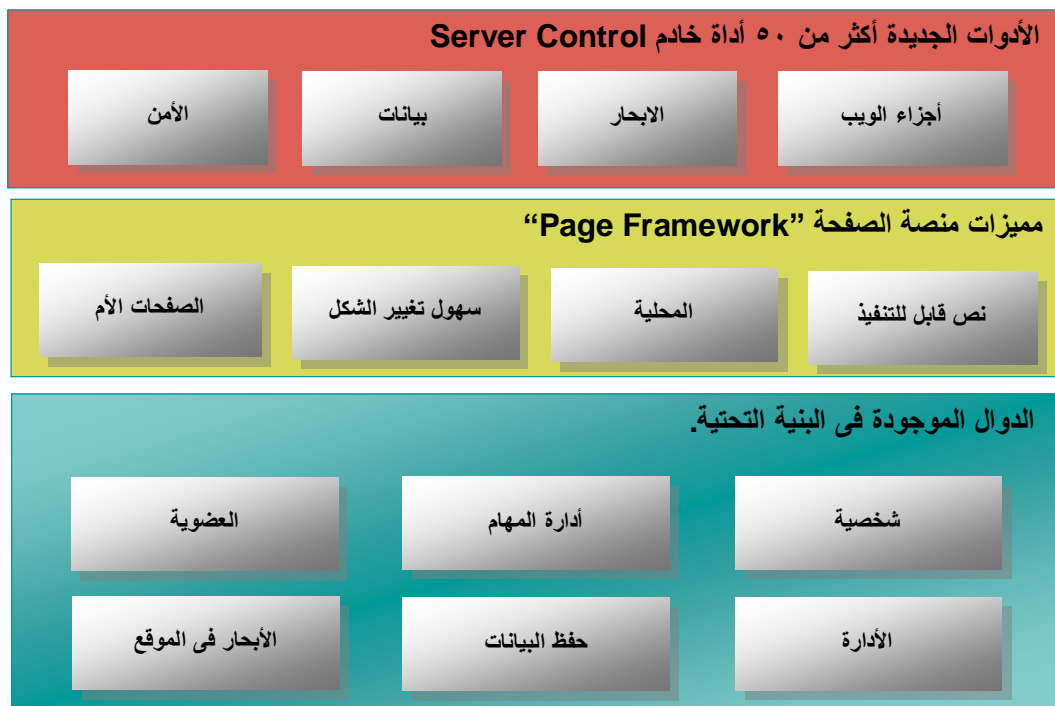


الفصل الأول : المميزات الجديدة فى فيجوال ستوديو ٢٠٠٥.

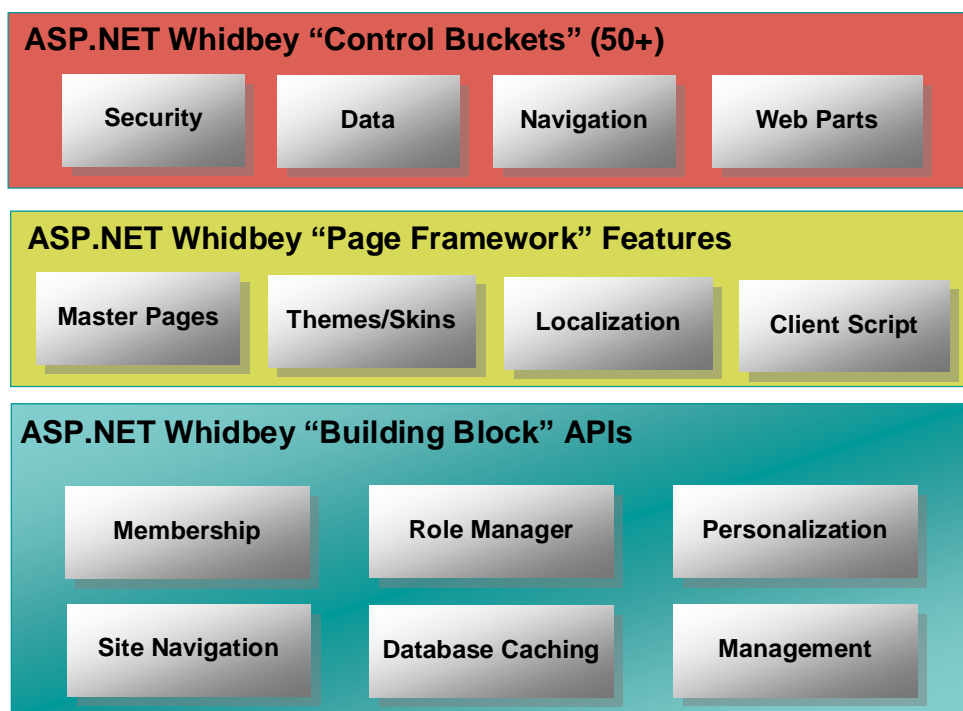
في البداية أود عرض رسم تعبيري بسيط يوضح ما هو الجديد بشكل عام في الأصدار الجديد من VS.net 2005

الجديد في ASP.net 2.0



الشكل ١

ASP.NET 2.0 Developer Stack



الشكل ٢

هذه المميزات الجديدة و المكتبات و الأدوات سوف توفر أكثر من ٣٠% من وقت المطور مما سيزيد تركيز المطور في إنشاء حلول أقوى و أفضل، أريد أن أوضح شئ صغير لفهم وهو يعتبر أساس كل جديد في الأصدار ٢٠٠٥ وهو أن خلف كل وحدة من هذه الوحدات الموجودة في الشكل ١ و ٢ ما يسمى بمكتبة المخصص **Provider Class** و التي تقوم بعمل كل ما كنت تقوم به أنت كمطور وسوف تأتي أمثلة لشرح هذا الموضوع بالتفصيل بالإضافة الى أنه سوف يكون هناك فصل كامل يتحدث عن **Provider Class** كيفية عملها بالإضافة الى كيفية أنشائها.

إنشاء مشروع جديد في الـ VS.net 2005

في الأصدارات السابقة لكي تقوم ببدأ مشروع جديد كان عليك أن تقوم بـ :

File à New à Solution

و بعد ذلك تظهر نافذة التي منها تختار لغة المشروع ثم نوع البرنامج الذي تود إنتاجه.

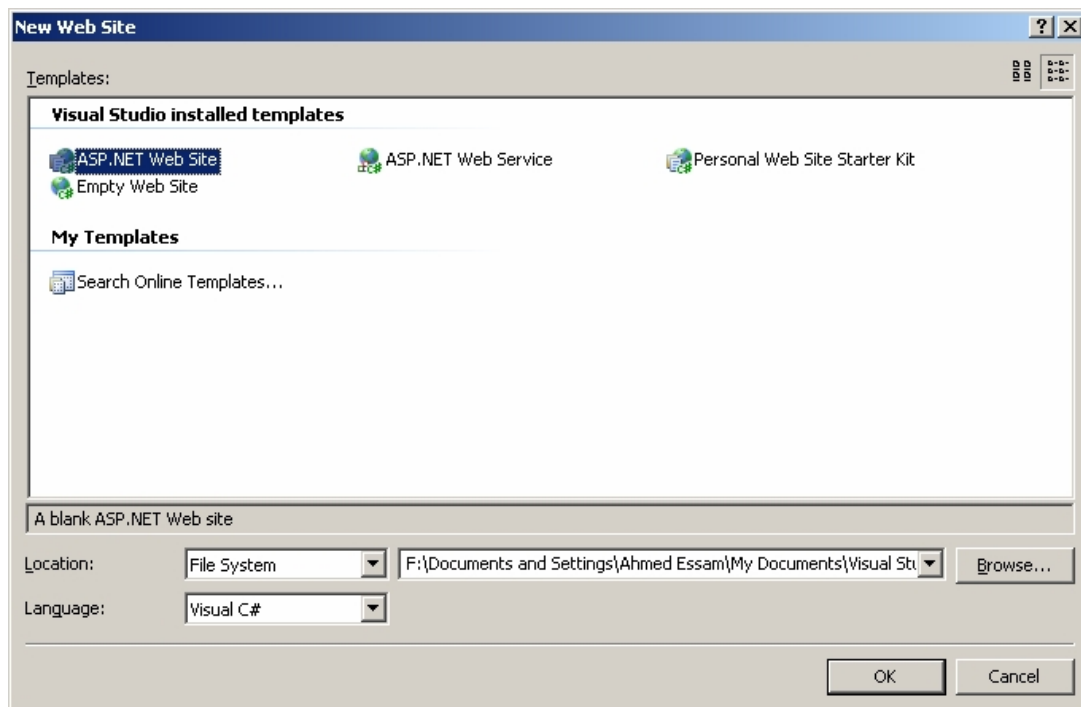
الآن في فيجوال ستوديو ٢٠٠٥ الوضع مختلف و من وجهة نظري فهو للأحسن حيث أن كل ما عليك الآن هو

File à New à Web site



الشكل 3

و بعد ذلك تقوم باختيار نوع المشروع و لغته و طريقة الحفظ.



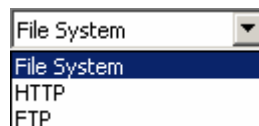
الشكل 4

الآن لنشرح ما أمامنا في هذه الشاشة.

في القسم العلوي نجد ما يسمى بالـ **Templates** (قوالب) و من هنا تختار نوع برنامج الويب سواء كان برنامج ويب عادي "موقع" أو كان خدمات "web services" أما الاختيار الثالث فهو موجود في حالة أنك تود أن ترى مثال حي و الرابع هو مشروع ويب فارغ يمكنك توجيهه حسب ما تريد.

في جزء الموقع **Location**

هناك جزء مهم جدا لا بد من توضيحه وهو الجزء الأول من **Location** و الذي يتكون من قائمة منزلة و محتوياتها كالآتي



الشكل 5

و الآن لنشرحهم بالتفصيل

• أولاً : File System

○ في حالة أنك تقوم بتجربة الـ **VS.net** فهذه هو أنسب اختيار لك حيث لا يتطلب منك وجود خادم **IIS**

internet information server – هنا يمكنك أستنتاج شئ آخر وهو أنه من الممكن الآن

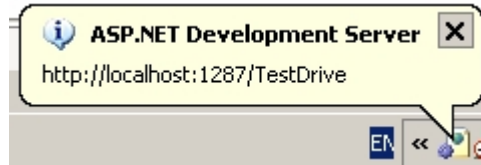
لمستخدمي **Windows XP home Edition** العمل بـ **VS.net 2005** حيث كان هناك مشكلة

في مشاريع الويب أنه لا بد من توافر خادم **IIS** وكان هذا الخادم غير موجود في نسخة المنزلية من

ويندوز **XP** . رائع جدا و لكن كيف يعمل مشروع الويب "لا بد من وجود خادم؟؟؟" بكل بساطة الآن في

الأصدار الجديد من فيجوال ستوديو يوجد خادم (**Server**) صغير يقوم بهذه المهمة كما هو موضح

في الشكل ٤ و ٥.



الشكل 6



الشكل 7

• ثانياً : HTTP

○ Local هذا الخيار جيد في حالة وجود خادم محلي Local IIS وهذا سوف يتيح للمستخدمين الموجودين في الشبكة الاطلاع على مشاريعك

• ثالثاً : FTP

○ هذا الخيار جيد إذا ما كنت تود التعامل مباشرة مع مستضيف الموقع Host حيث سوف يمكنك من خلاله التعامل مباشر و الحي مع مشروعك في بيئة عمل فعلية و حيه.

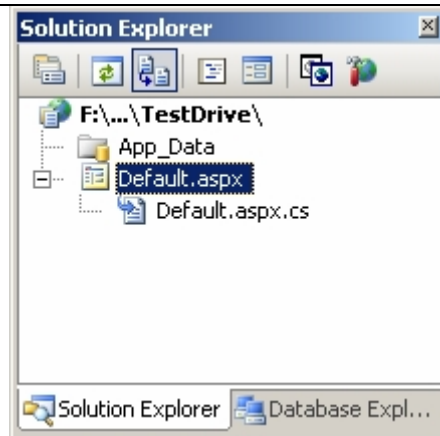
بعد أن تم الجزء الخاص بشرح موقع و طريقة الحفظ و التعامل مع الملفات الآن ننتقل الى اختيار اللغة و المقصود هنا هو اختيار اللغة التي سوف تنفذ بها مشروعك و هذا طبعا خيار بسيط و يرجع اليك و حسب أهتمامك كمطور برامج و ويب.

تم تغيير CodeFile الى CodeBehind

CodeFile="Default.aspx.cs"

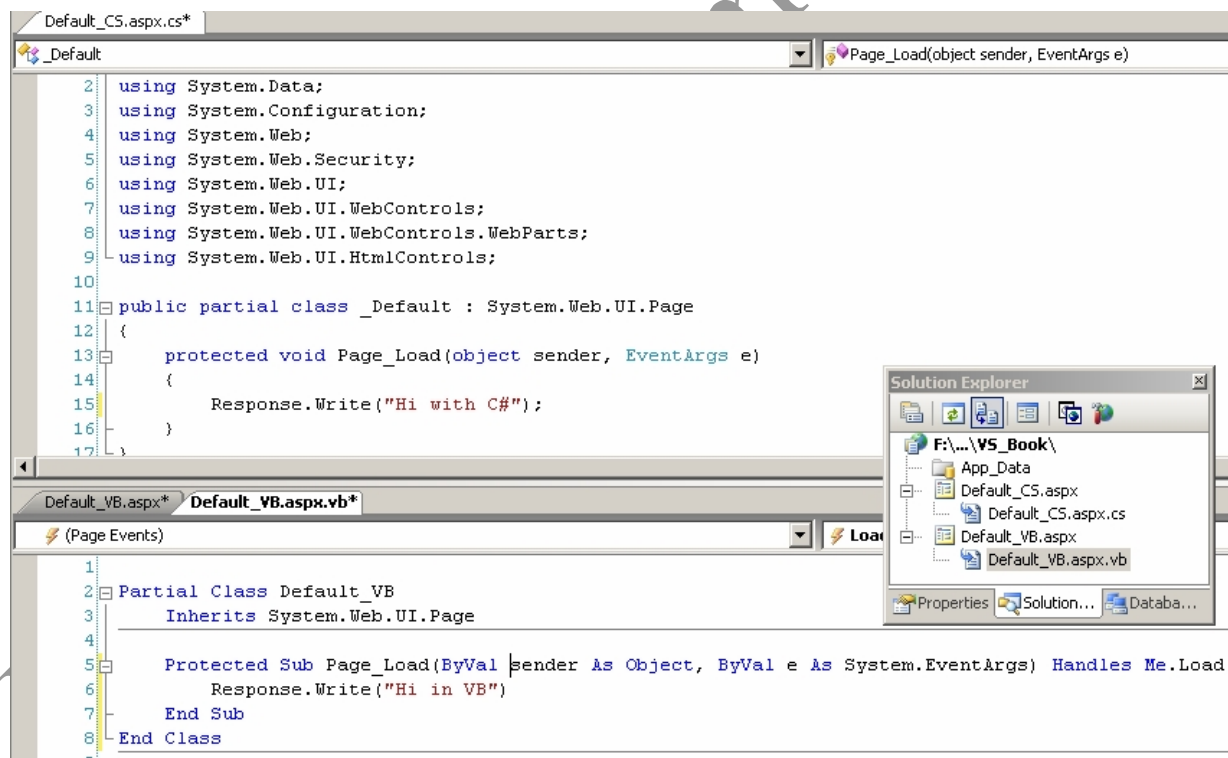
الشكل ٨

عند إنشاء المشروع لأول مرة سوف يقوم VS.net بإنشاء مجلد APP_Data و ملف Default.aspx و
ملف الكود الكود Default.aspx.cs أو Default.aspx.vb



الشكل 9

الآن الى أكثر الخواص الجديدة أشارة و هي تعدد اللغات في مشروع واحد
في الأصدارات السابقة من فيجوال ستديو كان هذا الموضوع معقد الى حد منا حيث كان عليك أن تقوم ببعض
الحيل لكي تقوم بدمج أكثر من لغة في مشروع واحد الآن في الأصدار الجديد من فيجوال ستوديو دوت نت الأمر أبسط ما
يكون لدجة أنك لن تشعر بأن هناك فرق أو أن هناك أكثر من لغة في المشروع الواحد



