

# Re: an old geezer running visual studio for the first time

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2005-06/msg02820.html>

---

- *From:* "TB" <[tbpostbox-googlegroups@xxxxxxxxx](mailto:tbpostbox-googlegroups@xxxxxxxxx)>
  - *Date:* Wed, 15 Jun 2005 17:33:39 +0200
- 

Please refer to my reply to Jeremy.

Anyway, I am beginning to realize that a decision to adopt VS.NET is not caused by a love-at-first-sight / wow experience, because it is about the most unsexy, userunfriendly program I have seen in a long time. But my experience in programming has taught me to keep my eye on the ball, which in this case is called long-term productivity gains, a area in which VS.NET will hopefully prove its worth.

TB

"Kevin Spencer" <[kevin@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:kevin@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)> wrote in message [news:e%23Tvn\\$DcFHA.456@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:e%23Tvn$DcFHA.456@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)

>I agree completely with Jeremy. A good programmer is definitely a control  
>freak, and wants to be in control of his/her app as much as possible. The  
>"training wheels" analogy is perfect. For beginners, these types of tools  
>can be the difference between getting started in ASP.Net and not getting  
>started at all. But they are hardly intended to be a substitute for real  
>hard-core programming, which is why the SDK has articles and tutorials on  
>such subjects as designing and building custom Server Controls, etc. Some  
>of the greatest aspects of Visual Studio.Net is that it can allow one to  
>build something quick and dirty in a short period of time, or it will allow  
>the developer to delve as deeply into the technology as one can go, all the  
>way down to assembler and MSIL if necessary or desired.

>

> --

> HTH,

>

> Kevin Spencer

> Microsoft MVP

> .Net Developer

> Ambiguity has a certain quality to it.

>

> "Jeremy S" <[A@xxxxx](mailto:A@xxxxx)> wrote in message

> [news:ejj2gqDcFHA.3040@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:ejj2gqDcFHA.3040@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)

>> Hi TB

Re: an old geezer running visual studio for the first time

>>  
>> From cruising through your various comments in this thread I gather that  
>> you are very comfortable with the old school way to do things. One  
>> thought is to just stay there. Seriously. If you need to get stuff done,  
>> then might as well just stay where you are most productive (at least  
>> until you are comfortable with the new stuff).  
>>  
>> Now, the new way to do things will involve somewhat of a learning curve –  
>> but once you've traversed it, you'll be \*way more\* productive than you  
>> ever could have been with the old stuff.  
>>  
>> From your other comments it appears that you view ASP.NET and ADO.NET as  
>> somehow limiting what a developer can do – especially if the developer  
>> wants to "get down and dirty" and write some code. Not true at all. Much  
>> of what you have likely read is touting the "training wheels" that have  
>> been included in Visual Studio – like drag-and-dropping DataAdapters into  
>> a visual surface. You should understand that most serious developers (at  
>> least the ones I've met) simply do not use such training wheels. You can  
>> ignore those and dive right into the code. When it comes to ADO.NET – you  
>> can actually write more code than you could have in classic ADO. Remember  
>> .UpdateBatch() in classic ADO? That actually writes a bunch of dynamic  
>> SQL under the covers for you that you never get to manipulate. ADO.NET  
>> actually gives you much more control and more granular control of  
>> inserts, updates, and deletes. I highly recommend to you another book  
>> "ADO.NET" by David Sceppa – Microsoft Press. A "must read" IMHO for  
>> anyone familiar with classic ADO moving to ADO.NET. Sceppa presents the  
>> new stuff and compares/contrasts it with classic ADO. Remember, ADO.NET  
>> is not simply the next version of classic ADO... it's a whole new way to  
>> interface with your data. And, to answer your particular question, yes –  
>> ADO.NET will let you work as easily with PHP as you can with SQL Server,  
>> Oracle, Access, or any other database or data store (even XML files).  
>>  
>> About the starter kits – I recommended the Commerce one because it is,  
>> IMHO, the most straight-forward; it shows you a lot of the nuts-and-bolts.  
>> This is on contrast to the Portal kit which shows some very cool stuff  
>> (like serving up an entire non trivial site through only one aspx page) –  
>> but it's stuff most projects likely won't need.  
>>  
>> -HTH  
>>  
>>  
>> "TB" <tbpostbox-googlegroups@xxxxxxxx> wrote in message  
>> [news:eaWTn63bFHA.3492@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:eaWTn63bFHA.3492@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)  
>>> Thank your very much for bothering to answer this "old newbie".  
>>>  
>>> I will follow your advice. Particularly I will desect some of the sample  
>>> projects on [www.asp.net](http://www.asp.net) to learn how code-behind truly works. Actually  
>>> it sounds like an interesting concept, but I suppose (without knowing  
>>> anything yet) that there are several degrees of code-behind depending on  
>>> how puritanic you are or how much time you have left to finish a job. I  
>>> guess that it also difficult for .NET developers to resist the

