Subject: Re: saving variables between calls to a procedure?
Posted by Mark Hadfield on Wed, 31 Jul 2002 23:12:46 GMT

View Forum Message <> Reply to Message

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.17b21214ff7d9b35989944@news.frii.com...

```
>  Paul van Delst (paul.vandelst@noaa.gov) writes:
>
>>>  pro define,ptr
>>>   *ptr=[*ptr,10]
>>>  end
>>
>>  Hmm. That seems like an extremely dangerous thing to do - couldn't
>>  you clobber something by concatenating like that? If IDL is smart
>>  enough to recognise that the next bit of memory may be used by
>>  something else it then seems that you would end up with a
>>  non-contiguous data structure (in the figurative).
>
>  This doesn't seem dangerous to me (perhaps because I use the
>  construct all the time). It seems like one of those wonderful things
>  IDL occasionally does that makes you think to yourself "Now, by God,
>  that's how software *ought* to work!"
>
>  In any case, it works, over and over and over. And it never occurred
>  to me that non-contiguous data storage could be involved, even
>  remotely.
```

Extending an array a with the a = [a,b] syntax doesn't create a non-contiguous data structure. What is does is create a new array a, insert the elements from the existing a and b into it, then delete the old a. This is fine for small arrays but it slows down on large arrays because of all the memory allocation & deallocation. It is *very* bad practice to create a large array by growing it one element at a time.

What do I mean by "large" in this context. I don't know, a few thousand I guess. Here is an exercise for the reader: time the following code for various values of n:

```
  a = [0]
  for i=1,n-1 do a = [a,0]
```

--
Mark Hadfield   "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)