

Web Controls for Financial Calculations in Visual Studio 2015

Cosma Emil

"Ovidius" University of Constanta, Faculty of Economic Sciences
ecosma@univ-ovidius.ro

Abstract

ActiveX controls can be placed in Word documents, Excel spreadsheets, PowerPoint slides or Web Forms applications. These controls are selected from the **Toolbox** and they allow user interaction. A control has adjustable attributes (properties) and it detects events (it is "alive"). Controls are associated with code sequences, in the form of certain procedures, called event procedures.

Key words: web, event, control, module.

J.E.L. Classification: C63.

1. Introduction

Microsoft Studio is an application that allows the user to employ advanced calculation means. **Microsoft Visual Basic** is used to improve its default calculation functions. As it relies on visual programming, it allows the users/programmers to assemble the program visually, in a window, using components/elements found in a **Toolbox**, which are useful when creating applications. Among others, the list of instruments includes a text editor for the source program, a language compiler and other facilities to test and verify the application. If **Microsoft Studio** is configured, **Basic** is gets installed as well. This allows you to write code. Therefore, to employ **Basic**, you must launch the **Studio** editing environment and, if you want to write code, you must open **Visual Basic**.

2. Web Forms

The process of creating an **ASP.NET (Web Forms)** application is similar to that of **Windows Forms** Type projects. The steps leading to the creation of a new application are:

File ➤ **New Web Site** ➤ **Visual Basic** ➤ **ASP.NET Empty Web Site**

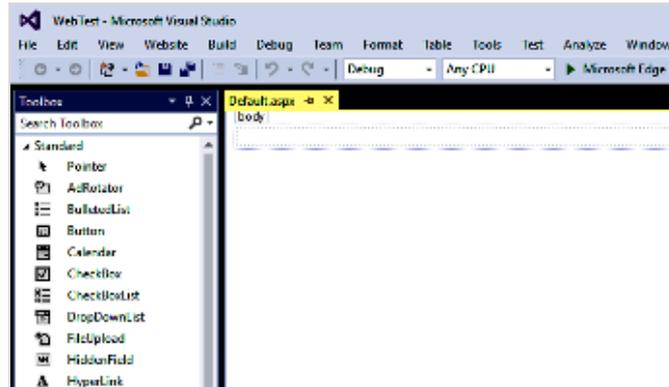
In the **Web Location** box, the user specifies the name of the folder where the project files will be placed (**WebTest**, for instance). Implicitly, the work folder is **Documents\Visual Studio 2015**.

In order to create a **Web** form/document:

Website ➤ **Add New Item** ➤ **Visual Basic** ➤ **Web Form** ➤ 

The **Design (Default.aspx)** file of the editing environment for **Web** forms is where the controls are placed on the form. When a form (document) is active in the editing environment, the **Toolbox** offers controls and components that can be used in the application. A form can contain many controls (objects), and it could also be considered an object itself. The **Properties** window allows the user to select the object and the particular property they wish to edit (for instance, the **Text** property, which determines the title (**Name**) of a form or object (Figure no. 1.).

Figure no. 1. Web Form Window



Editing **Web** forms is similar to editing texts – it involves using the **Enter** and **Space** keys. By **double-clicking** on the control or using your **mouse to drag@drop**, the controls can be included in the text.

3. Web Controls

One can add **Web controls** to **Web Forms**, next to the data they control, so that the act of operating them can be effectuated in a way that is as smooth and explicit as possible.

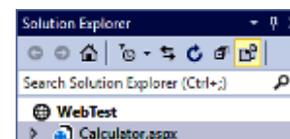
The most important **Web controls**:

- A Label** The **Label** Control displays descriptive texts (titles, short explanations and so on). This control is also useful when it comes to describing other controls, like **ListBox** or **TextBox**. The messages displayed using this type of control could be an answer in real time to an event or process of the running application.
- ab Button** The **Button** control is inserted with the purpose of running a macro and the managed event is a **Click event**. The text displayed on the button is related to the **Text** property, while its name is related to the **Name** property. If the text exceeds the length of the button, it is separated in lines (in this case, one might have to adjust the height of the button).
- abl TextBox** This control is one of the most commonly used controls when it comes to programming graphical interfaces, due to the fact that it offers multiple advantages and alternatives, both to the user, when inserting data, as well as the programmer, when manipulating these data. The aspect of the text is determined by the **Font** and **ForeColor** properties. Similar to other controls, these properties can be edited using a source code, as the application is running. A text box can be used in an active way (the user can access its content) or in a passive way (the user can only see its content, without being able to edit it, since the text is protected).

4. Examples

4.1 Arithmetic calculations

- ☞ Create a **Web** document.
- ☞ The name of the document appears in the **Solution Explorer (Default.aspx)** window. **Rename** it to **Calculator.aspx**.



☞ Activate the **Design** file. The **DOCUMENT** control appears in the list of controls of the **Properties** window.

