

Re: Who needs Tony Snow...

Source:

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2006-05/msg00018.html>

- *From:* "Randy Birch" <rgb_removethis@xxxxxxxx>
 - *Date:* Sun, 30 Apr 2006 22:13:17 -0400
-

April 21, 2006

Visual Studio 2005: Unstable and Highly Recommended
Expert opinions based on real-world experiences
(Page 1 of 8)

Scott Swigart

Dr. Dobb's recently sat down with some industry experts who use Visual Studio 2005 on a daily basis to get their impressions about its stability and usefulness.

Visual Studio 2005 started shipping in November of last year. On behalf of Dr. Dobb's, Scott Swigart recently sat down with some industry experts who use the product on a daily basis to get their impressions about product stability and usefulness. Rockford Lhotka, Billy Hollis, Bill Vaughn, and Kathleen Dollard took the time to chat with Scott on this topic. During the conversation, they describe in detail stability problems with Visual Studio 2005, and point out resolutions to many of the specific issues. It's also worth noting, up front, that despite the notable stability issues with Visual Studio 2005, the panel strongly and unanimously recommended Visual Studio 2005 over earlier versions of Visual Studio. Microsoft has also provided additional information.

DDJ: Thanks everyone for making time to talk. I wanted to get some people together who work on a wide spectrum of things. I know with this group, I've got representation for everything from stand-alone applications to distributed systems. Across this group, there's Windows Forms, Web, VB.NET, and C#. As people evaluate Visual Studio 2005, as people start using it, I figured that you would have some great advice and insight for them that might save them a lot of time and trouble. Specifically, I want to focus on the stability of Visual Studio 2005. Right after it released, there were some prominent postings indicating that the product was buggy. It's been out for a few months, you all are using it daily, what's your impression? Is there an issue around stability?

Billy H.:I'll go first. There are three significant issues that I've run into with Visual Studio 2005. One is the 99 percent CPU problem. (Editors note: While working in Visual Studio, the IDE becomes unresponsive for tens of seconds, and the CPU shows 100 percent utilization.) That seems to be

Re: Who needs Tony Snow...

fixed now. The possible things that could have been the fix for that are the hotfix, uninstalling Refactor!, and turning off edit-and-continue. (Editors note: Any issues where Refactor! was causing 100 percent CPU have since been fixed by DeveloperExpress. If you are using the latest version of Refactor! you will not experience this issue as a result of the Refactor! add-in. See the response from Microsoft for more details.) I heard from various sources that one of those things might make a difference, and in fact, at least one of them did, because I no longer see that problem.

The second problem is the Windows Forms designer loses its sanity and tells me that there's something wrong with a form. That problem has been reduced, but it's not entirely gone. It doesn't seem to be predictable, it just happens some times for no apparent reason. If I shut the development environment down, and bring it back up, it will be okay again for a while.

The third problem is when it loses its internal references, and it claims to no longer know about an assembly that is referenced. This happens regularly, and predictably, and I can pretty much play around and make it happen. It happens when I'm working with controls and extender providers. The controls appear in the toolbox, and when I drag one over to the form and drop it, sometimes this problem appears. That is frequent enough that I have to tell my audience that this is likely to happen during the course of a demonstration. I do demonstrations a lot, and it happens in front of audiences, and I shut the development environment down, and bring it back up, and the problem is gone.

Beta Debris?

DDJ: Okay, you mentioned three there. The first is that you're working along, and the CPU pegs at 99 percent, and that was fixed by the hotfix, or turning off edit-and-continue, or uninstalling Refactor!, correct?

Rocky: I can shed a little light on that. I have not yet applied the hotfix, and I have not turned off edit-and-continue, but I did uninstall and reinstall Refactor!, and that did solve the problem for me. It appears that one of the beta versions of Refactor! didn't remove everything from the GAC when you do an upgrade. A lot of us installed beta versions of Refactor! When the RTM version came out, I installed the RTM version of Refactor!, but I had a DLL left over in the GAC. (Editors note: Again, this problem has since been solved. If you are using the latest version of Refactor! you will not experience this.)

DDJ: Ah, so some beta debris. Is anyone else running into common problems? Is anyone seeing lots of: "It did this weird thing once, I can't get it to repeat, but I just keep running into strange instabilities. I shutdown, and restart, and it's okay again for a while."

