Subject: Re: IDL Error GCPC data
Posted by Wout De Nolf on Thu, 24 Dec 2009 08:59:18 GMT
View Forum Message <> Reply to Message

On Wed, 23 Dec 2009 23:27:27 -0800 (PST), sampton <simo2005@gmail.com>
wrote:

> but can't make it to run properly under idl71

Change *V2.1* in the routine names to something without a ".", for
example *V2_1*

Subject: Re: IDL Error GCPC data
Posted by sampton on Thu, 24 Dec 2009 16:57:26 GMT
View Forum Message <> Reply to Message

Thanks for your reply,
What I use is:
- Type "idl" at the UNIX command prompt to start IDL.
- Type: .run read_v2_1_file.pro to compile the procedures.
- Type: read_V2_1, "Bin_DATA_FILE_NAME", "STRUC"
and I changed the file names inside the pro file accordingly. At the
end no output file is generated, I'm expecting to read a binary file
but no file is generated.
Did I miss something ?

On Dec 24, 3:59 am, Wox <s...@nomail.com> wrote:
> On Wed, 23 Dec 2009 23:27:27 -0800 (PST), sampton <simo2...@gmail.com>
> wrote:
>
>> but can't make it to run properly under idl71
>
> Change *V2.1* in the routine names to something without a ".", for
> example *V2_1*

Subject: Re: IDL Error GCPC data
Posted by David Fanning on Thu, 24 Dec 2009 17:38:55 GMT
View Forum Message <> Reply to Message

sampton writes:

> What I use is:
> - Type "idl" at the UNIX command prompt to start IDL.
> - Type: .run read_v2_1_file.pro to compile the procedures.

> - Type: read_V2_1, "Bin_DATA_FILE_NAME", "STRUC"
> and I changed the file names inside the pro file accordingly. At the
> end no output file is generated, I'm expecting to read a binary file
> but no file is generated.
> Did I miss something ?

Quite a lot, actually. :-)

You won't have to do step 2, with the .RUN if you
move the read_V2_1 module to the bottom of the file and
name the file "read_v2_1.pro".

http://www.dfanning.com/tips/namefiles.html

If you do include this step, .COMPILE would be a better choice,
unless you always want to always see that printed help
message, which I think contains inaccurate information.

Then, on this command:

IDL> read_V2_1, "Bin_DATA_FILE_NAME", "STRUC"

Is that *really* the name of the file you want to open? That
is an *extremely* weird name. Normally, on LINUX systems filenames
will have all lowercase letters, unless there is some extraordinary
reason for not doing so.

In any case, you want "struct" to be a variable, not a string,
so you don't want to put quotes around it. If that filename
is a valid one, you probably want to call your program like this:

IDL> read_V2_1, "Bin_DATA_FILE_NAME", struct

This will pass the (undefined?) variable struct into the program
in a pass-by-reference (rather than pass-by-value) way. When the
program works, you *should* have struct defined as something that
got read out of the file. Is that was you intend?

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.")

Subject: Re: IDL Error GCPC data
Posted by sampton on Thu, 24 Dec 2009 18:18:05 GMT
View Forum Message <> Reply to Message

Thanks,
Yes I know that I'm missing a lot it's my first steeps with IDL ;)
As of the file name no I just want to mention that there is a binary
data file and the real name is: gpcp_v2.1_psg.1979
and the year extension changes from 1979 to 2009.
The command line I used are those contained in the pro file
(Instruction part) and yes I'll try your suggestion and see what I
get.
you suggested: .COMPILE do I have to include the file name in the
path or just the command ?
Thanks again.