☞ Introduce the **TextBox**, **Label** and **Button** controls (Figure no. 2., Table no. 1).

Figure no. 2. Document (Calculator) in Design Mode

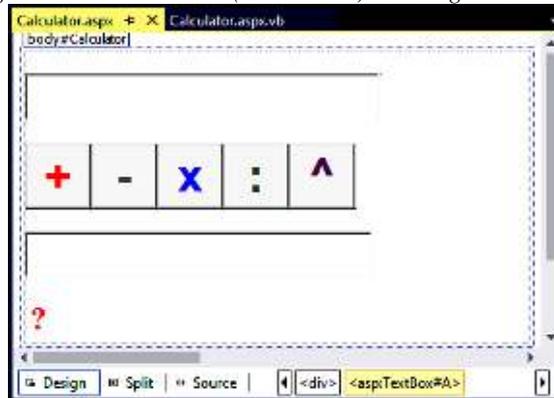


Table no. 1 Controls for Document (Calculator)

<i>Controls</i>	<i>Properties</i>
DOCUMENT	ID Calculator Title Calculator
TextBox	ID A Width 288px Height 34px Font, Bold True Font, Size 15pt
TextBox	ID B Width 288px Height 34px Font, Bold True Font, Size 15pt
Label	ID R Text ? Width 288px Height 34px Font, Bold True Font, Size 15pt
Button	ID Gat Height 55px Width 55px Font 30pt Font, Size 30pt ForeColor Red
Button	ID Dow Height 55px Width 55px Font 30pt Font, Size 30pt ForeColor Gray
Button	ID Mul

Button	Height	55px
	Width	55px
	Font	28pt
	Font, Size	28pt
	ForeColor	Blue
	ID	Div
Button	Height	55px
	Width	55px
	Font	30pt
	Font, Size	30pt
	ForeColor	Green
	ID	Pow
Button	Height	55px
	Width	55px
	Font	30pt
	Font, Size	30pt
	ForeColor	Purple
	ID	

☞ Event procedures:

```
Protected Sub Gat_Click(sender As Object, e As EventArgs) Handles Gat.Click
    A.ForeColor = Drawing.Color.Red
    B.ForeColor = Drawing.Color.Red
    R.ForeColor = Drawing.Color.Red
    R.Text = Val(A.Text) + Val(B.Text)
End Sub
Protected Sub Dow_Click(sender As Object, e As EventArgs) Handles Dow.Click
    A.ForeColor = Drawing.Color.Gray
    B.ForeColor = Drawing.Color.Gray
    R.ForeColor = Drawing.Color.Gray
    R.Text = Val(A.Text) - Val(B.Text)
End Sub
Protected Sub Mul_Click(sender As Object, e As EventArgs) Handles Mul.Click
    A.ForeColor = Drawing.Color.Blue
    B.ForeColor = Drawing.Color.Blue
    R.ForeColor = Drawing.Color.Blue
    R.Text = Val(A.Text) * Val(B.Text)
End Sub
Protected Sub Div_Click(sender As Object, e As EventArgs) Handles Div.Click
    A.ForeColor = Drawing.Color.Green
    B.ForeColor = Drawing.Color.Green
    R.ForeColor = Drawing.Color.Green
    R.Text = Val(A.Text) / Val(B.Text)
End Sub
Protected Sub Pow_Click(sender As Object, e As EventArgs) Handles Pow.Click
    A.ForeColor = Drawing.Color.Purple
    B.ForeColor = Drawing.Color.Purple
    R.ForeColor = Drawing.Color.Purple
    R.Text = Val(A.Text) ^ Val(B.Text)
End Sub
```

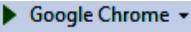
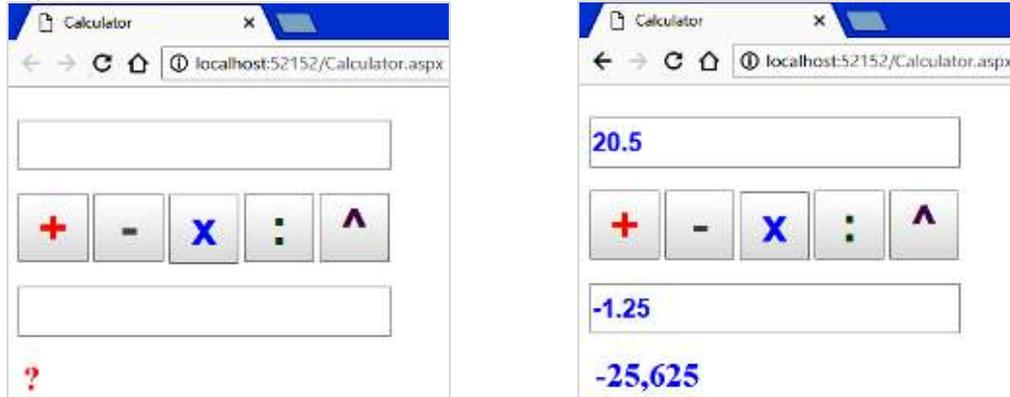
☞ Activate the  button, in order to launch the **Internet browser** (in this case, **Google Chrome**). The project will be compiled, and the generated code will be transferred to the browser (*Figure no. 3.*).

