
Subject: Re: MOD43B3 Col.4 Processing. unexpected error message. why?
Posted by [devin.white](#) on Thu, 12 Jul 2007 00:07:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

I build this sample program based on his code and it ran without throwing an error. The four output files look fine, too. Very interesting....

```
pro mod43b3_test
compile_opt idl2
```

```
;Simulate a MOD43B3 albedo SD data cube returned using HDF_SD_GETDATA
data_array = intarr(2,10,1200,1200)
```

```
;Get the simulated black and white sky albedo arrays
albedo_black = fltarr(1200,1200)
albedo_white = fltarr(1200,1200)
albedo_black[*,*] = data_array[0,9,*,*]
albedo_white[*,*] = data_array[1,9,*,*]
```

```
;Set up map information
mc = [0.5, 0.5, 11119968.509D, 4447338.766D]
ps = [926.6254331D, 926.6254331D]
units = envi_translate_projection_units('Meters')
params1 = [6371007.181D, 0, 0, 0]
Projection_Name1 = 'SIN_MODIS'
map_info = ENVI_MAP_INFO_CREATE(type=16, name=Projection_Name1,
params=params1, $
    UNITS = units, MC = mc, PS = ps)
```

```
black_out_name = 'c:\albedo_black_map_4.img'
white_out_name = 'c:\albedo_white_map_4.img'
```

```
ENVI_WRITE_ENVI_FILE, albedo_black, out_name=black_out_name,
map_info=map_info, out_dt=4, $
    r_fid=albedo_black_map, sensor_type=32, ns=1200, nl=1200,
nb=1, interleave=0
```

```
ENVI_WRITE_ENVI_FILE, albedo_white, out_name=white_out_name,
map_info=map_info, out_dt=4,$
    r_fid=albedo_white_map, sensor_type=32, ns=1200, nl=1200,
nb=1, interleave=0
```

```
;Get basic file info for both images, since they are identical
envi_file_query, albedo_black_map, ns=ns, nl=nl, nb=nb, dims=dims
pos = lindgen(nb)
```

```

out_name_albedo_black = 'c:\albedo_black_1km_4.img'
out_name_albedo_white = 'c:\albedo_white_1km_4.img'

DATUM = 'Tokyo mean'
Projection_Name2 = 'Korea - TM (Middle)'
Params2 = [6377397.2, 6356079.0, 38.000000D, 127.002890D,
200000.0, 500000.0, 1.000000]

OUT_Proj = ENVI_PROJ_CREATE(type=3, name=Projection_Name2,
datum=Datum, params=Params2)

envi_convert_file_map_projection, fid=albedo_black_map, pos=pos,
dims=dims, o_proj=OUT_Proj, $
    o_pixel_size=[1000, 1000], out_name=out_name_albedo_black,
warp_method=2, r_fid=albedo_black_TM, $
    resampling=0, background=0

envi_convert_file_map_projection, fid=albedo_white_map, pos=pos,
dims=dims, o_proj=OUT_Proj, $
    o_pixel_size=[1000, 1000], out_name=out_name_albedo_white,
warp_method=2, r_fid=albedo_white_TM, $
    resampling=0, background=0

;Get rid of intermediary files
envi_file_mng, id=albedo_black_map, /remove, /delete
envi_file_mng, id=albedo_white_map, /remove, /delete

end

```

```

On Jul 11, 2:59 pm, kuyper <kuy...@wizard.net> wrote:
> devin.wh...@gmail.com wrote:
>> This is just an initial stab at what might be the problem, but I
>> noticed that you're defining your albedo arrays as floating point:
>
>>   albedo_black = fltarr(1200, 1200)
>>   albedo_white = fltarr(1200, 1200)
>
>> MOD43B3 albedo is stored as integer, which is half the size.
>
> As a result, we have:
>
> IDL> HELP,albedo
> ALBEDO      INT      = Array[2, 10, 1200, 1200]
> IDL> help,albedo_black
> ALBEDO_BLACK  FLOAT   = Array[1200, 1200]
>

```

> However, an automatic conversion between INT and FLOAT occurs on the
> following line:
>
> albedo_black[:,*]=albedo[0, 9, *,*]
>
>> Your
>> data are most likely getting written into the file, but ENVI is
>> expecting them to take up more room.
>
> The INT array 'albedo' is never passed to ENVI. Only the albedo_black
> and albedo_white arrays are passed to ENVI_WRITE_ENVI_FILE, and those
> are both float arrays. In any event, ENVI_WRITE_ENVI_FILE is supposed
> to convert it's input array into whichever data type is specified by
> the OUT_DT parameter. In this case, that parameter is set to 4, so no
> conversion is needed.
>
> Whatever the problem is, that can't be the cause..
>
> ...
>
>> retrieve). You don't get floating point output unless you take the
>> four-dimensional albedo array and apply the necessary scale and offset
>> factors stored in the SD. Just a thought.
>
> The scale_factor and add_offset do need to be applied to calculate the
> correct albedo, as indicated by the filespec:
>
>> Data conversions:
>> file data =(Albedo / scale_factor) + add_offset
>> Albedo =(file data - add_offset) * scale_factor
>
> These values may be read from the file using code like the following:
>
> idx_scale = HDF_SD_AttrFind(sdsID_albedo,'scale_factor')
> HDF_SD_AttrInfo, sdsID_albedo, idx_scale, DATA=scale_factor
> idx_offset = HDF_SD_AttrFind(sdsID_albedo, 'add_offset')
> HDF_SD_AttrInfo, sdsID_albedo, idx_offset, DATA=add_offset
>
> However, the problems he's been having don't seem to be related to the
> values stored in his variables. The message said that the problem was
> that something had a dimension of 0. Whether or not he applied the
> scale factor and offset shouldn't have any affect on that problem.
