

ASP .NET Features (420-PS4-AB)

Aref Mourtada

Summer 2018

Outline

- Web Form Features
- Page Directive
- Web Controls
- User Input Validation
- Setting Defaults
- User Controls

Web Form Features | Page Directive

ASP .NET Web Forms Features

ASP .NET Control Categories

Security

Data

Navigation

Web Parts

Key Features

Master Pages

Themes/Skin

Localization

Adaptive UI

Application Service APIs

Membership

Role Manager

Personalization

Site Navigation

Caching

Management

What is in a Web Form?

- Web Forms can contain several different items:

- Directives:

```
<%@ Page Language="C#" AutoEventWireup="true" %>
```

- Code Blocks:

```
<script language="C#" runat="Server"> ... </script>
```

- Render Blocks:

```
<%=UserDetails%>
```

- Server Controls:

```
<asp:Label id="lblHelloWorld" runat="server" / >
```

What is in a Web Form? (2)

- Web Forms can contain several different items:

- User Controls:

```
<acme:Header id="myHeader" runat="server" />
```

- ASP .NET Expressions:

```
<%= $ConnectionString: NorthwindConnectionString %>
```

- Data Binding Expressions:

```
<%=# Eval("DBFileName") >
```

Page Directive

- The Page directive is added to the top of each ASP.NET page.

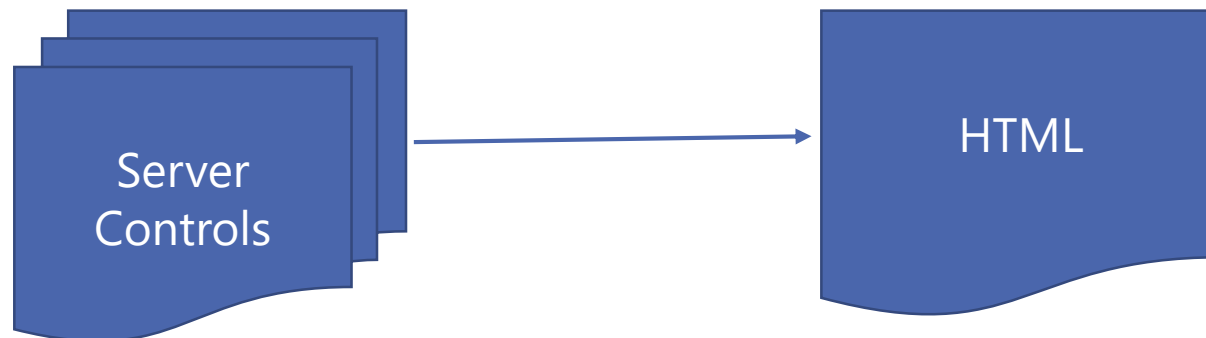
```
<%@ Page Language="C#" %>
```

- Page Key Page directive features:
 - **Title**: Set the page title
 - **Language**: Specify the page's language
 - **MaintainScrollPositionOnPostBack**: Maintain scrollbar positions
 - **CodeBehind**: Identify code file paths
 - **Trace**: Turn on or off tracing (logging)
 - **MasterPageFile, Theme**: Identify themes or master pages used.
 - **ErrorPage**: Identify an error page
 - **MetaDescription**: improve search-engine listings

ASP .NET Web Controls

Web Server Controls

- Boost productivity compared to html
- Web control \leftrightarrow Server Control
(interchangeable terminologies)
- Main purpose of all controls is to generate HTML