Figure no. 3. Document (Calculator) in Run Mode



☞ Stop the running application, using the  button (**Stop Debugging**).

4.2 Financial Calculations

☞ Create a new **Web** document (**FinancialCalculations.aspx**) - Figure no. 4, Table no. 2.

Figure no. 4. Document (Financial Calculations) in Design Mode

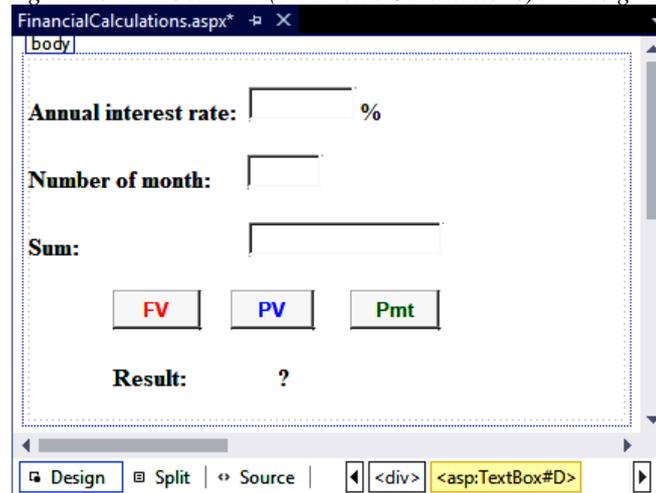


Table no. 2. Controls for Document (Financial Calculations)

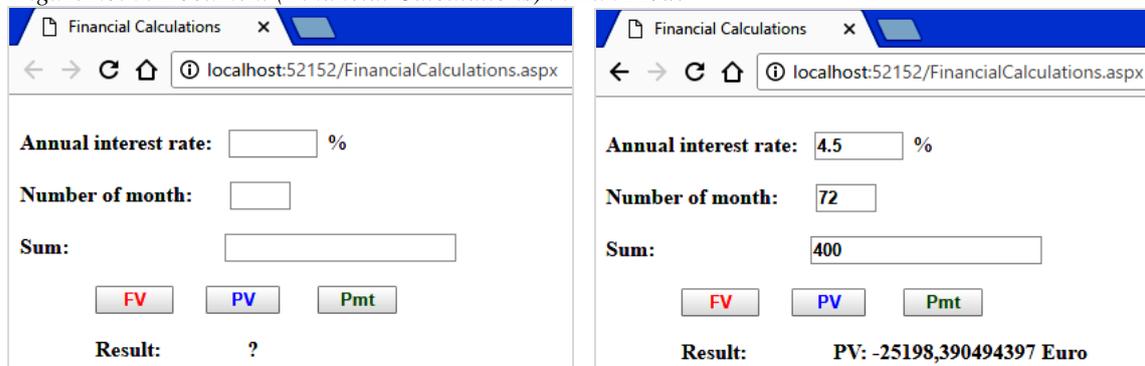
<i>Controls</i>	<i>Properties</i>
DOCUMENT	ID FinancialCalculations Title Financial Calculations
Label	ID R Text ?
TextBox	ID D
TextBox	ID P
TextBox	ID V
Button	ID bFV
Button	ID bPV
Button	ID bPmt

☞ Event procedures:

```
Protected Sub bFV_Click(sender As Object, e As EventArgs) Handles bFV.Click
    R.Text = "FV: " & FV(Val(D.Text) / 100 / 12, Val(P.Text), Val(V.Text)) & " Euro"
End Sub
Protected Sub bPV_Click(sender As Object, e As EventArgs) Handles bPV.Click
    R.Text = "PV: " & PV(Val(D.Text) / 100 / 12, Val(P.Text), Val(V.Text)) & " Euro"
End Sub
Protected Sub bPmt_Click(sender As Object, e As EventArgs) Handles bPmt.Click
    R.Text = "Pmt: " & Pmt(Val(D.Text) / 100 / 12, Val(P.Text), Val(V.Text)) & " Euro"
End Sub
```

☞ Activate the  button, in order to launch the **Internet browser** (in this case, **Google Chrome**). The project will be compiled, and the generated code will be transferred to the browser (*Figure no. 5.*).

Figure no. 5. Document (Financial Calculations) in Run Mode



5. References

- Welcome to Visual Studio 2015 - MSDN - Microsoft, [online] Available at: <<https://msdn.microsoft.com/en-us/library/dd831853.aspx>> [Accessed 22 may 2017]
- Cosma E., Visual Basic 2015, <http://stec.univ-ovidius.ro/>, [online] Available at: <<https://onedrive.live.com/?id=1A0E73A3A86C93F%21107&cid=01A0E73A3A86C93F>>