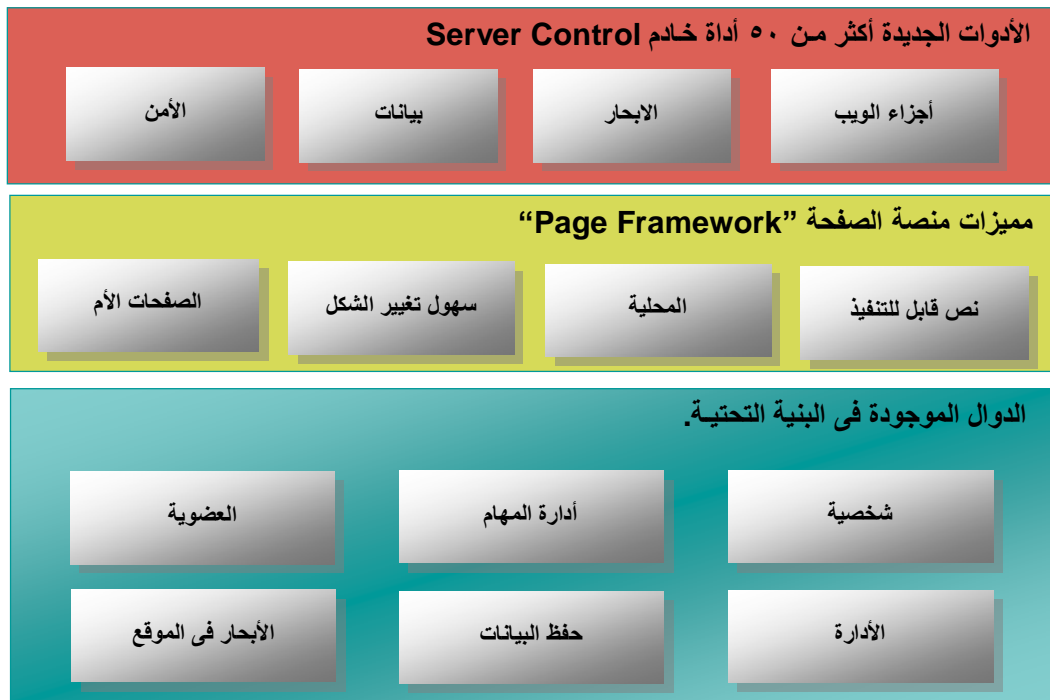


## الفصل الأول : المميزات الجديدة فى فيجوال ستوديو ٢٠٠٥.

---

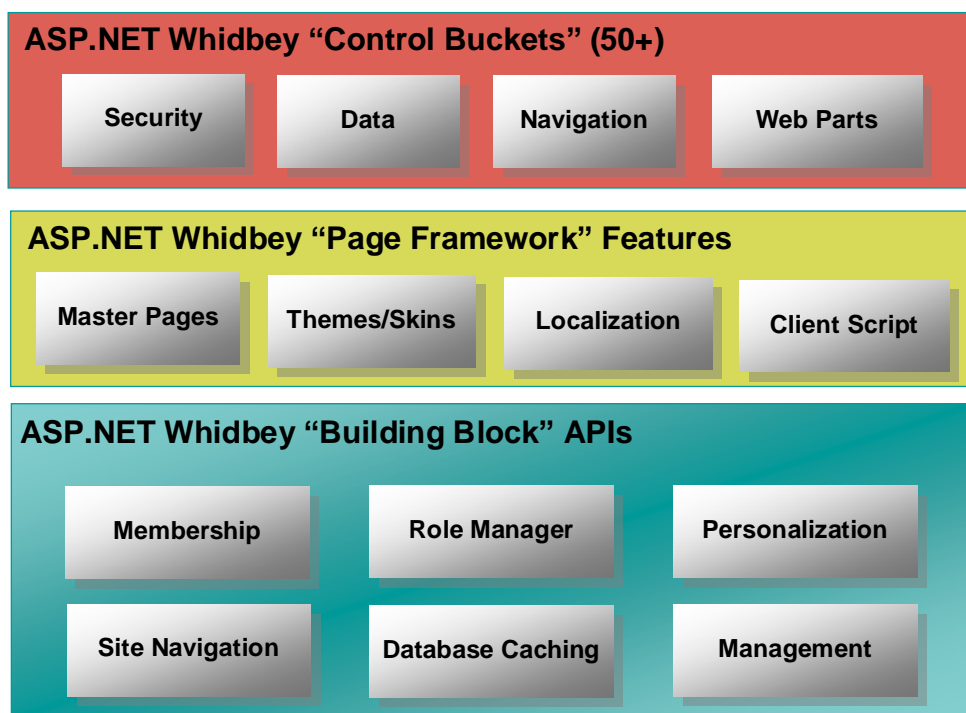
في البداية سيتم عرض رسم توضيحي لما هو جديد – بشكل عام- في الإصدار الجديدة من VS.net

## الجديد في ASP.net 2.0



الشكل ١

## ASP.NET 2.0 Developer Stack



الشكل ٢

هذه المميزات الجديدة والمكتبات والأدوات سوف توفر أكثر من ٣٠% من وقت مستخدميها – المطور- مما سيرفع من تركيزه في إنشاء حلول أقوى وأفضل .

ولابد من إيضاح أن خلف كل وحدة من هذه الوحدات الموجودة في الشكل [١] و [٢] – التي تعتبر أساس كل جديد في إصدار ٢٠٠٥ – ما يسمى بمكتبة المخصص ( Provider Class ) والتي تقوم بعمل كل ما كنت تقوم به أنت كمطور ؛ وسيتم إدراج أمثلة لشرح هذا الموضوع بالتفصيل ، وإضافة ملحق كامل خاص بالـ (Provider Class) وكيفية عملها وإنشائها

كيفية إنشاء مشروع جديد في الـ (VS.net 2005)

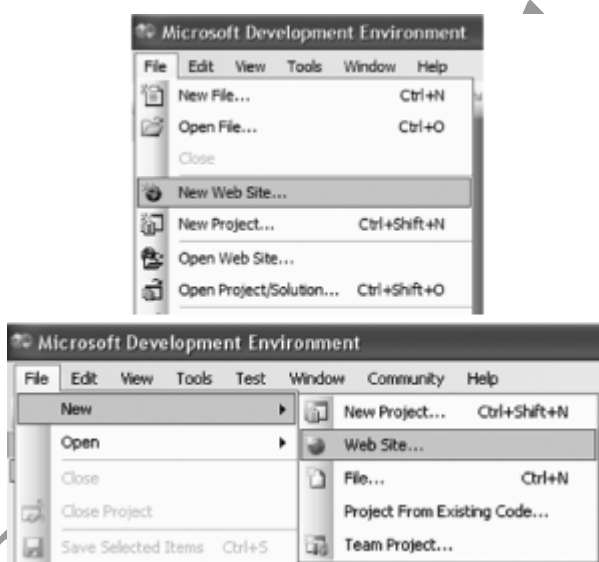
في الإصدارات السابقة للقيام ببدأ مشروع جديد كان عليك القيام بـ :

### File à New à Solution

ثم تظهر نافذة تختار منها لغة المشروع ثم نوع البرنامج الذي تود تصنيعه .

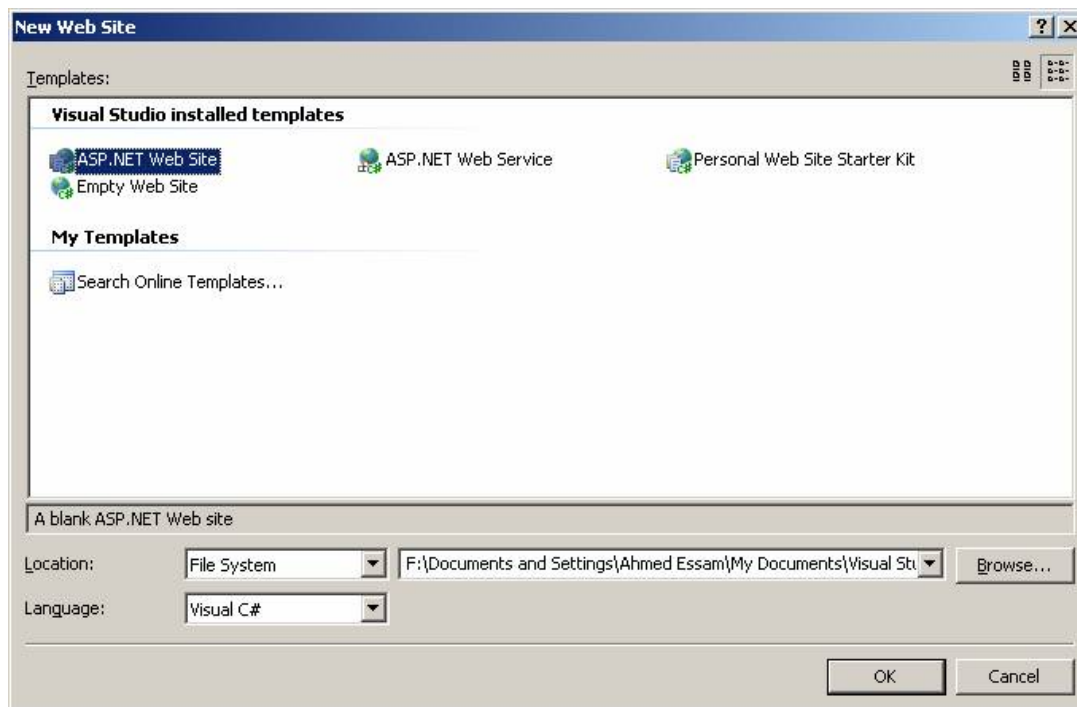
والآن – من وجهة نظري – فالوضع تحسن في فيجوال ستوديو ٢٠٠٥ عما كان عليه سابقا حيث أن كل ما عليك أن تقوم به هو :

### File à New à Web site



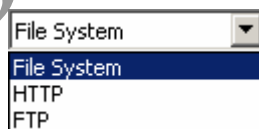
الشكل 3

وبعد ذلك تقوم باختيار نوع المشروع ولغته وطريقة الحفظ .



الشكل 4

- دعونا نفصل محتويات الشاشة السابقة فنجد بها : -
- القسم العلوي : نجد ما يسمى بالـ (Templates) قوالب ؛ ومنها تختار نوع برنامج الويب من ضمن أربعة أنواع وهي - نوع البرنامج -
- ١ - برنامج ويب عادي "موقع" (Web Site) .
- ٢ - خدمات الويب (Web Services) .
- ٣ - الاختيار الثالث (Personal web Site Starter Kit) يستخدم في حالة إن كنت تود أن ترى مثال حي .
- ٤ - الاختيار الرابع (Empty Web Site) و هو لمشروع ويب فارغ يمكنك من توجيهه حسب ما تريد .
- قسم الموقع (Location) : نجده يتكون من جزئين ؛ وسيتم إيضاح الجزء الأول لأهميته ، فهو يتكون من قائمة منزلة ومحتوياتها كالآتي : -



الشكل 5

و الآن لنشرحهم بالتفصيل

- ١ - File System
- ويتم اختياره في حالة قيامك بتجربة الـ (VS.net) حيث لا يتطلب منك وجود خادم (Internet Information Server) (IIS) ، وبهذا أمكن لمستخدمي (Windows XP Home Edition) العمل بـ (VS.net 2005) حيث إن مشاريع الويب لابد أن يتوافر بها خادم (IIS) الذي لم يكن يتوافر في النسخة المنزلية من ويندوز (win XP) . لكن ... هنا لنا وقفة .. وهي كيفية عمل مشروع الويب !! - هذا صحيح لابد من وجود خادم - وهذا ما تم توافره في الإصدار الجديد من فيجوال ستوديو حيث يوجد به خادم (Server) صغير يقوم بهذه المهمة كما هو موضح بالشكل [4] و



الشكل 6



الشكل 7

## ٢- HTTP

ويتم اختياره في حالة وجود خادم محلي (Local IIS) حيث سيتيح للمستخدمين المتواجدين على الشبكة الإطلاع على مشاريعك .

## ٣- FTP

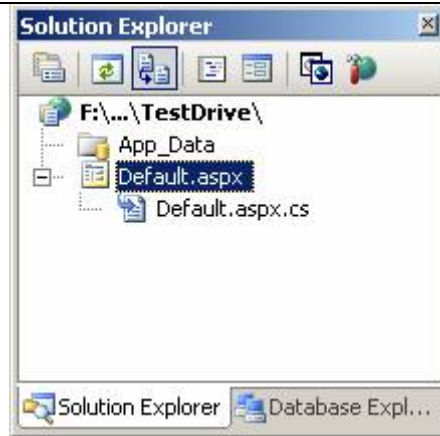
ويتم اختياره إذا ما كنت تود التعامل مباشرة مع مستضيف (Host) حيث ستمكن من خلاله التعامل مباشرة مع مشروعك في بيئة عمل فعلية وحية .

