

## Re: VB6, VB2005, or Something Else?

---

*Source:*

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2006-04/msg01416.html>

---

- *From:* Andre Kaufmann <[andre.kaufmann.bei@xxxxxxxxxxxx](mailto:andre.kaufmann.bei@xxxxxxxxxxxx)>
  - *Date:* Sat, 15 Apr 2006 04:22:40 +0200
- 

Ralph wrote:

>> [...]

I shouldn't have butted in. As you all are having having great fun and I have no desire to go into detail to defend what "might have been".

I just think your argument is totally specious from top to bottom.

I don't think any of the gurus was serious suggesting that changing the COM specification is a prerequisite to accommodate a new VB.,

Not a prerequisite. I'm something lost in this thread now, because my newsreader doesn't show any tree layout but only a flat list, because this thread got way too long now :-9

So let me recall what I'm thinking:

- a) It wouldn't be a good idea to enhance the COM model
- b) So any VB6 successor is free to break compatibility to COM
- c) The current VB6 syntax regarding objects is influenced by COM and IMHO it wouldn't be a good idea to extend it either.
- d) A new object model could support both native and managed

May be that I'm wrong in c). My language background is Basic (C64, Amiga), Assembler, Pascal, C++ and Java. I have written more code in VB.Script than in VB, though I assume the syntax is quite the same (?)

And I had to write some sample code for a C++ COM library in VB6 and VB.NET. And I could deal with the VB.NET syntax better than with the VB6 one.

That doesn't imply that the "old syntax" couldn't be supported and that the new one couldn't support native code compilation.

nor do I think

anyone was suggesting that code written for a managed environment should be 'backward compatible'.

## Re: VB6, VB2005, or Something Else?

It has been said that code could be compiled to both native and managed. And IIRC Delphi has been cited as an example for that. I don't want to rise any language wars here. Delphi.NET is very well done, no doubt.

But I think it's a myth that you simply can take your native code and compile it to managed one and expect it to be the same as if you would re-engineer the whole application and rewrite it in .NET.

Just to use .NET I don't need to recompile my whole application to managed code. This is why I stated VB6 could have done the same and compile all the code which wouldn't make sense or because it won't fit very well in .NET to native one and generate a mixed application.

If I need my code exposed and compiled to managed one I could choose to do so, surely with some restrictions.

While the analogy is imperfect, it is similar to DAO & ADO. They accomplish almost identical tasks – yet the internal details are very different.

Do you think it would be all that difficult to create an ADO.Net for VB? Pop in a managed provider, an adaptor thingy, a couple of controls to accept xml, and bang! – you got ADO.Net a la VB. Is it a requirement that the code would be identical? Did ADO have to 'look' like DAO? Does ADO code have to be backward compatible?

As I already stated this would have been possible if Microsoft would have written some wrappers for the current GUI and database stuff, like Borland has done with Delphi.NET. This would lead to the same discussion like in the Delphi newsgroups. Because there's no need to use it, many Delphi developers don't use .NET. Because code that is not written for .NET is slower and may be much slower if compiled to managed code, they are even more convinced that .NET is "crap".

O.k. many (classic) VB6 developers think the same, but for other reasons.

Currently VB is using OLE services to manage objects.

Don't know what you mean exactly.

It used to use DDE. VB could have easily offered managed objects using managed services. What really changed in a TextBox's behavior when they became based on OLE? (ie, the difference between a TextBox in VB3 and one in VB6) – Oh, oh, I know – VB6 TextBoxes also support DDE. It is called indirection and enhancement – programmers write that stuff all the time.

Or take implementation inheritance. What magic does C++ do here. The compiler simply takes the various class directives, applies some rules, and creates at runtime a single object which combine elements of several classes – do you really believe – and want to tell the world that this couldn't have been done in VB without 're-writing' the COM specification. <?>

Re: VB6, VB2005, or Something Else?

No just the opposite. COM could have been just the same for the native world as .NET (objects) is for the managed one.

But I also stated that this would break compatibility in many ways. You cannot call this enhanced model COM anymore, because one would expect it to be compatible.

Wrapping services and objects in another layer to be compatible is IMHO not the same as using the model natively.

As a sidnote, what is really funny to me, is if VB6, back in '98, had implemented implementation inheritance, but added the same restrictions dotnet imposes,

Which restrictions do you refer to ?

there would have been such a hue and cry – "VB is still not a true OOPL". <G>

May be. I know what you mean ;-)

–ralph

Andre

.