
Subject: Question regarding HDF file
Posted by [None\[1\]](#) on Sun, 05 Aug 2007 19:50:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi. I have a problem rewriting the data to the original hdf file (i.e., to the same dataset which i had read from, earlier) I am getting negative data values. Also I am getting the same negative data values if i put a dummy array instead of the data in the HDF_SD_ADDDATA command. I got correct values when I output the result to a tiff file, but not able to get the correct values if I write it back to the original file. I have posted my code bellow. Can anybody please help me with a solution

```
sdFileID2 = HDF_SD_Start(fname, /RdWr)
sdsIDsm = HDF_SD_Select(sdFileID2,im[j]); im[j] are different bands
HDF_SD_ADDDATA, sdsIDsm, rowp, start=[0,0], count=[x[j],y[j]]
; rowp is the dataset and is of type float
; x,y are the array dimensions
HDF_SD_EndAccess, sdsIDsm
```

Thank you

Rajesh

Subject: Re: Question regarding HDF file
Posted by [James Kuyper](#) on Mon, 06 Aug 2007 21:30:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

None wrote:

...

> Thanks for replying. No I have not missed that command. This is what I
> am doing. I am reading the dataset into a text file. the text file
> contains 3 columns seperated by spaces. I read the text file line by
> line to get the 3 values (D,A and G). use it in the formula to get the
> results and write it back to the HDF file. ...

If you're not doing anything to that text file between writing it and reading it, what is the purpose of the text file? Why not use the data directly? The only reason I can think of is type conversion, but that doesn't make sense because direct type conversion is simpler and faster.

> ... I get the correct results
> if I output the results to tiff file. But I am not able to get the
> same results when I write it back to the HDF file. this is what my
> code looks like
>

```

> sdFileID2 = HDF_SD_Start(fname, /RdWr)
> sdsIDsm = HDF_SD_Select(sdFileID2,im[j]) ; Image data
> sdsIDsm2 = HDF_SD_Select(sdFileID2,bd[j]) ; recorection data

```

I mentioned earlier that I think that a type conversion might be the cause of the problem. If you don't know what the HDF data types are in your input file, please insert the lines shown below to find out. I'm also asking you to print out the ranges of the variables, because that will make it clearer whether or not your problems are caused by a type conversion.

```

> HDF_SD_GETDATA, sdsIDsm, rr, start=[0,0], count=[x[j],y[j]],
> stride=[0,0]

```

```

HDF_SD_GetInfo, sdsIDsm, NAME=name, HDF_TYPE=htype, TYPE=itype
PRINT, im[j], name, htype, itype, min(rr), max(rr)

```

```

> HDF_SD_GETDATA, sdsIDsm2, r2

```

```

HDF_SD_GetInfo, sdsIDsm2, NAME=name, HDF_TYPE=htype, TYPE=itype
PRINT, bd[j], name, htype, itype, min(r2), max(r2)

```

```

> ; Writing it into a text file
> openw, lun, out_path+'Step1\' +strcompress(fn[3],/remove_all)
> +'-'+strcompress(j,/remove_all)+' .txt', /get_lun
> printf, lun, r2
> close, lun
> free_lun, lun
>
> results=fltarr(x[j],y[j])
> rowp=fltarr(x[j],y[j])
> openr, lun1, out_path+'Step1\' +strcompress(fn[3],/remove_all)
> +'-'+strcompress(j,/remove_all)+' .txt', /get_lun
> ; Reading line by line to D, A, G values
> while ((~eof(lun1)) && (n lt x[j])) do begin
>     readf, lun1, temp1
>     sr = strsplit(temp1,' ', /Extract)
>     d=float(sr[0])
>     a=float(sr[1])
>     g=float(sr[2])
>     print, d, a, g
>     results[n,*]=(float(a)*(rr[(n),*])/float(g))+float(d)
>     rowp[n,*]=(3.14159*results[n,*]*sd*sd)/(esun[j]*COS((90-
> theta)*3.14159/180))

```

```

PRINT, min(rowp), max(rowp)

```

Note: you should use !PI instead of 3.14159, and !DTOR instead of !PI/

180. It's marginally more accurate, but the main reason for doing so is that it replaces "magic" numbers with named constants whose name implies what they mean. That makes it easier to maintain your code.

```
>      n=n+1
> endwhile
> HDF_SD_ADDDATA, sdsIDsm, rowp, start=[0,0], count=[x[j],y[j]],
> stride=[0,0]
> HDF_SD_EndAccess, sdsIDsm
```

```
HDF_SD_EndAccess, sdsIDsm2
```

```
> HDF_SD_END, sdFileID2
```