On Dec 24, 12:38 pm, David Fanning <n...@dfanning.com> wrote:
> sampton writes:
>> What I use is:
>> - Type "idl" at the UNIX command prompt to start IDL.
>> - Type: .run read_v2_1_file.pro to compile the procedures.
>> - Type: read_V2_1, "Bin_DATA_FILE_NAME", "STRUC"
>> and I changed the file names inside the pro file accordingly. At the
>> end no output file is generated, I'm expecting to read a binary file
>> but no file is generated.
>> Did I miss something ?
>
> Quite a lot, actually. :-)
>
> You won't have to do step 2, with the .RUN if you
> move the read_V2_1 module to the bottom of the file and
> name the file "read_v2_1.pro".
>
> http://www.dfanning.com/tips/namefiles.html
>
> If you do include this step, .COMPILE would be a better choice,
> unless you always want to always see that printed help
> message, which I think contains inaccurate information.
>
> Then, on this command:
>
> IDL> read_V2_1, "Bin_DATA_FILE_NAME", "STRUC"
>
> Is that *really* the name of the file you want to open? That
> is an *extremely* weird name. Normally, on LINUX systems filenames
> will have all lowercase letters, unless there is some extraordinary
> reason for not doing so.
>
> In any case, you want "struct" to be a variable, not a string,

> so you don't want to put quotes around it. If that filename
> is a valid one, you probably want to call your program like this:
>
> IDL> read_V2_1, "Bin_DATA_FILE_NAME", struct
>
> This will pass the (undefined?) variable struct into the program
> in a pass-by-reference (rather than pass-by-value) way. When the
> program works, you *should* have struct defined as something that
> got read out of the file. Is that was you intend?
>
> 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.")

Subject: Re: IDL Error GCPC data
Posted by David Fanning on Thu, 24 Dec 2009 19:07:19 GMT
View Forum Message <> Reply to Message

sampton writes:

> Yes I know that I'm missing a lot it's my first steeps with IDL ;)

OK, then. Expected. :-)

> As of the file name no I just want to mention that there is a binary
> data file and the real name is: gpcp_v2.1_psg.1979
> and the year extension changes from 1979 to 2009.

I don't know what you mean by 'changing the year extension',
since there is nothing like that in your code. Your code opens
the file you supply it and reads a structure out of it. If this
is what you have in mind, then the command you would use is:

  IDL> read_v2_1, 'gpcp_v2.1_psg.1979', struct

This *assumes* the file is located in your current IDL directory.
Almost always a *bad* assumption, as it turns out. But we will
save that for a future lesson.

Your code then goes to a lot of trouble to do some byte swapping,
in needed. This can be MUCH more easily done by using the
SWAP_IF_*** keywords on the OPENR command. Look up the OPENR

statement in the on-line help.

> you suggested: .COMPILE do I have to include the file name in the
> path or just the command ?

If your file is named correctly, and is located somewhere in
your IDL path, then you just have to supply the "root" name
of the file:

IDL> .compile tvimage

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.")

Subject: Re: IDL Error GCPC data
Posted by sampton on Fri, 25 Dec 2009 02:29:10 GMT
View Forum Message <> Reply to Message

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.
S.S.

On 24 déc, 14:07, David Fanning <n...@dfanning.com> wrote:
> sampton writes:
>> Yes I know that I'm missing a lot it's my first steeps with IDL ;)
>
> OK, then. Expected. :-)
>
>> As of the file name no I just want to mention that there is a binary
>> data file and the real name is: gpcp_v2.1_psg.1979
>> and the year extension changes from 1979 to 2009.
>
> I don't know what you mean by 'changing the year extension',
> since there is nothing like that in your code. Your code opens
> the file you supply it and reads a structure out of it. If this
> is what you have in mind, then the command you would use is:
>

> IDL> read_v2_1, 'gpcp_v2.1_psg.1979', struct
>
> This *assumes* the file is located in your current IDL directory.
> Almost always a *bad* assumption, as it turns out. But we will
> save that for a future lesson.
>
> Your code then goes to a lot of trouble to do some byte swapping,
> in needed. This can be MUCH more easily done by using the
> SWAP_IF_*** keywords on the OPENR command. Look up the OPENR
> statement in the on-line help.
>
>> you suggested: .COMPILE do I have to include the file name in the
>> path or just the command ?
>
> If your file is named correctly, and is located somewhere in
> your IDL path, then you just have to supply the "root" name
> of the file:
>
> IDL> .compile tvimage
>
> 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.")

