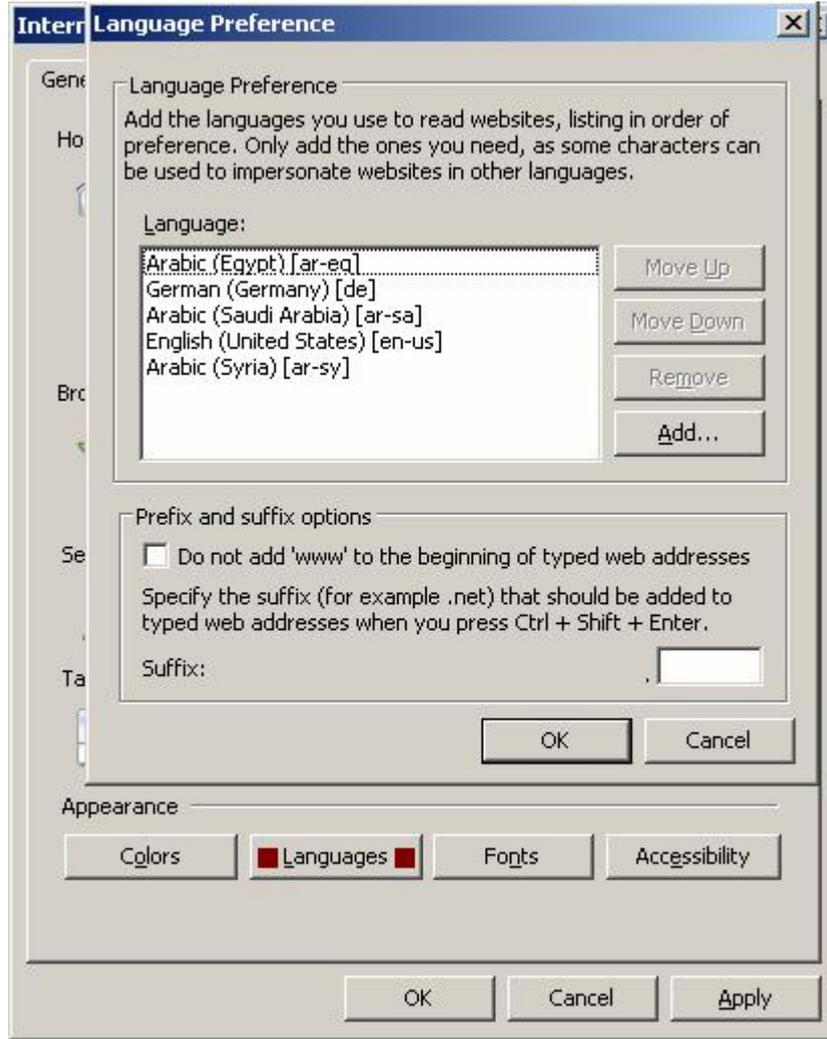
الشكل 8] و الآن هذا هو شكل ملف المصدر، و- كما نرى- يحتوى على مقابل لكل العناصر الموجودة على الصفحة ، و يمكننا هنا من تغيير المفردات الموجودة فى الصفحة ، و كما نرى فهذه الصفحة مخصصة للعرض للغة العربية للزائرين المصريين]



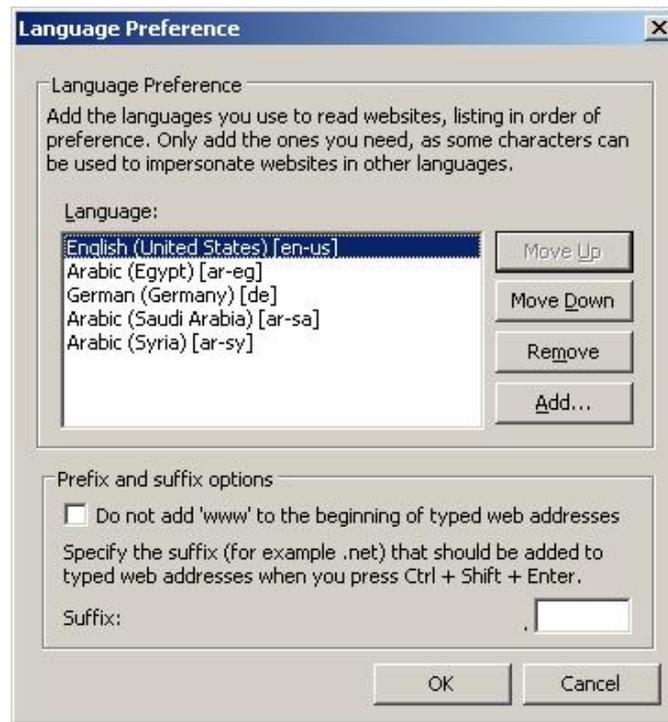
الشكل ٩ [هذه الصفحة مخصصة للزائرين متحدثين باللغة الإنجليزية و من أمريكا]



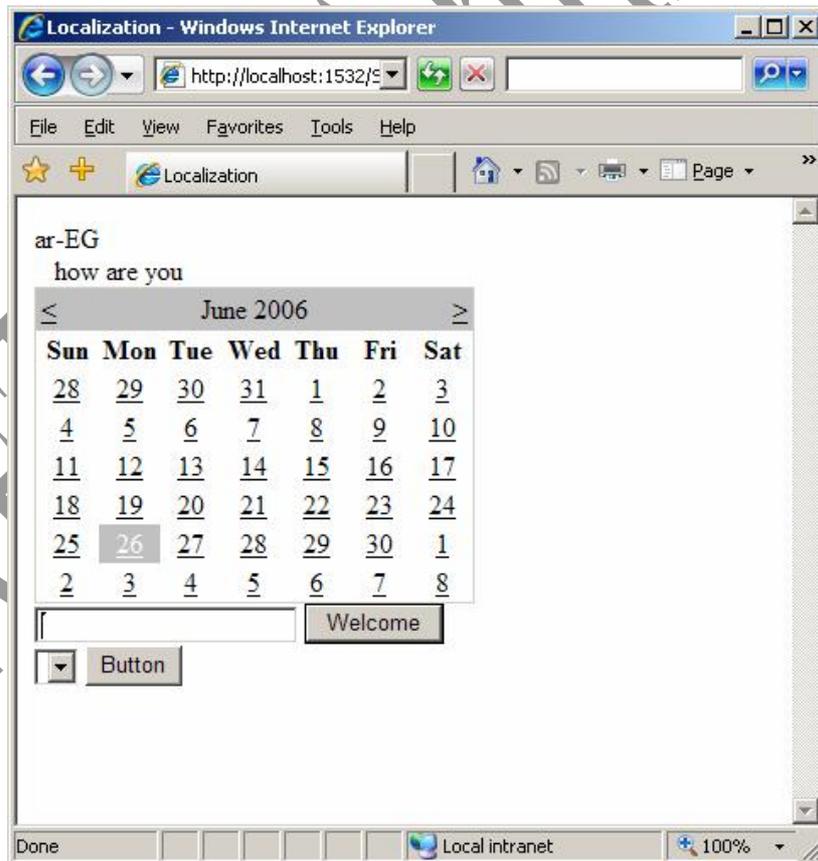
الشكل 10 [و هذا هو شكل الصفحة في حالة العرض للغة العربية للمصريين]



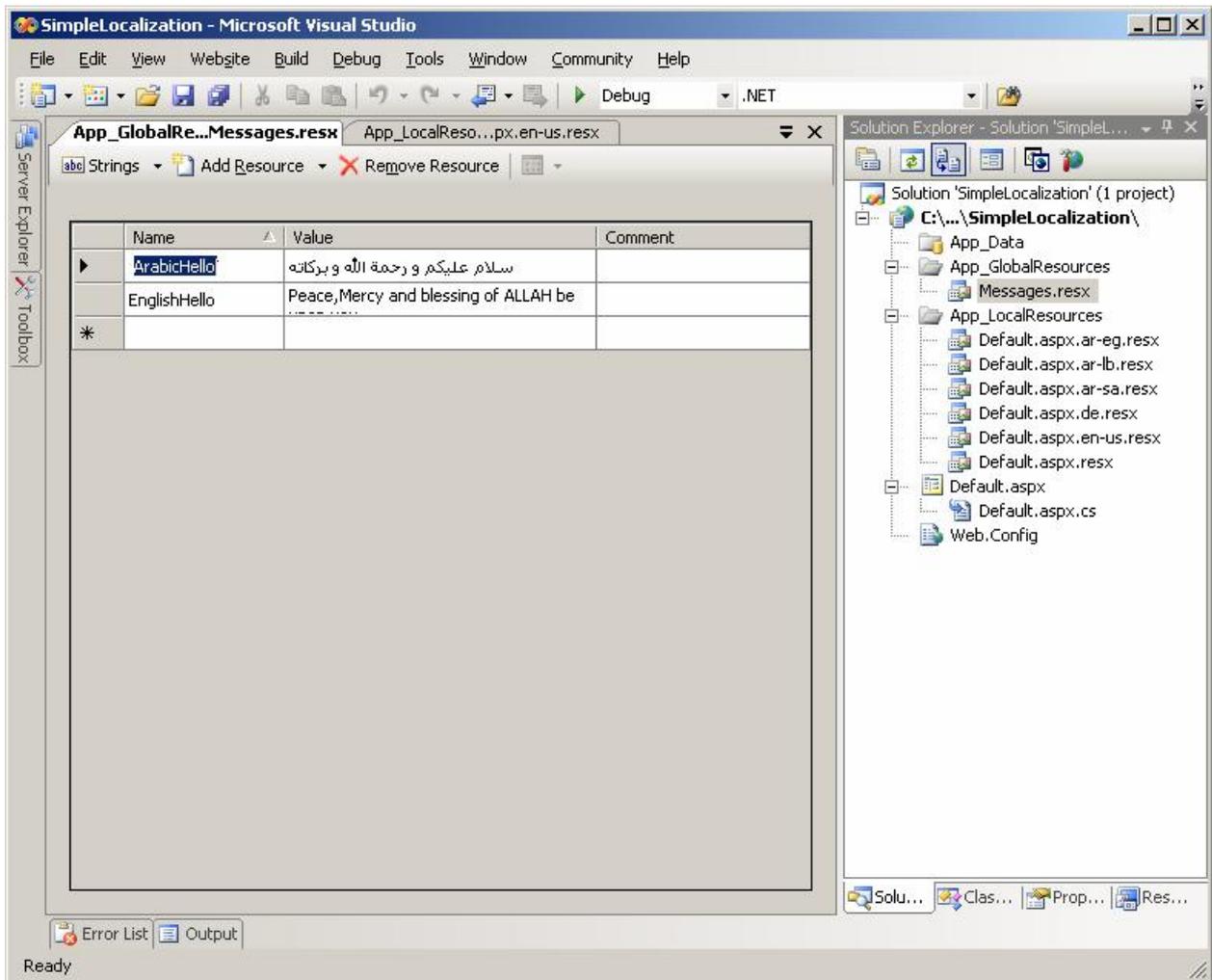
الشكل ١١ [أو بكل بساطة لتغيير اللغة من إنترنت اكسبلورر نقوم باختيار (Tools ثم Options) ، و بعد ذلك (Languages) ، و منها نقوم باختيار لغة المتصفح المراد التعامل بها ، و إذا لم تكن موجودة في القائمة يمكنك بكل بساطة الضغط على (Add) و إضافة اللغة]



