

Re: Letting go of Hungarian Notation

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2004-06/3286.html>

From: One Handed Man \ (OHM – Terry Burns) ("One")

Date: 06/15/04

Date: Tue, 15 Jun 2004 08:36:58 +0100

I agree with you.

However, I am too lazy to prefix identifiers with s, i etc and rely more on appropriately commented and well written code to assist in maintainability and readability.

Therefore I tend not to use prefixes except with controls such as btn, lbl as you do. I do allways use upper case for constants though

```
--
OHM ( Terry Burns )
. . . One-Handed-Man . . .
<cmoyaX@nospam.com> wrote in message
news:umS7fmqUEHA.3540@TK2MSFTNGP11.phx.gbl...
> Just seeking some advice (or solace)...
> Am I the only who's having trouble letting go of notation? Having
extensive
> experience in C++ years ago (both before I jumped into VB3 in college and
at
> various times during my VB career), I welcomed the "richly-typed" .NET
> framework. But, after almost *two years* of "giving it a chance," I still
> insist that using notation for the "common" (mostly value-types) types
> (string, integer, boolean, etc.) is extremely beneficial. I can instantly
> see that I'm dealing with a declared variable in my procedure rather than
a
> property or some framework object. Even using "o" to denote other
> instantiated (reference) objects is beneficial because I can instantly see
> that I'm dealing with an actual instantiated object rather than some
> "Shared" Class.
>
> Here are some of the standards I've adopted after almost two years of
> experience in .NET.
> 1) Use all the old VB standards for intrinsic types (s,i,b,dt, etc.-- yeah
I
> dropped the 3 letter notation for simple types years ago) with some new
> additions for commonly used Framework objects (sb for StringBuilder, for
> instance).
> 2) "o" for all other lesser used framework types and instantiated objects.
> 3) All the old VB notations for the most common controls (lbl,btn,txt,pic,
> etc.). Derived controls and fancy custom controls get noted according to
> their purpose or roots (i.e. "fancy buttons" still get the "btn"
notation).
> 4) "m_" or the more c#-like "_" to denote module-level variables.
> 5) "g_" for global variables.
```

microsoft.public.dotnet.languages.vb: Re: Letting go of Hungarian Notation

> 6) All Caps for global constants.
> 7) I even tried to keep notation out of my procedure parameters... but
> decided I liked those too (although I agree that they're ugly when seen
> through intellisense).
>
> So on and so forth. I should also note that I'm not a "resistor".... I
tried
> very hard to be open-minded. I've enthusiastically accepted a lot of the
> .NET conventions (structured error handling, options strict on, using
> framework objects as opposed to the legacy VB equivalents (where
> appropriate... i.e. Mid/Left/Right still beats String.Substring() anyday
if
> only because of the graceful error handling)).
>
> Any comments, advice? (P.S. I'm not an old fogey (28).... but I have been
> coding since I was 13. Maybe that's the problem!) *Sigh.*
>
>