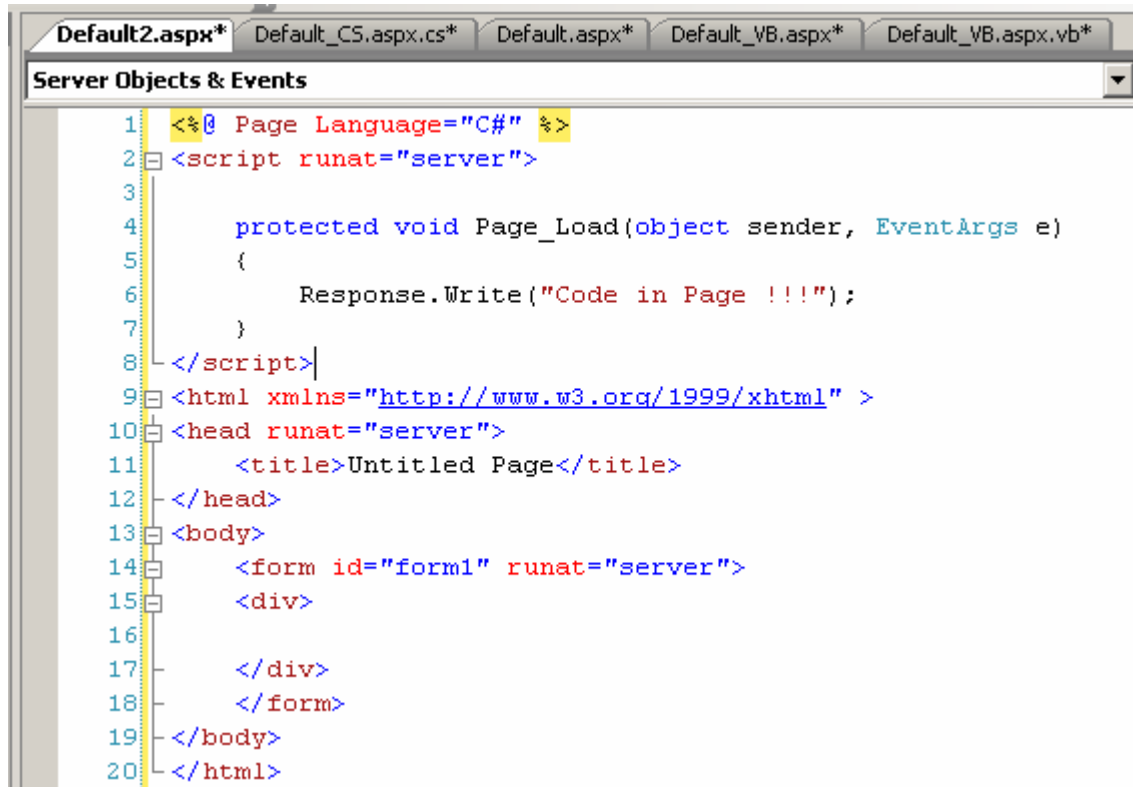
الشكل ١٠

هذه الخاصية لها ميزة و عيب في نفس الوقت



الميزة هي أنه ليس على كل أفراد الفريق أن يستعملوا نفس اللغة. مما يسهل العمل في المشروعات الكبيرة.
العيب هو أنه في المشروعات كبيرة سوف يكون هناك مشكلة وهي أنه في حالة الصيانة سوف يكون على من
يقوم بعملية الصيانة يتقن اللغات المستخدمة في المشروع و الذي من الممكن أن يكون غير متاح.

خاصية أخرى جديدة و مفيدة و هي إمكانية دمج الصفحة و ملف الكود



الشكل ١١

أهمية هذه الخاصية تكمن في أنه في بعض الأحيان لا يحتاج المطور أن يكون هناك ملف كامل للكود نظراً لأنه لن يكون هناك الكثير من وحدات الكود **Code Blocks** و هذا يظهر على سبيل المثال في حالة صفحات الـ **Log and redirect** و التي تستخدم لحفظ بيانات تحرك المستخدم داخل التطبيق لأغراض متابعة التطبيق و تحسينه.

من إحدى الخواص المهمة و المفيدة جيد هي خاصية الترجمة التلقائية لملفات الكود **AutoCompilation**

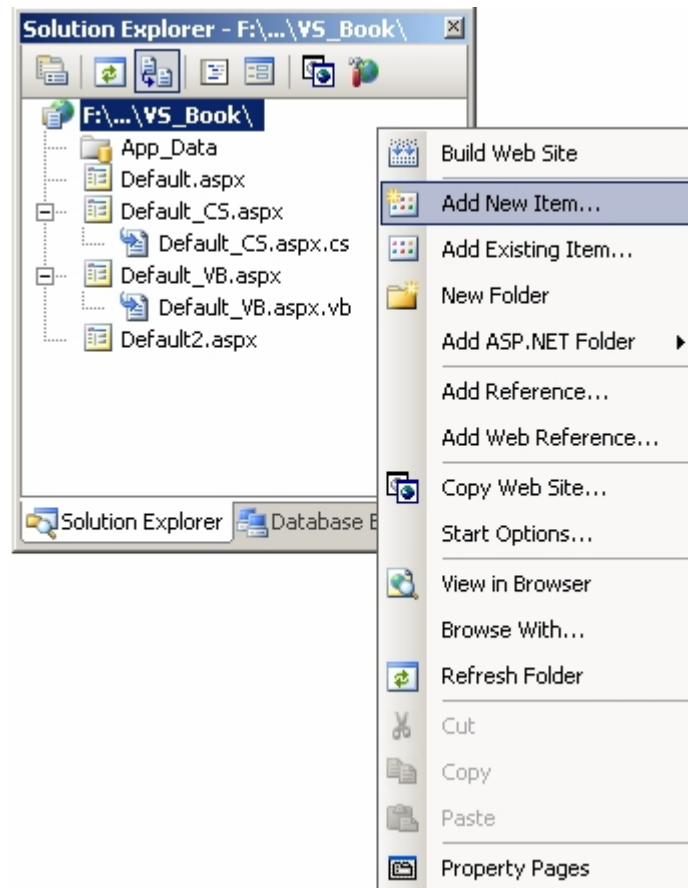
حيث أنه بمجرد حفظ الملف يقوم المترجم بأعادة ترجمة ملف الكود دون أنهاء أى من جلسة **Session Object** أو التطبيق **Application Object** هذه الخاصية مفيدة للمطورين حيث توفر الوقت الذى كان يستهلك في إعادة الخطوات التى تقوم بأعداد كلا من الـ **Sessions** و الـ **Application objects**.

كثير من الخواص المتعلقة بالكود و إعادة الترجمة ترجع الى سبب فنى مهم جداً لابد أن يعرفه المطور وهو أن كل صفحة الآن يتم ترجمتها على حدى بمعنى أنه إذا قمت بتعديل ما فى صفحة معينة من صفحات المشروع سوف يقوم المترجم التلقائي بترجمة الصفحة التى تم التعديل فيها فقط، يمكنك الاستفادة من هذه الخاصية فى المشاريع الضخمة حيث فى حالة الصيانة من المستحيل إيقاف الخدمة بالإضافة الى أنه فى الإصدارات القديمة عند إعادة ترجمة مشروع بشكل ما يؤدي الى فقد كلا من الـ **Session** و الـ **Application** مما يسبب بعض المشاكل فى بعض الأحيان. إذا فكرنا قليلاً سوف نجد الكثير من الفوائد التى تأتى من وراء خاصية كهذه كل ما نحتاجه هو بعض الوقت لكى تظهر "خبرة التعامل" مع هذه الخاصية الجديدة و التى سوف تسهل الكثير.



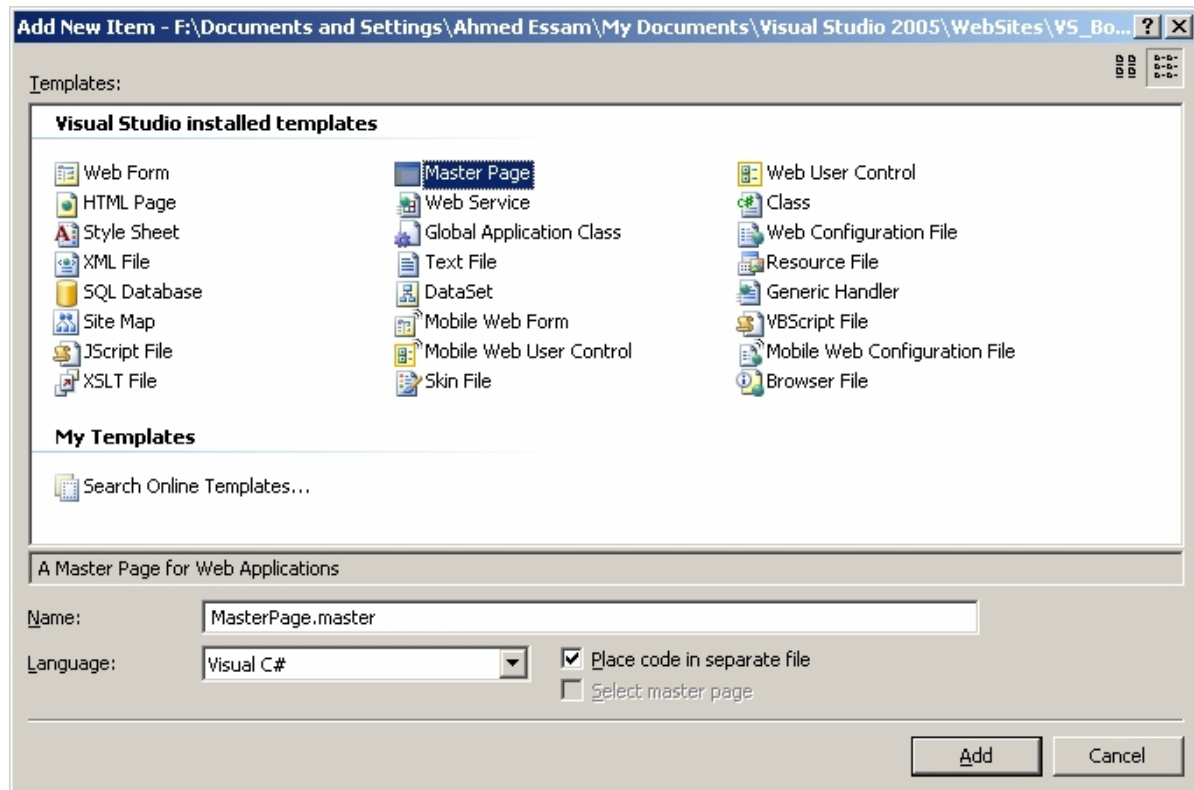
هناك ميزة جديدة و من أهم المميزات التي سوف توفر الكثير من العناء على كلا من مصمم الجرافيك للمشروع و مطورة
وهي خاصية وراثثة من الصفحة الأم !!! الأسم يبدو غريب و لكن لا تقلق سوف يتم شرح هذا الموضوع بالتفصيل وسوف
يظهر هذه الخاصية فى حالات معينة أولاً و لكي تقوم بتجربة بسيطة لهذه الميزة الجديدة نتبع الأتى:

من نافذة Solution Explorer قم بالضغط بالزر الأيمن للماوس و أختار Add new item



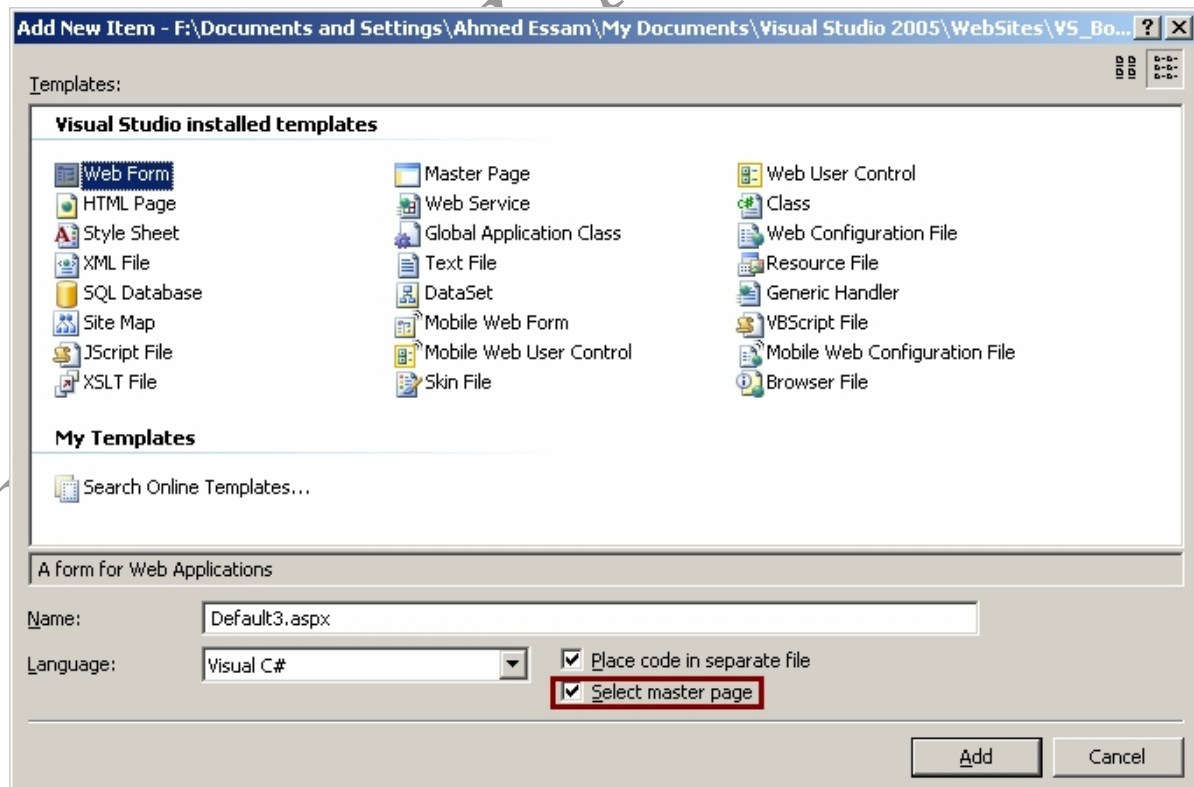
الشكل ١٢

و بعد ذلك نختار من نافذة Add New Item فى القسم Templates نقوم بأختيار Master Page



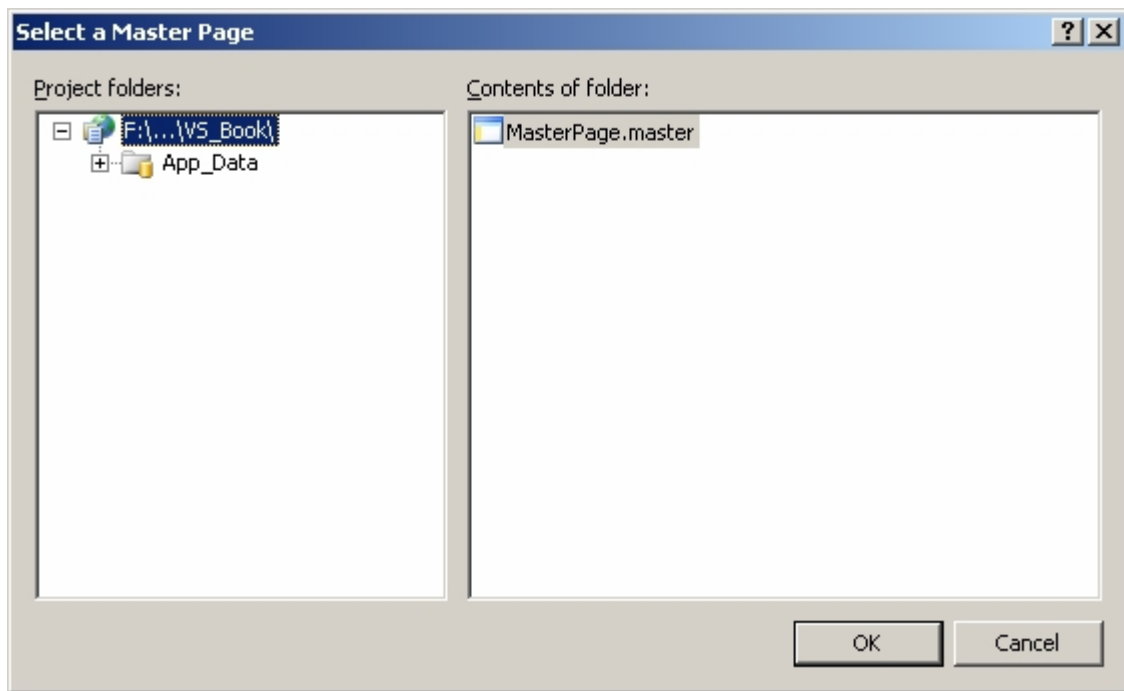
الشكل ١٣

نقوم بإضافة عنصر جديد بنفس الطريقة السابقة و لكن بدلا من اختيار **Master Page** سوف نختار **Web Form** و سوف نقوم بتفعيل الخيار **Master Page**



الشكل 14

بعد ذلك سوف يظهر شاشة **Select Master Page** و التي سوف تختار منها الصفحة الأم التي تود أن ترث منها



الشكل ١٥

بهذا نكون قد قمنا بعمل نوع ما من ربط الصفحة الأم بالصفحة الفعلية كما ذكرت أنه هناك فصل كامل يتحدث عن هذا الموضوع نظرا لكثرة التفاصيل به.

