

# Early Feedback on Visual Studio 2010 Beta

*So far, the reviews of VS 2010 and .NET Framework 4 are encouraging.*

**By Kathleen Richards, Jeffrey Schwartz 07/01/2009**

Developers expected to see Visual Studio 2010 and .NET Framework 4 beta 1 in the Tech-Ed time frame -- and Microsoft didn't disappoint. On Monday, May 18, the first business day following the educational conference, the company made Visual Studio 2010 and .NET Framework 4 betas available for MSDN subscribers to download. Public downloads of the bits were released a few days later.

Beta 1 gives the broad .NET community a closer look at what's coming in Visual Studio 2010 and the revamped code editor built using Windows Presentation Foundation (WPF). As reported in last month's cover story, ".NET 4 Revealed," the updated framework promises maturing class libraries, new parallelism capabilities and a major upgrade to the ADO.NET Entity Framework (EF) that debuted in .NET 3.5 Service Pack 1 (SP1).

In late May, Microsoft continued its emphasis on the cloud, releasing an updated community technology preview of its Azure Tools for Visual Studio, which added support for the Visual Studio 2010 beta. The Azure Tools provide C# and VB templates for cloud-based projects, debugging of the local development fabric and storage, and access via a WindowsLive ID to the Azure Services Developer Portal. The Visual Studio extension installs the Azure SDK, which Microsoft also updated in May. The caveat: Windows Azure and .NET Services work with .NET 3.5 SP1-they do not yet support .NET 4.

## **Trials and Triumphs**

Early reports on the Visual Studio 2010 and .NET Framework 4 beta are fairly positive, considering these are still technology previews. Developers should be able to install the Visual Studio 2010 beta side-by-side with Visual Studio 2008 SP1 on the Windows 7 RC (not the Windows 7 beta) without any issues, according to Microsoft.

Joseph Megkousoglou, associate lead software architect at property services firm Knight Frank LLP, lauds the clear separation between managed and native code in the new IDE's installer. He is also pleased with the performance of Visual Studio 2010 on the Windows 7 RC. "I would happily use it as my main IDE, speedwise," Megkousoglou says.

The Team Foundation Server (TFS) 2010 installation is another matter, however. "[It was] quite long and involved," he explains. "As a Subversion guy, I'm still trying to see the benefits of all the features TFS offers."

Still, the new bits have not caused tremendous headaches. "I've already loaded our main projects in Visual Studio 2010 and compiled them with .NET 4," Megkousoglou says. "No major problems there. Performance is very similar to .NET 3.5. The main solution I loaded-comprised of nine different projects-is an ASP.NET application, uses LINQ and Enterprise Library components, and contains a total of 120K lines of code. This is one of our main applications, which receives a huge [number] of visitors every day."

Other developers ran into performance issues with the Visual Studio 2010 beta. "It's sometimes more sluggish to use than Visual Studio 2008 on the same hardware, but I'm hoping that will be resolved by the time it's released," observes Vaibhav Gadodia, a .NET architect at outsourcer Nagarro Inc.

"The biggest problem I have with it is stability," says David Barnhill, a senior consultant with Lab49 Inc. "It has crashed at random times on my machine. I haven't found a pattern to it. But the stability is not bad for a beta 1 product; I've had much worse experiences."

## Working WPF

The much-ballyhooed WPF editor part of the new Visual Studio shell is not the showstopper that some developers feared. "While I think the performance is OK, I'm sure it will get better," says Stephen Forte, chief strategy officer of Telerik Inc. "I'm sure beta 2 will have a full-fledged performance."

Forte is a fan of the new WPF shell. "The WPF editor will give us a ton of benefits," he notes. "You can tell right way when you look at it: It looks different and quite frankly it's easier on the eyes."

Despite the new look and feel, Visual Studio 2010 remains familiar and easy to navigate. "It's actually close enough to the development environment of 2008 where you can find your way around really easily," Barnhill says.

Developers who've downloaded the beta have commonly voiced dismay over certain issues. One is too much white space, or glyphs, which can make it harder to read code. In a posting in the Visual Studio Editor blog a week after the beta dropped, Brittany Behrens, program manager on the Visual Studio Platform team, noted that Microsoft will revert back to the Visual Studio 2008-style glyphs in the final release. Similarly, the widely used Box Selection-missing in beta 1-will be updated with new multi-line editing capabilities and put back into the final release.

Another point of concern for many developers is blurry fonts and fuzziness, especially for lower point sizes. "One of the big things they're going to fix is the text in the display; [it's] a little blurry now," Barnhill says. "There are some problems that are in WPF's font rendering that they'll be fixing in the beta 2 of .NET Framework version 4-that's something that a lot of people really hate."

