Subject: Re: Arrays in structures; workarounds?
Posted by Ed Santiago on Thu, 04 May 2000 07:00:00 GMT
View Forum Message <> Reply to Message

> Does anybody know a way to work around this? [...] The
>only thing I can come up with is to parse the result of HELP,
>OUTPUT=out, but that seems like the crappiest solution ever.

Yep. Crappy indeed... but I couldn't find an alternative, either.

Below is a copy of "esmsize", a function I wrote last year when I
absolutely needed to obtain true dimensions within a struct.

Hope it helps,
^E

```
;+
; NAME:
;     ESMSIZE
;
; IDENT:
;     $Id: esmsize.pro,v 1.3 2000/01/31 14:38:23 esm Exp $
;
; PURPOSE:
;     Front end to SIZE, which will preserve unary dimensions
;
; AUTHOR:
;     Ed Santiago
;
; CALLING SEQUENCE:
;     xx = esmsize( struct, index )
;
; INPUTS:
;     struct      IDL structure
;     index       string or integer index into structure
;
; OUTPUTS:
;     IDL SIZE thingy
;
; REASON FOR THIS BULLSHIT:
;     IDL collapses unary dimensions wherever it can.  For example:
;
;       Pepe> trashme = { foo:reform(indgen(10),10,1) }
;       Pepe> help,trashme,/st
;       ** Structure <81cab34>, 1 tags, length=20, refs=1:
;          FOO             INT       Array[10, 1]
;       Pepe> print,size(trashme.foo)
```

```
;               1         10         2         10
;
;     See?  There is simply no fscking way to get IDL to recognize that
;     last dimension, even though the HELP command sees it.  Therefore,
;     this code was written to parse the HELP output.  Barf city.
;
;       Pepe> print,esmsize(trashme,'foo')
;                 2         10         1         2         10
;-
FUNCTION esmsize, struct, index_orig

  On_Error, 2

  ; We'll never get here, since the StRegeg()'s will not compile
  IF !Version.Release LT 5.3 THEN MESSAGE, myname() + ' requires IDL 5.3'

  ; Check the input args.  First arg must be a structure, and second
  ; must be an index.  If it's a string, convert to integer.
  IF size(struct, /TName) NE 'STRUCT' THEN MESSAGE, 'Arg 1 must be struct'

  CASE size(index_orig, /TName) OF
    'STRING': index = (where(Tag_Names(struct) EQ StrUpCase(index_orig)))[0]
    ELSE:       index = index_orig
  ENDCASE


  ; Obtain IDL's interpretation of the size...
  ss = size(struct.(index))

  ; ...as well as the HELP command's version.  Find the corresponding line.
  Help, struct, /Struct, out=foo
  foo = foo[index+1]

  ; If this structure element is an array, obtain the dimensions
  array_string = 'Array['
  pos = StrPos(foo, array_string)
  IF pos NE -1 THEN BEGIN
    pos = pos + StrLen(array_string)

    undefine, esmdims

    ; Keep looking for digits, and add them to our own "esmdims".
    WHILE StRegex(StrMid(foo,pos), '[0-9]+', /Bool) NE 0 DO BEGIN
      num = StRegex(StrMid(foo,pos), '[0-9]+', Len=len)

      tmp = long(StrMid(foo,pos+num,len))
      IF N_Elements(esmdims) EQ 0 THEN esmdims=tmp ELSE esmdims=[esmdims,tmp]
      pos = pos + num + len
```

```
    ENDWHILE

    ; If the dimensions don't match, override with our own.
    ndims = N_Elements(esmdims)
    IF ss[0] NE ndims THEN ss = [ndims, esmdims, ss[ss[0]+1:*]]
  ENDIF

  RETURN, ss
END
```

--
Eduardo Santiago Software Type esm@lanl.gov RKBA!