

---

Subject: reading in binary data

Posted by [Lazzar](#) on Wed, 15 Sep 1999 07:00:00 GMT

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

---

I have problem that maybe someone out there can help with,

I'm trying to read in a binary data file that is broken down into various tuples (these are blocks of different data within the file). Each tuple starts with a specific header byte to identify it, which is proceeded by a byte that can be used to check if the data in that tuple is the correct type. The way I currently read in these files is to read in an entire file then using a WHILE loop check each byte for a given tuple type, if yes then read in the tuple and store it, if no check for the next tuple type. On the surface this seems to work with no problems, but it is very slow and limited based on file size. Some of the files I'm trying to read in create arrays that are 1000x1500 and that is a relatively small size. I would like to find a faster way to read this data, as well as find a way not to bog IDL down with huge arrays. Is it possible to access a binary file without reading it all in at once? Also is it possible to write to an IDL save file incrementally, so that I can unload some of the array to disk and free up the extra memory? Along the same line is it possible to read in only a portion of an IDL save file if, for example, I only want certain variables from it but not all of the variables?

Another issue of note is the size of the IDL save files. When I convert one of my binary files to an IDL save file it increases the size of that file by about 3 times (a 953k binary file equals an equivalent IDL save file of 3.02meg). Is there any way to reduce the size of the IDL save file (I already remove any zeros from the array by indexing it and store the index and values in separate variables), or to save in a different format that is better on compression but still is quick to read?

Thanks for any help you can provide,  
Brian

---

---

Subject: Re: reading in binary data

Posted by [Lazzar](#) on Thu, 16 Sep 1999 07:00:00 GMT

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

---

David Fanning wrote:

- > This surprises me. There is obviously some overhead in
- > creating an IDL save file, but this seems a bit excessive.
- > Usually in these situations I always blame the programmer.
- > I'm wrong about 0.05 percent of the time. :-)
- >

> And, anyway, why are you so enamored with IDL SAVE files?  
> I use them occasionally, of course, but I'm scratching my  
> head to think of the need for them here. What exactly are  
> you trying to do?  
>  
> Cheers,  
>  
> David  
>

It's not so much that I'm enamored with IDL SAVE files, but I'm looking for speed when opening a file. It takes IDL around 2-3 minutes to open the raw binary file and put the data into the appropriate arrays, but only 30 sec. or so for it to open the IDL SAVE file. I've been in contact with RSI tech support and they've indicated that a new 5.3 feature of having a binary file template might be able to help me with this problem. I'm dealing with hydroacoustic data (like a fish finder) that has a vertical resolution of 53 samples per meter of water and it collects upwards of 5 soundings of data per second. The files can get very big very quickly and speed is very important when process this data. Right now I have a program that reads in the data and converts it from an binary file to an IDL SAVE file so that future processing is faster. IDL is the perfect language to do my data processing in, as I can deal with my data as an image and do whole array processing in one step, but it isn't the best at reading in the data. Also as a non programmer I've been able to learn enough IDL to get by, but seem to be pushing my limits (not necessary a bad thing).

Brian

---

Subject: Re: reading in binary data  
Posted by [kobayash](#) on Tue, 21 Sep 1999 07:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Wed, 15 Sep 1999 10:47:24 -0500, Brian Nagy <lazzar@gte.net> wrote:

> is the correct type. The way I currently read in these files is to read  
> in an entire file then using a WHILE loop check each byte for a given  
> tuple type, if yes then read in the tuple and store it, if no check for  
> the next tuple type. On the surface this seems to work with no  
> problems, but it is very slow and limited based on file size. Some of  
> the files I'm trying to read in create arrays that are 1000x1500 and  
> that is a relatively small size. I would like to find a faster way to  
> read this data, as well as find a way not to bog IDL down with hugh  
> arrays. Is it possible to access a binary file without reading it all  
> in at once?

Are you sure the file I/O is the bottleneck? I recently tried

something similar - reading an entire file (100 KB or so) and looping through the data one byte at a time to look for the beginning/end of each data set. I found that it takes a couple of seconds to read the entire file, and many minutes for the loop. A 'where' command to identify the delimiters was of course much faster.

As someone else mentioned, you can use associated variables (the ASSOC command) to read parts of the file, if you know where each data segment starts without reading the whole file.

- > Also is it possible to write to an IDL save file
- > incrementally, so that I can unload some of the array to disk and free
- > up the extra memory? Along the same line is it possible to read in only
- > a portion of an IDL save file if, for example, I only want certain
- > variables from it but not all of the variables?

Can't you just write a plain binary file incrementally?

- Ken

---