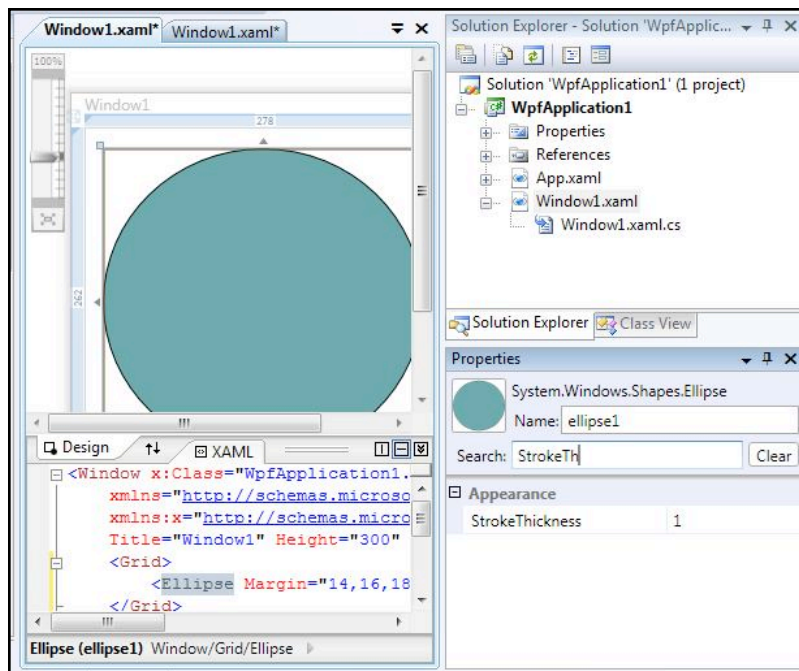


Christmas comes early: Introducing Visual Studio .NET 2008!

After a long beta period Microsoft pushed Visual Studio .NET 2008 (code named “Orcas”) out to MSDN in November - keeping their promise to deliver it by the end of the year. We’ve been using Orcas in many of our .NET classes for a while now and I for one, am pretty excited that it’s finally here. There’s a ton of new features and enhancements in this release that makes it almost a no-brainer to upgrade - I thought I’d take a moment and list my top ten favorites in no particular order:

#10: WPF designer (“Cider”)

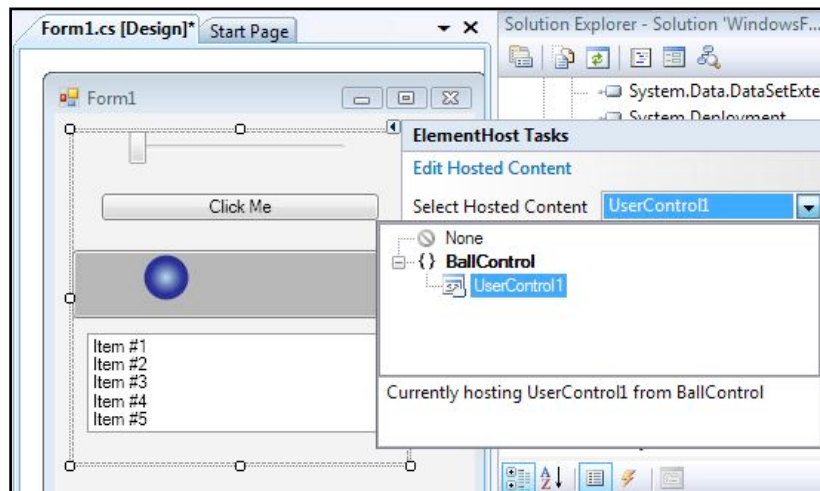
I spend a lot of time on rich-client island and compared to the embarrassing support for WPF provided for VS2005 (through an add-on) VS2008 just plain rocks. Not only does the designer *just work*, the XAML intellisense is full featured and driven through a real XAML parser and not an XSD file. That means custom types and namespace



completion actually functions how you would expect them to! The visual designer and XAML views stay synchronized and “mini” thumbnails and scaling bars are used to help you see what you are building. You can also see how the new Cider environment has borrowed from Blend - a Search box is present to filter properties down and the selection in the XAML pane is synchronized with the visual designer so if you highlight a tag, it selects the appropriate object in the visual pane.

