

## Re: Advice Needed...

---

*Source:*

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2007-05/msg01883.html>

---

- *From:* "Robert Morley" <[rmorley@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:rmorley@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Tue, 22 May 2007 11:30:29 -0400
- 

I use API all the time for enhanced functionality in vb6 (eg userdrawn items in a listbox). What about access controls is better?

I don't need userdraw functionality in a listbox, so it's never been a concern for me. As for what's better about them, I suppose it depends on what you're trying to do. Certainly data binding's much easier in Access, though that's to be expected. I think the biggest single thing that I find useful in Access listboxes (though there are probably others if I go looking) is that they can handle multi-column data, where a VB6 listbox can't, AFAIK.

What instability?

The only real issue is the problems when an app shuts down unexpectedly. I'll admit, they could've done a bit better of a job with that in Access.

We get between 2 and 5 corrupt databases sent to us a week out of 500 users. Some can be recovered, some can't. Access basically doesn't work over wireless (most of our users are not on

Ah, yes, that would be a problem. They're almost certainly dropping the connection, even if only momentarily, and Access' biggest single flaw is coming into play (the previously-mentioned issue of an app terminating unexpectedly, or in this case, the connection to it terminating unexpectedly). In this sort of scenario, yeah, you'd want to avoid Access at all costs! :-)

It will ask to upgrade etc. If they upgrade then they can't go back. I know you can get around it but you need to do that.

It does? I've never had a database prompt me to upgrade, unless it was v2.0 or before...which is more than a little old at this point.

## Re: Advice Needed...

With access the end result feels a lot less professional to me than a vb6 exe.

I suspect that comes back to what you're doing with it, and how you're doing it. I've yet to see a VB6 app that I thought felt more professional than an Access app (in fact, most that I've seen come across as dinky little flunky apps...though I blame that entirely on their designers, not VB6 itself). That said, I've seen a few that have come across as equal, and only rarely one or two that I thought felt more professional, and those few were because they were using snazzy outside controls and the like, that one normally wouldn't think of in Access because their predecessors are available and will get the job done...they just don't look/feel quite as nice.

That's not the way many of us see it.

Your analogy would hold more water if vb6 didn't work on vista but it does.

With a few problems, as I understand it. Certainly since the release of Vista, I've seen a lot of posts here talking about problems getting either VB6 itself or a VB6 app running on Vista.

Ah, but in VB.NET, they didn't keep the old one also, did they? They simply trashed the old one and told you to re-write with the new structure.

The old one is still available in vb6.

Which was sort of the point...migrate your apps to our new structure, or continue to use the old one and just ignore the new one...but you can't use your old stuff in our new stuff. (Or get bogged down in COM interop, but that's a whole 'nother set of problems.)

I've never heard of database tables supporting that functionality...though admittedly, most of my experience is with the MS products. That sounds like something that should be the work of an OO language, not a database itself.

It could have huge potential in a database.

It could, yes, but I'm not entirely sure if that would be a good thing or

Re: Advice Needed...

Re: Advice Needed...

not. I suspect that would be a very inefficient design at the low level, and would cause a lot of slowdowns. Besides, when you get right down to it, that's essentially what Views are for anyway. Your table is the "base class", and the views allow you to inherit from it and expand it into multiple different types of objects, each looking and behaving much the same as the original, with the option to add new fields, override existing ones, or hide them at will.

Thanks for that. I at least understand the thought behind it now, though given that C# and VB.NET go through so much else, being managed languages, I wouldn't have thought the performance hit (or size hit, if they'd followed Karl's suggestion in the commentary) of a non-zero array would really be that big of an issue.

I wouldn't think it would be that big a hit but I guess it has to subtract that number for every single array access.

Yeah, I can see it adding up in some cases, but I'm surprised that that would really be that big of an issue.

We should ask Rick Rothstein to do it. He'll manage it in just one line!  
;-)

Would have to be 1 big line of code.

Wouldn't it, though!

Rob

.