LATEST RESEARCH REPORT
Dissecting the .NET Framework

ASP 101
internet.com

Home
News
Samples
Forum
* **Articles**
Resources
Lessons
Links
Search

Please visit our Partners

activePDF

Advertising Info

Subscribe Now!
The journal for software developers using Microsoft technologies

internet.com

QUICK TIP: Using " in your Strings

Using ASP.NET Code-Behind Without Visual Studio.NET

by John Peterson

Introduction

I'm sure you've probably heard (since I've done nothing but talk about it) that we recently went to San Jose to put on our very own ASP.NET Developer Conference & Expo. While we were there, we talked to a number of developers and got a good feel for what the attendees were most looking forward to in ASP.NET. One of the things that kept coming up was the ability to use code-behind to separate the display and layout of an ASP.NET page from it's code and application logic.

Apparently some developers worry about what their sites look like and as a result they actually work with a graphic designer or layout artist (I'm not up on the politically correct title) to get it looking spiffy... who knew! ;) This is where code-behind really is a godsend. You just send the graphics person the Web Form portion of your page and they work with that. They never get to see (or mess with and break) the code that you've spent days working to get just right.

In Visual Studio.NET (VS.NET), this magic is all done for you. Now here's the rub... what if you don't have VS.NET? We've always maintained that while all the fancy tools are nice and might make you more productive, you don't need them to develop working solutions using ASP or ASP.NET. An email from one show attendee brought this whole topic to a head... here's an excerpt:

> hey, john - i was in the San Jose developer's conference, i believe i got a chance to talk to you there a bit...i was searching through asp101 and not quite able to come up with an answer to a question that's bothering me, maybe it's so obvious no one is answering it, but i can't quite work through it alone.
>
> i understand that there is a way to run a code-behind page inheriting from a class in a compiled DLL, yes? from what i've read visual studio .net does this 'automagically', but i can't find any explanation as to what the magic is, really. i've got VS on order, but anyway i'd like to understand how this stuff works with or without it.

Internet News
Internet Investing
Internet Technology
Windows Internet Tech.
Linux/Open Source
Web Developer
ECommerce/ Marketing
ISP Resources
ASP Resources
Wireless Internet
Downloads
Internet Resources
Internet Lists
International
EarthWeb
Career Resources

Search internet.com
Advertise
Corporate Info
Newsletters
E-mail Offers

internet.commerce
Be a Commerce Partner
Buy Books Here
Email Marketing
E-Commerce Solutions
Computer Training
Promote Your Website
Search the Web
Free Barter Account
Find a Consultant
For CGI Scripts
Managed DNS Service

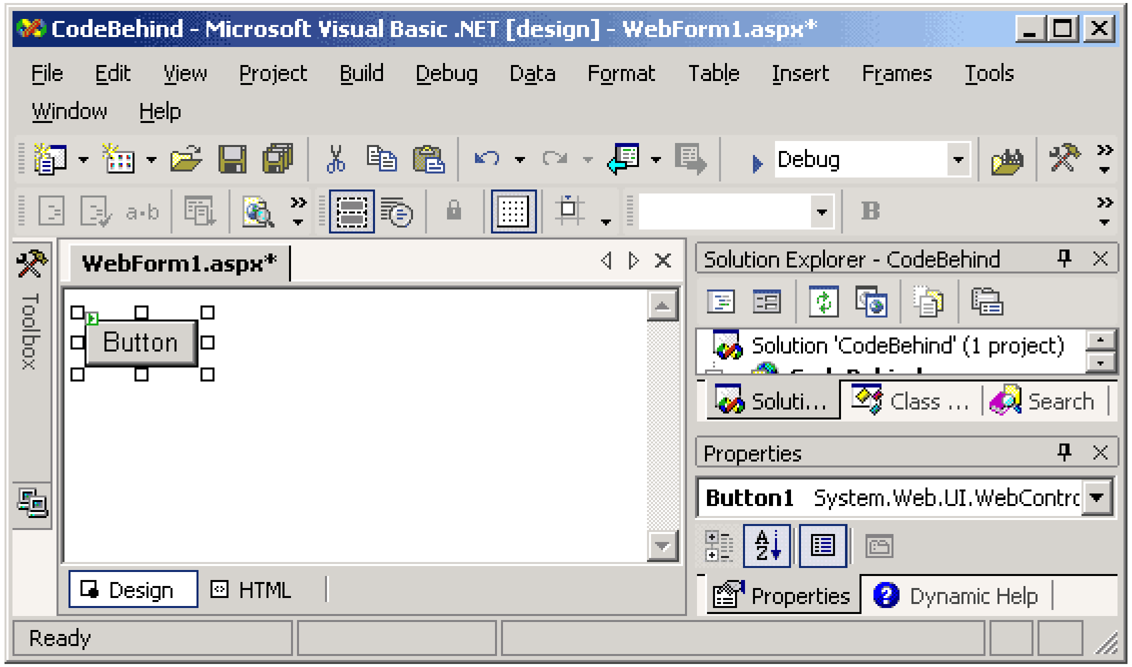
Windows Internet Tech.
15 Seconds
4GuysFromRolla.com
ASP 101
ASPWatch
ASP Wire
BHS.com
Swynk.com
VB Wire
WinDrivers.com
Win Planet