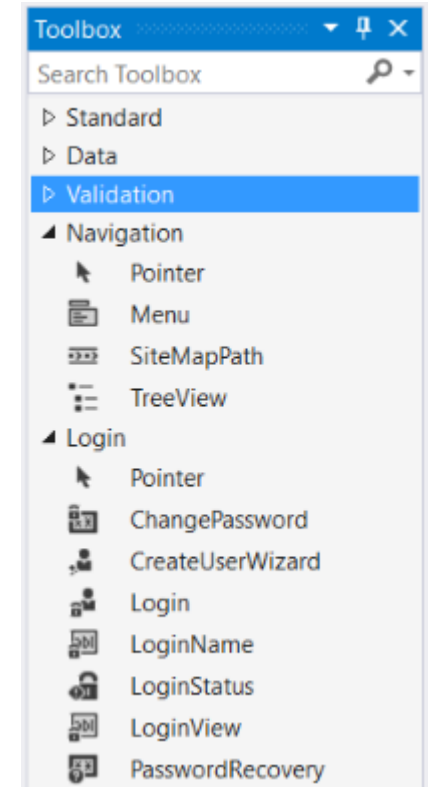
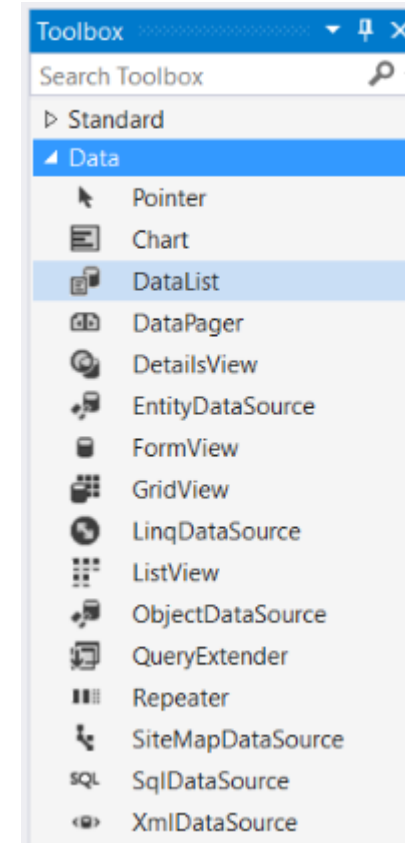
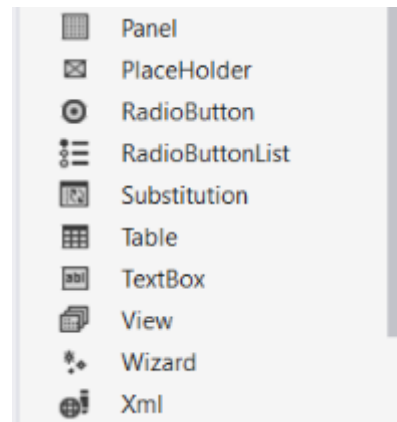
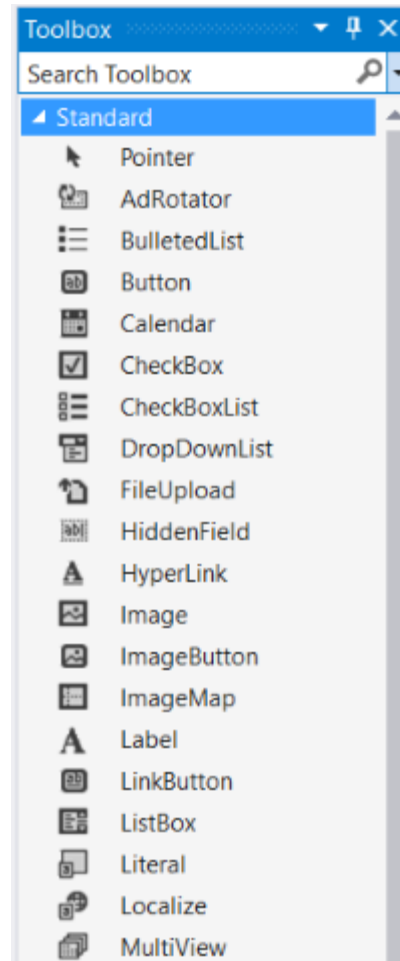
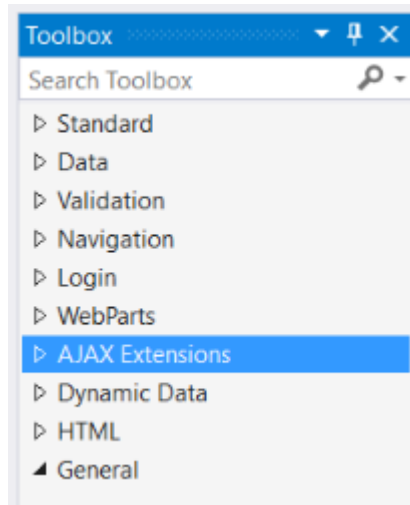
ASP.NET Web Server Controls