Subject: Re: Question regarding HDF file
Posted by [None\[1\]](#) on Tue, 07 Aug 2007 15:21:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

```
>> HDF_SD_GETDATA, sdsIDsm, rr, start=[0,0], count=[x[j],y[j]],
>> stride=[0,0]
>
> HDF_SD_GetInfo, sdsIDsm, NAME=name, HDF_TYPE=htype, TYPE=itype
> PRINT, im[j], name, htype, itype, min(rr), max(rr)
>
>> HDF_SD_GETDATA, sdsIDsm2, r2
>
> HDF_SD_GetInfo, sdsIDsm2, NAME=name, HDF_TYPE=htype, TYPE=itype
> PRINT, bd[j], name, htype, itype, min(r2), max(r2)
>
> PRINT, min(rowp), max(rowp)
>
```

Hi. I did the corrections which you had mentioned, but still I am getting the same results. I also changed the !PI and !DTOR. I have changed it to read from the variable directly instead writing to the text file and again reading from it. these are the information which I got.

```
sdsIDsm=      9 ImageData DFNT_UINT8 BYTE  40 249
sdsIDsm2=     10 RadiometricCorrTable DFNT_FLOAT32 FLOAT
-3.57843      2.47200
```

```
Max(rowp) & Min(rowp) =      0.506396      0.000000
```

But if i open the hdf file, I am getting values like -3.5123 -1.5232
-3.5138

code:

```
sdFileID2 = HDF_SD_Start(fname, /RdWr)
sdsIDsm = HDF_SD_Select(sdFileID2,im[j]) ; Image data
sdsIDsm2 = HDF_SD_Select(sdFileID2,bd[j]) ; recorection data
HDF_SD_GETDATA, sdsIDsm, rr
HDF_SD_GETDATA, sdsIDsm2, r2
sz=size(rr, /DIMENSIONS)
x[j]=fix(strcompress(sz[0],/remove_all))
y[j]=fix(strcompress(sz[1],/remove_all))
results=fltarr(x[j],y[j])
rowp=fltarr(x[j],y[j])

; Reading D, A, G values
i=0
while ((i lt (y[j]-1)) && (n lt (x[j]-1))) do begin
    d=r2[0,i]
    a=r2[1,i]
    g=r2[2,i]
    results[n,*]=(float(a)*(rr[(n),*])/float(g))+float(d)
    rowp[n,*]=(!PI*results[n,*]*sd*sd)/(esun[j]*COS((90-theta)*!
DTOR))
    n=n+1
endwhile
HDF_SD_ADDDATA, sdsIDsm, rowp, start=[0,0], count=[x[j],y[j]],
stride=[0,0]
HDF_SD_EndAccess, sdsIDsm
HDF_SD_END, sdFileID2
```

Thanks
Rajesh

Subject: Re: Question regarding HDF file
Posted by [James Kuyper](#) on Tue, 07 Aug 2007 17:43:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

None wrote:

```
>>> HDF_SD_GETDATA, sdsIDsm, rr, start=[0,0], count=[x[j],y[j]],
>>> stride=[0,0]
>>
>> HDF_SD_GetInfo, sdsIDsm, NAME=name, HDF_TYPE=htype, TYPE=itype
>> PRINT, im[j], name, htype, itype, min(rr), max(rr)
>>
>>> HDF_SD_GETDATA, sdsIDsm2, r2
>>
>> HDF_SD_GetInfo, sdsIDsm2, NAME=name, HDF_TYPE=htype, TYPE=itype
>> PRINT, bd[j], name, htype, itype, min(r2), max(r2)
```

```
>>
>> PRINT, min(rowp), max(rowp)
>>
>
> Hi. I did the corrections which you had mentioned, but still I am
> getting the same results. I also changed the !PI and !DTOR.
```

Those were debugging printouts, not corrections. They're not intended to solve the problem, they're intended to help figure out what the problem is.

```
> I have changed it to read from the variable directly instead writing
> to the text file and again reading from it. these are the information
> which I got.
>
> sdsIDsm=      9 ImageData DFNT_UINT8 BYTE  40 249
> sdsIDsm2=     10 RadiometricCorrTable DFNT_FLOAT32 FLOAT
> -3.57843      2.47200
>
> Max(rowp) & Min(rowp) =      0.506396      0.000000
```

