

## Re: vba vs VB dotnet

---

*Source:*

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.formscoding/2008-03/msg00826.html>

---

- *From:* Souris <[Souris@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:Souris@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Mon, 17 Mar 2008 05:06:00 -0700
- 

Thanks millions for the information,

"Albert D. Kallal" wrote:

"Souris" <[Souris@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:Souris@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)> wrote in message  
[news:D5A25F3F-CDCC-4F18-9C8C-9B743A2726D2@xxxxxxxxxxxxxxxxxx](mailto:news:D5A25F3F-CDCC-4F18-9C8C-9B743A2726D2@xxxxxxxxxxxxxxxxxx)

I have a new project to use MS access as backend.

I would like to know what adre the difference between VBA and VB dotnet  
as  
front end?

We probably should make a little bit of a distinction and clarification here. When we talk about VB6, or vb.net, your are as a general rule referring to a system that lets you build forms and is an application development environment.

When you refer to something like visual basic for applications, you're talking about the actual programming language used for quite a few different products in the Microsoft's stable (mainly office). In a sense you can not use VBA to develop an application, without having chosen the product you going to use to be developing with (such as ms-access).

The above is a bit of a minor distinction between the two, I suspect for the most part you are simply asking for some pros and cons in developing an application using the VB.net, as opposed to using MS access as the development tool.

By the way if you ask your question in the .net groups, I'm sure you'll get a completely different answer then when you get in this "pro access" group. If you go ask a vb.net developer to make your application, he's not going to spend the next 6-12 months learning access to become really pro efficient, he's going to build that application in vB.net (and the developer will

## Re: vba vs VB dotnet

likely be able to do so in less time it takes to learn access).

So while we can make a comparison between one or the other, often it's what developer skills and people you have available that will force this choice.

The critical thing to keep in mind here is that access is specifically designed to build forms that edit data. You can very well write a PAC man game in the vb.net, but I would not recommend that you write that PAC man game in ms-access. And if you need to write some advanced web services, then once again you're going to be using the.net environment.

The other significant approach that's different in the.net it is a OO (object orientated) development tool. This type of development is superior when you're going to have several developers working on the project at the same time. (in other words this different type of development approach tends to work better when you have more then one person working on the project at the same time).

MS access is built around the concept of a bound data form. This concept means that when you put your fields on the form they are bound to underlining query or table. Since MS access has (forces) a consistent way of navigating through records, and bulding forms to edit records, you actually in many cases get a more consistent looking application when you develop under MS access as opposed to VB.net.

In other words in the.net environment means you have considerably more flexibility in how you build a user interface, but often that increased flexibility is a downside since now you as a developer must adopt a consistent layout from form to form. (And of course more experienced developers will do this by nature).

Another thing to keep in mind is that when you develop under vb.net you going to have to write more code to load up the data for those text boxes you place on the forms. With some of the new data connector options, the vb.net enviornment has improved quite a bit in this regards, but it still a LOT MORE work then ms-access. You'll also missed the integrated report writer in ms-access.

I think much the choice comes down to here is how complex and how difficult of a project you're tackling. For example if you're taking on a project that needs five to ten developers to complete, then I probably would recommend the .net environment. As mentoned the philosophical approach to developing in the.net environment is differnt, and is what we called object oriented. For smaller project this OO doesn't matter, but if you have several developers each creating different parts of the application, then an object oriented approach scales better with more developers.

If you just have a lesser and simple application, then the benefits of this type of programming and development approach does not help you a lot.

Re: vba vs VB dotnet

For example having a welder teaching you how to weld and use high quality hide tensile steel is not going to be much of the help if you construction project is simply to build a dog house. In fact that advanced skill level might not even help you if you're building a garage. However start building a tower, a parkade, etc, then you'll instantly see how you need these advanced skills.

what is the best choise and what I need conside to use front end bwteen 2 choices?

You can only evaluate this On a case by case, or project by project basis.

Furthermore the other issue is what tool do you use now? You really for the most part can't pick a tool out of the air and choose it to use your project unless you have a good deal of experience.

The following list will give you some ideas here as to what I mean:

The skill level of the developer will make or break the the success of project.

I am a better Golf coach when Tiger Woods is on the team.

Hence the #1 consideration is at what level the developer is at. There are certainly more levels then just "trained" or "not trained". Generally there are a "lot" of skill levels, but the following breakdown is sufficient. \*\*

Stage 1 Innocent (never heard of the product)

Stage 2 Aware (Has read an article about X)

Stage 3 Apprentice (has attended a three-day seminar)

Stage 4 Practitioner (ready to use X on a real project)

Stage 5 Journeyman (uses X naturally and automatically in his job)

Stage 6 Master (has internalized X, knows when to break the rules)

Stage 7 Expert (writes books, gives lectures, looks for ways to extend x)

One should NEVER attempt a project with a team consisting with Stage 3 or lower people. This is a sure fire formula for failure.

So perhaps you know one tool already, and you don't have the three to four

Re: vba vs VB dotnet

months, or even more time to learn the tools at hand.

The other significance here is that the .net environment is superior if you plan to build a web portal, or web integrated system for your application (MS access can help you in this regards).

At the end of the day it really is a question of the right hourse for the right course. You don't need a bunch of welding equipment to build a dog house, or even a small garage.

If you just building a bunch of forms that for the most part present and edit data, you'll find an MS access is going to allow you to build that type of application in 1/5th the time it takes in .net.

On the other hand if you building an application that requires geological data integrated with a clickable map that presents the maintainace records of buildings around town, then you will not be building that type of visual interface in ms-access, and you need the .net envirment.

So probably at the end of the day the real deciding factor here is the type of visual interface you are going to need in your application.

The other significant issue is who going to use this application? are you writing an application that's going to be commercially distributed to many people all over the world, or is this simply an application you plan for your own personal use? (or somewhere in between). MS access is weak in terms of building a product for wide distribution and wide use.

--

Albert D. Kallal (Access MVP)  
Edmonton, Alberta Canada  
pleaseNOOSpamKallal@xxxxxxx