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 سيتم تناول أكثر الخواص الجديدة إثارة وتميزاً وهي : -  
**١ - تعدد اللغات في مشروع واحد .**

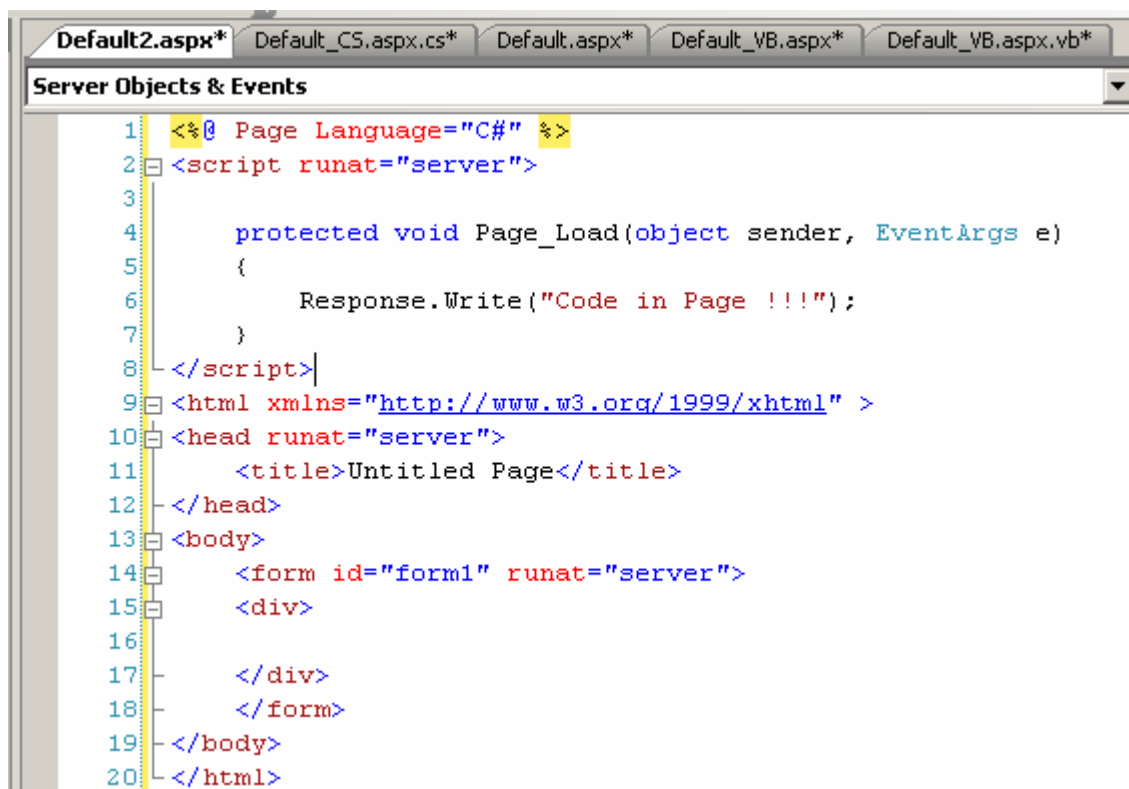
ففي الإصدارات السابقة من فيجوال ستديو كان هذا الموضوع معقد بعض الشيء ، حيث كان عليك القيام ببعض الحيل لتقوم بدمج أكثر من لغة في مشروع واحد ؛ بينما الآن في الإصدار الجديد من فيجوال ستوديو دوت نت فالأمر أبسط كثيراً ؛ فلن تشعر بوجود فرق أو وجود عدة لغات في المشروع الواحد .

الشكل ١٠

هذه الإمكانية – المتاحة في هذه الإصداره – لها ميزة وعيب في نفس الوقت : -  
 الميزة : أنه ليس على كل أفراد الفريق أن يستعملوا نفس اللغة. مما يسهل العمل في المشروعات الكبيرة.  
 العيب : يظهر في المشروعات الكبرى تحديداً في حالة الصيانة تظهر مشكلة للقائم بعملية الصيانة حيث لا بد أن يتقن اللغات المستخدمة في المشروع والذي من الممكن أن يكون غير متاح .



## ٢ - إمكانية دمج الصفحة وملف الكود



```

1 <%@ Page Language="C#" %>
2 <script runat="server">
3
4     protected void Page_Load(object sender, EventArgs e)
5     {
6         Response.Write("Code in Page !!!");
7     }
8 </script>
9 <html xmlns="http://www.w3.org/1999/xhtml" >
10 <head runat="server">
11     <title>Untitled Page</title>
12 </head>
13 <body>
14     <form id="form1" runat="server">
15     <div>
16
17     </div>
18     </form>
19 </body>
20 </html>


```

الشكل ١١

أهميتها تكمن في أنه في بعض الأحيان لا يحتاج المطور لوجود 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.

### ٣- الترجمة التلقائية لملفات الكود (AutoCompilation)

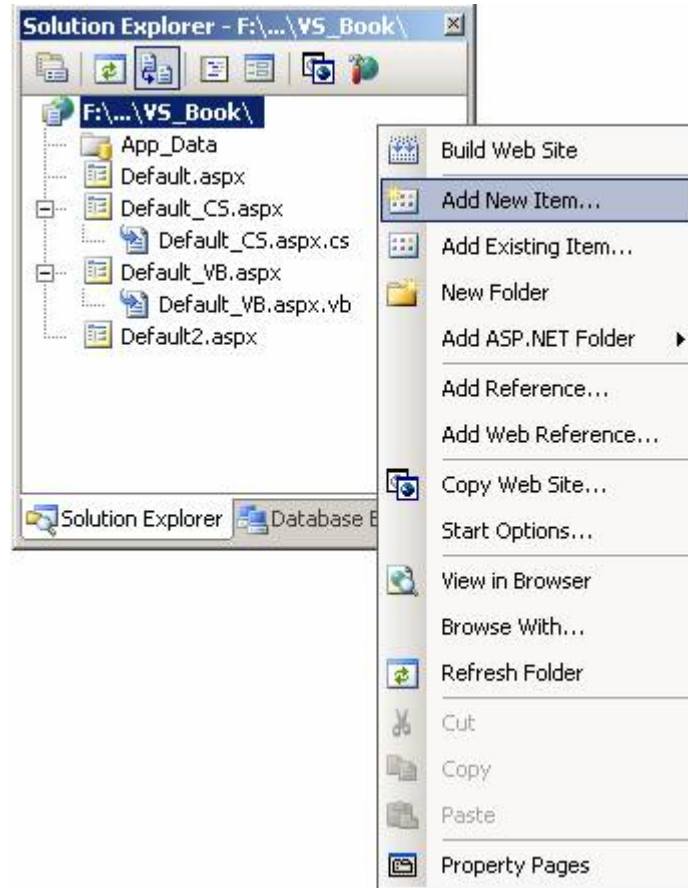
وفيه بمجرد حفظ الملف يقوم المترجم بإعادة ترجمة ملف الكود دون أنهاء أى من جلسة (Session Object) أو التطبيق (Application Object) ، هذه الخاصية مفيدة للمطورين لتوفيرها للوقت الذي كان يستهلك في إعادة الخطوات التي تقوم بإعداد كلاً من الـ (Sessions) و الـ (Application objects) .

<p>كثير من الخواص المتعلقة بالكود وإعادة الترجمة ترجع إلى سبب فني مهم جداً لابد أن يعرفه المطور وهو أن كل صفحة الآن يتم ترجمتها على حدى ؛ بمعنى أنه إذا قمت بتعديل ما يوجد في صفحة معينة من صفحات المشروع سوف يقوم المترجم التلقائي بترجمة الصفحة التي تم التعديل فيها فقط ، يمكنك الاستفادة من هذه الخاصية في المشاريع الضخمة ؛ حيث أنه في حالة الصيانة يستحيل إيقاف الخدمة ، بالإضافة إلى أنه في الإصدارات القديمة عند القيام بترجمة مشروع بشكل ما كان هذا يؤدي إلى فقط كلاً من الـ (Session) و الـ (Application) مما يسبب بعض المشاكل في بعض الأحيان.</p> <p>إذا فكرنا قليلاً ستظهر لنا الكثير من الفوائد التي تأتي باستخدام خاصية كهذه الخاصية ؛ كل ما نحتاج إليه هو بعض الوقت لتظهر "خبرة التعامل" مع هذه الخاصية الجديدة و التي سوف تسهل الكثير.</p>	
--	---

#### ٤ - خاصية الوراثة من الصفحة الأم<sup>(١)</sup>

قد يبدو الاسم غريباً ولكن لا تقلق سوف يتم شرح هذا الموضوع بالتفصيل وسوف تظهر هذه الخاصية في حالات معينة ، أولاً يجب إيضاح أن هذه الميزة من أهم المميزات التي سوف توفر الكثير من العناء على كلاً من مصمم الجرافيك للمشروع ومطوره ، ثانياً دعونا نقوم بتجربة بسيطة لهذه الميزة الجديدة فلنتبع الآتي : -  
من نافذة (Solution Explorer) قم بالضغط بالزر الأيمن للماوس و أختار (Add new item) كما هو موضح

بالشكل [12]

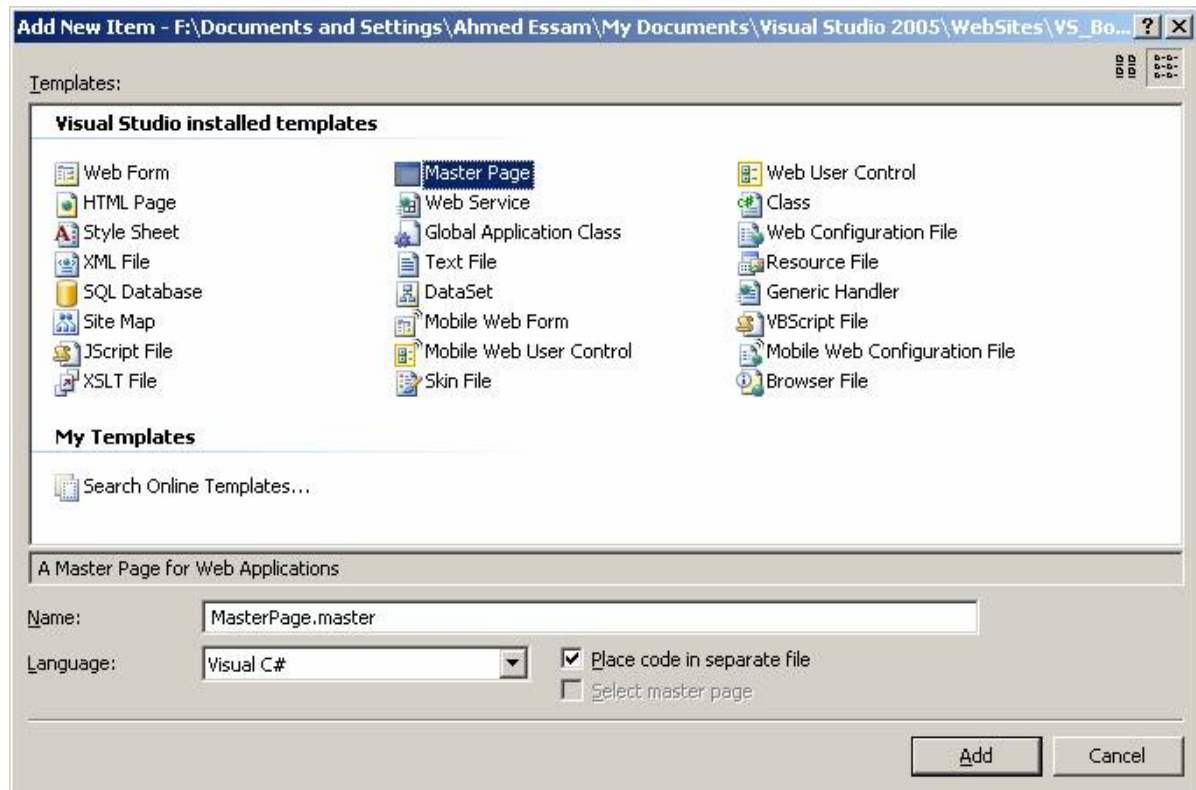


الشكل ١٢

و بعد ذلك نختار من نافذة (Add New Item) - في القسم (Templates) - لنقوم باختيار (Master Page) كما هو موضح بالشكل [13] .

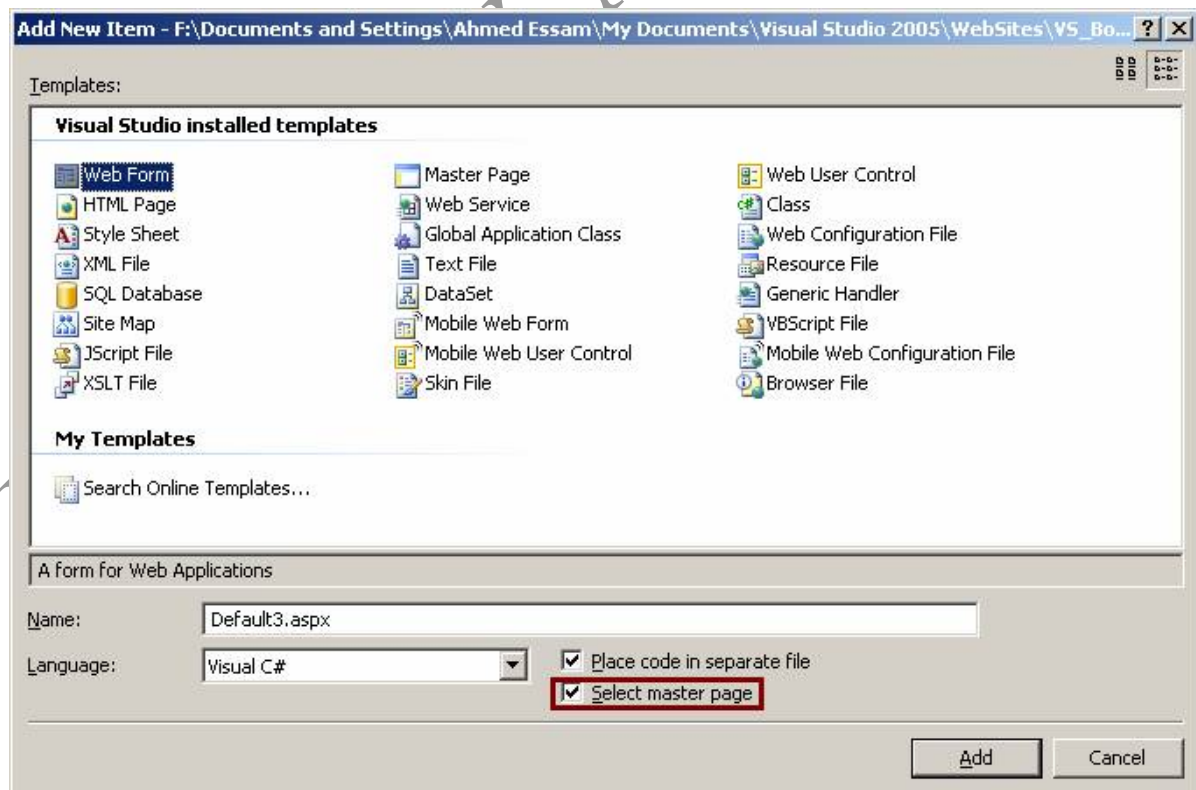
(1) سيتم تناول هذا جزء بالتفصيل في الفصل الثاني