I see a very serious problem right here, and it's a problem at the design level, not at the level of coding. You're attempting to write floating point values that range from 0.0 to 0.5 to an output array whose type is an 8-bit unsigned integer. The best you can hope for is automatic conversion from the floating point value to the 8-bit unsigned integer; that would result in the output array being filled with 0s.

```
IDL> filename = 'test.hdf'
IDL> sd_name = 'Bytes'
IDL> file = HDF_SD_Start(filename, /CREATE)
% Loaded DLM: HDF.
IDL> Dims = [30,40]
IDL> sds = HDF_SD_Create(file, sd_name, dims, /DFNT_UINT8)
IDL> data = BYTE(40+randomu(Seed, dims)*209)
IDL> print,min(data),max(data)
  40 248
IDL> HDF_SD_AddData, sds, data
IDL> HDF_SD_EndAccess, sds
IDL> HDF_SD_End,file

IDL> file = HDF_SD_Start(filename, /RDWR)
IDL> idx = HDF_SD_NameToIndex(file, sd_name)
IDL> sds = HDF_SD_Select(file, idx)
IDL> fdata = 0.506396*randomu(Seed, dims)
IDL> HDF_SD_AddData, sds, fdata
IDL> HDF_SD_EndAccess, sds
```

```
IDL> HDF_SD_End, file
```

```
IDL> idx = HDF_SD_NameToIndex(file, sd_name)
```

```
IDL> sds = HDF_SD_Select(file, idx)
```

```
IDL> HDF_SD_GetData,sds, data
```

```
IDL> print, min(data), max(data)
```

```
0 0
```

```
IDL> HDF_SD_EndAccess, sds
```

```
IDL> HDF_SD_End, file
```

> But if i open the hdf file, I am getting values like -3.5123 -1.5232

> -3.5138

How exactly are you reading the file? If you use the same methods shown in this code, you should be reading in the ImageData SDS as a BYTE array, which should have such values.

Subject: Re: Question regarding HDF file

Posted by [James Kuyper](#) on Tue, 07 Aug 2007 17:52:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

I accidentally sent this before it was complete; please ignore previous version.

None wrote:

```
>>> HDF_SD_GETDATA, sdsIDsm, rr, start=[0,0], count=[x[j],y[j]],
```

```
>>> stride=[0,0]
```

```
>>
```

```
>> HDF_SD_GetInfo, sdsIDsm, NAME=name, HDF_TYPE=htype, TYPE=itype
```

```
>> PRINT, im[j], name, htype, itype, min(rr), max(rr)
```

```
>>
```

```
>>> HDF_SD_GETDATA, sdsIDsm2, r2
```

```
>>
```

```
>> HDF_SD_GetInfo, sdsIDsm2, NAME=name, HDF_TYPE=htype, TYPE=itype
```

```
>> PRINT, bd[j], name, htype, itype, min(r2), max(r2)
```

```
>>
```

```
>> PRINT, min(rowp), max(rowp)
```

```
>>
```

```
>
```

> Hi. I did the corrections which you had mentioned, but still I am

> getting the same results. I also changed the !PI and !DTOR.

Most of the changes I suggested were debugging printouts, not corrections. They're not intended to solve the problem, they're intended to help figure out what the problem is. My other suggestions caused minor improvements in your code, but I didn't expect those improvements to solve the main problem.

```

> I have changed it to read from the variable directly instead writing
> to the text file and again reading from it. these are the information
> which I got.
>
> sdsIDsm=      9 ImageData DFNT_UINT8 BYTE  40 249
> sdsIDsm2=     10 RadiometricCorrTable DFNT_FLOAT32 FLOAT
> -3.57843      2.47200
>
> Max(rowp) & Min(rowp) =      0.506396      0.000000

```

I see a very serious problem right here, and it's a problem at the design level, not at the level of coding. You're attempting to write floating point values that range from 0.0 to 0.5 to an output array whose type is an 8-bit unsigned integer. This should cause an automatic conversion from the floating point value to the 8-bit unsigned integer; which should result in the output array being filled with 0s. If you want to store values other than 0, you're going to have to figure out a different way to calculate them, or a different way to store them. Just to confirm my assumptions, I performed the following test:

```

IDL> filename = 'test.hdf'
IDL> sd_name = 'Bytes'
IDL> file = HDF_SD_Start(filename, /CREATE)
% Loaded DLM: HDF.
IDL> Dims = [30,40]
IDL> sds = HDF_SD_Create(file, sd_name, dims, /DFNT_UINT8)
IDL> data = BYTE(40+randomu(Seed, dims)*209)
IDL> print,min(data),max(data)
    40 248
IDL> HDF_SD_AddData, sds, data
IDL> HDF_SD_EndAccess, sds
IDL> HDF_SD_End,file

