

بسم الله الرحمن الرحيم

## آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

آموزش ایجاد کنترل های سفارشی / ASP.NET – Custom controls

مدرس : مهندس افشین رفوآ

آموزش ایجاد کنترل های سفارشی / ASP.NET – Custom controls

ASP.NET این امکان را به کاربر می دهد که کنترل های اختصاصی خود را تعریف و ایجاد کند. کنترل هایی که توسط کاربر تعریف می شوند (یا به عبارتی **user-defined** هستند) به دو دسته ی کلی تقسیم می شوند:

User controls

Custom controls

### User Controls

**User control** عملکردی همانند صفحات کوچک یا فرم های وب (web form) ASP.NET دارند که مورد استفاده ی صفحات دیگر قرار می گیرد. کنترل مزبور از کلاس **System.Web.UI.UserControl** مشتق می شود.

User control دارای ویژگی های زیر می باشد:

دارای پسوند **ascx** می باشد.

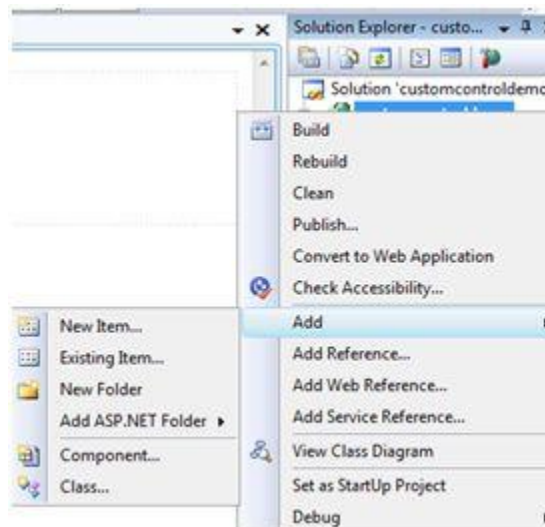
ممکن است دربردارنده ی تگ های **<html>**، **<body>** و **<form>** نباشد.

بجای **page directive** دارای **Control directive** باشد.

به منظور درک بهتر این مفهوم، یک user control ساده ایجاد می کنیم که به عنوان footer برای صفحه ی وب مورد نظر بکار گرفته می شود. به منظور ایجاد و استفاده از user control، مراحل زیر را دنبال کنید:

ابتدا یک اپلیکیشن تحت وب (web application) ایجاد کنید.

روی پوشه ی پروژه در پنجره ی Solution Explorer راست کلیک کرده و گزینه ی Add New Item را انتخاب کنید.



گزینه ی Web User Control را از پنجره ی معاوه Add New Item انتخاب کرده و آن را footer.ascx نام

گذاری کنید. در ابتدای امر، footer.ascx تنها دارای یک Control directive می باشد.

کد زیر را به فایل مورد نظر اضافه کنید:

```
<%@ Control Language="C#" AutoEventWireup="true" CodeBehind="footer.ascx.cs"
Inherits="Asp_Net_Simple_UserControl.footer" %>
<table>
  <tr>
    <td align="center"> Copyright ©2010 TutorialPoints Ltd.</td>
  </tr>
  <tr>
    <td align="center"> Location: Hyderabad, A.P </td>
  </tr>
</table>
```

به منظور افزودن user control به صفحه ی وب، ابتدا باید Register directive و نمونه ای از user control را به

صفحه مورد نظر اضافه کنید. کد زیر نشانگر content file می باشد:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Simple UserControl.aspx.cs"
Inherits="Asp_Net_Simple_UserControl.Simple_UserControl" %>
<%@ Register Src="~/footer.ascx" TagName="footer" TagPrefix="Tfooter" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```

<html xmlns="http://www.w3.org/1999/xhtml" >
  <head runat="server">
    <title>
      Untitled Page
    </title>
  </head>
  <body>
    <form id="form1" runat="server">
      <div>
        <asp:Label ID="Label1" runat="server" Text="Welcome to ASP.Net Tutorials
"></asp:Label>
        <br /> <br />
        <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
Text="Copyright Info" />
      </div>
      <Tfooter:footer ID="footer1" runat="server" />
    </form>
  </body>
</html>

```

پس از اجرا، صفحه ی وب **footer** را نمایش می دهد. این کنترل را می توان در تمامی صفحات وب سایت خود بکار  
ببرید:

Welcome to ASP.Net Tutorials

Copyright Info

Copyright ©2010 TutorialPoints Ltd.

Location: Hyderabad, A.P

به نکات زیر توجه کنید:

**Register directive** علاوه بر **tag name** یک **tag prefix** برای کنترل مورد نظر تعریف می کند.