المكتبة الجزئية **Partial Class**:

خاصية جديدة و في نظري سوف توفر الكثير من العناء التي كان يعينها المطورين في المشاريع الضخمة. والآن لنقوم بتوضيح الأمر لكي يتضح لكم مدى أهمية هذه الميزة. في البداية عن رؤية صفحة الكود التي ينتجها **VS.net 2005** سوف تجدها مختلفة تماما عما سبق !!! السبب يرجع الى أنه تم أخفاء كل الكود الخاصة بأنشاء المكونات على الصفحة و هذا لانه لم يتواجد أهمية حقيقية لأظهار هذا الكود، عند محاولة الدخول لكتابة كود أى حدث نلاحظ الأتي

```
1 using System;
2 using System.Data;
3 using System.Configuration;
4 using System.Web;
5 using System.Web.Security;
6 using System.Web.UI;
7 using System.Web.UI.WebControls;
8 using System.Web.UI.WebControls.WebParts;
9 using System.Web.UI.HtmlControls;
10
11 public partial class _Default : System.Web.UI.Page
12 {
13     protected void Page_Load(object sender, EventArgs e)
14     {
15     }
16 }
17
```

الشكل ١٦

```

1 Partial Class Default_VB
2     Inherits System.Web.UI.Page
3
4     Protected Sub Page_Load(ByVal sender As Object _
5         , ByVal e As System.EventArgs) Handles Me.Load
6
7     End Sub
8 End Class

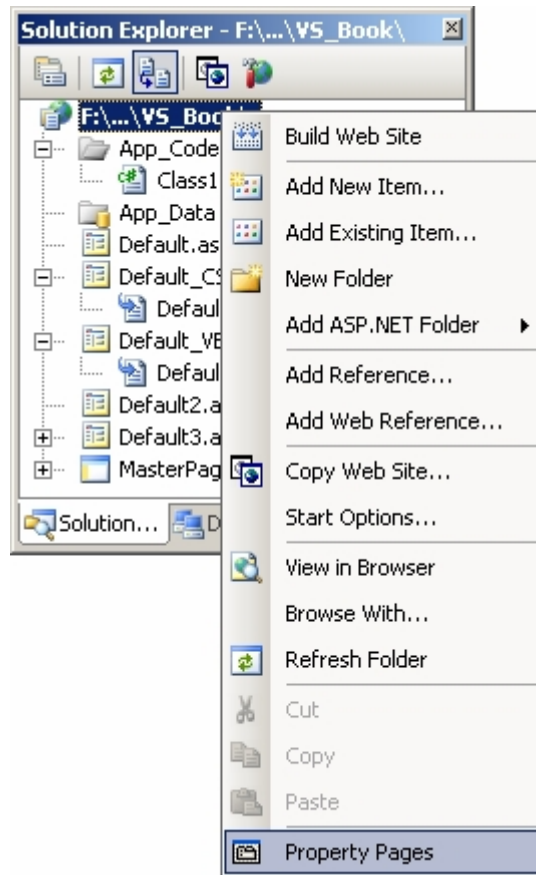
```

الشكل ١٧

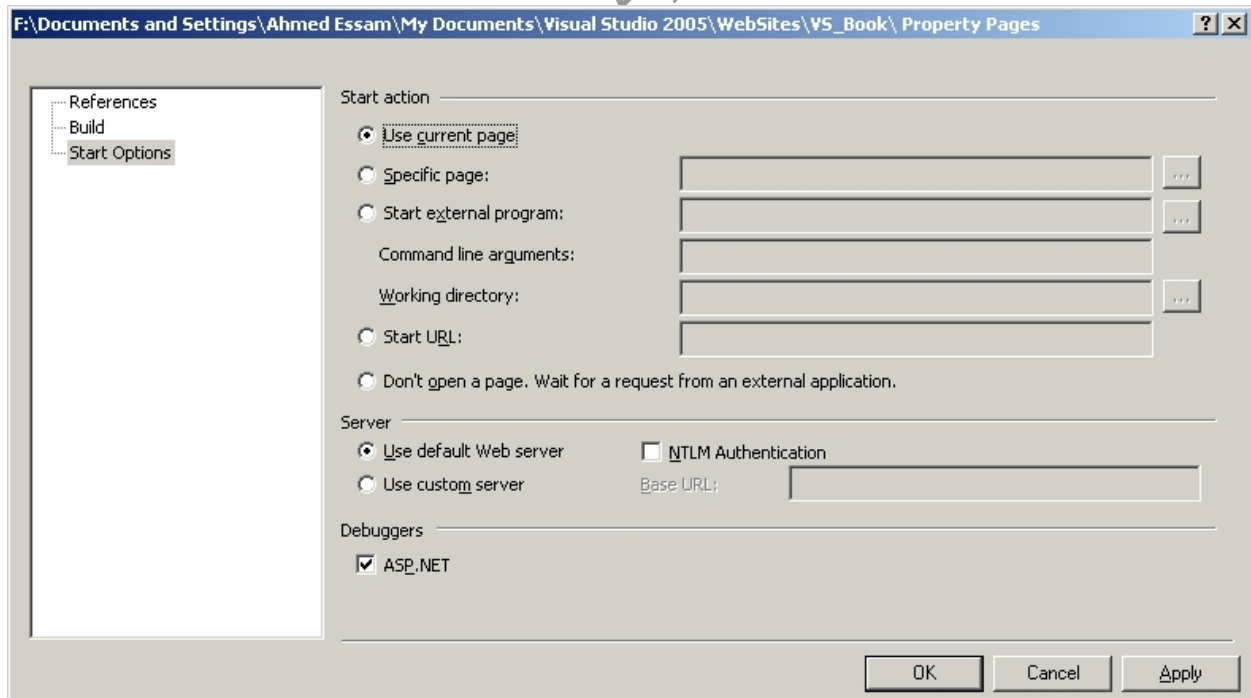
كما تلاحظ أنه تم إضافة الكلمة **Partial** الى معرف المكتبة، المعنى المقصود هنا هو أنه هذه المكتبة فى الواقع جزء من المكتبة و ليس المكتبة كلها مما يتيح عمل أكثر من مطور فى مكتبة واحدة دون حدوث المشاكل معهودة عند دمج الأكواد، فى الواقع يقوم مترجم الـ **VS.net 2005** بتجميع المكتبة و ترجمتها بالإضافة الى أن الـ **VS.net 2005** يقوم بدمج الكود المخفى الخاص بانشاء المكونات على الصفحة، هذه الميزة سوف تعطي مزيد من التحكم لرئيس مجموعة التطوير من ناحية التحكم فى كود **Source Control**.

تحسين التنقيح فى الأصدار الجديد Enhancing Debugging

فى الأصدارات السابقة من فيجوال ستديو دوت نت كان عليك بشكل واضح تحديد صفحة البدء **Set start page** التى سوف تظهر عند بدأ تنقيح التطبيق غالباً عند الضغط على **F5** (ذلك حسب أعدادات خارطة لوحة المفاتيح). فى الأصدار الجديد مازال بإمكانك تحديد صفحة البدء بالإضافة الى أنه من الممكن الصفحة الحالية هى صفحة البدء – المقصود بالصفحة الحالية هى الصفحة التى تعمل فيها- بالإضافة الى إمكانية جعل الصفحة المفعلة فى **Solution Explorer** هى صفحة البدء، هذه الميزة أهميتها تكمن فى توفير العناء على المطور عند اختبار صفحة معينة يعمل عليها حيث كان فى السابق موضوع شاقاً فى حالة المشاريع الكبيرة، كما أنه كانت تسبب مشاكل فى حالة عدم تحديد الصفحة فعلياً للمترجم أما الآن فبمجرد اختيارك للصفحة من **Solution Explorer** تكون قد اخترت صفحة البدء كل هذا يمكن تعديله و ضبطه حسب ما تريد من خلال صفحة الخصائص.



الشكل ١٨



الشكل ١٩

كما ترى الخيار يمكنك اختيار منها ما يناسبك و يلائم متطلباتك.

ضبط تركيز العناصر Focus of Controls.

أولا لابد من توضيح معنى هذه الجملة حيث أنها تبدوا للوهلة الأولى غير مفهومه و الحل لتوضيح هذه الجملة هو مثل حي أظن أن جميعا يستخدم محرك البحث جوجل هل لاحظ أحدكم أنه عند أنتهاء تحميل الصفحة يتم نقل المؤشر الى حقل نص البحث، هذا هو المقصود بضبط تركيز العناصر، في الإصدارات السابقة من فيجوال ستوديو دوت نت كان لعمل شئ مشابه كان هذا يتطلب كتابة كود على العارض "جافا سكربت" الآن في الإصدار الجديد في فيجوال ستوديو دوت نت أصبح الأمر في غاية البساطة كما هو موضح في الشكل التالي

```
protected void Page_Load(object sender, EventArgs e)
{
    Page.SetFocus(txtTry);
}
```

الكود بصيغة C#

```
Protected Sub Page_Load(ByVal sender As Object _
    , ByVal e As System.EventArgs) Handles Me.Load
    Page.SetFocus(txtTry)
End Sub
```

الكود بصيغة Visual Basic

كل ما عليك للقيام بهذا العمل هو استدعاء الدالة **SetFocus** من المكتبة **Page** ، هذه هي الطريقة الأولى لتحديد التركيز عند التحميل هناك طريقة أخرى يتم ضبطها من خلال كود الـ **HTML**

```
<form id="form1" runat="server" defaultfocus=txtOrderID defaultbutton=tnGetOrderData>
    &nbsp;Order ID :
    <asp:TextBox ID=txtOrderID runat=server></asp:TextBox>
    <br />
    <asp:Button ID="btnGetOrderData" runat="server" Text="Get Order Data" />
    <asp:Button ID="btnCancel" runat="server" Text="Cancel" />
</form>
```

كما هو واضح من إحدى خواص الـ **Form** ما يسمى بالـ **DefaultFocus** وهذه الخاصية مسأولة عن ضبط التركيز و تخصيص أداة معينة عند التحميل، و لكن هناك أيضا ما يسمى الـ **DefalutButton** هذه الخاصية مفيدة جدا و التي كانت تسبب الكثير من المشاكل و الحيرة في نفس الوقت عندما يقوم المستخدم بملء الـ **Form** و بعد ذلك قام بالضغط على زر الإدخال **Enter** و لنفرض أن هناك أكثر من زر أى منهم سوف يتم تنفيذه هنا يتم بكل بساطة تحديد هذه النقطة و التي سوف تريحك من عناء كتابة **Client side code** لحل المشاكل المشابهة.

تعريف الكثير من مجموعات التحقق **Validation Groups**.

في الإصدارات السابقة كانت الصفحة و الحدة يتم التحقق منها مرة واحدة مما قد يسبب بعض المشاكل في الصفحات التي يوجد بها أكثر من خاصية و في نفس الوقت تحتاج الى تحقيق كلا منها على حدى، الآن في الإصدار الجديد أمكن أن يتم

فصل كل مجموعة من الأدوات في مجموعة تحقق خاصة، حتى الآن هذه الخاصية تبدو غير واضحة سوف نقوم بتوضيح هذه الخاصية بمثال صغير

```
<asp:Panel ID="Panel1" runat="server" Height="50px" Width="253px">
&nbsp;Order ID :
<asp:TextBox ID=txtOrderID runat=server></asp:TextBox>
<br />
<asp:Button ID="btnGetOrderData" runat="server" Text="Get Order Data"
ValidationGroup="grpOrder"/>&nbsp;
<br />
<asp:RequiredFieldValidator
ID="RequiredFieldValidator2"
runat="server"
ControlToValidate="txtOrderID"
ErrorMessage="RequiredFieldValidator"
ValidationGroup="grpOrder">Enter the Order ID
</asp:RequiredFieldValidator><br />

<asp:RegularExpressionValidator
ID="RegularExpressionValidator1"
runat="server"
ControlToValidate="txtOrderID"
ErrorMessage="RegularExpressionValidator"
ValidationExpression="\d{6}"
ValidationGroup="grpOrder">Please Enter the right Order ID format
</asp:RegularExpressionValidator></asp:Panel>
```

لا بد من أتبع بعض الخطوات للاستفادة من هذه الخاصية و جعلها تعمل بشكل صحيح، بكل بساطة يتم تعريف كل محقق Validator المجموعة التي ينتمي اليها ValidationGroup و بعد ذلك نقوم بضبط الخاصية لجعلها تساوى نفس القيمة فى الزر الذي من المفروض أن يقوم بمهمة معينة بعد التحقق من المدخلات، قمت بتمييز الخواص المتعلقة بهذا الموضوع فى كود الـ HTML المذكور أعلى هذا النص.

أضافة جافا سكربت :

فى بعض الأحيان نحتاج بشدة الى توافر هذه العملية أضافة كود جافا سكربت لتقوم ببعض الأعمال عند جهاز العميل فى السابق كانت موجودة و لكن فى الأصدار الجديد من فيجوال هى موجودة أيضا و لكن تم تخصيص مكتبة كاملة للتعامل مع الجافا سكربت و الآن سوف أعرض مثالين للتعامل مع الجافا سكربت

```
protected void Page_Load(object sender, EventArgs e)
{
    string strScript;
```

```
strScript = "window.alert(' Server time -> " + DateTime.Now.ToString() + "');";  
Page.ClientScript.RegisterClientScriptBlock(this.GetType(), "testScript", strScript, true);  
}
```

الكود بصيغة C#

```
Protected Sub Page_Load(ByVal sender As Object _  
    , ByVal e As System.EventArgs) Handles Me.Load  
Dim strScript As String  
strScript = "window.alert(' Server time -> " & Now & "');"  
Page.ClientScript.RegisterClientScriptBlock(Me.GetType, "testScript", _  
    strScript, True)  
End Sub
```

الكود بصيغة Visual Basic

كما نرى أن في الأصدار الجديد وجود مكتبة كود الجافا سكربت التي تسمى **ClientScript** و التي تتيح لك التعامل مع كود الجافا سكربت و الآن سوف أعرض مثال آخر و الي سوف أعرض فيه كيفية تضمين ملف كود جافا سكربت **js file**

```
protected void Page_Load(object sender, EventArgs e)  
{  
    string strScript;  
    strScript = ".\\code.js";  
    Page.ClientScript.RegisterClientScriptInclude("testFileCode",strScript);  
}
```

الكود بصيغة C#

```
Protected Sub Page_Load(ByVal sender As Object _  
    , ByVal e As System.EventArgs) Handles Me.Load  
Dim strScript As String  
strScript = ".\\code.js"  
Page.ClientScript.RegisterClientScriptInclude("testFileCode", strScript)  
End Sub
```

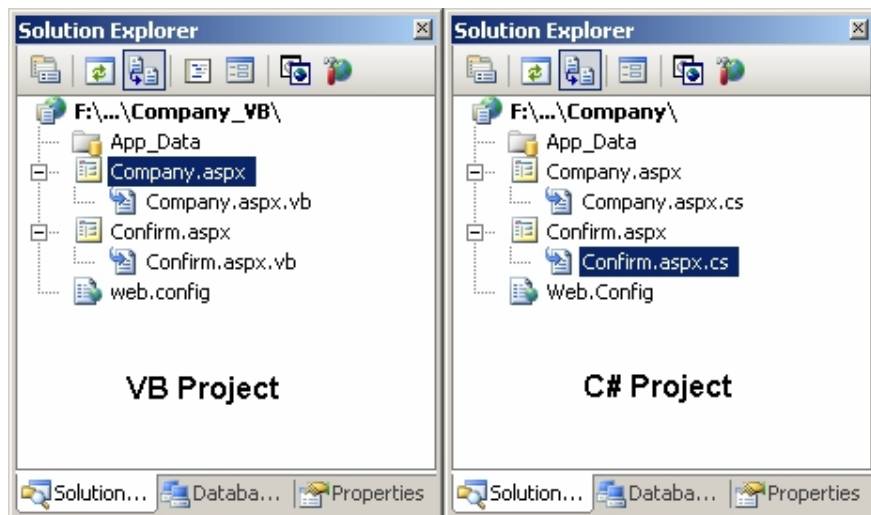
الكود بصيغة Visual Basic

كما نرى هنا أنه الآن من السهل جدا تضمين ملف جافا سكربت بدون مشاكل و الآن مع المكتبة الجديدة التي تم إضافتها الى العديد من مكتبات الدوت نت ٢٠٠٥ يمكن أستكشاف الكثير عن التكامل بين كود (C# أو Visual Basic) و الجافا سكربت كل ما عليكم هو البحث في الـ **MSDN** وسوف تجد الكثير من المواضيع التي سوف تكون مفيدة.

ميزة رهيبة و جديدة PreviousPage:

أولا نفهم فكرة هذه الميزة ثم نقوم بعرض السابق و الحالي أولا هذه الفكرة تعتمد على ربط الصفحات ببعضها عن طريق مكتبة جديدة أسمها الصفحة السابقة و التي تحتاج الى تعريف لى الصفحة المستقبلية و الآن سوف أعرض مشكلة و كيفية حلها بهذه الميزة الجديد.