Kathleen: It's that, but it's more than that. I do a lot in C# and Visual

Re: Who needs Tony Snow...

Re: Who needs Tony Snow...

Basic, and I see problems in both. I was a little surprised to hear that the 99-percent CPU problem has gone away. There's a similar problem, though not quite as severe, in C#. For me, the IDE comes up with a little gear icon, and becomes non-functional for 5-20 seconds. It does tend to come back in C#, which is different than the way the problem manifests itself in VB. There are also problems with references that are hard to track.

I have one problem in a form, which I've been unable to solve. I have a form that contains a treeview control. When I open that form in the designer, Visual Studio removes the designer generated line of code that instantiates the treeview. I have to manually put that line back in the designer code section every time I work on that form. There's definitely weird stuff that goes on in other corners of the product.

DDJ: Is it possible that a lot of the problems that people are experiencing are just a result of beta debris? I know that Microsoft provided an uninstall tool to try to help keep previous problems. Is it possible that if people do a clean install of Visual Studio 2005, that they won't see any of these problems?

Kathleen: No, this machine was built clean with Visual Studio 2005.

Billy H.: With my two laptops, one was a clean build, and one wasn't, and they both experience this drag and drop problem.

Bill V.: I'm with Kathleen. Mine are all clean builds to start with. My recent area of focus is a lot different than what I think a lot of other people are working with. I've been working with the CLR integration in SQL Server, using .NET languages to write stored procedures. I've crashed Visual Studio countless times. Most of the time, it's when I'm trying to do remote debugging, when I'm attached to the SQL Server process behind the scenes. (Editors note: Bill Vaughn has great information on troubleshooting connection problems here.) This works reliably about 90 percent of the time. But every so often, the system gets in a mode where it looks like it's starting remote debugging, but it gets lost and doesn't seem to know what to do. I thought I had traced it down to not assigning a default test script, but that doesn't always help. The only way out of this is to shut down Visual Studio. If you just try to stop debugging, that invariably crashes Visual Studio. I've also hit some new nuances that I can't reproduce after rebooting, where the debug manager crashes at the end of a test script. There are a number of things that are flakey. I'm also seeing the "Form is confused" issue now.

The White Screen of Darn

DDJ: Is that the "White Screen of Darn" (WSOD) issue? (See Figure 1.)

Re: Who needs Tony Snow...

Re: Who needs Tony Snow...

Bill V.: Yes, that's the "White Screen of Darn".

DDJ: What are some recommendations on ways to get back from the WSOD? I run into that one a lot myself.

Billy H.: That one is the most baffling to me. It seems so intermittent, and there doesn't seem to be anything specific that causes it. But, restarting the IDE seems to fix it.

Kathleen: The ability to go into the Windows Forms designer generated code is required. For example, with the form that eats the TreeView instantiation.

DDJ: That's the file that shows up in the Solution Explorer under the form file, the one that's hidden in VB projects? (See Figure 2.)

Kathleen: Right.

Bill V.: I've found that if I wear white socks, the problems don't happen nearly as often.

(laughter)

DDJ: For me, it's in working with user controls. When you're developing user controls, it's inevitable that you will modify the interface to the control. You'll rename properties, for example. If this control is sited on a form, then the designer generated code in that form will no longer be valid. When this happens, the form will no longer open in the designer. To get the form to open again, you have to go into the designer file that says all over it, "Designer generated, don't modify this!" You have to modify that code to fix the problem and get your form showing again. In the past, if there was a problem with a user control you could still at least see the form. The user control would show up with a red "X". You could delete the control, and re-add it, and probably fix the problem.

Kathleen: I've been lucky in the mentoring stuff I do that we don't have to go into that file very often. Unfortunately at this stage, even with folks new to .NET, you can't pretend that the designer file doesn't exist. You absolutely will have to go in there from time to time to fix problems.

DDJ: Just to narrow this down, some people are running third-party products, but there are still issues even if you're running a clean load of Visual Studio, with nothing third-party installed, correct?

Bill V.: I'm still seeing issues without anything third-party installed.

Re: Who needs Tony Snow...

Re: Who needs Tony Snow...