```
<%@ Register Src="~/footer.ascx" TagName="footer" TagPrefix="Tfooter" %>
```

tag prefix و tag name باید به هنگام افزودن user control بر روی صفحه بکار گرفته شود:

```
<Tfooter:footer ID="footer1" runat="server" />
```

## Custom Control

**Custom control** ها به عنوان اسمبلی های فردی **deploy** (نصب یا مستقر) می شوند. این نوع کنترل ها در قالب

یک فایل **Dynamic Link Library** (کتابخانه با پیوند پویا) کامپایل شده، سپس مثل هر کنترل سمت سرور

(server control) ASP.NET دیگری مورد استفاده قرار می گیرد.

**Custom control** ها را می توان به یکی از روش های زیر ایجاد کرد:

با مشتق کردن **custom control** از کنترل موجود.

از طریق ایجاد یک **custom control** جدید با ترکیب کردن دو کنترل موجود.

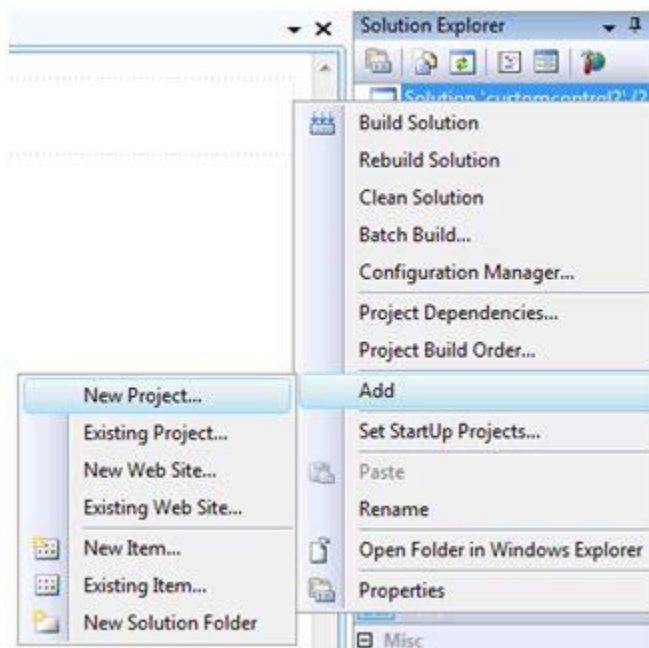
با به ارث بردن **base control class**.

**حال برای درک عمیق تر مفاهیم فوق یک custom control** ایجاد می کنیم. این کنترل یک پیام متنی را در پنجره ی

مرورگر بار گذاری کرده و به نمایش (render) می گذارد. به منظور ایجاد این کنترل، مراحل زیر را دنبال کنید:

ابتدا یک وب سایت جدید ایجاد کنید. اکنون مشابه تصویر زیر روی **solution** (و نه روی پروژه) در پنجره ی

**solution explorer** راست کلیک کنید.



در پنجره ی محاوره New Project، گزینه ی ASP.NET Server Control را از میان قالب های پروژه موجود

(**project template**) انتخاب کنید.

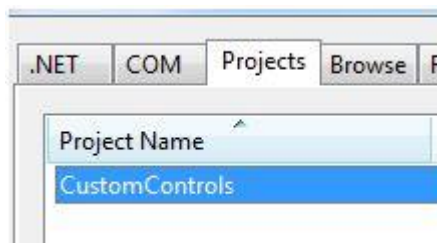


این کار یک پروژه جدید اضافه کرده و یک **custom control** به نام **ServerControl1** به **solution** اضافه می کند.

در این مثال پروژه ی خود را **CustomControls** نام گذاری می کنیم.

به منظور استفاده از کنترل مزبور، باید آن را به عنوان یک ارجاع (**reference**) پیش از ثبت (**register**) آن بر روی صفحه به وب سایت مورد نظر اضافه کنید. حال جهت افزودن ارجاع به پروژه ی موجود، روی آن پروژه (و نه **solution**) راست کلیک کرده و گزینه ی **Add Reference** را انتخاب کنید.

پروژه **CustomControls** را از **Projects tab** پنجره ی **Add Reference** انتخاب کنید. در صورت مراجعه به پنجره ی **Solution Explorer**، خواهی دید که ارجاع مربوطه در آن نمایش داده می شود.



برای استفاده از **control** در صفحه، کافی است **Register directive** را زیر **@Page directive** بکار ببرید.

```
<%@ Register Assembly="CustomControls" Namespace="CustomControls" TagPrefix="ccs" %>
```

می توان کنترل ایجاد شده را درست مثل هر کنترل دیگری بکار برد.

```
<form id="form1" runat="server">
  <div>
    <ccs:ServerControl1 runat="server" Text = "I am a Custom Server Control" />
  </div>
</form>
```

همان طور که در تصویر زیر مشاهده می کنید، خاصیت (**property**) **Text** کنترل مورد نظر پس از **execution** (اجرا) بر روی پنجره ی مرورگر به نمایش گذاشته می شود:



## کار با custom control ها

همان طور که در مثال قبلی مشاهده کردید، مقدار خاصیت Text کنترل مورد نظر تنظیم شده بود. این مقدار را به صورت پیش فرض به هنگام ایجاد شدن کنترل، اضافه می کند.

### Code behind file کنترل:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace CustomControls
{
    [DefaultProperty("Text")]
    [ToolboxData("<{0}:ServerControl1 runat=server>/{0}:ServerControl1>")]
    public class ServerControl1 : WebControl
    {
        [Bindable(true)]
        [Category("Appearance")]
        [DefaultValue("")]
        [Localizable(true)]
        public string Text
        {
            get
            {
                String s = (String)ViewState["Text"];
                return ((s == null)? "[" + this.ID + "]" : s);
            }

            set
            {
                ViewState["Text"] = value;
            }
        }

        protected override void RenderContents(HtmlTextWriter output)
        {
            output.Write(Text);
        }
    }
}
```

کد فوق به صورت خودکار برای custom control ایجاد می شود. event ها (رخدادها) و توابع را می توان به کلاس

custom control اضافه کرد.