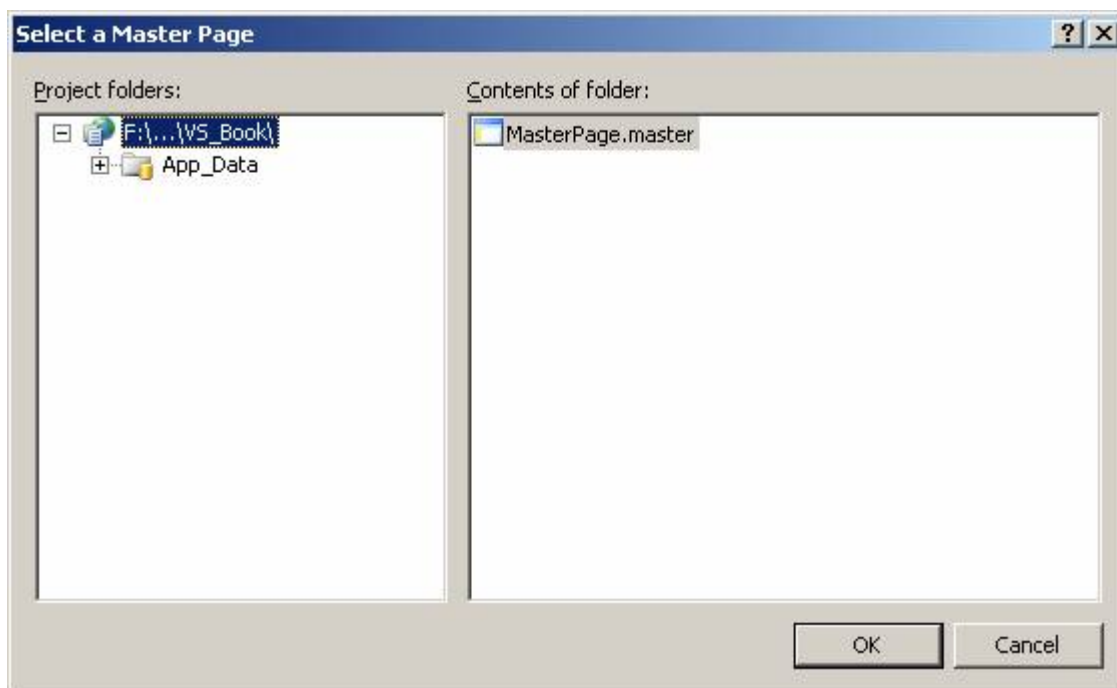
الشكل ١٣

سنقوم بإضافة عنصر جديد بنفس الطريقة السابقة ولكن بدلا من اختيار (Master Page) سوف نختار (Web Form) و سوف نقوم بتنفيذ الخيار (Master Page) كما هو موضح في الشكل [14].



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 ، وسيتم إيضاح هذا الموضوع بشكل مفصل في فصل كامل خاص به نظراً لكثرة التفاصيل به .

## ٥- المكتبة الجزئية (Partial Class) :

وهي خاصية جديدة – من وجهة نظري - و سوف توفر الكثير من الغناء التي كان يعيها المطورين في المشاريع الضخمة. والآن لنقوم بتوضيح الأمر لكي يتضح لكم مدى أهمية هذه الميزة.

في البداية ستلاحظ عند رؤية صفحة الكود التي ينتجها (VS.net 2005) أنها مختلفة عما سبق !!! .. ويرجع السبب إلى أنه تم إخفاء كل الاكواد الخاصة بإنشاء المكونات على الصفحة ؛ حيث لا توجد أهمية حقيقية لإظهار هذا الكود . وفي الشكل [16] و [17] تلاحظ ما يلي عند محاولة الدخول لكتابة كود خاص بأي حدث :

```

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

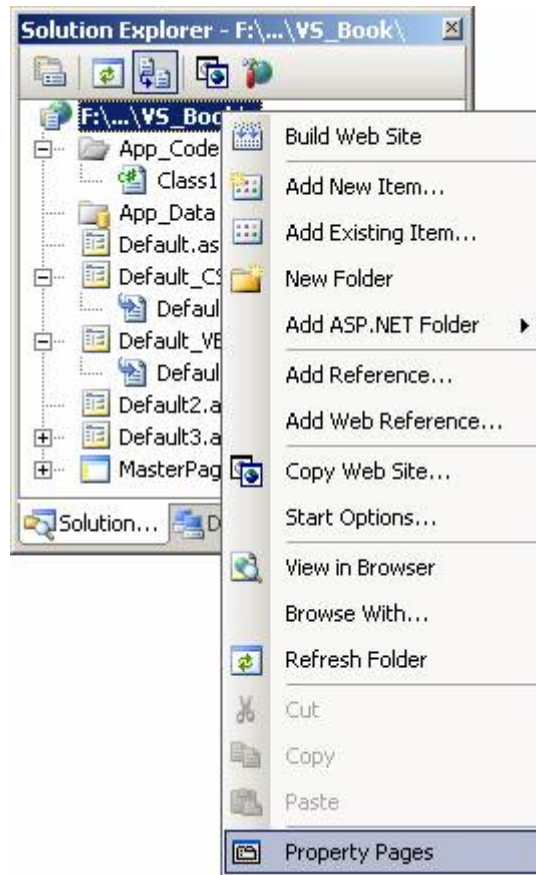
كما تلاحظ أنه تم إضافة الكلمة **Partial** الى معرف المكتبة، المعنى المقصود هنا هو أنه هذه المكتبة تعتبر في الواقع جزء من المكتبة و ليس المكتبة كلها ؛ مما يتيح عمل أكثر من مطور في مكتبة واحدة دون حدوث المشاكل المعهودة عند دمج الأكواد.

في الواقع يقوم مترجم الـ (VS.net 2005) بتجميع المكتبة و ترجمتها بالإضافة إلى أن الـ (VS.net 2005) يقوم بدمج الكود المخفي الخاص بإنشاء المكونات على الصفحة، هذه الميزة سوف تعطي مزيد من التحكم لرئيس مجموعة التطوير من ناحية التحكم في كود (Source Control) .

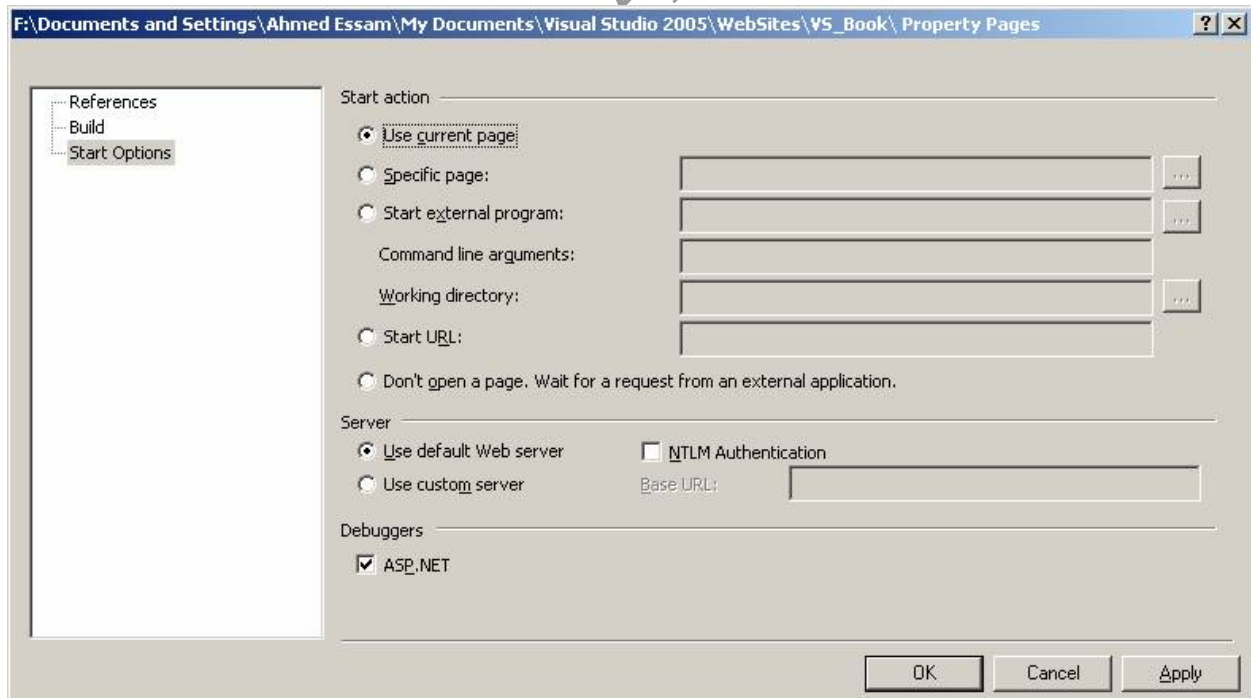
## ٦- تحسين التنقيح في الإصدار الجديد Enhancing Debugging.

كان عليك في الإصدارات السابقة- من فيجوال ستيديو موت نت- تحديد صفحة البدء (Set start page) التي سوف تظهر عند بدأ تنقيح التطبيق غالبا عند الضغط على F5 - ذلك حسب إعدادات خارطة لوحة المفاتيح - .

بينما في الإصدار الجديدة يمكنك أن تحدد صفحة البدء بالإضافة إلى إمكانية جعل الصفحة الحالية هي صفحة البدء . المقصود بالصفحة الحالية : هي الصفحة التي تعمل فيها- بالإضافة الى إمكانية جعل الصفحة المفعلة في (Solution Explorer) هي صفحة البدء، هذه الميزة أهميتها تكمن في توفير العناء على المطور عند اختبار صفحة معينة يعمل عليها حيث كان في السابق موضوع شاقا في حالة المشاريع الكبيرة، كما أنه كانت تسبب مشاكل في حالة عدم تحديد الصفحة فعليا 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 ، كما هو موضح بالشكل [18] و [19] .



الشكل ١٨



الشكل ١٩

فكما ترى في الشكل [19] يمكنك اختيار الخيار الذي يتناسب مع متطلباتك .  
**٧- ضبط تركيز العناصر (Focus of Controls) .**

أولا لابد من توضيح معنى هذه الجملة حيث أنها تبدو للوهلة الأولى غير مفهومه و الحل لتوضيح هذه الجملة هو مثل حي ؛ أظن أن جميعا استخدم محرك البحث جوجل .... هل لاحظ أحدكم أنه عند أنتهاء تحميل الصفحة يتم نقل المؤشر الى حقل نص البحث-هذا هو المقصود بضبط تركيز العناصر- ، ففي الإصدارات السابقة – فيجوال ستيديو دوت نت – كان الامر يتطلب كتابة كود على العارض "جافا سكربت" لعمل شئ مشابه ، أما الآن في الإصدار الجديد – فيجوال ستيديو كوت 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

هناك طريقتان لتحديد التركيز عند التحميل :-

١ - إستدعاء الدالة (SetFocus) من المكتبة

٢ - يتم ضبطها من خلال كود الـ HTML وتوضح في الشكل التالي .

```
<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>
```

وكما هو واضح من إحدى خواص الـ (Form) يوجد بها :

- ١ - ما يسمى بالـ (DefaultFocus) وهي مسؤولة عن ضبط التركيز و تخصيص أداة معينة عند التحميل
- ٢ - و ما يسمى بالـ (DefalutButton) وهي مفيدة جدا لكنها كانت تسبب الكثير من المشاكل حيث أنه في عندما يقوم المستخدم بملأ الإستمارة (Form) ، وعند قيامه – بعدها – بالضغط على زر الإدخال (Enter) ففي حالة إن كان هناك أكثر من زر لاد من تحديد أى منهم سيتم تنفيذه !!! . لذا نجد انه هنا- ضبط تركيز العناصر (Focus of Controls) - يتم تحديد هذه النقطة والتي سوف تريحك من عناء كتابة (Client side code) لحل المشاكل المشابهه .

#### ٧- 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 إلى تحقيق كلاً منها على حدى، بينما في الإصدار الجديد تمكن من فصل كل مجموعة من الأدوات في مجموعة تحقق خاصة .  
وسيتم إيضاح هذه الخاصية بشكل أكبر بالمثال التالي

```
<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 إتباع بعض الخطوات للاستفادة من هذه الخاصية وجعلها تعمل بشكل صحيح ؛ فيجب القيام بما يلي :-
- 1 - تعريف كل محقق (Validator) والمجموعة التي ينتمي اليها (ValidationGroup).
  - 2 - نقوم بضبط الخاصية الخاصة لجعلها تساوى نفس القيمة في الزر الذي من المفروض أن يقوم بمهمة معينة بعد التحقق من المدخلات
  - 3 - القيام بتمييز الخواص المتعلقة بهذا الموضوع في كود الـ HTML الذى سبق الإشارة إليه في أعلى هذا النص .

