Subject: skip the corrupted hdf file and continue the for lood for uncorrupted hdfs
Posted by Brent Fallarcuna on Mon, 27 Jan 2014 08:22:26 GMT
View Forum Message <> Reply to Message

Hello everyone!

I'm extracting values on a specific pixel from hundreds of hdf files and used for loop.

This is my code:

```
;Daily NPP/GPP Plotting
cd, ('C:\Users\Brent\Documents\NPP Annual Files\MODIS17A2v5')
filelist = file_search('*.*.h29v07.005.*.hdf')
i = 0
arr = strarr(1,637)

FOR i = 0, 636 DO BEGIN
  ; Open HDF and Import SDS
  filename = filelist[i]

  ;access hdf file
  hdf_id = hdf_sd_start(filename);this is where my program stops

  ;return the index of 'Npp_1km' variable
  index = hdf_sd_nametoindex(hdf_id, 'PsnNet_1km')

  ;access the dataset
  varid = hdf_sd_select(hdf_id, index)

  ;read the contents of the variable
  hdf_sd_getdata, varid, data

  ;endaccess
  hdf_sd_endaccess, varid
  hdf_sd_end, hdf_id

  ;rotate hdf
  data = rotate(data, 7)
  ;scale the data
  kaliwa_tile = data * (0.0001)
  point = kaliwa_tile[880,566]

  print, point
  arr[0,i] = point
ENDFOR

cd, ('C:\Users\Brent\Documents\MOD17A3UTM')
openw, lun, 'outputfilefor8dayPsnNet1km.csv', /get_lun
```

```
printf, lun, arr, format= '(a)'
close, lun
free_lun, lun

END
```

After the values from several hdfs are printed, there's an error on the console saying "HDF_SD_START: Unable to start the HDF-SD interface."
As I checked the variables tab, the i stopped at the 24th hdf file. As I checked the file using hdf browser, it didn't open, thus the file seems to be corrupted.

My question is, can I continue my for loop and proceed reading the other uncorrupted hdf files every time I got the said error?

I've tried catch statement but I can't figure it out on how to use and implement it correctly.

Brent

Subject: Re: skip the corrupted hdf file and continue the for lood for uncorrupted hdfs
Posted by Yngvar Larsen on Mon, 27 Jan 2014 09:25:45 GMT
View Forum Message <> Reply to Message

On Monday, 27 January 2014 09:22:26 UTC+1, Brent Fallarcuna wrote:

> My question is, can I continue my for loop and proceed reading the other uncorrupted hdf files every time I got the said error?
>
> I've tried catch statement but I can't figure it out on how to use and implement it correctly.
>

The solution should involve CATCH. Something like this:

```
for i=0, nfiles-1 do begin
  ;; Establish error handler
  catch, error
  if error ne 0 then begin
     print, filename[i]
     print, !error_state.msg
     ;; continue with next item in loop if error was encountered
     catch, /cancel
     continue
  endif

  hdf_id = hdf_sd_start(filename[i])
```

```
  catch, /cancel
  ;; Do other stuff with file here if no error was encountered.
  ;; ...
endfor
```

--
Yngvar

Subject: Re: skip the corrupted hdf file and continue the for lood for uncorrupted hdfs
Posted by Brent Fallarcuna on Tue, 28 Jan 2014 05:14:03 GMT
View Forum Message <> Reply to Message

