

One page ASP.NET application

In Visual Studio 2005, a web application is called a web site. After you open the New Web Site dialog box, you select the language you want to use for the web site and you specify the location where the web site will be created.

Three location options for ASP.NET web site

Option	Description
<i>File System</i>	A web site created in a folder on your local computer or in shared folder on a network. You can run the web site directly from the built-in development server or create an IIS virtual directory for the folder and run the application under IIS.
<i>HTTP</i>	A web site created under the control of an IIS server can be on your local computer or on a computer that's available over a local area network
<i>PTP</i>	A web site created on a remote hosting server

When you start a new web site, Visual Studio creates a solution file for the web site in the default location for the solution files, which is normally MyDocuments\Visual Studio 2005\Projects.

Three ways to open a web site

- Use the *File* → *Open Project* command
- Use the *File* → *Recent Projects* command
- Use the *File* → *Open Web Site* command

How to close a project

- Use the *File* → *Close Project* command

Flow layout

By default, you develop web forms in **flow layout**. When you flow layout, the text and controls you add to a form are positioned from the left to right and from top to bottom. Because of that, the position of the controls can change when the form is displayed depending on the size and resolution of the browser window.

The best way for positioning your controls on the form is using a table as a frame for the controls. Use the *Layout* → *Insert Table* command to display the *Insert Table* dialog box. Then, set the number of rows and columns that you want in the table, set any other options that you want, and click *Ok*. To add text to a table, type the text into the cell of the table. To format the text in a table, select the text, and then use the controls in the Formatting toolbar or the commands in the Format menu to apply the formatting.

Common web server control properties

Property	Description
<i>AutoPostBack</i>	Determines whether the page is posted back to the server when the value of the control changed. Available with controls such as a check box, drop-down list, radio button, or text box. The default value is <i>False</i> .
<i>CausesValidation</i>	Determines whether the validation that's done by the validation controls occurs when you click on the button, link button, or image button. The default value is <i>True</i> .
<i>EnableViewState</i>	Determines whether the control maintains its view state between HTTP requests. The default value is <i>True</i> .
<i>Enabled</i>	Determines whether the control is functional. The default value is <i>True</i> .
<i>Height</i>	The height of the control.
<i>ID</i>	The name that's used to refer to the control.
<i>Runat</i>	Indicates that the control will be processed on the server by ASP.NET.
<i>TabIndex</i>	Determines the order in which the controls on the form receive the focus when the Tab key is pressed.
<i>Text</i>	The text that's displayed in the control.
<i>ToolTip</i>	The text that's displayed when the user hovers the mouse the control.
<i>Visible</i>	Determines whether a control is displayed or hidden.
<i>Width</i>	The width of the control.

Common properties of drop-down list and list controls

Property	Description
<i>Items</i>	The collection of <i>ListItem</i> objects that represents the items in the control. Although you can set the values for these list items at design time, normally use code to add, insert, and remove the items in a drop-down list or list box
<i>SelectedItem</i>	The <i>ListItem</i> object for the currently selected item
<i>SelectedIndex</i>	The index of the currently selected item. If no item is selected in a list box, the value of this property is -1
<i>SelectedValue</i>	The value of the currently selected item

You need to know, that when buttons are clicked, they always post back to the server. That's why they don't have AutoPostBack properties.

The aspx code for the Future Value form

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default"
%>

<!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>Future Value</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <br /><br />
            <strong><span style="font-size: 20pt; color: blue">401K
                Future Value Calculator</span></strong><br /><br />
            <table>
                <tr>
                    <td style="width: 149px">Monthly investment</td>
                    <td style="width: 71px">
                        <asp:DropDownList ID="ddlMonthlyInvestment"
                            runat="server" Width="106px">
                        </asp:DropDownList>
                    </td>
                </tr>
                <tr>
                    <td style="width: 149px">Annual interest rate</td>
                    <td style="width: 71px">
                        <asp:TextBox ID="txtInterestRate" runat="server" Width="100px">6.0</asp:TextBox>
                    </td>
                </tr>
                <tr>
                    <td style="width: 149px">Number of years</td>
                    <td style="width: 71px">
                        <asp:TextBox ID="txtYears" runat="server" Width="100px">10</asp:TextBox>
                    </td>
                </tr>
                <tr>
                    <td style="width: 149px">Future value</td>
                    <td style="width: 71px">
```