## ٨- إضافة جافا سكربت

في بعض الأحيان نحتاج بشدة الى توافر هذه العملية - إضافة كود جافا سكربت - للقيام بعض الأعمال عند جهاز العميل .  
في الإصدارات السابقة كانت هذه الإمكانيه متوفرة ، ولكن في الإصدار الجديدة -فيجوال ستيديو- تم تواجدها ولكن مع تخصيص مكتبة كاملة للتعامل مع الجافا سكربت .  
وفيما يلي سيتم عرض 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 ، والآن سوف أعرض مثال آخر يوضح 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

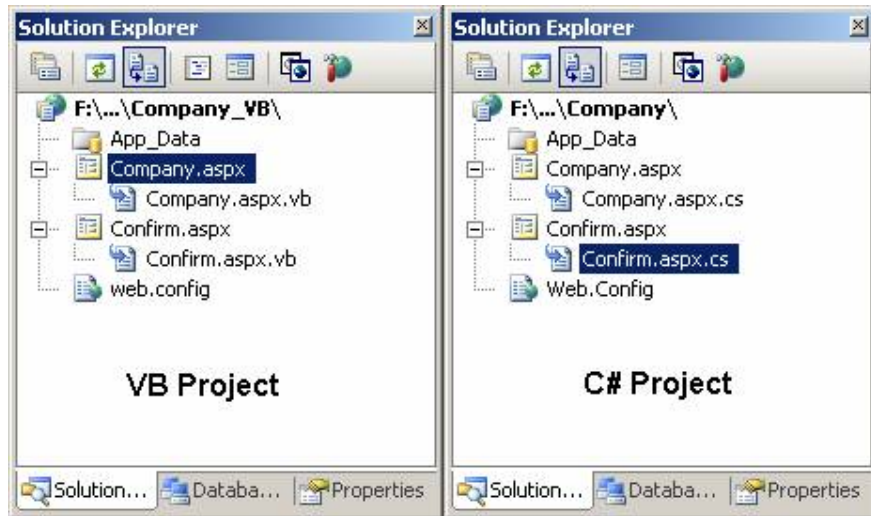
كما لاحظنا - بالتأكد - أنه من السهل جدا تضمين ملف جافا سكربت بدون مشاكل ، وأيضا بالمكتبة الجديدة التي تم إضافتها للعديد من مكتبات الدوت نت ٢٠٠٥ يمكن إستكشاف الكثير من التكامل بين كود (C#) أو (Visual Basic) والجافا سكربت ، كل ما عليكم هو البحث في الـ MSDN وسوف تجد الكثير من المواضيع التي سوف تكون مفيدة.

#### ٩ - ميزة أخرى جديدة (PreviousPage).

فكرة هذه الميزة تعتمد على ربط الصفحات عن طريق مكتبة جديدة أسمها الصفحة السابقة و التي تحتاج الى تعريف لى الصفحة المستقبلية .

وليتم إيضاح هذه الميزة وكيفية الإستفادة منها سيتم عرض مشكلة وكيفية حلها .

المشكلة :- إذا فرضنا أن لدينا شركة و نريد إدخال كلا من أسمها و عنوانها وبعد ذلك نريد تأكيد البيانات .  
الحل :- لاحظ الأشكال التالية [20] و [21] و [22] .



الشكل ٢٠



الشكل 21 (صفحة إدخال بيانات الشركة)



الشكل 22 (صفحة تأكيد البيانات)

أود ذكر شئ صغير وهو أنه في كود الـ (HTML) لا يوجد اختلاف بين صفحة مسجل فيها بواسطة لغة الـ (Visual Basic) أو الـ (C#).



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



بكل بساطة سنقوم بتحديد الصفحة التي سيتم 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 أضافة 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)

دعونا الآن نقوم بتوضيح بعض الأمور:-

- نقوم بضبط أعدادات الكائن (PreviousPage) و التي تحتوى جميع أعضاء الموجودة

- بالصفحة السابقة لذلك قمنا بعمل خاصيتين للقرأ 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

التالى :-

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

وبهذا نكون عرفنا الصفحة السابقة للصفحة الحالية ، و الآن يمكننا استخدام مكونات الصفحة السابقة في الصفحة الحالية

كما هو الحال في كود بدء التحميل في صفحة (Confirm.aspx) والشكل [23] و [24] 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 كثيرا عند استكشافك لهذا الموضوع في فيجوال ستوديو دوت نت ٢٠٠٥ .

## ١٠ - تعدد منظر العرض Multiple view for controls groups

هذه الميزة حقا رائعة ، فهي توفر الكثير من الوقت لكثير من المطورين ، يبدو وصف هذه الميزة غامضا لحد ما لذا تم الاستعانة بمثال ليوضح المعنى لهذه الميزة بشكل موضح .

المثال — :- هو أنك تريد عمل مساعد (Wizard) يقوم بعمل تسجيل لعضو فيجب القيام الخطوات التالية و هي:-

١- عرض رخصة التعامل مع موقع من قبل الشركة المنتجة للموقع .

٢- أقرار بأن السن تجاوز ١٥ سنة .

٣- إدخال الاسم الحركي و البريد الالكتروني .

٤- أدخل بيانات الاسم و العنوان .

٥- شكر للعضو على تسجيله في الموقع .

ويتضح هذا في الشكل رقم [28]

في الإصدارات السابقة للقيام بعمل مثال كهذا هناك أكثر من طريقة :-

- عمل أكثر من لوحة (Panel) و توضع عليها الأدوات ثم نقوم بعمل إظهار و إخفاء للوحة

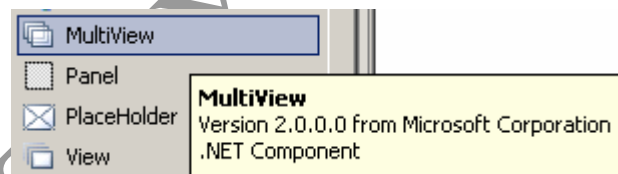
(Show and Hide)

- عمل ٦ صفحات تقوم بالمهمة .

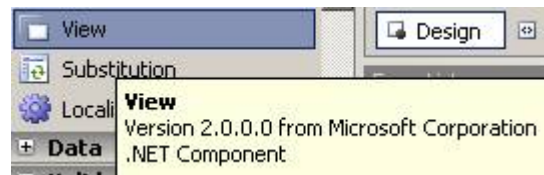
الآن يختلف الوضع - في فيجوال ستوديو ٢٠٠٥ - بوجود أداة جديدة أسمها (MultiView) فهي تحتوى على مجموعة

من عنصر يسمى (View) الذى يحتوى على مجموعة من الأدوات ، و من الأداة (MultiView) يتم تحديد الـ (View)

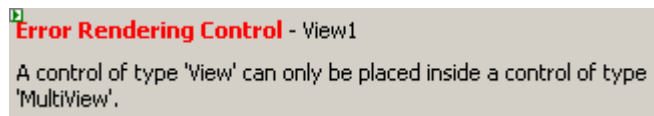
النشط بحيث يظهر و يخفي باقى الـ (View).



الشكل ٢٥ (أختيار العنصر MultiView)



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



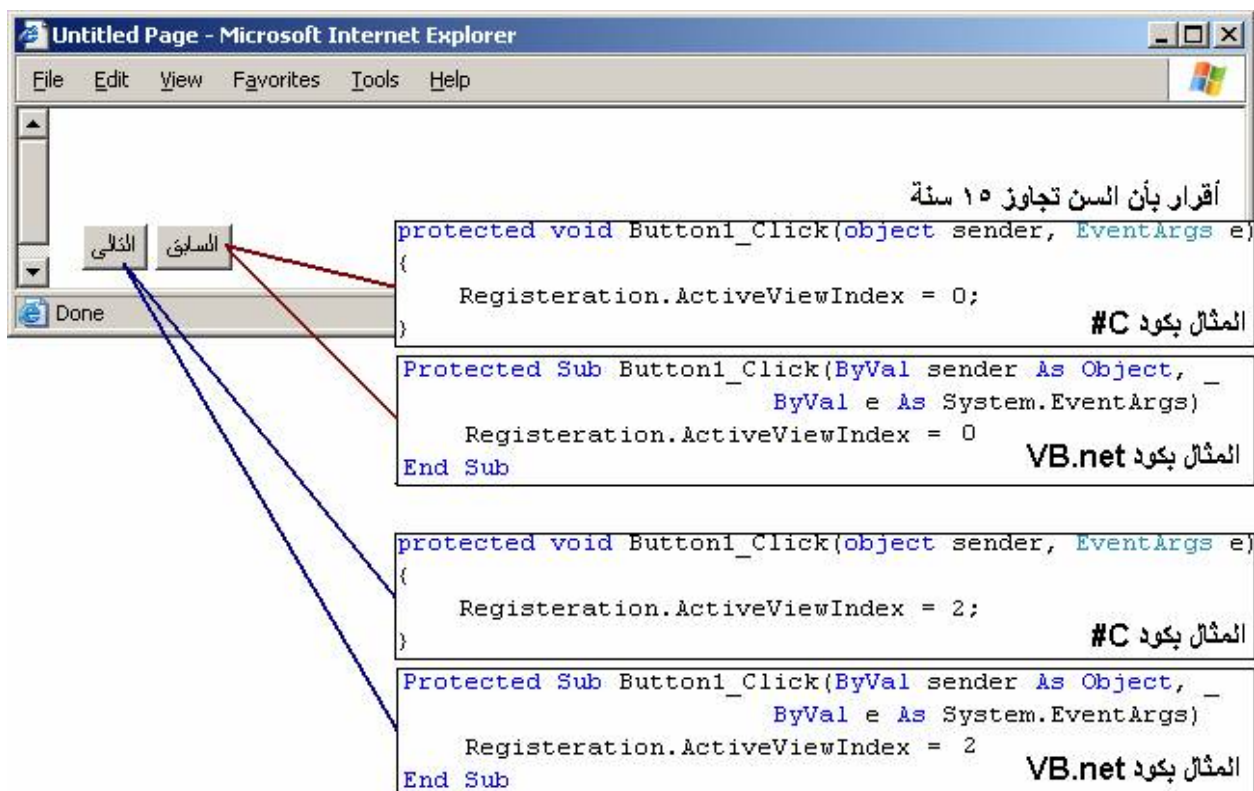
الشكل 27 [ عند محاولة وضع الـ (View) خارج الـ (MultiView) تظهر رسالة خطأ ]



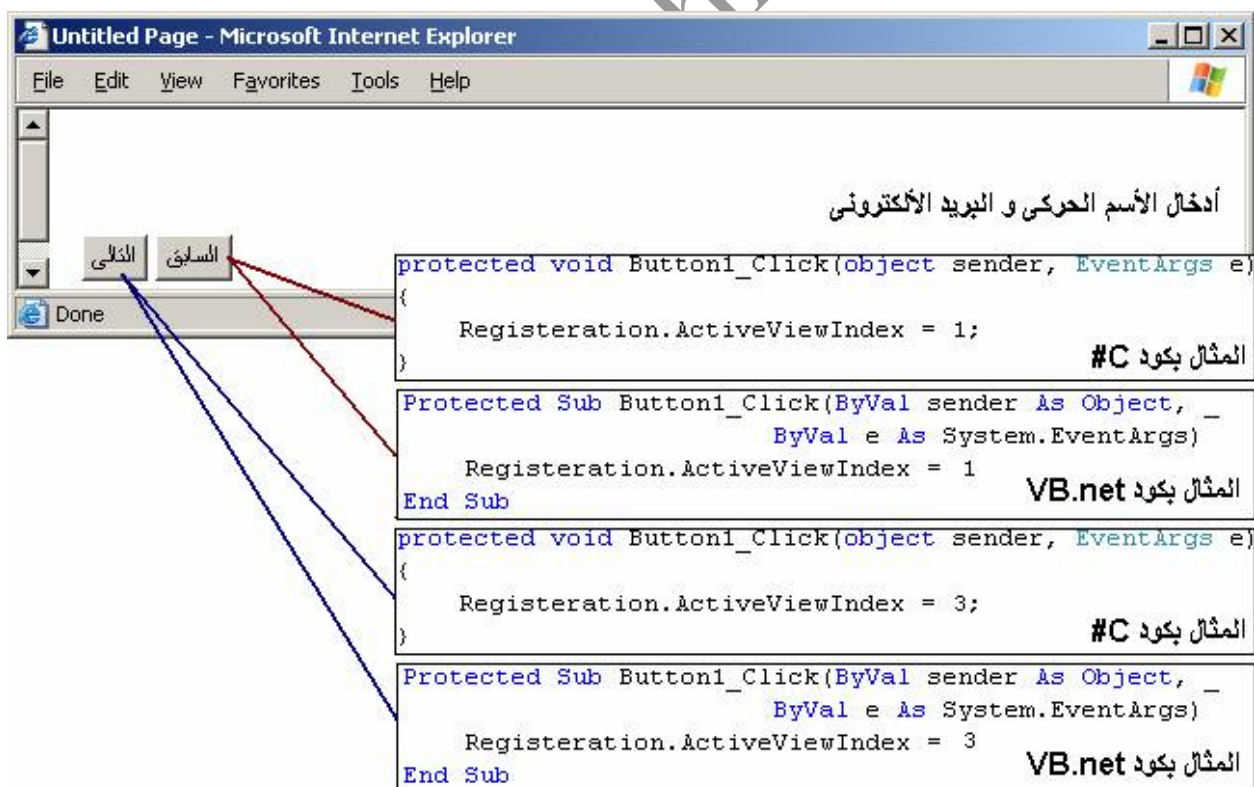
الشكل 28 [هنا يتم تحديد مجموعات الأدوات في الـ (View) و يجب هنا ملاحظة ترتيب العناصر لان أساس التعامل مع العنصر (MultiView) هو الترتيب (View Index) سوف يتم توضيح هذه النقطة في شرح الكود]



الشكل 29 [المنظر الأول (viw\_License)]