DDJ: It can't be attributed to beta debris. I can't be attributed to third-party products. Those might exacerbate the problem, but even without those, there's significant instability. Is it possible that maybe this is a perception issue? Is it possible that Visual Studio 2005 is really just as stable as any other release of Visual Studio? When the previous version shipped, blogging hadn't taken off. Every time someone found a bug, it didn't get broadcast around the world. Is blogging the problem? Has it just created an impression that the product is less stable than previous releases, or is there really something different about this one?

Kathleen: It's different with this one.

Rocky: I think it has to be put in perspective. I think it's roughly comparable to the very first version of Visual Studio .NET.

Bill V.: I was about to say the same thing.

Rocky: I think it's quite similar to the original Visual Studio .NET. But then the next version, Visual Studio 2003, had to be one of the best releases for stability that Microsoft has ever done. It just came out of the gate really solid.

Bill V.: 2003 didn't have a lot of new features, but they really did stabilize the product with that release.

Rocky: Yep, I agree. I think the blogging makes it seem worse than it is. And if you're like me, and you get hit with some beta debris, it's really bad. But, having fixed that, and hopefully after I install the hotfix, I'm hoping that almost all of my common issues will be solved.

Kathleen: I did not personally have as many problems with the original Visual Studio .NET as I'm having with Visual Studio 2005. Maybe I'm banging on the product differently now, but I did not have to close and reopen Visual Studio nearly as often. I don't think it crashed as often. And the weird stuff, like having a dialog appear on the left, when it's docked on the right, and then when I click on it and give it focus, it pops back over to the right where it's supposed to be. I don't recall seeing that kind of weird stuff in the original Visual Studio .NET. Now, I did switch to 2003 as quickly as I could, so it's possible that there were more problems than I'm remembering. It's a matter of degree. To me, it's somewhat less stable than the original Visual Studio .NET, and no where nearly as stable as Visual Studio .NET 2003.

Re: Who needs Tony Snow...

Re: Who needs Tony Snow...

Billy H.: I have to pretty much come in at the same level as Kathleen. I did a lot of production work in the original Visual Studio .NET, and my recollection is that I'd have to restart the IDE to get it to regain its sanity maybe twice a week. With Visual Studio 2005, it's more like twice a day.

(PAGE 4)

High Expectations

DDJ: Another hypothesis that I was going to forward, but I think you've preemptively shot it down, is that maybe Visual Studio 2003 set expectations artificially high. In a lot of ways, Visual Studio 2003 was just a massive service pack for Visual Studio .NET. It's logical that 2003 was very stable because it didn't introduce very many new features. It was mainly a big cleanup of the original Visual Studio .NET. However, even so, compared to the original Visual Studio .NET, Visual Studio 2005 is less stable.

Kathleen: It's worse by degree. It's not night and day. Even with the stability problems of 2005, I would choose to work with 2005 over 2003 any day of the week because of the features in 2005. I won't go back.

Bill V.: Agreed.

Kathleen: 2005 isn't so bad that you feel like you can't use the product. However, I do know people who are not moving to 2005 because of what they've heard about stability. They don't want to use something that still feels like a beta.

DDJ: I think some people would say some of these issues aren't technically stability problems. Instead, it's working as designed, the proverbial, "It's a feature!" For example, if the user control throws an exception at design time, or if there's an error in the designer generated code because you've changed the control and that code is no longer valid, then the WSOD, by design, is what you're supposed to see. Do you feel that this is really stability problems, or are things working as designed, but the design isn't really adequate in some areas?

Bill V.: I think a lot of these issues did show up in Ladybug (Editors note: also known as the MSDN Product Feedback Center.), and they were tagged as "Won't fix", or "By design". This happened for issue after issue after issue. Now we're looking at Orcas, and we're talking all these new features, we're talking about LINQ and things like that, and I think Orcas should be the 2003 type of release. It should be bug fixes, stability issues, and working

Re: Who needs Tony Snow...

Re: Who needs Tony Snow...

really hard on the documentation and discoverability of features that they already have. Push all those new features out to the Hawaii version.

DDJ: Is there any way, today, to obtain the hotfix without actually picking up the phone and calling PSS?

Bill V.: That's totally ridiculous. I called PSS; I talked to three different people. It was a painful waste of almost an hour of my time that I'll never get back.