```

        <asp:Label ID="lblFutureValue" runat="server" Font-Bold="True"></asp:Label>
    </td>
</tr>
<tr>
    <td style="width: 149px; height: 25px;"></td>
    <td style="width: 100px; height: 25px;"></td>
</tr>
<tr>
    <td style="width: 149px">
        <asp:Button ID="btnCalculate" runat="server"
            OnClick="btnCalculate_Click" Text="Calculate" Width="100px" />
    </td>
    <td style="width: 100px">
        <asp:Button ID="btnClear" runat="server"
            OnClick="btnClear_Click" Text="Clear" Width="100px" />
    </td>
</tr>
</table>
<br />
<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
    ControlToValidate="txtInterestRate" Display="Dynamic"
    ErrorMessage="Interest rate is required.">
</asp:RequiredFieldValidator>
<asp:RangeValidator ID="RangeValidator1" runat="server"
    ControlToValidate="txtInterestRate"
    Display="Dynamic" ErrorMessage="Interest rate must range from 1 to 20."
    MaximumValue="20" MinimumValue="1" Type="Double">
</asp:RangeValidator><br />
<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
    ControlToValidate="txtYears" Display="Dynamic"
    ErrorMessage="Number of years is required.">
</asp:RequiredFieldValidator>
<asp:RangeValidator ID="RangeValidator2" runat="server" ControlToValidate="txtYears"
    Display="Dynamic" ErrorMessage="Years must range from 1 to 45." MaximumValue="45"
    MinimumValue="1" Type="Integer">
</asp:RangeValidator>
</div>
</form>
</body>
</html>

```

Briefly about validation controls

Validation test are typically done on the client side before the page is posted back to server. That way, a round trip to the server isn't required to display error messages if any invalid data is detected.

In most cases, client-side validation is done when the focus leaves an input control that has validators associated with it. That can happen when the user presses the *Tab* key to move to the next

control or clicks another control to move the focus to that control. Validation is also done when the user clicks on a button that has its *CausesValidation* property set to *True*.

To perform client-side validation, a browser must support Dynamic HTML, or DHTML. Because most browsers in use today support DHTML, validation can usually be done on the client side. However, validation is always done on the server too when a page is submitted. ASP.NET does this validation after it initializes the page.

When ASP.NET performs the validation tests on the server, it sets the *IsValid* property of each validator to indicate if the test was successful. In addition, after all the validators are tested, it sets the *IsValid* property of the page to indicate if all the input data is valid. You can test this property in the event handler for the event that causes the page to be posted to the server.

Common validation control properties

Property	Description
<i>ControlToValidate</i>	The ID of the control to be validated.
<i>Display</i>	Determines how an error message is displayed. Specify <i>Static</i> to allocate space for the message in the page layout. <i>Dynamic</i> to have the space allocates when an error occurs, or <i>None</i> to display the errors in a validation summary control.
<i>ErrorMessage</i>	The message that's displayed in the validation control when the validation fails.

Additional properties of a range validator

Property	Description
<i>MaximumValue</i>	The maximum value that the control can contain
<i>MinimumValue</i>	The minimum value that the control can contain
<i>Type</i>	The data type to use for range checking (<i>String</i> , <i>Integer</i> , <i>Double</i> , <i>Date</i> , or <i>Currency</i>)

Writing a programming code in a form

To add the functionality required by a web form, you add C# or VB code to its code-behind file. This code responds to the events that the user initiates on the form. This code also responds to events that occur as a form is processed.

Three ways to start an event handler

- Double-click on a blank portion of a web form to start an event handler for the Load event of the page.
- Double-click on a control in the Web Forms Designer to start an event handler for the default event of that control.
- Select a control in the Web Forms Designer, click the Event button in the Properties window (the button with the lightning bolt), and double-click the event you want to create an event handler for.

ASP.NET pages and control events

Common ASP.NET page events

Event	Method name	Occurs when ...
<i>Init</i>	<i>Page_Init</i>	A page is requested from the server. This is raised before the view state of the page controls has been restored.
<i>Load</i>	<i>Page_Load</i>	A page is requested from the server, after all controls have been initialized and view state has been restored. This is the event you typically use to perform initialization operations such as retrieving data and initializing form controls.
<i>PreRender</i>	<i>Page_Prerender</i>	All the controls events for the page have been processed but before the HTML that will be sent back to the browser is generated.

Common ASP.NET controls events

Event	Occurs when ...
<i>Click</i>	The user clicks a button, link button, or image button.
<i>TextChanged</i>	The user changes the value in a text box.
<i>CheckedChanged</i>	The user selects a radio button in a group of radio buttons or selects or unselects a check box
<i>SelectedIndexChanged</i>	The user selects an item from a list box, a drop-down list, a check box list, or a radio button list

Code for the *Click* event of the *btnClear* button

