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