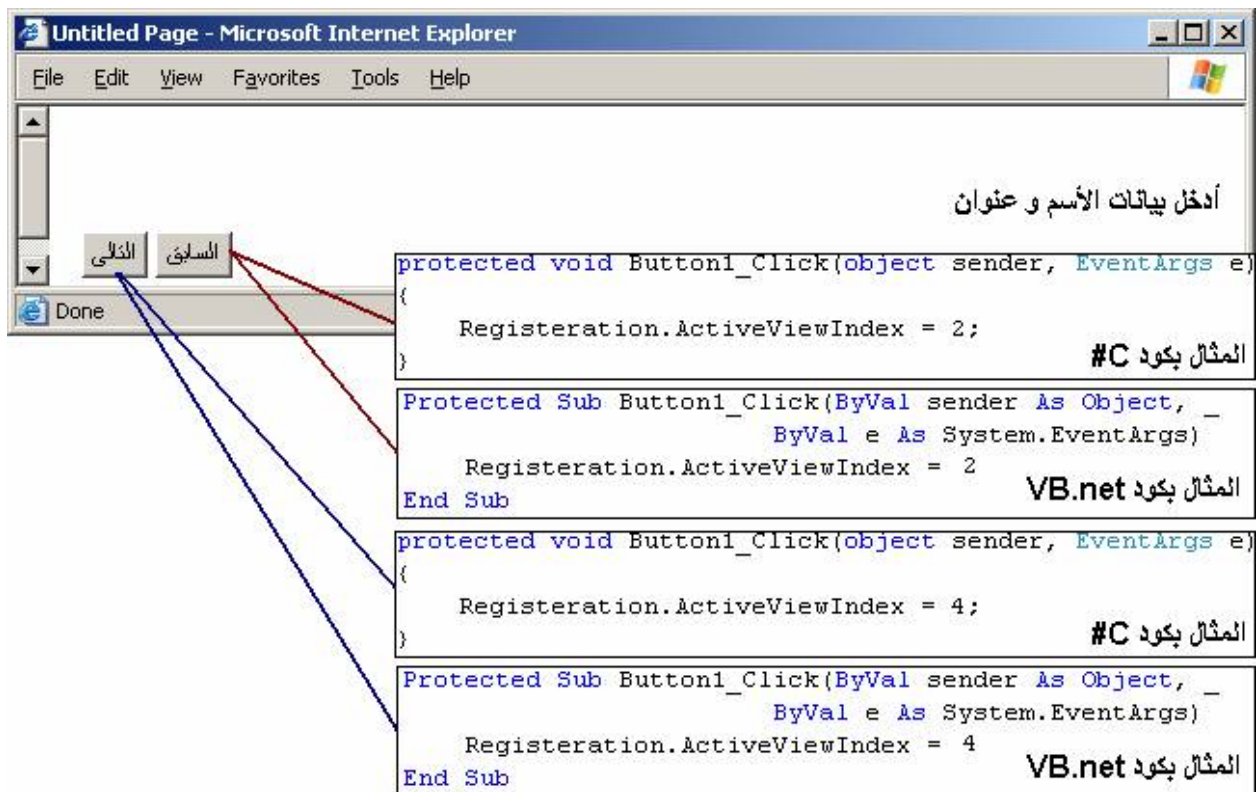


الشكل 30 [المنظر الثاني (view\_Admition)]



الشكل 31 [المنظر الثالث (view\_MainInfo)]





الشكل 32 د المنظر الرابع (( view\_Info))



الشكل 33 المنظر الخامس (( view\_Thanks))

كما نرى أن هذه الاداة وفرة الكثير من الوقت و العناء 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 وضع بداخله مجموعه من الView حسب ما تريد ثم وضع في كل View مجموعة العناصر التي تؤدي الغرض المراد تنفيذه.

## ١٠ - المساعد (Wizard):

هذه الخاصية هامة جداً- من وجهة نظري- حيث أنها تقوم بنفس عمل الاداة السابقة - (Multiple view) - و لكن بشكل مختلف حيث أنها تقوم بوضعه ؛ و من ثم تقوم بتحديد الخطوات المطلوبة ؛ ثم تجدها تقوم بكل شيء . معنى هذا أنك لست مضطر إلى كتابة أى كود لإتمام 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 ]

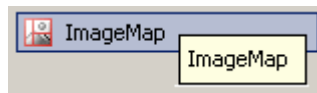


الشكل 37 [ جانب من المثال بعد التنفيذ ]

هناك الكثير لتتحدث 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.

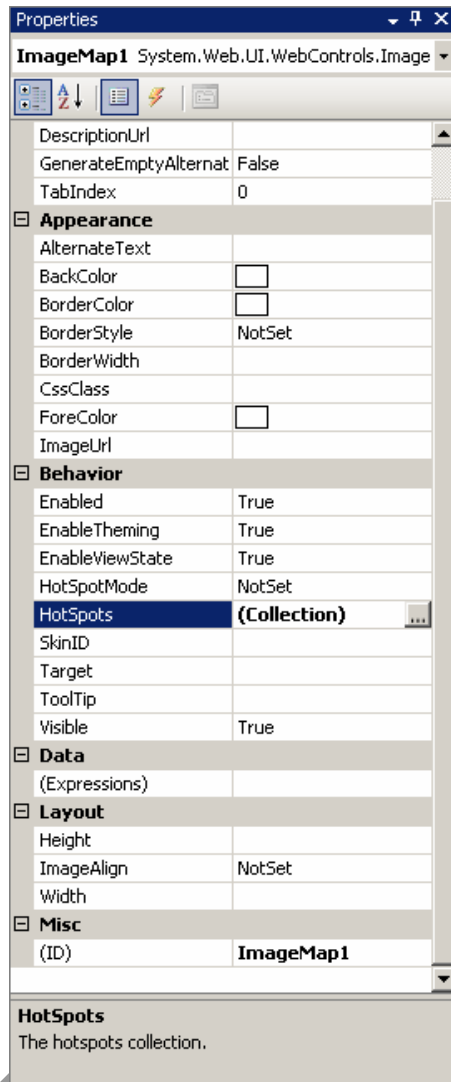
## ١١ - الأداة خارطة الصور (Image Map):

هي أداة جديدة تم تزويدها في الإصدار الجديد - الفيجوال ستوديو - ، فهي توفر عليك عناء التعامل مع البرامج التي تقوم بالعمل المشابه ، كما يمكن جعل هذه الأداة تتفاعل مع بقية العناصر الموجودة على الصفحة . و الآن سيتم عرض كيفية التعامل مع هذه الأداة الجديدة .

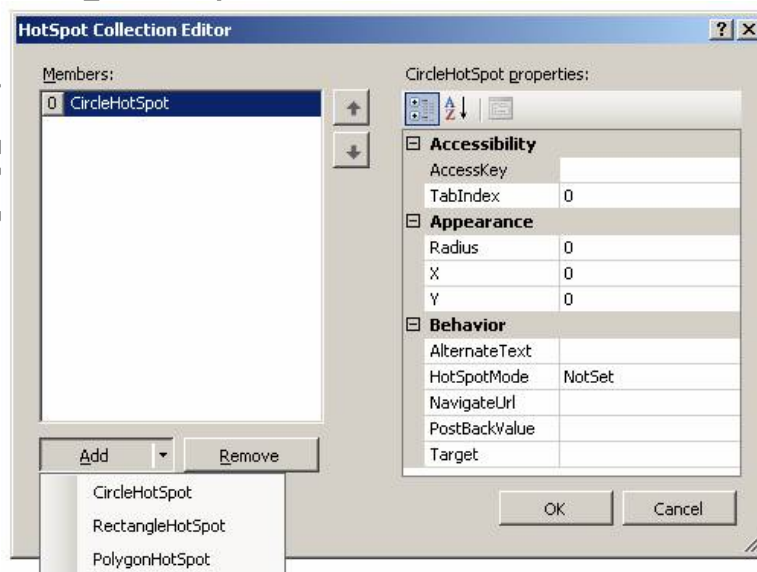


الشكل ٣٨ قم باختيار الأداة من التبويب (Standard)

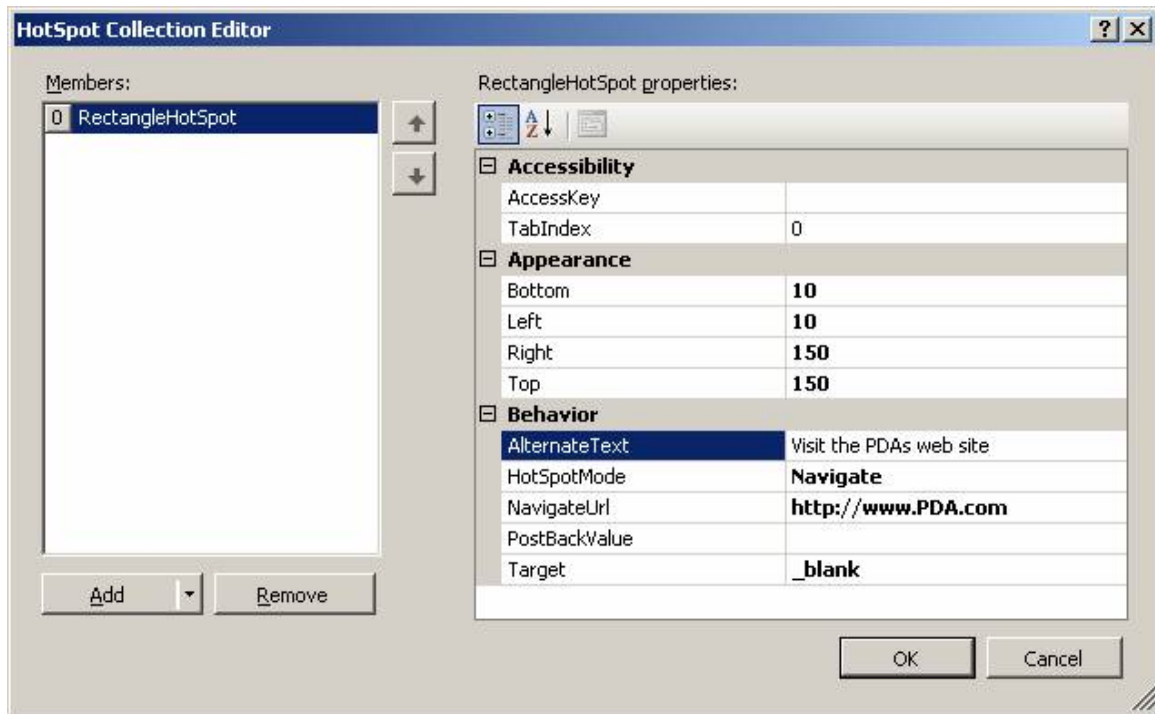




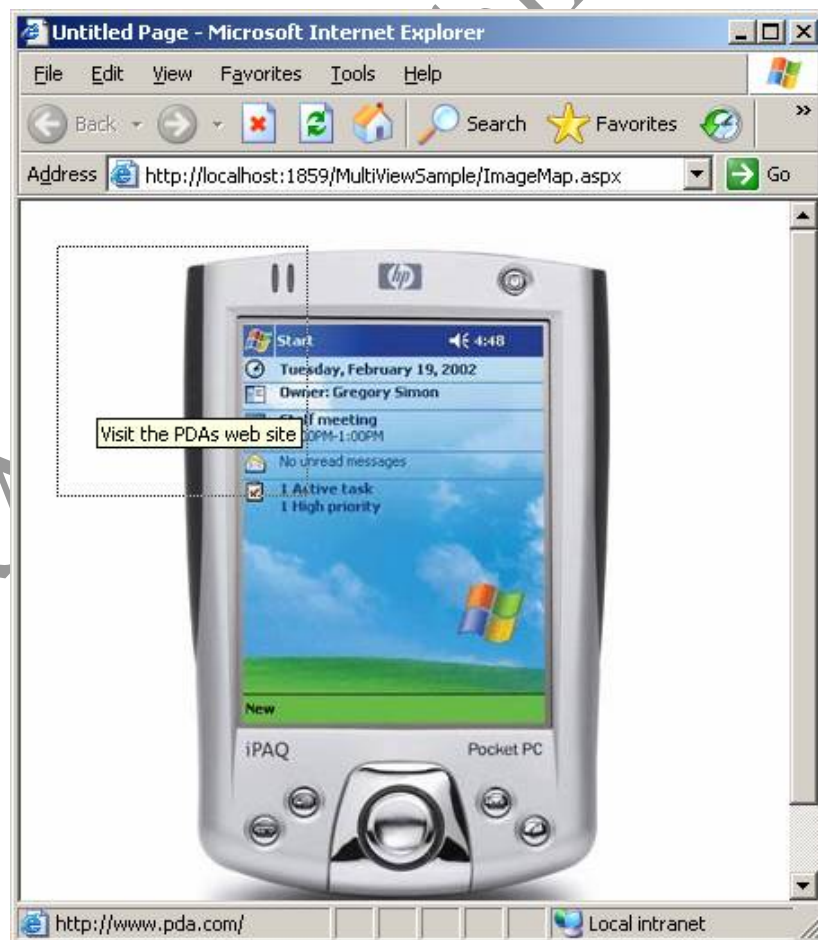
الشكل 39 عليك بعد ذلك تحديد أهم خاصية و هي (HotSpots) و هي الرسومات الغير المرئية و التي حين يمر عليها المستخدم تظهر كوصلات



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



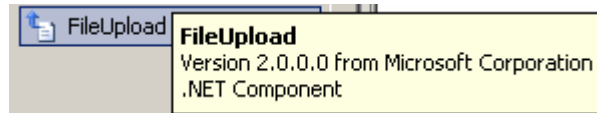
الشكل 41 يعد ذلك قم بتحديد خواص الشكل كالطول والعرض والوصلة المراد الذهاب إليها ، عند ضغط المستخدم على الشكل بالإضافة إلى مكان عرض الوصلة هل يتم ذلك في نافذة جديدة أم في نفس النافذة



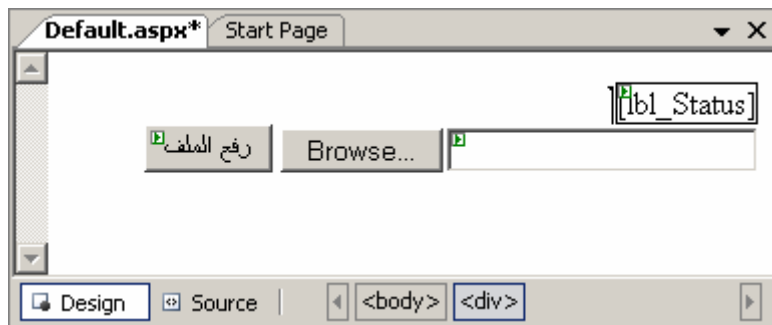
الشكل ٤٢ و كما نرى في المثال أننا قمنا بوضع صورة و بمجرد المرور على الشكل الخفى تظهر الوصلة.