```
protected void btnClear_Click(object sender, EventArgs e)
{
    ddlMonthlyInvestment.SelectedIndex = 0;
    txtInterestRate.Text = "";
    txtYears.Text = "";
    lblFutureValue.Text = "";
}
```

The C# code for the Future Value form

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
```

```

using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
            for (int i = 50; i <= 500; i += 50)
                ddlMonthlyInvestment.Items.Add(i.ToString());
    }

    protected void btnCalculate_Click(object sender, EventArgs e)
    {
        if (IsValid)
        {
            int monthlyInvestment = Convert.ToInt32(ddlMonthlyInvestment.SelectedValue);
            decimal yearlyInterestRate = Convert.ToDecimal(txtInterestRate.Text);
            int years = Convert.ToInt32(txtYears.Text);

            int months = years * 12;
            decimal monthlyInterestRate = yearlyInterestRate / 12 / 100;

            decimal futureValue = this.CalculateFutureValue(monthlyInvestment,
                monthlyInterestRate, months);

            lblFutureValue.Text = futureValue.ToString("c");
        }
    }

    protected decimal CalculateFutureValue(int monthlyInvestment,
        decimal monthlyInterestRate, int months)
    {
        decimal futureValue = 0;

        for (int i = 0; i < months; i++)
        {
            futureValue = (futureValue + monthlyInvestment) * (1 + monthlyInterestRate);
        }
        return futureValue;
    }

    protected void btnClear_Click(object sender, EventArgs e)
    {

```



```

ddlMonthlyInvestment.SelectedIndex = 0;
txtInterestRate.Text = "";
txtYears.Text = "";
lblFutureValue.Text = "";
}
}