الشكل 12 الآن نقوم بتغيير اللغة إلى الإنجليزية و الضغط على (OK)



الشكل 13 [كما نرى أن لغة الصفحة تغيرت إلى الإنجليزية]



الشكل 14 [نعود مرة أخرى للفيجوال ستوديو ، لنضيف بعض المفردات العامة للتطبيق]

```
<%@ Page Language="C#"
    AutoEventWireup="true"
    CodeFile="Default.aspx.cs"
    Inherits="_Default"
    meta:resourcekey="PageResource1"
    UICulture="auto" Culture="auto"%>
```

لا بد من ملاحظة شئ بسيط وهو أن التطبيق يقوم بتغيير اللغة بشكل تلقائي و ذلك بفضل المعرفين (Culutre) و (UICulutre) حيث أنهما يعطيا إمكانية التحويل التلقائي للصفحة

و الآن لنضيف بعض الأكواد و نرى ماذا يمكننا أيضا فعله مع الفجوال ستوديو

```
using System.Threading;
using System.Globalization;
using System.Resources;
protected void Page_Load(object sender, EventArgs e)
{
```

نضيف حدث بداية تحميل الصفحة

```
lblHello.Text =  
    Thread.CurrentThread.CurrentCulture.Name;  
نقوم بضبط النص (Label) بمحتوى اللغة الحالية للمستعرض و ذلك من خلال إحدى مكتبات  
الـ (Thread) و التي تحتوى على معلومات عن الثقافة الحالية للمستعرض  
if (!IsPostBack)  
{  
    التحقق من أنه إذا كان رد من نفس الصفحة (PostBack)  
    CultureInfo[] AllCul =  
        CultureInfo.GetCultures(CultureTypes.AllCultures);  
    حجز مصفوفة تحتوى على كل الثقافات المتاحة في منصة العمل (Frame work)  
    foreach (CultureInfo temp in AllCul)  
    {  
        دورة (Loop) نقوم بتحميل محتويات المصفوفة إلى قائمة منزلقة  
        ListItem t = new ListItem(temp.DisplayName, temp.Name);  
        DropDownList1.Items.Add(t);  
        تحميل بيانات اللغة إلى القائمة المنزلقة  
    }  
}  
}   
protected void Button2_Click(object sender, EventArgs e)  
{  
    حدث للزر (Button)  
    Thread.CurrentThread.CurrentCulture =  
        new CultureInfo(DropDownList1.SelectedValue);  
    تطبيق الثقافة و اللغة المختارة على التطبيق الخالي  
}
```

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As EventArgs)  
    نضيف حدث بداية تحميل الصفحة  
    lblHello.Text = Thread.CurrentThread.CurrentCulture.Name  
    نقوم بضبط النص (Label) بمحتوى اللغة الحالية للمستعرض و ذلك من خلال إحدى مكتبات  
الـ (Thread) و التي تحتوى على معلومات عن الثقافة الحالية للمستعرض  
If Not IsPostBack Then  
    التحقق من أنه إذا كان رد من نفس الصفحة (PostBack)  
    Dim AllCul() As CultureInfo = CultureInfo.GetCultures( CultureTypes.AllCultures)  
    حجز مصفوفة تحتوى على كل الثقافات المتاحة في منصة العمل (Frame work)  
    For Each temp As CultureInfo In AllCul  
        دورة (Loop) نقوم بتحميل محتويات المصفوفة إلى قائمة منزلقة
```

```
Dim t As ListItem = New ListItem(temp.DisplayName, temp.Name)
DropDownList1.Items.Add(t)
```

تحميل بيانات اللغة إلى القائمة المنزلقة

```
Next
```

```
End If
```

```
End Sub
```

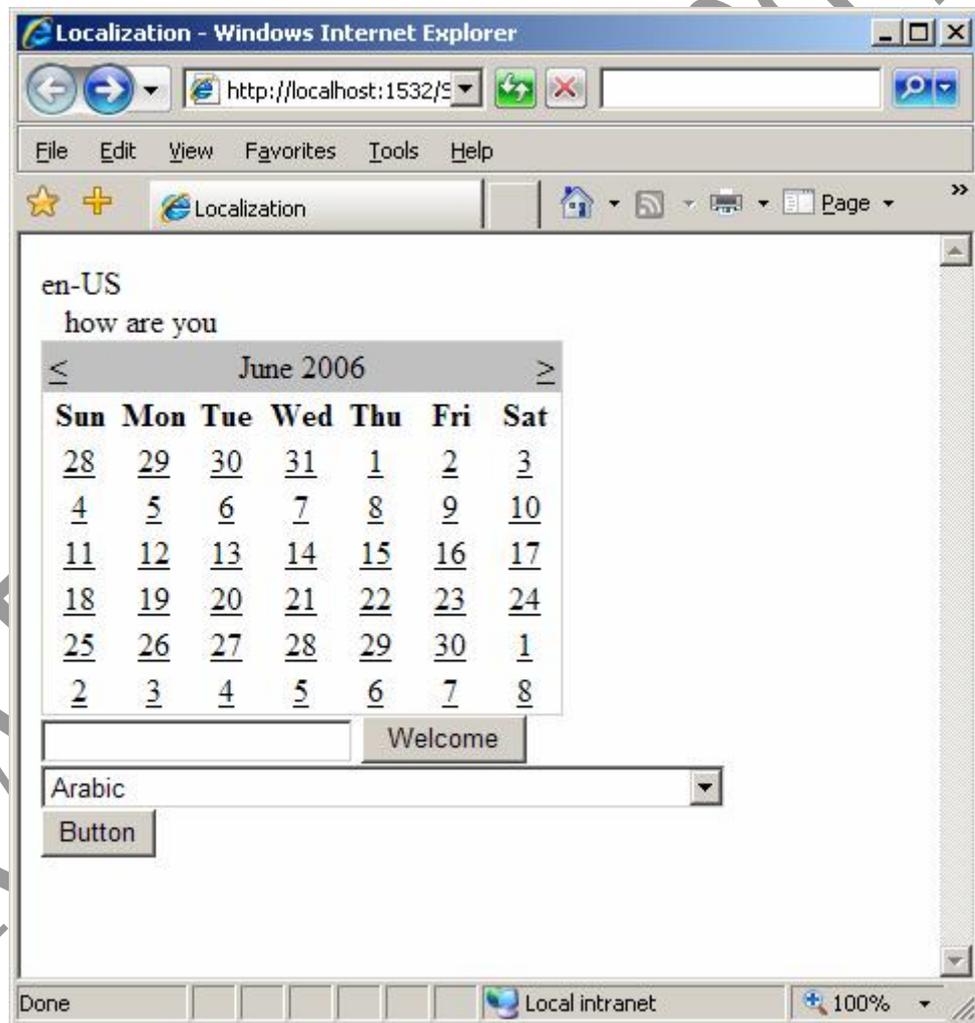
```
Protected Sub Button2_Click(ByVal sender As Object, ByVal e As EventArgs)
```

حدث للزر (Button)

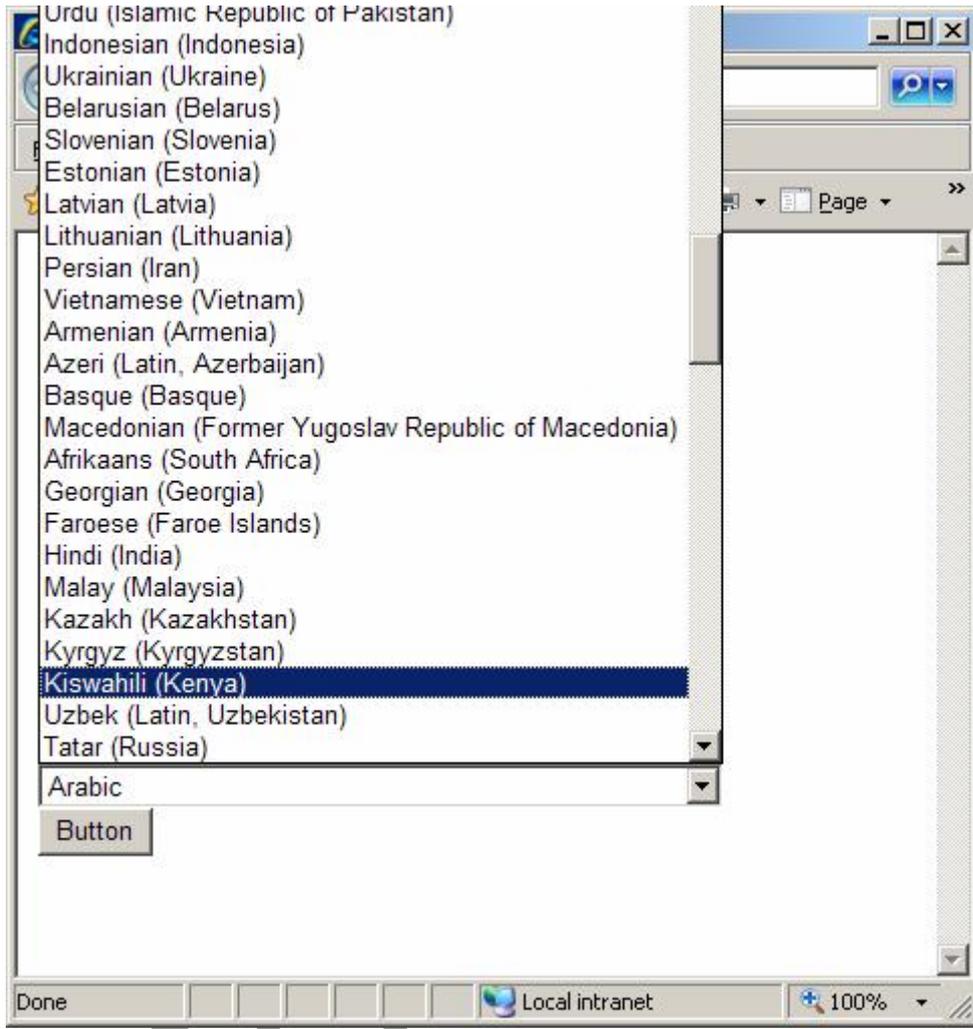
```
Thread.CurrentThread.CurrentCulture = New CultureInfo(DropDownList1.SelectedValue)
```

تطبيق الثقافة و اللغة المختارة على التطبيق الحالي

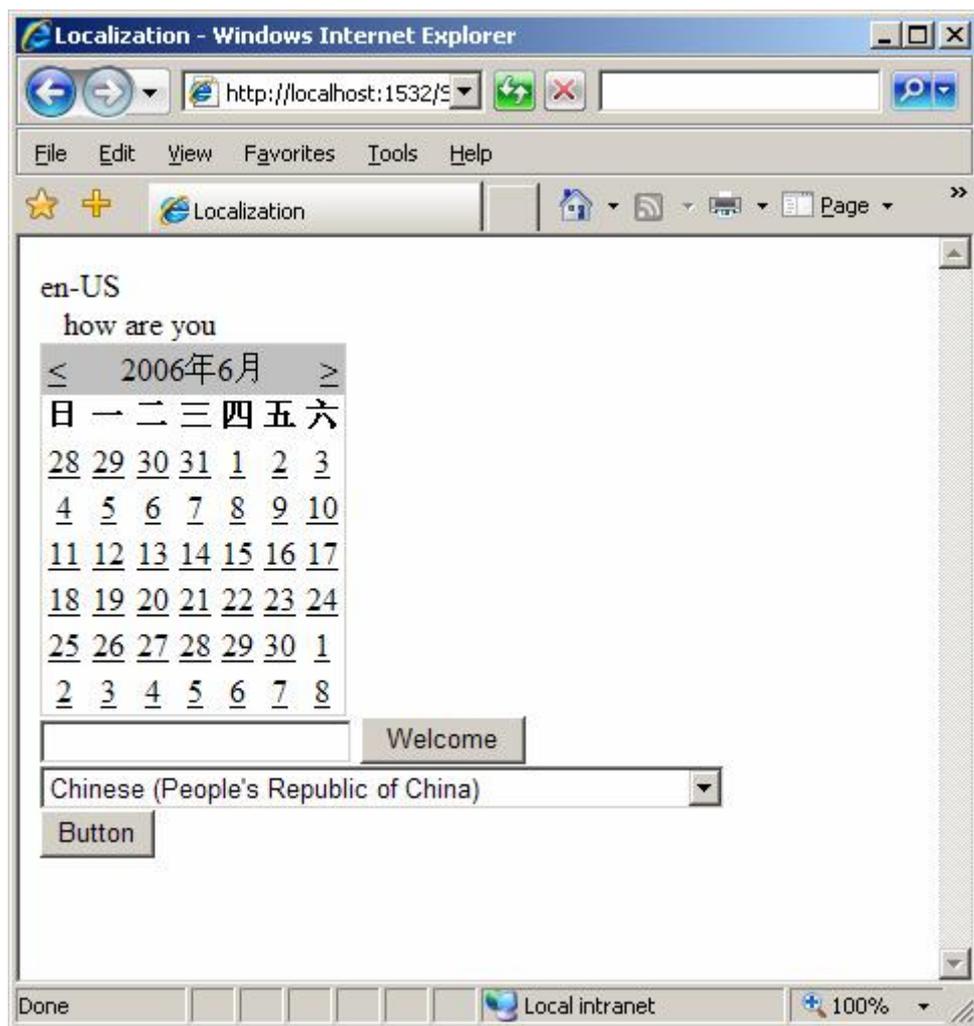
```
End Sub
```



الشكل 15] و- كما نرى أن- القائمة المنزلقة الآن تحتوي على بعض البيانات للغات]



الشكل 16 [هذه هي قائمة اللغات التي يمكن التعامل معها]



الشكل 17 [نلاحظ أننا اخترنا اللغة الصينية و أنها تعمل ؛ هذا لأن أداة التقويم مصممة بالعديد من اللغات ، و لأن منصة العمل تدعم ذلك مع ملاحظة أن بقية الصفحة باللغة الإنجليزية، و هذا لاننا قمنا بعمل إغلاق للتعريف التلقائي على اللغة و الثقافة . كما هو في المقطع التالي و المأخوذ عن الصفحة المعروضة]