إذا فرضنا أن لدينا شركة و نريد أدخل كلاً من أسمها و عنوانها و بعد ذلك نريد تأكيد البيانات، و الآن الى الحل،



الشكل ٢٠



الشكل 21 (صفحة إدخال بيانات الشركة)



الشكل 22 (صفحة تأكيد البيانات)

أود ذكر شيء صغير جداً وهو أنه في كود الـ HTML لا يوجد اختلاف بين صفحة منتجة بواسطة لغة الـ Visual Basic أو الـ C#.



```
<asp:Button ID="btnBranchInfo" runat="server" Text="الفرع بيانات إدخال"
PostBackUrl="~/Confirm.aspx"/>
```


بكل بساطة نقوم بتحديد الصفحة التي ستم نقل البيانات اليها أو ما يسمى بالـ **PostBack** في الجزء الخاص بـ **PostBackUrl** و بعد ذلك نقوم بكتابة بعض الكود في صفحة المصدر و التي سوف تصبح الصفحة السابقة ، و الآن لنكمل المثال

```
public string CompanyName
{
    get
    {
        return txtName.Text;
    }
}
public string CompanyAddress
{
    get
    {
        return txtAddress.Text;
    }
}
```

الكود بصيغة C#

```
Public ReadOnly Property CompanyName() As String
    Get
        Return txtName.Text
    End Get
End Property
Public ReadOnly Property CompanyAddress() As String
    Get
        Return txtAddress.Text
    End Get
End Property
```

الكود بصيغة Visual Basic

سيتم إضافة هذا الكود في الصفحة **Company.aspx**

```
protected void Page_Load(object sender, EventArgs e)
{
    ComName.Text = PreviousPage.CompanyName;
    ComAddress.Text = PreviousPage.CompanyAddress;
}
```

الكود بصيغة C#

```
Protected Sub Page_Load(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles Me.Load
    ComName.Text = PreviousPage.CompanyName
```

```
ComAddress.Text = PreviousPage.CompanyAddress
```

End Sub

الكود بصيغة Visual Basic

سيتم إضافة هذا الكود في الصفحة **Confirm.aspx**

و الآن لنقوم بتوضيح الأمور نقوم بضبط أعدادات الكائن **PreviousPage** و التي تحتوى جميع أعضاء الموجودة فالصفحة السابقة لذلك قمنا بعمل خاصيتين للقراءة فقط و هما كلا من **CompanyName** و **CompanyAddress** ونقوم بتعريف الصفحة الحالية بالصفحة السابقة عن طريق الموجة الموجود في أعلى الصفحة كما هو موضح في الكود التالي

```
<%@PreviousPageType VirtualPath="~/Company.aspx"%>
```

بذلك نكون عرفنا الصفحة السابقة للصفحة الحالية و الآن يمكننا أستهلاك مكونات الصفحة السابقة في الصفحة الحالية كما هو الحال في كود بدء التحميل في صفحة **Confirm.aspx** و هذا هو مثال حي عند تنفيذ المثال



الشكل 23



الشكل 24

هناك أيضا الكثير يمكنك الاستفادة منه في ميزة دعم مكتبة الصفحة السابقة ، وهناك الكثير من الخواص التي سوف تساعدك كثير عند أستكشافك لهذا الموضوع في فيجوال ستوديو دوت نت ٢٠٠٥ .

تعدد منظر العرض Multiple view for controls groups

هذه الميزة حقا رائعة و سوف توفر الكثير من الوقت لكثير من المطورين يبدو وصف هذه الميزة غامضا لحد ما و لكن كل شئ سوف يكون واضح بعد أول مثال.

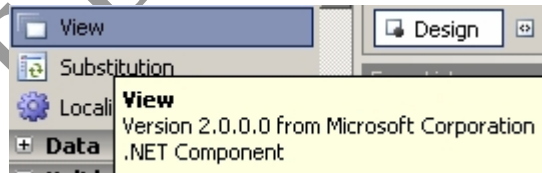
المثال لدين هو أنك تود عمل مساعد Wizard يقوم بعمل تسجيل لعضو و لتسجيل عضو هناك خطوات و هي:

- ١- عرض رخصة التعامل مع موقع من قبل الشركة المنتجة للموقع
 - ٢- أقرار بأن السن تجاوز ١٥ سنة
 - ٣- إدخال الأسم الحركي و البريد الإلكتروني
 - ٤- ادخل بيانات الأسم و عنوان
 - ٥- شكر للعضو على تسجيله في الموقع
- في السابق لعمل مثال كهذا هناك أكثر من طريقة :
- عمل أكثر من لوحة Panel و وضع عليه الأدوات و نقوم بعمل أظهار و أخفاء للوحة Show and Hide
 - عمل ٦ صفحات نقوم بالمهمة

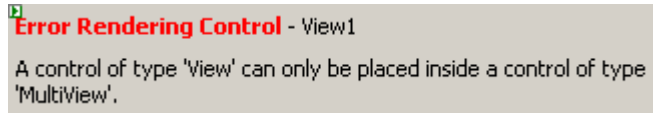
الآن في فيجوال ستوديو ٢٠٠٥ الوضع مختلف مع وجود أداة جديدة أسمها "MultiView" و هذه الأداة بكل بساطه تحتوى على مجموعة من عنصر يسمى View و العنصر View يحتوى على مجموعة من الادوات و من الاداه MultiView يتم تحديد النشط بحيث يظهر و يخفي باقى الـ View.



الشكل ٢٥ (اختيار العنصر MultiView)



الشكل ٢٦ (ثم اختيار العنصر View و وضعة داخل MultiView)



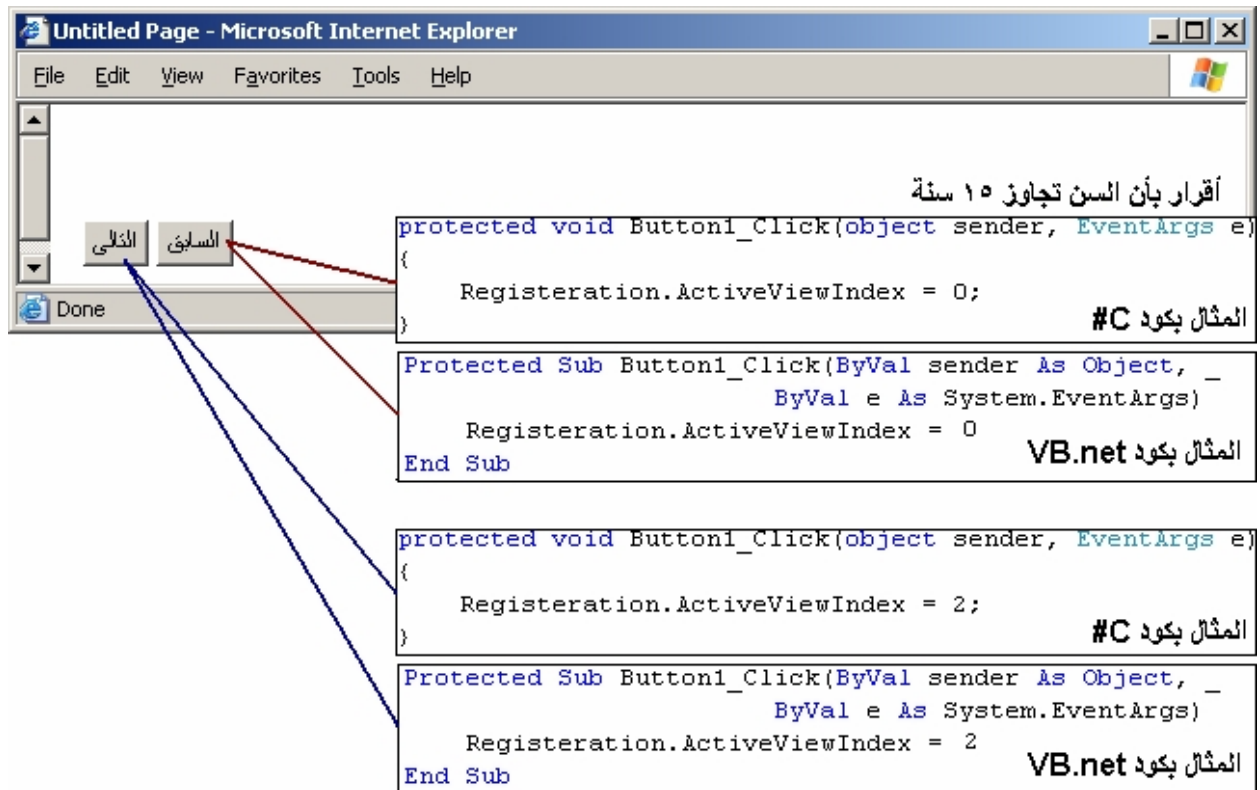
الشكل 27 (عند محاولة وضع الـ View خارج الـ MultiView تظهر رسالة خطأ)



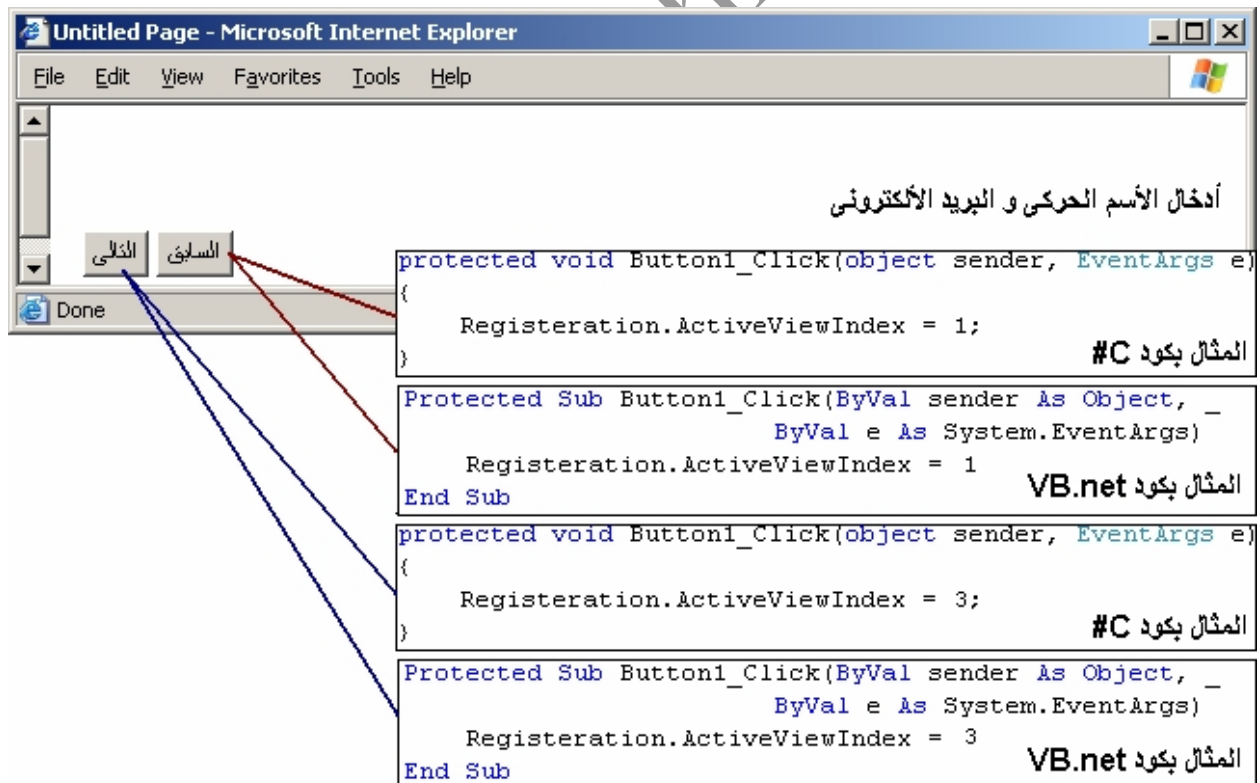
الشكل 28 (هنا يتم تحديد مجموعات الأدوات في الـ **View** و يجب هنا ملاحظة ترتيب العناصر لأن أساس التعامل مع العنصر **MultiView** هو الترتيب **View Index** سوف يتم توضيح هذه النقطة في شرح الكود)



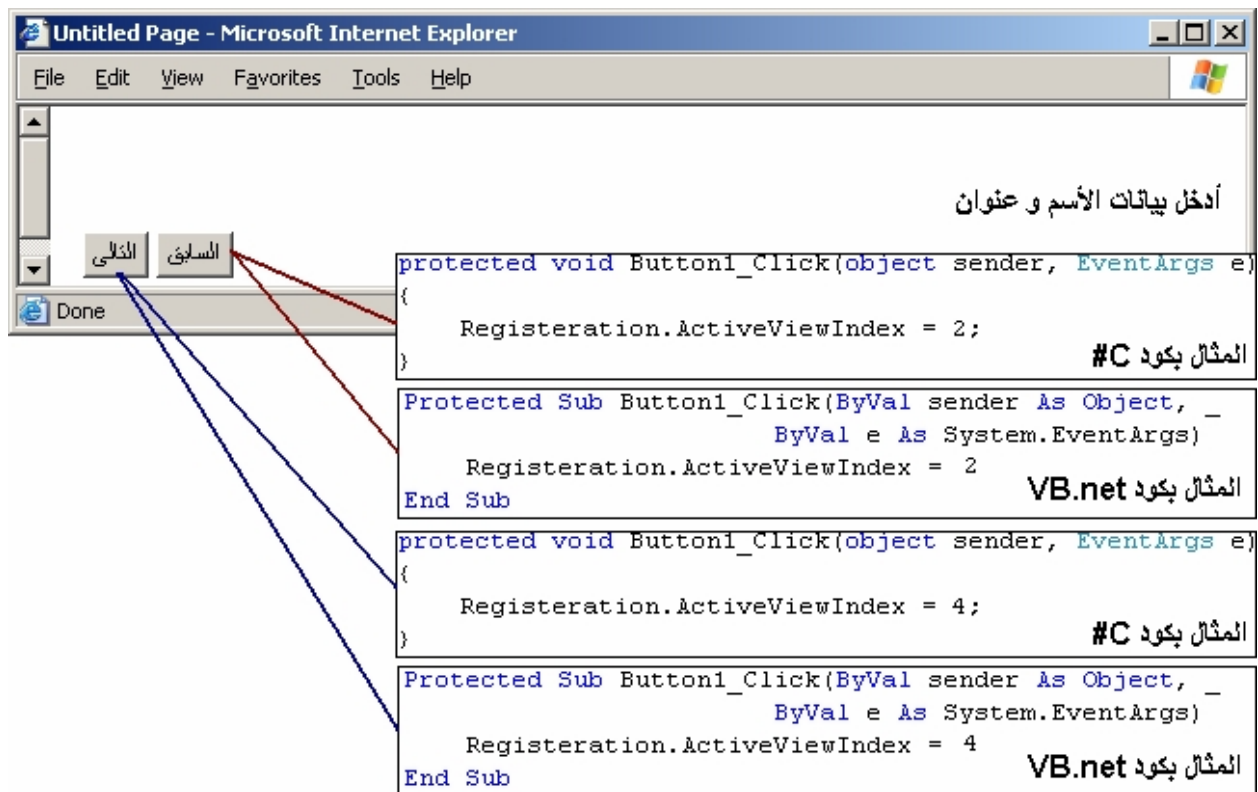
الشكل 29 (المنظر الأول **viw_License**)



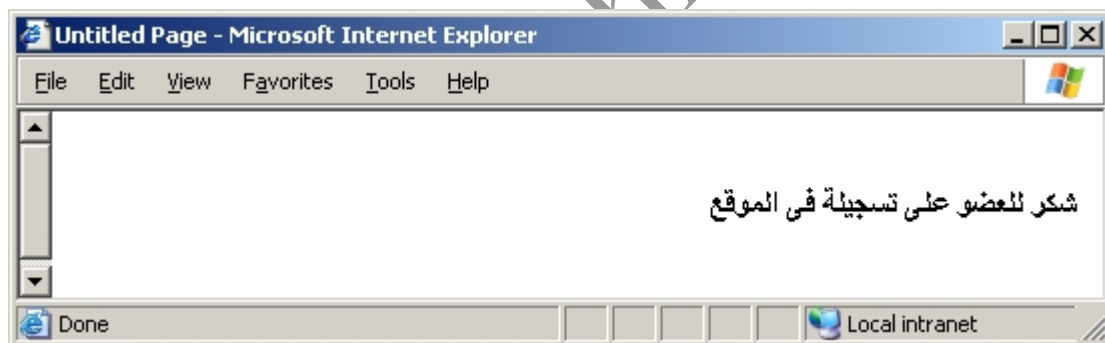
الشكل 30 (المنظر الثاني view_Admission)



