

---

Subject: Re: IDL- variable variable names  
Posted by [meron](#) on Sun, 10 Nov 1996 08:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

In article <davidf-ya023080000711961510130001@news.frii.com>, davidf@fortnet.org (David Fanning) writes:

> I wrote in a previous article:

>

>> But..., I got about this close:

>>

>> FUNCTION MAGIC, file

>> number = 0

>> ok = EXECUTE(file + ' = 0')

>> RESTORE, file

>> RETURN, number

>> END

>>

>> IDL> a = 10

>> IDL> number = a

>> IDL> Save, a, number, Filename='a'

>> IDL> b = 'a'

>> IDL> c = Magic(b)

>> IDL> IF c EQ 10 THEN Print, "It's magic!"

>>

>> It's magic!

>

> Here is an obvious modification that I overlooked that nudges it  
> even closer. And I added some help for the 8.3 crowd! :-)

>

> FUNCTION MAGIC, file

> number = 0

> RESTORE, file + '.sav'

> RETURN, number

> END

>

> IDL> a = 10

> IDL> number = a

> IDL> Save, number, Filename='a.sav'

> IDL> b = 'a'

> IDL> c = Magic(b)

> IDL> IF c EQ 10 THEN Print, "It's magic!"

>

> It's magic!

>

> This almost borders on the useful now. What do you have  
> in mind for this, Hendrik?

>

Oh, you can get useful stuff using EXECUTE, and there is no need to go

through saving and restoring. Here is an example

Function One\_of, v\_0, v\_1, v\_2, v\_3, v\_4, v\_5, v\_6, v\_7, value = val

```
;+
; NAME:
; ONE_OF
; PURPOSE:
; Called with up to 8 variables V_0 through V_7 , ONE_OF checks which
; variable is defined (only one is supposed to be).
; CATEGORY:
; Programming.
; CALLING SEQUENCE:
; Result = ONE_OF( V_0 [,V_1, ... V_7] [, VALUE = VAL])
; INPUTS:
;   V_0 through V_7
; Arbitrary, all are optional.
; OPTIONAL INPUT PARAMETERS:
; See above.
; KEYWORD PARAMETERS:
;   VALUE
; Optional output, see below.
; OUTPUTS:
; Returns the serial number (range 0 through 7) of the defined variable,
; or -1 if none is defined. If more than one variable is defined, ONE_OF
; issues an error message and returns to the main level.
; OPTIONAL OUTPUT PARAMETERS:
;   VALUE
; The name of the variable to receive the value of the single defined
; variable, or a null string if none is defined.
; COMMON BLOCKS:
; None.
; SIDE EFFECTS:
; None.
; RESTRICTIONS:
; Currently ONE_OF is restricted to a maximum of 8 variables. If needed,
; the number can be increased.
; PROCEDURE:
; Straightforward.
; MODIFICATION HISTORY:
; Created 15-JUL-1991 by Mati Meron.
; Modified 30-JUL-1991 by Mati Meron. The dependence of the original
; code on the EXECUTE system routine has been eliminated in order to
; assure compatibility with the OUTPUT routine.
; Modified 15-NOV-1993 by Mati Meron. Since IDL now allows for recursive
; calls to EXECUTE, the original code has been restored.
;-
```

```
on_error, 1
vnams = ['v_0','v_1','v_2','v_3','v_4','v_5','v_6','v_7']
exlist = lonarr(8)
exind = -1
val = "

for i = 0, n_params() - 1 do idum = $
execute('exlist(i) = n_elements(' + vnams(i) + ')')
  wex = where(exlist gt 0, nex)
  if nex eq 1 then begin
exind = wex(0)
idum = execute('val = ' + vnams(exind))
  endif else if nex gt 1 then message, 'Only one variable may be defined!'
  return, exind
end
```

Mati Meron | "When you argue with a fool,  
meron@cars.uchicago.edu | chances are he is doing just the same"

---