
Subject: Temporary variables still checked out ...
Posted by [Martin Schultz](#) on Thu, 22 Oct 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

To all,

even after about an hour or so, I still cannot figure out why I
get the error message "% Temporary variables are still checked out -
cleaning up..." with the program attached below. The idea of the program
is to return a free logical unit number both as function result and
parameter so that it can be used immediately as well as later on
(like in OPEN_FILE,name,get_freelun(ilun) & free_lun,ilun).
Interestingly, the result itself is correct, and if you first open a
file (e.g. with openr,1,name), then a subsequent call to get_freelun
yields no error message. I searched the online help but couldn't find
anything.

Thanks for any input,

Martin.

--

Dr. Martin Schultz
Department for Engineering&Applied Sciences, Harvard University
109 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu
Internet-homepage: <http://www-as.harvard.edu/people/staff/mgs/>

; \$Id: get_freelun.pro,v 1.1 1998/10/09 19:53:32 mgs Exp mgs \$
;-----
;+

```

; NAME:
;   GET_FREELUN (function)
;
; PURPOSE:
;   Return next available logical unit number. Unlike
;   the internal GET_LUN procedure, this function is not
;   restricted to unit numbers above 100, and it will
;   detect any blocked unit number.
;
; CATEGORY:
;   I/O tools
;
; CALLING SEQUENCE:
;   lun = GET_FREELUN([LUN])
;
; INPUTS:
;   none
;
; KEYWORD PARAMETERS:
;   none
;
; OUTPUTS:
;   The lowest available logical unit number. This number is
;   also returned in the LUN parameter for later use.
;
; SUBROUTINES:
;
; REQUIREMENTS:
;
; NOTES:
;
; EXAMPLE:
;   openw,get_freelun(lun),filename
;
; MODIFICATION HISTORY:
;   mgs, 17 Sep 1998: VERSION 1.00
;
;-
; Copyright (C) 1998, Martin Schultz, Harvard University
; This software is provided as is without any warranty
; whatsoever. It may be freely used, copied or distributed
; for non-commercial purposes. This copyright notice must be
; kept with any copy of this software. If this software shall
; be used commercially or sold as part of a larger package,
; please contact the author to arrange payment.
; The copyright is granted if this program becomes part of the
; IDL distribution.
; Bugs and comments should be directed to mgs@io.harvard.edu

```

```
; with subject "IDL routine get_freelun"  
;-----
```

```
function get_freelun,lun  
  
    help,/files,output=list  
  
    newlun = 1  
    lun = newlun  
  
    ; at least one file open  
    ; find lowest available unit number  
    if (n_elements(list) gt 1) then begin  
  
        ; maximum allowed number of open files exceeded?  
        if (n_elements(list) gt 99) then $  
            message,'Cannot handle any more open files'  
  
        ; extract numbers and compare to expectation  
        for i=1,n_elements(list)-1 do begin  
            usedlun = fix(strmid(list[i],0,3))  
            if (usedlun gt i) then begin  
                newlun = i  
                lun = newlun  
                return,newlun ; this one's free  
            endif  
        endfor  
        ; next free unit is greater than all used ones  
        newlun = i  
        lun = newlun  
        return,newlun  
  
    endif else begin ; no file opened  
        return,newlun  
    endelse  
  
end
```

File Attachments

1) [get_freelun.pro](#), downloaded 145 times

Subject: Re: Temporary variables still checked out
Posted by [Peter Mason](#) on Wed, 26 May 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

philaldis@geocities.com wrote:

```
<...> Every time I run it, if I hit the okay button,  
> I get the message from IDL 'Temporary variables still checked out', or  
> at least it's virtually like that, after the program exits.  
<...>  
> I don't know what's going on, but is it something to do with the  
> structures and passing by value etc.
```

Phil, you're essentially doing your assignment to a temporary variable when you do something like (structure.member)=val instead of structure.member=val. The brackets tell IDL to evaluate their contents, and this incurs the creation of a temporary variable. The assignment is lost when the temp variable is destroyed, and you get that inscrutable "programmer's revenge" error message in lieu of a warning that something hasn't worked out quite as you might have expected.

```
> ((*info.ptr)).optIndex[i,*] = 0  
=> Instead of this, try (*info.ptr).optIndex[i,*]=0; etc.
```

Things DO get a bit unsettling when (structures of) pointers to structures are involved, but a bit of command line action will normally clarify IDL's workings with brackets soon enough. (Basically, you must restrict your bracketing to just the pointer component(s).) There was quite a bit of discussion about all this some time back (at least a year) on the NG. As I recall, Stein Vidar pretty much sussed out how it all works - it may be worth your while searching a news archive for this thread.

Peter Mason

```
==== Sent via Deja.com http://www.deja.com/ ====  
---Share what you know. Learn what you don't.---
```