```

Review the HTML that's sent to the browser

The HTML for the Future Value for after a post back

```

<!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><title>
    Future Value
</title></head>
<body>
    <form name="form1" method="post" action="Default.aspx" onsubmit="javascript:return
        WebForm_OnSubmit();" id="form1">
    <div>
    <input type="hidden" name="__EVENTTARGET" id="__EVENTTARGET" value="" />
    <input type="hidden" name="__EVENTARGUMENT" id="__EVENTARGUMENT" value="" />
    <input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
value="/wEPDwUJNjg4ODQ0MzA4D2QWAgIDD2QWAgIBDxBkDxYKZgIBAgICAwIEAgUCBgIHAggCC
RYKEAUCNTAFajUwZxAFazEwMAUDMTAwZxAFazE1MAUDMTUwZxAFazlwMAUDMjAwZxAFazI1M
AUDMjUwZxAFazMwMAUDMzAwZxAFazM1MAUDMzUwZxAFazQwMAUDNDAwZxAFazQ1MAUDND
UwZxAFazUwMAUDNTAwZ2RkZla8lx4pGARce4WvHLA1jD1rdSrb" />
    </div>

    <script type="text/javascript">
    //
    var theForm = document.forms['form1'];
    if (!theForm) {
        theForm = document.form1;
    }
    function __doPostBack(eventTarget, eventArgument) {
        if (!theForm.onsubmit || (theForm.onsubmit() != false)) {
            theForm.__EVENTTARGET.value = eventTarget;
            theForm.__EVENTARGUMENT.value = eventArgument;
            theForm.submit();
        }
    }
    //]]&gt;
    &lt;/script&gt;
</pre>
<p style="text-align: right;">View state</p>
<img alt="An arrow pointing from the text 'View state' to the ViewState value in the hidden input field of the HTML code." data-bbox="670 580 800 630"/>
</div>
<div data-bbox="84 926 488 941" data-label="Page-Footer">
<p>C:\UTech\Courses\EC\Year2009-2010\Sem1\LecNotes\EC1_Lecture2.docx</p>
</div>
<div data-bbox="898 928 920 945" data-label="Page-Footer">
<p>8</p>
</div>
```



```

<script src="/Ch02FutureValue/WebResource.axd?d=NROzXF-
W_MP4tGrZCxDYTA2&amp;t=633757020057031250" type="text/javascript"></script>
<script
src="/Ch02FutureValue/WebResource.axd?d=xu1qSRlLeuORNNXYnsVpWS8x6wtgnmLybWS2E4KRa
yU1&amp;t=633757020057031250" type="text/javascript"></script>
<script type="text/javascript">
//
function WebForm_OnSubmit() {
if (typeof(ValidatorOnSubmit) == "function" &amp;&amp; ValidatorOnSubmit() == false) return false;
return true;
}
//]]&gt;
&lt;/script&gt;

&lt;div&gt;
    &lt;input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION"
value="/wEWDwKU0PLhDgLOplaxDgLW5eHmCQLW5ZXICQLX5eHmCQLX5ZXICQLY5eHmCQLY5ZXI
CQLZ5eHmCQLZ5ZXICQLa5eHmCQKrm5kLArParM00AtqshokKAq2S5aIKbyVll/1uJx9A78m1NaR3Tx
MVbcs=" /&gt;
&lt;/div&gt;
&lt;div&gt;
    &lt;img src="Images/UTECH-crest.gif" alt="Utech" /&gt;&lt;br /&gt;&lt;br /&gt;
    &lt;strong&gt;&lt;span style="font-size: 20pt; color: blue"&gt;401K
    Future Value Calculator&lt;/span&gt;&lt;/strong&gt;&lt;br /&gt;&lt;br /&gt;
    &lt;table&gt;
    &lt;tr&gt;
    &lt;td style="width: 149px"&gt;Monthly investment&lt;/td&gt;
    &lt;td style="width: 71px"&gt;
        &lt;select name="ddlMonthlyInvestment" id="ddlMonthlyInvestment" style="width:106px;"&gt;
            &lt;option value="50"&gt;50&lt;/option&gt;
            &lt;option value="100"&gt;100&lt;/option&gt;
            &lt;option value="150"&gt;150&lt;/option&gt;
            &lt;option value="200"&gt;200&lt;/option&gt;
            &lt;option value="250"&gt;250&lt;/option&gt;
            &lt;option value="300"&gt;300&lt;/option&gt;
            &lt;option value="350"&gt;350&lt;/option&gt;
            &lt;option value="400"&gt;400&lt;/option&gt;
            &lt;option value="450"&gt;450&lt;/option&gt;
            &lt;option value="500"&gt;500&lt;/option&gt;
        &lt;/select&gt;
    &lt;/td&gt;
    &lt;/tr&gt;
    &lt;tr&gt;
    &lt;td style="width: 149px"&gt;Annual interest rate&lt;/td&gt;
    &lt;td style="width: 71px"&gt;
</pre>
</div>
<div data-bbox="697 678 829 695" data-label="Text">
<p>Drop-down list</p>
</div>
<div data-bbox="83 926 488 941" data-label="Page-Footer">
<p>C:\UTech\Courses\EC\Year2009-2010\Sem1\LecNotes\EC1_Lecture2.docx</p>
</div>
<div data-bbox="898 928 919 945" data-label="Page-Footer">
<p>9</p>
</div>
```

```

        <input name="txtInterestRate" type="text" value="6.0" id="txtInterestRate"
            style="width:100px;" />
    </td>
</tr>
<tr>
    <td style="width: 149px">Number of years</td>
    <td style="width: 71px">
        <input name="txtYears" type="text" value="10" id="txtYears" style="width:100px;" />
    </td>
</tr>
<tr>
    <td style="width: 149px">Future value</td>
    <td style="width: 71px">
        <span id="lblFutureValue" style="font-weight:bold;"></span>
    </td>
</tr>
<tr>
    <td style="width: 149px; height: 25px;"></td>
    <td style="width: 100px; height: 25px;"></td>
</tr>
<tr>
    <td style="width: 149px">
        <input type="submit" name="btnCalculate" value="Calculate"
            onclick="javascript:WebForm_DoPostBackWithOptions(new
                WebForm_PostBackOptions(&quot;btnCalculate&quot;, &quot;&quot;, true,
                &quot;&quot;, &quot;&quot;, false, false))" id="btnCalculate" style="width:100px;" />
    </td>
    <td style="width: 100px">
        <input type="submit" name="btnClear" value="Clear"
            onclick="javascript:WebForm_DoPostBackWithOptions(new
                WebForm_PostBackOptions(&quot;btnClear&quot;, &quot;&quot;, true,
                &quot;&quot;, &quot;&quot;, false, false))" id="btnClear" style="width:100px;" />
    </td>
</tr>
</table>
<br />
<span id="RequiredFieldValidator1" style="color:Red;display:none;">Interest rate is
    required.</span>
<span id="RangeValidator1" style="color:Red;display:none;">Interest rate must range from 1 to
    20.</span><br />
<span id="RequiredFieldValidator2" style="color:Red;display:none;">Number of years is
    required.</span>
<span id="RangeValidator2" style="color:Red;display:none;">Years must range from 1 to
    45.</span>
</div>