```
<%@ Page Language="C#"
AutoEventWireup="true"
CodeFile="Default.aspx.cs"
Inherits="_Default"
meta:resourcekey="PageResource1" %>
```

نلاحظ أنه لا وجود للـ (Culture) و الـ (UICulture)

و الان ماذا لو أردنا أن نقوم بإستدعاء رسائل من مصدر اللغات إعتياداً على اللغة الخاصة بالمستخدم و الآن سوف نضيف قطعة كود إلى زر الترحيب كالتالي :-

```
protected void Button1_Click(object sender, EventArgs e)
```

```
{
```

حدث زر الترحيب

```
string strMSG = "window.alert('{0}')";
```

محتوى النص الفعال الذي سوف يظهر الرسالة

```
if (Thread.CurrentThread.CurrentCulture.Name == "ar-EG" )
```

التأكد من أن لغة المستعرض هي اللغة العربية

```
strMSG = string.Format(
```

```
strMSG, Resources.Messages.ArabicHello);
```

إختيار رسالة الترحيب باللغة العربية و كما نلاحظ أنها (Strongly Typed) بمعنى أنه أصبحت من ضمن خصائص المكتبة

```
else
```

```
strMSG = string.Format(
```

```
strMSG, Resources.Messages.EnglishHello);
```

في حالة أي لغة غير العربية و الثقافة المصرية يتم عرض الرسالة باللغة الإنجليزية
Page.ClientScript.RegisterClientScriptBlock(

```
this.GetType(), "JustHello", strMSG, true);
```

يتم أمر عرض الرسالة

```
}
```

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As EventArgs)
```

حدث زر الترحيب

```
Dim strMSG As String = "window.alert('{0}')"
```

محتوى النص الفعال الذي سوف يظهر الرسالة

```
If (Thread.CurrentThread.CurrentCulture.Name = "ar-EG") Then
```

التأكد من أن لغة المستعرض هي اللغة العربية

```
strMSG = String.Format(strMSG, Resources.Messages.ArabicHello)
```

إختيار رسالة الترحيب باللغة العربية و كما نلاحظ أنها (Strongly Typed) بمعنى أنه أصبحت من ضمن خصائص المكتبة

```
Else
```

```
strMSG = String.Format(strMSG, Resources.Messages.EnglishHello)
```

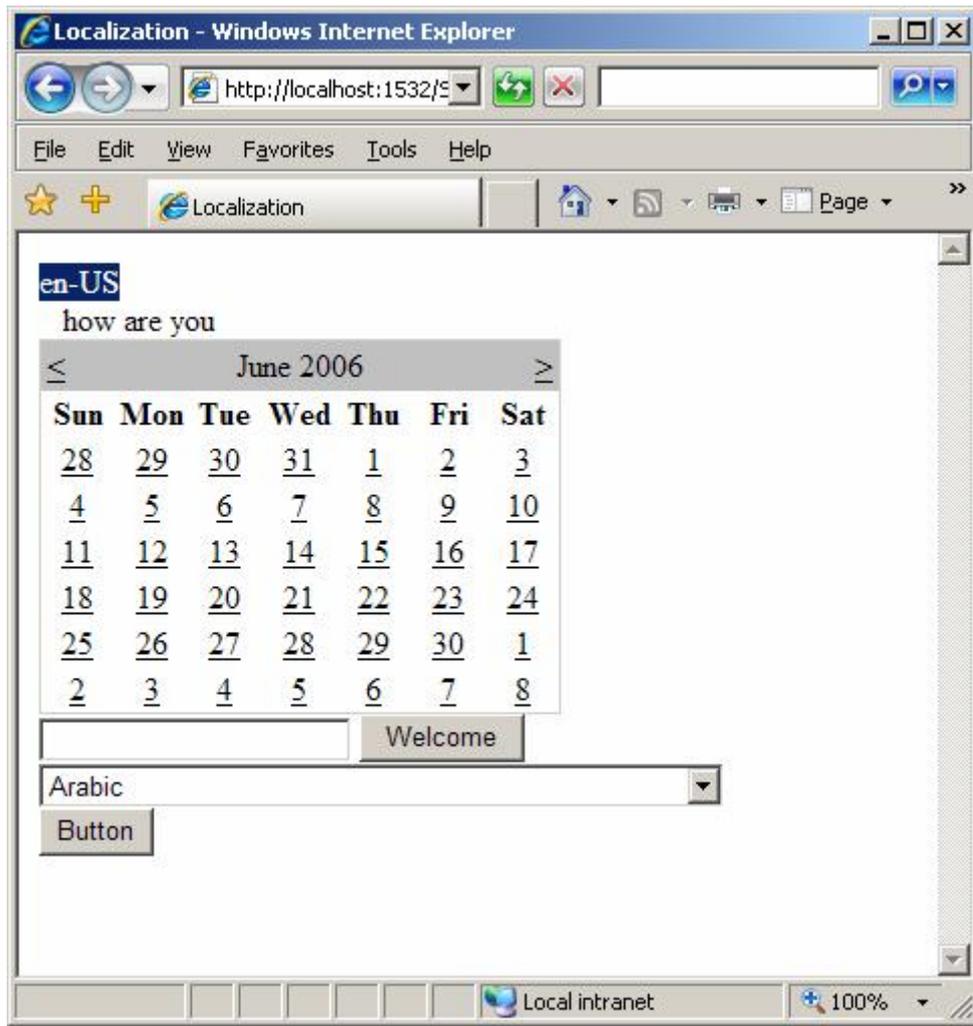
في حالة أي لغة غير العربية و الثقافة المصرية يتم عرض الرسالة باللغة الإنجليزية

```
End If
```

```
Page.ClientScript.RegisterClientScriptBlock(Me.GetType, "JustHello", strMSG, True)
```