- **Capabilities:** ASP .NET controls provide different capabilities:
 - Collect data
 - Display data
 - Validate data
- **Structure:** Server controls are standard classes with properties, methods and events.
- **Creation:** Server controls can be *static* or *dynamically* generated.
- **Adaptive:** target browser.

ASP.NET Control Types

- ASP.NET controls are a key technology used by the Page class to dynamically generate HTML output.
- Four basic types of server controls exist:
 - **Web Server Controls:**
 - Strongly-typed programmable objects.
 - **HTML Server Controls:**
 - Similar to regular HTML elements.
 - By adding `runat="Server"`, can be manipulated from the server side
 - **Validation Controls:**
 - JavaScript
 - Used to validate Web Form submissions.
 - **User Controls:**
 - Custom controls such as headers, footers and menus.

ASP .NET Web Server Controls



ASP.NET Control Examples

- Web Server Controls:

```
<asp:TextBox id="txtName" runat="server" />
```

- HTML Server Controls:

```
<input type="hidden" id="hidVal" name="hidVal" runat="server" />
```

- Validation Controls:

```
<asp:RequiredFieldValidator id="rfvName" runat="server"  
ControlToValidate="txtName" />
```

- User Controls: (ascx files)

```
<acme:Header id="ucHeader" runat="server" />
```

Declaring Server Controls

- Server Controls are used inside a web form by prefixing the control name with "asp" namespace prefix:

```
<asp:Label id="lbOutcome" Text="Hello World"  
        runat="server" />
```

Class Name

Namespace Prefix

- Remember that use an ASP control if you need to dynamically change on server side, otherwise just use HTML.

Server Control Properties

- Server Control properties can set declaratively using attributes:

```
<asp:GridView ID="gvRecords" runat="server"  
    BorderColor="Silver"  
    BorderWidth="1"  
    GridLines="Both"  
    CellPadding="0"  
    Font-Names="Verdana"  
    Font-Size="8pt"  
/>
```

Web Server Control Events

- Server Controls expose different events that can be handled in Web Forms
- The OnClick attribute can be added to hook a Button Web Server control to a Click event handler:



```
public void btnSubmit_Click (object sender, EventArgs e)
{
    lblMessage. Text = "You clicked btnSubmit!" ;
}
```

```
<asp:Button id="btnSubmit" OnClick="btnSubnit_Click"
    runat="server" Text="Submit" />
```


Demo: Investment Calculator (1)

- Create an ASP .NET Web form to calculate investment value given a monthly investment value, the annual interest rate and the number of year. Form fields:
 - Name
 - Predefined values of monthly investment.
 - [50,100,150,200....500]
 - Annual Interest Rate
 - Number of years
 - Calculate button: get information from the form and display in a label.
 - Clear button: clear the form.
- Add a logo and let it look nice.

