

---

Subject: Re: Writing a structure containing empty strings  
Posted by [JD Smith](#) on Mon, 07 Aug 2006 21:34:12 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Mon, 07 Aug 2006 08:12:58 -0700, Wayne Landsman wrote:

> This message and the next one address long-standing problems I've had  
> with IDL software to handle the FITS data format used in astronomy.  
>  
> When writing an IDL structure, str, to FITS format, the following  
> simple code \*almost\* works.  
>  
> openw,lun, 'output.dat',/get\_lun  
> writeu,lun, str  
> free\_lun, lun  
>  
> The only problem occurs if one of the structure tags contains an empty  
> string. What I want is for IDL to write a zeroed byte (0b), but IDL  
> simply ignores the tag and doesn't write anything. Note that IDL  
> does understand 0b as the byte representation of an empty string:  
>  
> IDL> print,strlen(string(0b))  
> 0  
> IDL> print,byte("")  
> 0  
>  
> but it simply skip over an empty string when performing I/O. So my  
> first thought is to convert the string tags to bytes, but this has two  
> problems: (1) one can't directly change the data type of a structure  
> tag and (2) converting a string array to byte forces everything to a  
> fixed length. So right now, I loop over every element and every tag  
> of the structure and write it to disk:, if it is a string tag I test  
> for the presence of an empty string, and then further loop over the  
> string array to convert every element to a byte array before writing to  
> disk. So simple code that \*almost\* works, requires a major revision  
> to completely work. Or am I missing a simpler solution?

The manual has this to say:

When writing fields containing strings with the unformatted procedure  
WRITEU, IDL writes each character of the string and does not append a  
terminating null byte.

Since it's not appending \0 to any string, having it do so only in the  
special case of a zero-length string would actually be surprising. It's  
not ignoring the string, it's realizing it contains zero bytes and writing  
out zero bytes. On input, it would read in those zero bytes again ;).

Do FITS files really want non-zero terminated strings in all cases except for 0 length strings? Pretty strange.

JD

---