```

```

<script type="text/javascript">
//
var Page_Validators = new Array(document.getElementById("RequiredFieldValidator1"),
document.getElementById("RangeValidator1"),
document.getElementById("RequiredFieldValidator2"),
document.getElementById("RangeValidator2"));
//]]&gt;
&lt;/script&gt;
</pre>
</div>
<div data-bbox="83 251 810 587" data-label="Text">
<pre>
&lt;script type="text/javascript"&gt;
//<![CDATA[
var RequiredFieldValidator1 = document.all ? document.all["RequiredFieldValidator1"] :
document.getElementById("RequiredFieldValidator1");
RequiredFieldValidator1.controltovalidate = "txtInterestRate";
RequiredFieldValidator1.errorMessage = "Interest rate is required.";
RequiredFieldValidator1.display = "Dynamic";
RequiredFieldValidator1.evaluationfunction = "RequiredFieldValidatorEvaluateIsValid";
RequiredFieldValidator1.initialvalue = "";
var RangeValidator1 = document.all ? document.all["RangeValidator1"] :
document.getElementById("RangeValidator1");
RangeValidator1.controltovalidate = "txtInterestRate";
RangeValidator1.errorMessage = "Interest rate must range from 1 to 20.";
RangeValidator1.display = "Dynamic";
RangeValidator1.type = "Double";
RangeValidator1.decimalchar = ".";
RangeValidator1.evaluationfunction = "RangeValidatorEvaluateIsValid";
RangeValidator1.maximumvalue = "20";
RangeValidator1.minimumvalue = "1";
</pre>
</div>
<div data-bbox="83 603 810 729" data-label="Text">
<pre>
var RequiredFieldValidator2 = document.all ? document.all["RequiredFieldValidator2"] :
document.getElementById("RequiredFieldValidator2");
RequiredFieldValidator2.controltovalidate = "txtYears";
RequiredFieldValidator2.errorMessage = "Number of years is required.";
RequiredFieldValidator2.display = "Dynamic";
RequiredFieldValidator2.evaluationfunction = "RequiredFieldValidatorEvaluateIsValid";
RequiredFieldValidator2.initialvalue = "";
</pre>
</div>
<div data-bbox="83 746 685 906" data-label="Text">
<pre>
var RangeValidator2 = document.all ? document.all["RangeValidator2"] :
document.getElementById("RangeValidator2");
RangeValidator2.controltovalidate = "txtYears";
RangeValidator2.errorMessage = "Years must range from 1 to 45.";
RangeValidator2.display = "Dynamic";
RangeValidator2.type = "Integer";
RangeValidator2.evaluationfunction = "RangeValidatorEvaluateIsValid";
RangeValidator2.maximumvalue = "45";
RangeValidator2.minimumvalue = "1";
</pre>
</div>
<div data-bbox="83 926 488 941" data-label="Page-Footer">
<p>C:\UTech\Courses\EC\Year2009-2010\Sem1\LecNotes\EC1_Lecture2.docx</p>
</div>
<div data-bbox="889 928 918 945" data-label="Page-Footer">
<p>11</p>
</div>
```

```

//]]>
</script>

<script type="text/javascript">
//

var Page_ValidationActive = false;
if (typeof(ValidatorOnLoad) == "function") {
    ValidatorOnLoad();
}

function ValidatorOnSubmit() {
    if (Page_ValidationActive) {
        return ValidatorCommonOnSubmit();
    }
    else {
        return true;
    }
}
//]]&gt;
&lt;/script&gt;
&lt;/form&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre>
</div>
<div data-bbox="83 533 187 551" data-label="Section-Header">
<h3>Description</h3>
</div>
<div data-bbox="112 552 909 731" data-label="List-Group">
<ul>
<li>• To view the HTML for a page, use the View → Source command in the browser's menu.</li>
<li>• The HTML that the browser receives consists entirely of standard HTML tags because all of the asp tags are converted to standard HTML when the page is rendered.</li>
<li>• View state data is stored in a hidden input field within the HTML. This data is encrypted so you can't read it.</li>
<li>• If the page contains validation controls and client scripting is enabled for those controls, the HTML for the page contains script to perform the validation on the client if the client supports DHTML.</li>
<li>• Values that the user enters into a page are returned to the browser as part of the HTML for the page.</li>
</ul>
</div>
<div data-bbox="83 926 488 942" data-label="Page-Footer">
<p>C:\UTech\Courses\EC\Year2009-2010\Sem1\LecNotes\EC1_Lecture2.docx</p>
</div>
<div data-bbox="889 927 920 946" data-label="Page-Footer">
<p>12</p>
</div>
```