WPF only supports TrueType fonts, and other bitmap fonts still appear on the menu in Visual Studio 2010 beta 1. Those fonts will be removed in the final release, according to Behrens. (To learn more about common editor issues in beta 1 and Microsoft's fixes, you can take a look at Behrens' June blog posting at <http://tinyurl.com/p93xhm>.)

Gadodia says he's willing to wait for Microsoft to add the Compatible Width Layout stack in beta 2 to address WPF's font rendering. "It would be crazy to sit in front of an IDE that has these problems," he notes. "Since a lot of the UI is WPF-based in Visual Studio 2010, it's definitely more painful."

Despite these issues, Barnhill applauds the new shell, which is built on WPF and the new Managed Extensibility Framework. "It's just a much better technical base to start with than before," he says. "It's a lot easier for developers to make extensions to the editor and make it work differently without running a little C++ code. You can do it right in WPF and C#."

The historical debugger and new multithreading debugging functionality that Barnhill calls "parallel stacks" are also impressive in beta 1. "I really like the historical debugger, but one issue with that is it's only available in the Visual Studio Team System (VSTS) Developer Edition. It doesn't come with Visual Studio Pro," he explains. "I like, overall, the design."

The "parallel stacks" used for debugging multithreaded applications enable developers to set a break point in their source code so that they can figure out where each thread is at a point in time. "The performance of that is fine," says Barnhill. "When you run it and get to that break point, that's the only time you can see your stacks; that parallel stack only makes sense when you're at a break point in your code."

The early looks at the first version of Parallel LINQ (PLINQ), which is rather limited, are encouraging. "It worked as advertised as far as being able to easily take a collection of objects and do parallel operations on those objects," says Barnhill. "It was very easy to do, and more [efficient] than the old way you had to do it."

### 'Oslo' Preview Debuts 'Quadrant'

Microsoft released a new CTP of "Oslo" in May, giving developers a first look at "Quadrant," the code name for the graphical tool intended to provide browsing of models. Oslo is the code name for Microsoft's model-driven development platform that runs on a SQL Server 2008 repository.

The May 2009 community technology preview (CTP) updates the SDK with runtime and provides the first public look at Quadrant. (An alpha version of Quadrant appeared in the Virtual PC handed out to attendees of Microsoft's Professional Developers Conference 2008, but the graphic modeling tool was not in the public CTPs.)

"What's interesting about the [Quadrant] tool is, now you can put things into the repository and you now have the ability to view them graphically, as opposed to just using command lines or SQL statements," says Stephen Forte, chief strategy officer at Telerik Inc. "It's obviously in an early form but it looks pretty good. You can really work with your application data and metadata much better."

The latest CTP also adds support for UML 2.1 domain models-use case, component diagrams, activity, sequence-and the respective XMI importer.

Also noteworthy is the M editor called Intellipad. It enables developers to compile DSLs created in the M modeling language into image files, rather than having to use a command-line interface, explains Forte.

"You can actually author M in Visual Studio much easier now with this SDK," Forte says. "You could in the past, but integration is even tighter. For example, if you create and end the project in Visual Studio, you can now open that same M project in Intellipad, which you weren't able to do before. So the tooling definitely interoperates nicely."

-J.S.

## Entity Framework 4 Unleashed

With the release of the Visual Studio 2010 and .NET 4 betas, developers are getting their first look at the ADO.NET EF 4, which is Microsoft's preferred model for building applications that access databases.

Given much of the backlash about the first release, the update has been eagerly awaited. EF 4 adds support for n-tier APIs and templates, increases Plain Old CLR Objects (POCO) coverage and improves Persistence Ignorance, in addition to other improvements, according to Microsoft.

"The Entity Framework itself has pretty much undergone a radical transformation," Forte says. "It really addressed some of the concerns of the community in that respect."

Nagarro's Gadodia is among those who passed on the first version of EF. "We were pretty happy with LINQ to SQL," he notes. "We had invested a lot of training effort as well, since it was-and is-such a shoe-in for that ORM layer."

Of course, many who invested in LINQ to SQL felt jilted last fall when Microsoft shifted its focus to EF. With EF 4, Gadodia notes that a lot of the features of LINQ to SQL are now available within EF. "It almost feels as if Microsoft copied some of these into EF from LINQ to SQL as a result of developer feedback," he says. "We'll continue to use LINQ to SQL in the short term in our current projects, for the simple reason that most of our developers are trained in the technology."

Among other features he likes in the updated EF are things like the modeling support, which now generates DDL based on the model. "There are other changes which are useful; for instance, POCO support," he adds. "We're developing a new internal application framework to use across projects, and changes in EF have put it in the front-running for using it in our framework."