
Subject: Re: [Q] structure definition with variable array size
Posted by [Matthew J. Sheats](#) on Thu, 25 Jun 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

> I wish to create a structure for holding a slice from a 3d volume along
> with domain info, e.g.,
>
> pro slice__define
> tmp = {slice, data:fltarr(100,100), x:fltarr(100), y:fltarr(100), z:0.0}
end
>
> This definition is fine if the data dimensions are known. However, what
> if the array sizes are only known at runtime when the data file is opened
> and brought into IDL. Is it possible to have structures with arbitrary
> length arrays ala F90, C, or Pascal?
>
> Cheers,
> Darran.

Unfortunately, the only way I have found to defeat this limitation is to use pointers. For example:

```
tmp = { slice, data:PTR_NEW(0), x:PTR_NEW(0), ... } etc for all dynamic arrays.
```

Then in run time...

```
data = PTR_NEW(FLTARR(100,100))
```

Now that just actually allocates the pointer, you haven't actually allocated the array yet, you need one more call like this:

```
(*data) = FLTARR(100,100)
```

Now you have a dynamically set array in a structure. Bit of a pain, but it works. And it works great with IDL's OOP. Just remember you have to use the pointer dereference whenever you access the array:

```
(*data)[10,15] = 5.0
```

or whatever.

Hope this help,

Matthew Sheats
Los Alamos National Laboratory
