

Visual Studio 2008 Web Application Development

Content:

1. Visual Studio 2008 Web Application Development



Contents

1	VISUAL STUDIO 2008 WEB APPLICATION DEVELOPMENT	3
1.1	RAPID WEB APPLICATION DEVELOPMENT – ASP .NET STARTER KITS.....	4
1.2	VISUAL STUDIO 2008 IDE ENHANCEMENT.....	8
1.3	VISUAL STUDIO 2008 NEW CONTROLS	21
1.4	VISUAL STUDIO 2008 CLIENT SIDE IMPROVEMENT	40

1 VISUAL STUDIO 2008 WEB APPLICATION DEVELOPMENT

Overview

- Rapid Web Application Development – ASP .NET Starter Kits
- Visual Studio 2008 IDE Enhancement
- Visual Studio 2008 New Controls
- Visual Studio 2008 Client Side Improvement

1.1 Rapid Web Application Development – ASP .NET Starter Kits

■ ASP .NET Starter Kits Web Site

<http://www.asp.net/community/projects/>

Microsoft **ASP.NET** English Deutsch Español Français 汉语 Sign In | Join

Home Get Started Learn Downloads AJAX **Community** Forums

Home > Community > Starter Kits and Community Projects

do everything you want to do on the web...
and more download FREE TRIAL

Starter Kits and Community Projects

The ASP.NET 2.0 Starter Kits for Visual Web Developer are fully functional sample applications to help you learn ASP.NET 2.0 and accomplish common Web development scenarios. Each sample is complete and well-documented so that you can use the code to kick start your Web projects today! These kits, once downloaded are integrated directly into the Visual Web Developer 2005 Express Edition or Visual Studio 2005 experience. After downloading and installing the starter kits, please provide feedback on the [Starter Kit forums](#).

Discover some great open-source projects created by members of the ASP.NET community, which you may use in your own applications.

Starter Kits

[Starter Kits](#)
[By the Community, For the Community](#)
[Open-Source Applications](#)

DotNetNuke® Portal Starter Kit

DotNetNuke is a Web Application Framework ideal for creating and deploying projects such as commercial web sites, corporate intranets and extranets, online publishing portals, and custom vertical applications.

ASP.NET 2.0 StarterKit Hosting

ASP.NET WEB HOSTING

3 Months Free on Award Winning, Developer Ready ASP.NET Web Hosting compatible with the following ASP.NET 2.0 Starter Kits

- **Small Business Site Starter Kit**
- **Paypal eCommerce Site Starter Kit (CSK)**
- **Club Web Site Starter Kit**
- **Personal Web Site Starter Kit**
- **Classified Web Site Starter Kit**
- **Time Tracker Starter Kit**
- **Job Site Starter Kit**
- **And MORE!**

[Click Here to Learn More...](#)

■ Small Business Starter Kit

<http://www.asp.net/downloads/starter-kits/small-business/>

Microsoft **ASP.NET** English Deutsch Español Français 汉语 Sign In | Join

Home Get Started Learn Downloads AJAX **Community** Forums

Home > Community > Starter Kits and Community Projects > Small Business Starter Kit

Small Business Web Site

The Small Business Starter Kit provides a sample of a business promotion website suitable for small and medium-sized businesses. It provides a template for customizing and creating a site for your own business out-of-the-box, with advanced features including integration with SQL and XML data sources for content and data management.

- [Download the Small Business Site Starter Kit](#)

[Try it Live!](#)

The key pages of the Small Business Site starter kit are:

- **Catalogs**
Display complete range of products and services offered by your business.
- **News**
Press room for the business where the relevant news and press releases can be posted.
- **Testimonials**
Showcase testimonials given by customers regarding their experience with your product or service.
- **People**
Provide summary and in-depth info about the professionals running your business.

ASP.NET 2.0 StarterKit Hosting

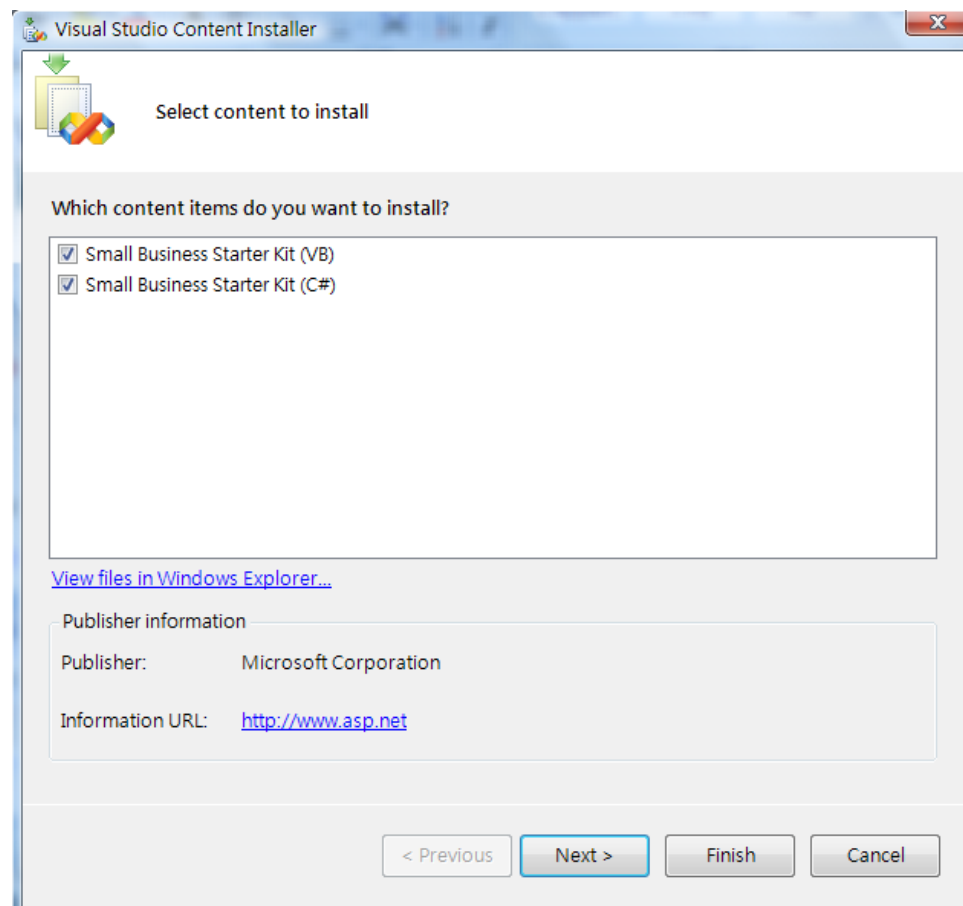
ASP.NET WEB HOSTING

3 Months Free on Award Winning, Developer Ready ASP.NET Web Hosting compatible with the following ASP.NET 2.0 Starter Kits

- **Small Business Site Starter Kit**
- **Paypal eCommerce Site Starter Kit (CSK)**
- **Club Web Site Starter Kit**
- **Personal Web Site Starter Kit**
- **Classified Web Site Starter Kit**
- **Time Tracker Starter Kit**
- **Job Site Starter Kit**
- **And MORE!**

[Click Here to Learn More...](#)

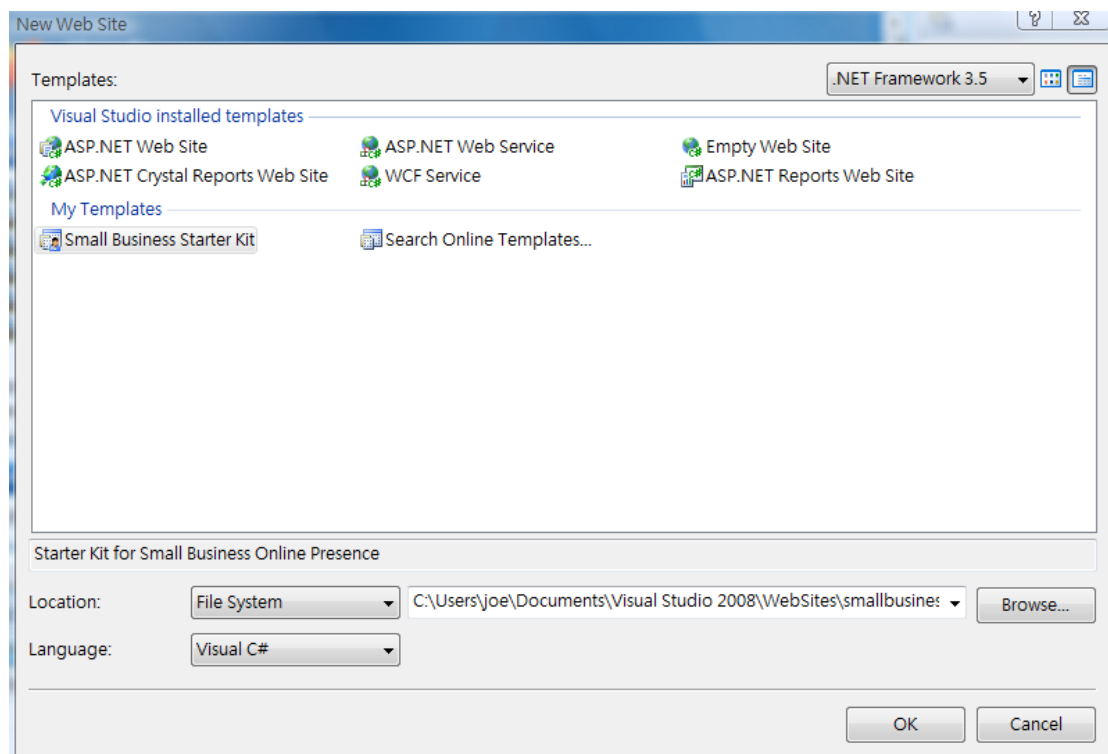
■ **Download and install the Small Business Starter Kit.**



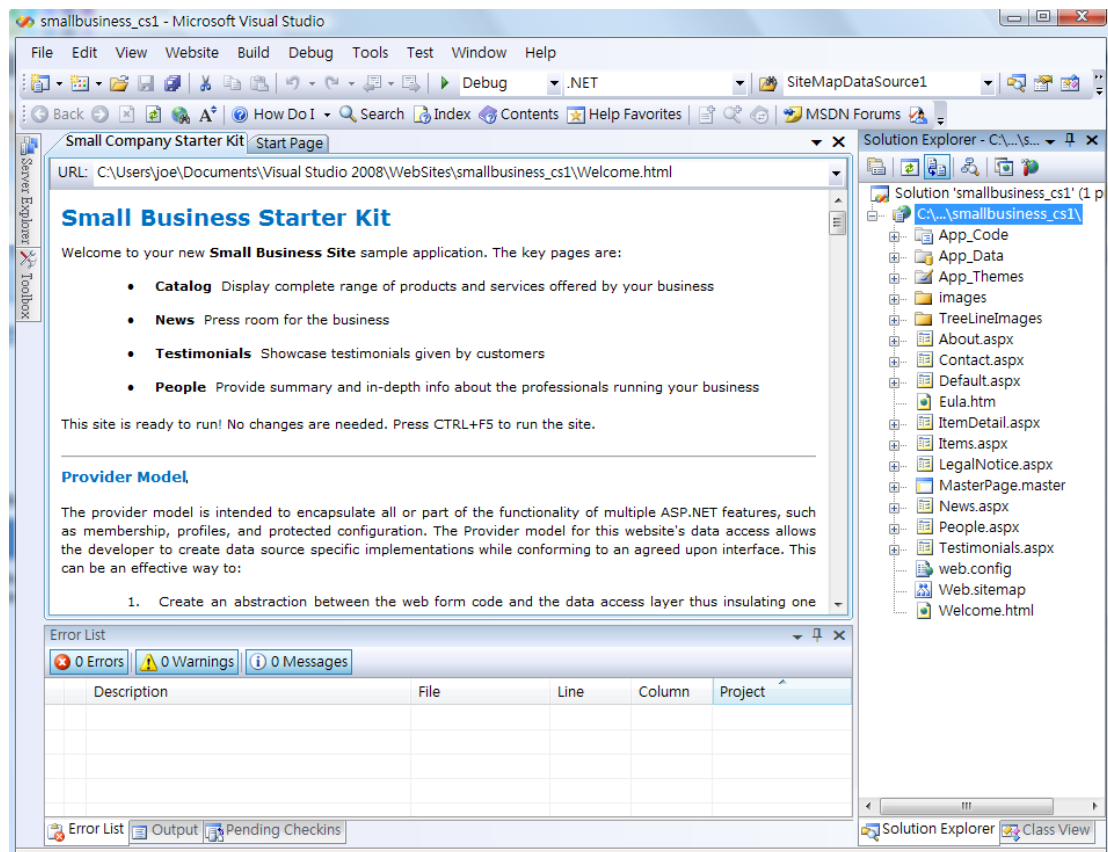
Create a New Web Site by Small Business Starter Kit

■ Create a New Web Site

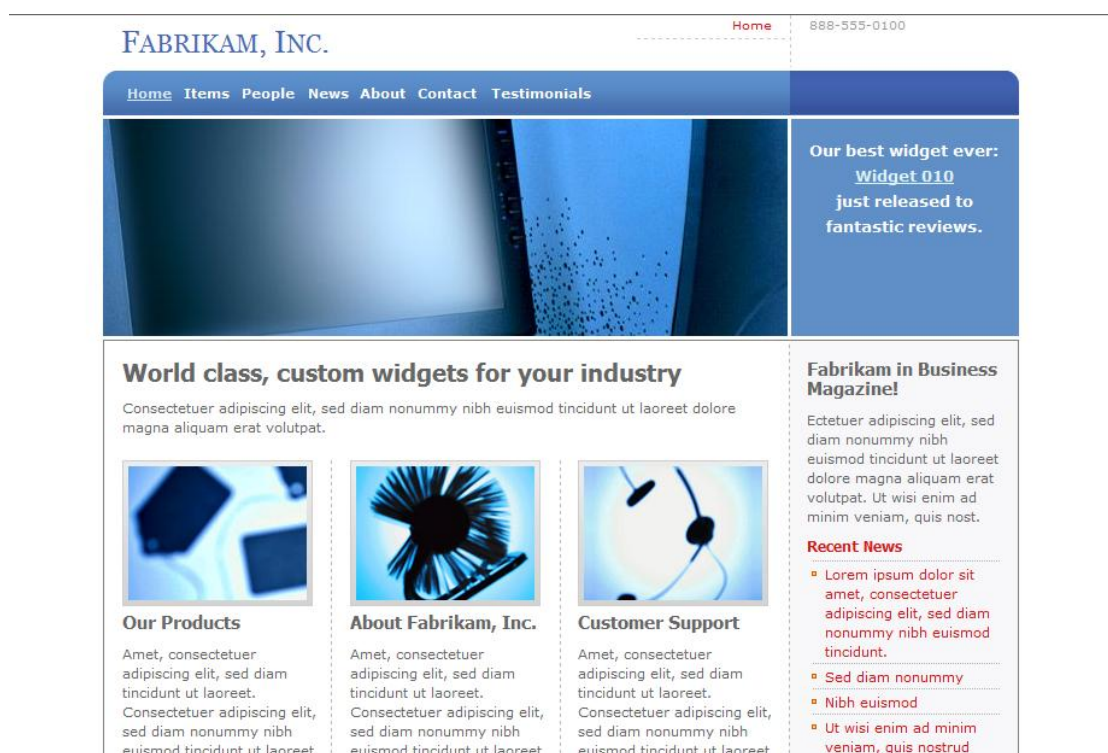
1. Start **Visual Studio .Net 2008 RTM** and select the **File | New | Web Site** menu command. In New Web Site Dialog Box. For **Templates** select Small Business Starter Kit. For **Languages** select C# and then click **OK** button.



