Subject: Re: static variable mayhem
Posted by wmc on Tue, 12 May 1998 07:00:00 GMT
View Forum Message <> Reply to Message

In article 1@filet.nrlssc.navy.mil, "T Bowers" <tbowers@nrlssc.navy.mil> writes:
> function foofunc, x
> common CBlock, firstTimeInFooFunc ;I think this'll init to 0
>
> if (NOT firstTimeInFooFunc) then begin ;init'd to 0, so I'll just NOT
> the bastard
> firstTimeInFooFunc = 1 ;Set so this'll never happen again on subsequent
> calls
> return, x = x - 1
> endif $
> ....
>
> No such luck. It compiles, but when I single step to this line, I get
> % Variable is undefined: FIRSTTIMEINFOOFUNC (CBLOCK).
> % Execution halted at: FOOFUNC blah, blah, blah

Well I don't think it compiles, since return,x=x-1 is odd.
But if you swap that to return,x-1 then it should compile.

To make it *work*, say
common CBlock...
at the main prompt (or somewhere above foofunc) and then say
first...func=0 (or 1, or whatever you want...)

Then it will do what you want. Ugly, though.

- William

William M Connolley | wmc@bas.ac.uk | http://www.nbs.ac.uk/public/icd/wmc/
Climate Modeller, British Antarctic Survey | Disclaimer: I speak for myself

Subject: Re: static variable mayhem
Posted by Theo Brauers on Tue, 12 May 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Hi:

you may the following:

FUNCTION foo, x
COMMON foo, called
; look if called is defined, what ever value it has!

```
IF n_elements(called) EQ 0 THEN BEGIN
; define called
    called=1
    RETURN, x-1
ENDIF
RETURN, x+1
END
```

I hope it solves your problem.
Theo

T Bowers wrote:
>
> Hi.
> I'm trying to do something very simple, but IDL 5.02 is fighting me. I'm
> trying
> to have what would be a static variable in c/c++ in a function set as a flag
> so
> that it is initialized when i come in, like so:
>
> function foofunc, x
> ;In c/c++ I would put this next line to init my flag the 1st time this fn
> is called
> ; from foopro below.
> ;static int firstTimeInFooFunc = 1; ;Set to 1 ONLY the 1st time foofunc
> called
>
> if (firstTimeInFooFunc) then begin
> firstTimeInFooFunc = 0 ;Set so this'll never happen again on subsequent
> calls
> return, x = x - 1
> endif $
> else return, x = x + 1
> end
>
> pro foopro
> x = foofunc(0)
> print, x
> x = foofunc(x)
> print, x
> x = foofunc(x)
> print, x
> return
> end
>
> the output I want is:
> -1

> 0
> 1
>
> I tried:
> function foofunc, x
> common CBlock, firstTimeInFooFunc = 1
> ...
>
> but IDL won't let me initialize firstTimeInFooFunc like this.
> Then I thought probably IDL would initialize it for me automatically to 0.
> So i just changed the code to:
>
> ...
> function foofunc, x
> common CBlock, firstTimeInFooFunc ;I think this'll init to 0
>
> if (NOT firstTimeInFooFunc) then begin ;init'd to 0, so I'll just NOT
> the bastard
> firstTimeInFooFunc = 1 ;Set so this'll never happen again on subsequent
> calls
> return, x = x - 1
> endif $
> ...
>
> No such luck. It compiles, but when I single step to this line, I get
> % Variable is undefined: FIRSTTIMEINFOOFUNC (CBLOCK).
> % Execution halted at: FOOFUNC blah, blah, blah
>
> I'm *sure* I'm missing something here. Doesn't IDL have static variables?
> Can anybody help, please?