DDJ: There was a petition. I looked up on the MSDN Product Feedback center this morning, and there was a recommendation that Microsoft not ship Visual Studio 2005. The recommendation was, "Have a Beta 3. It's not ready to release." I don't know when the voting closed on that feedback, but this morning, it had something like 230 votes. For something on the MSDN Product Feedback center, that's a lot. Writers for trade publications picked up on it and wrote articles saying that Microsoft's customers feel that it's not ready. On the other hand, all we had to look at were CTPs, and with CTPs, all bets are off. They are a snapshot, but the quality of a CTP is unlikely to accurately reflect the quality of the final product. Did people see this coming? Should Microsoft have known? Did Microsoft know that the quality wasn't going to be near that of 2003?

Kathleen: I think to get a stable product, we have to have something in the field that's about as good as what we have right now. Now I would prefer to call a Beta, and shake down something that's 98 percent of the way there. I think that would get the final product correct. But we desperately needed this releases to ship last fall. That's particularly true in Windows Forms because we can't release production stuff on Beta software. If I was inside of Microsoft, and someone had said to me, "You make the call. Do you ship now or wait?" I'm not sure I would have said to wait. Even though it's a pain to work with, there were a lot of people out there waiting for this to release. They had stuff that they wanted to get to the field. The .NET runtime is stable, so I'm not sure I would have delayed the release because of the development environment issues.

DDJ: Let me get other people's impressions on that. Given what we know now, if you had to make the call to ship back in November, would you have pulled the lever to ship it? (silence)

Bill V.: That's a tough call.

Billy H.: It is.

DDJ: Let me maybe make a distinction. Are we talking about something that's

Re: Who needs Tony Snow...

Re: Who needs Tony Snow...

a tool problem, or is it the tool and the Framework? In other words, is it an issue where the development environment is unstable, but the underlying .NET Framework is solid, so that you can have very high confidence that the applications that you build against the .NET Framework will run well, and be stable for your customers, even if the tool that you're using to build them gives you grief?

Bill V.: Some of the problems related to Data Access are in the Framework, and the tools can only be as good as the Framework. I think it's both.

Billy H.: I don't have anything in production yet on 2005, but I haven't seen any issues with the Framework that concern me.

Kathleen: In my experience with Windows Forms, it's pretty good. There have been a couple issues, but I'm not worried about release.

Bill V.: One thing about the Framework, if you do run into an issue, you just code around it. You just know that there's a pothole in the road at mile 45, and you drive around it. It doesn't have to affect the stability of your application.

(PAGE 5)

Framework Issues

DDJ: So any Framework issues that you have seen fall into the category of things that are repeatable, not intermittent?

Bill V.: The thing I've seen have all been, "Okay, the Framework does it this way, so I'll have to code it like this."

Kathleen: I feel the framework does hit the mark for stability.

DDJ: I've focused on stability, but looking at the product as a whole, and you've alluded to the fact that there's so much new functionality in here. What do you recommend to people? Do you recommend that they wait for the first service pack? Do you recommend that they wait for the next release? Or, do you recommend that they move forward to 2005, as-is?

Kathleen: Move forward.

Bill V.: Move forward.

Rocky: Move forward.

Re: Who needs Tony Snow...

Re: Who needs Tony Snow...

Billy H.:Move forward.

DDJ: Wow. The responses came pretty quick, and were unanimous that people move forward. So even though we've spent a lot of time taking about stability issues, when taken as a whole, you feel the product is a big improvement? You feel the productivity enhancements and the new Framework are that compelling? And is that across the board? Do you make that recommendation for Web and Windows, for C# and Visual Basic?

Kathleen: I can't speak for Web, but for Windows Forms in C# or Visual Basic, I recommend moving forward to 2005.

Rocky: In my mind, it includes all of them. Magenic has been doing Web, Windows, Visual Basic, and C#, and no one is willing to go back to 2003. Kathleen: I won't take a job that makes me go back. I will turn down work before I will develop using Visual Studio 2003.

