

Re: VS 6 to VS.Net justification

Source:

<http://www.tech-archive.net/Archive/VisualStudio/microsoft.public.vsnet.general/2007-05/msg00042.html>

- *From:* Sandy <Sandy@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 4 May 2007 11:49:03 -0700
-

Thank you very much for your detailed response, Andrew. It's exactly what I was looking for.

My only serious complaint about VS2005 and .NET is that some of the features (or even removal of from VS6) kind of give the impression that MS wanted to encourage NON-DEVELOPERS to start developing.

I very definitely got that impression too!

—
Sandy

"Andrew D. Newbould" wrote:

In message <7A2AE7E3-7890-4978-A31A-3653FC2ABD7A@xxxxxxxxxxxxxx>, Sandy <Sandy@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx> writes

So far, VB6 has been pretty good as far as the needs of the organization.

I am concerned, however, that we are using Sql Server 2005. Seems to me there is a whole lot of functionality that is more compatible with VS.Net, although I don't know how to present that argument.

More "Integrated" would be a better term to use. Personally, I like SQL 2005; it has some really good new features. There is a comparison sheet on MSDN somewhere. In saying that, we still use SQL 2000 for all our production databases (over 50 at the last count) and probably won't start migrating till towards the end of the year. IMO past experience has proven its better to wait for the first couple of service packs to be released which has now happened.

Just about all of our apps involve saving in a database and retrieving

Re: VS 6 to VS.Net justification

information for reports, etc.

This is supposed to be easier in .NET however I have to admit that IMO there is no real difference over classic ADO. Subsequently, we use both depending on what we are trying to achieve. The biggest difference is that ADO.NET ONLY ever uses DISCONNECTED recordsets (ie: no server side cursors at all). There are times when a server side cursor does have an advantage. There are still some things that ADO.NET can't do so you will be forced back to Classic ADO.

Also, by way of example, one solution I am in the middle of right now is writing an application that incorporates MS Word. It basically grabs records from an outside Oracle database, pulls it into Sql Server, and it is used to provide information to form letters in MS Word (amongst other things it does).

We are in the middle of migrating to Windows Server 2003 and Windows XP for client machines. We will probably be upgrading Office applications in the future as well. I wonder if my application incorporating MS Word may "break."

With MS its always likely that it will break. All you can do is test the new environments and patch your software if required. Using Virtual Server to create these environments is far better than the old Ghosting method. Last time I looked 70% of Office is still traditional COM based, so MS certainly are not practising what they preach.

Future applications will need RAD, but I have to admit, I am not convinced on that issue from what I've seen yet; let alone convince someone else.

For Windows Forms applications this is true. Apart from some subtle but useful differences, you will not see much improvement over VS6. However, when it comes to Web Services (ie: ASP.NET) or even BI (using the new Reporting Services in SQL 2005) things get interesting and have improved ALOT.

Because the new technology relies on real O-O technology, what is going to happen to all of the old dlls? Will VB6 apps "break"?

The MS line is that current COM dlls etc will still work in Vista's life cycle but they won't guarantee that for future versions on Windows. So your safe for the time being. That being said, Vista has alot more security features which can quickly break existing traditional

Re: VS 6 to VS.Net justification

applications and therefore you will almost certainly need to look at the Best Practices for developing on Vista. Specially, when it comes to what areas of the "Program Files" and the "Registry" you can / can not access.

Moving to OO; this is where .NET has come full circle and offers many benefits for developers and the application life cycle in general. For example, we write plug-ins for our core components to satisfy individual clients needs. In VS6 and the like this leads to much replicated / similar code for each plug-in etc. In .NET we have defined a few Base Classes which our new / converted plug-ins Inherit from. This has removed an awful lot of repetative / similar code. We only need to Override the elements that we really need to tweak etc. Error handling is another major improvement. We have probably reduced our coding to about 1/3 of the normal requirement under VS6. This allows us to concentrate of other areas like improving usability, performance, new projects etc.

As far as my professional career is concerned, it is certainly in my best interests to receive .Net training and advance to the next level; I certainly don't want to become obsolete . . . yet another one of my concerns, but definitely something I cannot use as an argument.

Why not!!!

Would they prefer to lose valuable team members with important knowledge of their systems etc. Secondly, as the company grows you are going to find it harder to recruit new staff with the right skill set unless the company moves forward.

We still have millions of lines of VB6 code to maintain. If it ain't broke then don't fix it. Nobody likes being forced to change (specially by a behemoth like MS) however, there comes a point when you need to make a decision as to what's best for the future of the company. We have built our systems in such a way that we can continue to use our old code and migrate it when a changes is required or we have the time (its fairly well planned).

After all there are other choices. If you are going to have to rewrite you code and systems then there are plenty of alternatives to .NET. What about Java or Delphi !!!

Another question I have to ask is, due to Microsoft's inclination to release tools before they are completed is (i.e. service packs and fixes), what recommendation would you give me for the Visual Studio version to go to?

IMO your nearly always better applying the Service Packs etc in a test

Re: VS 6 to VS.Net justification

environment first, before deploying. We still have not deployed SP1 for Visual Studio 2005 as there seems to be concerns over its stability and changes to functionality relied upon. Personally, I prefer to wait a month or two and see what problems everyone else has first.

I am sure you can see from my response how difficult it is for me to implement a sales pitch . . . I am not even convinced myself. Perhaps the best way to approach this is for someone to try to convince me to upgrade and then use their arguments!

(I do have to admit, though, that I have worked with .Net on Web development for about a year outside of work and I think it's fun. That certainly will not work as an argument!!)

Any help in this regard will be appreciated -- and please forgive my rambling!!

No problem. Sorry my first response was a little sarcastic but these questions get asked alot in newsgroups. I am not 100% convinced myself, however there are some nice features and functionality that make you want to use it all.

My only serious complaint about VS2005 and .NET is that some of the features (or even removal of from VS6) kind of give the impression that MS wanted to encourage NON-DEVELOPERS to start developing. That's never a good idea unless someone has some formal training first. Its only good for the people making money from the sale. After all, would you let someone fly a Jumbo Jet without a pilots license? In saying that, REAL DEVELOPERS now have some pretty good tools to do their job with.

Kind Regards,

—

Andrew D. Newbould E-Mail: NO-usenet-SPAM@xxxxxxxxxxxx

ZAD Software Systems Web : www.zadsoft.com