

Re: globals?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2007-01/msg02479.html>

- *From:* "Herfried K. Wagner [MVP]" <hurf-spam-me-here@xxxxxx>
 - *Date:* Sat, 27 Jan 2007 13:57:15 +0100
-

"Fabio" <znt.fabio@xxxxxxxxxxxx> schrieb:

There are several reasons why someone might use the VB methods in favor of the framework methods

1. the VB methods are sometimes shorter (saves typing , and thus coding speed)

But you loose much more in readability and in OOP point of view.

OOP is a tool, not a goal. Yes, you are loosing object-orientation, but I believe that object-orientation is not always the best tool. Take the 'Math' class as an example.

```
\\  
x = Sin(a) * 2  
//
```

seems more natural than

```
\\  
x = Math.Sin(a) * 2  
//
```

In the latter case readability gets worse and the only way to make it better is by importing the 'Math' class. In the case of the mathematical functions it's absolutely irrelevant that the functions are members of a certain class. Qualification by class name only blows up code and thus reduces readability.

And as said by Tom, you make your code difficult to understand to non-vb "players".

Does that really matter? There are so many things which are different in various programming languages. You can also say that other programming languages not providing the functionality VB provides make understanding VB code hard to non-vb "players", which is certainly often the case in practice. So that's only a POV.

Re: globals?

2. The VB methods sometimes behave different as there .Net counterparts
- 3 The VB methods often have built in validation (where you should write a try catch block for the .Net counterpart)

I see that an an advantage in certain cases. Why write

```
\\  
If s IsNot Nothing AndAlso s.Length > 0 Then  
...  
End If  
\\
```

if I can simply write

```
\\  
If Len(s) > 0 Then  
...  
End If  
\\
```

—
M S Herfried K. Wagner
M V P <URL:<http://dotnet.mvps.org/>>
V B <URL:<http://dotnet.mvps.org/dotnet/faqs/>>

.