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.