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

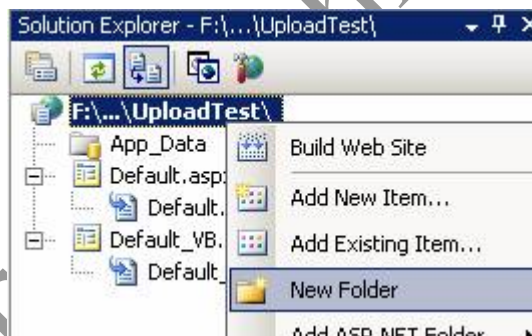
## ١٢- أداة رفع الملفات (FileUpload):



الشكل 43 قم بأختيار الأداة من نافذة الأدوات من التبويب (Standard)



الشكل 44 قم بعمل الشكل التالي بأأراج (Button) و (Label) للصفحة



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



الشكل 46 قم بتغيير أسمى حسب ما تريد

```
protected void btn_Upload_Click(object sender, EventArgs e)
{
    هنا نقوم بتعرف متغير من نوع سلسلة حرفية ؛ حيث سيضع فيه المسار الحقيقي للملف .
    string strRealPath;
```

هنا نقوم بأخذ المسار الحقيقي من العنصر (Request) و نضعة في المتغير الذي قمنا بحجزه

```
strRealPath = Request.PhysicalApplicationPath;
```

هنا نقوم بأضافة المجلد الذي قمنا بأنشاءه إلى السلسلة الحرفية

```
strRealPath += "Uploads\\";
```

نقوم باختبار ما إذا كان المستخدم قام باختبار ملف أم لم يتم بالإختيار

```
if (fu_Uploader.HasFile)
{
```

في حالة إذا ما كان المستخدم اختار ملف نقوم باختبار هل مجلد الملفات موجود على الخادم أم لا يوجد

```
if (System.IO.Directory.Exists(strRealPath))
{
```

في حالة إذا ما كان المجلد موجود نقوم باختبار هل المجلد يحتوي على نفس الملف أم لا

```
if (!System.IO.File.Exists(strRealPath + fu_Uploader.FileName))
{
```

في حالة إذا ما كان الملف غير موجود نقوم باختبار الملف هل حجم الملف أقل من 15000 حرف أم لا

```
if (fu_Uploader.PostedFile.ContentLength < 15000)
{
```

في حالة إذا ما كان 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 إذا ما كان الملف أكبر من 15000 حرف تظهر للمستخدم رسالة تخبره بهذا

```
lbl_Status.Text = "حرف 15000 من أكبر الملف حجم";
```

```
}
```

```
}
```

```
else
```

```
{
```

في حالة وجود ملف على الخادم بنفس الاسم يتم اخبار المستخدم.

قد يتسأل الكثيرون 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";
```

```
};
```

```

    }
}
else
{
    في حالة إذا ما قام المستخدم بعدم اختيار ملف للرفع
    lbl_Status.Text = "ملف أختَر 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 بتعريف متغير من نوع سلسلة حرفية حيث سوف نضع فيه المسار الحقيقي للملف
    Dim strRealPath As String
    هنا نقوم بأخذ المسار الحقيقي من العنصر (Request) ونضعه في المتغير الذي قمنا بحجزة
    strRealPath = Request.PhysicalApplicationPath
    هنا نقوم بإضافة المجلد الذي قمنا بإنشاءه الى السلسلة الحرفية
    strRealPath += "Uploads\"
    نقوم باختبار ما إذا كان قام المستخدم باختيار ملف أم لا
    If fu_Uploader.HasFile Then
        في حالة إذا ما كان المستخدم اختار ملف نقوم باختبار هل مجلد الملفات موجود على الخادم أم لا يوجد
        If System.IO.Directory.Exists(strRealPath) Then
            في حالة إذا ما كان المجلد موجود نقوم باختبار هل المجلد يحتوي على نفس الملف أم لا
            If Not System.IO.File.Exists(strRealPath) Then
                في حالة إذا ما كان الملف غير موجود نقوم باختبار الملف هل حجم الملف أقل من 15000 حرف أم لا
                If fu_Uploader.PostedFile.ContentLength < 15000 Then
                    في حالة إذا ما كان 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 إذا ما كان الملف أكبر من 15000 حرف تظهر رسالة للمستخدم تخبره بهذا
                    lbl_Status.Text = "حرف 15000 من أكبر الملف حجم"
                End If
            Else

```

```

في حالة وجود ملف على الخادم بنفس الاسم يتم اخبار المستخدم بهذا
وقد يتسأل الكثيرون 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"
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 اختيار 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 اختيار 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 اختاره 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

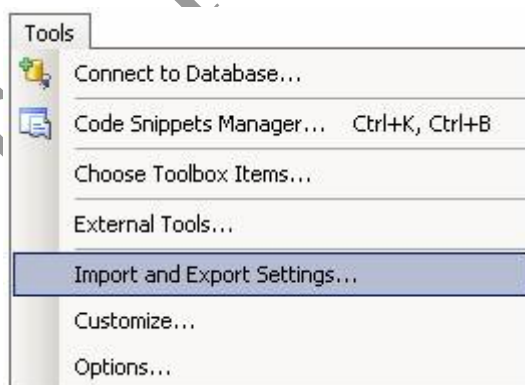


الشكل 51 في حالة مطابقة كل الشروط و عم حدوث مشاكل و رفع الملف بشكل سليم.

هناك من يرى أن الموضوع معقد جدا لقيامنا 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 ، هذا بالإضافة إلى أن معظم المخترقين ينتظرون أخطاء الآخرين فمن الممكن أن تكون إحدى أخطاءك مدخل لأحد المخترقين و التى من الممكن ثوصله إلى أجزاء لا يسمح له بدخولها.

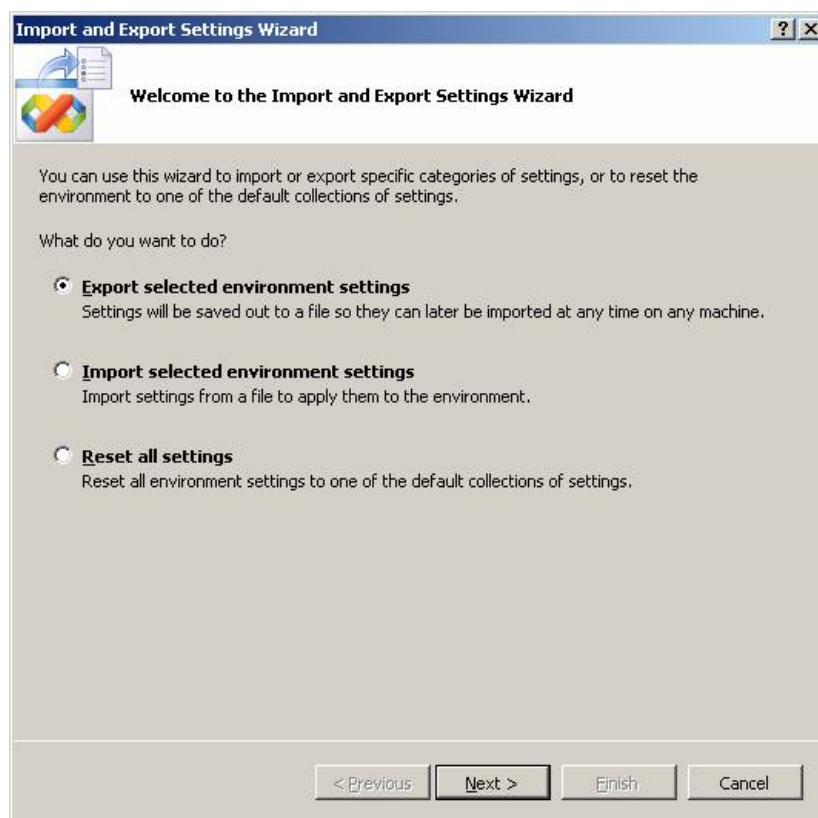
### ١٣ - حفظ إعدادات فيجوال ستوديو ٢٠٠٥ :

فى الإصدارات السابقة إن تم تغيير جهاز كنت تحتاج إلى أن تقوم بإعدادات فيجوال ستوديو مرة أخرى لتعمل عليه كما اعتدت من قبلما كان يضع الكثير من الوقت لأن هذه العملية عملية روتينية لا يوجد فيها أى جديد .  
و الآن بالإصدار الجديدة - فيجوال ستوديو ٢٠٠٥ - تم حل هذه المشكلة عن طريق إستيراد و تصدير و حفظ إعدادات فيجوال ستوديو و فى أقل من ثوانى .

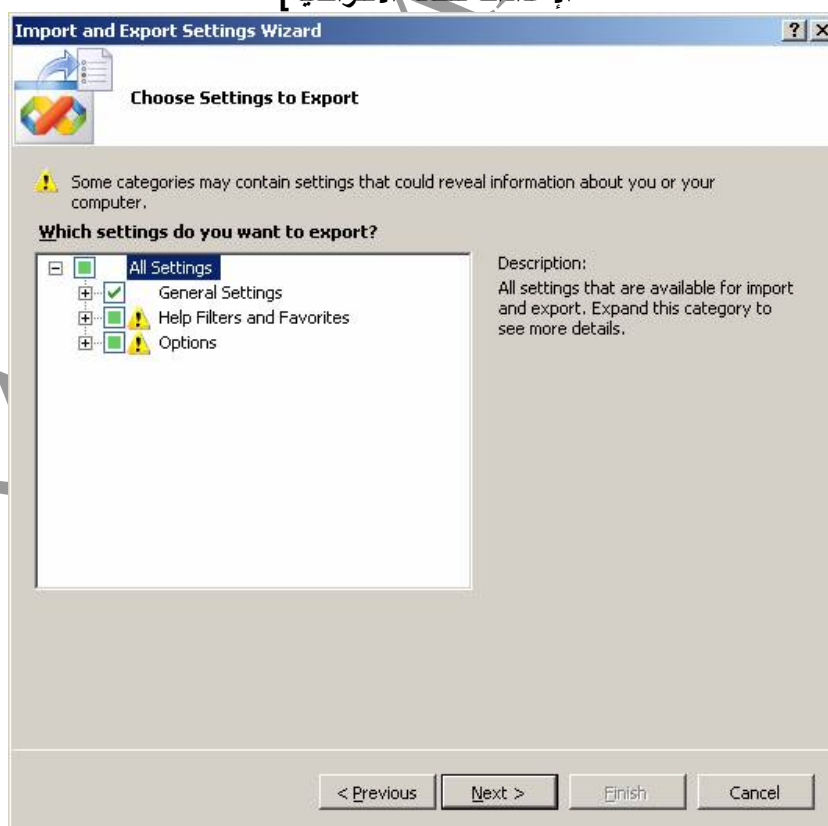


الشكل 52 [أولا للتعامل مع هذه الميزة الجديدة كل ما عليك هو الدخول على القائمة (Tools) و من ثم اختيار (Import and Export Settings) و بعدها سيظهر لك معالج إعدادات فيجوال ستوديو.

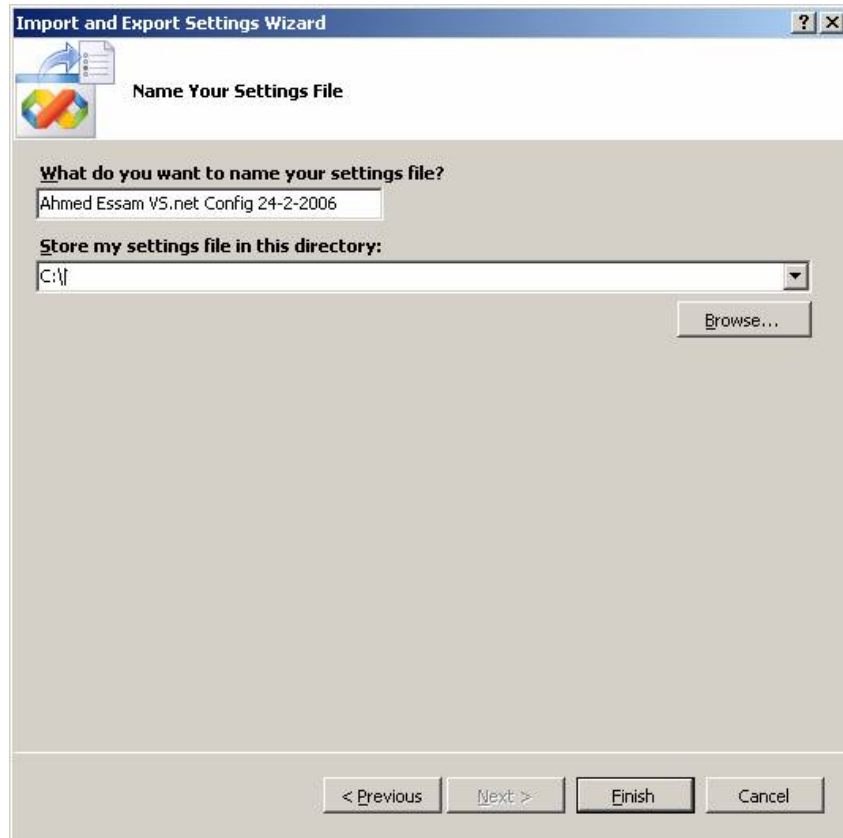




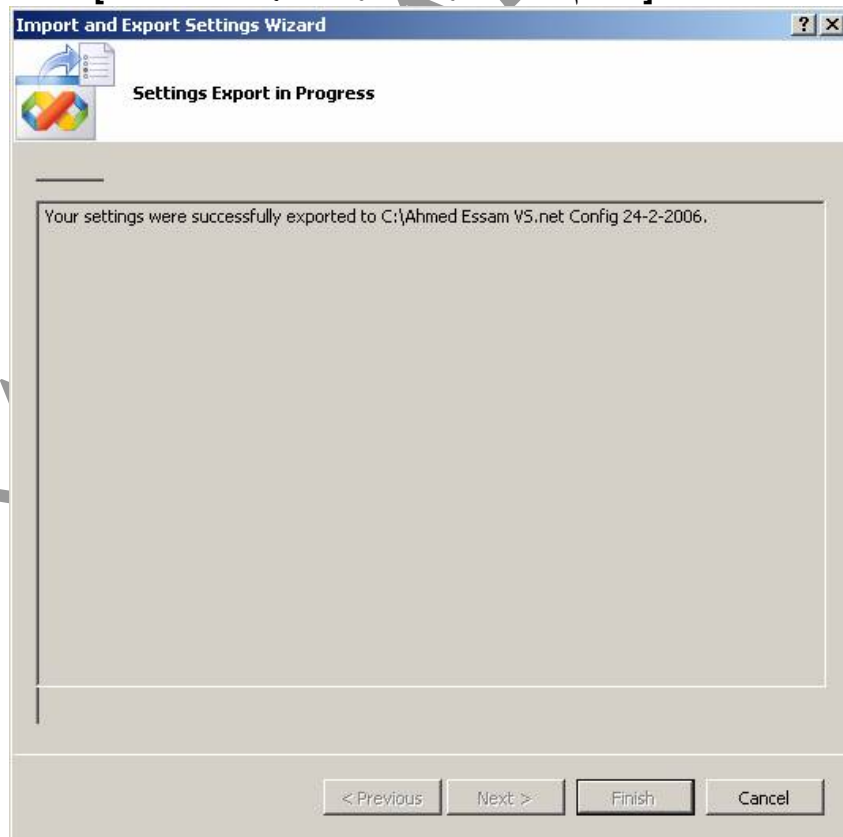
الشكل 53 ] فى هذه النافذة سوف يسألك ... ١- هل تريد إستيراد إعدادات ٢- حفظ ملف إعدادات ٣- إنك تريد إعادة الإعدادات للحالة الافتراضية