Re: an old geezer running visual studio for the first time

>>> temptation to get dirty and throw in a bit of inline code here and there  
>>> just to get a project out of the door.  
>>>  
>>> I will also take a serious look at the book you suggest.  
>>>  
>>> Another question, now that I got you "on the line": Is it possible to  
>>> make VS.NET work perfectly / natively / transparently with MySQL?  
>>>  
>>> TB  
>>>  
>>>  
>>> "Jeffrey Todd" <Me@xxxxxxxxxxxxxx> wrote in message  
>>> [news:eM04AAsbFHA.1040@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:eM04AAsbFHA.1040@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)  
>>> These 3 suggestions will hopefully help a lot:  
>>>>  
>>>> 1. Consider developing your aspx and ascx files in FrontPage,  
>>>> Dreamweaver, Notepad, or whatever is your favorite HTML and/or CSS  
>>>> editor – and limit your use of VS.NET to (1) all programming work  
>>>> (code-behind logic) and (2) launching you application for testing  
>>>> purposes. VS.NET 2003 most definitely NOT the place to write HTML – so  
>>>> no need for you to think about grid vs flow layouts. Reason being that  
>>>> VS.NET 2003 will rewrite your HTML for you – often with disastrous  
>>>> results. This link from a Microsoft insider explains:  
>>>> <http://blogs.msdn.com/MikhailArkhipov/archive/2004/05/16/132886.aspx>  
>>>> The upshot is that many of us use FrontPage or Dreamweaver for all  
>>>> HTML-specific work (i.e., ASPX and ASCX file development) and use  
>>>> VS.NET only for programming/code-behind work. The way I develop is that  
>>>> I keep a copy of Dreamweaver running at all times along side VS.NET  
>>>> 2003 – and switch between the two. One last point here is that VS.NET  
>>>> 2005 – due out in November or so – is expected to no longer have the  
>>>> HTML rendering and management issues that VS.NET 2003 has.  
>>>>  
>>>> 2. Get this book: Programming ASP.NET – by Dino Esposito – Microsoft  
>>>> Press. It puts everything in context – IMHO ideal for your situation.  
>>>> It's not a step-by-step tutorial, but it is more valuable because tells  
>>>> you how to understand ASP.NET.  
>>>>  
>>>> 3. Familiarize yourself with the starter kits at the following link:  
>>>> <http://www.asp.net/Default.aspx?tabindex=8&tabid=47>. I'd especially  
>>>> recommend that you download and dissect the Commerce starter kit. The  
>>>> starter kits are complete ASP.NET Web applications and include all  
>>>> source code – which is very readable and well-documented. The Commerce  
>>>> starter kit is one of the simpler – so it would be a good starting  
>>>> point as it incorporates many fundamentals.  
>>>>  
>>>> –HTH  
>>>>  
>>>> –JT  
>>>>  
>>>>  
>>>>