What I think Microsoft should do for their long term plan is have a point release planned for one year after a major release like 2005. For clients that I'm working with, it would be really nice if I could say, "Look, lets just wait three months. A year after release, you're going to see an update. I understand that you don't want to feel like you're working with a Beta product." If we knew that was coming, and that was the focus of the next release, then those people who need that comfort level could wait. And then another year, or year and a half after that, we could see spectacular new features, because like you said, we do think everyone should move forward.

Rocky: This might be a bit of a tangent, but I think that historically with ..NET, Microsoft has tried to synchronize the release of Visual Studio to a curve based on a new release of the .NET platform. That didn't use to be the case, especially in the VB world. COM and Windows didn't really change much, and we got VB 4, 5, and 6. I would like to see us return to that model. The focus has shifted so much to be on the framework that we've lost sight of what the dev tools can do for us. In some ways, that's too bad. Instead of Visual Studio leading the way, and saying, "Hey, we could solve this issue, or that issue." And then later the Framework could come along and incorporate the idea. Instead, it's the other way around, and the .NET platform always seems to be ahead of the tool.

DDJ: And that's what's driving a lot of this problem. I'm hearing people say, "Make Orcas the stability release," but we know that a lot of new technology is coming. We know that LINQ, WPF, WCF, and WWF are coming. We know what people will say if there's no designer experience for those. Do we really want Microsoft to focus on stability in Orcas, at the expense of support for these new Framework things? Do we want Microsoft to focus primarily on great tool support for these new paradigms, or is stability

Re: Who needs Tony Snow...

Re: Who needs Tony Snow...

more important, even if the tools don't support those paradigms as fully as we might want?

Kathleen: Well, I could wait for things like LINQ.

Rocky: With Orcas, they're focusing, hopefully, on stability, and then only support for the new layers in WinFX. I think if they get in the habit of doing that, and this is probably a pipe dream, but then maybe we could see a release that's post-Orcas, that similarly says, "We're going to do a release of Visual Studio, and it can't change WinFX, it can just add new developer productivity."

Eclipse is an interesting model. It's independent of Java and any of the other technology that it targets. The upside is that they're able to do some things in the tool that Microsoft has locked themselves out of doing. But then to be fair, the down side is that Eclipse can't really integrate as deeply with a platform.

DDJ: We've all released software, and we all know that there's always a little bit of holding your breath for the first 24 hours after you've shipped. You hope that all your testing and work that you've done proves itself in the field, and the product works without a hitch. But we've all been part of things that have shipped that probably haven't worked as well initially as we would have hoped.

Let's just say, for the sake of argument, that anyone can ship a buggy release every once in a while. So now that it's out there, what do you, as a customer, want from your software vendor?

Kathleen: You mean in addition to a service pack that fixes these issues that we've discussed?

DDJ: Okay, so a service pack, is that what people are looking for at this point? Microsoft has said that they'll have a service pack in Q3 of this year. Is that good enough?

Kathleen: Other than just sticking their finger in a hole in the dike here and there, I think 9 months to a year is reasonable, and it's just what we'll have to live with. I want the service pack done right.

Billy H.:It depends on accessibility to the things that the service pack will contain. If I have to be on the phone with PSS to get a hotfix, then I want the service pack sooner.

Kathleen: I think the hotfixes are the fingers in the dike. If they need us to sign something saying that we understand that the hotfixes have risks, and they haven't been tested to the same level as a service pack, if that's what it takes to get them into our hands sooner, then I'm all for that.

Billy H.:And that takes the problem of blogs, where information spreads very

Re: Who needs Tony Snow...

Re: Who needs Tony Snow...

quickly, and turns it around so that blogs help instead of hurt. That's where people can find out about a hotfix, and if the hotfix is easily available, and you don't have to call PSS to get it, then that's good. On the other hand, if you see in a blog that there's a hotfix, but you have to call PSS, then it just makes the whole fact that you even need this hotfix more frustrating.

(PAGE 6)

Decoupling the Tools

DDJ: Rocky, one thing that you touched on was that you'd like to see the tools decoupled from the Framework so that they could rev independently.

Rocky: I don't think it's healthy for the platform to change radically and often the way it's been changing.

