Subject: Re: size of a structure
Posted by Peter Mason on Wed, 29 Jan 1997 08:00:00 GMT
View Forum Message <> Reply to Message

On Mon, 27 Jan 1997, Phil Williams wrote:
> 1) What is the "length" of a struture in the help,/st mean? It's not
> the "size" which is what I thought it was. i.e.
> IDL> help,/st,info
< ... structure with several string fields & some other fields, with
< help,/struct showing length=124 >
< ... code showing that structure written to disk is 162 bytes long >

It seems that the structure's "length" is correct except when it comes to structure members which are strings. A string member contributes 16 bytes to the structure's "length", regardless of string length.
This is probably because strings are dynamic, while everything else about an IDL structure's size (#members, member datatypes and dimensions) is static: the 16 bytes shown for a string member is probably some sort of string descriptor with a pointer to the actual string.

> 2) Is there a way, besides brute force, to determine the size of a
> structure w/in a procedure?

Not that I know.
But brute force is often OK, especially for a computer (which is by nature a brute :)
e.g.,

```
; Return the size of variable V in bytes
forward_function sizeof     ;make sure our recursion will work
function sizeof,v
j=size(v) &t=j(j(0)+1) &n=j(j(0)+2) &j=0
case t of
  0:rv=0L
  1:rv=1L
  2:rv=2L
  3:rv=4L
  4:rv=4L
  5:rv=8L
  6:rv=8L
  7:begin       ;string
    rv=0L &for i=0L,n-1L do rv=rv+strlen(v(i)) &n=1L
   end
  8:begin       ;structure
     rv=0L &nt=n_tags(v(0))-1L
     for ii=0L,n-1L do begin  ;loop here because of string members
       vv=v(ii)
```

```
        for i=0,nt do rv=rv+sizeof(vv.(i)) ;loop thru members
      endfor
      n=1L
    end
  9:rv=16L
  else:rv=0L
endcase
return,rv*n
end
```

Peter Mason
