Subject: When Ptr_New doesn't work
Posted by badenes on Tue, 22 Jan 2002 11:42:25 GMT
View Forum Message <> Reply to Message

I have the following problem:

```
rCoefficients=PtrArr(nElems)
FOR i=0, nElems-1 DO BEGIN
  rCoefficients[i]=Ptr_New(PtrArr(elems[i]))
ENDFOR
```

since elems is a vector of integers, each element i of
rCoefficients is a pointer to an array of elems[i] pointers.
But, for some reason,

```
((*rCoefficients[j])[k]) = Ptr_New(FltArr(2), /NO_COPY)
```

doesn't work. k and j are within the allowed range, of course, and
((*rCoefficients[j])[k]) is a null pointer, as expected from the
initialization above. Ptr_New is supposed to allocate memory for the
specified pointer to store an array of 2 floats, but I get the message

Expression must be named variable in this context: <POINTER
(<NullPointer>)>.

I must be doing something wrong. Can you help?

Thanks,
Carles

Subject: Re: When Ptr_New doesn't work
Posted by Pavel A. Romashkin on Wed, 23 Jan 2002 18:07:14 GMT
View Forum Message <> Reply to Message

I think the key here is, anything in parenthesis gets *evaluated* and as
such is an expression. And it is not really surprising that one is not
allowed to assign a value to an expression. This is the strength of IDL
- it creates temporary variables as expressions, and allows to use them
just like named variables. Unlike, say, IGOR Pro that requres you to
define a named variable to even plot it.

Pavel

Richard Younger wrote:
>
> This seems very strange to me. I think of parenthesis as not modifying
> their arguments at all, since they should (in my own little idl, er,

> ideal world) change only precedence and not value.

Subject: Re: When Ptr_New doesn't work
Posted by John-David T. Smith on Wed, 23 Jan 2002 19:13:12 GMT
View Forum Message <> Reply to Message

"Pavel A. Romashkin" wrote:
>
> I think the key here is, anything in parenthesis gets *evaluated* and as
> such is an expression. And it is not really surprising that one is not
> allowed to assign a value to an expression. This is the strength of IDL
> - it creates temporary variables as expressions, and allows to use them
> just like named variables. Unlike, say, IGOR Pro that requres you to
> define a named variable to even plot it.
>

Right. Another example of parentheses forming an expression is:

IDL> a=(b=1)

which is the only way to do chained assignments.

But as for the necessity of this, compare Perl, where things like:

($first, $second, @rest)=make_list();

are allowed. But heh, it's IDL.

JD