الشكل 31 (المنظر الثالث view_MainInfo)



الشكل 32 (المنظر الرابع الرابع (viw_Info)

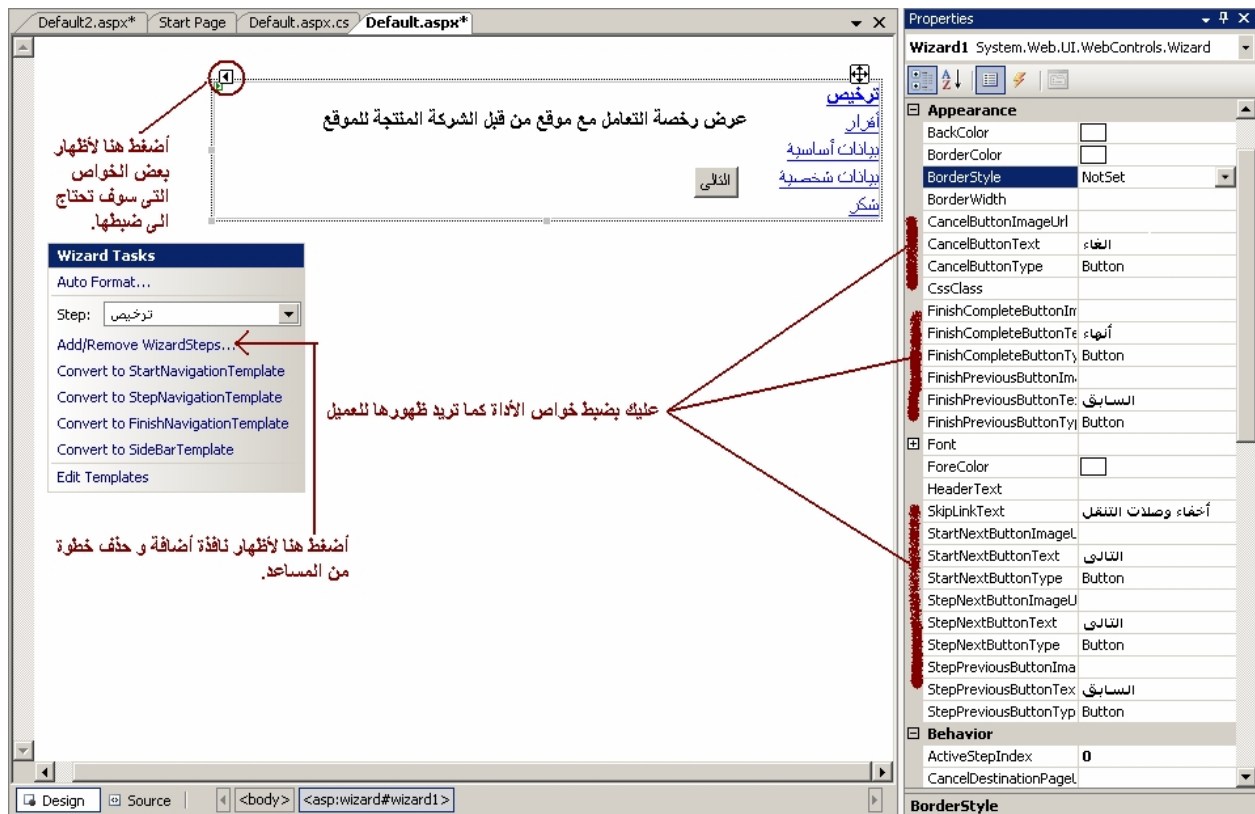


الشكل 33 (المنظر الخامس (viw_Thanks)

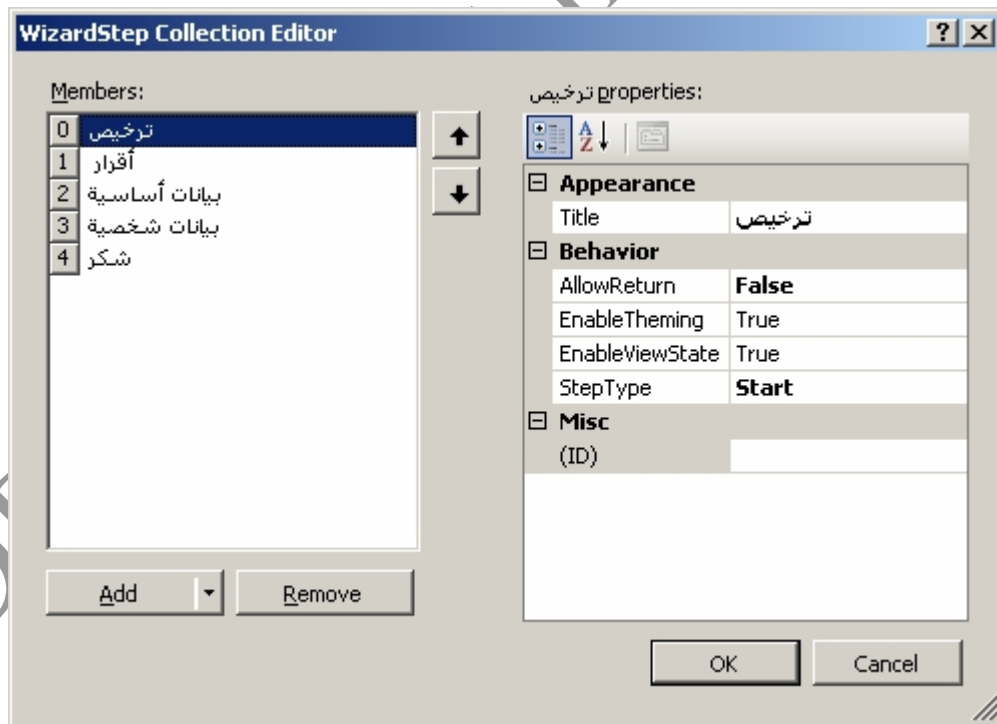
كما نرى أن هذه الاداة وفرة الكثير من الوقت و العناية الذي كان على المطور عملة لظهور بعض العناصر في شكل مجموعات و كما ترى أنه واضح جدا كيفية التعامل مع الاداة حيث أن كل ما عليك القيام به هو وضع MultiView و بداخله مجموعه من View حسب ما تريد ثم ضع في كل View مجموعة العناصر التي تؤدي الغرض المراد تنفيذه.

المساعد Wizard

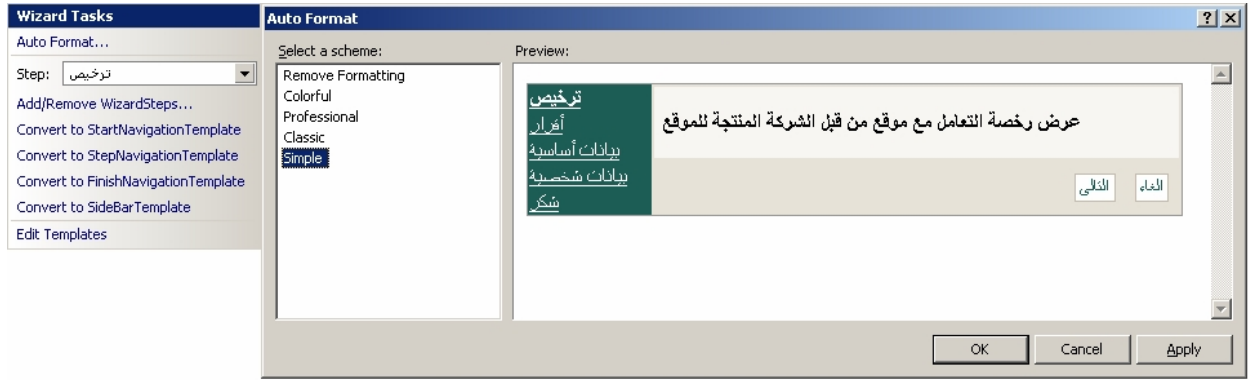
هذه الخاصية جبارة من وجهة نظري حيث أنها تقوم بنفس عمل الاداة السابقة و لكن بشكل مختلف حيث أن هذه تقوم بوضعه و من ثم تقوم بتحديد الخطوات المطلوبة و بعد ذلك هذه الاداة سوف تقوم بكل شئ ، بمعنى أنك لست مضطر الى كتابة أي كود لآتمام العملية كما هو الحال في الاداة MultiView.



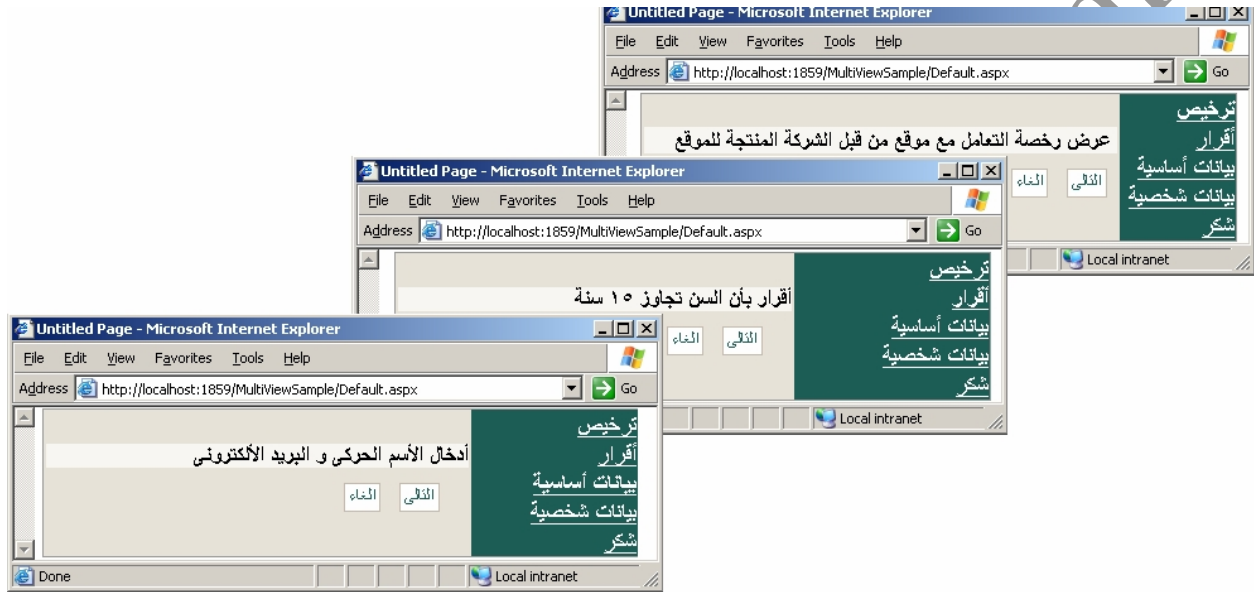
الشكل 34 (شكل عام لكيفية التعامل مع الأداة)



الشكل 35 (شاشة إدخال خواص الخطوات)



الشكل 36 (استخدام الخاصية AutoFormat لتحسين الشكل في العرض)

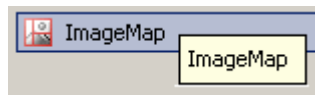


الشكل 37 (جانب من المثال بعد التنفيذ)

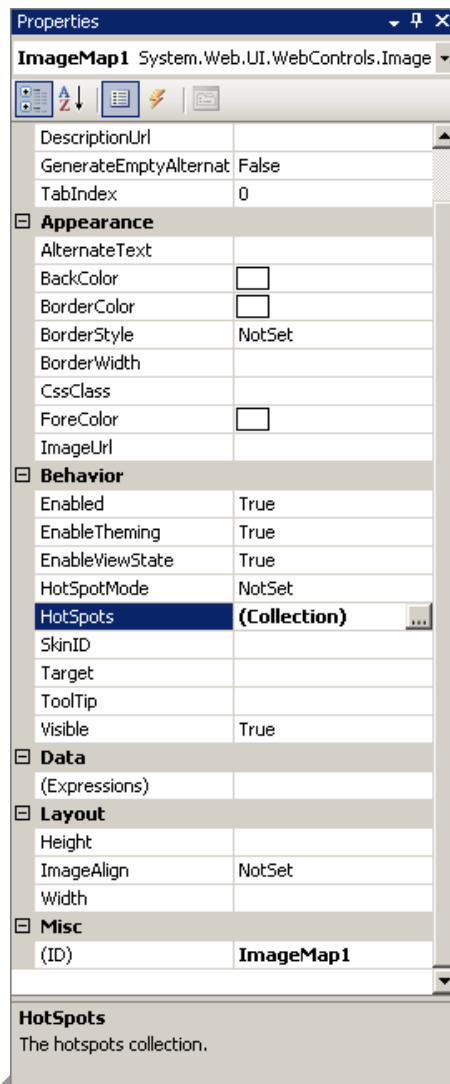
هناك الكثير لتحدث عنه في فيجوال ستوديو ٢٠٠٥ و سوف ترى في الفصول التالية الكثير من الأمكانيات التي كانت تستهلك الكثير من الوقت لكي يتم تنفيذها في الإصدارات السابقة من فيجوال ستوديو.

الأداة خارطة الصور Image Map:

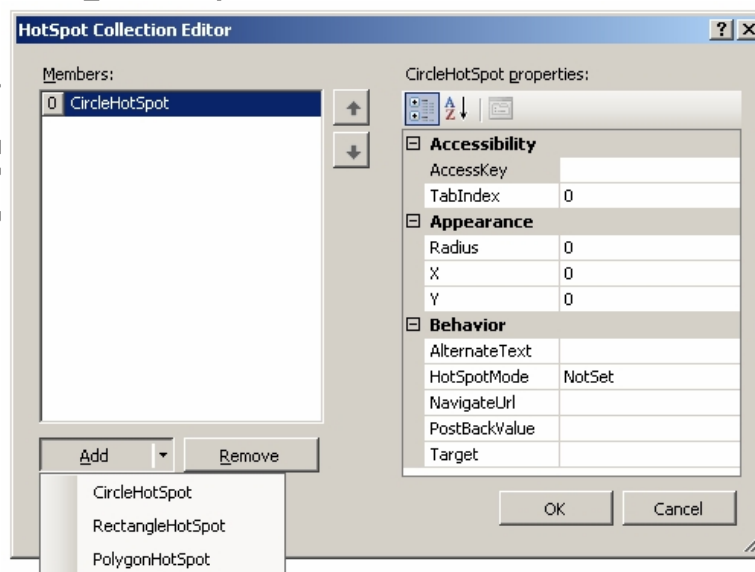
هذه الأداة هي أداة جديدة تم تزويدها في الإصدار الجديد من الفيجوال ستوديو هذه الاداة سوف توفر عليك عناء التعامل مع البرامج التي فتقوم بالعمل المشابه بالإضافة الى أنه يمكن أن جعل هذه الأداة تتفاعل مع بقية العناصر الموجودة على الصفحة و الآن سوف أعرض كيفية التعامل مع هذه الأداة الجديدة.