On Monday, January 27, 2014 4:22:26 PM UTC+8, Brent Fallarcuna wrote:

```
>  Hello everyone!
>
>
>
>  I'm extracting values on a specific pixel from hundreds of hdf files and used for loop.
>
>
>
>  This is my code:
>
>
>
>  ;Daily NPP/GPP Plotting
>
>  cd, ('C:\Users\Brent\Documents\NPP Annual Files\MODIS17A2v5')
>
>  filelist = file_search('*.*.h29v07.005.*.hdf')
>
>  i = 0
>
>  arr = strarr(1,637)
>
>
>
>  FOR i = 0, 636 DO BEGIN
>
>    ; Open HDF and Import SDS
>
>    filename = filelist[i]
>
>
>
>    ;access hdf file
```

```
>
>   hdf_id = hdf_sd_start(filename);this is where my program stops
>
>
>
>   ;return the index of 'Npp_1km' variable
>
>   index = hdf_sd_nametoindex(hdf_id, 'PsnNet_1km')
>
>
>
>   ;access the dataset
>
>   varid = hdf_sd_select(hdf_id, index)
>
>
>
>   ;read the contents of the variable
>
>   hdf_sd_getdata, varid, data
>
>
>
>   ;endaccess
>
>   hdf_sd_endaccess, varid
>
>   hdf_sd_end, hdf_id
>
>
>
>   ;rotate hdf
>
>   data = rotate(data, 7)
>
>   ;scale the data
>
>   kaliwa_tile = data * (0.0001)
>
>   point = kaliwa_tile[880,566]
>
>
>
>   print, point
>
>   arr[0,i] = point
>
> ENDFOR
```

>
>
>
> cd, ('C:\Users\Brent\Documents\MOD17A3UTM')
>
> openw, lun, 'outputfilefor8dayPsnNet1km.csv', /get_lun
>
> printf, lun, arr, format= '(a)'
>
> close, lun
>
> free_lun, lun
>
>
>
> END
>
>
>
>
>
> After the values from several hdfs are printed, there's an error on the console saying "HDF_SD_START: Unable to start the HDF-SD interface."
>
> As I checked the variables tab, the i stopped at the 24th hdf file. As I checked the file using hdf browser, it didn't open, thus the file seems to be corrupted.
>
>
>
> My question is, can I continue my for loop and proceed reading the other uncorrupted hdf files every time I got the said error?
>
>
>
> I've tried catch statement but I can't figure it out on how to use and implement it correctly.
>
>
>
>
>
>
> Brent

Hello.

I've run this code, wherein I inserted the for loop. The output on the command line says "MOD17A2.A2000065.h29v07.005.2010162162935.hdf
Attempt to subscript FILENAME with K is out of range." and it only prints a single value.

Perhaps IDL is confused with my code structure. Is it ok to use for loop within the for loop? What I needed is to identify those corrupted files (to re-download them again) and put the values in my csv output file.

```
;Daily NPP/GPP Plotting

cd, ('C:\Users\Brent\Documents\NPP Annual Files\MODIS17A2v5')

filelist = file_search('*.*.h29v07.005.*.hdf')

k = 0
arr = strarr(1,637)

FOR k = 0, 636 DO BEGIN
  ; Open HDF and Import SDS
  filename = filelist[k]

     for i=0, 636 do begin ;<====This is where I inserted the for loop
     ;establish error handler
     catch, error
     if error ne 0 then begin
        print, filename[i]
        print, !error_state.msg
        ;continue with next item in loop if error was encountered
        catch, /cancel
        continue
     endif
     ;access hdf file

     hdf_id = hdf_sd_start(filename[k])
     catch, /cancel
     ;Do other stuff with file here if no error was encountered.

     ;return the index of 'Npp_1km' variable
     index = hdf_sd_nametoindex(hdf_id, 'PsnNet_1km')

     ;access the dataset
     varid = hdf_sd_select(hdf_id, index)

     ;read the contents of the variable
     hdf_sd_getdata, varid, data

     ;endaccess
     hdf_sd_endaccess, varid
     hdf_sd_end, hdf_id

     ;rotate hdf
```

```
    data = rotate(data, 7)
    ;scale the data
    kaliwa_tile = data * (0.0001)
    point = kaliwa_tile[880,566]

    print, point
    arr[0,k] = point
    endfor
ENDFOR

cd, ('C:\Users\Brent\Documents\MOD17A3UTM')
openw, lun, 'outputfilefor8dayPsnNet1km.csv', /get_lun
printf, lun, arr, format= '(a)'
close, lun
free_lun, lun

END
```

