

فصل چهارم :

معرفی کنترل های وب و نحوه استفاده از آنها در صفحات ASP.NET

مقدمه :

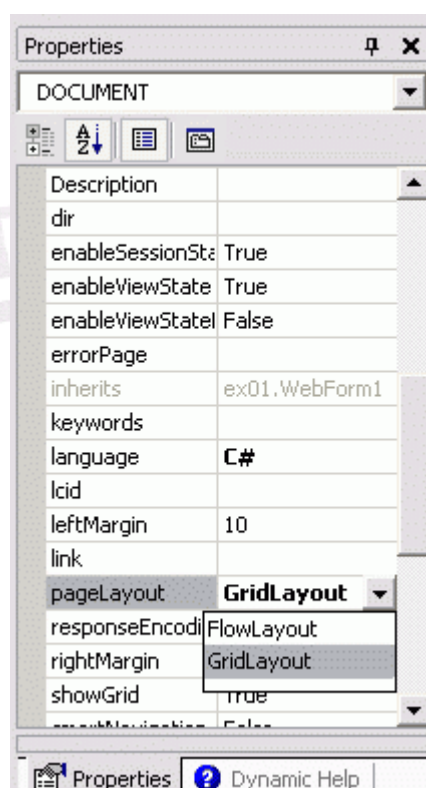
پس از مطالعه ی فصول قبلی که صرفاً مقدمه ای برای آغاز کار بودند ، از این فصل به بعد برنامه نویسی جدی ASP.NET شروع خواهد شد. در فصل جاری مروری خواهیم داشت بر نحوه ی استفاده از کنترل های وب در برنامه ها.

انتخاب Layout :

- هنگامیکه شما کنترلی را بر روی فرم قرار می دهید دو گزینه پیش رو خواهید داشت :
- Grid Layout : (پیش فرض می باشد). در این حالت مکان کنترل ها مطلق می باشند و بیش تر شبیه به طراحی ظاهر برنامه های ویندوز می باشد که با مقدار زیادی متن مخلوط نیستند .
 - Flow Layout : در این حالت کنترل ها ، نسبت به یکدیگر روی صفحه قرار می گیرند. برای مثال اگر شما کنترلی را در زمان اجرا به برنامه اضافه کنید ، کنترل های بعد از آن به سمت پایین حرکت

خواهند کرد. از این حالت بیش تر برای مواردی که مخلوطی از متن و کنترل ها نیاز است ، استفاده می گردد.

برای تنظیم این موارد در پنجره ی خواص که در سمت راست صفحه قرار دارد ، شیء Document را انتخاب کنید و سپس خاصیت PageLayout آنرا تغییر دهید (شکل ۱).



شکل ۱- انتخاب Layout نهایی فرم وب و تعیین نحوه ی قرار گیری کنترلها نسبت به هم.

انتخاب کنترل صحیح :

می توان کنترل های سرور وب و یا کنترل های HTML را روی فرم های وب قرار داد. چه تفاوتی در این زمینه وجود دارد؟ کنترل های سرور وب مزایای قابل توجهی را نسبت به کنترل های HTML ارائه می دهند که در جدول زیر مرور شده اند:

جدول ۱- مقایسه کنترل های سرور وب و کنترل های HTML.

ویژگی	Server Controls	HTML Controls
رخدادهای سرور	می توانند به رخدادهای مربوط به کنترل پاسخ دهند.	تنها می توانند به رخدادهایی در سطح صفحه عکس العمل نشان دهند.
حفظ حالت	داده ی وارد شده در کنترل بین درخواست ها ثابت باقی می ماند.	داده ها نگهداری نمی شوند و باید به صورت دستی و با برنامه نویسی اینکار صورت گیرد.
سازگاری	به صورت خودکار نوع مرورگر را تشخیص می دهد و خود را هماهنگ می کند.	هیچگونه سازگاری اتوماتیکی وجود ندارد و باید با برنامه نویسی اینکار انجام شود.
خواص	از NetFrameWork. به ارث رسیده شده است.	تنها ویژگی های مربوط به HTML در آنها وجود دارد.

سؤال : با این مقایسه چرا میکروسافت کنترل های HTML را که در فصل پیشین مرور شدند ، ارائه داده است؟ دلایل آن به شرح زیر هستند:

- مهاجرت از ASP قدیمی به ASP.NET به سادگی صورت گیرد. زیرا ASP قبلی تنها از عناصر HTML و یا همان کنترل های HTML جدید می توانست استفاده کند .
- تمام کنترل ها نیازی به رخدادهای سمت سرور و یا حفظ حالت ندارند.
- کنترل کاملی در مورد شکل نهایی صفحه با کنترل های HTML وجود دارد ، زیرا به صورت خودکار نمی تواند نوع مرورگر را حدس بزند (برخلاف کنترل های سرور وب) و خود را هماهنگ با آن نماید.

در حالت کلی ، استفاده از کنترل های سرور وب ساده تر و کارآتر می باشد. در جدول زیر کنترل هایی را که در ToolBox ویژوال استودیو دات نت می بینید با هم مقایسه شده اند و عملی را که هر کدام انجام می دهند ، مرور گردیده است.



جدول ۲- مقایسه عملکرد کنترل های وب و کنترل های HTML.

HTML Controls	Server Controls	عملکرد
Label, TextField, TextArea, PasswordField	Label, TextBox, Literal	نمایش متن
Table	DataGrid, Table	نمایش جدول
DropDown, ListBox	ListBox, DropDownList, Repeater, DataList	انتخاب از لیست
Button, ResetButton, SubmitButton	Button, LinkButton, ImageButton	انجام دستورات
Button, ResetButton, SubmitButton	CheckBoxList, CheckBox, RadioButtonList, RadioButton	تنظیم مقادیر
Image	ImageButton, Image	نمایش تصاویر
- (فقط تگ <a>)	HyperLink	حرکت بین صفحات
FlowLayout, GridLayout	Placeholder, Panel	کنترل های گروهی
-	Calendar	کار با تاریخ
-	AdRotator	نمایش تبلیغات
Horizontal rule	Literal	نمایش خط افقی
FileField	-	دریافت نام فایل از کلاینت
Input Hidden	بوسیله ی مدیریت حالت و به صورت خودکار انجام می شود.	ذخیره سازی داده ها روی صفحه
-	RequiredFieldValidator, CompareValidator, RangeValidator, RegularExpressionValidator, CustomValidator, ValidationSummary	ارزیابی داده ها

در قسمت های آتی نحوه ی استفاده از کنترل های وب مرور خواهند شد.