يتم أمر عرض الرسالة

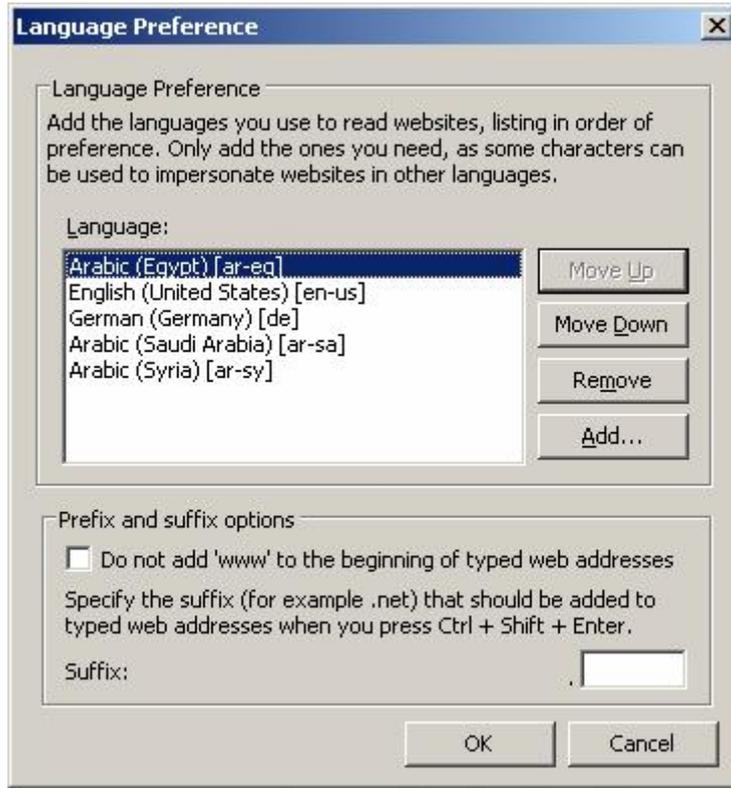
```
End Sub
```



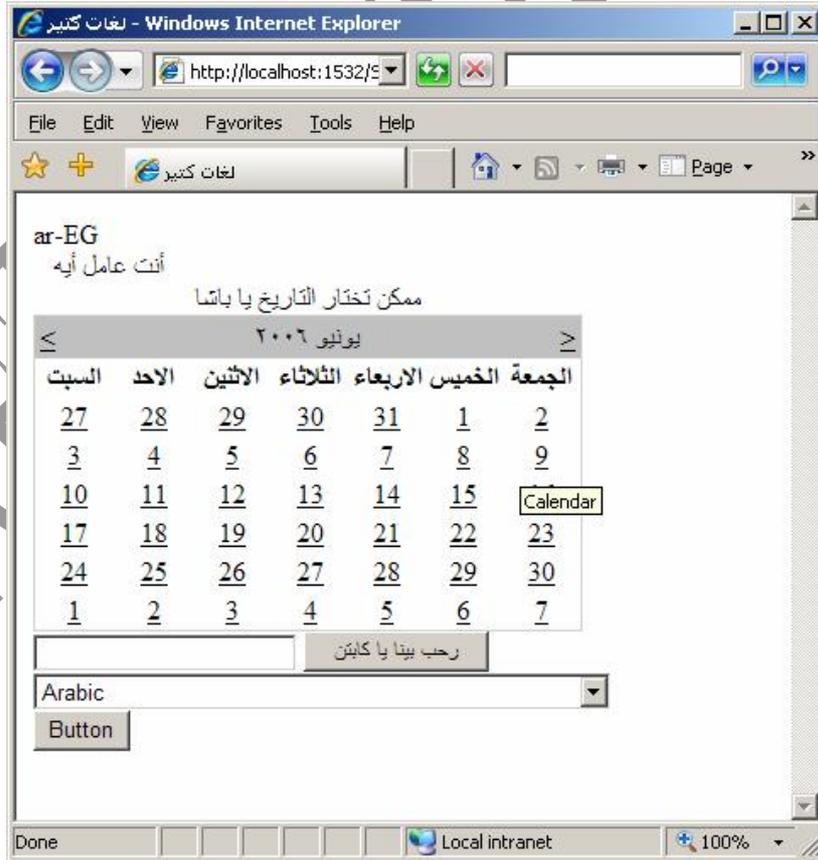
الشكل 18] هذه هي الصفحة في نمط عرض ثقافة أمريكية ولغة إنجليزية]



الشكل 19] سوف تظهر هذه الرسالة عند الضغط على زر (Welcome)]



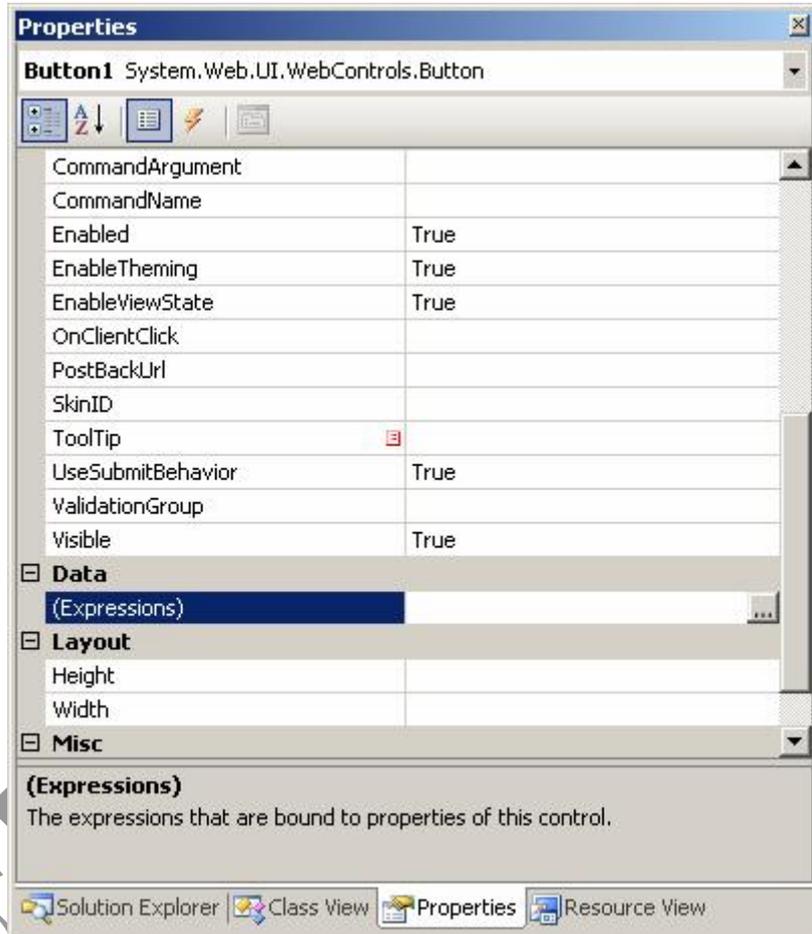
الشكل 20 [الآن نقوم بتغيير اللغة و الثقافة إلى اللغة العربية و الثقافة المصرية ثم نضغط على (OK)]



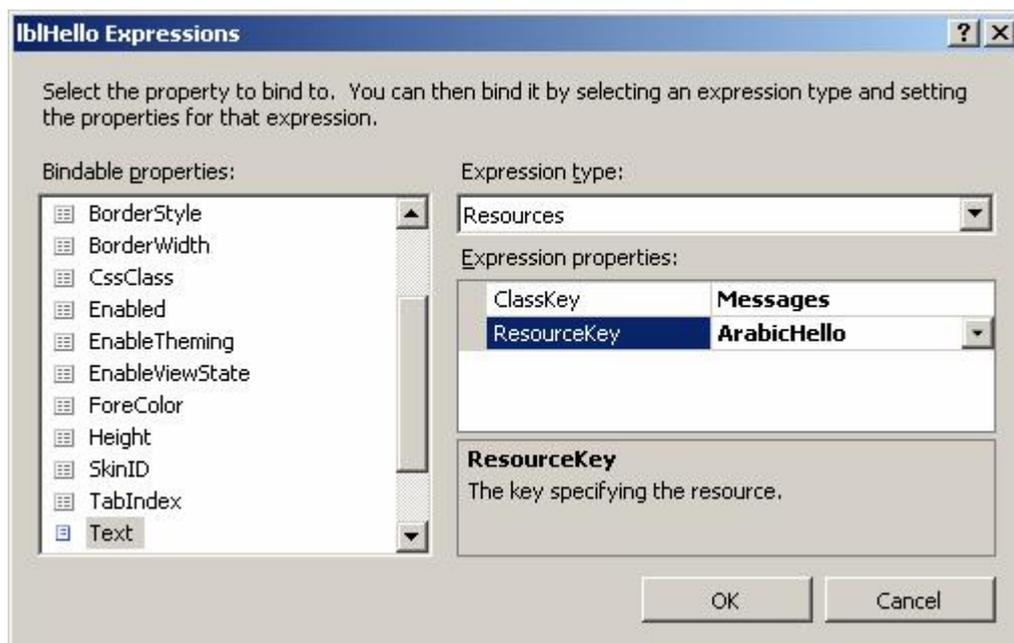
الشكل 21 [بعد أن تم عرض الصفحة باللغة العربية و الثقافة المصرية ، نضغط على زر "رحب بيينا يا كابتن"]



الشكل 22 [سوف تظهر هذه الرسالة عند الضغط على الزر]



الشكل 23] في حالة إضافة عنصر جديد إلى التطبيق و لا يوجد له مرادفات في معجم المفردات - الملفات التي تحتوي على مفردات الصفحة - ،
يمكننا بكل سهولة تحديد مصدر لهذا العنصر و ذلك من خلال الـ (Expressions) من صفحة خصائص العنصر [



