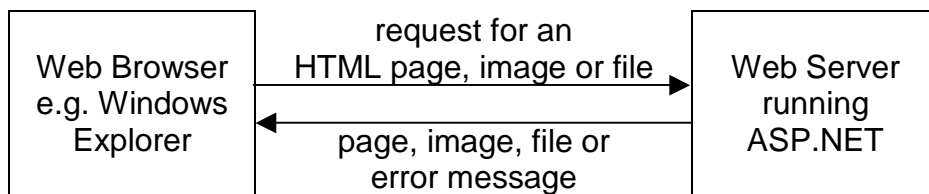


Visual Web Development

Terry Marris January 2009

12 State

Our ASP.NET pages are repeatedly passed to the browser (e.g. Windows Explorer) from the web server running our ASP.NET code. By default, the server does not preserve variable values from one request for a page to the next. We need a mechanism for preserving these values.



12.1 Counter

We create a Click counter. Every time a button is clicked, the count is increased by one.



Here, we have pressed the Click button five times.

12.2 State Variables

State variables are global variables whose values are maintained during the lifetime of program execution.

We declare `intCounter` as a global variable; it can be accessed directly by any procedure or function within the program.

```
Partial Class _Default
    Inherits System.Web.UI.Page
    ' GLOBAL VARIABLES
    Dim intCounter As Integer = 0
```

We use the Load and Unload events and a Session object to preserve the `intCounter` value between page requests. This is discussed in §12.3 and §12.4 below.

12.2 Click

The code behind the Click button is straightforward.

```
Protected Sub btnClick_Click(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles btnClick.Click
    intCounter = intCounter + 1
    txtCounter.Text = intCounter.ToString()
End Sub
```

12.3 Page Unload

Pages are loaded and unloaded at the server. When a page is unloaded we store the value of `intCounter` in a Session object.

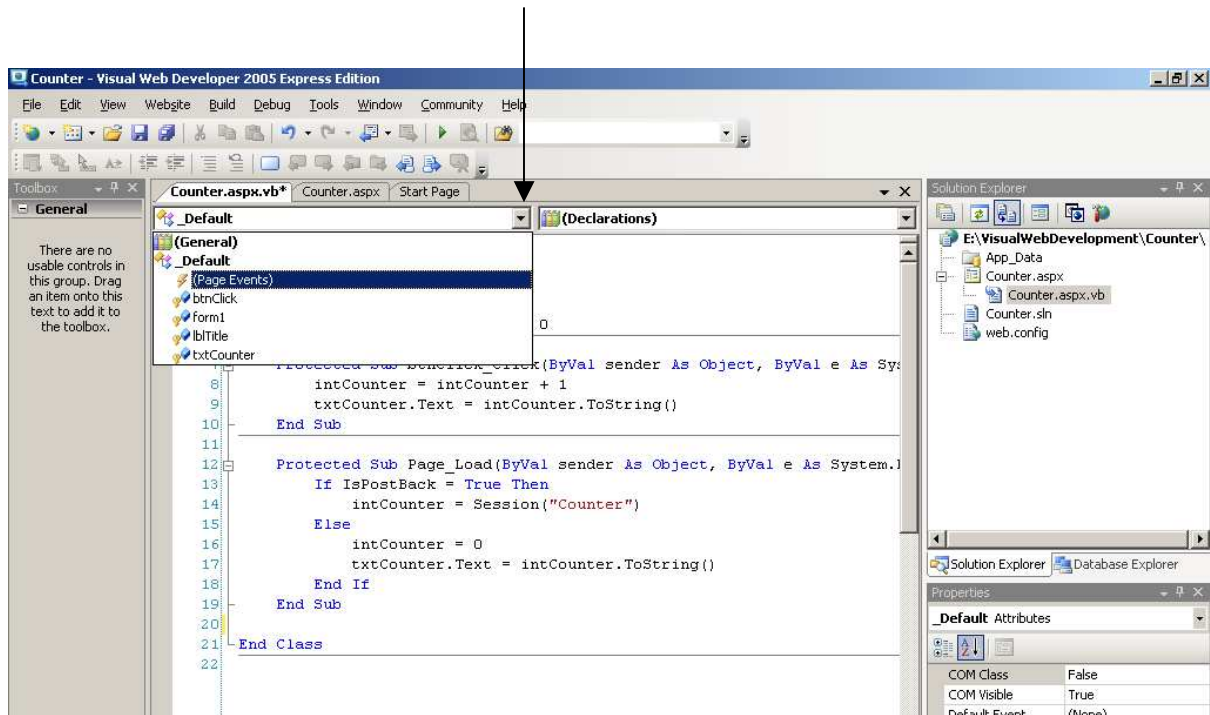
```
Session("Counter") = intCounter
```

We choose a descriptive name e.g. `Counter`, for the session.

