

Re: Get/Set vs Public Variables

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2008-02/msg00540.html>

- *From:* Paul Werkowitz <newsgroups@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 5 Feb 2008 22:14:54 +0100
-

Am Mon, 4 Feb 2008 05:54:37 -0800 (PST) schrieb Jon Skeet [C# MVP]:

On Feb 4, 1:48 pm, JB <jamesb...@xxxxxxxxxx> wrote:

I was always taught in C++ that using public variables was a bad idea, which i could understand, so some generic get/set methods were usually wrote to access the data.

It's exactly the same in C#. I don't really see how you can understand why it's a good thing in C++, but not in C#.

First, more tahn 90% of properties I saw in my life are simple setters and getters without any useful functionality.

Second, chances are really low that a construct like

```
class Person
{
  string name;
  string firstname;
}
```

EVER will need any functionality in getters and setters for the two string fields.

So, why is it a bad idea to expose them directly to clients?

– dirty management for business objects? No. This requires to explicitly call some function (IPropertyChanged) in the setter, only for that purpose. There are better ways.

– Binding to external code? No. When someone ref-binds to a public variable and this is later changed to a property, we must change the statement and recompile. Big deal? IMO, no.

Re: Get/Set vs Public Variables

– Databinding etc? No. It is a conceptual lack of WinForms (and WPF) databinding that it can bind only to properties and not to variables. Technically, it is no problem to include variables in binding. But LINQ alleviates most of that.

Of course, there are situations where this is simply wrong. If I must anticipate the situation that my implementation might change, I definitely am better off with properties. But for > 90% of all classes with data members this is NOT the case. And if I make a mistake here, the cost is low – mostly recompilation, and some syntax errors. Definitely not worth to make everything private.... – at least until auto-properties were invented.

Greetz
Paule