الشكل 24] و هذا هو شكل إعدادات العنصر بعد ضبطها [

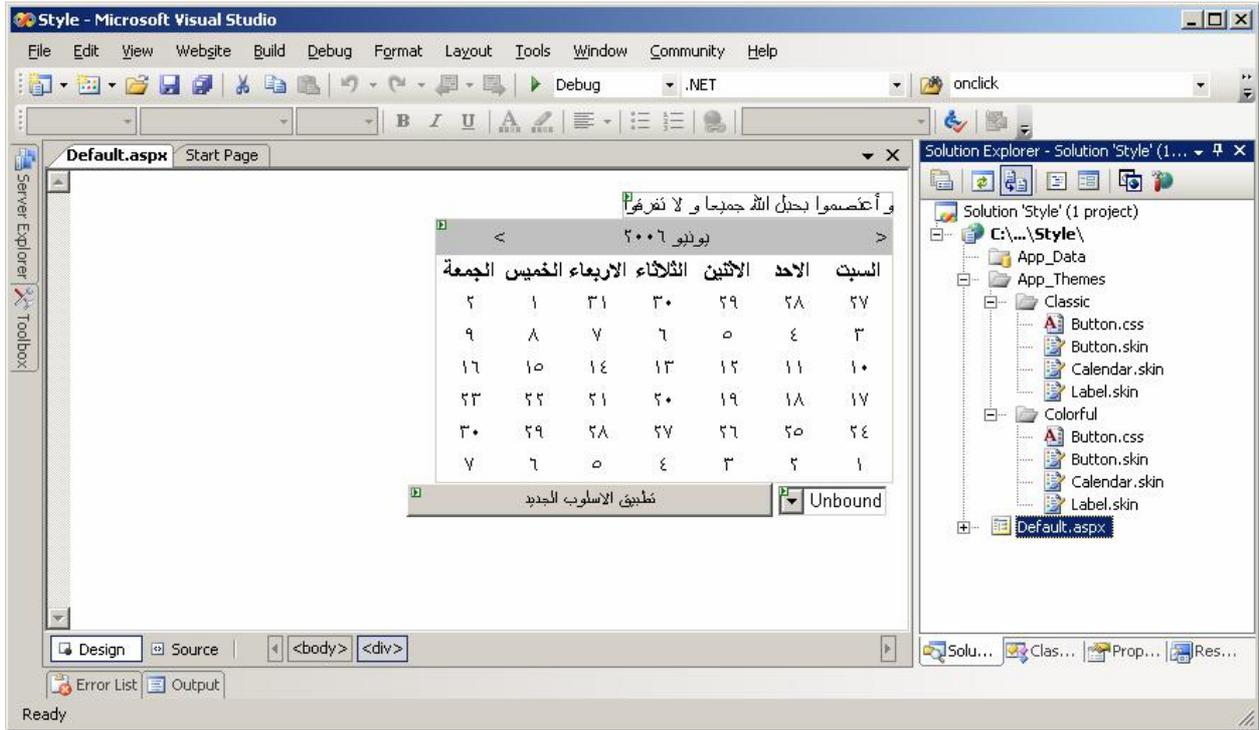
```
<asp:Label ID="lblHello" runat="server" Text="<%"$ Resources:Messages,
ArabicHello %>"></asp:Label>
```

كما نلاحظ في كود الـ (HTML) أن الخاصية (Text) تغيرت إلى توجيه إلى ملفات المصدر

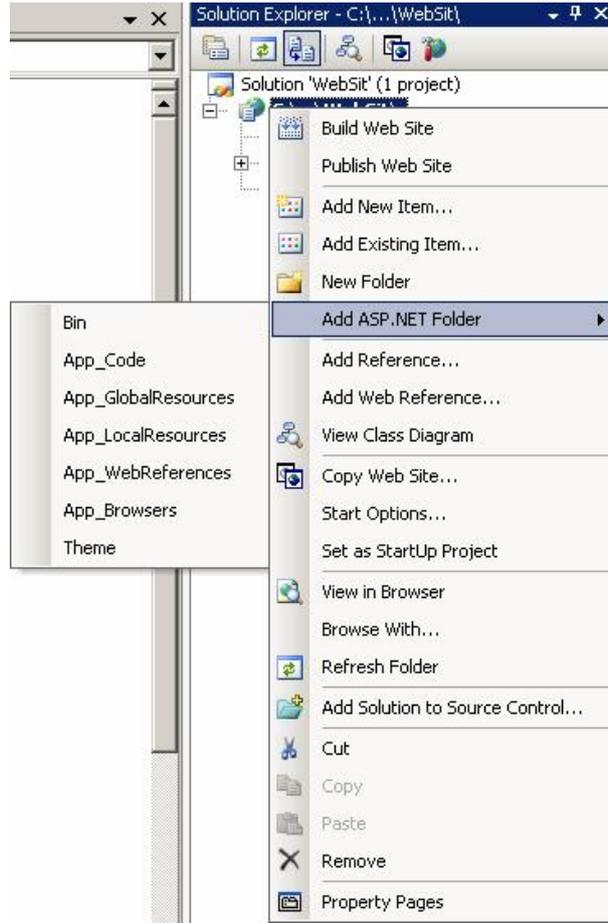
§ و بعد أن أنتهينا من تطبيق تعدد اللغات سننتقل إلى التحدث عن موضوع آخر ، ولكن قبل الانتقال إليه يجب الإشارة إلى أن موضوع تعدد اللغات كبير جدا و له جوانب كثيرة ؛ لذا سيتم إدراج بعض النصائح الهامة - في آخر هذا الفصل - عن كيفية تصميم التطبيقات و المشاكل التي دائما تحدث بسبب اللغة .

§ الآن سوف نتحدث عن ما يسمى بالقابلية لتغيير نمط الموقع (Skinning) هذه الميزة الجديدة - في وجهة نظري - من أفضل و أحسن المميزات التي وفرتها لنا ميكروسوفت - في فيجوال ستوديو ٢٠٠٥ - هذه الميزة سوف توفر لك إمكانية تغيير نمط الموقع ككل في أقل من ثواني معدودة و بسهولة و بأقل مجهود .

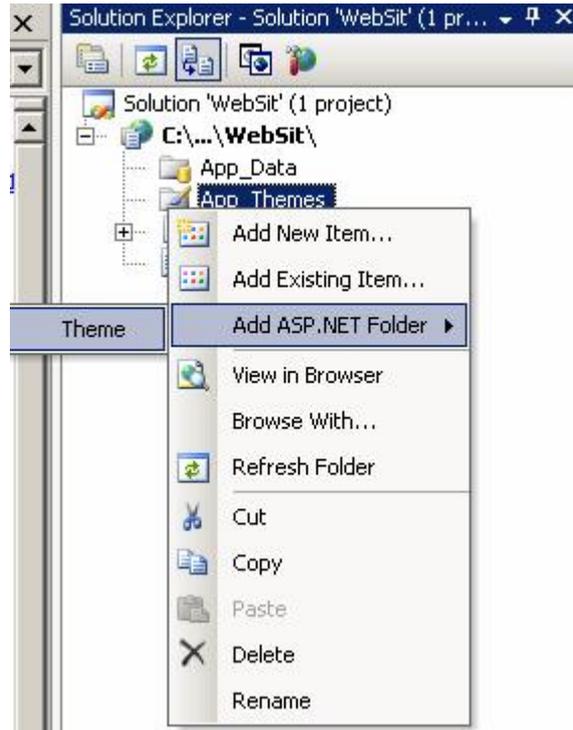
§ و كما تعودنا أيضا دائما نقوى شرحنا بالمثل فهيا بنا لنرى المثال التالي ونرى مدى أهميته و قوة هذه الميزة الجديدة .



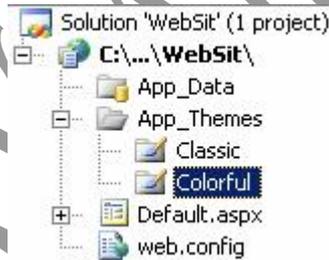
الشكل 25] و الآن لنقم بفتح مشروع جديد و لنضع أدوات كما هو موضح بالشكل - نضع أداة التقويم و أداة النص الثابت و قائمة منزلقة و زر - و بالنسبة للمشروع نقوم بإضافة مجلد ال (App_Theme) - كما هو موضح في الشكل التالي -]



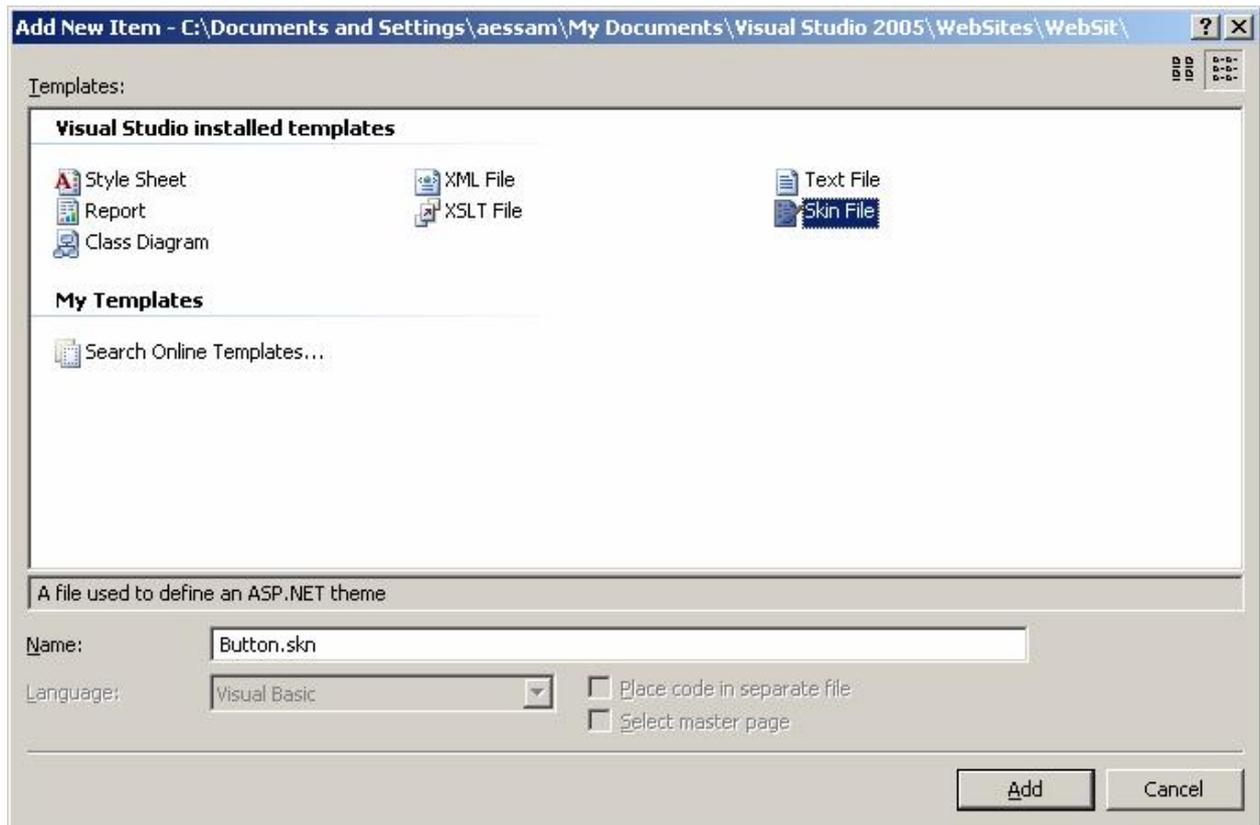
الشكل 26 [نضغط بالزر الأيمن للماوس على نافذة مستكشف الحلول (Solution Explorer) و بعد ذلك نختار (Add ASP.NET Folder) و منها نختار (App_Theme) و بعد إضافة المجلد إلى المشروع نقوم بإضافة مجلد الجلد (Skin Folder) كما هو موضح في الشكل التالي]