Brent

Subject: Re: skip the corrupted hdf file and continue the for lood for uncorrupted hdfs
Posted by David Fanning on Tue, 28 Jan 2014 05:37:12 GMT
View Forum Message <> Reply to Message

Brent Fallarcuna writes:

> I've run this code, wherein I inserted the for loop. The output on the command line says "MOD17A2.A2000065.h29v07.005.2010162162935.hdf
> Attempt to subscript FILENAME with K is out of range." and it only prints a single value.
>
> Perhaps IDL is confused with my code structure. Is it ok to use for loop within the for loop?
> What I needed is to identify those corrupted files (to re-download them again) and put the values in my csv output file.

Oh, dear. :-(

It is OK to use loops within loops if you have to, but I see no evidence that you have to do so in anything you have told us or in this code. I can't even figure out where the number 637 comes from. Doesn't the size of your loop depend on how many files you have to open?

I would write something along these lines:

```
filelist = file_search('*.*.h29v07.005.*.hdf', Count=count)
```

```
array = StrArr(count)
FOR j=0,count-1 DO BEGIN

  Catch, theError
  IF theError NE 0 THEN BEGIN
     void = cgErrorMsg()
     Print, 'Bad File: ', filename[j]
     Message, /Reset
     Continue
   ENDIF

   thisFile = filelist[j]
   hdf_id = hdf_sd_start(thisFile)
   ...

ENDFOR
```

Cheers,

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

Subject: Re: skip the corrupted hdf file and continue the for lood for uncorrupted hdfs

Posted by David Fanning on Tue, 28 Jan 2014 05:47:22 GMT
View Forum Message <> Reply to Message

David Fanning writes:

```
>  I would write something along these lines:
>
>  filelist = file_search('*.*.h29v07.005.*.hdf', Count=count)
>  array = StrArr(count)
>  FOR j=0,count-1 DO BEGIN
>
>     Catch, theError
>     IF theError NE 0 THEN BEGIN
>        void = cgErrorMsg()
>        Print, 'Bad File: ', filename[j]
>        Message, /Reset
>        Continue
>      ENDIF
```

>
>   thisFile = filelist[j]
>   hdf_id = hdf_sd_start(thisFile)
>   ...
>
> ENDFOR

Whoops! I just violated the cardinal rule of error handling: Don't introduce errors into your error handling code! Kill your IDL session with your mouse when you get into the infinite loop I put you into, then make this change to your code. Sorry! :-(

```
filelist = file_search('*.*.h29v07.005.*.hdf', Count=count)
array = StrArr(count)
FOR j=0,count-1 DO BEGIN

  Catch, theError
  IF theError NE 0 THEN BEGIN
     void = cgErrorMsg()
     Print, 'Bad File: ', thisFile
     Message, /Reset
     Continue
   ENDIF

   thisFile = filelist[j]
   hdf_id = hdf_sd_start(thisFile)
   ...

ENDFOR
```

Cheers,

David

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

Subject: Re: skip the corrupted hdf file and continue the for lood for uncorrupted hdfs
Posted by Yngvar Larsen on Tue, 28 Jan 2014 17:52:02 GMT
View Forum Message <> Reply to Message

On Tuesday, 28 January 2014 06:14:03 UTC+1, Brent Fallarcuna wrote:

> On Monday, January 27, 2014 4:22:26 PM UTC+8, Brent Fallarcuna wrote:

> Hello.
>
> I've run this code, wherein I inserted the for loop. The output on the command line says "MOD17A2.A2000065.h29v07.005.2010162162935.hdf
>
> Attempt to subscript FILENAME with K is out of range." and it only prints a single value.