Demo: Investment Calculator (2)

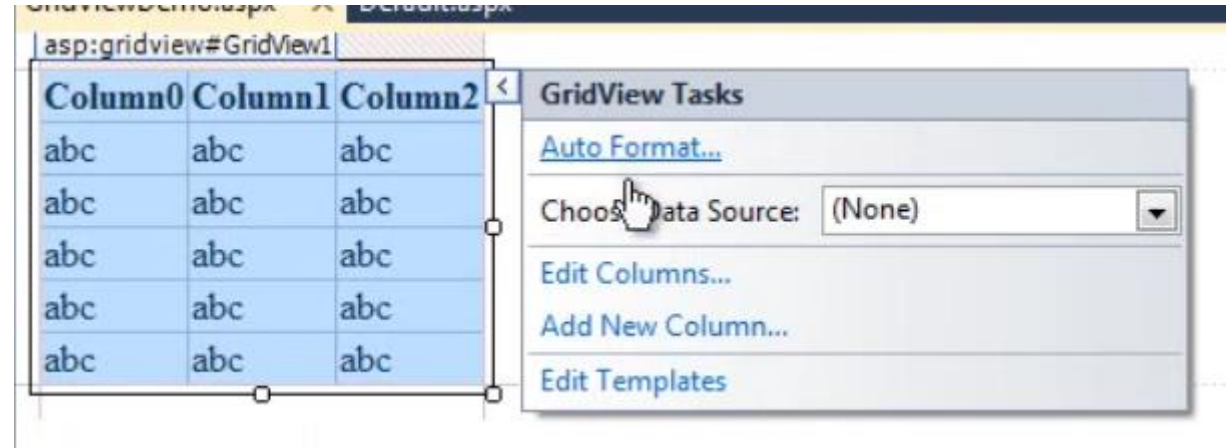
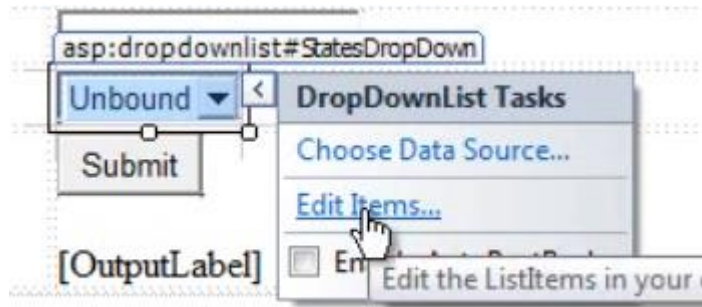
- Method for the investment value.

```
protected decimal CalculateValue(int monthlyInvestment, decimal
yearlyInterestRate, int years)
{
    int months = years * 12;
    decimal monthlyInterestRate = yearlyInterestRate / 12 / 100;
    decimal futureValue = 0;
    for (int i = 0; i < months; i++)
    {
        futureValue = (futureValue + monthlyInvestment) * (1 +
            monthlyInterestRate);
    }
    return futureValue;
}
```

Build Project

- If you make changes in the .aspx file, no need to build the project.
- Changes in the aspx file will reflect automatically in browser.
- If make changes to the .cs file, you will need to build the project to reflect the changes in the code on the ASP .NET application.

Smart Tags on Controls



Exercise: User Information

1. Create a web form to ask user for the following:
 - Name
 - Date of Birth
 - Email
 - Provenance (drop down)
 - City (drop down)
2. Print information to label
3. Add the following:
 - Clear form upon submitting (write a method)
 - Track all submitted users in a ListBox
 - Hidden value to track number of entries in the current session

Web Control Status

- The **EnableViewState** property is set to True for all controls.
- **EnableViewState** enable a control to automatically save its state for the use is round-trips (post backs).
- If you need a control to reset to default set it to False.
- Not all controls rely on View State all the time because they are rendered as standard HTML form controls in the browser once the aspx page loads.

User Input Validation

ASP .NET Validation Controls

- Validation Controls allow user input submitted through Web Forms to be validated easily.
- Supports client-side and server-side validation.
- JavaScript based.
- **Types of validation available:**
 - Required entry
 - Validating specific criteria (value range, etc.)
 - Comparing control values (text1 = text2 ?)
 - Range checking
 - Pattern matching
 - Custom Validation

ASP .NET Validation Controls

- Validation Controls that can be used in ASP .NET Web Forms include:

<asp:CompareValidator>

Multi-purpose: Comparing two values, validate dates, integers..etc.