2. A New Web Site has been created by Small Business Starter Kit.



3. Press **F5** to run the Web Site and view the function.

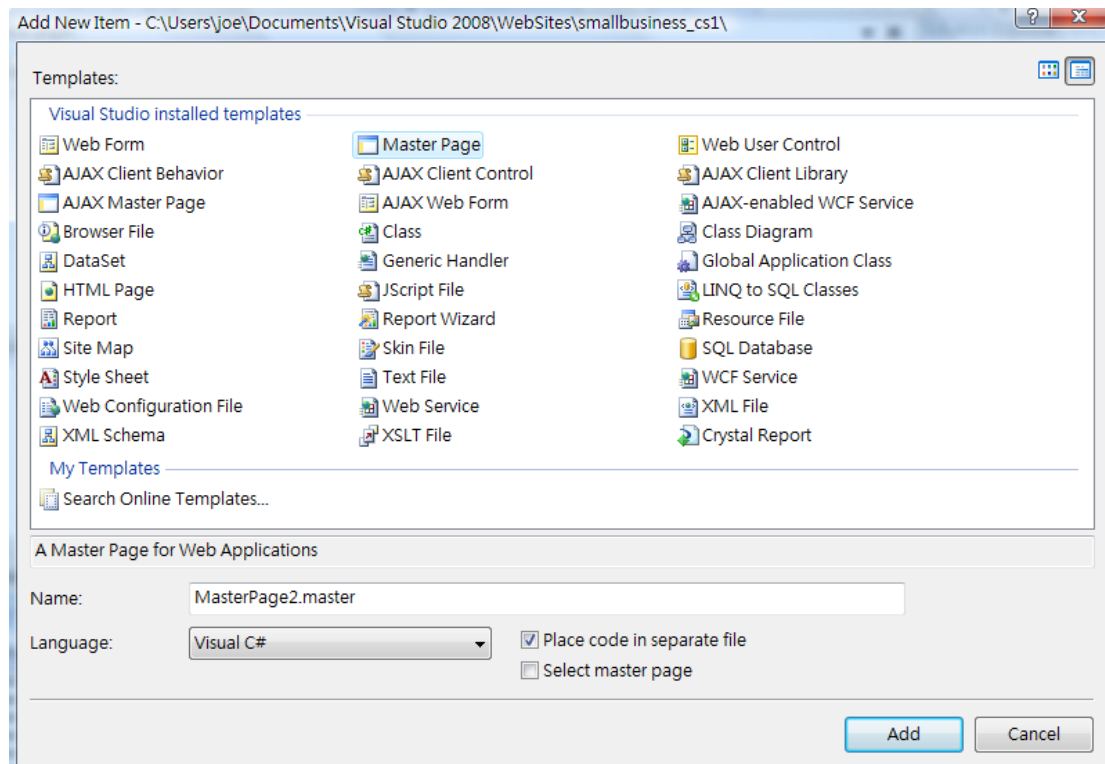


1.2 Visual Studio 2008 IDE Enhancement

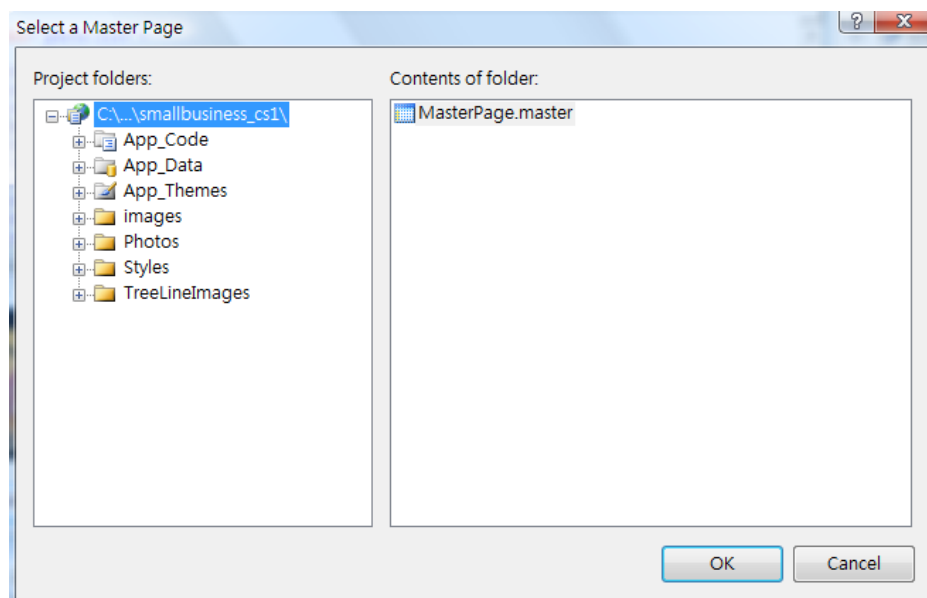
- Nested Master Pages Support
- HTML Designer Improvements
- Advanced CSS Support

Nested Master Pages Support

4. Select the Web Site, right click and select **Add New Item** menu command. In Add New Item Dialog Box, select **Master Page** template, checked **Select master page** checkbox and then click **Add** button.



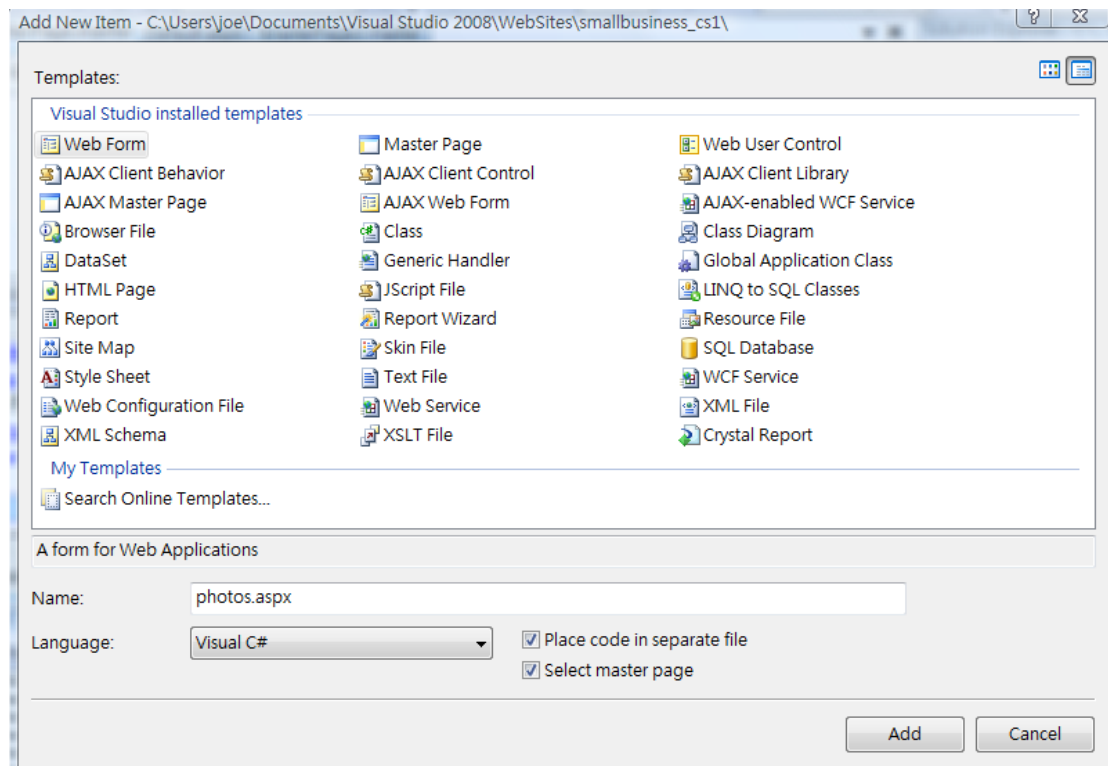
5. In Select a Master Page dialog box, select MasterPage.master and then click **OK** button.



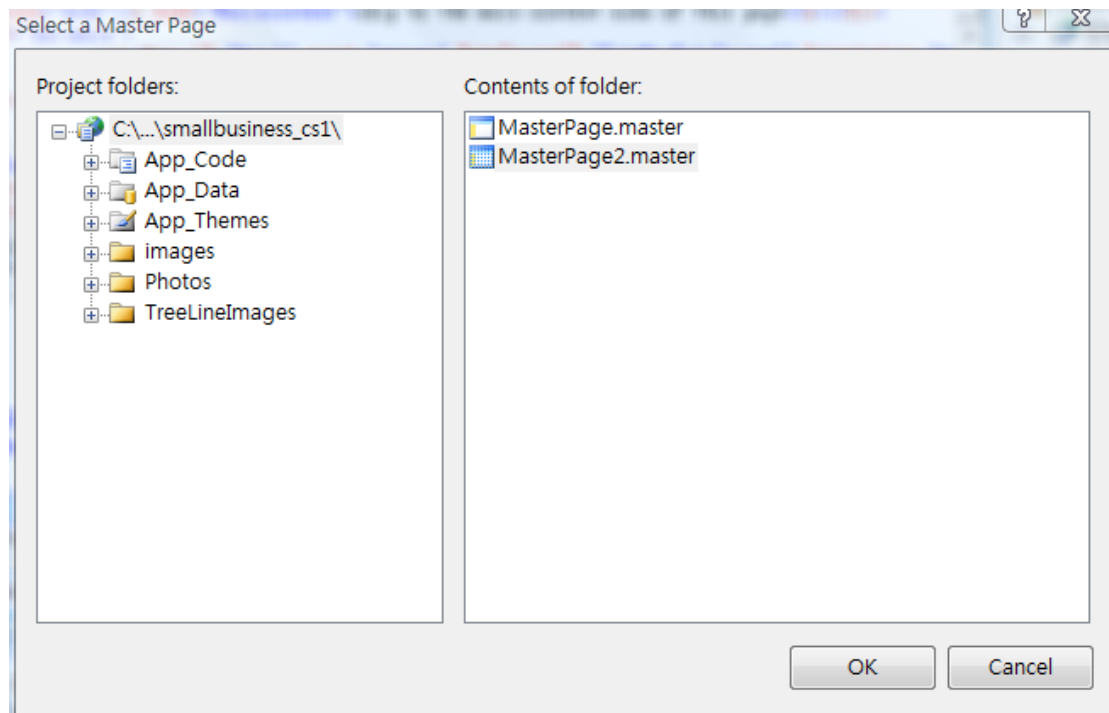
6. Open the **MasterPage2.master** and Insert below code inside the **ContentPlaceHolder**

```
<div id="content-container-two-column">
  <table width="100%">
    <tr>
      <td style="border-style:dotted;border-color:#0E0E6C">
        <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
        </asp:ContentPlaceHolder>
      </td>
    </tr>
  </table>
</div>
```

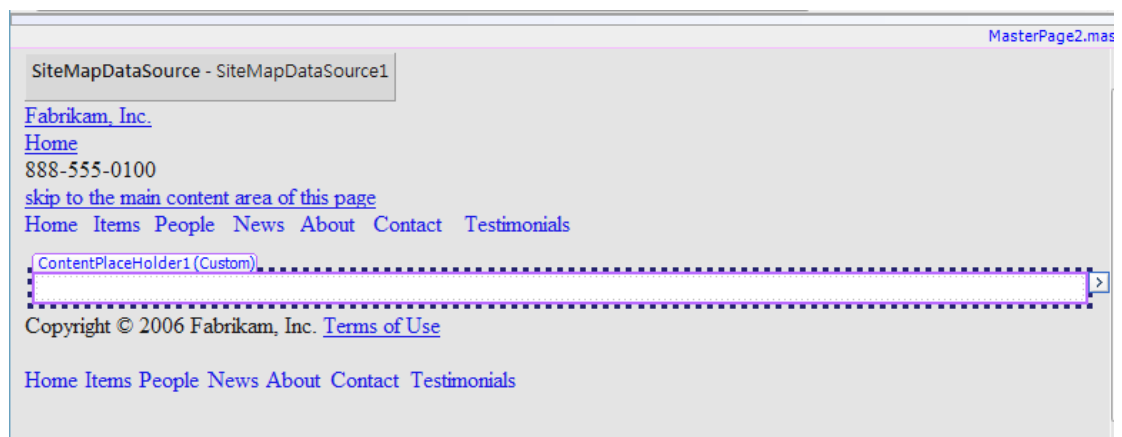
7. Select the Web Site, right click and select **Add New Item** menu command. In Add New Item Dialog Box, select **Web Form** template, checked **Select master page** checkbox and input name as **photos.aspx** and then click **Add** button.