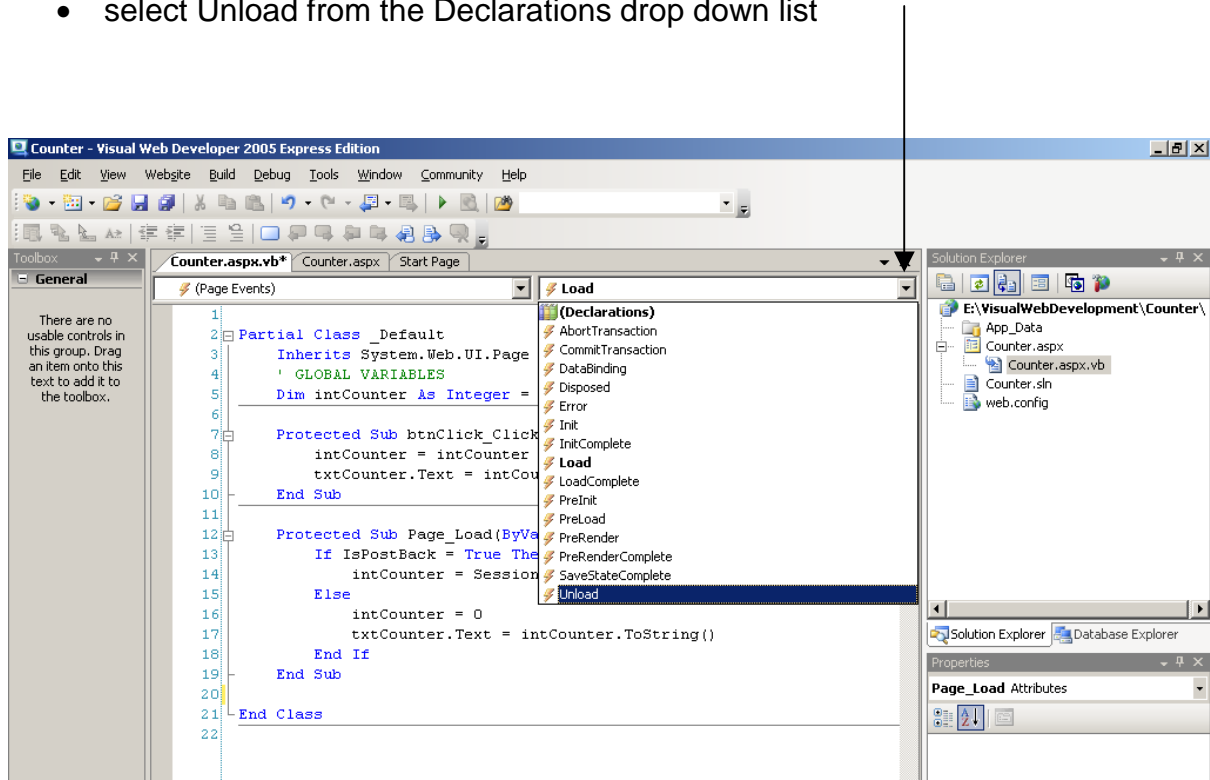
```
Protected Sub Page_Unload(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles Me.Unload
    Session("Counter") = intCounter
End Sub
```

