

---

Subject: Re: Am I stupid?

Posted by [Paul van Delst](#) on Wed, 11 Jul 2001 14:21:28 GMT

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

---

Jaco van Gorkom wrote:

>  
> Paul van Delst wrote:  
>> Jaco van Gorkom wrote:  
>>> Craig Markwardt wrote:  
>>>> "Liam E. Gumley" <Liam.Gumley@ssec-nospam.wisc.edu> writes:  
>>>> > Colin Rosenthal wrote:  
>>>> > > ... to have a keyword AXISCOLOR in a routine that also accepts  
>>>> > > the AX keyword. ...  
>>>> > ...  
>>>> > "Keywords can be abbreviated to their shortest unique length"  
>>>> >  
>>>> > In your case, the keyword AX precludes the use of another keyword which  
>>>> > begins with the letters "AX". ...  
>>>>  
>>>> ... I want keywords like TIME, TIMEUNIT, TIMESTEP,  
>>>> and so on. My suggestion is that the above policy should hold,  
>>>> \*unless\* there is an exact match to a specific keyword.  
>  
> On second reading, the documentation ("IDL> ? abbreviating keywords") exactly  
> describes Craig's suggestion. The full keyword TIME cannot be abbreviated to a  
> shorter unique length, whereas TIMESTEP can be abbreviated to the shortest  
> unique length TIMES. Nowhere in the documentation can I find the behaviour that  
> the keyword TIME may not be used, simply because... because... well, for no reason  
> at all actually.

Umm, I haven't had my requisite 4 cups of coffee yet this morning so I may be fading in and out here, but if you specify a keyword TIME=something, how is the interpreter supposed to know if you are specifying TIME, TIMEUNIT abbreviated to "TIME", or TIMESTEP abbreviated to "TIME"? My initial fix would be to rename TIME to TIMEVALUE and stick an easy to see comment somewhere in the header warning users about abbreviating these keywords too much.

> So I would consider this to be a bug, not a policy.

A bug... hmmm. In the user-written code maybe for using ambiguous keyword definitions that allows newcomers to IDL (or people that are trying to switch from Matlab :o) to get quite frustrated as to why the code they just downloaded won't work right.

>> It will make code harder to read and understand. Once you give your code to others, they  
>> may use it in other code with abbreviated keywords and not enough comments as to what they  
>> are doing.  
>

> I fully agree, \*if\* you are referring to the general (over-)use of keyword abbreviation.

I guess. I personally never abbreviate keywords in code, but I also have found the 31 character limit on variable names in f90/95 to be limiting on occasion. :o)

If you are concerned about writing robust code that can't be broken easily, making any keyword `_roots_` (like "TIME" in Craig's example) unambiguous would, to me at least, seem like the easiest first step.

>>> So who is frustrated enough to put in a feature  
>>> request with RSI (/VNI?) for this, for the benefit of us all?  
>> I wonder if there is a way for me to write a little Perl script to automatically send in a  
>> Feature Un-Request whenever this Feature Request is emailed to RSI..... :o)  
>  
> :-). `Feature request' was the wrong wording. I should have said `bug report'.  
> Paul, would you take the honours? :-)

My bug report would reflect Craig's point: How come I can compile .pro files with ambiguous keyword `_roots_` ? (that's my phrase for today... :o)

paulv

--

Paul van Delst            A little learning is a dangerous thing;  
CIMSS @ NOAA/NCEP       Drink deep, or taste not the Pierian spring;  
Ph: (301)763-8000 x7274   There shallow draughts intoxicate the brain,  
Fax:(301)763-8545        And drinking largely sobers us again.  
                         Alexander Pope.

---