
Subject: Re: IDL Error GCPC data
Posted by [sampton](#) on Fri, 25 Dec 2009 17:38:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks a lot David,
Santa is here today so I'll get back to IDL prog. tommorow and
Hopefully I can do the homework ;)
Greeting,
Sam.

On 24 déc, 23:21, David Fanning <n...@dfanning.com> wrote:

```
> sampton writes:
>> OK ! you're the expert then ;)
>> As of the file extention what I mean is that I have more than one
>> binary data file ending by 'year': gpcp_v2.1_psg.1979, gpcp_v2.1_psg.
>> 1980 and so on.
>> I'll try those command and get back to you in case,
>> Thanks again and greeting fo the Holidays season.
>
> I am sitting around waiting for Santa to show up
> tonight, so I thought I'd just make a couple of
> changes to your code. This is not *exactly* how I
> would have written it, but I tried to keep the
> flavor of the original.
>
> Here it is:
>
> ;*****
> ;
> function create_v2_1_struct, $
>   NUM_LON=num_lon, $ ; Number of longitude values. Input. Default:
> 144
>   NUM_LAT=num_lat, $ ; Number of latitude values. Input. Default: 72
>   NUM_MON=num_mon, $ ; Number of months. Input. Default: 12
> ;-----
> ; Set up the data structure for output.
> ;-----
>   if n_elements(num_lon) eq 0 then num_lon = 144
>   if n_elements(num_lat) eq 0 then num_lat = 72
>   if n_elements(num_mon) eq 0 then num_mon = 12
>
>   struc = { header: bytarr(num_lon*4), $
>               data:  fltarr(num_lon,num_lat,num_mon) }
>
>   return, struc
> end
>
> function read_v2_1, file,
>   HELP=help, $ ; Print help message.
```

```

> HEADER=header, $ ; File header. Output.
> NUM_LON=num_lon, $ ; Number longitude values. Input. Default: 144
> NUM_LAT=num_lat, $ ; Number latitude values. Input. Default: 72
> NUM_MON=num_mon, $ ; Number months. Input. Default: 12
> ;-----
> ; The main procedure; create the data structure, read both
> ; header and data, and swap bytes if needed. Returns the data
> ; structure, and the header, if requested.
> ;-----
>
> ; Need some help?
> if keyword_set(help) then begin
>   print, 'struct = read_v2_1_file(filename, [HEADER=header], $
>   print, ' [NUM_LON=num_lon], [NUM_LAT=num_lat], [NUM_MON=num_mon]'
>   print, "
>   print, 'Arguments:----'
>   print, 'filename: The name of the file to read.'
>   print, "
>   print, 'Keywords:----'
>   print, 'HEADER: Output variable contains file header.'
>   print, 'NUM_LON: Number of longitude values. Input. Default: 144'
>   print, 'NUM_LAT: Number of latitude values. Input. Default: 72'
>   print, 'NUM_MON: Number of months. Input. Default: 12'
> endif
>
> ; Need a file?
> if n_elements(file) eq 0 then begin
>   file = dialog_pickfile(title='Select file to read...')
>   if file eq "" then return, -1
> endif
>
> ; Assign keyword default values.
> if n_elements(num_lon) eq 0 then num_lon = 144
> if n_elements(num_lat) eq 0 then num_lat = 72
> if n_elements(num_mon) eq 0 then num_mon = 12
>
> ; Read the header.
> header = bytarr( num_lon*4 )
> openr, lun, file, /get_lun
> readu, lun, header
> free_lun, lun
> header = str_sep( strtrim(string(header),2), ' ' )
>
> ; If the word "Silicon" is in the header, then the file
> ; order is big-endian.
> bigEndian = (strpos( header, 'Silicon') ne -1) ? 1 : 0
>
> ; Read the structure from the file. Swap bytes, if necessary.

```

```

> struct = create_v2_1_struct(NUM_LON=num_lon, $
>   NUM_LAT=num_lat, NUM_MON=num_mon)
> IF bigEndian THEN BEGIN
>   openr, lun, file, /get_lun, /swap_if_little_endian
> ENDIF ELSE BEGIN
>   openr, lun, file, /get_lun, /swap_if_big_endian
> ENDELSE
> readu, lun, struct
> free_lun, lun
>
> ; Return structure.
> return, struct
>
> end
> ;*****
>
> This will be called something like this:
>
> IDL> struct = Read_V2_1_File('gpcp_v2.1_psg.1979')
>
> If you want a help message, call it like this:
>
> IDL> void = Read_V2_1_File(/HELP)
>
> You can pass it the number of longitude values, number of latitude
> values, and number of months as keywords. If you want the header
> to come back separate from the structure, you can use the HEADER
> keyword as an output variable.
>
> IDL> file = 'gpcp_v2.1_psg.1979'
> IDL> struct = Read_V2_1_File(file, HEADER=thisHeader)
> IDL> Help, struct
> IDL> Print, thisHeader
>
> If you wanted to run this in a loop, you might try something
> like this:
>
> IDL> structs = PtrArr(10)
> IDL> files = 'gpcp_v2.1_psg' + StrTrim(Indgen(10)+1979,2)
> IDL> for j=0,9 do structs[j] = Ptr_New(Read_V2_1_File(files[j]))
> IDL> Help, *structs[5], /Structure
>
> Then you would have a pointer array to 10 structures, read
> from the files ending in 1979 to 1988.
>
> No error handling in this code, and the modules are not named
> as I would name them if I were writing the code. But we will
> leave that as an exercise for the reader. :-)

```

>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:<http://www.dfanning.com/>
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
