

---

Subject: Re: Fragmented memory with IDL  
Posted by [thompson](#) on Wed, 19 May 1993 00:47:39 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

mayor@vaxine.larc.nasa.gov writes:

> IDLers,

> The following is a question that I sent to IDL tech support today.

> Below it is the reply from IDL. Thought I'd post it to see if

> anybody else had similar problems or solutions.

>> We've noticed that occasionally we get the following error message:

>>

>> % Unable to allocate memory: to make array.

>> not enough core

>>

>> The interesting part is that it never occurs when we run the program

>> immediately after starting IDL. It does happen after running the

>> program multiple times EVEN with a RETALL at the end of each run.

>> Somehow, it appears that consecutive runs cause it to reserve

>> more and more memory - even though each run shouldn't require any

>> more than the previous runs in the same IDL session. Is there a

>> command that I can use to assure that all memory is cleared out

>> even after a retall?

> Alan Youngblood of RSI replies:

> The circumstances you describe happen when memory becomes fragmented.

> Unfortunately, there is nothing you can do except use less memory in

> your application, or get more for the system to work with.

Yes, this is a well known problem. Memory gets fragmented just like disks do.

Suppose you allocate two megabytes for a temporary 1024x1024 integer array. Then you release this memory and create several smaller arrays. Chances are these new arrays will fall into some of the space used by the original large array. Now you want to allocate another 2 megabytes temporarily. You can't use the memory you used before because part of it is taken up. Worse, an array has to be stored contiguously, so any free memory that occurs between used sections can't be used to store this temporary array.

There was a long discussion of memory fragmentation in this newsgroup some time ago. It's probably worth putting some of the net wisdom on this issue in the new FAQ.

One particularly common cause of memory fragmentation is when large arrays are allowed to grow by concatenation. For instance, suppose you were reading a

large data file into a array with something like the following:

```
DATA = FLTARR(10)
READF,UNIT,DATA
ACCUMULATED = DATA
WHILE NOT EOF(UNIT) DO BEGIN
  READF,UNIT,DATA
  ACCUMULATED = [ACCUMULATED,DATA]
ENDWHILE
```

then every step through this loop would require that a new array be created. A better way to accomplish the same thing would be

```
ACCUMULATED = FLTARR(1000)
DATA = FLTARR(10)
N_DATA = 0
WHILE NOT EOF(UNIT) DO BEGIN
  IF N_DATA EQ N_ELEMENTS(ACCUMULATED) THEN $
    ACCUMULATED = [ACCUMULATED,FLTARR(1000)]
  READF,UNIT,DATA
  ACCUMULATED(N_DATA) = DATA
  N_DATA = N_DATA + 10
ENDWHILE
ACCUMULATED = ACCUMULATED(0:N_DATA)
```

Letting the array grow by fewer but bigger jumps will help minimize the effects of memory fragmentation.

Another way to avoid memory fragmentation is to use the TEMPORARY function where appropriate.

Hope this helps. I'm sure other people also have insights into memory fragmentation to share with the net.

Bill Thompson.

---