DDJ: What about other dependencies. If Visual Studio 2005 didn't ship, then SQL Server 2005 wouldn't ship, and Team System wouldn't ship. There's a log jam of Microsoft products backed up behind Visual Studio 2005. The dependencies don't just go up the stack from the Framework to the development tools, but also across some really large products. What are your thoughts about the number of things dependent on Visual Studio 2005, and version 2.0 of the .NET Framework. Does that make the problem worse?

Bill V.: Dependencies have always been an issue, and it's a growing issue. The operating system will be dependent on the Framework. Office will be more dependent on the Framework. As these other teams want change, it just makes things far more complex than it used to be.

Rocky: I agree with Bill, and yet I honestly think that it has the potential for an upside. One problem that we often face is that Microsoft often tends to hire a lot of young, smart people, who have never worked in business. They've never worked in IT. They build compilers, and they build tools, and I think they generally do a wonderful job, but they really don't have any idea what it's like for us. We build stuff that's always dependent on Windows, .NET, and SQL Server. By definition, all the stuff that we build, whether it's Windows or Web, is dependent on a broad set of Microsoft technologies. And we're vulnerable to them changing any one of those pieces. Well maybe, Microsoft itself, by accepting some of these dependencies, will start to understand our pain, and will come up with interesting ways to address it. Basically, if they fix their own problem, they will inadvertently fix many of our's also.

DDJ: That's an interesting perspective. On one hand, yes, the SQL Server team probably filed a lot of bugs. On the other hand, the SQL Server team was probably saying, "You have to ship, because we have to ship!" And I guess that's no different than customers who start building products based on a Beta version, because they've been "promised" a certain release date for Visual Studio. They also need the Framework to ship, because they can't

Re: Who needs Tony Snow...

Re: Who needs Tony Snow...

ship until it does.

Kathleen: As somewhat of an aside, the most expensive decision I ever saw regarding moving forward was to not move forward. Not moving forward can be as expensive, or even more expensive, than moving to the newest version. It turns into this big gamble for companies that have real people driving trucks or making doughnuts, to decide when to move forward to a new technology, and a lot of it comes down to how much they can trust the release dates announced by Microsoft or any other company.

Billy H.: I guess I'm just pessimistic that this won't get any better. The problem is that we have so many platforms now. We have the Windows platform, and the .NET platform, and the SQL Server platform. Soon we'll have changes in the form of WinFS, which changes the platform. You pretty much have to have all that stuff working right, before you put anything on top of it. As bad as it is to have products wait on other products, I think it's even worse to do things incrementally. Part of the problem today is that Windows XP doesn't have the .NET framework on it. The platforms got out of sync. As much as I hate the delays, I think the platforms have to be synced up, to be the fundamental foundation on which we build.

Rocky: I really don't think this current rate of change is sustainable in the long run. I don't think that the corporate world is going to put up with it. For any investment they make in Windows or Web right now, they have to be concerned about how much of that will change with Avalon. If Microsoft turns around after Avalon, and comes out with more radical changes, I think they really do run the risk of losing people.

DDJ: It sounds like the core question is quantity vs. quality. I think the question to Microsoft is, "You know that you're going to have tool support for all these new technologies, like WinFX, Atlas, and so on. These new technologies aren't small things. How can you add in tool support for all those technologies, in a relatively short time frame, and still insure that the tool is stable and high quality?" It seems like a tall order.

Kathleen: But we all said that we want the new features. We all said that we wouldn't go back.

Bill V.: I think the third parties could expand their role. If they knew that the platform would be stable for a long time, and they weren't going to have to rewrite their stuff every couple of years, they could go a lot farther.

DDJ: But we've all said, "Please, Microsoft, put this feature in the box so that I don't have to go to a third party for something that's this fundamental. I don't get the same support from a third party. Their stuff is often practically undocumented. It's not as stable. Loading add-ins can hurt the stability of Visual Studio." Don't get me wrong, I end up using 3rd party products in every application that I work on, but it's one more layer of stuff that can cause problems.

It almost sounds like we don't know what we want. On one hand we're saying,

Re: Who needs Tony Snow...

Re: Who needs Tony Snow...

