Subject: HDF_SD_ADDDATA problem
Posted by adfraser on Mon, 21 Apr 2008 23:24:17 GMT
View Forum Message <> Reply to Message

Dear Everyone,

I'm having trouble writing to an HDF file using HDF_SD_ADDDATA, but let me set the scene quickly first.

I'm using a multispectral satellite imagery dataset in its native L1B (HDF-EOS, it's MODIS for those who care!) format. When ordering the data, one has the option to channel subset the data to reduce size.

I've written IDL programs to cloud mask the data, which involves manipulating an SDS within the HDF. My programs all work fine using the full, un-subsetted data, but whenever I try them on the subsetted HDFs, they fall over with this error.

```
 ------------------------------------------------------------ -----------
% HDF_SD_ADDDATA: Unable to write the specified HDF-SD slice.
% Execution halted at: HDFEDITSDS        28 /array/work/adfraser/
MODIS
   images/18_4_08 processing of winter2 data/output/HDFeditSDS.pro
%                    $MAIN$
 ------------------------------------------------------------ -----------
```

Here is the code which I've been using to write successfully to the unsubsetted data but unsuccessfully to the subsetted data:

```
 ------------------------------------------------------------ ------------
PRO HDFeditSDS, filename, sdsname, newdata

sdfileid=hdf_sd_start(filename, /rdwr)

; Find the index of the "Gridded Data" SDS.
index = HDF_SD_NAMETOINDEX(sdFileID, sdsname)

; Select the Gridded Data SDS.
thisSdsID = HDF_SD_SELECT(sdFileID, index)

HDF_SD_AddData, thisSdsID, newdata

hdf_sd_end, sdfileid

END
 ------------------------------------------------------------ -------------
```

I've checked all the obvously stupid things (file access permissions
are identical, and the SDS name exists within the HDF). I'm at a loss,
please help!

Cheers,
Alex.

Alex Fraser
Ph.D. candidate
University of Tasmania
Antarctic Climate and Ecosystems Cooperative Research Centre
Sandy Bay Campus
Tasmania 7001

Subject: Re: HDF_SD_ADDDATA problem
Posted by jameskuyper on Thu, 24 Apr 2008 13:00:58 GMT

View Forum Message <> Reply to Message

adfra...@utas.edu.au wrote:
> Dear Everyone,
>
> I'm having trouble writing to an HDF file using HDF_SD_ADDDATA, but
> let me set the scene quickly first.
>
> I'm using a multispectral satellite imagery dataset in its native L1B
> (HDF-EOS, it's MODIS for those who care!) format. When ordering the
> data, one has the option to channel subset the data to reduce size.

I'm responsible for the code which creates the original L1B files, and
also for the code which performs the channel subsetting. Therefore I'm
very interested in resolving this problem.

> I've written IDL programs to cloud mask the data, which involves
> manipulating an SDS within the HDF. My programs all work fine using
> the full, un-subsetted data, but whenever I try them on the subsetted
> HDFs, they fall over with this error.
>
> -- -----------
> % HDF_SD_ADDDATA: Unable to write the specified HDF-SD slice.
> % Execution halted at: HDFEDITSDS   28 /array/work/adfraser/
> MODIS
>   images/18_4_08 processing of winter2 data/output/HDFeditSDS.pro
> % $MAIN$
> -- -----------
>
> Here is the code which I've been using to write successfully to the

```
>  unsubsetted data but unsuccessfully to the subsetted data:
>
>   ------------------------------------------------------------ ------------
>  PRO HDFeditSDS, filename, sdsname, newdata
>
>  sdfileid=hdf_sd_start(filename, /rdwr)
>
>  ; Find the index of the "Gridded Data" SDS.
>  index = HDF_SD_NAMETOINDEX(sdFileID, sdsname)
```

What's the actual value of sdsname in the cases where this fails?

The native MODIS L1B files are in HDFEOS Swath format, not Grid format. They contain no SDSs that are named as, or can accurately be described as, "Gridded Data". Did you select reprojection as a post-processing option when you ordered the data? That would created gridded data. I'm not responsible for the code which performs the reprojection, but I know the guy who is.