8. In Select a Master Page dialog box, select MasterPage2.master and then click **OK** button.



9. Open **photos.aspx**, you can see the Visual Studio 2008 IDE now support nested master page display.



HTML Designer Improvements

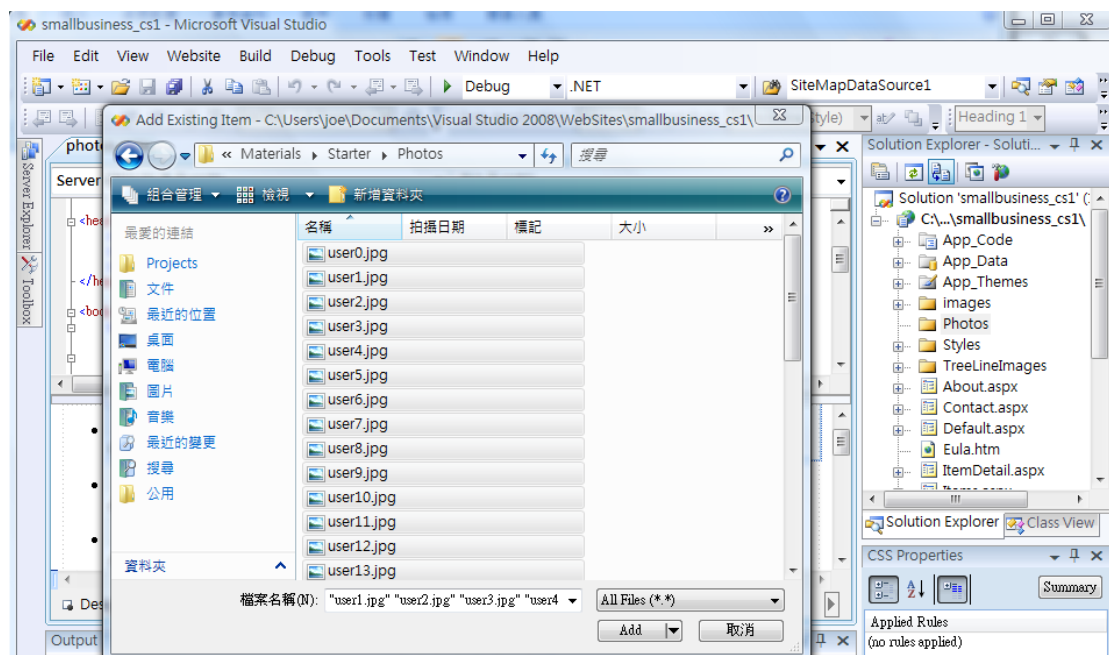
- Open the **photos.aspx** and Insert below code inside the **ContentPlaceHolder**

```
<ul>

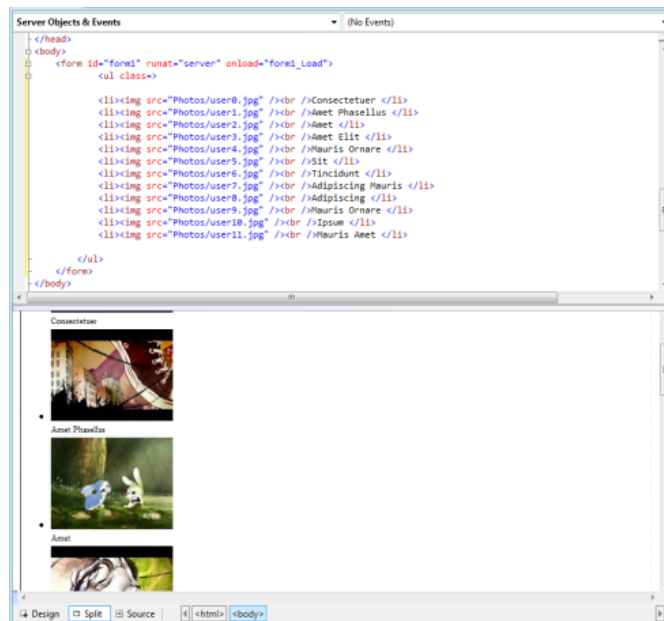
    <li><br />Consectetuer </li>
    <li><br />Amet Phasellus </li>
    <li><br />Amet </li>
    <li><br />Amet Elit </li>
    <li><br />Mauris Ornare </li>
    <li><br />Sit </li>
    <li><br />Tincidunt </li>
    <li><br />Adipiscing Mauris </li>
    <li><br />Adipiscing </li>
    <li><br />Mauris Ornare </li>
    <li><br />Ipsum </li>
    <li><br />Mauris Amet </li>

</ul>
```

- Select the Web Site, right click and select **New Folder** command to create a new folder "Photos".
- Select the **Photos** folder, right click and select **Add Existing Item** menu command. In Add Existing Item Dialog Box, select **All Images** in Starter/Photos folder and then click **Add** button.

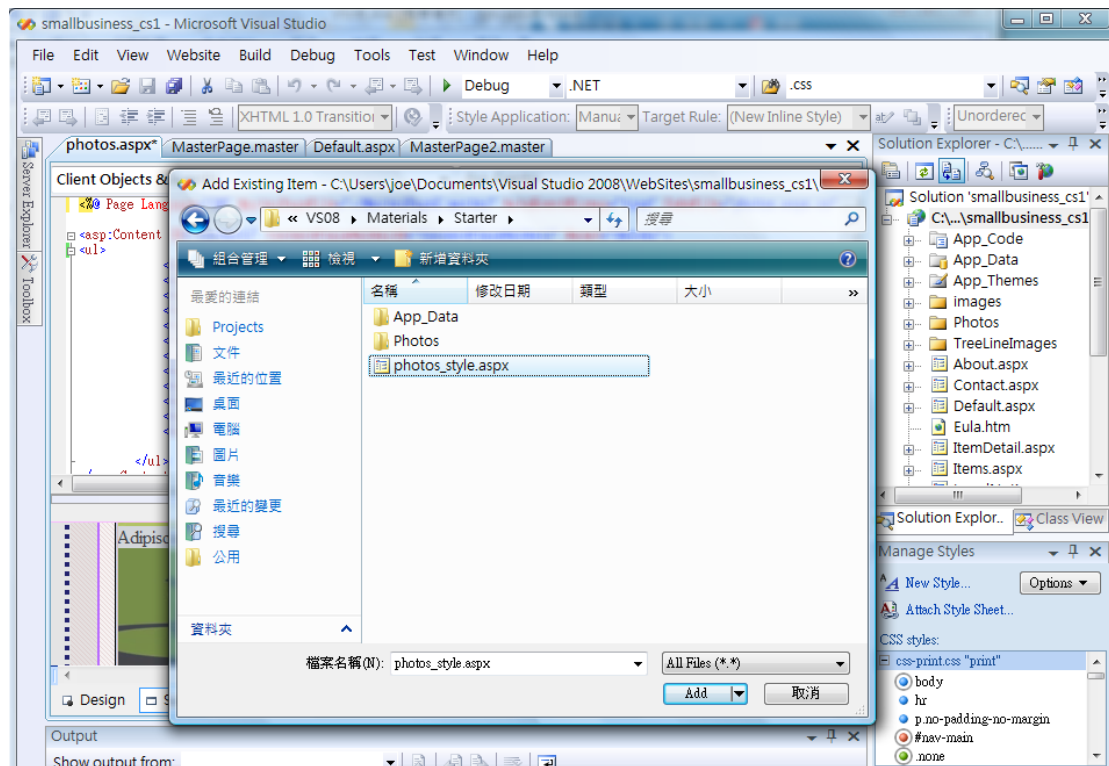


13. Open **photos.aspx**, you can see the **split view** of source and design.

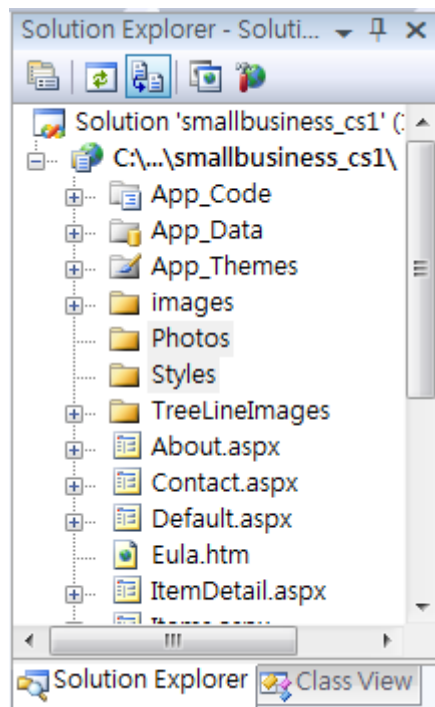


Advanced CSS Support

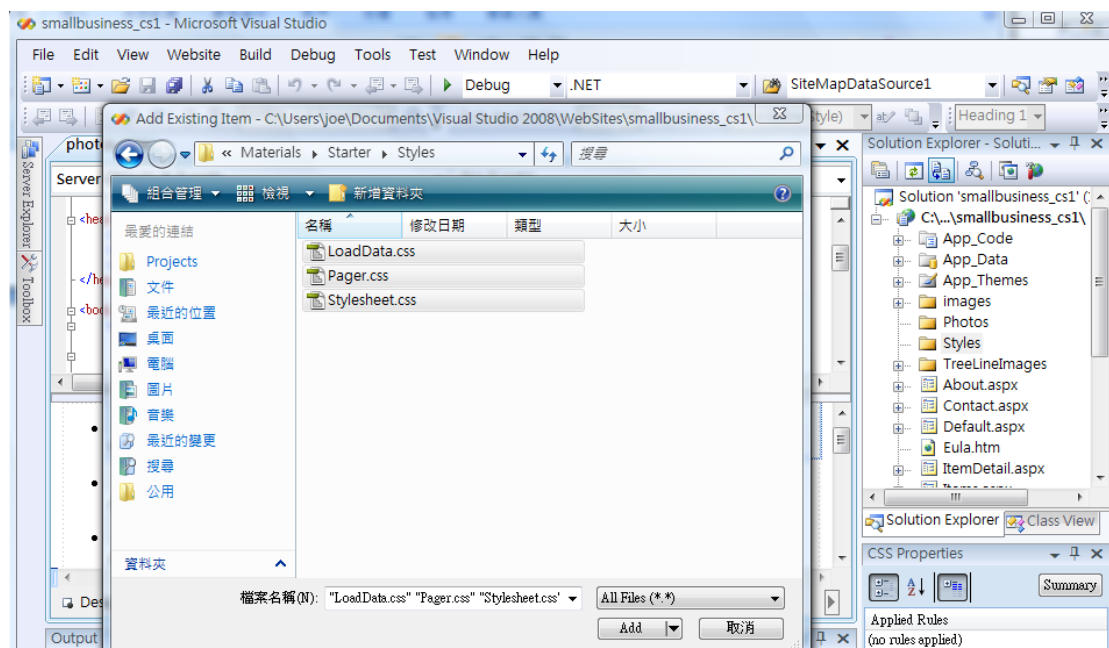
14. Select the Web Site, right click and select **Add Existing Item** menu command. In Add Existing Item Dialog Box, select **photos_style.aspx** in Starter folder and then click **Add** button.



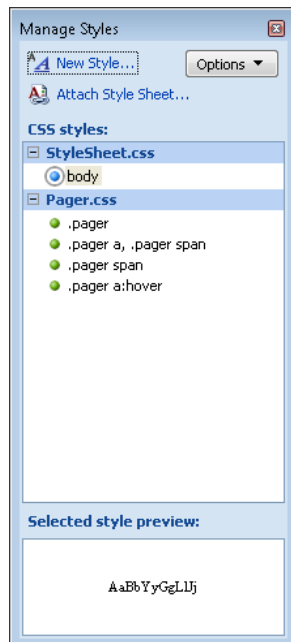
15. Select the Web Site, right click and select **New Folder** command to create a new folder “Styles”.



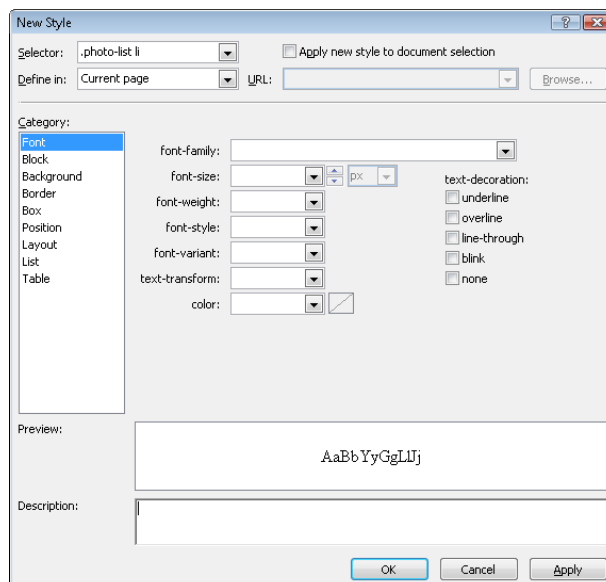
16. Select the **Styles** folder, right click and select **Add Existing Item** menu command. In Add Existing Item Dialog Box, select **LoadData.css**, **Pager.css** and **Stylesheet.css** in Starter/Styles folder and then click **Add** button.



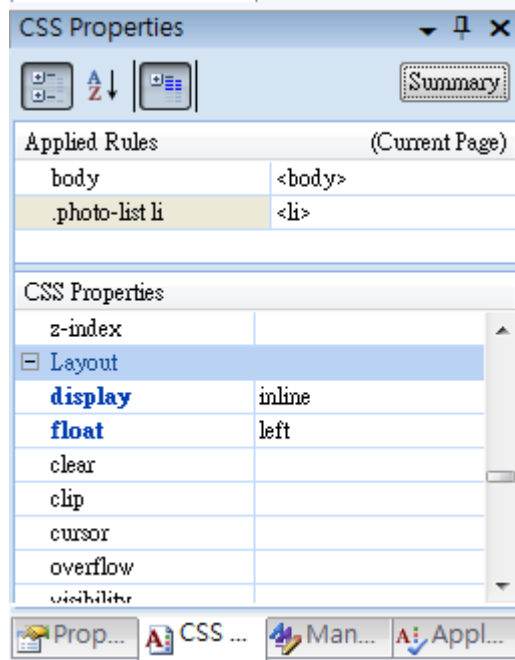
17. To open **Managed Styles**, click on the **View** menu, select **Manage Styles**.



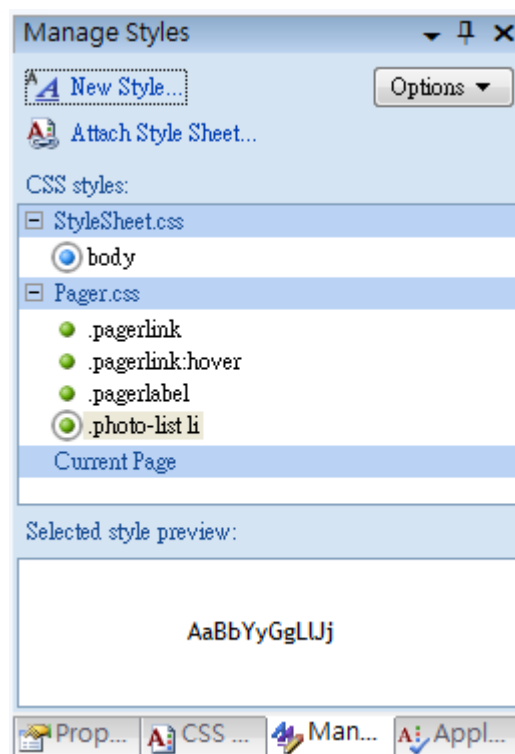
18. Select the **** element and click **New Style** from Manage Styles page.