#9: WPF and Windows Forms integration (“Crossbow”)

Hand in hand with the designer view, VS2008 improves the designer experience for Windows Forms to WPF integration. Specifically, **ElementHost** is now present in the toolbox and you can drag and drop WPF user controls right onto your Form! You get a live preview of the control and can edit the content right in the designer.

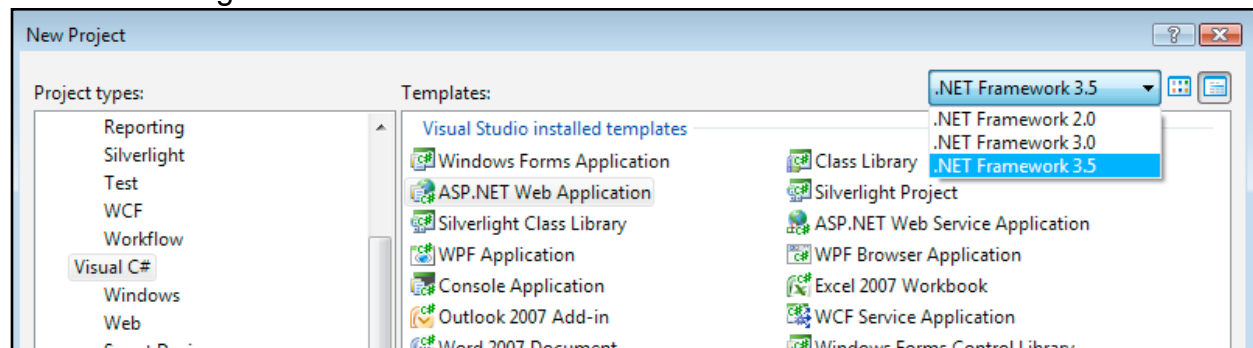


#8: ClickOnce improvements

Microsoft introduced a simplified way to deploy Windows Forms clients from the web with Visual Studio 2005 called ClickOnce. With 2008, they’ve improved the experience by allowing deployment through FireFox (previously only IE was supported), file associations so your application can be launched by activating a data file, better support for certificate expiration and changing the deployment location without resigning the application and support for Office add-in and WPF XAML Browser application deployment.

#7: Multi-framework targeting

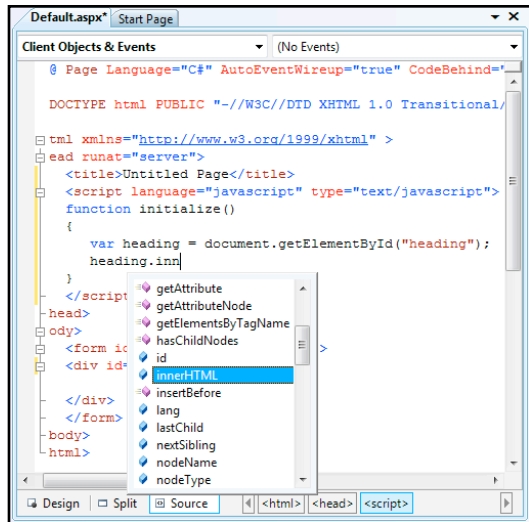
This release of Visual Studio has a much-needed feature that I wish had been there before - the ability to target multiple versions of .NET. Specifically, you can select your target framework (2.0 SP1, 3.0 or 3.5) and the project types, toolbox, references, intellisense and features will be appropriately set to that version. This selection can occur during project creation - in the project templates dialog or on the project properties dialog. Be aware that selecting .NET Framework 2.0 actually means 2.0 SP1 and not the original .NET 2.0 framework released with Visual Studio 2005.



#6: Better intellisense support

Web developers benefit from this release too - Visual Studio now has intellisense and code completion support for JavaScript!

It's smart enough to look at the underlying type currently assigned and correctly infer the methods which are available. So, for example if you assign a numeric type, the dropdown will be populated with the proper methods available for the type.



Visual Studio will also look at the comments applied to your methods - just like in C# and VB.NET, it will pull descriptions out of XML based comments on your JavaScript methods and display descriptions as tooltips when you are navigating the intellisense dropdown.