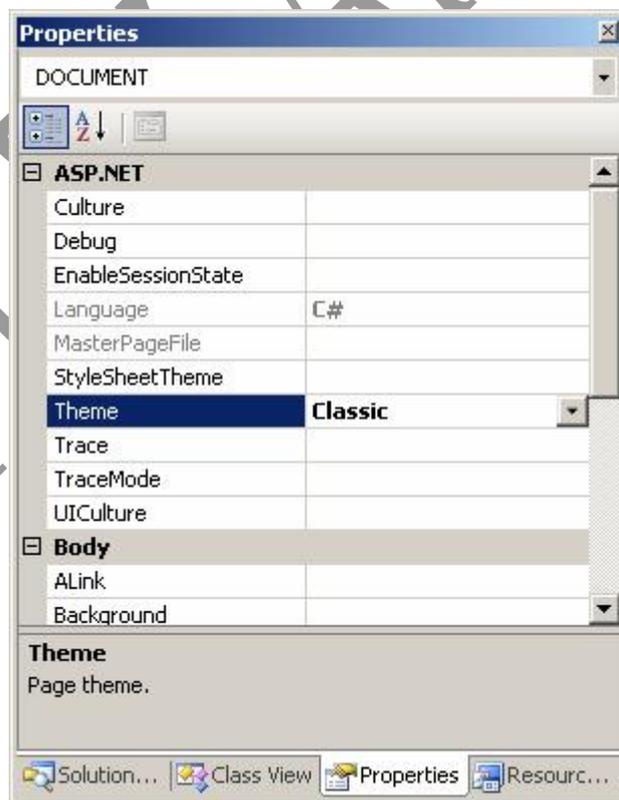
الشكل 27 [و بعد إضافة مجلد الأنماط ، نقوم بإضافة مجلد لكل نمط ، و كل مجلد سوف يحتوي على ملف لكل عنصر. و سوف يتم التوضيح في الخطوات التالية]



الشكل 28 [و بعد أن قمنا بإضافة مجلدات الأنماط ، نقوم بإضافة الملفات التي تحتوي على تصميم النمط لكل عنصر، بمعنى آخر أن لكل عنصر أو أداة (Label,Button,Text , ... etc) تصميم خاص به يتم تخزينه في ملف منفصل كما سنرى في بقية الأمثال]



الشكل 29 [هنا يتم إضافة الملفات التي تحتوي على توصيف كل عنصر]



الشكل 30 [في خصائص الصفحة لكي تقوم بتفعيل خاصية تغيير الجلد ؛ لا بد و أن تختار النمط الذي تود أن يتم تطبيقه على تطبيقك]



الشكل 31] هذا هو شكل الصفحة بعد تطبيق النمط (Classic) . و الآن لنرى ماذا تحتوية هذه الملفات [

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" Theme="Classic" %>
```

هنا في مقدمة الصفحة و في الجزء الخاص بالتوصيف يتم إضافة المعرف (Theme) و الذي يتم تحديد أسم النمط المراد تطبيقه و هنا يعتمد المحرك على أن الأنماط كلها موجودة في المجلد السابق ذكره (App_Theme)

```
<asp:Button runat="server" Width="190px" Font-Bold="True" />
<asp:Button skinid="Italic" runat="server" Width="190px"
Font-Bold="True" Font-Italic="True" />
<asp:Button skinid="Underline" runat="server" Width="190px"
Font-Bold="True" Font-Italic="True" Font-Underline="True" />
```

في الملف (Button.skn) هنا ترى توصيف النمط لعنصر الزر، و كما نرى انه أكواد (HTML) ، و نلاحظ أيضا انه يتم تعريف خصائص له ، و أن هناك ما يسمى (SkinID) وهو الذي يميز كل نمط عن الآخر في حالة إحتواء الملف على أكثر من نمط .

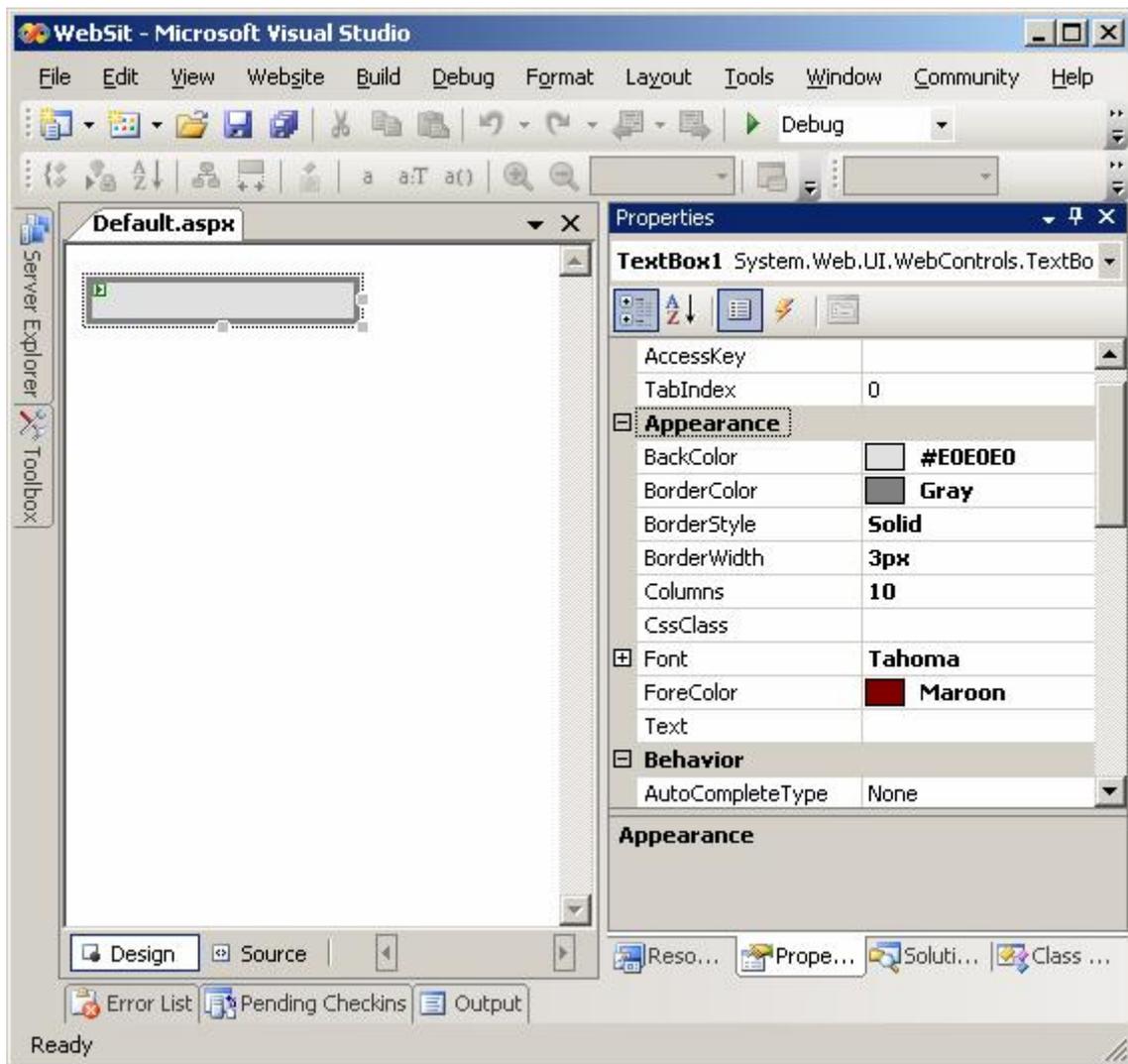
```
<asp:Calendar runat="server" Width="400px"
Font-Names="Times New Roman" Font-Size="10pt"
Height="220px" BorderColor="Black" ForeColor="Black"
```

```
BackColor="White" NextPrevFormat="FullMonth"
TitleFormat="Month" DayNameFormat="FirstLetter">
<SelectedDayStyle ForeColor="White" BackColor="#CC3333" />
<SelectorStyle ForeColor="#333333" Font-Names="Verdana"
Font-Size="8pt" Font-Bold="True"
Width="1%" BackColor="#CCCCCC" />
<OtherMonthDayStyle ForeColor="#999999" />
<TodayDayStyle BackColor="#CCCC99" />
<DayStyle Width="14%" />
<NextPrevStyle ForeColor="White" Font-Size="8pt" />
<DayHeaderStyle ForeColor="#333333" Height="10pt"
Font-Size="7pt" Font-Bold="True"
BackColor="#CCCCCC" />
<TitleStyle ForeColor="White" Height="14pt"
Font-Size="13pt" Font-Bold="True"
BackColor="Black" />
</asp:Calendar>
```

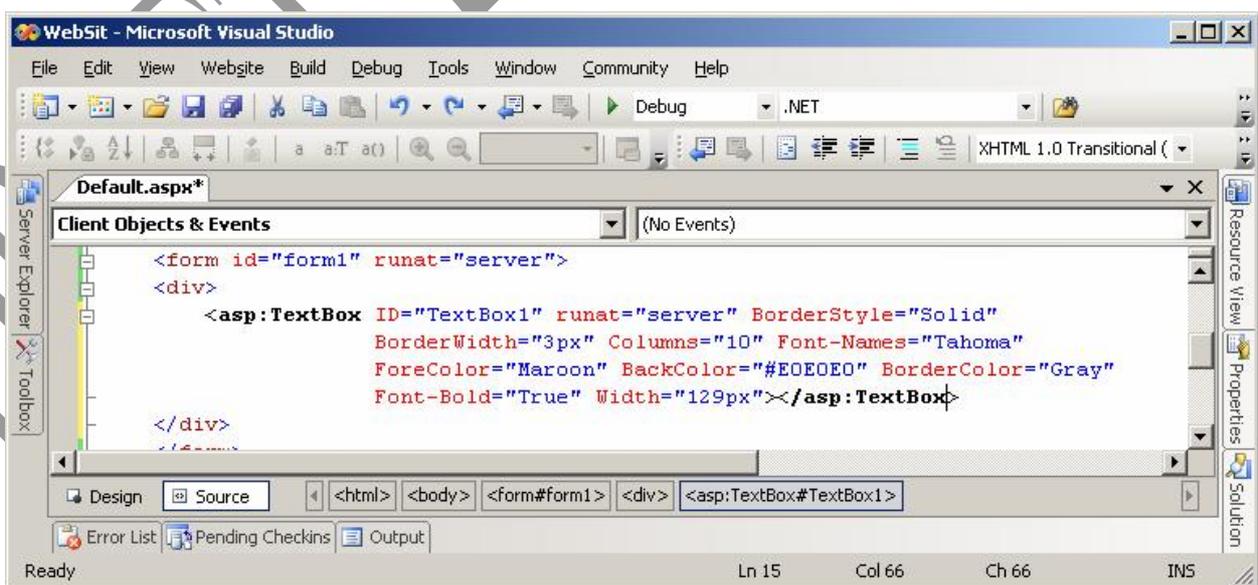
هذا توصيف لملف (Calendar.skf) و هذا الملف يحتوي على توصيف لعنصر التقويم