The problem might depend upon some post-processing option you selected which you haven't mentioned yet. If you don't know your order number, please identify precisely all of the post-processing options you selected when you ordered this data. The problem might depend one of those other options. If you do know your order number, we can find out precisely what options you selected, we don't have to rely upon you remembering it.

```
>  ; Select the Gridded Data SDS.
>  thisSdsID = HDF_SD_SELECT(sdFileID, index)
>
>  HDF_SD_AddData, thisSdsID, newdata
>
>  hdf_sd_end, sdfileid
>
>  END
>   ------------------------------------------------------------ -------------
>
>
>  I've checked all the obvously stupid things (file access permissions
>  are identical, and the SDS name exists within the HDF). I'm at a loss,
>  please help!
```

Well, the obvious question is, what is the type and and what are the dimensions of newdata? How do they compare with the type and dimensions of the SDS you're trying to add it to? Channel subsetting will reduce the size (and change the meaning of) the band dimension of any SDS which has a band dimension containing the bands which you didn't select.

Subject: Re: HDF_SD_ADDDATA problem
Posted by brodzik@nsidc.org on Thu, 24 Apr 2008 14:25:39 GMT
View Forum Message <> Reply to Message

On Apr 23, 8:17 pm, adfra...@utas.edu.au wrote:

> Also, I hadn't used the EOS_GD interface before, I must check that out
> one day. It looks interesting, is a part of the IDL standard
> distribution?
>

AFAIK, yes.   It's the IDL layer that handles the HDF-EOS Grid
abstraction.

Mary Jo

Subject: Re: HDF_SD_ADDDATA problem
Posted by jameskuyper on Fri, 25 Apr 2008 16:15:05 GMT
View Forum Message <> Reply to Message

adfra...@utas.edu.au wrote:

```
> Dear Everyone,
>
> I'm having trouble writing to an HDF file using HDF_SD_ADDDATA, but
> let me set the scene quickly first.
>
> I'm using a multispectral satellite imagery dataset in its native L1B
> (HDF-EOS, it's MODIS for those who care!) format. When ordering the
> data, one has the option to channel subset the data to reduce size.
>
> I've written IDL programs to cloud mask the data, which involves
> manipulating an SDS within the HDF. My programs all work fine using
> the full, un-subsetted data, but whenever I try them on the subsetted
> HDFs, they fall over with this error.
>
>  ------------------------------------------------------------ -----------
> % HDF_SD_ADDDATA: Unable to write the specified HDF-SD slice.
> % Execution halted at: HDFEDITSDS        28 /array/work/adfraser/
> MODIS
>   images/18_4_08 processing of winter2 data/output/HDFeditSDS.pro
> %                    $MAIN$
>  ------------------------------------------------------------ -----------
>
> Here is the code which I've been using to write successfully to the
> unsubsetted data but unsuccessfully to the subsetted data:
>
>  ------------------------------------------------------------ ------------
>  PRO HDFeditSDS, filename, sdsname, newdata
>
>  sdfileid=hdf_sd_start(filename, /rdwr)
>
>  ; Find the index of the "Gridded Data" SDS.
>  index = HDF_SD_NAMETOINDEX(sdFileID, sdsname)
>
>  ; Select the Gridded Data SDS.
>  thisSdsID = HDF_SD_SELECT(sdFileID, index)
>
>  HDF_SD_AddData, thisSdsID, newdata
>
>  hdf_sd_end, sdfileid
>
>  END
>   ---------------------------------------------------------- -------------
>
>
>  I've checked all the obvously stupid things (file access permissions
>  are identical, and the SDS name exists within the HDF). I'm at a loss,
>  please help!
```

SInce you were talking about Gridded data, I presume that you ordered the data with reprojection. I placed an order for MOD021KM for 2008-04-23 08:40, with channel subsetting for bands 5, 6, and 7, and with reprojection to a grid at 1km resolution in an azimulhal equal-area projection centered at 40E 35N. I used

```
filename = $
 'MOD021KM.A2008114.0840.005.2008114172926.bsrpcs_00050029480
8.EV_500_Aggr1km_RefSB_1_1.hdf'
sdsname = 'EV_500_Aggr1km_RefSB_0'
```

I use exactly the code you provided above, except that I added the required call to HDF_SD_EndAccess. I filled in newdata by executing the following commands:

```
HDF_SD_GetData, thisSdsID, olddata
newdata = olddata+500
```

Everything worked perfectly. Therefore, whatever problem you're running into depends upon some of the details you haven't given us. Could you provide some more details about how you triggered this problem?

Subject: Re: HDF_SD_ADDDATA problem

Posted by liamgumley on Fri, 25 Apr 2008 16:18:25 GMT
View Forum Message <> Reply to Message

On Apr 21, 6:24 pm, adfra...@utas.edu.au wrote:
[stuff deleted]
> I've checked all the obvously stupid things (file access permissions
> are identical, and the SDS name exists within the HDF). I'm at a loss,
> please help!

