Subject: Re: changing a datatype in a structure
Posted by Mark Hadfield on Thu, 11 May 2006 01:30:24 GMT
View Forum Message <> Reply to Message

mark.macauda@gmail.com wrote:
> Ok, here is my current very frustrating problem. I have ALOT of IDL
> work done the general idea is that I read in binary files that are
> stored in very specific format...blah blah, the details aren't
> important, the important part is that rewriting all of that becuase of
> the current problem is not a desirable solution. Here's the problem.
> All the info is read in in structures and the data types are dictated
> by the binary files. One is an array of long 32-bit integers. I have
> discovered that I need to do some calculations on this array, that
> require the data type to be something other than interger. Normally I
> would just use the float command and be on my merry way...but for all
> the other programs to work I need the float arrays to be back in the
> handy dandy structures...is there anyway I can easily change that
> datatype?

Changing the data type or shape of a field in a structure is impossible,
as is adding a new field. The only way to make it look like you've done
so to create a new structure and copy the data from the old one.

Your problem sounds like an obvious application for pointers:

```
IDL> x = {a: ptr_new(indgen(10))}
IDL> help, *x.a
<PtrHeapVar24115>
        INT      = Array[10]
IDL> *x.a = float(*x.a)
IDL> help, *x.a
<PtrHeapVar24115>
        FLOAT    = Array[10]
IDL> ptr_free, x.a
```

Note that the integer array and the float array that replaces it are
both associated with the same pointer heap variable. When you've
finished with the structure you will have to clean up the pointer
variable, but you don't have to worry about that there's still a copy of
the integer array lurking around somewhere.

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