کار با متن :

روش های زیادی برای نمایش متن روی یک صفحه وجود دارد. برای یک متن فقط خواندنی می توان از روش های زیر استفاده کرد:

- استفاده از دستور `Response.Write("Some Text");`
- استفاده از کنترل `Label`
- استفاده از کنترل `TextBox` با خاصیت `ReadOnly` مساوی `True`
- استفاده از کنترل `Literal`

برای نمایش یک متن قابل ویرایش ، می توان از کنترل سرور `TextBox` استفاده کرد. خواص کلیدی آن در جدول زیر مرور شده اند:

جدول ۳- خواص مهم کنترل `TextBox`

خاصیت	نحوه ی استفاده
<code>Text</code>	برای دریافت متن از آن و یا نوشتن متن در آن بکار برده می شود.
<code>TextMode</code>	حالت <code>SingleLine</code> و یا <code>MultiLine</code> که مانند <code>TextArea</code> می شود و یا حالت <code>Password</code>
<code>ReadOnly</code>	در صورت <code>True</code> بودن ، کاربر نمی تواند آنرا تغییر دهد.
<code>AutoPostBack</code>	تا زمانی که <code>True</code> نشود نمی توان از رخداد <code>TextChanged</code> آن کنترل استفاده کرد و به صورت پیش فرض <code>False</code> است.

چون نحوه ی استفاده از این کنترل در طی فصول قبلی در عمل مطالعه گردید ، لزومی به تکرار آن در اینجا نمی باشد.

کار با جداول و لیست ها :

برای آراستن متن در ردیف ها و ستون ها باید از یکی از کنترل های لیست که در جدول ۲ نامبرده شدند استفاده شود. از ListBox ، DropDownList و جدول (Table) برای جداول و لیست های دینامیک استفاده می گردد. از DataGrid ، DataList و Repeater برای نمایش جدول و لیست های پیچیده مانند آنهایی که حاوی کنترل ها هستند و یا متصل به پایگاه داده اند استفاده می گردد. در جدول زیر موارد استفاده از کنترل های لیست و جدول ، مرور شده اند.

جدول ۴- کنترل های لیست و جدول ASP.NET .

کنترل	موارد کاربرد
ListBox	نمایش متنی فقط خواندنی در یک لیست با قابلیت Scroll
DropDownList	نمایش متنی فقط خواندنی در یک Dropdownlist ساده
Table	نمایش متن و یا کنترل ها در ستون ها و ردیفها
DataGrid	نمایش داده ها و کنترلهای پیچیده در جداول

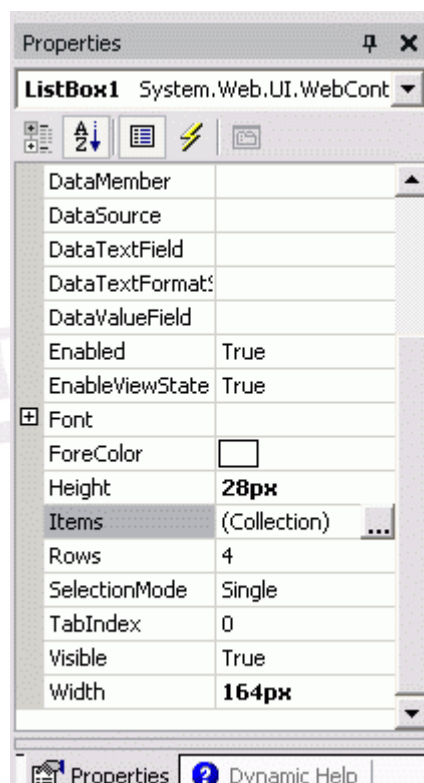
اضافه کردن آیتم ها به یک لیست یا جدول در زمان طراحی :

کنترل های ListBox ، DropDownList و Table اجازه اضافه کردن آیتم های استاتیک را در زمان طراحی می دهند.

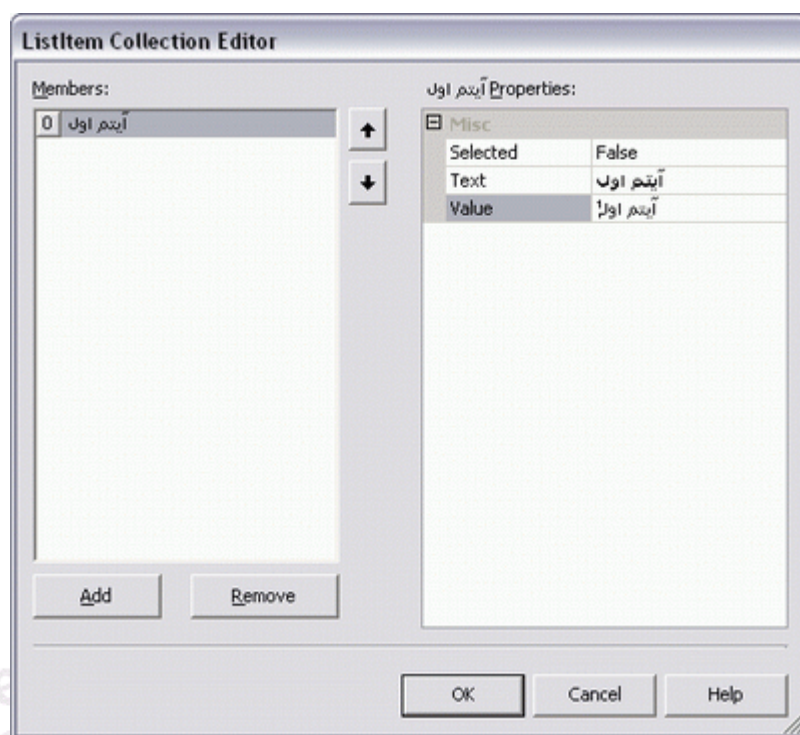
با استفاده از Collection Editor می توان آیتم های استاتیک را به یک ListBox ، DropDownList و یا جدول اضافه کرد (شکل ۲).

برای اضافه کردن آیتم های استاتیک به یک ListBox یا DropDownList ، خاصیت Items را در پنجره خواص آنها انتخاب کنید تا پنجره مربوطه باز شود (شکل ۳).

برای اضافه کردن آیتم های استاتیک به یک جدول ، خاصیت Rows آنرا در پنجره خواص کنترل انتخاب نمایید.



شکل ۲- برای اضافه کردن آیتم ها در زمان اجرا می توان از گزینه ی Items و سپس Collection Editor برای کنترل هایی مانند ListBox و DropDownList استفاده کرد.



شکل ۳- نمایی از Collection Editor یک ListBox که برای اضافه کردن آیتم های استاتیک به آن بکار برده می شود.