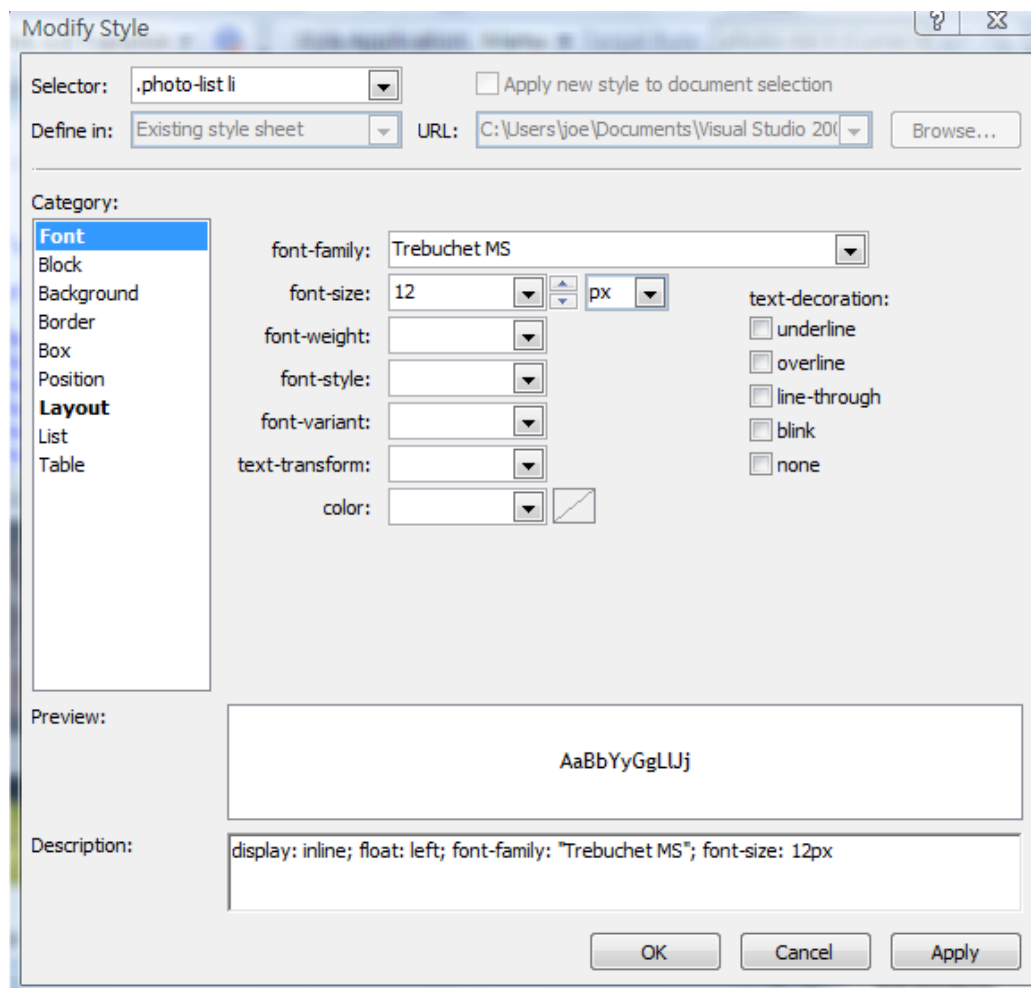
19. In New Style Dialog Box, In Selector type “**.photo-list li**” and click OK.
20. Select the **** element. In CSS Properties, in **Layout** category, set float = “left” and display = “inline”.



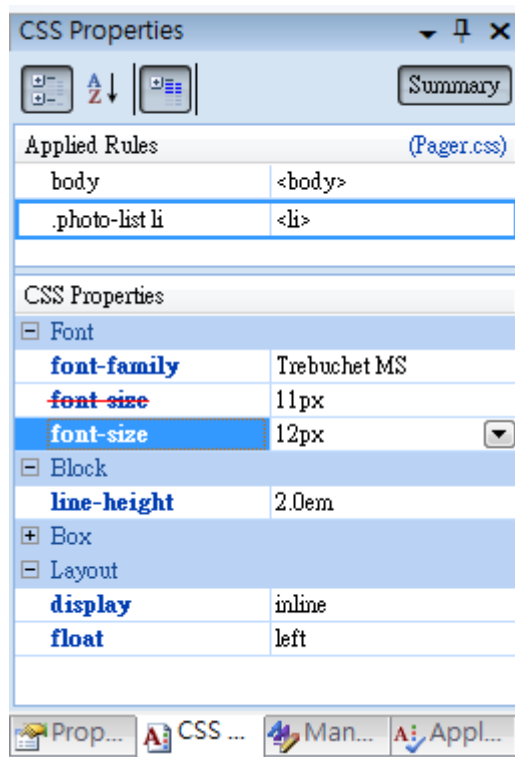
21. Drag **.photo-list li** style from page to **Pager.css** in Manage Styles.



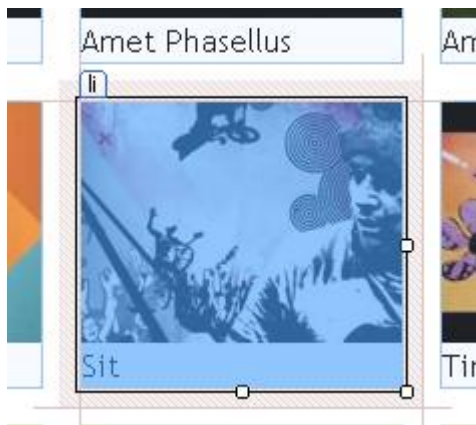
22. Select the style in Manage Styles and right click and click **Modify Style**



23. In **View** menu, select **CSS Properties**, click **Summary** and show the override of the font size. Click the element and show how IDE highlights where the rule comes from.

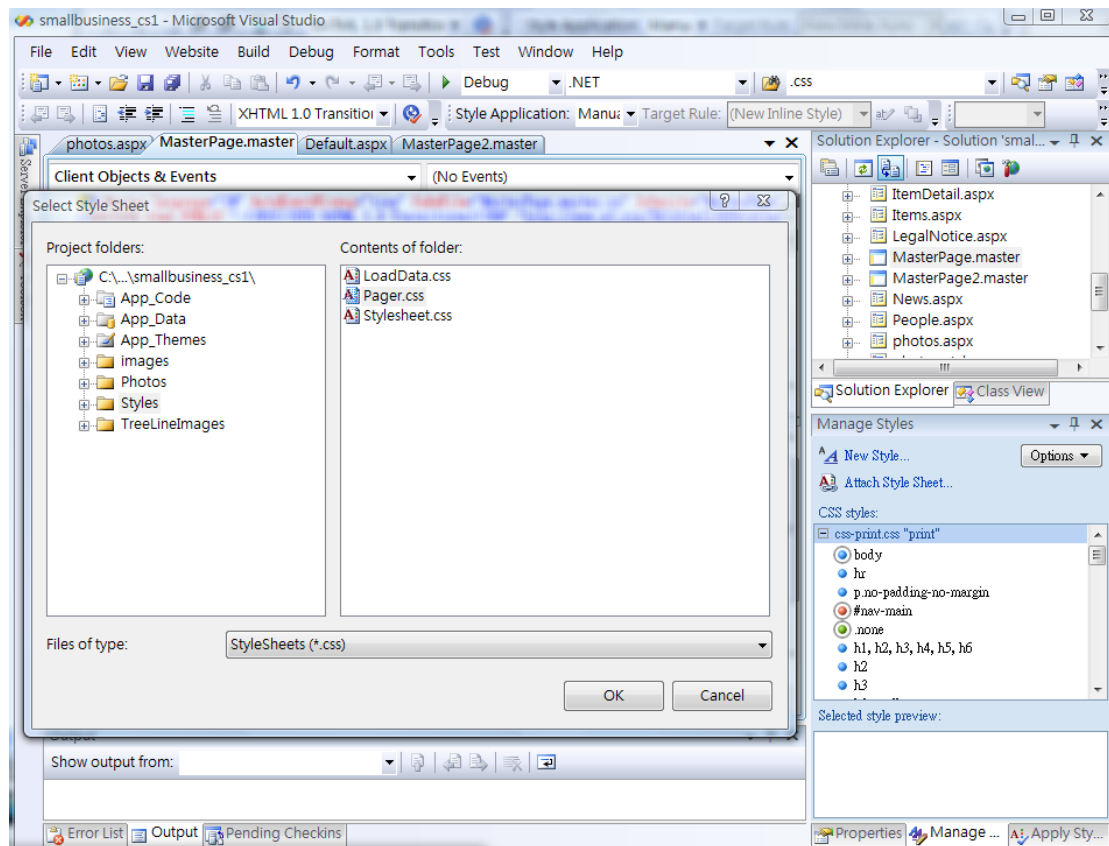


24. Switch to split view and select a **** tag in the source pane. In the design view drag the margin handles to visually set the margin to 10px.

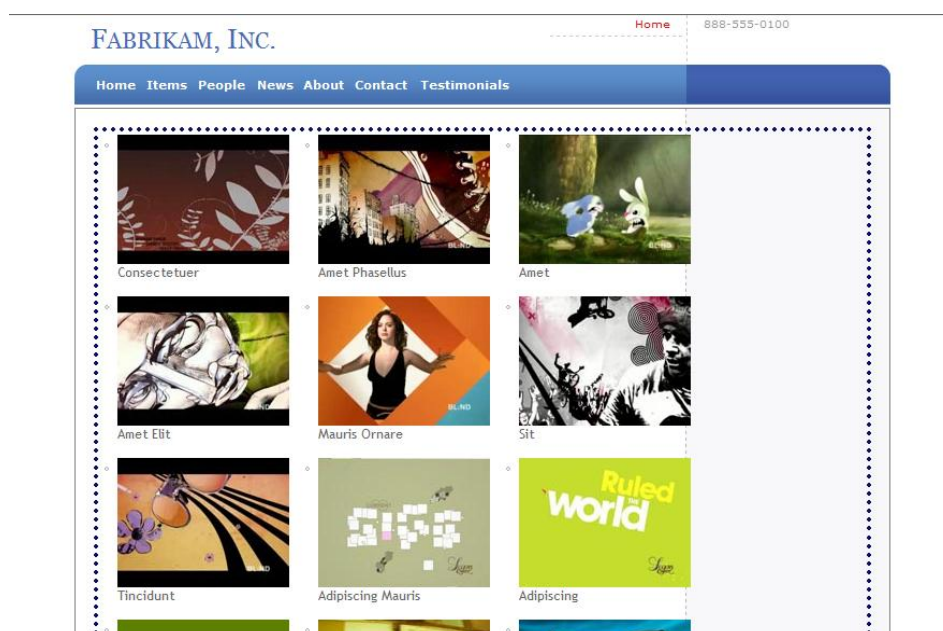


25. Open the **Pager.css** and show how the margins were applied to the target rule.

26. Select **MasterPage.master**, in Manage Styles click **Attach Style Sheet**, select **Pager.css** in Select Style Sheet Dialog Box.



27. In **photos.aspx**, in **ul** tag, apply the "**photo-list**" style.
28. CTRL-F5 and show the **photos.aspx** in IE.

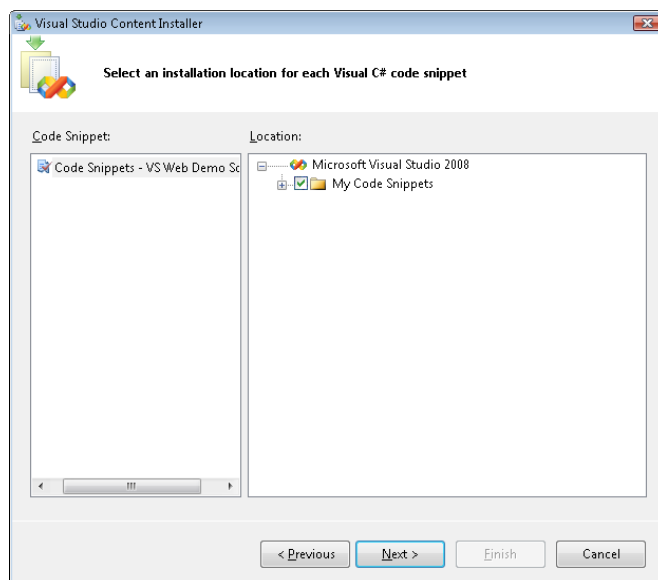
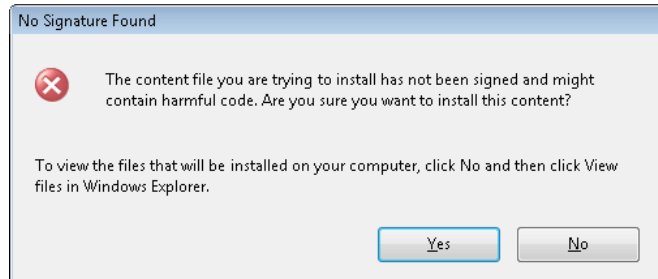
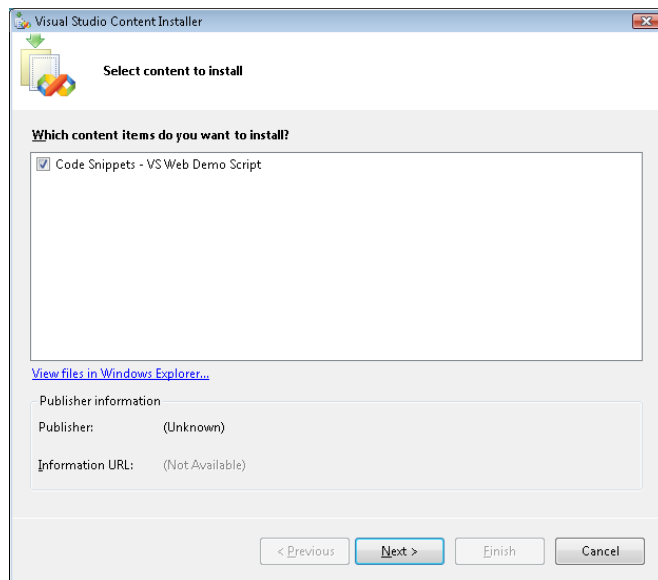


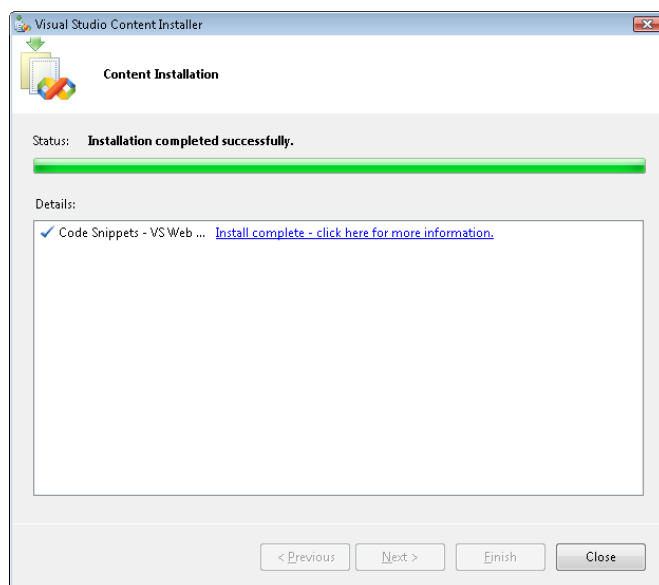
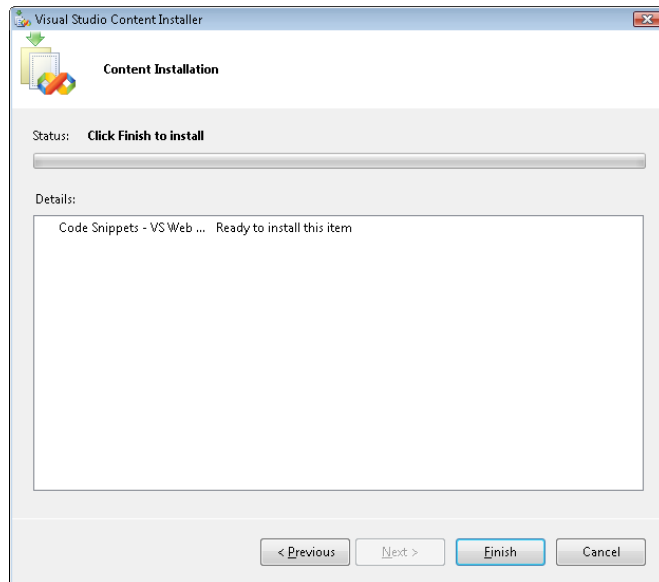
1.3 Visual Studio 2008 New Controls