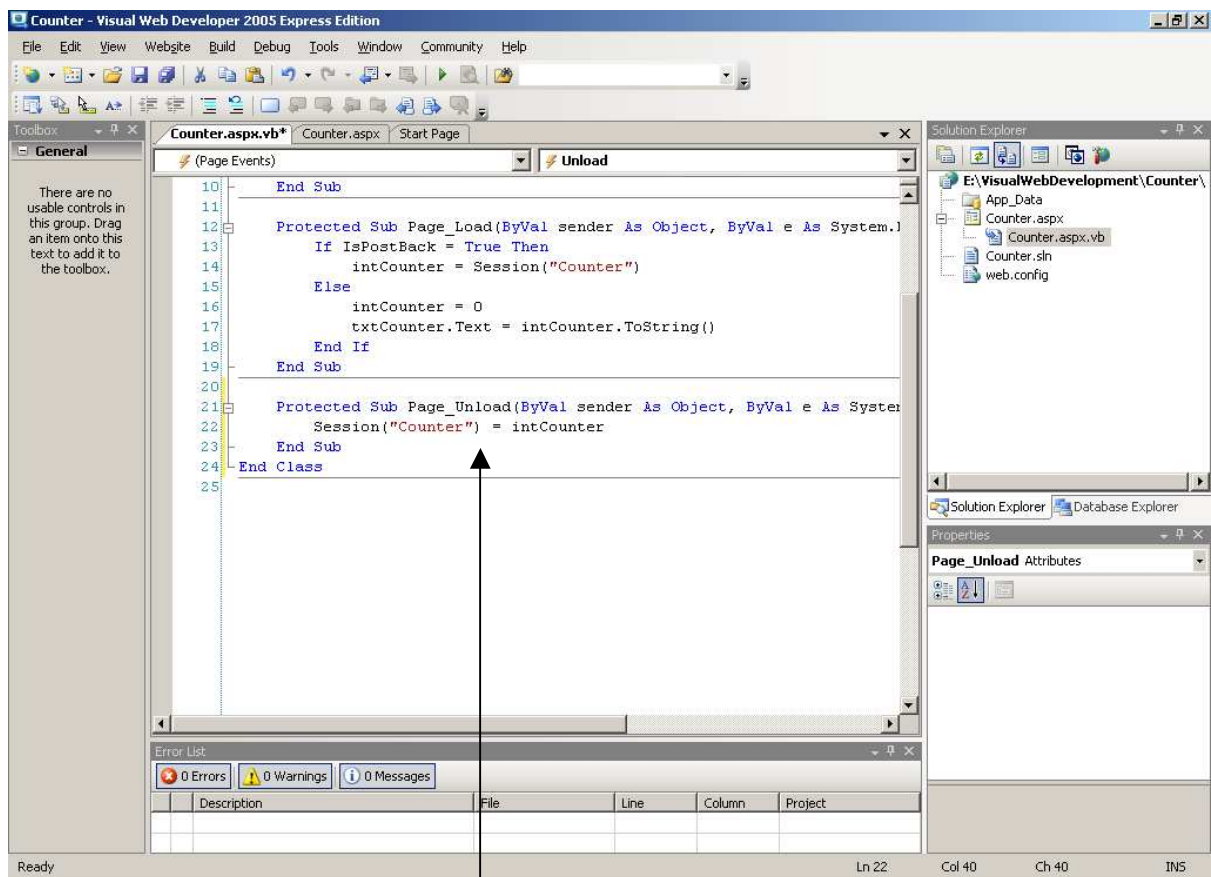
To obtain the skeleton for the Page_Unload event we:

- select PageEvents from the Default drop down list



- select Unload from the Declarations drop down list





- And enter `Session("Counter") = intCounter`

12.4 Page Load

The code behind the Page Load event is shown below.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles Me.Load
    If IsPostBack = True Then
        intCounter = Session("Counter")
    Else
        intCounter = 0
        txtCounter.Text = intCounter.ToString()
    End If
End Sub
```

The Page Load event runs whenever the server processes the page.

IsPostBack is a page property. It returns True if this is not the first time the page has been displayed.

If this is the first time the page has been displayed we initialise our state variable, intCounter, and copy it to our output text box, txtCounter.Text.

If this is not the first time the page has been displayed we retrieve the variable value from the Session object.

Of course, the Session name, "Counter" in our example, must be the same for both page load and unload procedures.

12.5 The Code

The entire code is shown below.

```
Partial Class _Default
    Inherits System.Web.UI.Page
    ' GLOBAL VARIABLES
    Dim intCounter As Integer = 0

    Protected Sub btnClick_Click(ByVal sender As Object, ByVal e As
        System.EventArgs) Handles btnClick.Click
        intCounter = intCounter + 1
        txtCounter.Text = intCounter.ToString()
    End Sub

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
        System.EventArgs) Handles Me.Load
        If IsPostBack = True Then
            intCounter = Session("Counter")
        Else
            intCounter = 0
            txtCounter.Text = intCounter.ToString()
        End If
    End Sub

    Protected Sub Page_Unload(ByVal sender As Object, ByVal e As
        System.EventArgs) Handles Me.Unload
        Session("Counter") = intCounter
    End Sub
End Class
```

12.6 Exercise

1. Try out the Click Counter program described above
2. Comment out entirely both the Page_Load and Page_Unload procedures and determine how the program then behaves.
3. Apply the preservation of variable values technique described above to any of your previously written programs that require it.