Well. This is a rather accurate error message, because this is exactly what is going on. You (or actually I since you copied my code verbatim) made an error within the error handler. The value to print in the error handler is FILENAME or FILELIST[i], not FILENAME[i]. Of course, in your case, you most likely would like to save the name of the erroneous file in the error handler instead of just printing the name.

```
>       catch, error
>       if error ne 0 then begin
>          print, filename[i]       <------------ ERROR IN ERROR HANDLER
>          print, !error_state.msg
>          ;continue with next item in loop if error was encountered
>          catch, /cancel
>          continue
>       endif
>
>       ;access hdf file
>       hdf_id = hdf_sd_start(filename[k])  <--------- ANOTHER ERROR
>       catch, /cancel
>
>       ;Do other stuff with file here if no error was encountered.
```

Also, I have no idea what you are trying to do with the double loop.

--
Yngvar

Subject: Re: skip the corrupted hdf file and continue the for lood for uncorrupted hdfs
Posted by Brent Fallarcuna on Wed, 29 Jan 2014 03:14:22 GMT
View Forum Message <> Reply to Message

On Tuesday, January 28, 2014 1:47:22 PM UTC+8, David Fanning wrote:
> David Fanning writes:
>
>
>
>> I would write something along these lines:
>

```
>>
>
>>  filelist = file_search('*.*.h29v07.005.*.hdf', Count=count)
>
>>  array = StrArr(count)
>
>>  FOR j=0,count-1 DO BEGIN
>
>>
>
>>     Catch, theError
>
>>     IF theError NE 0 THEN BEGIN
>
>>        void = cgErrorMsg()
>
>>        Print, 'Bad File: ', filename[j]
>
>>        Message, /Reset
>
>>        Continue
>
>>      ENDIF
>
>>
>
>>      thisFile = filelist[j]
>
>>      hdf_id = hdf_sd_start(thisFile)
>
>>      ...
>
>>
>
>>  ENDFOR
>
>
>
>  Whoops! I just violated the cardinal rule of error handling: Don't
>
>  introduce errors into your error handling code! Kill your IDL session
>
>  with your mouse when you get into the infinite loop I put you into, then
>
>  make this change to your code. Sorry! :-(
>
>
>
```

```
> filelist = file_search('*.*.h29v07.005.*.hdf', Count=count)
>
> array = StrArr(count)
>
> FOR j=0,count-1 DO BEGIN
>
>
>
>   Catch, theError
>
>   IF theError NE 0 THEN BEGIN
>
>      void = cgErrorMsg()
>
>      Print, 'Bad File: ', thisFile
>
>      Message, /Reset
>
>      Continue
>
>    ENDIF
>
>
>
>    thisFile = filelist[j]
>
>    hdf_id = hdf_sd_start(thisFile)
>
>    ...
>
>
>
> ENDFOR
>
>
>
>
>
> Cheers,
>
>
>
> David
>
>
>
> --
>
```

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

Thank you Sir Fanning, I've successfully ran my code and able to see corrupted hdf files for redownload.

On my code, my input x,y pixel location for my area of interest was based from your coyote page: http://www.idlcoyote.com/map_tips/latlon2pixel.php.
I've read it again and it says, "Note that we are not checking that the latitude and longitude is actually inside our image. We should be, since we will get incorrect results if this is the case."
How am I able to check if my input pixel location falls within my desired area (based field based lat long reading)?

Cheers,
Brent

Subject: Re: skip the corrupted hdf file and continue the for lood for uncorrupted hdfs
Posted by David Fanning on Wed, 29 Jan 2014 03:38:03 GMT
View Forum Message <> Reply to Message

Brent Fallarcuna writes:

> On my code, my input x,y pixel location for my area of interest was based from your coyote page: http://www.idlcoyote.com/map_tips/latlon2pixel.php.
> I've read it again and it says, "Note that we are not checking that the latitude and longitude is actually inside our image. We should be, since we will get incorrect results if this is the case."
> How am I able to check if my input pixel location falls within my desired area (based field based lat long reading)?

Well, in the usual way:

```
IF (myLat GE minlat) && (myLat LE maxlat) $
   THEN Print, 'Yea!' $
   ELSE Print, 'Yikes!'
```

Cheers,

David

--
David Fanning, Ph.D.