Subject: Re: IDL Error GCPC data
Posted by David Fanning on Fri, 25 Dec 2009 04:21:23 GMT
View Forum Message <> Reply to Message

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_LAT=num_lat, $  ; 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_LAT=num_lat, $  ; 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_LAT=num_lat]
      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_LAT:  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.")

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 Hopefelly 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_LAT=num_lat, $; 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_LAT=num_lat, $; 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_LAT=num_lat]
>       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_LAT:  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.")

Subject: Re: IDL Error GCPC data
Posted by sampton on Tue, 29 Dec 2009 19:28:06 GMT
View Forum Message <> Reply to Message

Hi,
Now that Santa is gone back to IDL ;)
Itried the code but get some syntax error handling in this code and
here is the error:

```
IDL> struct = Read_V2_1_file('gpcp_v2_1_psg.2009')

   if n_elements(num_lon) eq 0 then num_lon = 144
    ^
% Syntax error.
  At: ~/.../read_v2_1_file.pro, Line 44

   return, struc
             ^
% Return statement in procedures can't have values.
  At: ~/.../read_v2_1_file.pro, Line 51

function read_v2_1, file,
                    ^
% Syntax error.
  At: ~/.../read_v2_1_file.pro, Line 54

   endif
    ^
% Type of end does not match statement (END expected).
  At: ~/.../read_v2_1_file.pro, Line 79

      if file eq "" then return, -1
                              ^
% Return statement in procedures can't have values.
  At: ~/.../read_v2_1_file.pro, Line 84

   struct = create_v2_1_struct(NUM_LON=num_lon, $
                               ^
% Syntax error.
  At: ~/.../read_v2_1_file.pro, Line 104

   return, struct
             ^
% Return statement in procedures can't have values.
  At: ~/.../read_v2_1_file.pro, Line 115
% Compiled module: READ_V2_1_FILE.
% Attempt to call undefined procedure/function: 'READ_V2_1_FILE'.
% Execution halted at: $MAIN$
```

The initial code is working but still don't know how to save the

output in a .txt or .dat file.
Thanks again for your help.

Subject: Re: IDL Error GCPC data
Posted by David Fanning on Tue, 29 Dec 2009 19:35:25 GMT
View Forum Message <> Reply to Message

sampton writes:

> Now that Santa is gone back to IDL ;)
> Itried the code but get some syntax error handling in this code and
> here is the error:
> IDL> struct = Read_V2_1_file('gpcp_v2_1_psg.2009')

If you are going to call it as a function, you are going
to have to write it as a function. At the moment,
your code is written as a procedure. :-)

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.")

Subject: Re: IDL Error GCPC data
Posted by sampton on Tue, 05 Jan 2010 01:13:11 GMT
View Forum Message <> Reply to Message

On 29 déc 2009, 14:35, David Fanning <n...@dfanning.com> wrote:
> sampton writes:
>> Now that Santa is gone back to IDL ;)
>> Itried the code but get some syntax error handling in this code and
>> here is the error:
>> IDL> struct = Read_V2_1_file('gpcp_v2_1_psg.2009')
>
> If you are going to call it as a function, you are going
> to have to write it as a function. At the moment,
> your code is written as a procedure. :-)
>

> 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.")

Hi,
Thanks again for your help,
I'm just wondering how I can save struct to data.txt after executing
the command under IDL:
IDL> read_v2_1, 'gpcp_v2.1_psg.1979', struct
I want to save struct to data.txt to be oppened with text editor under
windows (so data.txt wil be located under a spedcific folder say: c:
\cgcp\data.txt).
Thanks again
Sam.