```
<asp:Label runat="server" Width="256px"
Font-Bold="True" Font-Names="Arial Narrow"
Font-Size="Medium" />
```

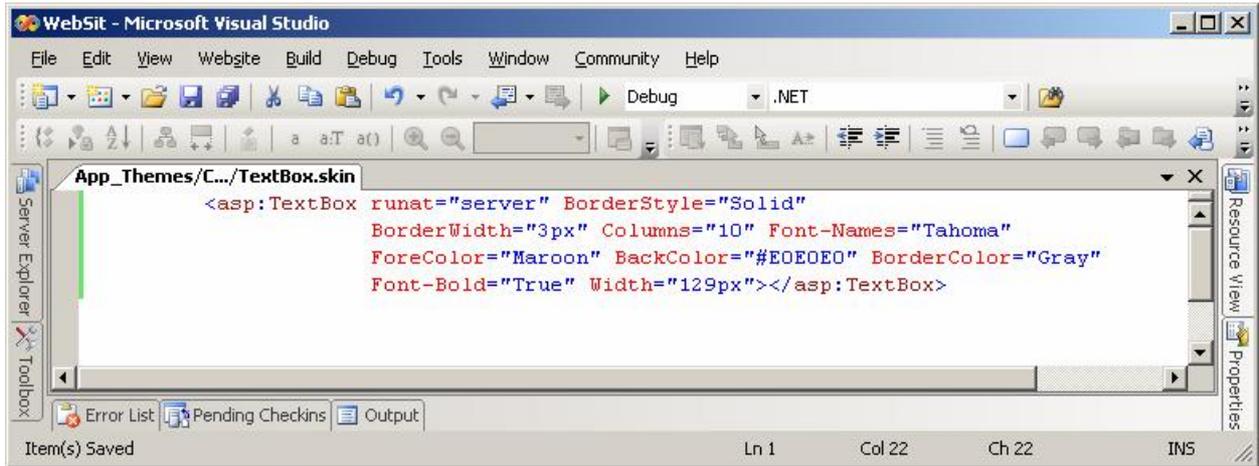
هذا توصيف لملف (Label.skf) و هو أيضاً ملف نرى أنه يحتوي على توصيف لأداة النص الثابت ، كما نرى أن العملية ليست معقدة و لا تحتاج إلى مجهود كبير، حيث أنك بكل بساطة من الممكن عمل ملف مشابه بدون كتابة حرف واحد وهو أن تقوم بالدخول على أى صفحة و تصميم ما تريده و ضبط الإعدادات كما تريد .



الشكل 32 [و الآن لنقم بعرض كيفية عمل نمط للعنصر (TextBox) ، قم بفتح صفحة جديدة و ضع عليها الأداة ، و قم بضبط إعدادات الشكل كما تريد و بعد ذلك أفتح الـ (Source View) - كما هو في الشكل التالي -]



الشكل 33 [قم بنسخ نص محتويات بيانات العنصر - كما هو موضح -]

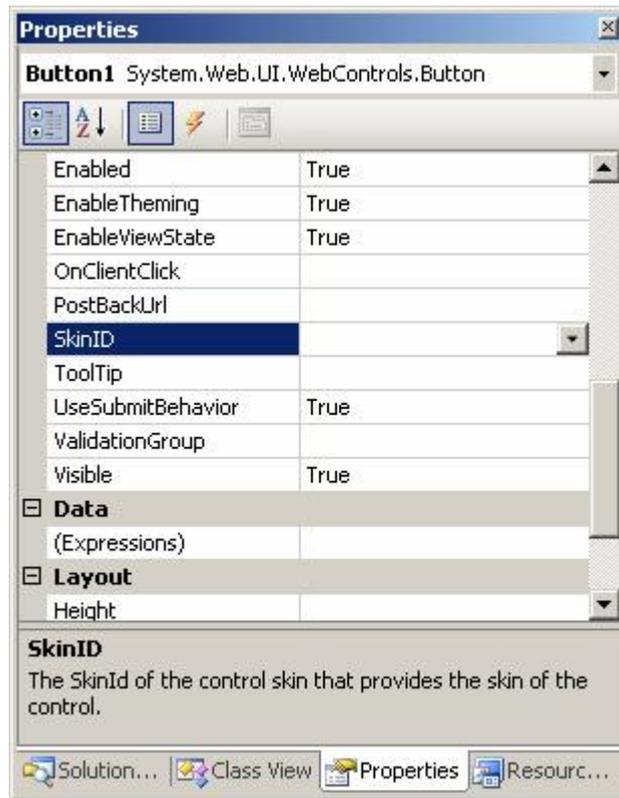


الشكل 34] قم بنسخ المحتويات في الملف (TextBox.skin) - كما هو موضح - مع الأخذ في الاعتبار أننا حذفنا المعرف (ID)



الشكل 36] وكما نرى أن النمط قد تم تطبيقه على العناصر التي لم نختار لها أي خصائص واجهة الاستعمال ، و هذا يرجع لقيامنا بتطبيق النمط كلاسيك و الذي يحتوي على ملف (TextBox.Skn) و هذا الملف يؤثر في جميع الـ (TextBox) الموجودة على الصفحة المطبق عليها النمط

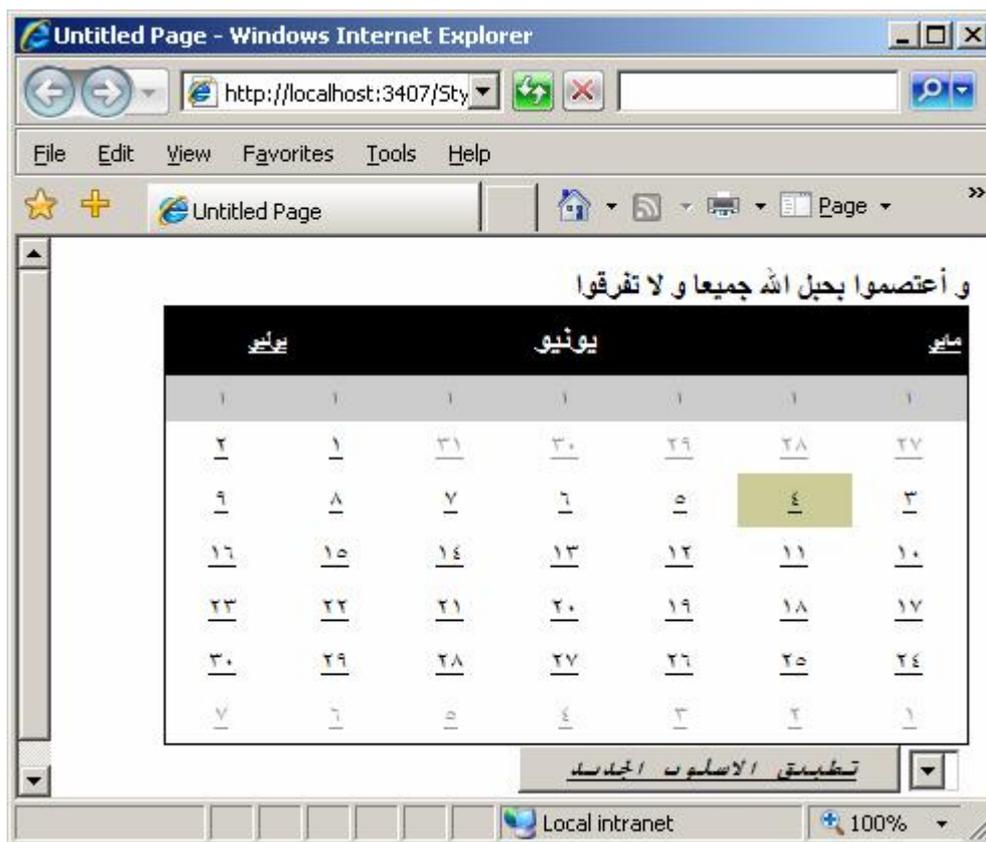
الشكل 35] بعد أن نكون ضبطنا إعدادات الأنماط في الصفحة نضع بعض الـ (TextBox) ، و كما نرى أنه لا يوجد أي إعدادات للشكل [



الشكل 37 [ميزة أخرى ممتازة و هي أنه يمكن للملف الواحد أن يحتوي على أكثر من نمط فرعي كما هو الحال في العنصر (Button)]



الشكل 38 [و هي الأنماط الفرعية الموجودة في النمط الأساسي (Classic)]



الشكل 39 [هو شكل الصفحة بعد تطبيق النمط (Classic) ، و تطبيق النمط الفرعي (Underline) للعنصر (Button)]

- § و بعد تعرفنا على كيفية عمل الأنماط ؛ نقوم – الآن - بتعلم شئ جديد و هو التعامل مع الأنماط برمجيا .
- § و المثال التالي هو مثال بسيط لكيفية تغيير النمط برمجيا و هو تكمله على المثال السابق .