Alex,

This may or may not be related to your problem, but it's worth a shot.

MODIS HDF4 files obtained from LAADS are normally compressed via HDF internal compression. It is not possible to write to an internally compressed SDS. Assuming you have the HDF4 binaries from

http://www.hdfgroup.org/products/hdf4_tools/

then the hrepack command can be used to remove HDF4 internal compression:

```
hrepack -i input_file -o output_file -t '*:NONE'
```

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/
Proudly sold in paperback form ONLY at all fine bookstores (and one or two ordinary ones as well).

Subject: Re: HDF_SD_ADDDATA problem
Posted by adfraser on Mon, 28 Apr 2008 01:05:57 GMT
View Forum Message <> Reply to Message

Hi James,
Thanks for your great comments. The "gridded data" referred to in the comments is an outdated comment, sorry for misleading you.

I'm talking about MODIS L1B data in its unsubsetted form (the routine works for these data) and its channel subsetted form (the routine doesn't work for these data). So no reprojection was ordered as part of the post processing, only channel subsetting.

Interesting that the code worked for your reprojected and channel subsetted data (I've since modified my code to include the endaccess call, but the error is still present). I've tried the code on two completely different systems (one windows, one unix, both the same IDL version, however), and I get identical errors on both. I guess it is the reprojection of the data which somehow makes the data fundamentally different so that it works with the provided routine. Maybe try the test again without reprojection?

As per your other question, the type of newdata is the same as olddata, and the dimensions are the same also (however, I've checked with the unsubsetted data, and you are able to write data with different dimensions than the existing array). For my testing, I've just been writing a smaller vector e.g., [1,2,3,4] to the existing SDS, which replaces the first 4 elements in the unsubsetted HDF, but causes the error in the subsetted HDF.

However, I may have uncovered the source of the error!

Is it possible that somehow the act of subsetting the data somehow subtly makes the file not fully HDF compliant? I mean that the new data are still HDF-EOS compliant, but not actually HDF compliant. And, of course, I've been using IDL's HDF routines to edit these files, whereas I should possibly have been using EOS_SW_WRITEFIELD (http:// www.astro.princeton.edu/~esirko/idl_html_help/EOS-routines13 4.html). What are your thoughts on this?

Cheers,
Alex.

Subject: Re: HDF_SD_ADDDATA problem
Posted by adfraser on Mon, 28 Apr 2008 01:08:32 GMT
View Forum Message <> Reply to Message

Thanks, Liam (nice book, by the way).

I haven't yet investigated this, but I'd hope that the IDL routines are smart enough to decompress before opening the files! Actually I'm pretty sure they do - I can read the SDS data fine using IDL. But you're right, writing may be a different story. I'll check it out.

So LAADS MODIS HDFs are HDF4, not HDF5? That's interesting.

Thanks again,
Alex.

Subject: Re: HDF_SD_ADDDATA problem
Posted by jameskuyper on Mon, 28 Apr 2008 02:52:31 GMT

View Forum Message <> Reply to Message

adfraser@utas.edu.au wrote:

> Thanks, Liam (nice book, by the way).
>
> I haven't yet investigated this, but I'd hope that the IDL routines
> are smart enough to decompress before opening the files! Actually I'm
> pretty sure they do - I can read the SDS data fine using IDL. But
> you're right, writing may be a different story. I'll check it out.

I believe that they decompress the data sets in memory; but the original data is still in the file in compressed format, and that's what causes the problem. The only way you can write to a compressed SDS is if it defined as having an unlimited dimension, and the write only appends data at the end, rather than replacing data within the SDS.

> So LAADS MODIS HDFs are HDF4, not HDF5? That's interesting.

The MODIS project standardized on HDF4 before HDF5 ever came out. HDF4 wasn't even stable yet; there's traces of bizarre workaround in some of our code for bugs that HDF4 no longer has.

When HDF5 did come out, it was decided that with a planned instrument lifetime of 5 years, it wasn't worthwhile going to the trouble of switching MODIS products to HDF5. The MODIS instruments launched later than originally planned :-(but have lasted longer than planned :-), so the decision looks a little less reasonable now than it did at the time. Hindsight is 20/20.