اضافه کردن آیتم ها به لیست یا جدول در زمان اجرای برنامه :

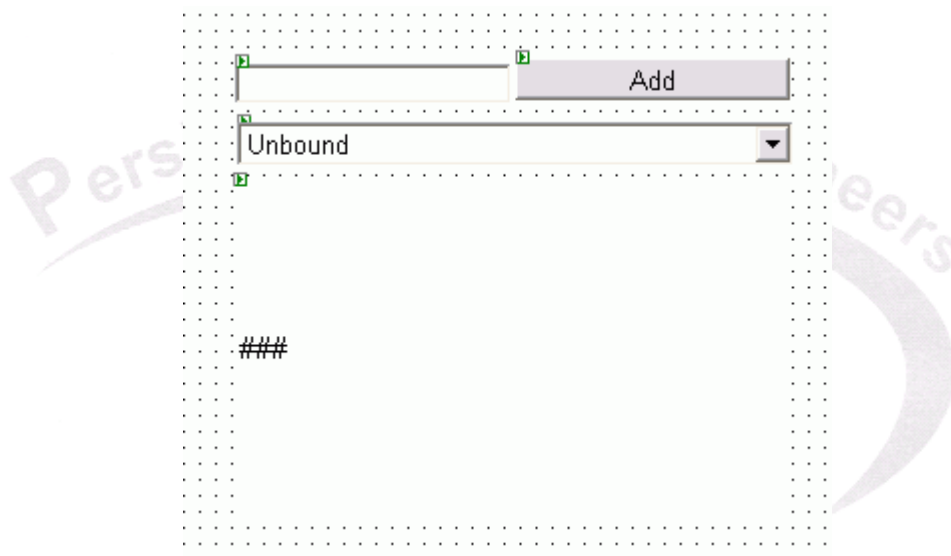
با استفاده از متد Add می توان به کلکسیون Items آنها ، عضو اضافه کرد.

`ListBox.Items.Add(....);`

بحث در مورد کنترل Table کمی مفصل تر می باشد. این کنترل تنها داده هایی را برای سلولهای جدولی ذخیره می کند که در زمان طراحی ایجاد شده اند. برای ایجاد سلولها و ردیف های بیشتر در زمان اجرا، باید دوباره جدول را با استفاده از داده های ذخیره شده در متغیر حالت ، ساخت. در این زمینه باید به یک مثال کامل توجه کرد (شکل ۴).

مثال اول :

مطابق شکل ۴ یک دکمه ، DropDownList ، TextBox ، Table روی فرم قرار دهید و نام آنها را به ترتیب به btnClick ، ddlItems ، txtAdd و tblEx01 تغییر دهید . برای طولانی نشدن فصل به راحتی می توانید به سورس همراه مراجعه کنید. در این مثال متنی که در TextBox نوشته می شود پس از کلیک شدن بر روی دکمه Add به DropDownList و Table اضافه می شود (مفهوم دوباره بازسازی کردن جدول در کد پیاده شده است) .



شکل ۴- تصویر مربوط به مثال اول.

دریافت آیتم انتخاب شده از یک لیست :

با استفاده از خاصیت SelectedItem می توان اینکار را انجام داد. برای نمونه در مثال قبل از دستور زیر می توان استفاده کرد تا به این خاصیت دسترسی پیدا کرده و در محلی مناسب از آن استفاده نمود.

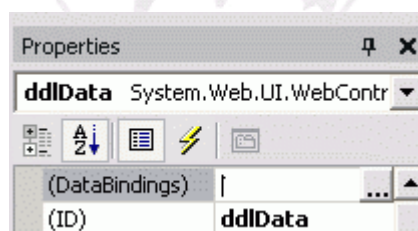
```
ddlItems.SelectedItem.Text
```

نحوه ی Data Binding ساده در کنترل لیست ها :

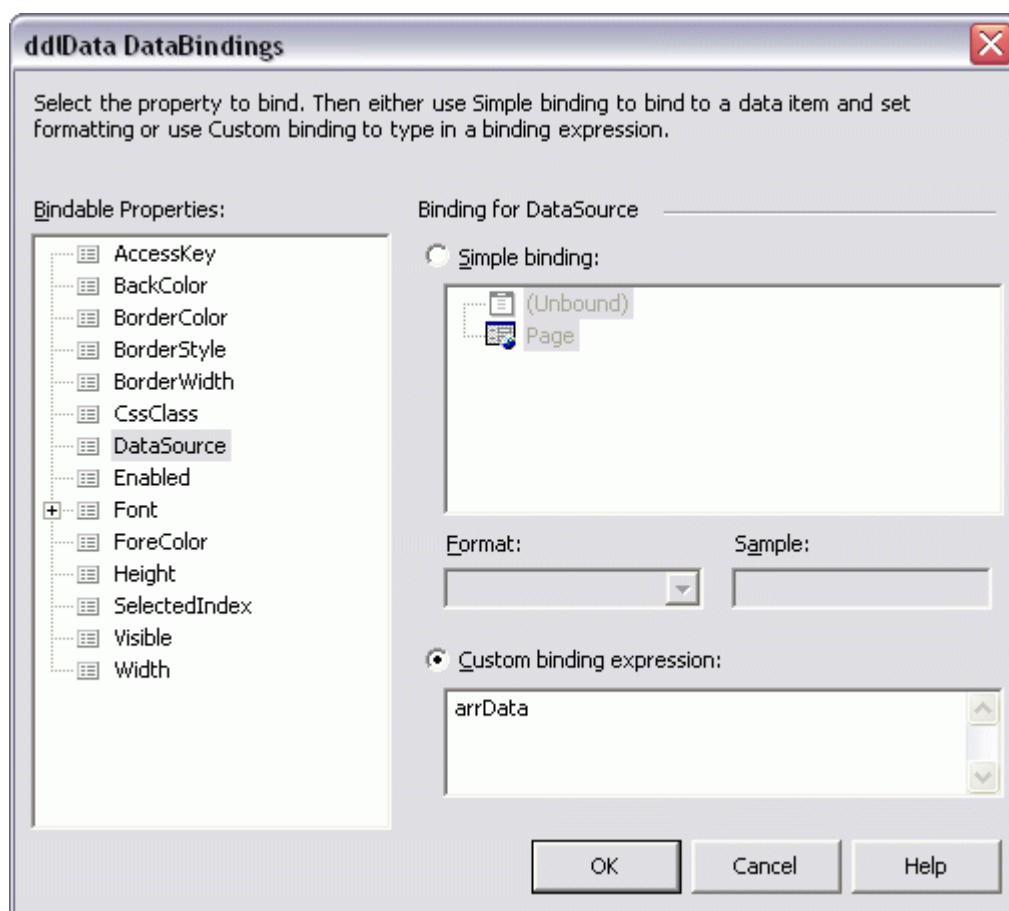
کنترل ها مقادیرشان را می توانند از هر منبع داده ای در برنامه شما دریافت کنند. برای مثال از یک بانک اطلاعاتی ، آرایه ، خاصیت یک شیء و غیره . در ساده ترین مرحله آن به مثال زیر توجه کنید:

مثال ۲ :

- ۱- یک فرم وب را با یک DropDownList درست کنید . کد مربوطه که آرایه ای به نام arrData را برای بایند کردن ایجاد می کند در سورس همراه برنامه ملاحظه نمایید.
- ۲- DropDownList را انتخاب نموده و روی دکمه خاصیت DataBinding در پنجره خواص (شکلهای ۶و۵) کلیک کنید.
- ۳- خاصیت DataSource را در این صفحه انتخاب کنید و در قسمت Custom Binding Exp نام آرایه arrData را بنویسید.
- ۴- برنامه را اجرا کنید.