To address the email and since I don't like to be proven wrong... here's how you can implement code -behind using just a plain old text editor and the tools included in the .NET Framework... no VS.NET required!

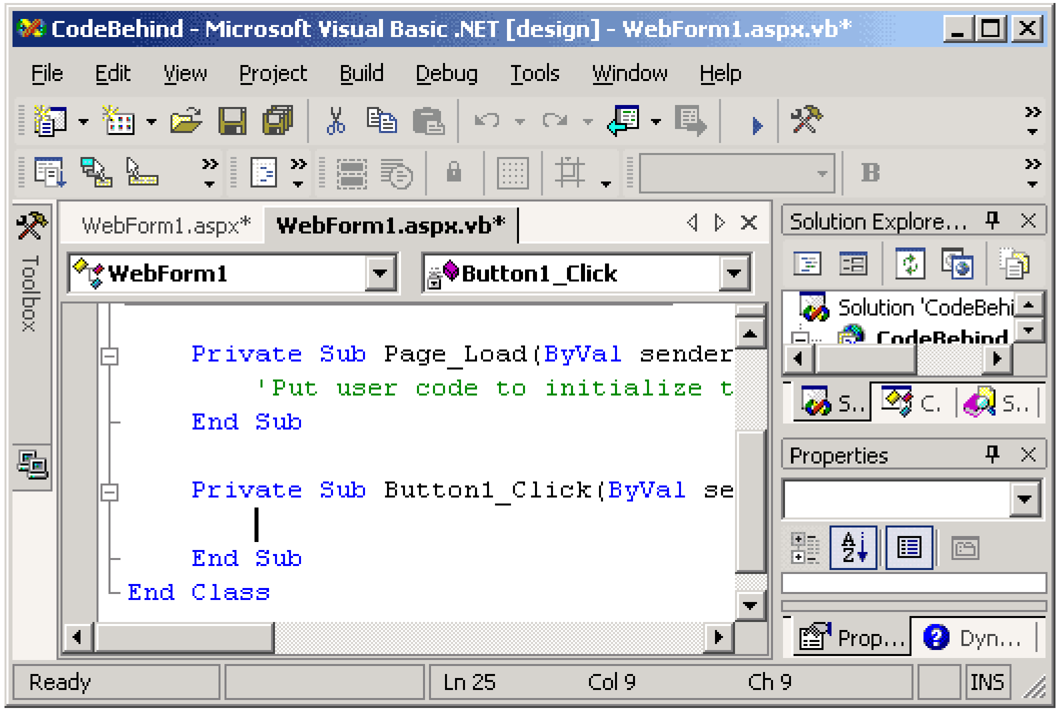
What Code-Behind Looks Like in Visual Studio.NET

Before we start, let me take this opportunity to illustrate what we're talking about. (I know I said no VS.NET, but it's just for illustration.)