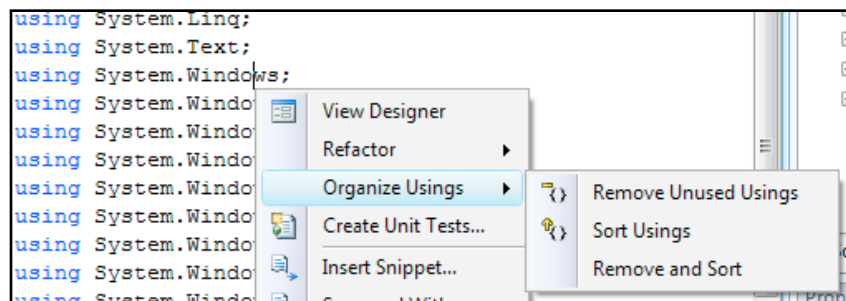
Another cool feature added to Intellisense is the filtering applied - now if you press a letter and start typing, it will begin to restrict your choices displayed, not just scroll to that section. This filters the output and makes it easier to find what you are looking for.

#5: Organize your using statements

Visual Studio now has the ability to organize your using statements. As a project evolves, you often end up

with a ton of using statements at the top of each file which are not really being used - either because the project template added it, or because you originally *did* use something from that namespace which was

then removed later. Now you can right click on your using statements and sort and/or remove unused namespaces.



#4: Refactoring enhancements

The refactoring support was a welcome addition to VS 2005 - and Microsoft has enhanced the support in 2008 to support C# 3.0 features and allow you to refactor anonymous types, extension methods and lambda expressions.

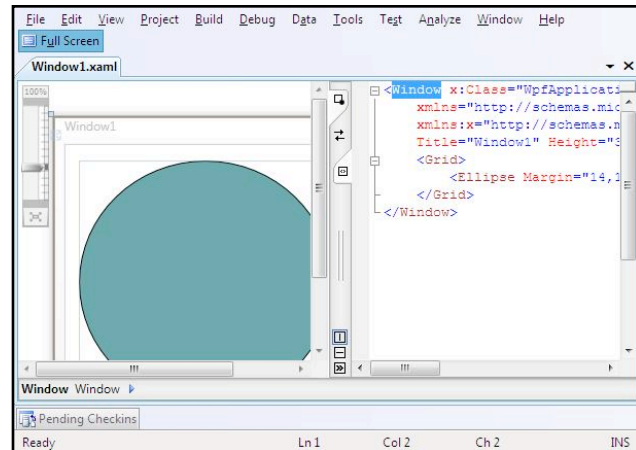
#3: C# 3.0 support

Speaking of C# 3.0, that has *got* to be one of the coolest features - functional programming seems to be all the rage these days and introducing these features into C# 3.0 will allow you to be the coolest programmer on your floor. Things like automatic

properties, lambda expressions (which reduce your typing for anonymous delegates), partial methods, anonymous types and extension methods will radically change how you can build applications. They can seem weird at first for some people, but don't be afraid of them! They will cut down your typing and provide for some really cool ways to enhance and evolve your projects.

#2: Visual Studio Split View

It is becoming increasingly more common to have multiple monitors on developers desks. Building on the "split" window feature of VS 2005, Visual Studio 2008 now allows you to tile the window horizontally *or* vertically so you can split your design and code view across monitors! This works in any of the designers (ASP.NET, WinForms, WPF, etc.)



#1: Debugging the .NET source code

There are actually several debugging enhancements (try debugging JavaScript on the client-side for example - it works great!) but I think the coolest feature by far has got to be the debugging enhancements which will allow you to actually debug *into* the source code of the .NET framework. Scott Guthrie announced a few weeks ago on his blog (<http://weblogs.asp.net/scottgu/archive/2007/10/03/releasing-the-source-code-for-the-net-framework-libraries.aspx>) that they intend to release the source code under the Microsoft Reference License. As part of that release, VS2008 will have integrated debugging support so that you can step into the framework libraries (ever want to know why your DataBind isn't working??). Visual Studio will automatically download the source file from Microsoft's server and you will get full watch and breakpoint support. This isn't quite ready yet, but it's coming soon and promises to be one of the biggest timesavers added to Visual Studio!

There's a ton of other features that make VS 2008 a worthwhile upgrade - you can experience it yourself by downloading a free copy today (Express editions have been released) or upgrading from MSDN. VS 2008 co-exists just fine with 2005 and 2003 but be aware that the project format has changed and it's not backward compatible so sharing projects with your coworkers may be slightly painful for you until everyone moves to the new release as you will have to maintain two project and solution files.