شکل ۵- انتخاب گزینه ی DataBinding کنترل DropDownList .



شکل ۶- نحوه ی تعریف آرایه arrData به صورت منبع داده ای برای Bind شدن به کنترل DropDownList .

نکته :

هنگامیکه از DataBinding در کنترل های سرور استفاده می کنید ، می توان حفظ مدیریت و حالت را خاموش کنید . این مورد کارایی را افزایش می دهد ، زیرا متد DataBind به صورت اتوماتیک این مدیریت خودکار را جایگزین می کند. برای اینکار ، خاصیت EnableViewState را False کنید.

اضافه کردن آیتم ها به DataGrid ، DataList و RepeaterControls :

با استفاده از data binding می توان به کنترل ها ، آیتم ها را اضافه نمود. این کنترل ها با استفاده از Templates ظاهر خودشان را در زمان اجرا تعریف می کنند. یک Template مجموعه ای از المان های HTML است یا کنترل های سرور و یا هر دو ، که برای هر آیتم داده در کنترل تکرار خواهد شد. برای اضافه کردن آیتم ها به این کنترل ها مراحل زیر را طی نمایید.

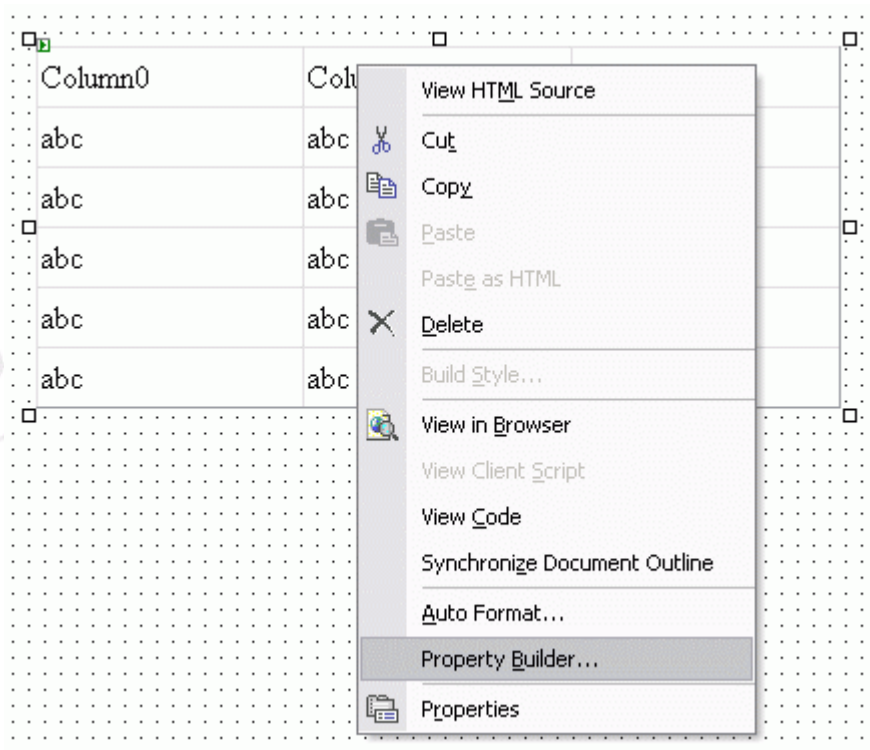
- ۱- تعریف data source .
- ۲- قرار دادن آنها روی فرم و Bind نمودن آنها به منبع داده.
- ۳- ویرایش Templates مربوط به کنترل برای اضافه کردن عناصر HTML یا کنترل های سرور که در Grid یا لیست تکرار خواهد شد.
- ۴- تنظیم خواص کنترل های سرور که در Grid یا لیست قرار داده شده اند برای bind کردن آیتم ها به آنها.

مثال ۳ :

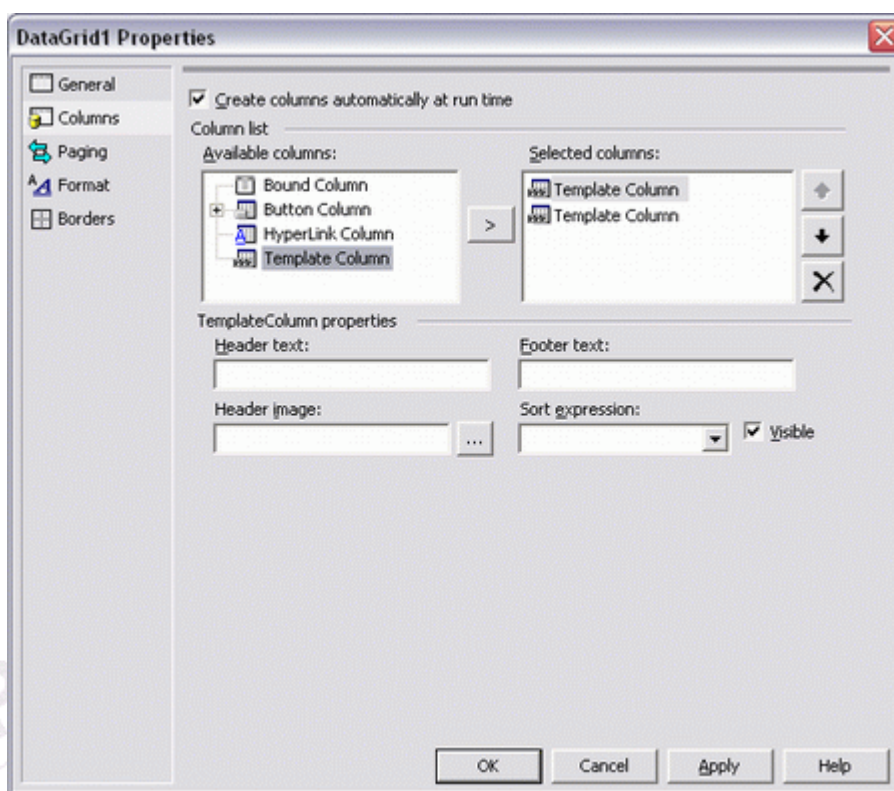
مثال زیر نحوه ی اضافه کردن ستون های Template را به DataGrid نشان می دهد و چگونگی Bind کردن کنترل های موجود در آن به یک منبع داده :

- ۱- ابتدا یک منبع داده عمومی در برنامه خود تعریف کنید (همانند برنامه قبلی) . لطفا به سورس همراه مراجعه کنید.
- ۲- کنترل DataGrid را روی فرم قرار دهید.
- ۳- ستون های Template را به آن با استفاده از Property Builder می توان اضافه کرد (روی کنترل کلیک راست کنید . شکل ۷).
- ۴- در صفحه ی ظاهر شده ، گزینه ی Columns را انتخاب کرده ، سپس ستون Template را در لیست ستون ها انتخاب کنید و بر روی دکمه Add (<) کلیک کنید. برای این مثال ۲ ستون Template را اضافه و سپس روی Ok کلیک نمایید (شکل ۸).