"Stop! You can't change this fast." And then on the other hand, we're saying, "The changes you made for 2005 are so good that I'll never go back." The changes that they make for 2007 will probably be just as great. As soon as we see them, we'll say that we can't live without them, and we'll say, "What's taking so long to ship the dang thing?" Once we get it get it, we'll say we won't go back. If it's not stable, we'll cuss about it not being stable, but we'll use it and recommend it anyways. Are we just at a point where we have to resign ourselves to a certain level of frustration? The technology is going to change fast, and while in abstract, we don't like the rate of change, when we see specific features, we want them very badly. We'll push to have it ship soon because we want to use it for new products, and then when it does ship, we'll point our fingers at it and say, "It should have been more stable."

Rocky: The problem, I think, is that it's all being driven by geeks, and we're geeks too. Most business applications are forms over data. All of these new technologies exist to feed the technology frenzy. They don't actually make it any easier for the user to type in the data that gets entered into the database. None of this stuff is really benefiting the user. It makes it prettier, whatever, but it's all technology driven for technology's sake. That's all very cool, and as geeks, we all love that. Microsoft loves it because they're all geeks too. But I just wonder how sustainable this rate of change is in the long run.

(PAGE 7)

Microsoft Response to "Visual Studio 2005: Unstable but Highly Recommended"

This article raises a number of different and widely varied issues regarding the release of the Visual Studio 2005. Microsoft is proud of the quality of Visual Studio 2005 and the .NET Framework. We worked with thousands of customers through the development process to ensure the quality of the product. At the same time, these are complex software projects and we recognize that some customers may run into issues. We take these issues very seriously and are working to ensure that they're addressed in a timely fashion.

There are a couple of interesting points that I want to highlight from the article. The first is that everyone who participated in the interview recommended that people immediately adopt Visual Studio 2005 and the .NET Framework 2.0. They feel, as we do, that the improved functionality is a tremendous step forward for developers regardless of whether they are building Windows, Web, Office or Mobile applications. The second interesting point is that the issues that were raised were almost universally around the behavior of the tool.

The article starts by raising issues that cause product instability with the designers and the compiler. Several of the solutions that the MVPs

Re: Who needs Tony Snow...

Re: Who needs Tony Snow...

recommend, such as uninstalling third-party products and disabling Edit and Continue, are things that we asked them to try as a part of helping us to debug the issues that had been raised by the community. Some of these issues are now addressed in the HotFix available by calling Microsoft's Product Support Services and referencing KB 915038. We are in the process of combining this HotFix with another fix that is currently being verified by customers and will post these as a freely available General Distribution Release (GDR). As the article points out, we are also working towards a general Service Pack that will be available in the third quarter of this year.

However, Microsoft has never recommended either uninstalling third-party products or disabling Edit and Continue as a part of a permanent solution to the problem. The interviewees also raised some questions about Developer Express' Refactor! product. There was an early issue with that product that impacted developers who had installed the Beta, but that has now been addressed. The Visual Basic team worked with Developer Express because we felt that this product added significant value to the VB development experience and we continue to believe this.

The White Screen of Darn (WSOD) is the screen that is displayed to users when a designer fails to load. In previous versions of Visual Studio we tried to show the designer at all costs, and only showed the WSOD in extreme cases where we could not locate a designer. In Visual Studio 2005 we changed this behavior, since showing a broken designer can cause loss of work where broken controls are deleted from the designer and all of the work put into creating and customizing them is lost. By using the WSOD we are trying to bring attention to the errors that appear in the Visual Studio task list so that the user does not continue to work on a broken project. Errors that can cause the WSOD include broken user code that is run at design time, or a deleted or missing reference that is relied upon by components in the project. We recognize that these issues are sometimes hard to diagnose, and we are working on ways to make the process easier.

Beyond the specific issues discussed above, we remain committed to producing the industry leading development experience in terms of functionality and product quality. One of the key initiatives in doing that has been the Developer Division's focus on increased transparency throughout the development cycle. This has manifested itself to developers through efforts like the MSDN Product Feedback Center and our frequent releases of Community Technology Previews (CTPs). Because of these and other community initiatives, we've received significant feedback about the product, which has given us an enormous amount of valuable information about what customers love and about areas where they feel that we can improve.