- LinqDataSource
- ListView
- DataPager

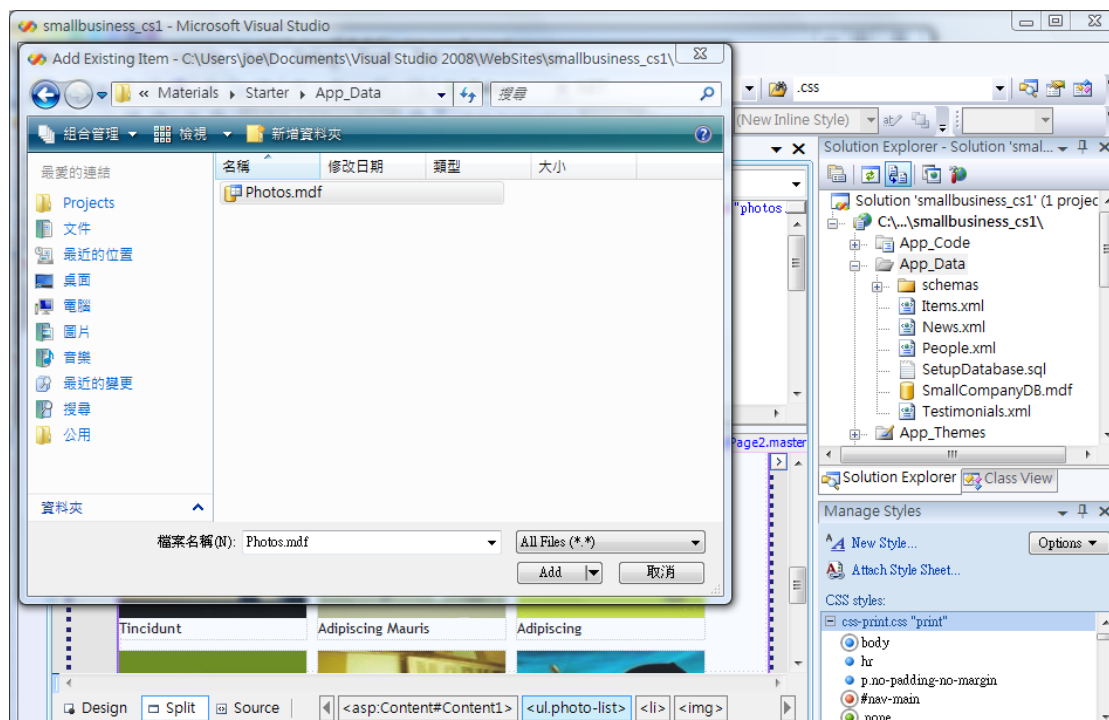
LinqDataSource

29. Double-click on the **SetupEx.cmd** file in the Setup folder. When prompted, choose to install the Visual Studio Code Snippets

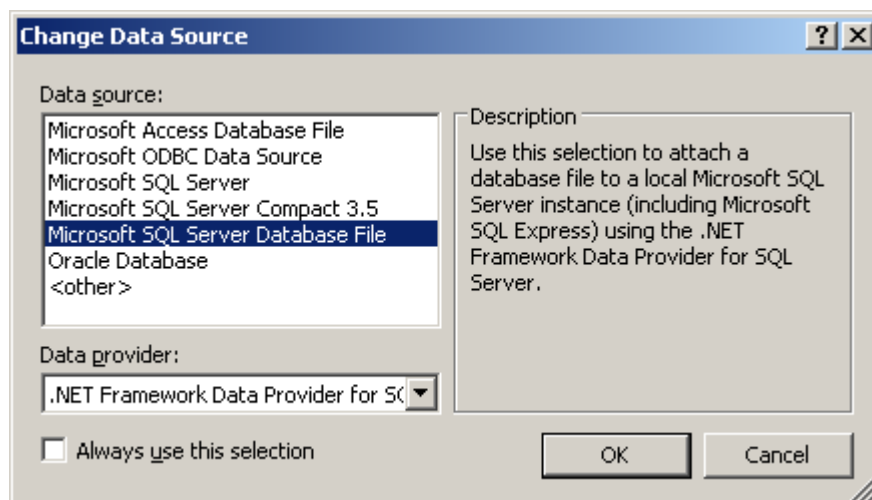




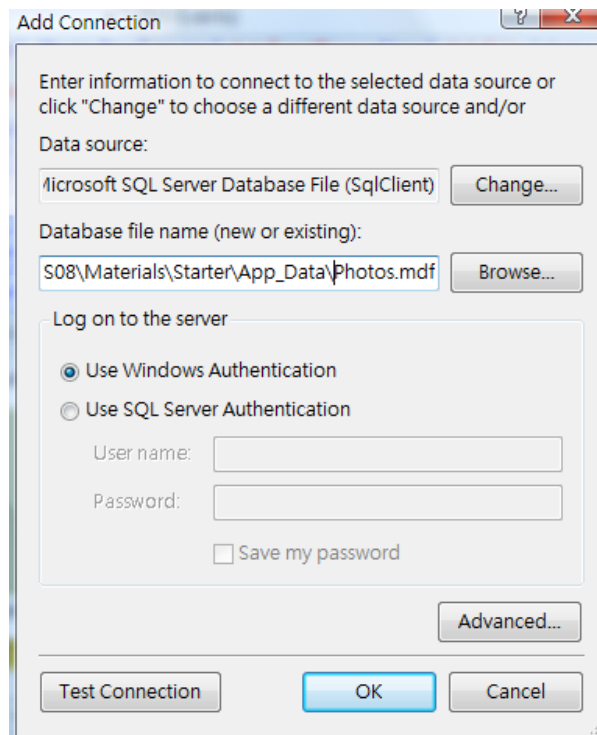
30. Select the **App_Data** folder, right click and select **Add Existing Item** menu command. In Add Existing Item Dialog Box, select **Photos.mdf** in Starter/App_Data folder and then click **Add** button.



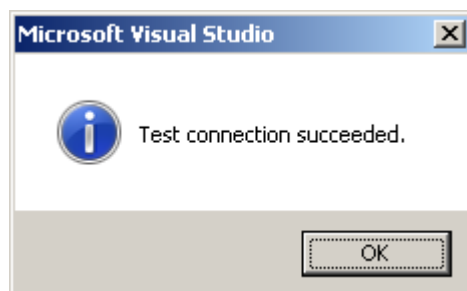
31. A new data connection should be added automatically to your environment when you open the web site in Visual Studio. Otherwise, in **Server Explorer**, add a data connection to the **Photos** database.
32. Choose the **Microsoft SQL Server Database File**



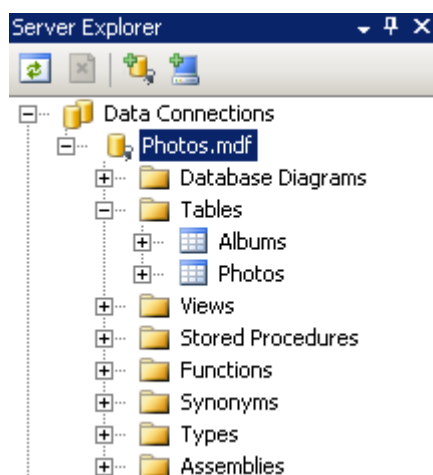
33. Select the database **Photos.mdf** under App_Data folder



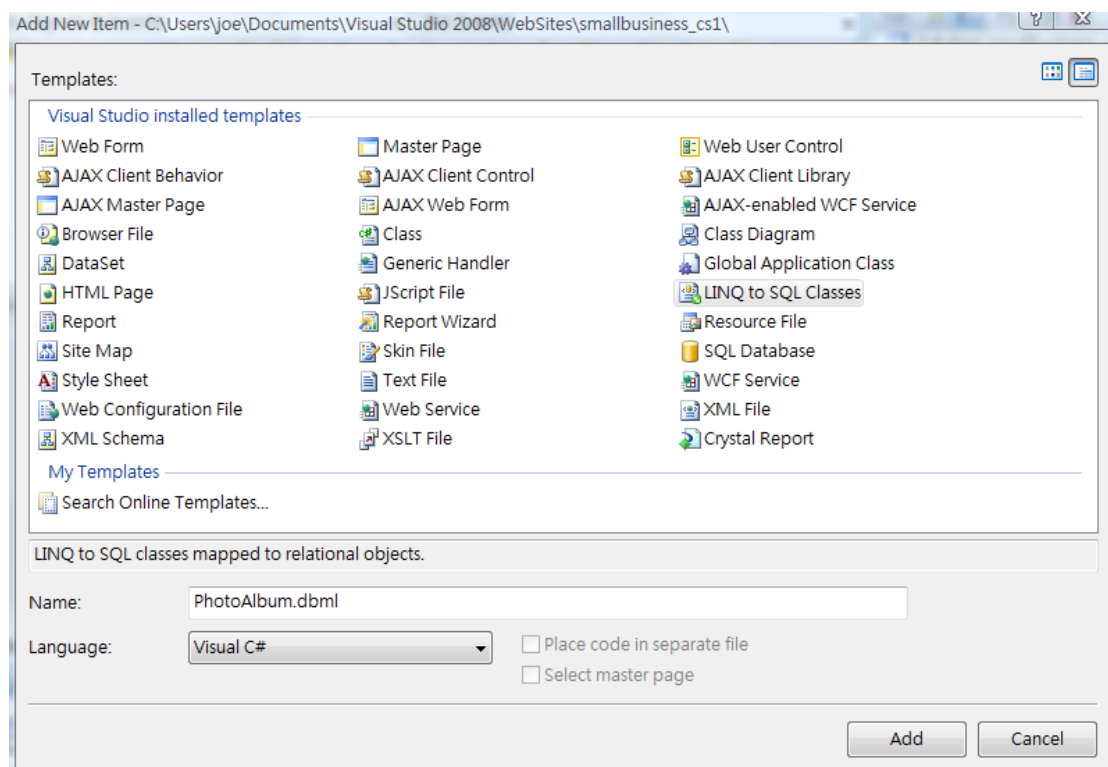
34. Test the new connection



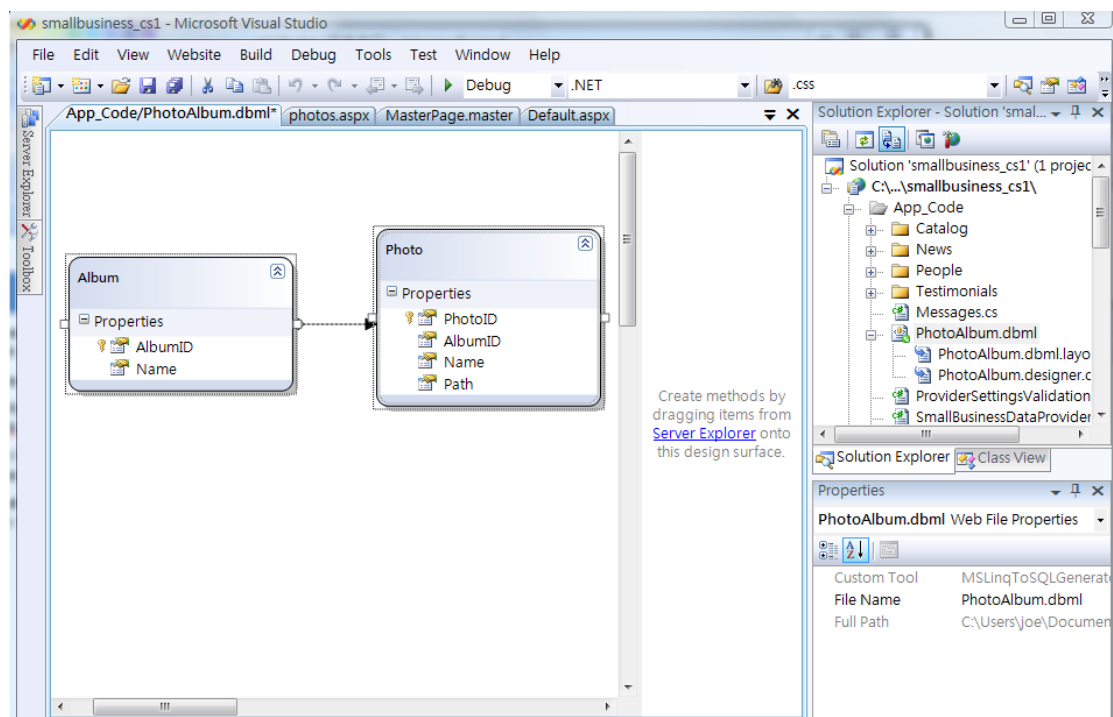
35. After creating the connection, expand it until all tables are shown.



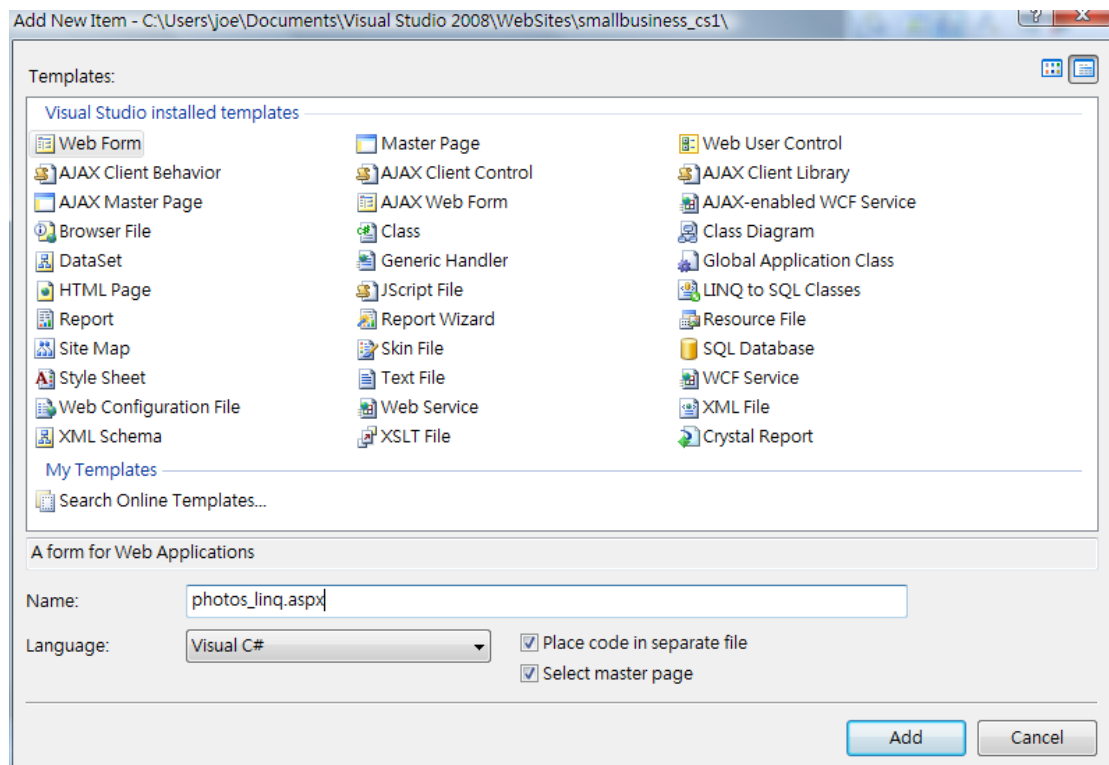
36. Select the Web Site, right click and select **Add New Item** menu command. In Add New Item Dialog Box, select **LINQ to SQL Classes** template, input "**PhotoAlbum.dbml**" as name and then click **Add** button.