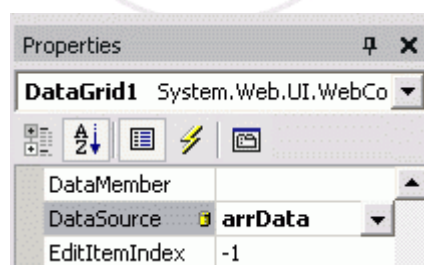
- ۵- در پنجره خواص ، خاصیت DataSource را انتخاب کنید و منبع داده را مشخص کنید (مانند مثال قبل) یعنی همان arrData (شکل ۹).
- ۶- روی DataGrid کلیک کنید و سپس Columns(0) را از منوی pop-up انتخاب نمایید. ظاهر کنترل به حالت Edit تغییر می کند (شکل ۱۰).



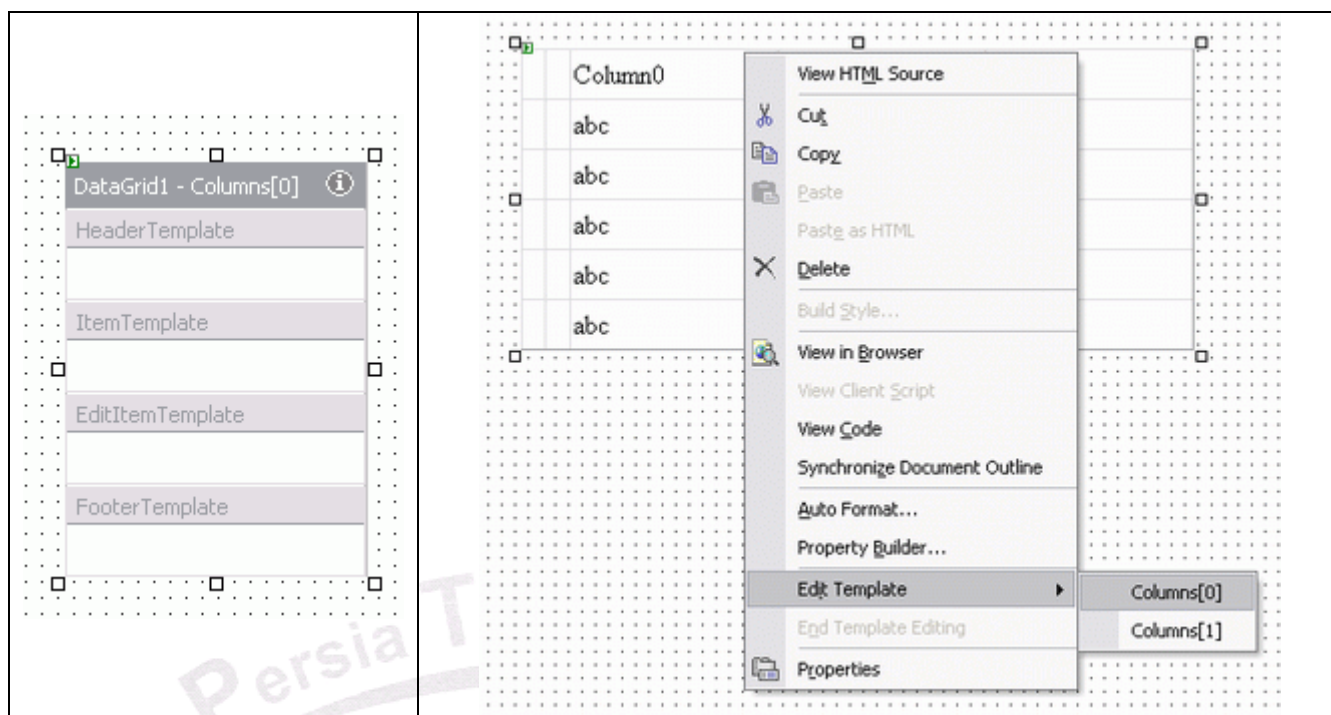
شکل ۷- انتخاب Property Builder مربوط به DataGrid .



شکل ۸- اضافه کردن دو ستون Template به دیتاگرید.



شکل ۹- انتخاب منبع داده برای دیتاگرید (که در اینجا یک آرایه می باشد).

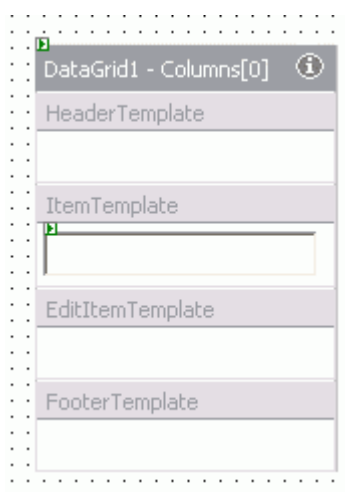


شکل ۱۰- انتخاب Columns[0] دیتا گرید برای ویرایش. در سمت چپ، این ستون را آماده ویرایش ملاحظه می کنید.