Re: an old geezer running visual studio for the first time

Re: an old geezer running visual studio for the first time

>>>> "TB" <tbpostbox-googlegroups@xxxxxxxx> wrote in message  
>>>> [news:eTCWZ3qbFHA.2212@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:eTCWZ3qbFHA.2212@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)  
>>>> Hi all:  
>>>>  
>>>> If you think that the following comments are absolute amateurish, then  
>>>> please bear with me, or simply skip this thread.  
>>>>  
>>>> A couple of months back I made the decision to initiate a gradual  
>>>> upgrade of my web programming skills from Classic ASP / VBS to ASP.NET  
>>>> / VB.NET.  
>>>>  
>>>> While the study of the language differences and all the new features  
>>>> in .NET has so far not been a traumatic experience, I am a bit  
>>>> shell-schocked after installing and running for the first time Visual  
>>>> Studio .Net (2003 edition).  
>>>>  
>>>> Until now I have programmed everything in the old-fashioned way, i.e.  
>>>> with a bare minimum of tools: a simple editor for coding, a bit of  
>>>> Frontpage for page layout, and my browser to see the results when  
>>>> running the page against a local instance of IIS.  
>>>>  
>>>> I have done quite a bit web programming, developing anything from  
>>>> yellow pages to accounting programs.  
>>>>  
>>>> But when I ran VS.NET for the first time, I simply didn't know where  
>>>> to start.  
>>>>  
>>>> The program appears to be a sort of Swiss-knife approach to  
>>>> programming, eliminating the need for external apps. During the past  
>>>> couple of years I have developed my own little collection of helpers  
>>>> which I use during development.  
>>>>  
>>>> VS.NET creates files for "project mangement". My project management  
>>>> normally consists notes and diagrams. My reusable code (subs and  
>>>> functions) is kept in separate files, which I either reference to or  
>>>> cut/paste – depending on the situation.  
>>>>  
>>>> I normally use a GUI tool (Frontpage for example) to generate a page  
>>>> layout. In VS.NET I am met with the question whether I want the  
>>>> designed to be grid-based or flow-based.  
>>>>  
>>>> Finally, I discovered that VS.NET saves code and html in different  
>>>> files. I am used to keeping everything together, grouping functions,  
>>>> subs etc above the <head> tag, and then insert code snippets between  
>>>> <% %> tags. This approach seems to be frowned upon by VS.NET.  
>>>>  
>>>> Am I an old-fashioned guy? No, because I have made the desicion to  
>>>> move to .NET, although a bit late perhaps. But I would like to find a  
>>>> book, a tutorial, anything that can teach this old geezer to use a new  
>>>> toy which apparently could improve my productivity.  
>>>>

Re: an old geezer running visual studio for the first time

>>>>> Any advice / comments from people who may or may not have been through  
>>>>> the same process would be most welcome.

>>>>>

>>>>> Thanks

>>>>>

>>>>> TB

>>>>>

>>>>>

>>>>>

>>>>

>>>>

>>>

>>>

>>

>>

>

>

.

---

• **Follow-Ups:**

- ◆ **[Re: an old geezer running visual studio for the first time](#)**  
    ◇ From: Kevin Spencer

• **References:**

- ◆ **[an old geezer running visual studio for the first time](#)**  
    ◇ From: TB
- ◆ **[Re: an old geezer running visual studio for the first time](#)**  
    ◇ From: Jeffrey Todd
- ◆ **[Re: an old geezer running visual studio for the first time](#)**  
    ◇ From: TB
- ◆ **[Re: an old geezer running visual studio for the first time](#)**  
    ◇ From: Kevin Spencer

- Prev by Date: **[Re: Server.Transfer vs. Response.Redirect](#)**
- Next by Date: **[Re:PostBack "Please Wait..." Message.](#)**
- Previous by thread: **[Re: an old geezer running visual studio for the first time](#)**
- Next by thread: **[Re: an old geezer running visual studio for the first time](#)**
- Index(es):
  - ◆ **[Date](#)**
  - ◆ **[Thread](#)**