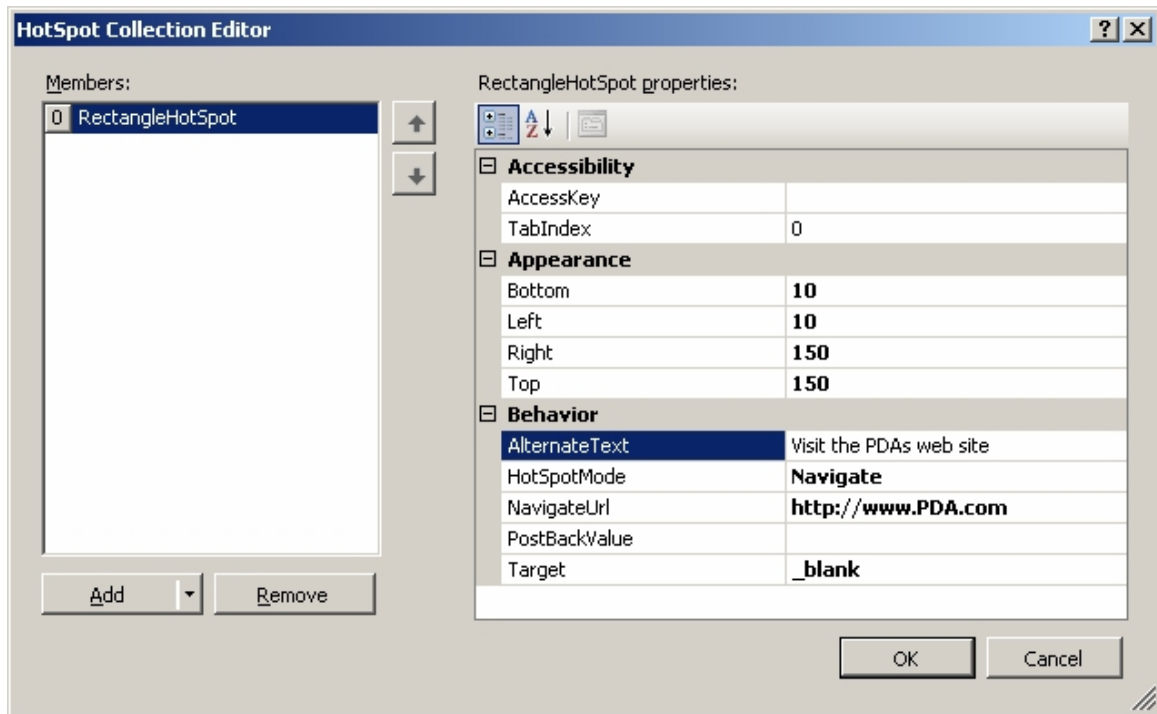
الشكل ٣٨ قم بأختيار الأداة من التبويب Standard



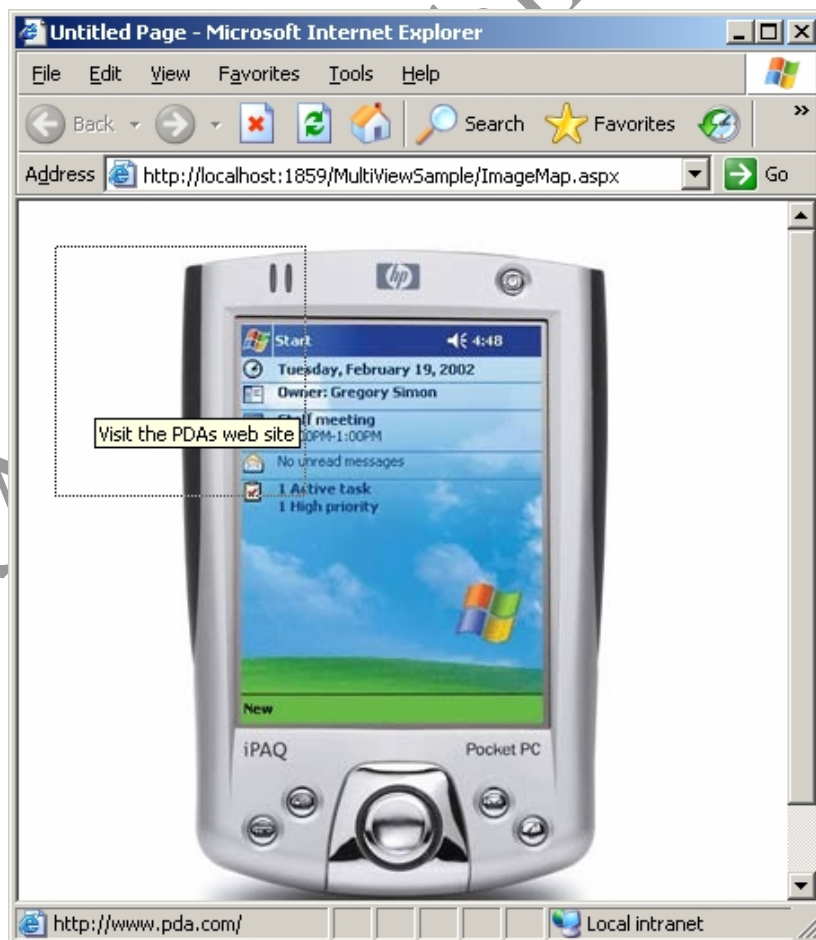
الشكل 39 عليك بعد ذلك تحديد أهم خاصية وهي HotSpots وهي الرسومات الغير المرئية والتي حين يمر عليها المستخدم تظهر كوصلات



الشكل 40 هنا يتم تحديد الأشكال التي سوف ترسم على الصورة سوء كانت دائرة أو مربع أو شكر غير معلوم عدد أضلعة



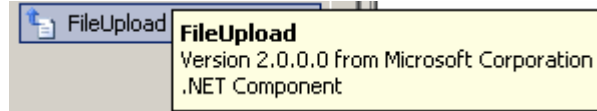
الشكل 41 يعد ذلك قم بتحديد خواص الشكل كالطول والعرض والوصلة المراد الذهاب إليها عند ضغط المستخدم على الشكل بالإضافة إلى مكان عرض الوصلة هل يتم ذلك في نافذة جديدة أم في نفس النافذة



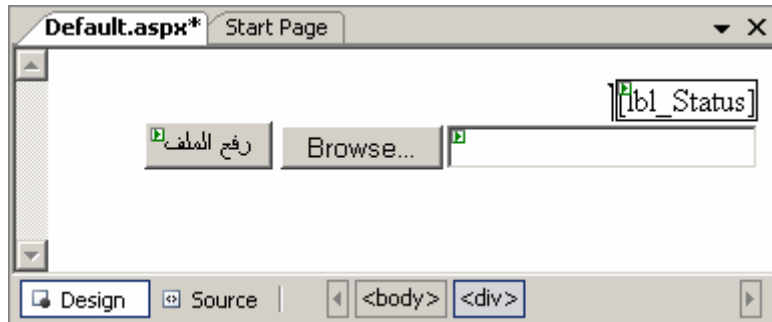
الشكل ٤٢ و كما نرى في المثال أننا قمنا بوضع صورة و بمجرد المرور على الشكل الخفي تظهر الوصلة.

يمكن استخدام هذه الخاصية في عمل تطبيقات غنية بالصور و الوسائط المتعددة التي سوف تساعد المستخدم في التعامل مع بعض الأشياء، على سبيل المثال عمل تطبيق لمحاكاة الهاتف المحمول، لكي يتيح للمستخدم التعرف على الأماكن الموجودة في الهاتف عن طريق هذا التطبيق.

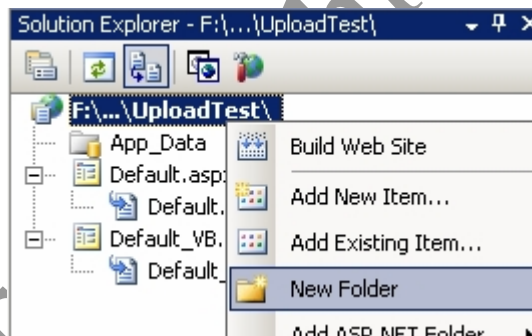
أداة رفع الملفات FileUpload:



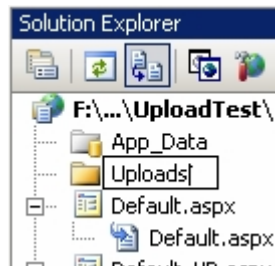
الشكل 43 قم بأختيار الأداة من نافذة الأدوات من التبويب Standard



الشكل 44 قم بعمل الشكل التالي بأدراج Label و Button للصفحة



الشكل 45 قم بإنشاء مجلد جديد و الذي سيتم رفع الملفات فيه



الشكل 46 قم بتغيير أسمية حسب ما تريد

```
protected void btn_Upload_Click(object sender, EventArgs e)
{
```

هنا نقوم بتعرف متغير من نوع سلسلة حرفية حيث سوف نضع فيه المسار الحقيقي للملف

```
string strRealPath;
```

هنا نقوم بأخذ المسار الحقيقي من العنصر Request و وضعه في المتغير الذي قمنا بحجزة

```
strRealPath = Request.PhysicalApplicationPath;

هنا نقوم بإضافة المجلد الذي قمنا بإنشائه الى السلسلة الحرفية

strRealPath += "Uploads\\";

نقوم باختبار ما إذا كان قام المستخدم باختيار ملف أم لا

if (fu_Uploader.HasFile)
{
    في حالة أن المستخدم أختار ملف نقوم باختبار هل مجلد الملفات موجود على الخادم أم لا

    if (System.IO.Directory.Exists(strRealPath))
    {
        في حالة أن المجلد موجود نقوم باختبار هل المجلد يحتوي على نفس الملف أم لا

        if (!System.IO.File.Exists(strRealPath + fu_Uploader.FileName))
        {
            في حالة أن الملف غير موجود نقوم باختبار الملف هل حجم الملف أقل من 15000 حرف أم لا

            if (fu_Uploader.PostedFile.ContentLength < 15000)
            {
                في حالة أن الملف أقل من المساحة المحددة نقوم برفع الملف و عرض رسالة تفيد بأن الملف قد تم رفعه

                fu_Uploader.SaveAs(strRealPath + fu_Uploader.FileName);
                lbl_Status.Text = "الملف رفع تم";
            }
            else
            {
                في حالة أن الملف أكبر من 15000 حرف تظهر للمستخدم رسالة تخبرة

                lbl_Status.Text = "حرف 15000 من أكبر الملف حجم";
            }
        }
        else
        {
            في حالة وجود ملف على الخادم بنفس الاسم يتم أخبار المستخدم هناك الكثير سوف يتسأل لماذا نقوم باختبار هل الملف موجود على الخادم أم لا ببساطة لتجنب فقدان أى ملفات

            lbl_Status.Text = "الملف أسم تغيير حاول موجود الملف هذا";
        }
    }
    else
    {
        في حالة عدم وجود مجلد الملفات المرفوعة

        lbl_Status.Text = "غير المرفوعة الملفات حفظ مجلد : الخادم من خطأ";
    }
}

"موجود";
}
```

```

}
else
{
    في حالة إذا قام المستخدم بعدم اختيار ملف للرفع
    lbl_Status.Text = "ملف أختار فضلك من لرفعه ملف يوجد لا";
}

يتم تغيير لون الرسالة لكي تبدو واضحة للمستخدم
lbl_Status.Text = "<font color=red><B>" + lbl_Status.Text +
"</B></font>";
}

```

الكود بصيغة C#

```

Protected Sub btn_Upload_Click(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles btn_Upload.Click
    هنا نقوم بتعرف متغير من نوع سلسلة حرفية حيث سوف نضع فيه المسار الحقيقي للملف
    Dim strRealPath As String
    هنا نقوم بأخذ المسار الحقيقي من العنصر Request و وضعه في المتغير الذي قمنا بحجزة
    strRealPath = Request.PhysicalApplicationPath
    هنا نقوم بإضافة المجلد الذي قمنا بإنشاءه الى السلسلة الحرفية
    strRealPath += "Uploads\"
    نقوم بأختبار ما إذا كان قام المستخدم بأختيار ملف أم لا
    If fu_Uploader.HasFile Then
        في حالة أن المستخدم أختار ملف نقوم بأختبار هل مجلد الملفات موجود على الخادم أم لا
        If System.IO.Directory.Exists(strRealPath) Then
            في حالة أن المجلد موجود نقوم بأختبار هل المجلد يحتوي على نفس الملف أم لا
            If Not System.IO.File.Exists(strRealPath) Then
                في حالة أن الملف غير موجود نقوم بأختبار الملف هل حجم الملف أقل من 15000 حرف أم لا
                If fu_Uploader.PostedFile.ContentLength < 15000 Then
                    في حالة أن الملف أقل من المساحة المحددة نقوم برفع الملف و عرض رسالة تفيد بأن الملف قد تم رفعه
                    fu_Uploader.SaveAs(strRealPath +
fu_Uploader.FileName)
                    lbl_Status.Text = "الملف رفع تم"
                Else
                    في حالة أن الملف أكبر من 15000 حرف نظهر للمستخدم رسالة تخبرة
                    lbl_Status.Text = "حرف 15000 من أكبر الملف حجم"
                End If
            Else

```

في حالة وجود ملف على الخادم بنفس الاسم يتم أخبار المستخدم هناك الكثير سوف يتسأل لماذا نقوم بأختبار هل الملف موجود على الخادم أم لا ببساطة لتجنب فقدان أى ملفات

```
lbl_Status.Text = "الملف أسم تغيير حاول موجود الملف هذا"
```

```
End If
```

```
Else
```

في حالة عدم وجود مجلد الملفات المرفوعة

```
lbl_Status.Text = "غير المرفوعة الملفات حفظ مجلد : الخادم من خطأ"
```

"موجود"

```
End If
```

```
Else
```

في حالة إذا قام المستخدم بعدم أختيار ملف للرفع

```
lbl_Status.Text = "ملف أختار فضلك من لرفعه ملف يوجد لا"
```

```
End If
```

يتم تغيير لون الرسالة لكي تبدو واضحة للمستخدم

```
lbl_Status.Text = "<font color=red><B>" + lbl_Status.Text +
```

```
"</B></font>"
```

```
End Sub
```

الكود بصيغة Visual Basic



الشكل 47 في حالة عدم أختيار المستخدم لملف



الشكل 48 في حالة عدم وجود المجلد الذي سيتم حفظ الملفات فيه على الخادم



الشكل 49 في حالة وجود ملف بنفس اسم الذي أختاره المستخدم و أهمية هذا الشرط هو أنه يحمي من فقد الملفات أو ظهور رسالة خطأ



الشكل 50 في حالة أن حجم الملف أكبر من ١٥٠٠٠ حرف

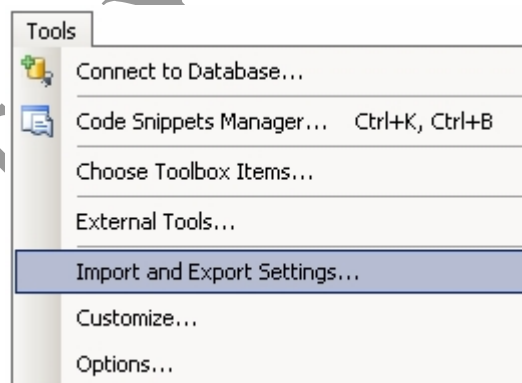


الشكل 51 في حالة مطابقة كل الشروط و عم حدوث مشاكل و رفع الملف بشكل سليم.

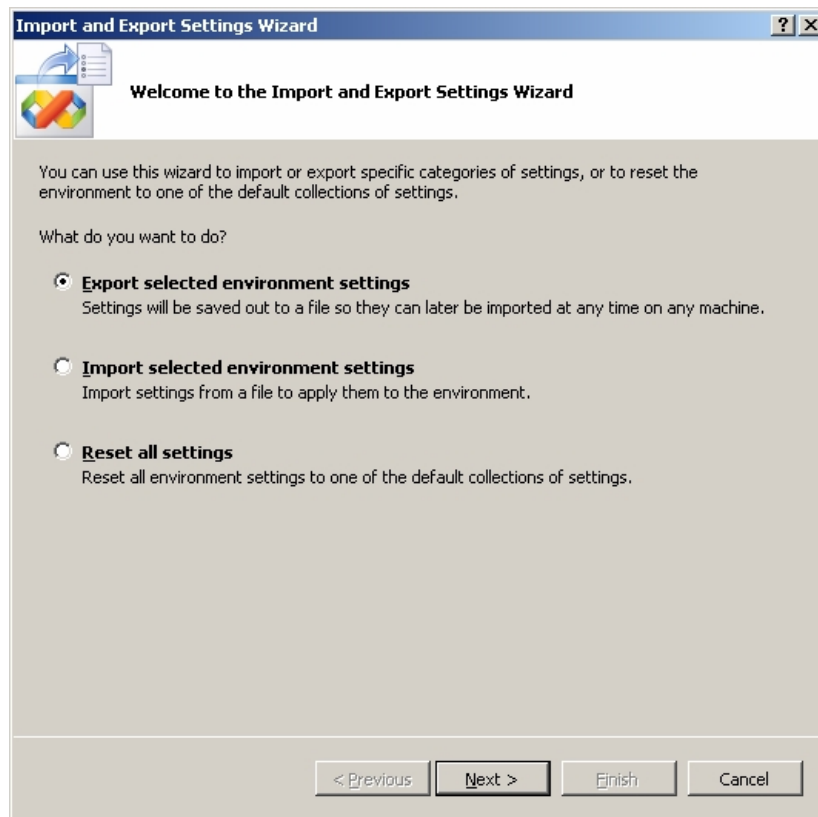
هناك من يرى أن الموضوع معقد جدا حيث أننا قمنا بعمل الكثير من التأكيدات **Validation** و لكن هل تعلم أن المبرمج الجيد لابد و أن يقوم بذلك حتى لا يحدث أى خطأ من شأنه تعطيل التطبيق بالإضافة الى أن معظم المخترقين ينتظرون أخطاء الآخرين فمن الممكن أن تكون أحدى أخطاءك مدخل لأحد المخترقين و التى من الممكن توصلة الى أخزء لا يسمح له بدخولها.

