Subject: getting data from a pointer
Posted by Vap User on Tue, 21 Jul 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Here's a question for the assembled cognoscente.

When one creates a pointer one can escape the overhead of creating a copy of the data by using the '/NO_COPY' keyword, e.g. if A = some_really_Big_ARRAY then B = Ptr_New(A,/no_copy) will disattach the data portion of A and attach it to B, thereby making A undefined, but avoiding the wasteful copy of the data that was attached to A.

The question is: Is there a similar method to disattach the data from a pointer and return it to a 'normal' IDL variable? Say one has done the above, and now one has the variable B which points to the data once contained in A, and one wants to reattach it to A but without the 'copy'. I hope everyone would agree that

```
A = *B
Ptr_Free,B
```

will result in a copy of B's data being made in the first statement before Ptr_Free is called to free the memory still retained by the pointer B.

I thought that perhaps

```
A=temporary( *B )
Ptr_Free, B
```

will work, and want confirmation of this conjection.

If this method doesn't work, perhaps we could ask for this functionality as an improvement?

Wadya think, Dave? (with the pipline to RSI)

I realize that there are many situation where I could do all necessary operations on the pointer without every putting the data back into a 'normal' IDL variable, thereby avoiding the problem. But humor me and answer this question.

William Daffer

--
I don't speak for JPL, it doesn't speak for me.
Well, not all the time, at least.
William Daffer <vapuser@haifung.jpl.nasa.gov>