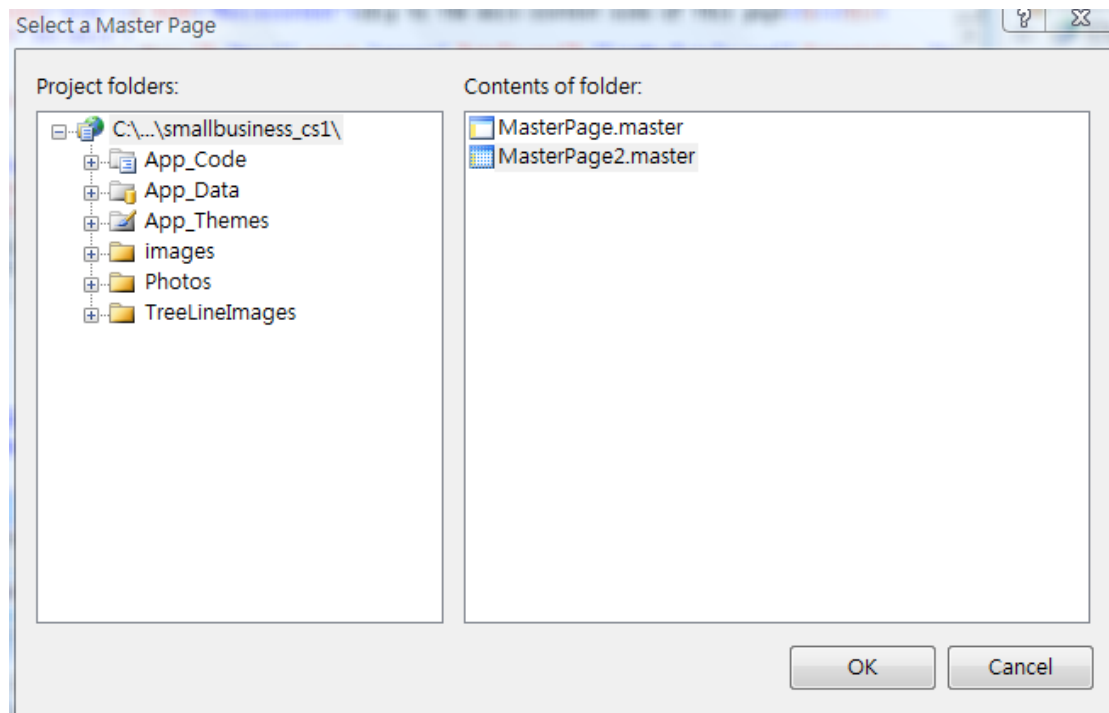
37. Open Server Explorer and drag **Album** and **Photo** tables onto design surface.



38. Select the Web Site, right click and select **Add New Item** menu command. In Add New Item Dialog Box, select **Web Form** template, checked **Select master page** checkbox and input name as **photos_linq.aspx** and then click **Add** button.



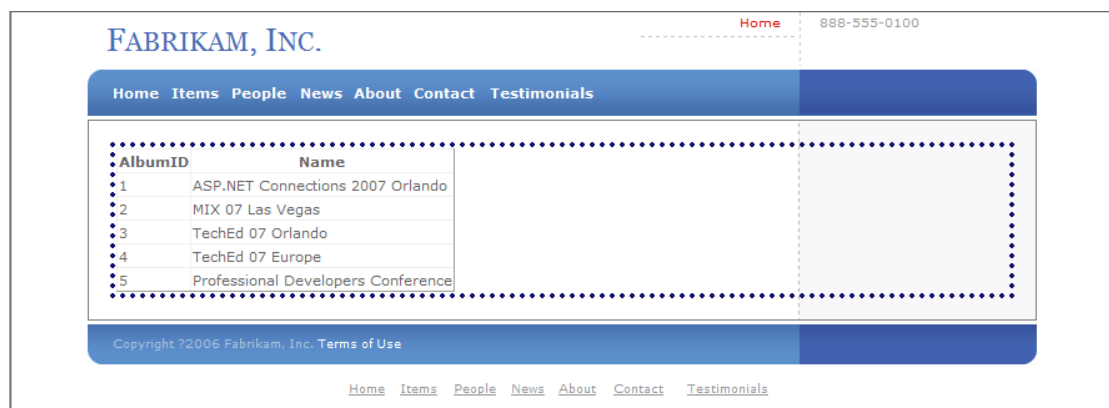
39. In Select a Master Page dialog box, select MasterPage2.master and then click **OK** button.



40. Drop a **GridView** onto the page.
41. Create a **Page Load** event and then create a simple select that takes from Album table and databind the statement to the GridView by input below code or insert the **QueryAlbums snippet**.

```
var ctx = new PhotoAlbumDataContext();
var q = from a in ctx.Albums
        select a;
GridView1.DataSource = q;
GridView1.DataBind();
```

42. CTRL-F5 and show the completed work in IE.



43. Show **projection** using new keyword. Replace the query by input below code or insert the **QueryAlbumsProjection** snippet.

```
var q =
  from a in ctx.Albums
  select new {
    Name = a.Name,
    Count = a.Photos.Count
  };
```

44. CTRL-F5 and show the completed work in IE.

FABRIKAM, INC. Home 888-555-0100

Home Items People News About Contact Testimonials

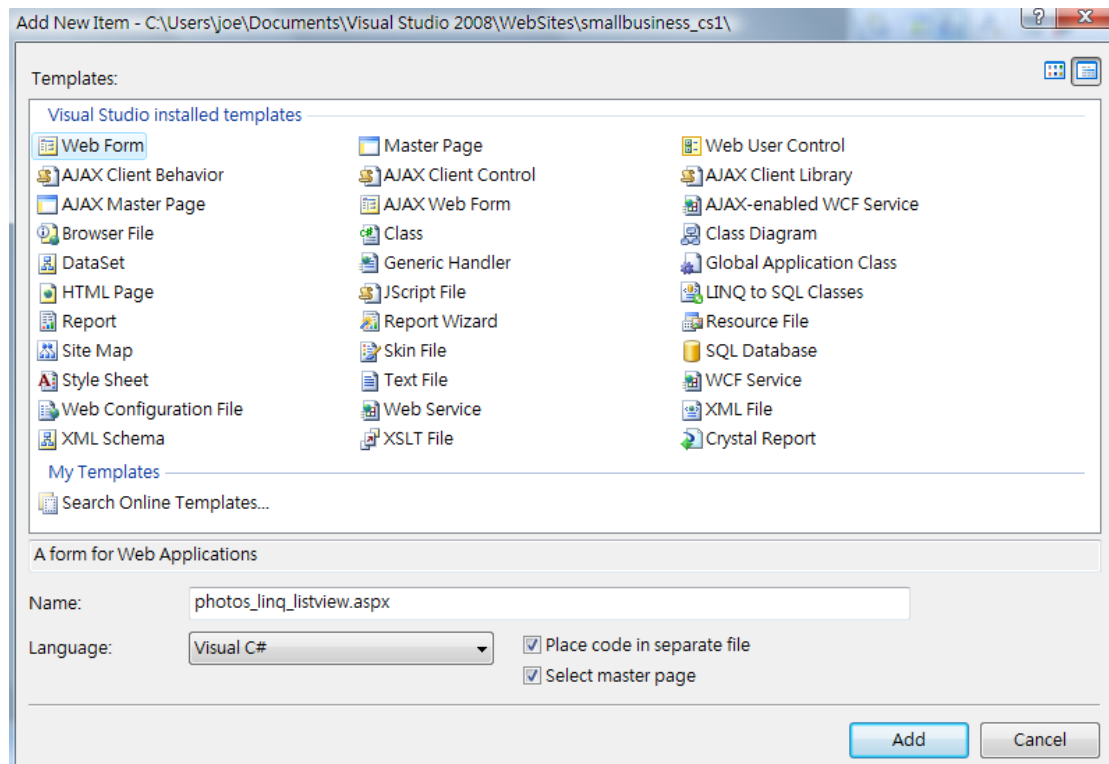
Name	Count
ASP.NET Connections 2007 Orlando	9
MIX 07 Las Vegas	13
TechEd 07 Orlando	5
TechEd 07 Europe	4
Professional Developers Conference	5

Copyright ©2006 Fabrikam, Inc. [Terms of Use](#)

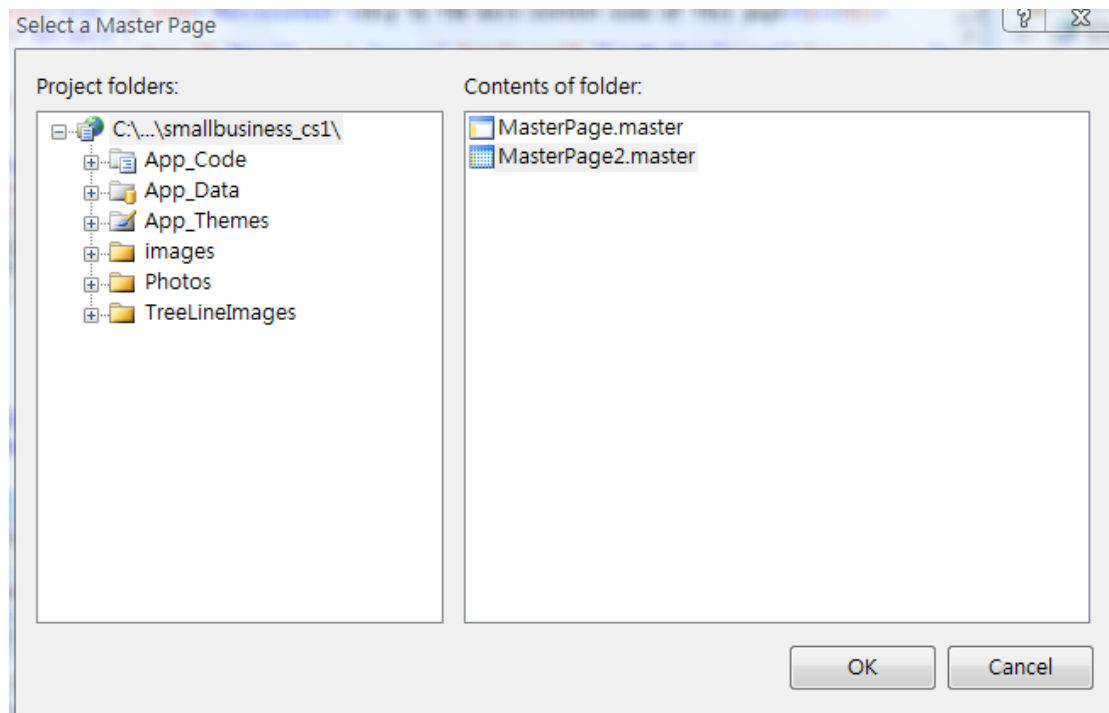
[Home](#) [Items](#) [People](#) [News](#) [About](#) [Contact](#) [Testimonials](#)

Listview

45. Select the Web Site, right click and select **Add New Item** menu command. In Add New Item Dialog Box, select **Web Form** template, checked **Select master page** checkbox and input name as **photos_linq_listview.aspx** and then click **Add** button.



46. In Select a Master Page dialog box, select MasterPage2.master and then click **OK** button.



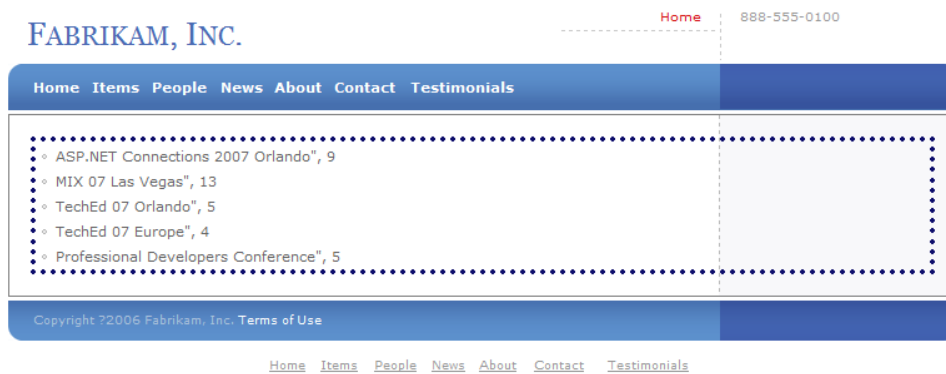
47. Put a **ListView** on the page.
48. Create **ItemTemplate** and **LayoutTemplate** for LI and UL elements respectively. Declaratively bind to the query (update to bind against ListView1 instead of GridView1).

```
<asp:ListView ID="ListView1" runat="server">
  <LayoutTemplate>
    <ul id="itemPlaceholder" runat="server">
    </ul>
  </LayoutTemplate>
  <ItemTemplate>
    <li>
      <%=Eval("Name")%> ",
      <%=Eval("Count") %>
    </li>
  </ItemTemplate>
</asp:ListView>
```

49. Create a **Page Load** event and then input below code.

```
var q =
    from a in ctx.Albums
    select new
    {
        Name = a.Name,
        Count = a.Photos.Count
    };
ListView1.DataSource = q;
ListView1.DataBind();
```

50. CTRL-F5 and show the HTML output, note clean layout and no extra 'stuff' generated.



51. Update the LINQ query to put pictures into results as part of a new anonymous type by input below code or insert **QueryAlbumWithPhotos** snippet.

```
var q =
    from a in ctx.Albums
    select new
    {
        Name = a.Name,
        Count = a.Photos.Count,
        Photos = a.Photos
    };
```


52. Nest another ListView into the previous one and show (run the application with Ctrl+F5) how it can handle LINQ nested values as well as how LINQ used the relationship to return all pictures by Album. Using img elements, the pictures will display under the LI header of the album name.

```
<asp:ListView ID="ListView1" runat="server">
  <LayoutTemplate>
    <ul id="itemPlaceHolder" runat="server">
    </ul>
  </LayoutTemplate>
  <ItemTemplate>
  <li>
    <%#Eval("Name") %>
    <asp:ListView ID="ListView2"
      runat="server"
      DataSource='<%#Eval("Photos")%>'>
    <LayoutTemplate>
      <ul id="itemPlaceHolder"
        runat="server"></ul>
    </LayoutTemplate>
    <ItemTemplate>
    <li>
      " />
      <br />
      <%#Eval("Name") %>
    </li>
    </ItemTemplate>
  </asp:ListView>
  </li>
</ItemTemplate>
</asp:ListView>
```

FABRIKAM, INC.

