
Subject: Re: How to read file to fill an array "partially" ?
Posted by [ben.bighair](#) on Thu, 18 Oct 2007 00:13:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Oct 17, 2:26 pm, mystea <idllear...@gmail.com> wrote:

> Hi Ben,
>
> Thanks a lot for providing the link and your code.
> I have two observations when I tried out David's ASCII reading tips.
>
> 1. the format code '(I0)' means integer of any length while '(A0)'
> literally means string of length zero.
> 2. A string can't start with a number. for example, if you type
> junk="3", IDL returns syntax error. The best one can do seems to be
> junk=" 3".
>
> Although I can swallow it and take them as simple facts, I don't feel
> very comfortable about it. Why is there such inconsistency in the
> design of format codes? And what can you do if you really want your
> string starts with a number?
>
> As to your bucket code, I need a little more time to digest. Do you
> recommend any reference in IDL object programming?
>
> Sincerely,
>
> Gene

Hi Gene,

The best place to research is this newsgroup! You did search here first, right? I tried searching this group via groups.google for "objects" and this <http://tinyurl.com/3bvg7d> looks promising to me.

If you plan on writing a lot of IDL code then I can assure you that your time will be well spent learning IDL's OO stuff. Having subsequently worked with other OO languages I can only now begin to see the strengths and weaknesses of IDL's OO implementation. I would say the 95% of my IDL code is now OO. I especially like it for working with complex files (headers as well as data), widgets (I can't even remember how to do it without objects), and command line interaction (which is paradoxical in a way.)

What jumped out to me about your problem was that you indicated your data lumps might be different sizes. To keep all those lumps associated together you just have to use IDL heap variable pointers - other wise it is a mess. Now I can never remember when to use *data vs

*(data) vs ... So, long ago I simply squished all that stuff into an object and wrote something like the BUCKET object with GET and SET methods. Write once - use over and over and over again. That is pretty darned handy.

MyBigBucket is simply to contain all the little buckets. Maybe I should have called it "dumptruck" or or "wheelbarrow" or something like that. You don't have to use that as you could use an object array instead.

So you should definitely wade into OO. You won't regret it.

Cheers,
Ben