```
protected void Page_PreInit(object sender, EventArgs e)
{
    حدث تحضير لتحميل الصفحة

    if (Request["DropDownList1"] != null)
    {
        التأكد من أنه هناك قيمة تعود من العنصر (DropDownList1) و التي تحتوى على أسماء
        الأنماط المتوافرة
        Page.Theme = Request["DropDownList1"];
        في حالة وجود قيمة في المتغير نقوم بضبط نمط الصفحة على قيمة العنصر
    }
}

protected void Page_Load(object sender, EventArgs e)
{
    حدث بداية التحميل

    if (!IsPostBack)
    {
```

التأكد أن هذا ليس إعادة طلب لصفحة معروضة أصلا (PostBack)

```
string[] strPath=Directory.GetDirectories(  
    Request.PhysicalApplicationPath + "app_themes");  
عرض كل المجلدات الموجودة في المجلد الفرعي (app_theme) و الموجود في مجلد المشروع  
foreach (string strTemp in strPath)  
دورة (Loop) تمر على محتويات المصفوفة (strPath) و تضع قيمة كل عنصر على حد في المتغير  
    (strTemp)  
    {  
DropDownList1.Items.Add(strTemp.Substring(strTemp.LastIndexOf('\\') + 1));  
        إضافة اسم المجلد و ذلك بأخذ آخر مقطع بعد الـ ( \ )  
    }  
}  
}
```

```
Protected Sub Page_PreInit(ByVal sender As Object, ByVal e As EventArgs)  
حدث تحضير لتحميل الصفحة  
If (Not (Request("DropDownList1")) Is Nothing) Then  
التأكد من أنه هناك قيمة تعود من العنصر (DropDownList1) و التي تحتوى على أسماء  
الأنماط المتوافرة  
    Page.Theme = Request("DropDownList1")  
في حالة وجود قيمة في المتغير نقوم بضبط نمط الصفحة على قيمة العنصر  
End If  
End Sub  
  
Protected Sub Page_Load(ByVal sender As Object, _  
    ByVal e As System.EventArgs) _  
    Handles Me.Load  
حدث بداية التحميل  
If Not IsPostBack Then  
التأكد أن هذا ليس إعادة طلب لصفحة معروضة أصلا (PostBack)  
    Dim themes As String() = _  
    Directory.GetDirectories( _  
    Request.PhysicalApplicationPath & "App_Themes")  
عرض كل المجلدات الموجودة في المجلد الفرعي (app_theme) و الموجود في مجلد المشروع  
    ddlThemes.Items.Clear()  
    For Each theme As String In themes  
دورة (Loop) تمر على محتويات المصفوفة (strPath) و تضع قيمة كل عنصر على حد في المتغير  
        (strTemp)
```

```
ddlThemes.Items.Add( _  
    theme.Substring(theme.LastIndexOf("\") + 1))  
    إضافة أسم المجلد و ذلك بأخذ آخر مقطع بعد الـ ( \ )
```

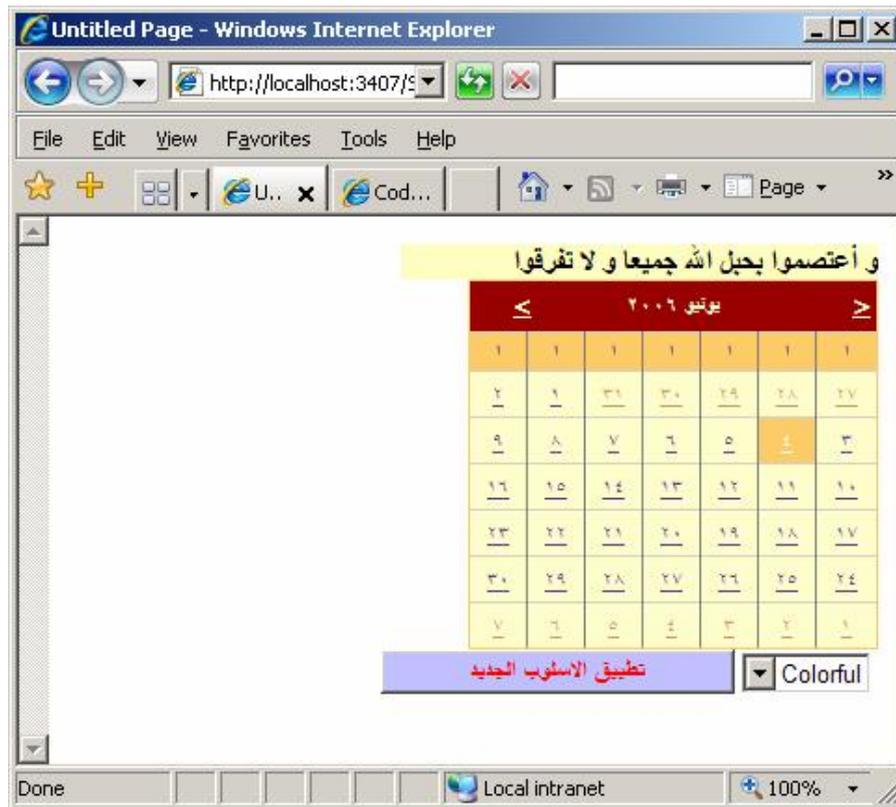
```
Next
```

```
End If
```

```
End Sub
```

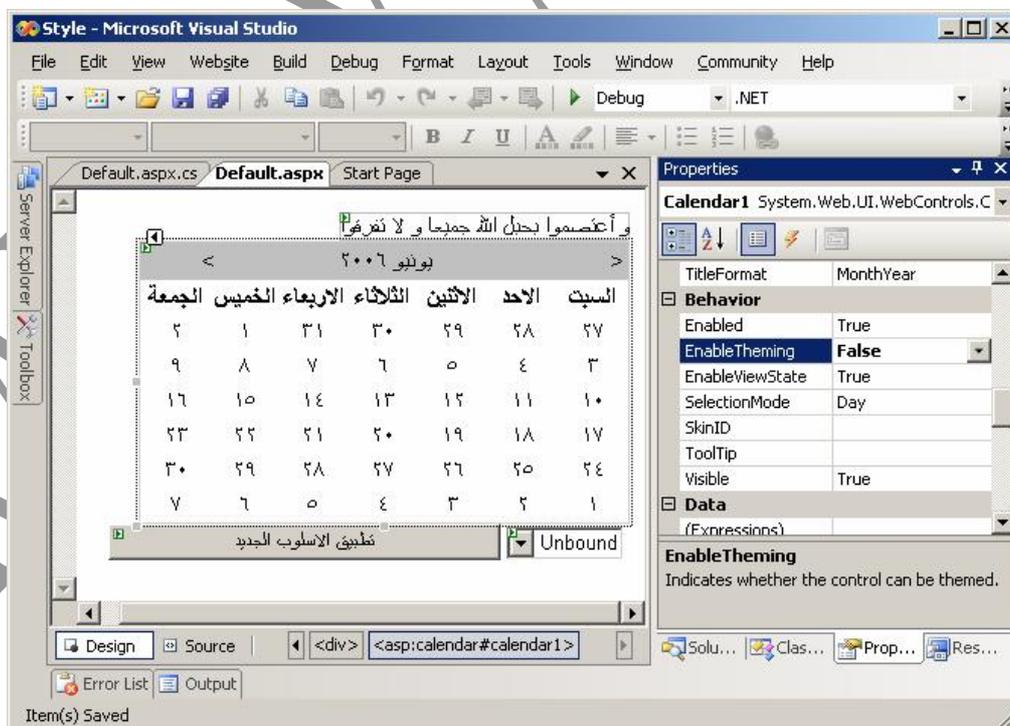


الشكل 40] هذا هو شكل المثال بعد أن قمنا بإضافة بعض الكود ، كل ما عليك هو أن تختار النمط ثم الضغط على الزر و سوف يلتقط الحدث التغيير و يقوم بتطبيق النمط الذي قمت باختياره [



الشكل 41 [هذا هو الشكل بعد اختيار النمط (Colorful)]

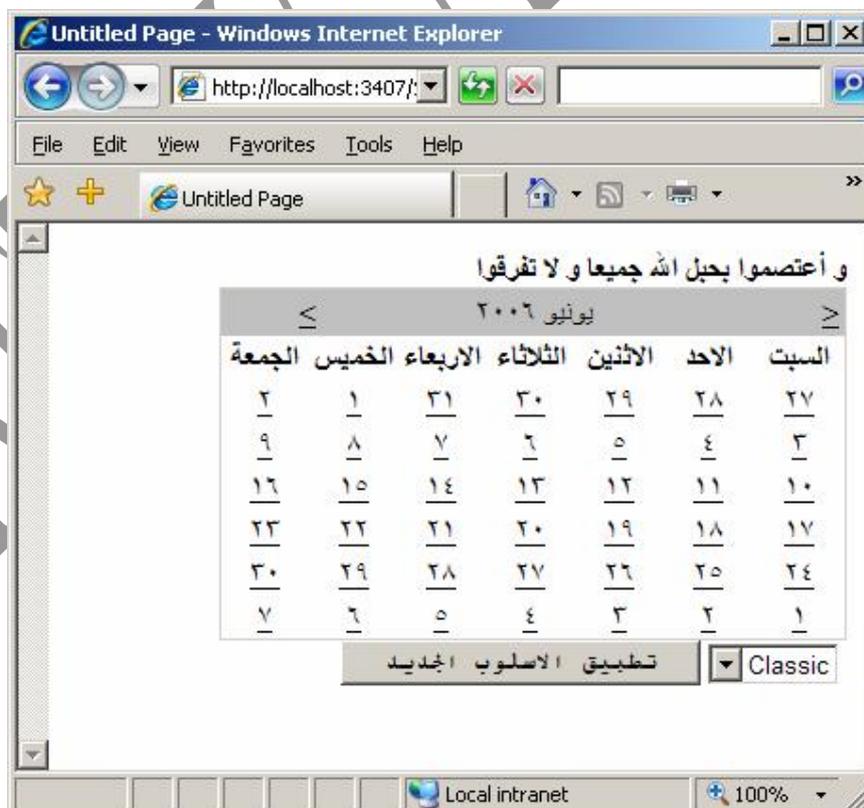
- آخر شئ أود ذكره- و من الممكن أن يكون مفيد- وهو أنه ماذا لو أردت أن توقف تطبيق النمط على بعض العناصر!



الشكل 42 [نقوم باختيار عنصر التقويم ، و نقوم بإيقاف الخاصية (EnableTheming) و هذه الخاصية مسنولة عن السماح بتطبيق النمط على العنصر أم لا]



الشكل ٤٣ [هذا هو شكل الصفحة قبل أن نوقف تطبيق النمط على عنصر التقويم ، و نرى أن النمط طبق على العنصر]



الشكل ٤٤ [هذا هو شكل الصفحة بعد إيقاف الخاصية (EnableTheming) ، و كما نرى أن النمط طبق على جميع العناصر ما عدا التقويم]

§ و الآن و بعد الانتهاء من هذا الفصل هناك بعض الأفكار التي سأعرضها عليكم - فمن الممكن أن تكون مفيدة - للقيام بتطبيقها ، وهي - على سبيل المثال - دمج إمكانية تغيير الأنماط مع الملفات الشخصية و من الممكن أيضا أن تقوم بعمل تطبيق صغير يقوم بصنع ملفات الأنماط لكي يقوم المستخدم بتصميم الألوان التي يحبها مما يجعله مرتبط أكثر بتطبيقك و هناك الكثير من الأفكار و الحيل التي تفيد كثيراً في هذا الموضوع .

DRAFT BY Ahmed Essam