<asp:RangeValidator>

Range of values (max & min)

<asp:RegularExpressionValidator>

Pattern matching: zip codes, IP addresses, phone numbers, email addresses

<asp:RequiredFieldValidator >

If a field is required to be enter

<asp:CustomValidator >

Write your own script

<asp:ValidationSummary>

Ties everything together

ASP .NET Validation Controls – Server Side

- Scripts at the client side can be stopped or stripped.
 - Property: EnableClientValidation (False)
- To force validation at the server:

```
Page.Validate();  
if (Page.IsValid)  
{  
    //your code  
}
```

Demo: Investment Calculator – Data Validation

- Add validations to the investment calculator created earlier:
 - Name: required
 - Annual Interest Rate: decimal [1.0 – 20.0]
 - Number of years: integer [1-45]
 - Validation summary
- Framework 4.5 and later
 - `UnobtrusiveValidationMode = UnobtrusiveValidationMode.None`

Exercise: User Information – Data Validation

- Add validation to the User Information form created earlier :
 1. Required field for all fields (including drop down menus)
 2. Date & email format validation.
 3. Add a summery validation.
 4. Add server-side validation.
- Add code to select a user from the ListBox and edit the information and update list box.

Setting Defaults

Setting a Default Button

- Setting the default button when a user hits the "enter" key can be done using the defaultButton attribute:

```
<form defaultButton="btnSearch" runat="server">
```

- The <asp:panel> control can override the defaultButton specified when the panel has focus:

```
<asp: Panel runat="server" defaultButton="btnOK">
```

...

```
</ asp : Panel>
```

Setting the Default Focus

- Setting the default focus for a page can be done using the defaultFocus attribute:

```
<form defaultFocus=" txtName" runat="server">
```

- Programmatic support for validating groups:
 - Page.SetFocus(control)
 - Page.SetFocus("ClientID")
 - txtName.Focus()

Demo: Investment Calculator

- Set default button to Calculate.
- Set default focus to name text box.

Exercise: User Information

- Set default button to submit.
- Set default focus to name text box.

User Controls

ASP.NET User Controls

- User Controls allow commonly used UI functionality to be consolidated and re-used across a site
- User Controls can be registered and used in ASP.NET Web Form pages
- Examples of user controls include.
 - Login pages used in multiple places
 - Headers or Footers in Web Forms (example: copy rights)
 - Repeating Menus

Creating User Controls

- User Controls are pages created by adding a special directive at the top of the page:

```
<%@ Control Language="C#" %>
```

- The file containing the User Control directive must be saved with a ".ascx" extension
- Cannot be viewed in browser by itself.
- To use it, just simply drag and drop into the page.

Demo: Investment Calculator – User Control

- Create a user control named *Header* and add to the Investment Calculator form.

Exercise: User Information – User Control

- Create a user control named *Header* that includes:
 - Form Name.
 - Logo
- Create a user control named *Footer* that includes:
 - Current date and time.

Dynamically Add Web Controls

- You can add a new control on the run.
 - But you do not want to be added in the right place in the page.

```
<asp:PlaceHolder ID="tempPlaceHolder" runat="server" />
```

- User Placeholder to add
 - ASP .NET Control: button, label ..etc.
 - A user defined control: `Page.LoadControl("~/path/name.ascx");`

Exercise: Shape Area Calculator

- Create an ASP .NET web form to allow user to choose among different geometric shapes and the ask for required measurement to calculate the area.
 - Choose a shape (use any suitable control)
 - Add images
 - Add proper validations
 - Square = width * width
 - Rectangle = width * height
 - Triangle = $\frac{1}{2}$ base * height
 - Circle = π * radius * radius
 - Ellipse = π * radius1 * radius2
 - Trapezoid = $\frac{1}{2}$ height (base1+base2)

Q & A

