

Microsoft Visual Studio 2008: An Insider's Personal View

Daniel Moth of Microsoft highlights some of his favourite new features.

For more of the same, check Daniel's blog, <http://www.danielmoth.com/blog>

By now you will have heard that Visual Studio 2008 was made available to MSDN subscribers in November 2007 and will be broadly available in February 2008. If you are not an MSDN subscriber you can still download a 90-day trial of Visual Studio 2008. After that, if you decide to purchase you will be able to upgrade in-place by entering your product key.

So what do we get with this release and is it worth the upgrade? In this short article I'll set things in context and then give you a tour of the new features pointing out things you should look for in your VS2008 installation.

First, let's take a trip down memory lane. Previous releases of Visual Studio were aligned and tied to a specific version of the .NET Framework, which itself was aligned with and tied to the Common Language Runtime (CLR) engine (Figure 1):

	2002	2003	2005
Tool (Visual Studio)	VS.NET 2002	VS.NET 2003	VS2005
Languages	C# v1.0 VB.NET (v7.0)	C# v1.1 VB.NET (v7.1)	C# v2.0 VB2005 (v8.0)
Framework Libraries	NetFx v1.0	NetFx v1.1	NetFx v2.0
Engine (CLR)	CLR v1.0	CLR v1.1	CLR v2.0

Figure 1 - Previous releases of Visual Studio and .NET Framework

In 2006 Microsoft broke the mould by releasing .NET Framework v3.0, but crucially it did not release a new version of the CLR or new versions of the language compilers or a brand new version of Visual Studio. .NET Framework v3.0 comprises of four technologies: Windows Presentation Foundation (WPF), Windows Communication Foundation (WCF), Windows Workflow Foundation (WF) and Windows CardSpace.

Also in 2006, Windows Vista and the 2007 Office System were released, which both are development platforms in their own right. So developers have been looking for a refreshed toolset to properly address those three "platforms". That is exactly the gap that VS2008 fills by offering premier support for .NET Framework v3.0, Windows Vista and Office 2007. A select list of new features for

.NET Framework v3.0 in VS2008 is: new WPF designer, ClickOnce and XBAP deployment to FireFox, WCF support for an HTTP SOAP-less programming model, syndication support, partial trust for services and JSON serialisation. There is also integration of WF and WCF both with the new Send/Receive activities and also through a new WorkflowServiceHost class.

At the same time as VS2008, a new version of the .NET Framework is released (v3.5) and a Service Pack to v2.0 and v3.0. We also get brand new compilers for both Visual Basic and C#. The really cool feature of VS2008 is that it can target whichever version of the framework you are interested in and thus it can be seen as a superset to VS2005 (Figure 2).

It is for this reason that I personally call it "the no brainer upgrade". Not only it can co-exist on the same machine with VS2005, not only

good visual summary of which features are new per language (for more on each, follow the links from item #4 on the 2nd November entry of my blog post).

There are many additional aesthetic and usability features in VS2008. For example, right click in the code editor to see "Organise Usings", hit Alt+Tab to see previews of windows, right click in solution explorer to discover "Open Folder in Windows Explorer" and many more. VS2008 was designed and built as a great Vista client (and of course works nicely on XP too!) and hence it is UAC-aware and uses the new Vista dialogs.

But for me, the most impressive usage of VS2008 for existing projects is when targeting ASP.NET 2.0 web sites. Web developers get support for script intellisense (just type some script in the editor to see what I mean) and debugging (just set a breakpoint in your script and hit F5). They also get unprecedented support for Cascading Style Sheets (CSS): Go to the "View" menu and open the three brand new CSS tool windows (CSS Properties, Manage Styles and Apply Styles) and you'll be pleasantly surprised. If you are ready to move to ASP.NET 3.5, simply select in the combobox in your project properties ".NET Framework 3.5" and you are ready. Now you can use the built-in ASP.NET AJAX including the ability to select AJAX control extenders in the designer. You can also use 3 new ASP.NET 3.5 controls (ListView, DataPager and LinqDataSource) that make it very easy to connect with data and fully define the HTML that gets emitted to the browser.