```

```

IDL> file = HDF_SD_Start(filename, /RDWR)
IDL> idx = HDF_SD_NameToIndex(file, sd_name)
IDL> sds = HDF_SD_Select(file, idx)
IDL> fdata = 0.506396*randomu(Seed, dims)
IDL> HDF_SD_AddData, sds, fdata
IDL> HDF_SD_EndAccess, sds
IDL> HDF_SD_End, file

```

```

IDL> idx = HDF_SD_NameToIndex(file, sd_name)
IDL> sds = HDF_SD_Select(file, idx)
IDL> HDF_SD_GetData,sds, data
IDL> print, min(data), max(data)
    0  0

```

```
IDL> HDF_SD_EndAccess, sds
IDL> HDF_SD_End, file
```

This result is consistent with my expectation that HDF_SD_AddData performs automatic conversion from the IDL data type to the HDF data type. However, this is inconsistent with the symptoms you describe:

```
> But if i open the hdf file, I am getting values like -3.5123 -1.5232
> -3.5138 .....
```

How exactly are you reading the file? If you use the same methods shown in this code, you should be reading in the ImageData SDS as a BYTE array, which should be incapable of representing either negative or fractional values.

Subject: Re: Question regarding HDF file
Posted by [None\[1\]](#) on Tue, 07 Aug 2007 21:25:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

```
>> But if i open the hdf file, I am getting values like -3.5123 -1.5232
>> -3.5138 .....
>
> How exactly are you reading the file? If you use the same methods
> shown in this code, you should be reading in the ImageData SDS as a
> BYTE array, which should be incapable of representing either negative
> or fractional values.
```

Hi. The dataset I am using is a kind of remote sensing dataset (ASTER). Sorry to let you know lately that I was opening the hdf file in ENVI 4.1 software. Yes, if I read it using HDF_SD_GETDATA, I will get all zeros. I understood your point, This result is consistent with my expectation that HDF_SD_AddData performs automatic conversion from the IDL data type to the HDF data type.

But is there a way to avoid this?

Subject: Re: Question regarding HDF file
Posted by [James Kuyper](#) on Tue, 07 Aug 2007 21:55:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

None wrote:
>>> But if i open the hdf file, I am getting values like -3.5123 -1.5232
>>> -3.5138
>>

>> How exactly are you reading the file? If you use the same methods
>> shown in this code, you should be reading in the ImageData SDS as a
>> BYTE array, which should be incapable of representing either negative
>> or fractional values.
>
> Hi. The dataset I am using is a kind of remote sensing dataset
> (ASTER). Sorry to let you know lately that I was opening the hdf file
> in ENVI 4.1 software. ...

Without being familiar with ASTER data, it sounds like ENVI is handling it the same way it handles MODIS L1B data when you choose the File/Open External File/EOS/MODIS option. A floating point scale factor and offset for each band is stored as an SDS attribute for each image SDS in a MODIS L1B file. ENVI knows how to apply those scale factors and offsets to calculate correct values.

If you want to take an existing ASTER image, transform it in some fashion, and then write it back to the file, you're going to have to find out how the corresponding scale and offset values are stored in the ASTER data. This should be documented in their file specifications. You'll have to apply that scale and offset before applying your transformation. After performing your transformation, you'll have to apply the scale and offset in reverse. Depending upon your transformation, the new values should also be within the range from 0 to 255, just like the original values, so they'll still fit in 8-bit unsigned integers. Those are the values you'll want to write back to the file.

> ... Yes, if I read it using HDF_SD_GETDATA, I will
> get all zeros. I understood your point,
> This result is consistent with my expectation that HDF_SD_AddData
> performs automatic conversion from the IDL data type to the HDF data
> type.
>
> But is there a way to avoid this?

You can't do what you wanted to do; not this way. You need to choose from a wide variety of alternatives which are NOT what you originally wanted to do.

You can't change the data type of an HDF SDS once it has been created. If you want to store actual floating point values in an SDS, you'll have to create a new SDS for that purpose, with the appropriate HDF data type, either in the same file or a new one. Of course, this loses whatever advantage you hoped to gain by re-writing the data in-place.

Subject: Re: Question regarding HDF file
Posted by [None\[1\]](#) on Wed, 08 Aug 2007 16:37:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi. Thanks for the help. I want the output to be in float. I understand that it cannot be written back to the same file. I have to figure out something else. Anyways I am getting the values I wanted writing it to the tiff file. so I guess I would stick with it.

Thank you
Rajesh

Subject: Re: Question regarding HDF file
Posted by [James Kuyper](#) on Thu, 09 Aug 2007 01:02:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

None wrote:

- > Hi. Thanks for the help. I want the output to be in float. I
- > understand that it cannot be written back to the same file.

You can write it back to the same file, you just have to create a new SDS within that file to store the floating point values.