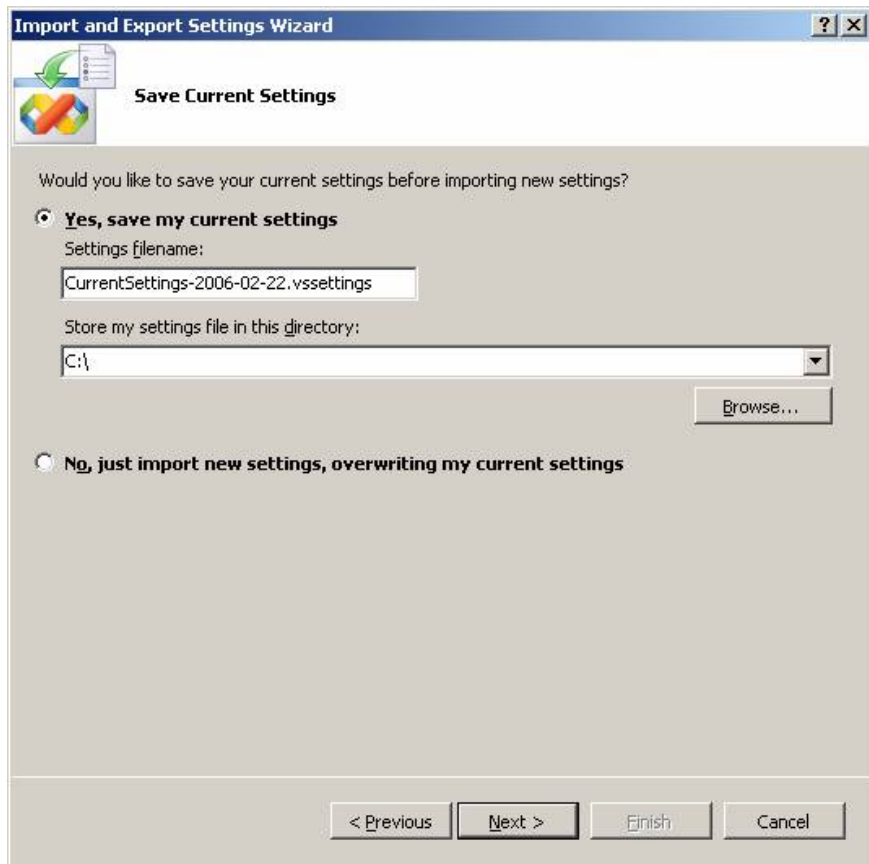
الشكل 54 ] قمنا باختيار حفظ ملف إعدادات ؛ و من هنا يتم إختيار الإعدادات التى تريد حفظها ، و تشير علامة المثلث الأصفر الى أن هناك إعدادات تحتوى على بيانات شخصية



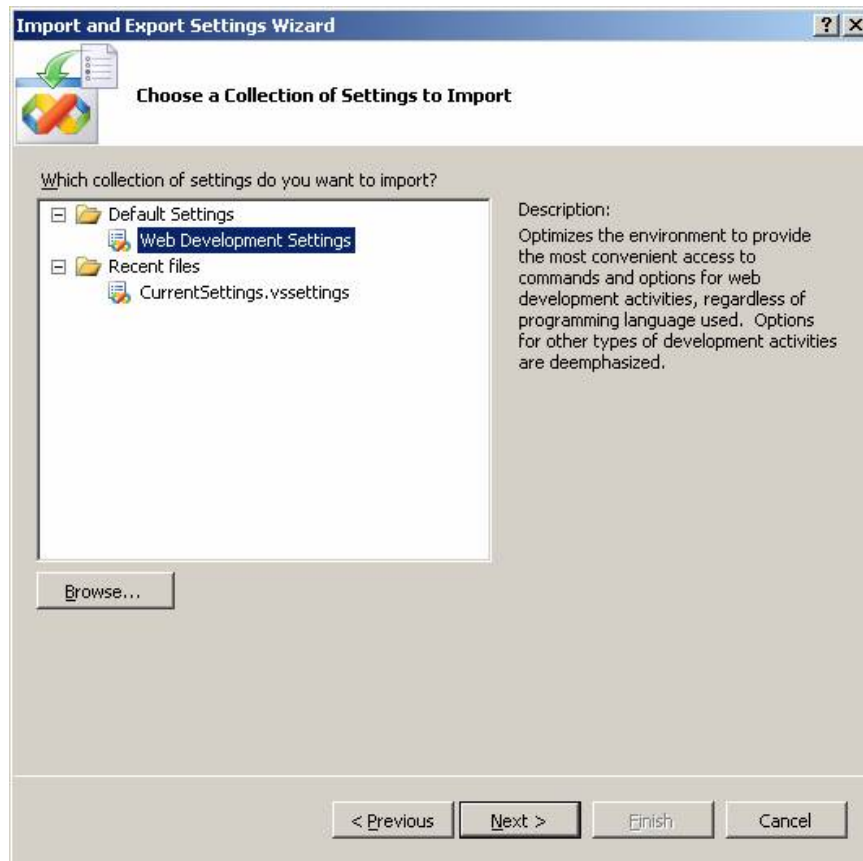
الشكل 55 [هنا يتم تحديد مسار الملف واسمه تهيئة لعملية الحفظ]



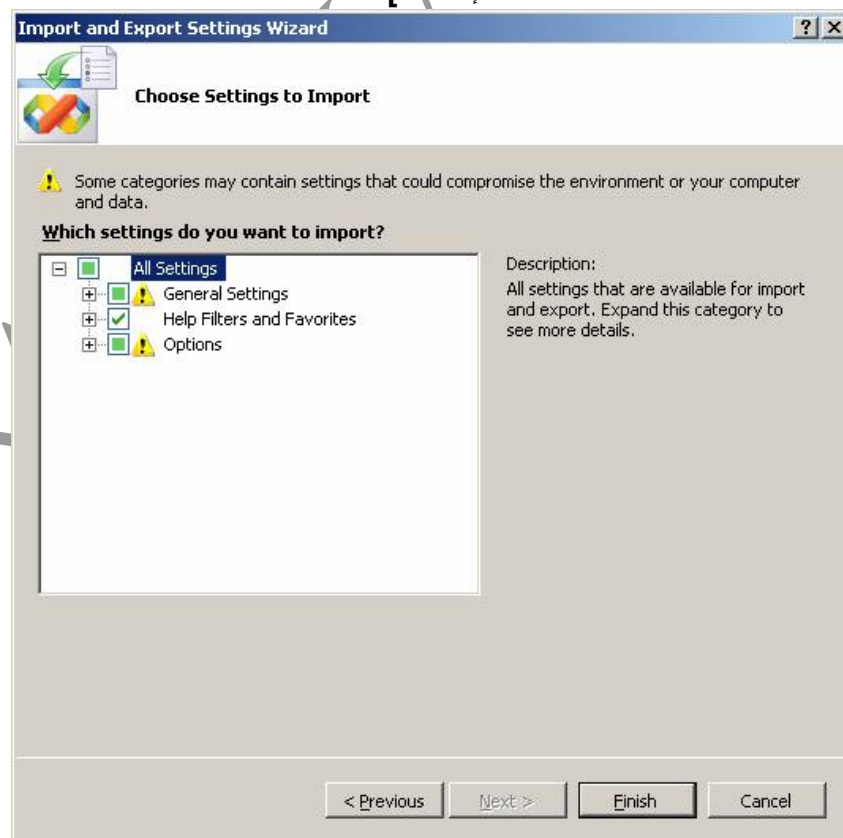
الشكل 56 [تظهر هذه النافذة عند الانتهاء من العملية]



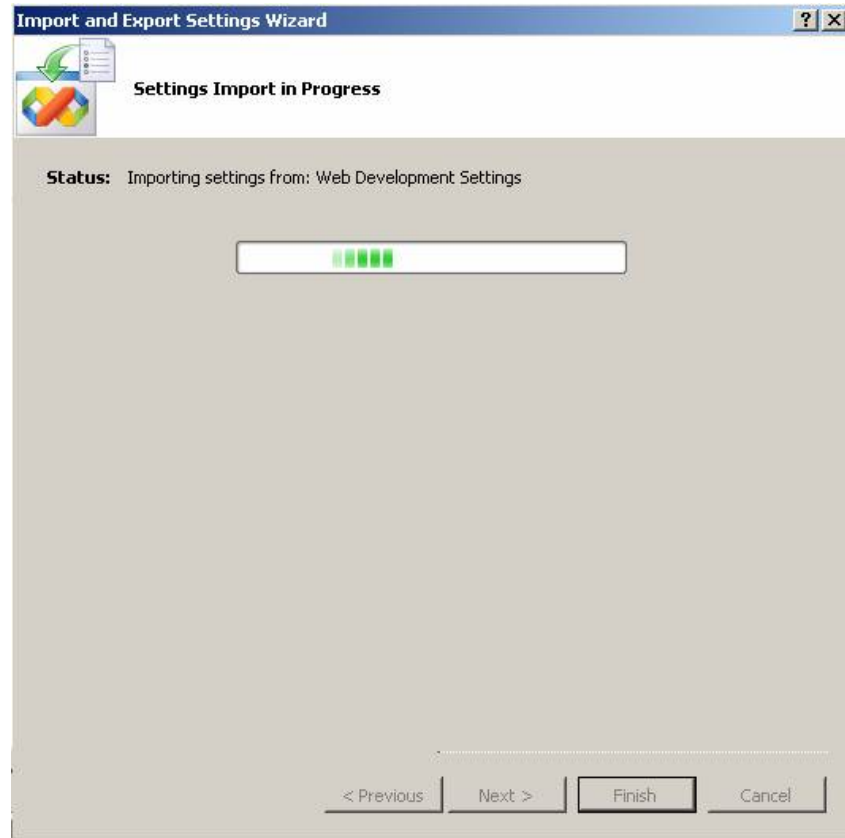
الشكل 57 [ هذه النافذة هي التي تلى الشكل [ ٥٣ ] فى حالة إختيار إستيراد ملف إعدادات ]



الشكل 58 [ يقوم فيه المعالج بتخييرك بين الإعدادات الافتراضية و الإعدادات الحالية و هناك إمكانية لأختيار ملف إعدادات ]



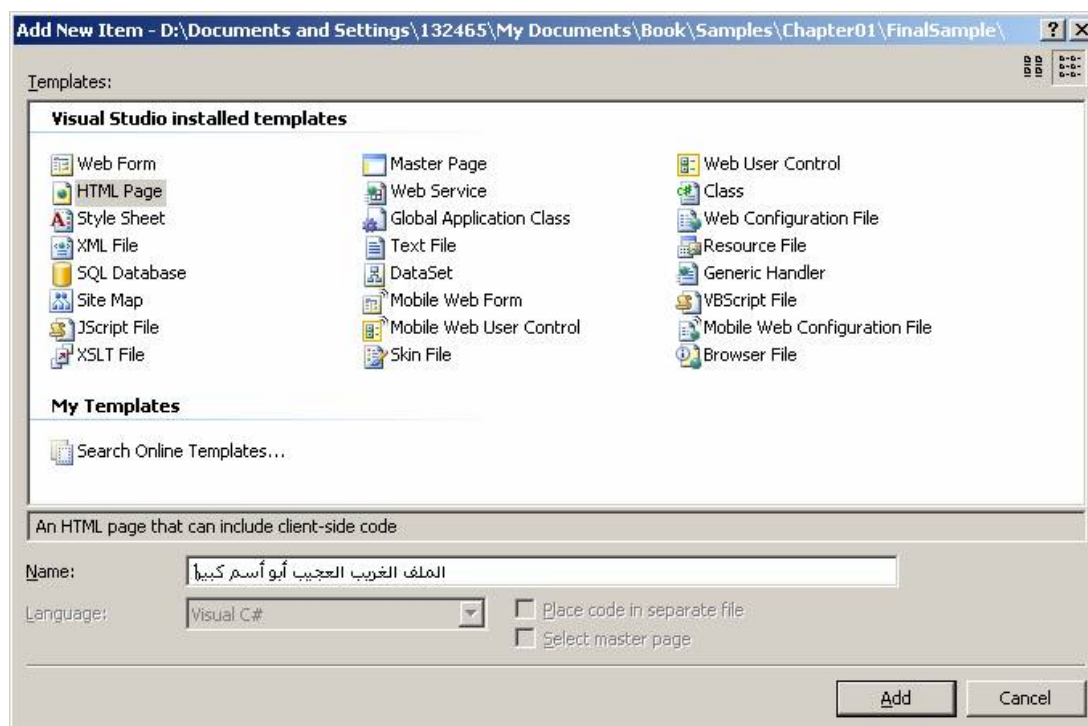
الشكل 59 [ سيظهر لك نافذة إختيار الأجزاء التي تريد إستيرادها ]



