Subject: Re: Creating pointer in structure
Posted by Pavel A. Romashkin on Thu, 01 Nov 2001 16:56:11 GMT
View Forum Message <> Reply to Message

I use both ways, but I prefer /ALLOCATE. The /ALLOCATE is very
convenient when you know your pointer must be defined, and it saves time
when the pointer is filled later. I sometimes do not allocate heap if
the pointer may not be required (like a UVALUE property of an object,
for example). I rarely check on validity of pointers because I make my
code issue an error earlier, at the INIT stage, if a vital pointer is
not filled with necessary data, so Paul's remark does not really apply
in my case. For writing object methods, having valid pointers makes it a
lot easier, especially if a field is changeable by several methods.

Cheers,
Pavel

"K. Bowman" wrote:

```
>
>  If I need to define a structure containing a pointer before I know the
>  characteristics of the associated heap variable, which of the following
>  is preferable?  Does it make any difference, or is it simply a matter
>  of programming taste?
>
>  For example:
>
>  a = {point: PTR_NEW()}              ;Create struct w/ null pointer
>   ... figure out what n is
>  a.point = PTR_NEW(FINDGEN(n))       ;Replace null pointer
>
>  or
>
>  b = {point: PTR_NEW(/ALLOCATE_HEAP)}   ;Create struct w/ pointer->undef
>   ... figure out what n is
>  *b.point = FINDGEN(n)                  ;Define heap var
>
>  Thanks, Ken
```