[Home](#)

888-555-0100

[Home](#)
[Items](#)
[People](#)
[News](#)
[About](#)
[Contact](#)
[Testimonials](#)

ASP.NET Connections 2007 Orlando



Amet Phasellus



Amet



Mauris Ornare

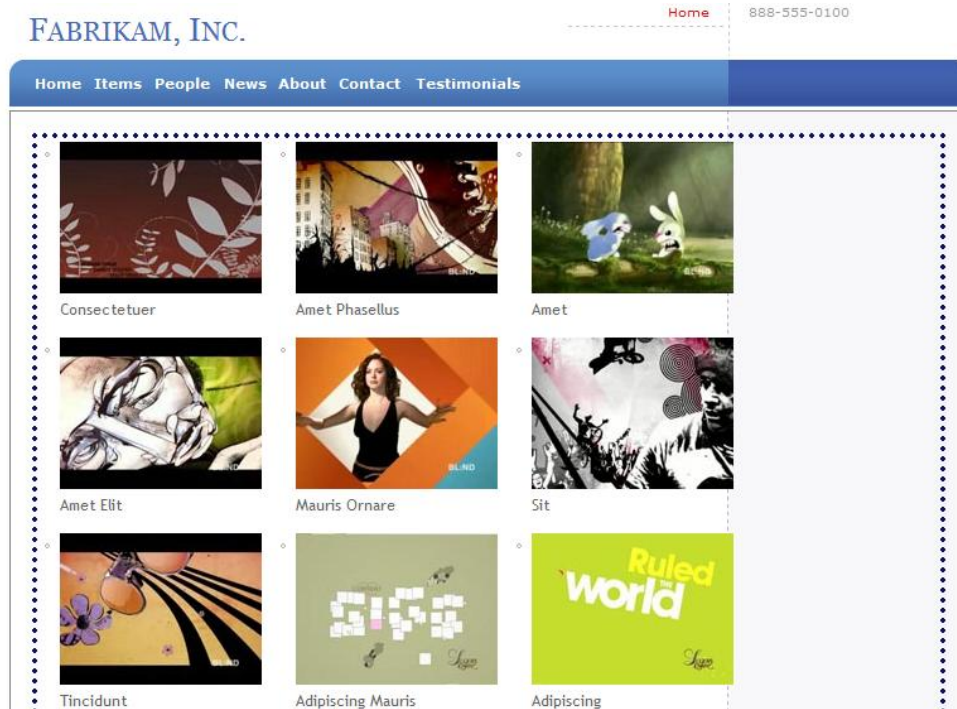
53. Remove the outer ListView and keep only the previously nested one. Clean it up, rename to ListView1 and add the class we previously created called 'photo-list' to the itemContainer in <LayoutTemplate>.

```
<asp:ListView ID="ListView1" runat="server">
  <LayoutTemplate>
    <ul id="itemPlaceHolder" runat="server"></ul>
  </LayoutTemplate>
  <ItemTemplate>
    <li class="photo-list">
      " />
      <br />
      <%=Eval("Name") %>
    </li>
  </ItemTemplate>
</asp:ListView>
```

54. Change the LINQ query to select all the Photos by input below code or insert **QueryPhotos** snippet.

```
var q = from p in ctx.Photos
  select p;
```

55. Show how we now are back to exactly the beginning, with clean code and CSS controlled UI with data bound pictures instead of static list. Rename the style **photo-list** to **photo-list**, as ListView does not support the css setting in , it need set the directly.



56. Use the **LINQPagging snippet** to insert paging code using LINQ to demonstrate Take and Skip using a query string. Place this after the DataContext declaration and before the binding to the ListView replacing the previous query.

```
int page, pagesize;

int.TryParse(
    Request.QueryString["page"],
    out page
);

pagesize = 5;

if (page == 0)
    page++;

//query database to get count
int count = ctx.Photos.Count();

int pages = count / pagesize;

if ((count % pagesize) != 0)
    pages++;

for (int i = 0; i < pages; i++)
{
    Response.Output.Write(
        "<a href={0}?page={1}>{1}</a>&nbsp;&nbsp; ",
        Request.Path,
        i + 1
    );
}

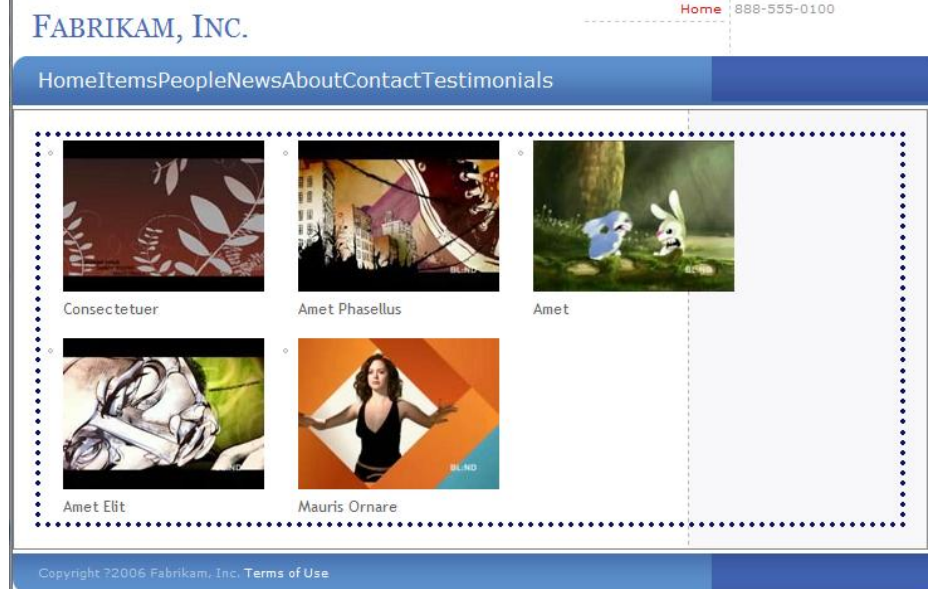
Response.Write("<br/><br/>");

var q = (from p in ctx.Photos
        select p).Skip(((page - 1) * pagesize)).Take(pagesize);
```

57. Uncomment the **.Log** line to see SQL generated and noting ROW usage as well as TOP usage.

-1- -2- -3- -4- -5- -6- -7- -8-

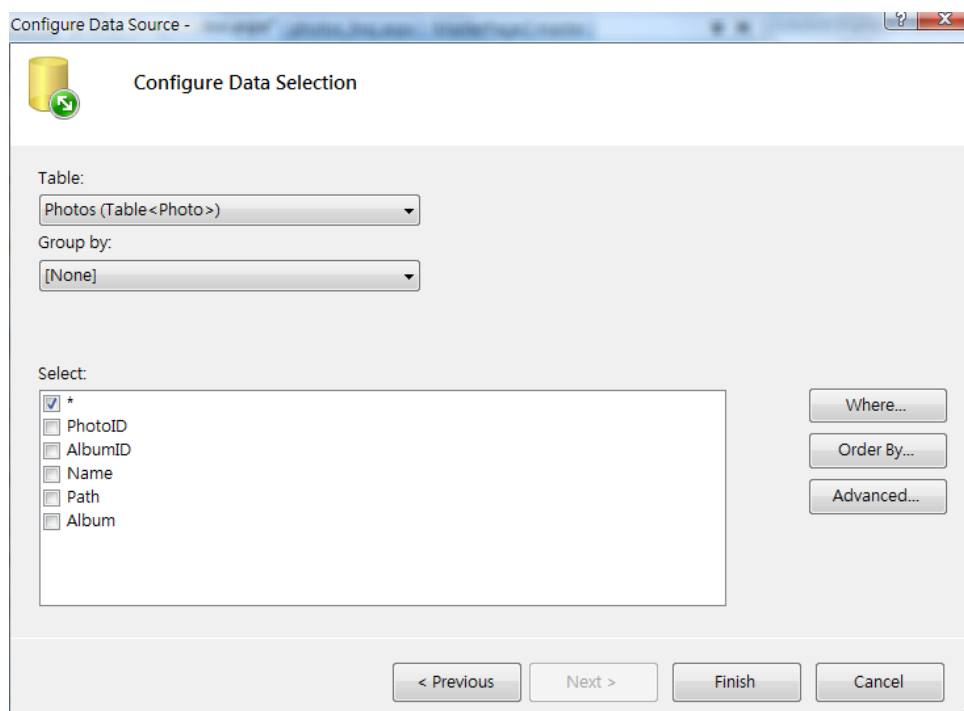
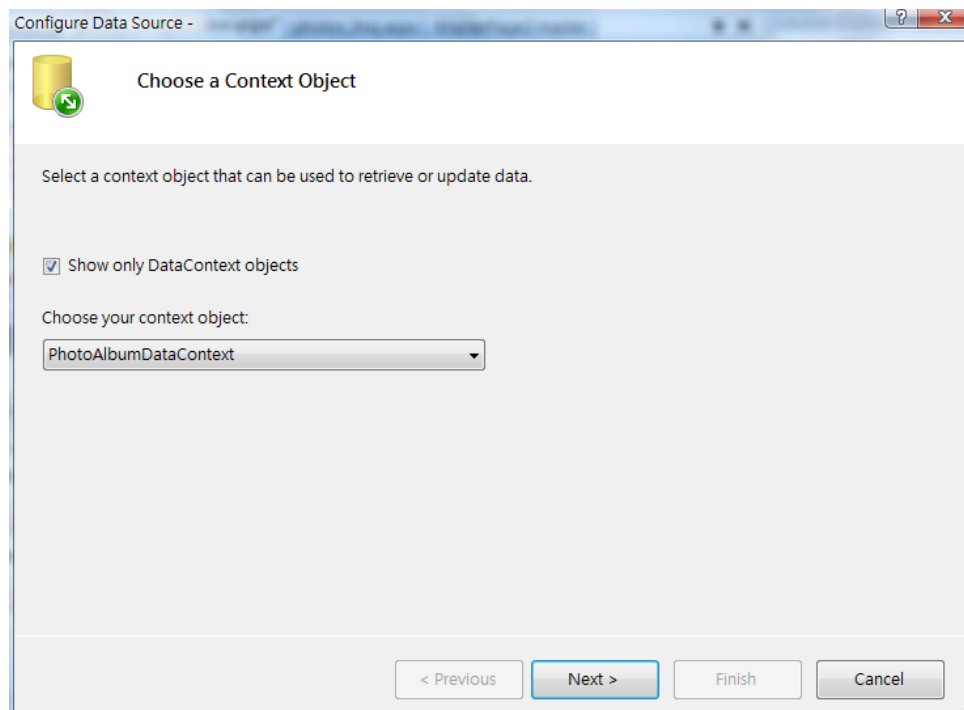
```
SELECT TOP 5 [t0].[PhotoID], [t0].[AlbumID], [t0].[Name], [t0].[Path] FROM [dbo].[Photos] AS [t0] -- Context: SqlProvider(Sql2005) Model:
AttributedMetaModel Build: 3.5.20706.1
```



58. Comment out the code now.

DataPager

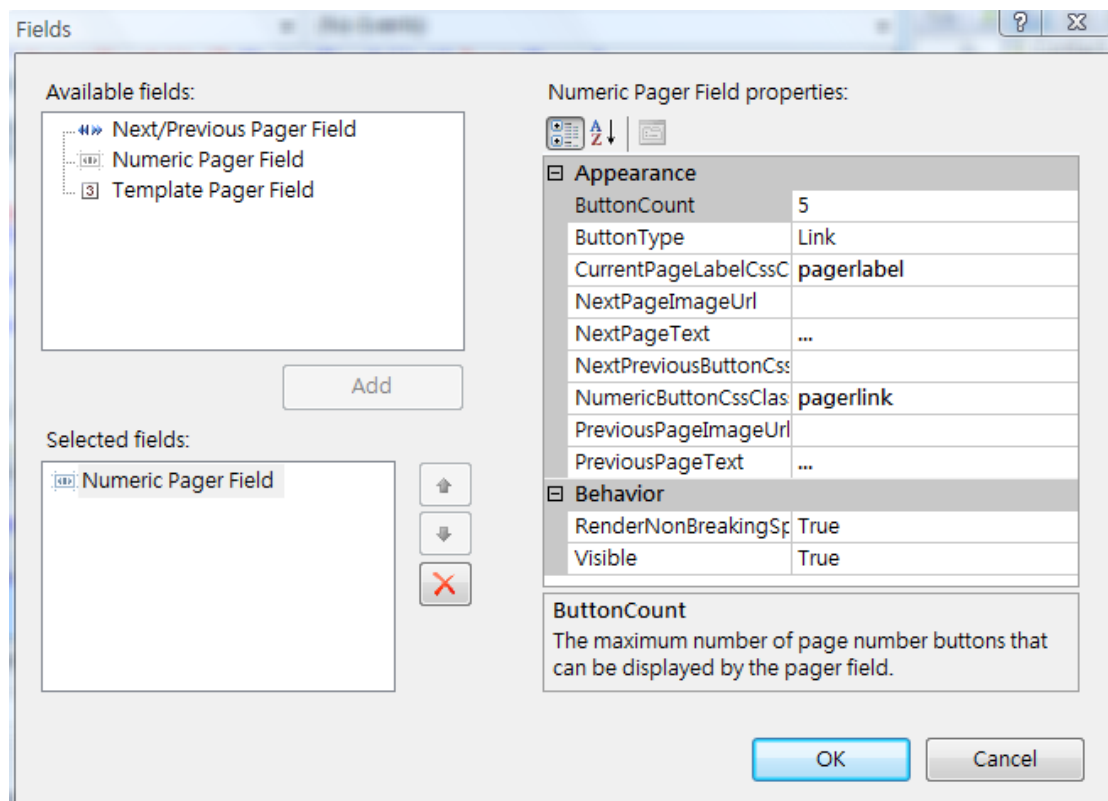
59. Add **LinqDataSource** control. In smart tag, select **Configure Data Source**.
60. In Configure Data Source Dialog Box, select **PhotoAlbumDataContext** and then select **Photos** Table and click Finish button.