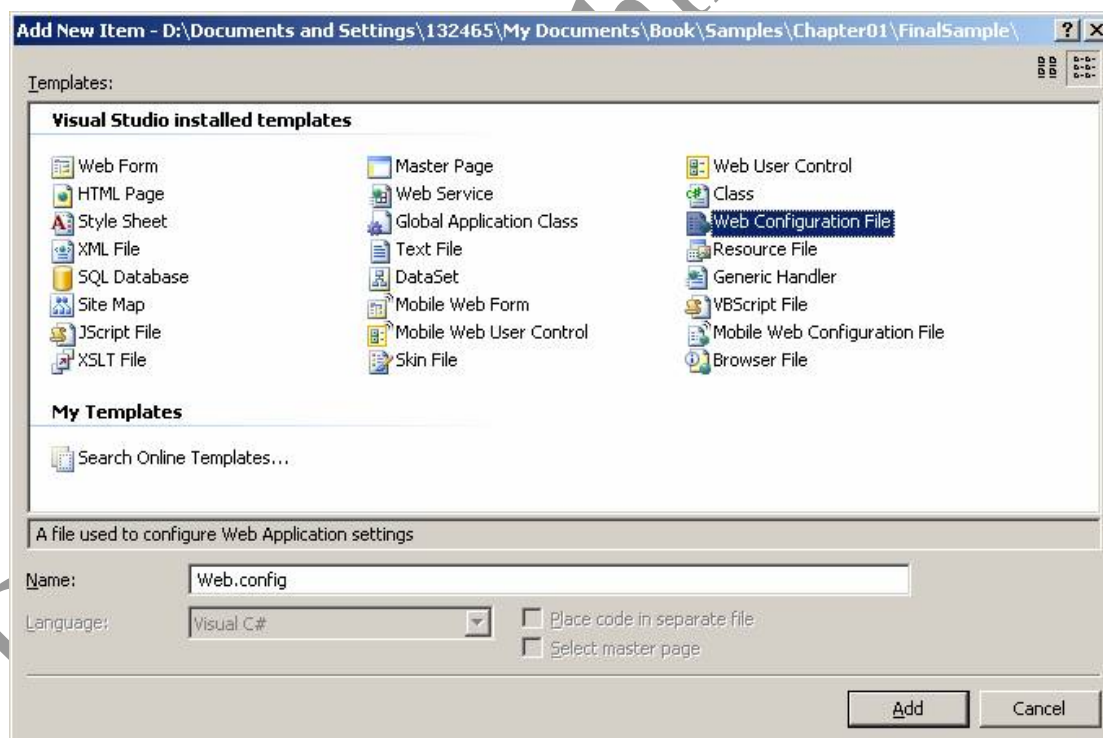
الشكل 60 [ بعد الإنتهاء ستجد أن الإعدادات عادة كما حفظتها في الملف ]

#### ١٤ - URL Mapping :

خاصية جديدة و مفيدة في بعض الأحيان- عن طريقها لن تضطر إلى وضع اسم الملف الحقيقي ؛ و سوف نرى كيف يتم إستغلال هذه الميزة بالمثال الموضح .  
على سبيل المثال -في بعض الأحيان- يسبب تسمية الملف باللغة العربية حدوث بعض المشاكل بوجود هذه الخاصية وُجد حل لهذا المشكلة كما تمكن أيضا من إستغلال هذه الخاصية في الكثير من الأعمال ؛ وسيوضح هذا بمثال عملي .



الشكل 61 [ أولا إضافة صفحة (Web Page) عادية لإختبار 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/" />
```

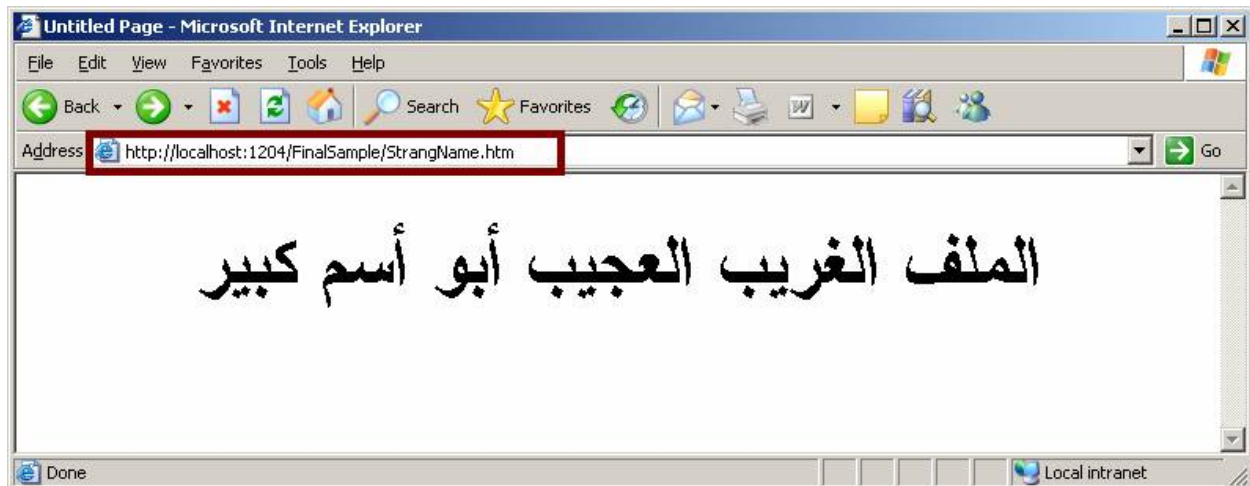
```
</urlMappings>
```

</system.web>

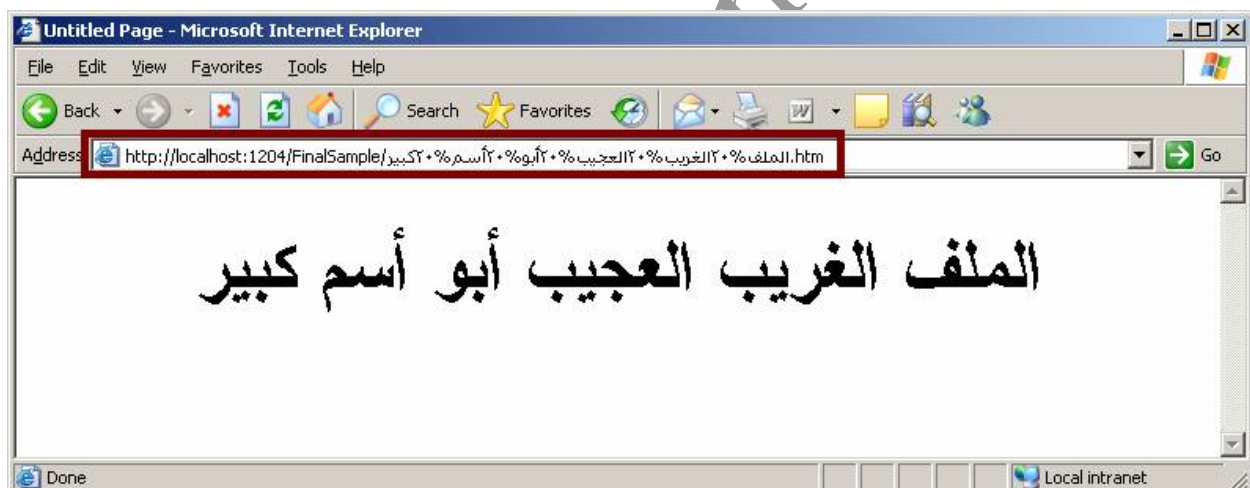
نقوم بفتح الـ (Web.Config) ثم نقوم بإضافة هذا الكود ؛ و الذي نحدد فيه الاسم الوهمي (URL) ، و أسم الملف المطلوب الإشارة إليه أو المطلوب عرضه (Mapped URL) .

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

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



الشكل ٦٣

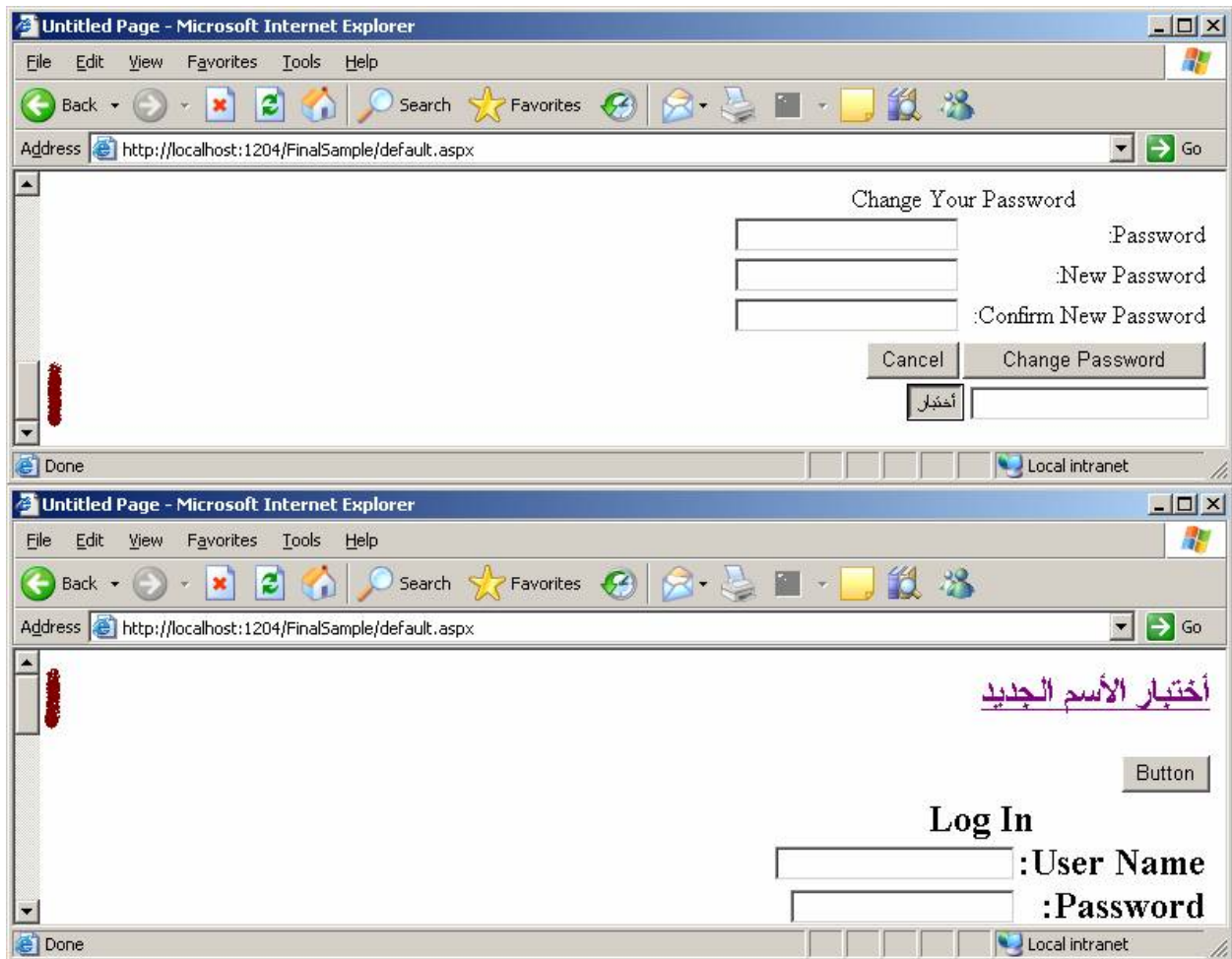


الشكل ٦٤ [ لاحظ الـ (Address) هنا و الفرق بينه و بين الشكل السابق ]

## ١٥ - MaintainScrollPositionOnPostBack

وهي ميزة جديدة توفر كتابة كود جافا سكريبت وقد تم إضافتها إلى موجة الصفحة (Page Directive) حيث تقوم هذه الصفحة بمعالجة موقع الصفحة في حالات إن قامت الصفحة بعمل (Post Back) ؛ حيث كان في السابق ترجع الصفحة إلى أولها ، كما هو موضح في الشكل التالي :-





الشكل 65 [ كما نلاحظ من الصورة أن العمود المنزلق عاد الى أول الصفحة عند الضغط على الزر "اختبار" و لتجنب هذه المشكلة نقوم بإضافة (MaintainScrollPositionOnPostBack) الى موجة الصفحة ]

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

الشكل ٦٦ [ بعد هذه الأضافة لن تظهر هذه المشكلة و لن تحتاج إلى كتابة كود جديد ليقوم بمعالجة هذه المشكلة ]



و في نهاية هذا الفصل أود أن نتذكر سويا ملخصا لما تعلمناه في هذا الفصل:

١. الجديد في (ASP.net) في كلاً من طبقات التعامل ( Server Control – Page Framework – Infrastructure API )
٢. إنشاء مشروع جديد
٣. الطرق الجديدة لحفظ المشروع
٤. الإضافة الجديدة (Partial Class)
٥. تغيير (CodeBehind) إلى (CodeFile)
٦. تعدد اللغات داخل المشروع الواحد
٧. (AutoCompliation)
٨. توارث واجهة التعامل (User Inference Inheritance)
٩. تحسين التنقيح في الإصدارة الجديدة
١٠. مكتبة الصفحة السابقة (Previouspage)
١١. الأدوات الجديدة في (VS.net 2005)
١٢. رفع الملفات
١٣. حفظ الإعدادات و إسترجاعها في (VS.net 2005)
١٤. إظهار العناوين بشكل مختلف (URL Mapping)
١٥. توفير كود جافا سكريبت باستخدام (MaintainScrollPositionOnPostBack)