مثال:

در این بخش همان **custom control** مثال قبلی (**ServerControl1**) را بسط می دهیم. متدی به نام **checkpalindrome** به آن اضافه می کنیم که قابلیت بررسی و پیدا کردن **palindrome** ها (وارخوانه) را به کنترل می دهد.

**Palindrome** واژه، جمله، عدد یا هر چیز دیگری گفته می شود که خواندن آن از چپ به راست یا از راست به چپ کاملاً یکسان باشد. به عنوان نمونه می توان واژه های **Malayalam**، **madam**، **Saras** را نام برد.  
برای این منظور لازم است کد **custom control** را همانگونه که زیر مشاهده می کنید، بسط دهید:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;
using System.Text;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace CustomControls
{
    [DefaultProperty("Text")]
    [ToolboxData("<{0}:ServerControl1 runat=server>/{0}:ServerControl1 >")]
    public class ServerControl1 : WebControl
    {
        [Bindable(true)]
        [Category("Appearance")]
        [DefaultValue("")]
        [Localizable(true)]
        public string Text
        {
            get
            {
                String s = (String)ViewState["Text"];
                return ((s == null) ? "[" + this.ID + "]" : s);
            }
            set
            {
                ViewState["Text"] = value;
            }
        }
        protected override void RenderContents(HtmlTextWriter output)
        {
            if (this.checkpanlindrome())
            {
                output.Write("This is a palindrome: <br />");
                output.Write("<FONT size=5 color=Blue>");
                output.Write("<B>");
                output.Write(Text);
                output.Write("</B>");
                output.Write("</FONT>");
            }
            else
            {
                output.Write("This is not a palindrome: <br />");
                output.Write("<FONT size=5 color=red>");
                output.Write("<B>");
                output.Write(Text);
                output.Write("</B>");
            }
        }
    }
}
```

```

        output.Write("</FONT>");
    }
}
protected bool checkpanlindrome()
{
    if (this.Text != null)
    {
        String str = this.Text;
        String strtoupper = Text.ToUpper();
        char[] rev = strtoupper.ToCharArray();
        Array.Reverse(rev);
        String strrev = new String(rev);
        if (strtoupper == strrev)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    else
    {
        return false;
    }
}
}
}

```

هنگامی که کد کنترل خود را اصلاح می کنید، باید **solution** را با کلیک روی **Build --> Build Solution** بسازید تا از این طریق تغییرات وارد آمده در پروژه نشان داده شوند. حال یک کنترل **textbox** و **button control** به صفحه اضافه کنید تا کاربر بتواند متن وارد کند. زمانی که کاربر روی دکمه کلیک می کند متن ورودی برای یافتن **Palindrome** کاملاً بررسی می شود.

```

<form id="form1" runat="server">
<div>
Enter a word:
<br />
<asp:TextBox ID="TextBox1" runat="server" style="width:198px"> </asp:TextBox>
<br /> <br />
<asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Check Palindrome"
style="width:132px" />
<br /> <br />
<ccs:ServerControl1 ID="ServerControl11" runat="server" Text = "" />
</div>

```

```
</form>
```

**Click event handler** کنترل دکمه، متن را از کنترل **textbox** کپی کرده و آن را در خاصیت **custom (property)** **control** می ریزد.

```

protected void Button1_Click(object sender, EventArgs e)
{
    this.ServerControl11.Text = this.TextBox1.Text;
}

```

پس از اجرا، کنترل مورد نظر **palindrome** ها را پیدا کرده و بررسی می کند.



Enter a word:

madam

Check Palindrome

This is a palindrome:

**madam**

به نکات زیر توجه کنید:

پس از اینکه ارجاع را به **custom control** اضافه می کنید، این **custom control** (کنترل سفارشی) بلافاصله به **toolbox** اضافه شده و شما می توانید آن را مانند هر کنترل دیگری به طور مستقیم مورد استفاده قرار بدهید.



در اینجا متد **RenderContents** کلاس **custom control** بازنویسی (**override**) شده و شما می توانید متدها و رخدادهای خود را به آن اضافه کنید.

متد **RenderContents** یک پارامتر از نوع **HtmlTextWriter** می گیرد که وظیفه ی اجرا و نمایش / **rendering** (بر روی مرورگر) را بر عهده دارد.