
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
```