حفظ أعدادات فيجوال ستوديو ٢٠٠٥:

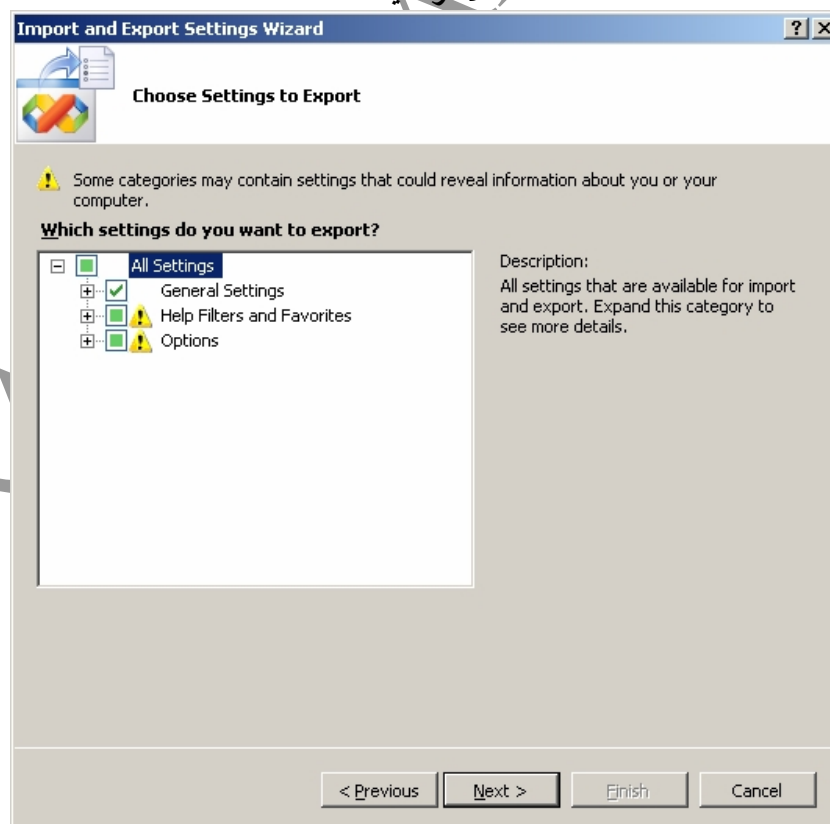
فى الأصدارات السابقة كانت فى حالة أنك قمت بتغيير جهاز تحتاج الى أن تقوم بأعدادات الفيجوال ستوديو لكي تعمل عليه كما أنت معتاد و كان هذه الأمر بالتأكد يأخذ الكثير من الوقت لان العملية هى عملية روتينية لا يوجد فيه الجديد و لكن الآن مع فيجوال ستوديو ٢٠٠٥ تم حل هذه المشكلة بكل بساطة يمكن أستيراد و تصدير و حفظ أعدادات الفيجوال ستوديو و فى أقل من ثوانى،



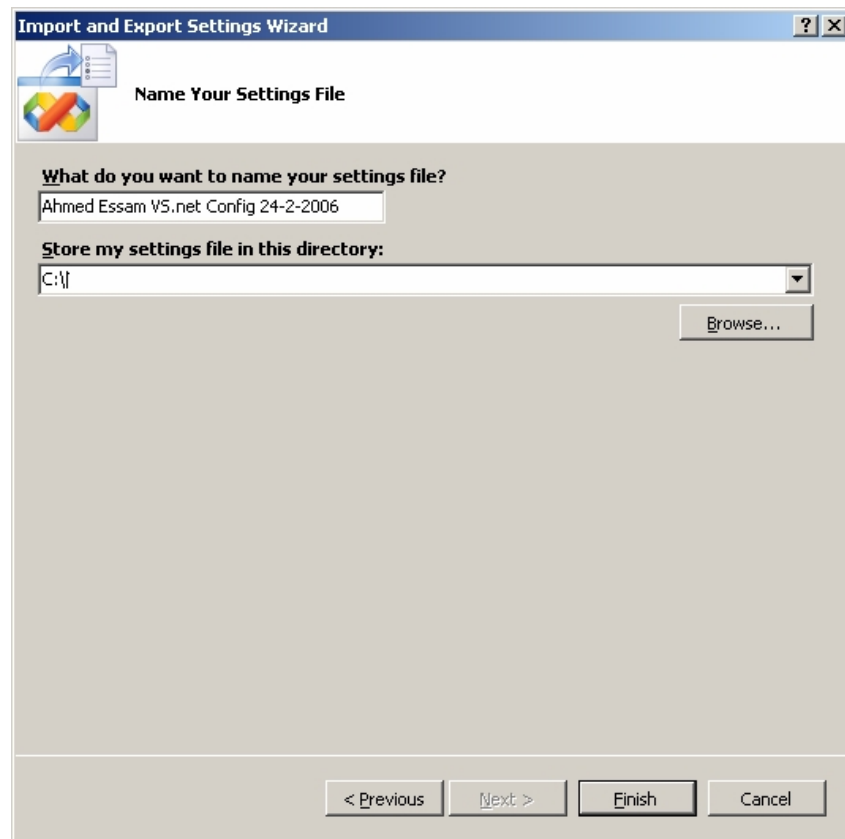
الشكل 52 أولا للتعامل مع هذه الميزة الجديدة كل ما عليك هو الدخول على القائمة **Tools** و من ثم اختيار **Import and Export Settings** و بعد ذلك سوف يظهر لك معالج أعدادات الفيجوال ستوديو.



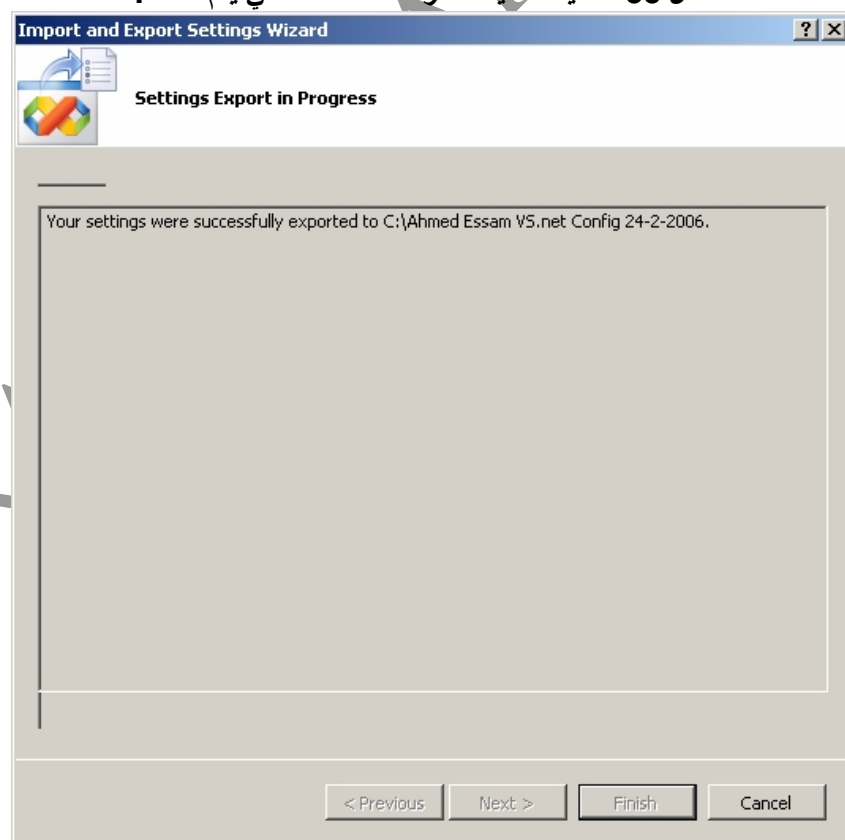
الشكل 53 في هذه النافذة سوف يسألك هل تريد استيراد إعدادات أم حفظ ملف إعدادات أم أنك تريد إعادة الإعدادات للحالة الافتراضية،



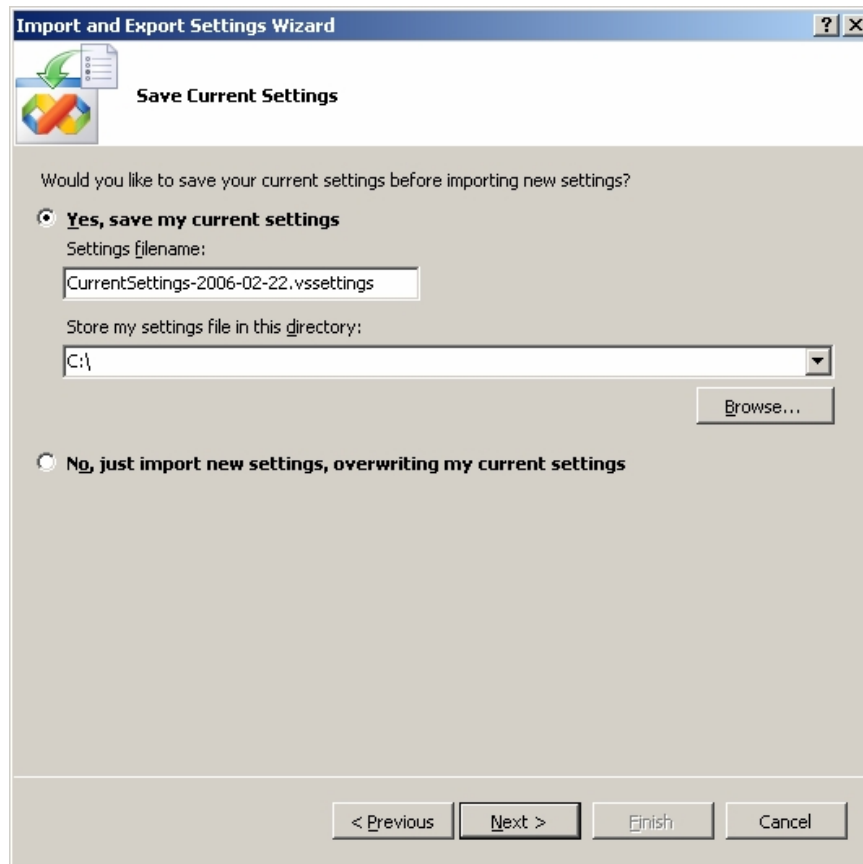
الشكل 54 قمنا باختيار حفظ ملف إعدادات و من هنا يتم اختيار الإعدادات التي تريد حفظها العلامة المثلث الأصفر تشير الى أن هناك إعدادات تحتوى على بيانات شخصية.



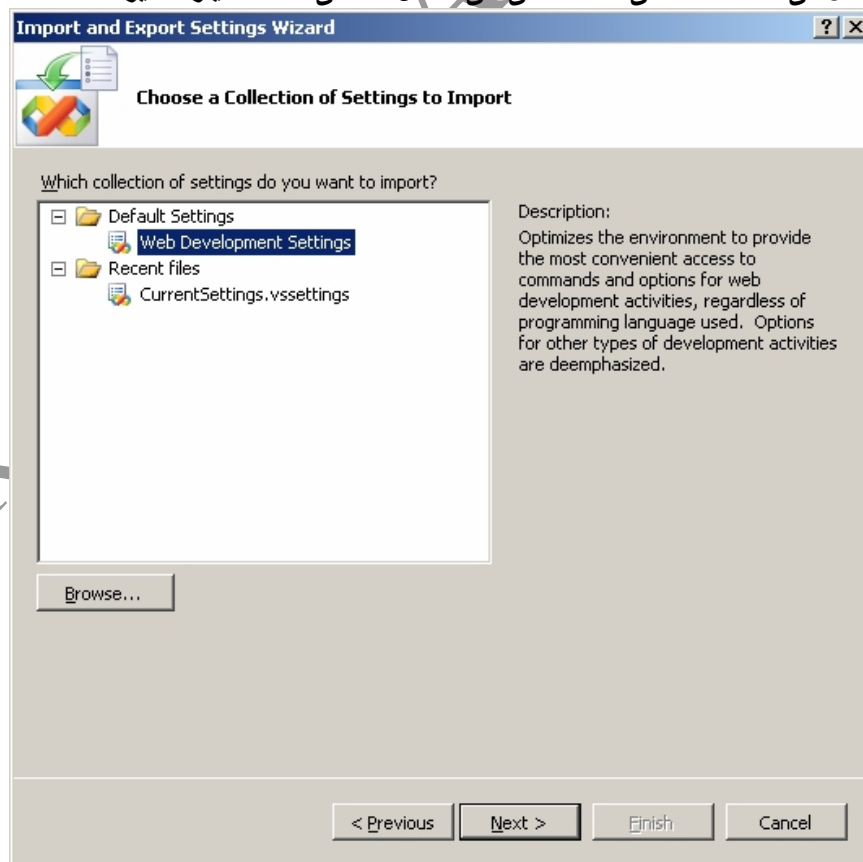
الشكل 55 هنا يتا تحديد مسار الملف أسمة لكي يتم حفظة.



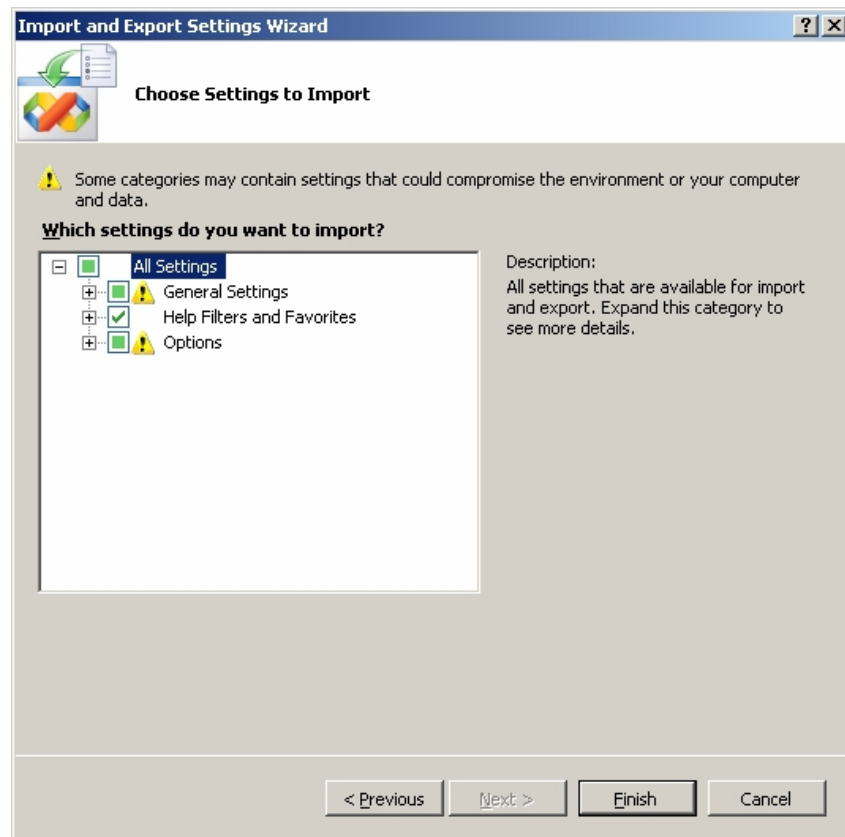
الشكل 56 هذه النافذة سوف تظهر عند الانتهاء من العملية.



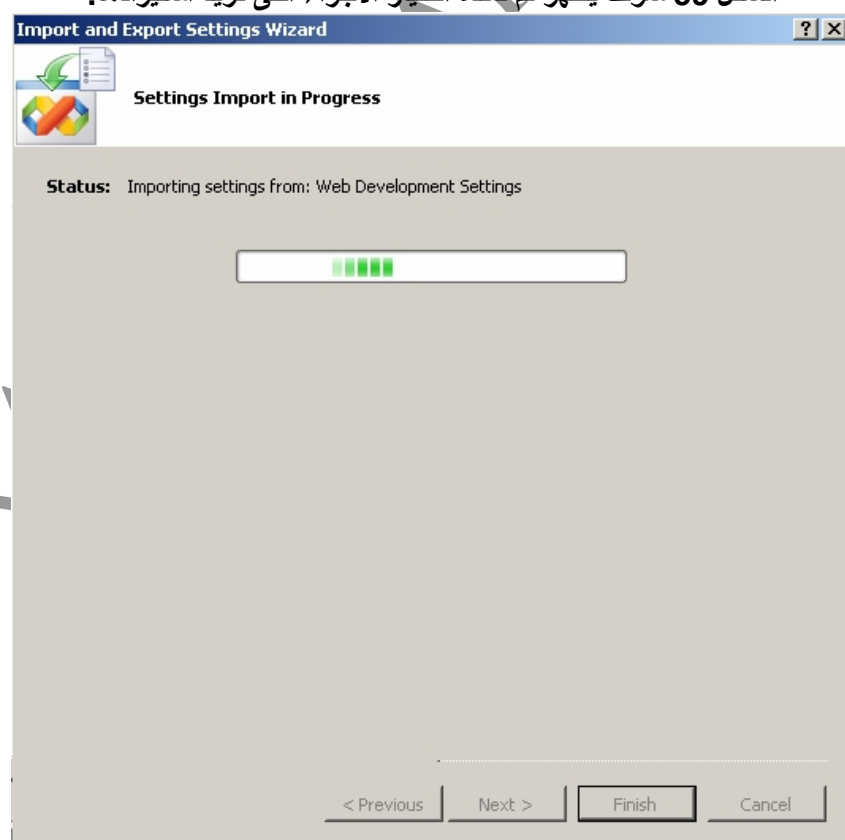
الشكل 57 في هذه النافذة هي النافذة التي تلي الشكل ٥٣ في حالة اختيار استيراد ملف إعدادات.