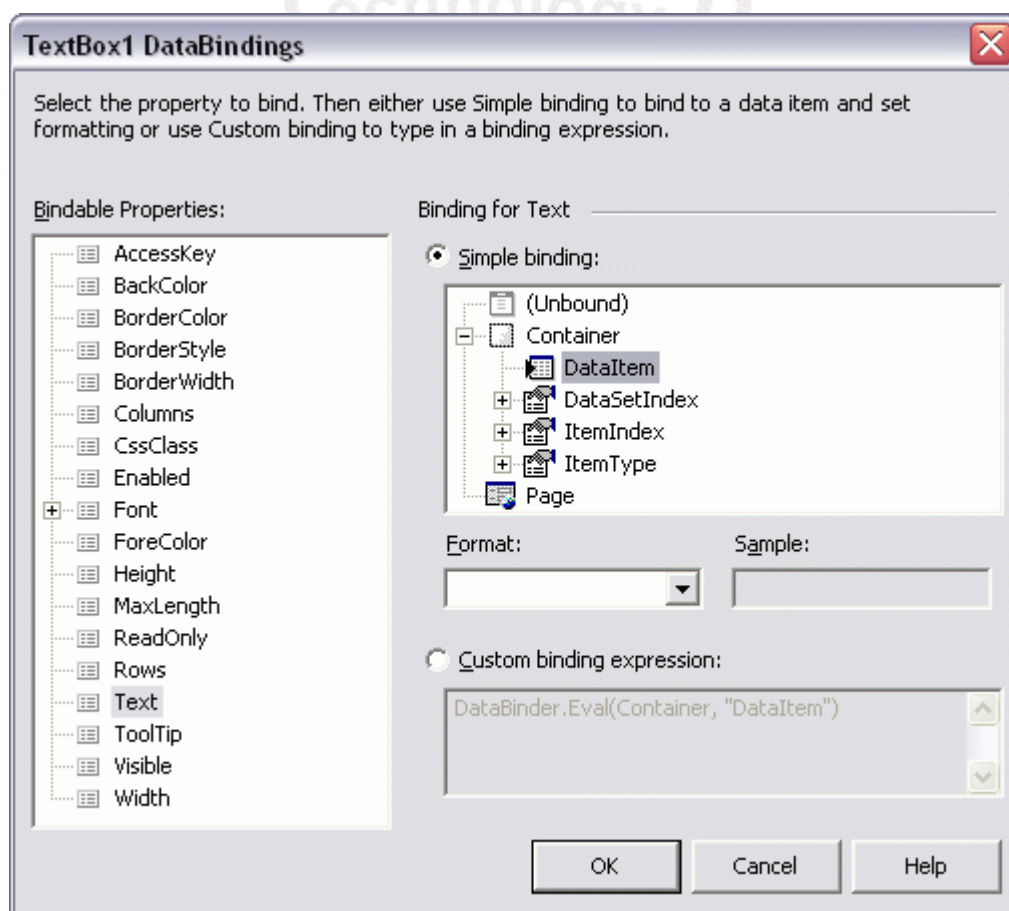
۷- سایر کنترل ها را روی فرم وب قرار دهید و سپس به Template مربوط به کنترل، Drag کنید تا به DataGrid اضافه شوند (شخصا از Cut/Paste استفاده می کنم). برای مثال یک TextBox روی فرم قرار دهید و سپس آنرا به Columns(0)، Drag کنید (شکل ۱۱).

۸- در پنجره ی خواص، خاصیت DataBinding را انتخاب کنید (مربوط به کنترل TextBox که به template اضافه کرده اید) (شکل ۱۲) و بر روی دکمه ظاهر شده کلیک کنید.

۹- در لیست خواص Bindable، خاصیتی را انتخاب کنید تا آیتم داده را دریافت کند. برای این مثال، Text را انتخاب کنید، Simple Binding را انتخاب کنید و سپس بر روی Container و DataItem کلیک کنید تا مشخص نمایید که کدام آیتم داده در خاصیت انتخاب شده قرار گیرد. روی Ok کلیک کنید تا این صفحه بسته شود.

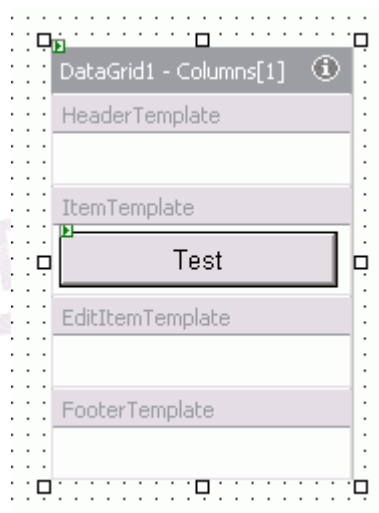


شکل ۱۱- قرار دادن یک TextBox بر روی Columns[0] در دیتاگرید.



شکل ۱۲- انتخاب نوع Binding برای کنترلی که دیتاگرید اضافه کرده ایم.

- ۱۰- دومین ستون Template را ویرایش کنید. برای اینکار بر روی کنترل DataGrid کلیک راست کنید و سپس Columns(1) را از منوی pop-up انتخاب نمایید.
- ۱۱- مراحل ۴ تا ۷ را برای این ستون تکرار کنید. برای این مثال ، یک دکمه ترسیم کنید و آنرا Drag کنید به Columns(1) در Template مربوط به DataGrid .



شکل ۱۳- قرار دادن یک دکمه در Columns[1] دیتا گرید.

- ۱۲- Template را هنگامیکه کار شما پایان یافته است ، ببندید. برای اینکار روی آن کلیک راست کنید و گزینه ی End Tempalate editing را از منوی pop-up انتخاب کنید. ویژوال استودیو کنترل های موجود در آنرا مانند شکل-۱۴ نمایش می دهد. برای تغییر خواص هر کدام از کنترل ها ، باید Template را همانگونه که تا بحال ویرایش کرده ایم ، تغییر داد.

		Column0	Column1	Column2
Databound	Test	abc	abc	abc
Databound	Test	abc	abc	abc
Databound	Test	abc	abc	abc
Databound	Test	abc	abc	abc
Databound	Test	abc	abc	abc

شکل ۱۴- شکل نهایی گرید پس از اتمام کار ویرایش .

انجام دستورات :

کنترل های سرور Button ، LinkButton و ImageButton برای انجام دستورات بکار برده می شوند. این کنترل ها سبب وقوع رخدادهایی به نام Post-Back می شوند. اینگونه رخدادهای از طرف مرورگر درخواست شده و سبب می شوند که سرور به آن پاسخ دهد. برای اینکه ترتیب رخدادهای اتفاق افتاده در یک صفحه را ببینیم به مثال زیر توجه کنید.

مثال ۴ :

یک TextBox ، ListBox و یک کنترل Button را روی فرم قرار دهید (شکل ۱۵). AutoPostBack تکست باکس و لیست باکس را True کنید. در رخداد Page_Load لیست باکس را با سه آیتم دلخواه پر کنید. سپس کدهای زیر را به رخدادهای مختلف صفحه اضافه کنید.