I've fired off VS.NET and added a blank VB Web Form to my project (named boringly enough WebForm1.aspx). Next I add a button (named Button1) to the page. Here's what it looks like:


CodeBehind - Microsoft Visual Basic .NET [design] - WebForm1.aspx*
File Edit View Project Build Debug Data Format Table Insert Frames Tools Window Help
Debug
WebForm1.aspx*
Toolbox
Button
Design
HTML
Solution Explorer - CodeBehind
Solution 'CodeBehind' (1 project)
Soluti...
Class ...
Search
Properties
Button1 System.Web.UI.WebContrc
Properties
Dynamic Help
Ready

We're still basically in that same one file. When I click the button to add code to it, all of a sudden, a new file (named WebForm1.aspx.vb after our WebForm1.aspx) opens and VS.NET drops me into it in order to write the code for the button's click event.


CodeBehind - Microsoft Visual Basic .NET [design] - WebForm1.aspx.vb*
File Edit View Project Build Debug Tools Window Help
Toolbox
WebForm1.aspx* WebForm1.aspx.vb*
WebForm1
Button1_Click
Solution Explore...
Solution 'CodeBehi
CodeBehind
S.. C. S..
Properties
Prop... Dyn...
Ready
Ln 25 Col 9 Ch 9
INS

```
    Private Sub Page_Load(ByVal sender
        'Put user code to initialize t
    End Sub

    Private Sub Button1_Click(ByVal se

    End Sub
End Class
```

Here are the code listings for the two resulting files:

WebForm1.aspx

```
<%@ Page Language="vb" AutoEventWireup="false"
  Codebehind="WebForm1.aspx.vb" Inherits="WebForm1"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
  <head>
    <title>WebForm1 Code-Behind Test</title>
    <meta name="GENERATOR" content="Microsoft Visual Studio.NET 7.0">
    <meta name="CODE_LANGUAGE" content="Visual Basic 7.0">
    <meta name=vs_defaultClientScript content="JavaScript">
    <meta name=vs_targetSchema
      content="http://schemas.microsoft.com/intellisense/ie5">
  </head>
  <body MS_POSITIONING="GridLayout">
```

```
    <form id="Form1" method="post" runat="server">
      <asp:Button id="Button1" runat="server" Text="Button">
      </asp:Button>
    </form>

  </body>
</html>
```

WebForm1.aspx.vb

```
Public Class WebForm1
    Inherits System.Web.UI.Page
    Protected WithEvents Button1 As System.Web.UI.WebControls.Button

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> _
      Private Sub InitializeComponent()

    End Sub

    Private Sub Page_Init(ByVal sender As System.Object, _
      ByVal e As System.EventArgs) Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Private Sub Page_Load(ByVal sender As System.Object, _
      ByVal e As System.EventArgs) Handles MyBase.Load
        'Put user code to initialize the page here
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, _
      ByVal e As System.EventArgs) Handles Button1.Click
        ' Do Something When The Button Is Clicked!
    End Sub
End Class
```

Once I have VS.NET build my project and request it from a browser, whatever I type in the button click event handler (Button1_Click) in WebForm1.aspx.vb will execute when the button on WebForm1.aspx is clicked. Let's look at how this is accomplished.

Hot Wiring Our Own Code-Behind Page

Just looking at the listings above... most of this stuff has nothing to do with the actual code-behind process. The obvious command to investigate is `Codebehind="WebForm1.aspx.vb"`. Oddly enough however this doesn't do anything in ASP.NET... it's there only so VS.NET can find the source code! Let me illustrate.

I stripped down the above files and added a label control. I also added a command, which modifies the label, to the button click event handler (in the code-behind file) so we can tell when the two files are communicating and have some indication that the event handler is actually running. As a final step I renamed the files and classes to WebForm2 to prevent it from working because of anything VS.NET does behind the scenes... we're trying to accomplish this on our own! Here are the resulting 2 file listings:

WebForm2.aspx

```
<%@ Page Language="vb" AutoEventWireup="false"
  Codebehind="WebForm2.aspx.vb" Inherits="WebForm2"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
  <head>
    <title>WebForm2 Code-Behind Test</title>
  </head>
  <body>
    <form id="Form1" method="post" runat="server">
           <asp:Button id="Button1" runat="server" Text="Button">
           </asp:Button>
           <br />
           <br />
           <asp:Label id="Label1" runat="server">Label</asp:Label>
    </form>
  </body>
</html>
```

WebForm2.aspx.vb

```
Public Class WebForm2
    Inherits System.Web.UI.Page
    Protected WithEvents Button1 As System.Web.UI.WebControls.Button
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label

    Private Sub Button1_Click(ByVal sender As System.Object, _
      ByVal e As System.EventArgs) Handles Button1.Click

        Label1.Text = "I was clicked at: " & System.DateTime.Now
    End Sub
End Class
```

Naturally that would be too easy... when you try and run WebForm2.aspx, you'll get an error something like this:

Parser Error Message: Could not load type 'WebForm2'.