A related issue that's raised in the article is how Microsoft can guarantee a quality release for Visual Studio "Orcas" given that we are adding new

Re: Who needs Tony Snow...

functionality to support WinFx. This effort continues to build on the product quality and transparency initiatives that we initiated in the Visual Studio 2005 release. Customers can expect to see us releasing CTPs throughout the product cycle—including some of the early CTPs that we released before the Visual Studio 2005 product even shipped. We encourage customers to download these CTPs and to provide feedback through the MSDN Product Feedback center. We realize that not all of our answers will be the "thing you want to hear" but we are committed to evaluating each issue that's raised and addressing it as appropriate.

As software, including Visual Studio, gets more complicated we need to enhance our processes in order to ensure that we continue to ship quality software. In an earlier blog entry, S. Somasegar, the vice president of the developer division, talked about our MQ efforts and the ways that we're improving our internal processes. Additionally, we've made a conscious effort to ensure that the "Orcas" release is focused on the right features to meet customer requirements. We know that professional developers will depend on us for providing tools support for Windows Vista and the 2007 version of the Office System, so we have concentrated team efforts on ensuring that our technology solutions in these spaces are rock solid.

(PAGE 8)

Panel Bios

Kathleen Dollard has been developing business applications for over 20 years, programming in Visual Basic for almost ten years, and working with .NET since the early betas. As an independent consultant, she has had the opportunity to work in a variety of domains, including the finance and justice sectors. Kathleen has worked extensively with application code generation and is the author of Code Generation in Microsoft .NET (Apress). Kathleen is also a long time Microsoft MVP, president of the Northern Colorado .NET SIG, and is an active member of the Denver Visual Studio User Group.

Billy Hollis is the author or co-author of a number of books on .NET, including the first book ever published on Visual Basic. NET. Billy is a Microsoft Regional Director and MVP, and was selected as a .NET Software Legend in 2002. Billy has his own consulting practice for training, consultation, and software development on the Microsoft .NET platform, focusing on .NET architecture and design, advanced smart-client systems, Tablet PC, and commercial packages.

Rockford Lhotka is the author of several books, including Expert VB and C# 2005 Business Objects and related CSLA .NET framework. He is a Microsoft Regional Director, MVP and INETA speaker. Rockford is the Principal Technology Evangelist for Magenic (www.magenic.com), a Microsoft Gold

Re: Who needs Tony Snow...

Re: Who needs Tony Snow...

Certified Partner.

William (Bill) Vaughn is an industry-recognized author, mentor and subject-matter expert on Visual Studio, SQL Server, Reporting Services and data access interfaces. He's written six editions of the Hitchhiker's Guide to Visual Basic and SQL Server. He and Peter Blackburn also wrote the best-selling and critically acclaimed Hitchhiker's Guide to SQL Server 2000 Reporting Services.

Scott Swigart spends his time consulting, authoring, and speaking about converging and emerging technologies. With development experience, going back over 15 years, and by staying in constant contact with future software development technologies, Scott is able to help organizations get the most out of today's technology, while preparing to leverage the technology of tomorrow. Scott is also the author of several .NET books, a certified Microsoft trainer (MCT) and developer (MCSD), and a Microsoft MVP. Feel free to contact the Scott at scott@xxxxxxxxxxxxxxxxxxxxxxxx

--

Randy Birch
MS MVP Visual Basic
<http://vbnet.mvps.org/>

Please reply to the newsgroups so all can participate.

"Grant" <email@xxxxxxxx> wrote in message
news:%23E%234khsaGHA.4916@xxxxxxxxxxxxxxxxxxxxxxxx
Can it be copy and paste here, my company has blocked that site.

Thanks.

--

Grant

Who gives a {censored} if I am wrong.
"Karl E. Peterson" <karl@xxxxxxxx> wrote in message
news:OokqmukaGHA.3612@xxxxxxxxxxxxxxxxxxxxxxxx

... when you have Rockford Lhotka, Billy Hollis, Bill Vaughn, and Kathleen Dollard?

Bush should hire this group of kool-aid drinkers, eh?!

Dr. Dobb's | Visual Studio 2005: Unstable and Highly Recommended

Re: Who needs Tony Snow...

Re: Who needs Tony Snow...

http://ddj.com/dept/windows/186500706?cid=RSSfeed_DDJ_Windows/.NET

The headline doesn't even *hint* at the absurdity of this piece.

Put down your drink before reading. You have been warned.

--

Working without a .NET?

<http://classicvb.org/>