61. Set the **DataSourceID** of the ListView to the new LinqDataSource1.

```
<asp:LinqDataSource ID="LinqDataSource1" runat="server"
    ContextTypeName="PhotoAlbumDataContext" TableName="Photos">
</asp:LinqDataSource>
```

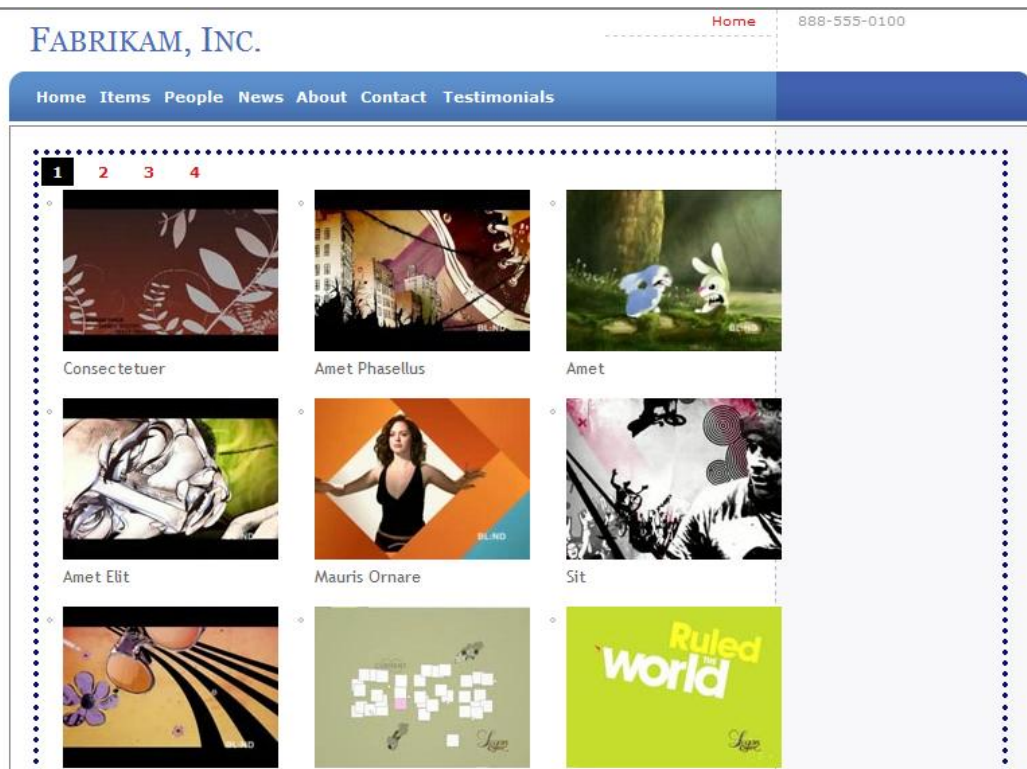
62. Add **DataPager** control and wire to ListView by setting the **PagedControlID** property to ListView1.
63. Select the DataPager. In smart tag, select **Edit Pager Fields**. Add the **Numeric Pager Field**. For styling, set the CurrentPageLabelCssClass to 'pagerlabel', and the NumericButtonCssClass to 'pagerlink'.



```
<asp:DataPager ID="DataPager1" runat="server" PagedControlID="ListView1">
    <Fields>
        <asp:numericpagerfield CurrentPageLabelCssClass="pagerlabel"
            NextPageText="..."
                NumericButtonCssClass="pagerlink" PreviousPageText="..." />
    </Fields>
</asp:DataPager>
```

64. Comment out all code in Page Load event. This was done all declaratively with no code.

65. CTRL-F5 and show the completed work in IE.



1.4 Visual Studio 2008 Client Side Improvement

- ASP.NET AJAX Support
- Improved Javascript Intellisense
- Improved JavaScript Debugging

ASP.NET AJAX Support

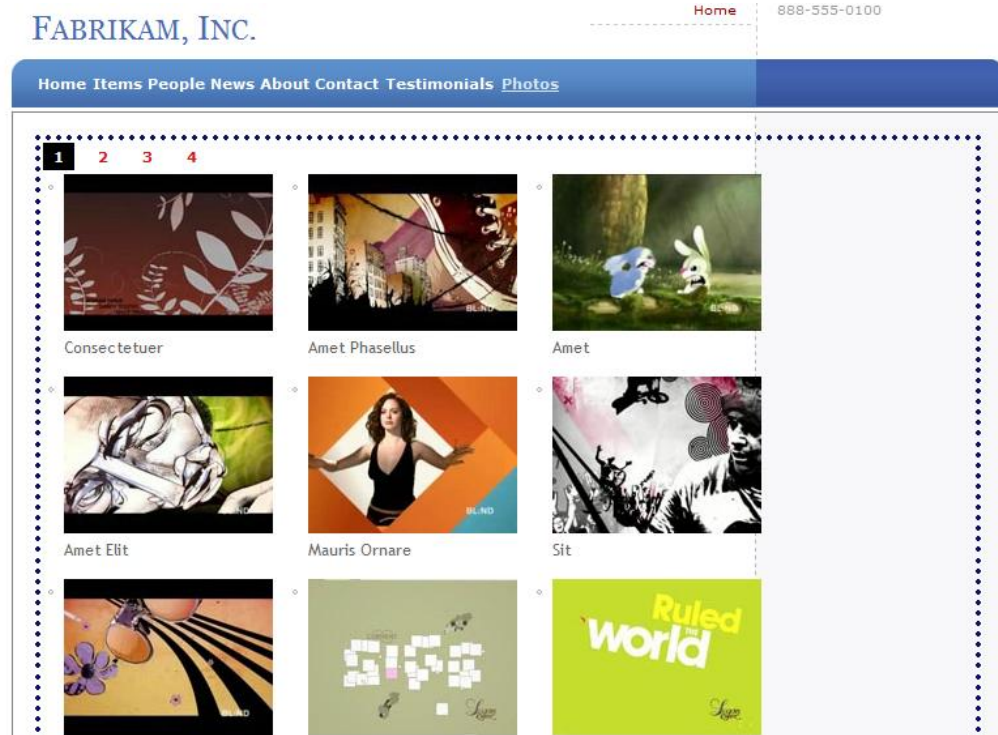
66. Add **ScriptManager** and **UpdatePanel** to the page. Put all content except LinqDataSource into update panel's template.

```
<asp:ScriptManager ID="ScriptManager1" runat="server">
</asp:ScriptManager>
<asp:LinqDataSource ID="LinqDataSource1" runat="server"
    ContextTypeName="PhotoAlbumDataContext" TableName="Photos">
</asp:LinqDataSource>
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
    <ContentTemplate>
        <asp:datapager ID="DataPager1" runat="server"
PagedControlID="ListView1">
            <Fields>
                <asp:numericpagerfield CurrentPageLabelCssClass="pagerlabel"
NextPageText="..."
                    NumericButtonCssClass="pagerlink" PreviousPageText="..."
            />
            </Fields>
        </asp:datapager>
        <asp:listview ID="ListView1" runat="server"
DataSourceID="LinqDataSource1">
            <layouttemplate>
                <ul id="itemContainer" runat="server" class="photo-list">
                </ul>
            </layouttemplate>
            <itemtemplate>
                <li>
                    <img src='<%=Eval("Path")%>' />
                    <br />
                    <%=Eval("Name") %></li>
                </itemtemplate>
            </asp:listview>
        </ContentTemplate>
    </asp:UpdatePanel>
```

67. CTRL-F5 and show the completed work in IE and now the paging would not refresh the whole page.
68. Open **web.sitemap**, add a new node at the end of sitemap.

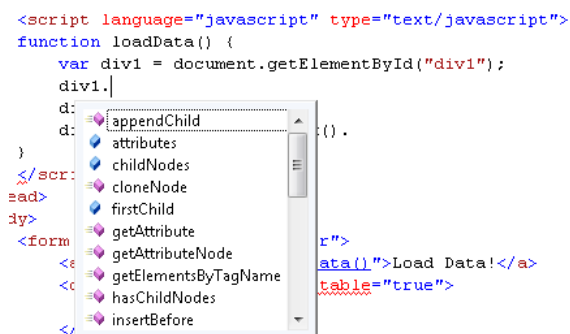
```
<siteMapNode url="~/photos_linq_listview.aspx" title="Photos" description="Our
Photos"/>
```

69. CTRL-F5 and show the completed work in IE and now a new node is added at the end of the site map.



Improved Javascript Intellisense

70. Select the Web Site, right click and select **Add Existing Item** menu command. In Add Existing Item Dialog Box, select **JavascriptSample.aspx** in Starter folder and then click **Add** button.
71. Open **JavascriptSample.aspx** page, show how simple it is.
72. Start to type in script element and note the **IntelliSense** that starts immediately.



73. Note how JS support only happens inside of client-side **<script>** tags and show how div1 is now typed as DOM element with IntelliSense.

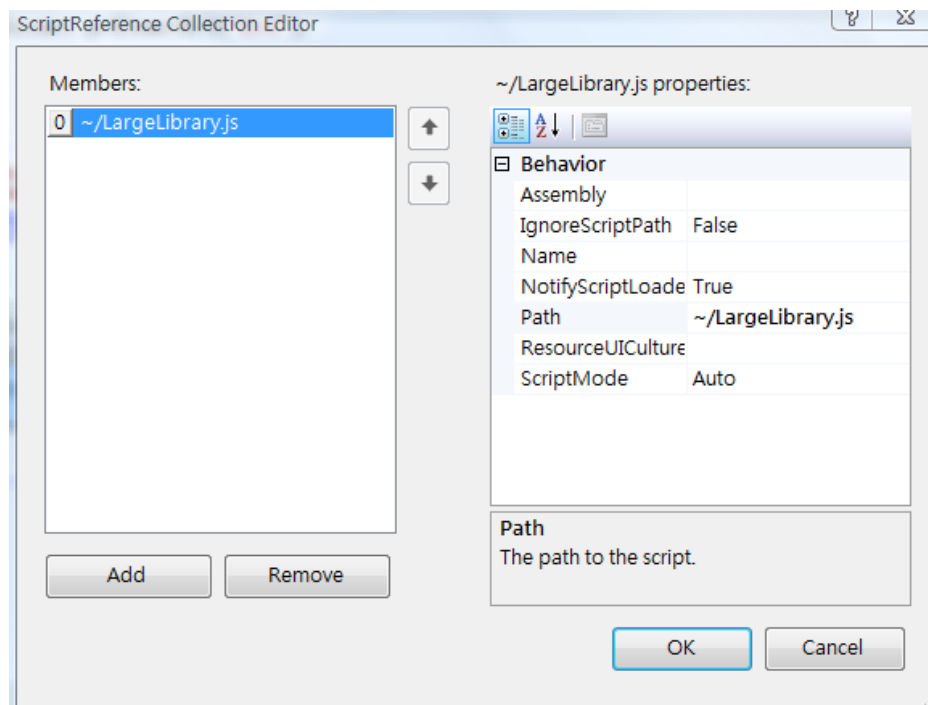
```
var div1 = document.getElementById("div1");
```

74. Next, change type of div1 by setting:

```
div1 = 10;
```

75. Show the following line where JS interprets it as a number now. Use intellisense before the assignment and how it is still a DOM element.
76. Select the Web Site, right click and select **Add Existing Item** menu command. In Add Existing Item Dialog Box, select **LargeLibrary.js** in Starter folder and then click **Add** button.
77. Open **LargeLibrary.js** and show how this contains a function with one param and returns some HTML.

78. Add **ScriptManager** to the **JavascriptSample.aspx** page. In Properties window, select Scripts property. In ScriptReference Dialog Box, Add a new reference to the LargeLibrary.js..



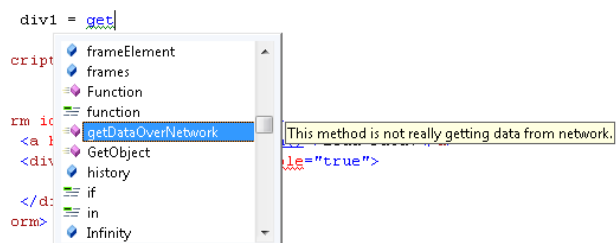
79. Now, show how the '**getDataOverNetwork()**' function shows in IntelliSense. Set div1 equal to the output of this function for remainder of script.

```
div1 = getDataOverNetwork();
```

80. Next open the .JS file and start to mark up with /// syntax. First, type an XML summary comment inside the function:

```
///<summary>this method is not really getting data from network</summary>
```

81. Go back to aspx page and show how method now has summary information in intellisense.



82. Go back to .JS file and input:

```
///<param name="path" type="String">Optional path.</param>
```

83. Go back to aspx page and show how intellisense now types and gives info for params.

```
<script src="Scripts/LargeLibrary.js" type="text/javascript">
<script language="javascript" type="text/javascript">
function loadData() {
    var div1 = document.getElementById("div1");
    var data = getDataOverNetwork();
    div1.innerHTML = Date.getDataOverNetwork(String path)
}
</script>
</script>
</script>
</script>
```

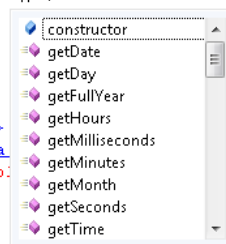
path:
Optional path.

84. Go back to .JS file and input:

```
///<returns type="Date" />
```

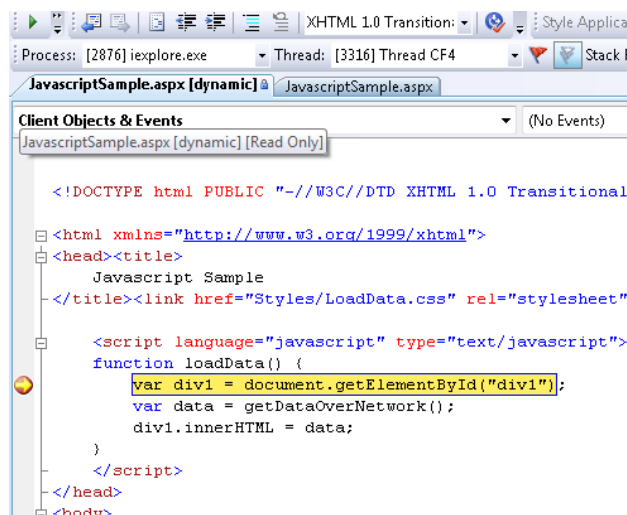
85. Go back to aspx page and note that while the return type is described wrong, show how VS08 interprets the output as DateTime now from aspx page.

```
<script language="javascript" type="text/javascript">
function loadData() {
    var div1 = document.getElementById("div1");
    var data = getDataOverNetwork();
    div1.innerHTML = data;
}
</script>
</script>
</script>
</script>
<form id="form1" runat="server">
    <a href="javascript:loadData">
        <div id="div1" contenteditable="true">
            </div>
        </div>
    </form>
```



Improved JavaScript Debugging

86. Place a breakpoint on the Javascript anywhere and hit F5.
87. If an error box about IE shows up, explain how it detects if your IE is not setup for debugging and exit error and show how to fix it.
88. Click on "Load Data" in your browser and bask in the applause as it breaks on your breakpoint.



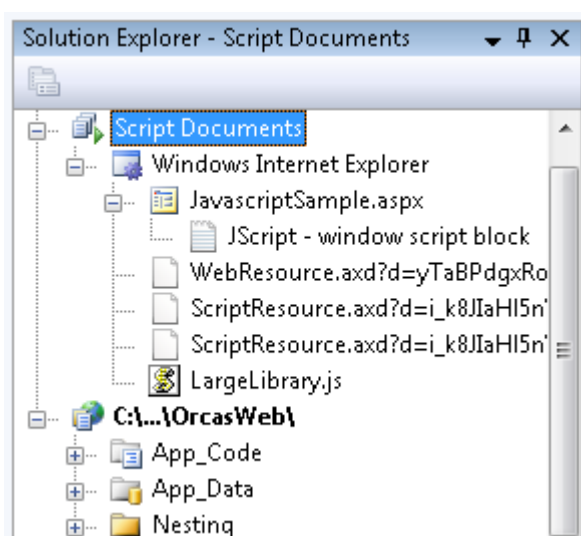
The screenshot shows the 'Client Objects & Events' window in Internet Explorer. The 'JavascriptSample.aspx [dynamic]' tab is selected. The 'Events' pane shows '(No Events)'. The 'Script' pane shows the following code:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>
  Javascript Sample
</title><link href="Styles/LoadData.css" rel="stylesheet"
<script language="javascript" type="text/javascript">
  function loadData() {
    var div1 = document.getElementById("div1");
    var data = getDataOverNetwork();
    div1.innerHTML = data;
  }
</script>
</head>
<body>

```

89. Demonstrate it is not same file now (the file is identified as JavascriptSample.aspx [dynamic] [Read Only] in the tab) and that it has broken on the file sent to the browser



90. Materials can download at http://www.fevaworks.com/course/msnet/VS08_Seminar.zip