Don't worry... all is not lost. The problem is simply that it can't find the class we define in our code-behind file. Normally VS.NET will automatically compile the .vb file into a .dll and place it in your application's /bin directory. Since we're not using VS.NET, it wasn't compiled and the application can't find the appropriate class. There are two solutions - compile it manually or tell the .aspx file where to find the .vb source file.

Compiling it manually is really pretty easy. The command will look something like this:

```
vbc /t:library /out:bin\WebForm2.dll /r:System.dll /r:System.Web.dll WebForm2.aspx.vb
```

I'm not going to go into all the compiler options, but basically we're taking WebForm2.aspx.vb and compiling it into a dll named WebForm2.dll and placing it in the application's /bin directory.

This option is the better approach if you need complete control over your compiler options or if you will be distributing the application without the .vb source files. Being the lazy type, I tend to go for option 2... check out this code listing:

WebForm3.aspx

```
<%@ Page Language="vb" AutoEventWireup="false"
  Src="WebForm2.aspx.vb" Inherits="WebForm2"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
  <head>
    <title>WebForm3 Code-Behind Test</title>
  </head>
  <body>
    <form id="Form1" method="post" runat="server">
           <asp:Button id="Button1" runat="server" Text="Button">
           </asp:Button>
           <br />
           <br />
           <asp:Label id="Label1" runat="server">Label</asp:Label>
    </form>
  </body>
</html>
```

Looks just like the last one doesn't it... well not quite... notice that instead of the `Codebehind="WebForm2.aspx.vb"` we now have `Src="WebForm2.aspx.vb"`. While `Codebehind` doesn't mean anything to ASP.NET, `Src` (short for source) does and the code-behind file will compile on the fly just like the .aspx file.

> Note: Don't worry... you're not missing a file... I've got WebForm3.aspx running off the same code-behind file as WebForm2.aspx since the two .vb files would be identical anyway.

Some Final Notes

In order for .NET to find your classes, make sure your compiled files are stored in the /bin directory off the root of your application. You need to make sure you've set your directory up as an IIS application or else ASP.NET will go up the tree until it finds one and end up at the /bin directory of the root application if it doesn't find one sooner.

Those of you using VS.NET might have noticed my `Inherits` statements are short a project name. That's because VS.NET creates a separate namespace for each project it creates. It's easy enough to do just use the Namespace command, but that's beyond the scope of this article. I'm only mentioning it so you don't panic when you see an inherits line that looks like this:

```
<%@ Page Language="vb" Inherits="ProjectName.WebForm2"%>
```

Wrap Up

So to sum everything up... all it really takes to do code-behind is one little inherits attribute in your page declaration line specifying the name of the class you want to inherit. If you're willing to precompile your classes into .dll files, that's where it stops. If you're lazy like me or like the "edit and run" simplicity that classic ASP gave you, add a src attribute pointed at your code-behind file and ASP.NET will compile it for you. That's really all there is to it.

If you don't want to copy and paste the code listings, you can get all 5 files in zip file format from here.

Other Code-Behind Resources On The Web

- ASP.NET Code Behind Pages from 4GuysFromRolla
- INFO: ASP.NET Code-Behind Model Overview`
- ASP.NET Unleashed Sample Chapter 6 - the section on code-behind is Separating Code from Presentation
- Reusability in ASP.NET: Code-behind Classes and Pagelets
- Developing User Controls in a Code-Behind File
- Working with Single-File Web Forms Pages in Visual Studio .NET - a little on not using code-behind in VS.NET

Home | **News** | **Samples** | **Forum** | **Articles** | **Resources** | **Lessons** | **Links** | **Search**

Copyright 1999-2002 INT Media Group, Incorporated All Rights Reserved.
Legal Notices, Licensing, Reprints, & Permissions, Privacy Policy.
http://www.internet.com