

Re: C# vs VB.Net

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2006-10/msg00187.html>

- *From:* "Michael D. Ober" <obermd.@.alum.mit.edu.nospam>
 - *Date:* Wed, 04 Oct 2006 04:18:24 GMT
-

I have updated your list for VS 2005.

"sloan" <sloan@xxxxxxxx> wrote in message
news:OV08Kev5GHA.4304@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

*** This is a 1.1 list ***. Some may or may not apply in 2.0.

VB.NET has no "using" directive.

Fixed.

VB.NET does not do automatic namespacing. Whenever you put in a new
folder

and/or class, C# puts in "namespace MyCompany.Technology.Args" for you.
In VB.NET you have to ~remember to put in "namespace
MyCompany.Technology.SomethingElse". Or everything defaults to the default
namespace.

True, but the VB module statement is an equivalent. In C# 2005 everything
must be in a class, but in VB 2005 you can place functions and subs in
modules and they will be global or accessible via the module name .
function/sub name.

VB.NET does not provide pre-build or post-build steps.
(there is a work around available for this , one is build rules ex

Re: C# vs VB.Net

<http://www.gotdotnet.com/Community/UserSamples/Details.aspx?SampleGuid=a3326eb3-a468-4f67-91a8-f84469f>

)

Haven't tried, so I don't know.

VB.NET has no built-in comment->documentation generator
(see previous forum post about GhostDoc)
(there is a work around available for the default c# functionality, but
there is no equivalent GhostDoc feature)
See demo at
<http://channel9.msdn.com/Showpost.aspx?postid=121822>

Unfortunately still true as far as I can tell.

VB.NET regions are not as flexible as c# regions.
The key here is that you can place a region ~~~inside a function (in C#),
thus breaking up the implementation into logical pieces.

VB is still missing this capability. Also, in VB 6, you could have the IDE
display only the function/property/sub you're working on. This would be a
nice feature to add back to the IDE.

VB.NET does not allow the application the <NonSerialized> attribute to
events (you can in C# by using the Field: modifier).
As a result, there is no simple way of telling the runtime not to

serialize

the event fields.
This results in serializing objects that you didn't expect, resulting in a
larger stream.
If the object handling the events is not Serializable, then the
serialization process will throw an exception.
(See <http://www.codeproject.com/vb/net/serializevbclasses.asp> for more

info

Re: C# vs VB.Net

and workaround)

Status unknown – I haven't used the serialize features of dotNET.

VB.NET has less features with its static code analysis (as in, when you go to Build / Build Solution).

Issues like : functions that don't return values, uninitialized variables, unused variable declarations, etc...

Development time (in c#) is slightly decreased by finding these issues at compile time.

No longer true. Also, the background compiler for VB 2005 is far superior to the one for C# 2005.

VB.NET does not have ... Increments/Decrements. a++;a--;

VB 2005 still doesn't have the increment/decrement operators, however, it does have the more general += and -= operators.

VB.NET does not have operator overloading. C# has Operator Overloading. (see <http://www.csharphelp.com/archives/archive135.html>)

VB 2005 has the same operator overloading abilities as C# 2005.

VB.NET has no multiline comment syntax

VB.NET has no within-the-line comment syntax

VB.NET has no multiline string syntax

All three of the above are still true, and all would be welcome additions..

VB.NET has a limit on line continuations (I think the limit is 10?)

In VB 2005, it's appears to be unlimited. I just tested 30 lines and the documentation says a single line can be 65535 characters long and you can create longer statements via line continuation..

In addition, VB 2005 has the following useful features:

Event Handlers added by the IDE include the clause "Handles <object.event>", making it real easy to find orphaned event handlers.

When inheriting a class with MustOverride or Implements items (abstract in C# terms), the IDE will automatically insert the correct prototype for you.

VB has inherited the VB6 C<type> features such as CInt, CLng, and has also abstracted the System.Convert class in a single CType(object, <class>) statement for a generalized C<type> interface.

The My classes, which can be used in C# via a using statement, are native to VB.

Case insensative code. Foo and foo are the same. The IDE will correct the case for you, but the compiler won't spit out a ton of errors if you forget.

One syntactical item sorely missing is the C# yield statement. It can be worked around, but it should have been part of the language. I'm sure there are other features of C# that VB is missing, but I'm currently unaware of them.

C# still tends to have higher salaries because of the stigma surrounding VB as a "toy" language, which it definitely hasn't been since at least VB 5.

One thing MS has done really well is supporting multi-language solutions in that each project of the solution can be written in a different language and they will seamlessly work together. Other venders have done this, but their solutions didn't command the market share that MS does.

Mike Ober.

.