Subject: Re: IDL 8.0 compile_opt changes
Posted by penteado on Mon, 21 Dec 2009 06:16:48 GMT
View Forum Message <> Reply to Message

On Dec 21, 3:33 am, ddegr...@stny.rr.com wrote:
```
>  No! negative index is a Bad Idea!
>
>  z= where(bad ne 0)
>   if z[0] ne -1 then begin
>      lots of stuff
>   endif
>
>  I have a lot of code like this, which would no longer be valid if -1
>  were a legal array index. I could go thought and include a count in
>  the where function but that's a lot more work than fixing all the ().
>  Actually I was surprised that you could still use () for arrays. I
>  tell my students to always use [].
>
>  david
```

I use that a lot, too. I do not know what the negative indexes will mean in IDL (the same as in Python?), but that does not by itself mean that where() would no longer return -1 for nothing found.

Also, I think that replacing those occurrences by checking where's count is easier than replacing () by [], since they can be found by text searches. In the case of (), simple text searches would find a lot of uses of () for precedence in expressions, and in function calls. And because of the function call ambiguity, it can be tricky when inspecting the code to determine if something is a function or a variable. In the case of -1 returned by where, there is no ambiguity, and if where's results are always test close to it (as is probably the case most often), it is not so difficult to find and replace all occurrences.

Still, if there would be no bad side effects, it would be better if where() will only return -1 meaning something other than not found if it some new keyword is set. That way, old constructs like the one you mentioned would be unaffected.

Somehow, this reminds me what it would also be really nice to have a syntax for accessing substrings with indexes, as in Python, Fortran or C/C++. Having to use strmid() all the time is a bit tiring, specially for writing to a piece of a string, when the pieces have to be put together with +, strjoin or strput.