ASP.NET programming

Web Forms User Controls

User Controls Overview

- Can be easily created by using the <u>same techniques as Web Forms pages</u> are developed with.
  - Can be authored with any text editor, or to be developed using code-behind classes.
- Offer an <u>easy way to partition and reuse</u> common user interface functionality across ASP.NET Web applications.
- Offer <u>greater flexibility than server-side includes</u> by accessing the object model support provided by ASP.NET.
- <u>Runtime behaviour same</u> as that of Web Forms pages
  - Are compiled when first requested and stored in server memory to reduce the response time for subsequent requests.
- <u>Cannot be requested</u> independently
  - Must be included in a Web Forms page to work.

Including a User Control in a Web Forms Page

- In the containing Web Forms page, an **@Register** directive must be declared including:
  - **TagPrefix** attribute, which associates a prefix with the user control. This prefix will be included in opening tag of the user control element.
  - **TagName** attribute, which associates a name with the user control. This name will be included in the opening tag of the user control element.
  - **Src** attribute, which defines the virtual path to the user control file that you are including in the Web Forms page.

```
<%@ Register TagPrefix="uc"
             TagName="TestUserControl"
             Src="~/TestUserControl.ascx" %>

<uc:TestUserControl ID="ThisTestUserControl"
                    runat="server" />
```

Accessing UI Controls

- Accessing UI controls of containing page from user control
  - Using Page or Parent properties and FindControl method

```csharp
((Label)Page.FindControl("Label1")).Text = TextBox1.Text;
```

  - Not recommendable, because for reusability user control shoud not be aware of its container

- Accessing UI controls of user control from containing page
  - Using FindControl method

```csharp
Label1.Text =
    ((TextBox)ThisTestUserControl.FindControl("TextBox1")).Text;
```

  - Using public properties and object model

    User control

```csharp
public partial class TestUserControl : System.Web.UI.UserControl {
    public TextBox TextBox1 = new TextBox();
    protected void Page_Load(object sender, EventArgs e) {
        PlaceHolder1.Controls.Add(TextBox1);
    }
}
```

    Containing page

```csharp
Label1.Text = ThisTestUserControl.TextBox1.Text;
```

Handling User Control Events in Containing Page

User control

```csharp
public partial class TestUserControl : System.Web.UI.UserControl {
    public event EventHandler onCopyButtonClick;
    public TextBox TextBox1 = new TextBox();
    protected void Page_Load(object sender, EventArgs e) {
        PlaceHolder1.Controls.Add(TextBox1);
    }
    protected void CopyButton_Click(object sender, EventArgs e) {
        if (this.onCopyButtonClick != null)
            onCopyButtonClick(this, e);
    }
}
```

Containing page

```csharp
protected void Page_Load(object sender, EventArgs e) {
    ThisTestUserControl.onCopyButtonClick +=
        new EventHandler(ThisTestUserControl_onCopyButtonClick);
}
void ThisTestUserControl_onCopyButtonClick(object sender, EventArgs e) {
    this.Label1.Text = ThisTestUserControl.TextBox1.Text;
}
```