Subject: Re: read_ascii
Posted by Benjamin Tupper on Thu, 09 Dec 2004 13:52:12 GMT
View Forum Message <> Reply to Message

Rick Towler wrote:

>
>> The problem is that sometimes the array is contained in field01,
>> sometimes in field01.
>
>
>> How come it varies ?
>
>
> You mean "field1" and "field01".
>
> When confronted with less than 10 fields, READ_ASCII will return
> field1-field9, when you have more than 10, field01-field99.
>
> It is unfortunate that READ_ASCII behaves this way but David's
> suggestion of addressing the structure by tag index instead of tag name
> will solve your problem.
>

Hi,

You could define the names of the fields using ASCII_TEMPLATE before calling READ_ASCII. You can define the template without using ASCII_TEMPLATE - a little study of the structure returned by ASCII_TEMPLATE should help.

Alternatively, you could modify READ_ASCII (call it something else, though, like MY_READ_ASCII). You can force the routine to always use N digits in the field names by modifying the following line (line 874 in my version) ....

```
digits_str = $
 strtrim(string(strlen(strtrim(string(fieldCountUse),2))),2)
```

to something like this...

```
my_dig_len = fieldCountUse > 2
digits_str = $
 strtrim(string(strlen(strtrim(string(my_dig_len),2))),2)
```

On a related note, I have drifted away from treating columnated ASCII

data files as nColumns of vectors. Instead, I treat them as vectors of structures where each row is a record and each column is a field of the record. This works fine for flatly organized data and I find a vector of structures MUCH easier to manage in IDL than a structure of vectors. If you are interested seeing my version of READ_ASCII then shoot me an email.

Cheers,
Ben