الشكل 58 هنا يقوم المعالج بتخييرك بين الإعدادات الافتراضية و الإعدادات الحالية و هناك إمكانية لأختيار ملف إعدادات



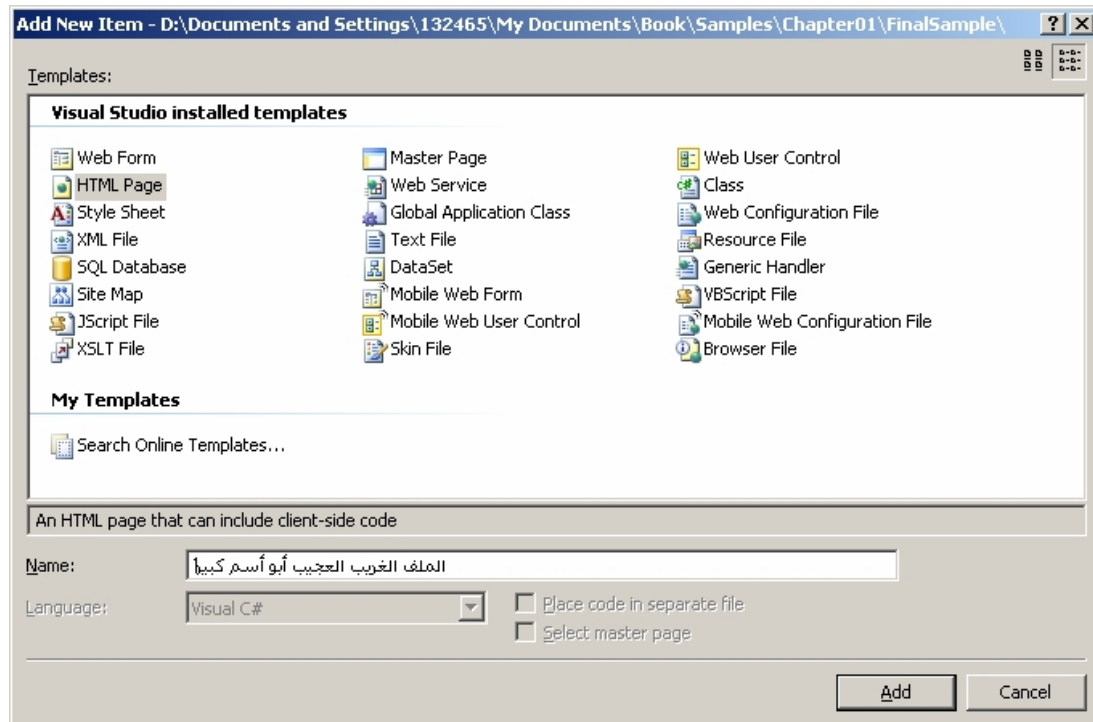
الشكل 59 سوف يظهر لم نافذة اختيار الأجزاء التي تريد أستيرادها.



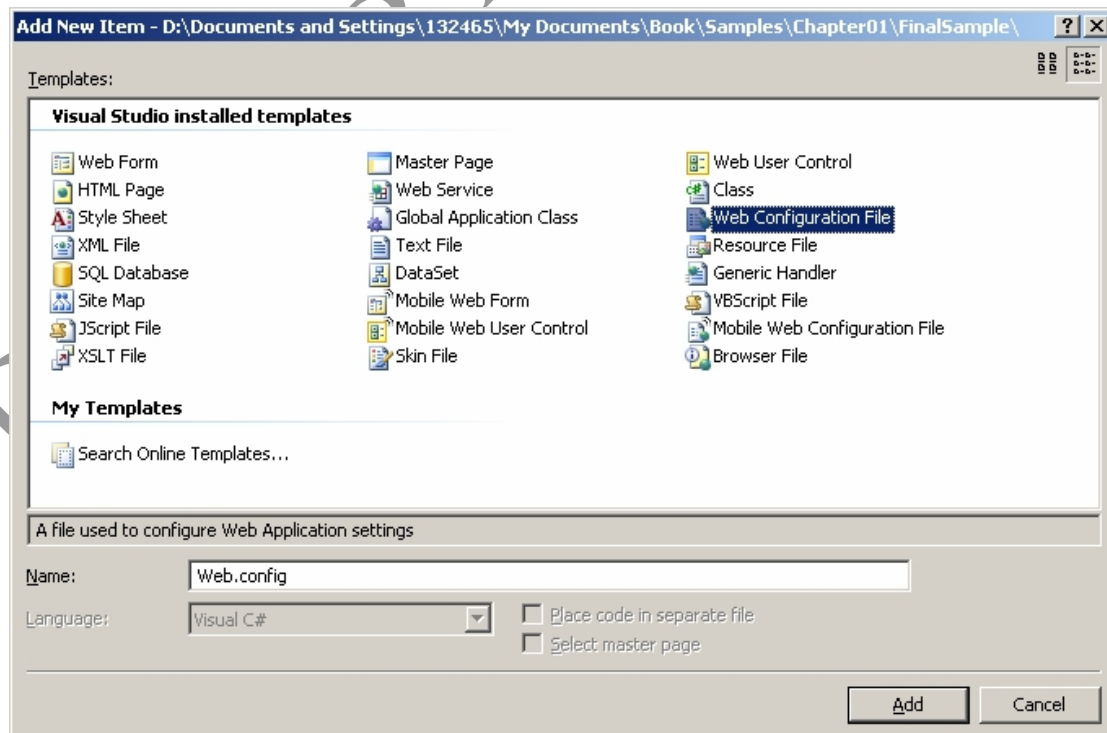
الشكل 60 بعد الانتهاء سوف تجد أن الإعدادات عادة كما حفظتها في الملف.

:URL Mapping

خاصية جديدة و مفيدة في بعض الأحيان حيث عن طريق هذه الميزة لن تضطر الى وضع أسم الملف الحقيقي و سوف نرى كيف يتم أستغلال هذه الميزة بمثال، على سبيل المثال في بعض الأحيان يسبب تسمية الملف باللغة العربية الى حدوث بعض المشاكل الآن سوف نرى حل لهذا المشكلة و يمكن أيضا أستغلال هذه الخاصية في الكثير الأعمال وسوف يتضح هذا بمثال عملي،



الشكل 61 أولا إضافة صفحة Web Page عادية لأختبار الخاصية



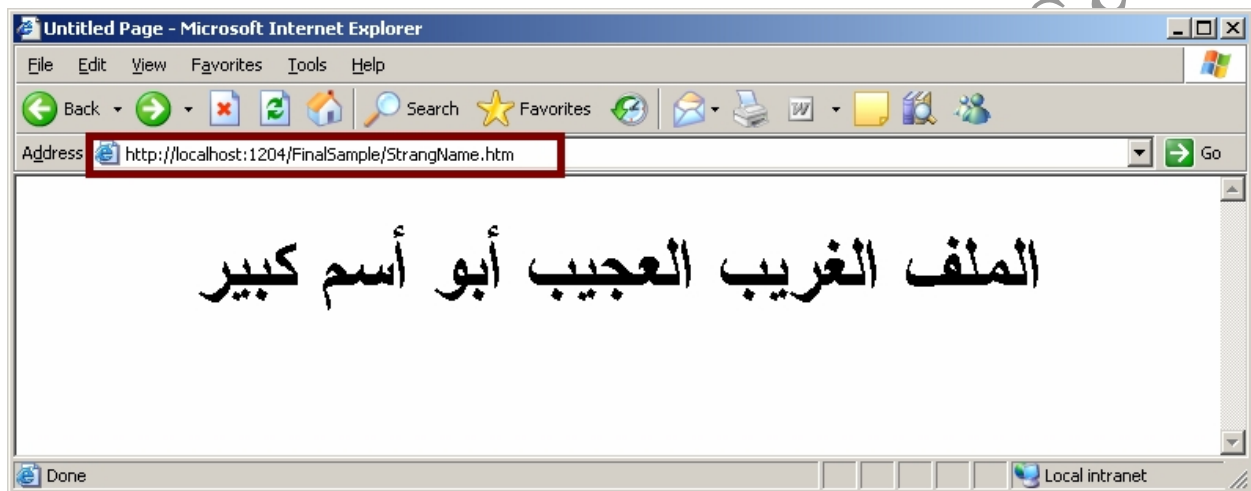
الشكل ٦٢ نقوم بإضافة ملف Web.Config في حالة عدم وجود ملف

```
<system.web>
    <urlMappings enabled="true">
        <add url="~/StrangName.htm"
            mappedUrl="~/العجيب الغريب الملف كبر.htm"/>
    </urlMappings>
</system.web>
```

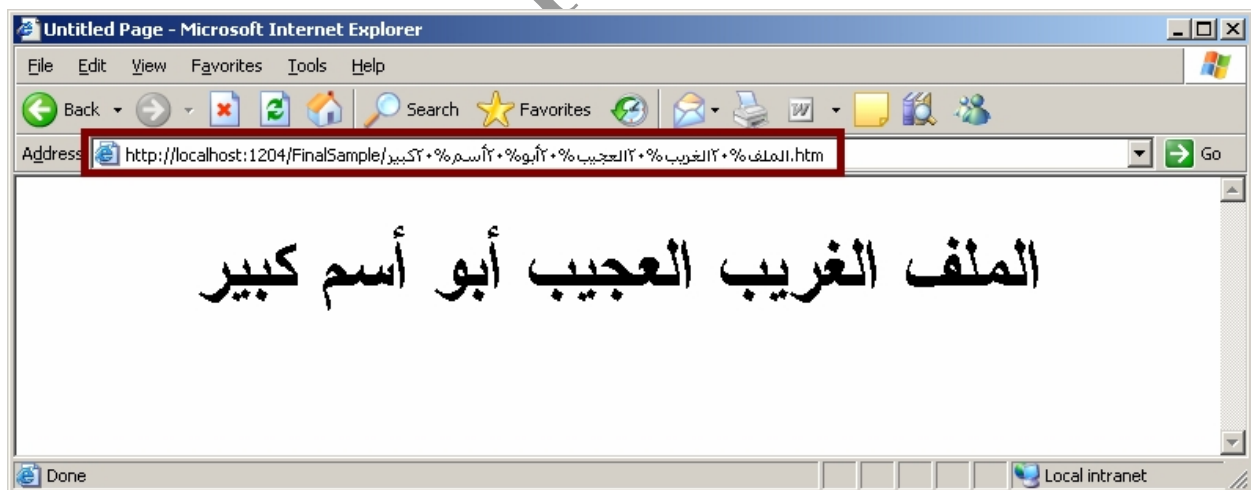
نقوم بفتح Web.Config ونقوم بإضافة هذا الكود والذي نحدد فيه الاسم الوهمي URL و اسم الملف المطلوب الإشارة اليه أو المطلوب عرضة Mapped URL

```
<a href="StrangName.htm">أختبار الاسم الجديد</a>
```

في الصفحة التي خصصناها للاختبار قم بإضافة هذا الكود والذي يقوم بفتح الصفحة.

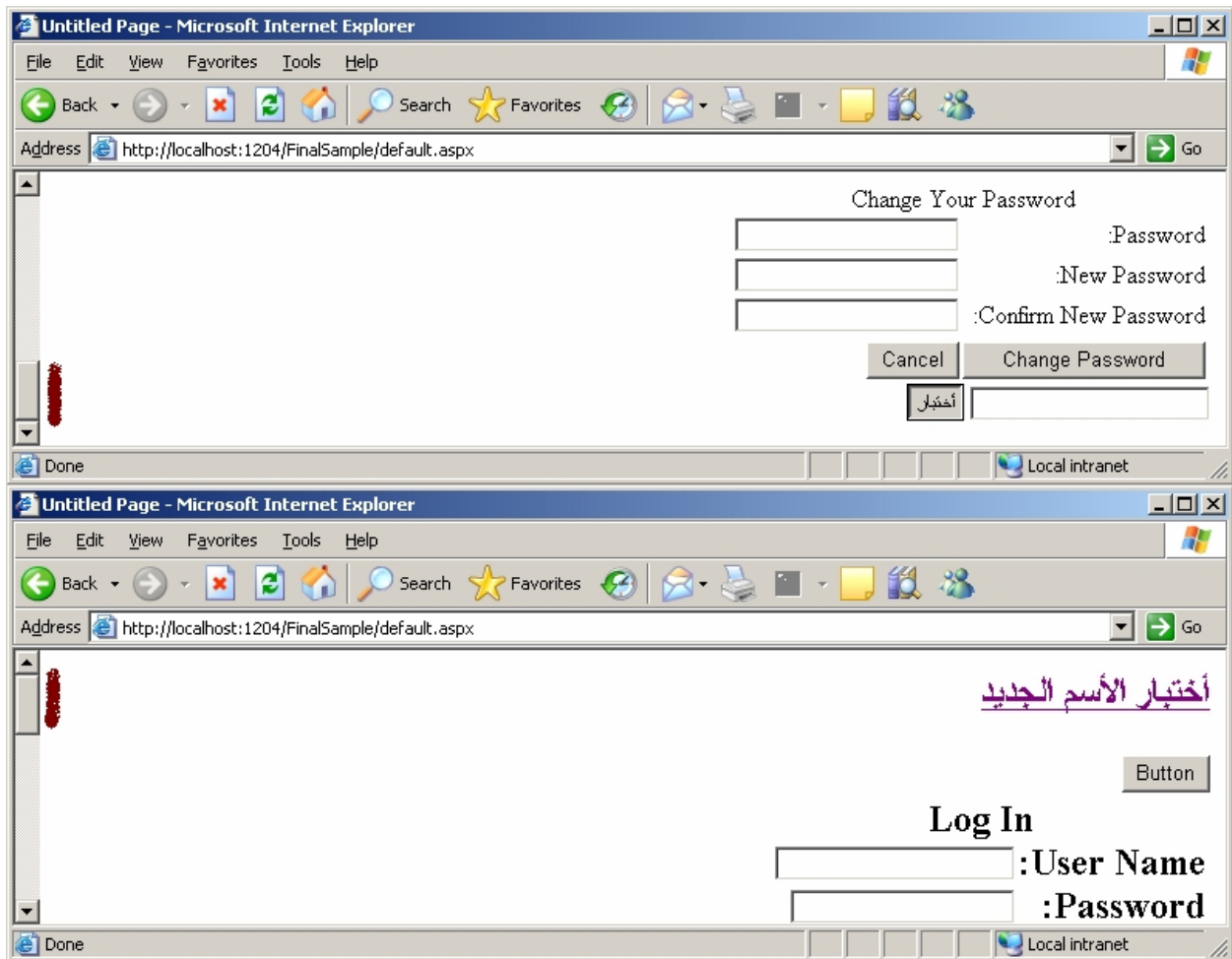


الشكل ٦٣



الشكل ٦٤ لاحظ Address هنا و الفرق بينه و بين الشكل السابق.

ميزة جديدة توفر كتابة كود جافا سكريبت هذه الخاصية هي **MaintainScrollPositionOnPostBack** هذه الميزة الجديدة تم إضافتها الى موجة الصفحة Page Directive حيث تقوم هذه الصفحة بمعالجة موقع الصفحة في حالات أن الصفحة قامت بعمل Post Back حيث كان في السابق ترجع الصفحة الى أولها كما هو في الشكل التالي



الشكل 65 كما نلاحظ من الصورة أن العمود المنزلق عاد الى أول الصفحة عند الضغط على الزر "أختبار" و لتجنب هذه المشكلة نقوم بإضافة **MaintainScrollPositionOnPostBack** الى موجة الصفحة

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

الشكل 66 بعد هذه الأضافة لن تظهر هذه المشكلة و لن تحتاج الى كتابة كود جديد ليقوم بمعالجة هذه المشكلة.

و في النهاية هذا الفصل أود أن نذكر سويا ملخصا لما تعلمناه في هذا الفصل:

١. الجديد في ASP.net في كلا من طبقات التعامل (Server Control – Page Framework – Infrastructure API)
٢. إنشاء مشروع جديد
٣. الطرق الجديدة لحفظ المشروع
٤. الأضافة الجديدة Partial Class
٥. تغيير CodeFile الى CodeBehind
٦. تعدد اللغات داخل المشروع الواحد
٧. AutoCompiation
٨. توارث واجهة التعامل User Inference Inheritance
٩. تحسين التنقيح في الأصدار الجديد
١٠. المكتبة الصفحة السابقة Previouspage
١١. الأدوات الجديدة في VS.net 2005
١٢. رفع الملفات
١٣. حفظ الأعدادات و أسترجاعها في VS.net 2005
١٤. أظهار العناوين بشكل مختلف URL Mapping
١٥. توفير كود جافا سكريبت بأستخدام MaintainScrollPositionOnPostBack