
Subject: Re: Yet again, The Sky is Falling!

Posted by [Paul Van Delst\[1\]](#) on Mon, 12 Mar 2007 22:30:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

Sven Geier wrote:

> mgalloy@gmail.com wrote:

>

>> I'm talking about two parameters that are both passed a single named

>> variable (like Paul's mypro example). While legal, I don't think this

>> is a good technique for clear code.

>>

>

> ...and as usual there's exceptions where this is a perfectly good thing to

> do. I have a routine in front of me that dynamically improves a "guess" of

> some number. It takes an input and an output parameter and in almost all

> cases you want to give it the same variable there. Schematically like this

>

> x = someOldGuess

> improve,x,x

>

> where "improve" takes the first "x" as its input, copies the values to a

> local variable, performs a bunch of magic and returns the result in

> the "second x". From the outside, the variable "x" simply has a new,

> improved value (which is the purpose of "improve").

But why do that? It makes no difference since you're overwriting the original value of x anyway. Why not just do

```
x = someOldGuess
```

```
improve,x
```

where "improve" just modifies "x" internally as required. The only advantage (that I can see) of

```
improve,x,x
```

over

```
improve, x
```

is that the former serves to confuse the reader of the code. There may be examples of exceptions where the construct in question is a good idea, but this isn't one of them.

> I second David's statement that one of the nice things about IDL is that one

> can do all these weird things. As someone once said (about C++, I

> think) "all the power and all the elegance of a hand grenade": It ain't

> always pretty but it gets things done.

Yes, well, just because something *can* be done....

But I like the analogy - much more graphic than the old "shoot yerself in the foot" sawhorse. :o)

cheers,

paulv

--

Paul van Delst Ride lots.

CIMSS @ NOAA/NCEP/EMC

Eddy Merckx