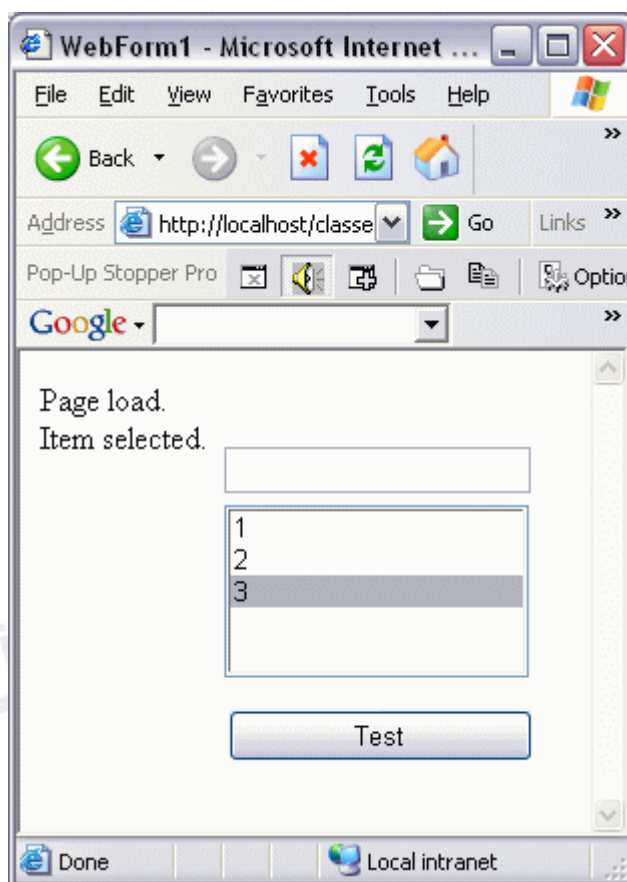
```
private void Page_Load(object sender, System.EventArgs e)
{
    Response.Write("Page load.<br>");

    if ( ! Page.IsPostBack ) // run 1 time
    {
        ListBox1.Items.Add("1");
        ListBox1.Items.Add("2");
        ListBox1.Items.Add("3");
    }
}

private void TextBox1_TextChanged(object sender, System.EventArgs e)
{
    Response.Write("Text changed.<br>");
}

private void ListBox1_SelectedIndexChanged(object sender, System.EventArgs e)
{
    Response.Write("Item selected.<br>");
}

private void Button1_Click(object sender, System.EventArgs e)
{
    Response.Write("Page load.<br>");
}
```



شکل ۱۵ - نمونه ای از اجرای برنامه ۴.

استفاده از Button و LinkButton بسیار واضح و سراسرست می باشد. کنترل ImageButton قابلیت های بیشتری را ارائه می دهد. رخداد کلیک آن حاوی آرگومانهای x و y مکانی هستند که با ماوس روی آن کلیک کرده اید و به آن ImageMaps هم می گویند.

دریافت مقادیر از کاربر :

با استفاده از کنترل های RadioButtonList ، RadioButton ، CheckBox یا CheckBoxList می توان داده های بولی و غیره را از کاربر دریافت کرد. همانند کنترل های ListBox و DropDownList



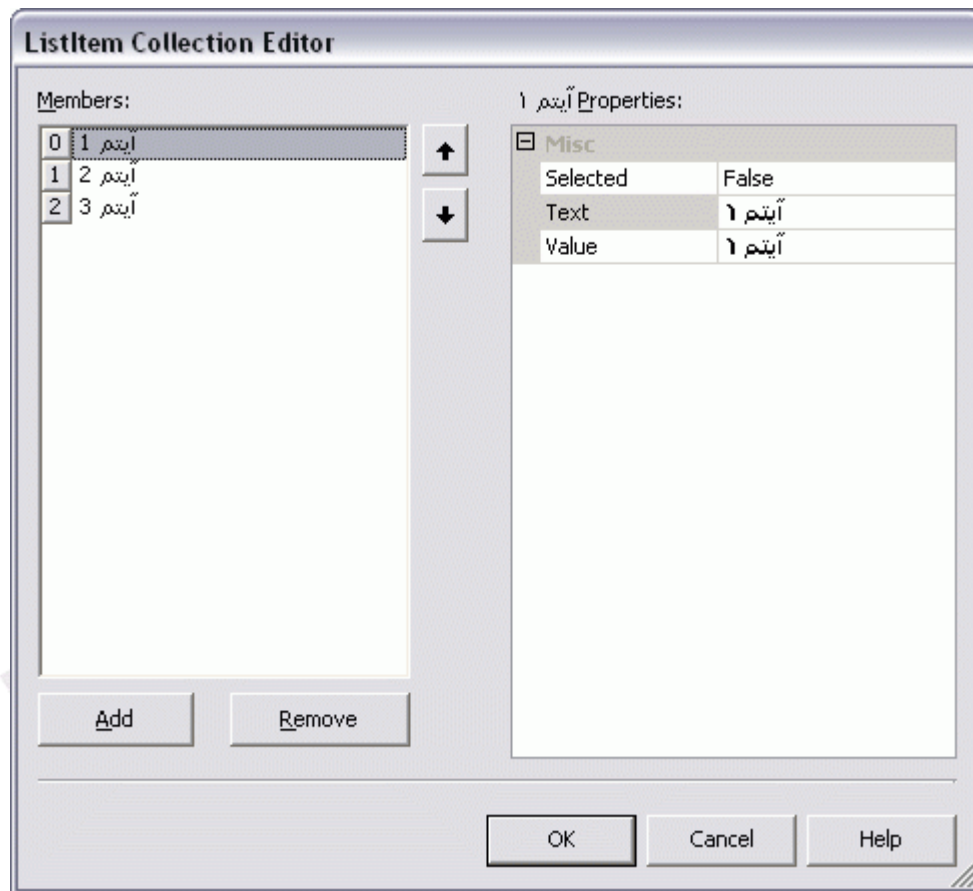
می توان از ویرایشگر Collection برای اضافه کردن آیتم به RadioButtonList یا CheckBoxList استفاده کرد. برای اینکار باید بر روی خاصیت Items آنها در پنجره ی خواص کلیک کرد .
با استفاده از خاصیت Checked در آنها می توان متوجه شد که آیا CheckBox یا RadioButton انتخاب شده اند یا خیر.

هنگامیکه شما یک RadioButton را روی فرم قرار می دهید با سایر RadioButton ها برخلاف کنترل های OptionButton ، برهم کنشی ندارد. برای این منظور باید خاصیت GroupName آنها را برای هر radioButton مشخص کرد.

برای دریافت و یا تنظیم مقادیر CheckBoxList و یا RadioButtonList ، از حلقه ی foreach می توان استفاده کرد. برای این کنترلها از خاصیت Selected آنها برای فهمیدن انتخاب شدن یا خیر، می توان استفاده کرد.

مثال ۵ :

می خواهیم یک مثال ساده برای آشنایی با نحوه استفاده از کنترل های RadioButtonList بنویسیم. یک RadioButtonList و یک دکمه را روی فرم قرار دهید. سپس روی گزینه ی Items آن در صفحه ی خواص کنترل کلیک کنید. سه گزینه ی دلخواه به آن اضافه نمایید (شکلهای ۱۶ و ۱۷).



شکل ۱۶ - اضافه کردن سه آیتم دلخواه به کنترل RadioButtonList.



شکل ۱۷ - تصویر نهایی فرم مثال ۵.