Developers ready to move to .NET Framework v3.5 will find a bunch of new features not necessarily related to web development. For example, Office developers will be delighted to hear that they can create Office (both 2007 and

2007			
Tool (Visual Studio)	Visual Studio 2008		
Languages	C# 3.0, VB 9.0		
Framework Libraries	NetFx v2.0 SP1	NetFx v3.0 SP1	NetFx v3.5
Engine (CLR)	CLR v2.0 SP1		

Figure 2 - Visual Studio 2008 can target multiple Framework versions

Microsoft Visual Studio 2008: An Insider's Personal View

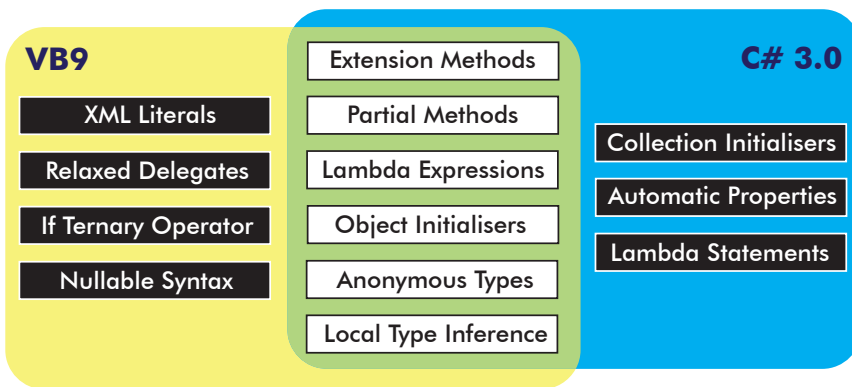


Figure 3 - C# and VB compiler features

2003) addins with the Professional Edition of VS2008. There are many new options available especially when targeting Office 2007 such as customising the new Office ribbon, custom task panes, form regions, actions panes, content controls and easy interop between managed and VBA code. From the feature list you can probably tell that in addition to addins at the application level (for Word, Excel, PowerPoint, Visio, InfoPath, Project and Outlook), you can also create addins at the document level (for Word

and Excel). One of the most requested features for Office addins has been an easier deployment model and we get that with full ClickOnce support. All in all, I would encourage all managed developers building client applications to give Office development a try because it has truly become a breeze with Visual Studio 2008.

I hope that by reading this article so far you can see that there is something for all types of developers (client, service, web and office) in VS2008. Worthy of note is that Unit Testing is supported

Daniel Moth has significant industry experience, holds a BSc in Computing for Business as well as an MSc with Distinction in Object Oriented Software technology, and he has been working with .NET since the first beta in 2000. Daniel was awarded the Most Valuable Professional (MVP) award for his community contributions in 2004 and still enjoys helping others wherever he can. In his spare time he loves to travel around the world, be it on city breaks, lazing on the beach or scuba diving the oceans. Reach him at <http://www.danielmoth.com/Blog>.

in the Professional Edition and it now includes support for device projects as well, which will make my mobility friends happy. Having said all that, there is one new feature in .NET Framework v3.5 that has deservedly stolen the spotlight. The feature can be used in web projects, client/office projects, mobile projects and anywhere where you write code really. It is called: the Language Integrated Query (LINQ).

LINQ (see Figure 4) introduces a SQL-like syntax into C# 3.0 and VB9 combined with rich extension method implementations that are part of Framework v3.5. As a developer you learn a single declarative pattern that encompasses filtering, grouping, ordering etc, and you can then apply it to all kinds of data sources: in-memory objects, XML, DataSets and SQL Server. The cool thing about LINQ is that it is fully extensible to incorporate any other data source and there are already 3rd party LINQ providers available and more from Microsoft in the works. In the future, it will be very easy for LINQ queries to be parallelised on multi-core/CPU machines with almost no code changes.

I hope you enjoy using VS2008 as much as I do. If you want to see the code in action for ALL the features I described above, please read my blog posts on <http://www.danielmoth.com/Blog> and watch my screencasts linked from the left on my blog page. Thanks for reading.

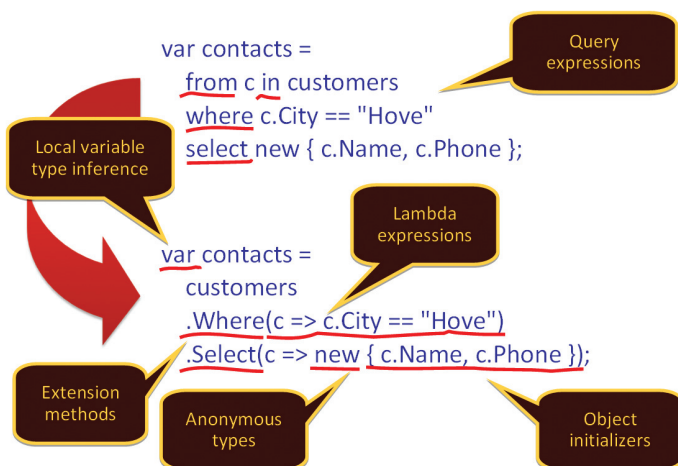


Figure 4 - LINQ and its relationship to language features