
Subject: Re: reading unformatted data into a structure
Posted by [Nigel Wade](#) on Tue, 19 Mar 2002 09:42:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Rick Towler wrote:

```
> I have an unformatted data file which I am trying to read and I am having
> only limited success. The data file was created on a Sun machine and it
> is being read on a win32 machine (I have set the swap_if_little_endian
> keyword
> on the open statement). As for the file format, all I have to go on is
> the C header file that describes the records in the file.
>
> Early on in the header I find this:
>
> // Variable types:
> //  int  is 32 bit integer.
> //  short is 16 bit integer.
>
>
> Here are the first few lines of the header (all of the data types found in
> the header are in these lines):
>
> typedef struct
> {
>   int fileFormatVersion; /* The file format version number. */
>   int points;           /* Number of xyz points on the file. */
>   int time;             /* UNIX time of file generation,
>                          seconds since 1/1 1970. */
>   unsigned short coordSys; /* The coordinate system used:
>   0 = Geographical lat/long.
>   1 = Projection */
>   char projection[32]; /* Projection name (see above) */
>   int projParam1;      /* 1. projection parameter (see above). */
>   int projParam2;      /* 2. projection parameter (see above). */
>   int projParam3;      /* 3. projection parameter (see above). */
>   int projParam4;      /* 4. projection parameter (see above). */
>   int projParam5;      /* 5. projection parameter (see above). */
>   char datum[32];      /* Name of the datum (see above). */
>   char ellLargeHalfAxis[16]; /* Large half axis of the ellipsoide. */
>   ....
>
>
> From the comments about variable types and the structure definition I
> created an IDL structure that I think matches the C struct (again, I'll
> list the first few lines):
>
> header = { fileFormatVersion:0L, $
```

```

>     points:0L, $
>     time:0L, $
>     coordSys:0, $
>     projection:bytarr(32), $
>     projParam1:0L, $
>     projParam2:0L, $
>     projParam3:0L, $
>     projParam4:0L, $
>     projParam5:0L, $
>     datum:bytarr(32), $
>     ellLargeHalfAxis:bytarr(16), $
>     ....
>
>
> I open the data file and read the header like so:
>
>     openr, lun, 'J:\hydrographic\1.xyz', /get_lun, $
>         /swap_if_little_endian
>     readu, lun, header
>
>
> Am I doing this correctly? I know I am close because I get what seems
> like
> a valid time and I know I get a valid datum. But my data doesn't seem
> right. I just want to check that I am using the correct types in my IDL
> struct and that I am reading in the data correctly.

```

The types appear to be correct at first glance. Do you know how the data were written out from the C program? Was it in a single write, or were they written as individual elements? If it was a single write there *may* be packing problems in the struct, although the definition above wouldn't appear to require any packing, the C compiler could have decided otherwise.

My favourite tool for debugging these sorts of problems is a binary/hex editor or dump tool such as `od` in UNIX. I don't know of similar tools for Windows as I don't use that platform. But there's no substitute for getting your hands dirty, delving into the data file and looking at the contents byte by byte to ensure you know *exactly* what its contents are. You really need at least one record converted to human readable form by some alternate means so you can compare that to the contents of the file and be sure you can identify the components in the binary structure.

I know from past experience that relying on the documentation can be an exercise in futility - it's all too easy for some well meaning individual to make an alteration to the code and not bother to change the documentation.

--

Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK

E-mail : nmw@ion.le.ac.uk

Phone : +44 (0)116 2523568, Fax : +44 (0)116 2523555
