
Subject: Re: Is there an easy way to write and read (large) structure?

Posted by [Chris\[6\]](#) on Fri, 14 Nov 2008 07:52:46 GMT

[View Forum Message](#) <> [Reply to Message](#)

You can use the WIDTH keyword in openw to override IDL's default of 80 columns per line.

You can also save / load binary copies of the structure using the SAVE/RESTORE procedures. If you aren't doing anything else with the ascii files, this is more efficient in both space and read/write time. I tried this out by writing findgen(5000,5000) to file - using SAVE was 5 times faster and resulted in a file that was 4 times smaller.

Chris

Subject: Re: Is there an easy way to write and read (large) structure?

Posted by [Vince Hradil](#) on Fri, 14 Nov 2008 14:54:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Nov 13, 11:30 pm, mystea <idllear...@gmail.com> wrote:

```
> Hi all,
>
> I am trying to write my data (which is a structure array) into a ascii
> file and to retrieve it later on.
>
> A most naive way to write to a file is as follows:
> .***
> ;
> openw,lun,'mydata.dat',/get_lun
> printf,lun,data
> free_lun,lun
> .***
> ;
> To read it, simply create a structure array of the same type:
> .*****
> ;
> record={properties,DOB:0.0,M:0.0,sec:0.0,xpos:0.0,ypos:0.0,z pos:0.0,$
> P:0.0,A:0.0,Q:0.0,Tc:0.0,Tm:0.0,clA:0.0,ms:0.0,status:'unbor n'}
>
> data=replicate(record,10000)
>
> openr,lun,'mydata.dat',/get_lun
> readf,lun,data
> free_lun,lun
> .*****
> ;
>
> This method actually works very well for short structures where each
> array element
> can be recorded in one line.
```

```

>
> However, the data structure I have in hand is very long, and the file
> 'mydata.dat' looks
> like:
>
> { 110.0015 1.09355 7.99650 2489.13
> 1962.11 1361.55
> 0.00000 12.00000 53.00000 0.00000
> 0.00000 0.00000
> 0.00000 unborn}{ 0.114715 0.803090 17.3017
> 660.172
> 3556.35 1409.37 0.00000 0.05000 0.00000
> 0.10000
> 0.00000 0.00000 0.00000 unborn}.....
>
> (10000 structure instances, 23000+ lines)
>
> so when I type in:
> readf,lun,data
>
> IDL returns
> % READF: Input conversion error. Unit: 100, File: mydata.dat
> % Execution halted at: $MAIN$
>
> .*****
> ,
>
> Is there any good way to get around this?

```

Does it have to be an ascii file? Why not just use WRITEU?