سپس از کد زیر استفاده نمایید :

```
private void Button1_Click(object sender, System.EventArgs e)
{
    foreach ( ListItem lstItem in RadioButtonList1.Items )
    {
        if ( lstItem.Selected )
            Response.Write(lstItem.Text + " is selected.<br>");
    }
}
```

نمایش گرافیک و تبلیغات :

روش های مختلفی برای نمایش گرافیک روی فرم وجود دارد.

- ۱- بعنوان پس زمینه : با استفاده از خاصیت Background فرم وب می توان یک تصویر را روی کل صفحه قرار داد. با استفاده از BackImageUrl یک کنترل Panel می توان در قسمتی از صفحه بجای کل صفحه تصویر را نمایش داد.
- ۲- بعنوان foreground : با استفاده از کنترل Image در زمان اجرا هم می توان خاصیت ImageURL آنرا تنظیم کرد.
- ۳- بعنوان یک دکمه : با استفاده از کنترل ImageButton .
- ۴- بعنوان تبلیغات : با استفاده از کنترل AdRotator برای نمایش تصاویر از لیستی از موارد تبلیغاتی .

استفاده ی معمول از تصاویر گرافیکی در صفحات ASP.NET برای تبلیغات است. کنترل AdRotator که برای این منظور بکار برده می شود از یک فایل XML برای به نوبت نمایش دادن تصاویر تبلیغی استفاده می کند. در این فایل XML آدرس تصاویر تبلیغی قرار می گیرد بعلاوه آدرس صفحه ای که اگر کاربر روی تصویر کلیک کرد باز شود و همچنین حق تقدم تصویر و سایر خواص اینگونه. در زمان اجرا ، کنترل AdRotator ، یکی از موارد تبلیغاتی موجود و لیست شده در فایل XML را انتخاب کرده و آنرا روی صفحه نمایش می دهد.



برای استفاده از کنترل AdRotator باید مراحل زیر طی شود:

- ۱- یک کنترل AdRotator روی فرم قرار دهید.
- ۲- یک فایل XML را از منوی پروژه ایجاد کنید.
- ۳- در پنجره خواص ، خاصیت TargetSchema را انتخاب کرده و سپس از منوی پایین افتادنی AdRotator Schedule file را انتخاب کنید.
- ۴- برای هر تصویری که قرار است نمایش داده شود ، تگ های <Ad> را به قسمت <Advertisement> فایل XML اضافه کنید.
- ۵- فایل XML را ذخیره کنید و به فرم وب بازگردید.
- ۶- در پنجره ی خواص کنترل AdRotator ، خاصیت AdvertisementFile را به فایل XML ایی که درست کرده اید ، ارجاع دهید.

برای مثال فایل XML زیر برای نمایش سه تبلیغ بکار می رود.

```
<Advertisements>
<Ad>
  <ImageUrl>./SampleBanner.gif</ImageUrl>
  <NavigateUrl>http://www.google.com</NavigateUrl>
  <AlternateText>Click me now!</AlternateText>
  <Keyword>ShowMe</Keyword>
  <Impressions>71</Impressions>
</Ad>
<Ad>
  <ImageUrl>./AnotherSample.gif</ImageUrl>
  <NavigateUrl>http://www.microsoft.com</NavigateUrl>
  <AlternateText>Go to Microsoft Site</AlternateText>
  <Keyword>ShowMe</Keyword>
  <Impressions>70</Impressions>
</Ad>
<Ad>
  <ImageUrl>./DoesNotExist.gif</ImageUrl>
  <NavigateUrl>http://www.microsoft.com</NavigateUrl>
  <AlternateText>Won't see me</AlternateText>
  <Keyword>DoNotShowMe</Keyword>
  <Impressions>2000</Impressions>
</Ad>
</Advertisements>
```



جدول ۵- تگ های تعریف شده برای AdRotator .

معنا	Tag
یک Ad را شروع می کند.	<Ad>
آدرس تصویری که باید نمایش داده شود.	<ImageURL>
با کلیک کردن کاربر بر روی تصویر به این آدرس هدایت می شود.	<NavigateURL>
به صورت ToolTip نمایش داده می شود و اگر تصویر به هر دلیلی نمایش داده نشود این متن جایگزین آن می شود.	<AlternativeText>
برای فیلتر کردن Ad ها به گروه های مختلف.	<Keyword>
نمایانگر احتمال نمایش فایل تبلیغی است. Ad هایی با اعداد بالاتر ، احتمال نمایش بیشتری دارند.	<Impression>

کنترل های گروهی :

برای مثال یکی از کاربردهای کنترل های گروهی این است که شما یک صفحه لاگین درست کنید و در یک قسمت صفحه کنترل های مربوط به صورت یک گروه قرار گیرند و پس از لاگین کردن آنها را مخفی کنید و قسمت دیگر صفحه را نمایش دهید.

از کنترل Panel برای اینکار استفاده می شود. روی این کنترل نمی توان کنترل ها را ترسیم کرد. باید ابتدا کنترل روی فرم قرار گیرد و سپس به روی آن Drag شود. این نوع کنترل ها از سیستم FolwLayout که در مورد آن صحبت شد ، استفاده می کنند و فقط از Space و یا Enter می توان برای تنظیم مکان آنها استفاده کرد.



کار با تاریخ :

با استفاده از کنترل تقویم می توان اطلاعات مربوط به روزها را نمایش داد. برای کار با این کنترل از رخدادهای SelectionChanged استفاده می شود و همچنین خواص SelectedDate و یا SelectedDates. رخداد SelectionChanged رخدادی post-back می باشد بنابراین به محض تغییر تاریخ ، سرور را مطلع می نماید.





تمرین :

- ۱- با استفاده از کنترل ImageButton تصویری متشکل از چهار دایره هم مرکز را روی فرم قرار دهید (بهتر است شعاع هر دایره نسبت به دایره مرکزی به مقدار ثابتی اضافه شود). برنامه ای بنویسید که مشخص کند کاربر بر روی کدام دایره کلیک کرده است.
- ۲- یک کنترل AdRotator به فرم برای نمایش دادن ۴ فایل گرافیکی تبلیغی اضافه کرده و برنامه را تکمیل نمایید.
- ۳- با استفاده از کنترل Panel یک ویزارد ۲ مرحله ای درست کنید. در مرحله ی اول نام و نام خانوادگی کاربر گرفته می شود. سپس این پنل مخفی شده و پنل دوم برای گرفتن آدرس ایمیل و محل اقامت او ظاهر می شود. سپس این پنل نیز مخفی شده و در یک پنل دیگر خلاصه اطلاعات دریافت شده نمایش داده می شود.
- ۴- یک کنترل تقویم روی فرم قرار دهید بطوریکه روز انتخاب شده از طرف کاربر در یک Label نمایش داده شود .