Subject: Re: HDF_SD_ADDDATA problem
Posted by jameskuyper on Mon, 28 Apr 2008 23:07:27 GMT

View Forum Message <> Reply to Message

adfra...@utas.edu.au wrote:

> Hi James,
> Thanks for your great comments. The "gridded data" referred to in the
> comments is an outdated comment, sorry for misleading you.
>
> I'm talking about MODIS L1B data in its unsubsetted form (the routine
> works for these data) and its channel subsetted form (the routine
> doesn't work for these data). So no reprojection was ordered as part
> of the post processing, only channel subsetting.
>
> Interesting that the code worked for your reprojected and channel
> subsetted data (I've since modified my code to include the endaccess
> call, but the error is still present). I've tried the code on two
> completely different systems (one windows, one unix, both the same IDL

> version, however), and I get identical errors on both. I guess it is
> the reprojection of the data which somehow makes the data
> fundamentally different so that it works with the provided routine.
> Maybe try the test again without reprojection?

I've tried it, and as long as I was re-writing the entire SDS, everything worked.

> As per your other question, the type of newdata is the same as
> olddata, and the dimensions are the same also (however, I've checked
> with the unsubsetted data, and you are able to write data with
> different dimensions than the existing array). For my testing, I've
> just been writing a smaller vector e.g., [1,2,3,4] to the existing
> SDS, which replaces the first 4 elements in the unsubsetted HDF, but
> causes the error in the subsetted HDF.

OK, now that's important information. I was able to duplicate your error message when overwriting only a portion of the SDS to the subset file. It worked fine when using the original file rather than the subset file. The original file was internally compressed and chunked, while the subset file was not. I don't know any reason why something like this should work on a compressed file and fail on an uncompressed file; if anything I'd have expected precisely the opposite pattern. I decompressed the original file, and removed the the chunking using

```
hrepack -i input_filename -o output_filename -t "*:NONE" and -c
"*:NONE"
```

and it still succeeds, so apparently the compression/chunking issue is a red herring.

These symptoms make no sense to me, but at least I can reproduce them. I'll continue my investigation, and get back to you if I can figure anything out.

However, I should say that making sure that our data can be overwritten is not a high priority; if anything, we might want to discourage it. Not simply as a matter of protecting the integrity of our data, but also to avoid confusing the original data with the modified data. I would strongly recommend writing the modified data to a new file, rather than overwriting the existing one, and inserting metadata where appropriate to document the modifications you've performed.

> However, I may have uncovered the source of the error!
>
> Is it possible that somehow the act of subsetting the data somehow
> subtly makes the file not fully HDF compliant? ...

That should not be the case. We use exclusively standard HDF and HDF-EOS C library functions to create these files. The HDF-EOS library is built on top of HDF, it uses HDF routines for all actual file I/O. If the resulting files are not HDF "compliant", then the cause should be a defect in the HDF library (if only because it failed to warn us about erroneous inputs from our code). However, I won't know for sure until I've investigated further.

> ... I mean that the new
> data are still HDF-EOS compliant, but not actually HDF compliant. And,
> of course, I've been using IDL's HDF routines to edit these files,
> whereas I should possibly have been using EOS_SW_WRITEFIELD (http://
> www.astro.princeton.edu/~esirko/idl_html_help/EOS-routines13 4.html).
> What are your thoughts on this?

It's supposed to be perfectly feasible to use the underlying HDF routines to read and write the data fields for an HDF-EOS swath, once the swath itself has been created with HDFEOS function calls. I'd expect you to get the same results with either interface, but if you find that the HDFEOS routine gives better results, please let me know.

Subject: Re: HDF_SD_ADDDATA problem
Posted by adfraser on Wed, 30 Apr 2008 05:57:17 GMT
View Forum Message <> Reply to Message

Thanks again, James, for another interesting reply.

I rewrote the above code to use only HDF-EOS calls instead of HDF_SD calls. You were right - the operation still fails (though instead of an error message, I now get a "-1" returned when using EOS_SW_WRITEFIELD, indicating a failure to write). Again, the new code works perfectly on the un-subsetted files.

Thanks for checking out the compression stuff too. You're right - very odd that the compressed data is the stuff that can be written too properly. A red herring, like you say.

So, you are able to write back to the subsetted file (only) when writing back something the same size as the full SDS? I wonder why I'm unable to do this.

If it's alright with you, can I contact you via email and send you one of my subsetted files I'm having trouble with?

Thanks a million,
Alex.

Subject: Re: HDF_SD_ADDDATA problem
Posted by adfraser on Wed, 30 Apr 2008 06:43:34 GMT

View Forum Message <> Reply to Message

Just an update: I ran the original code, line by line from the IDL
command line, but substituted in your two lines

```
HDF_SD_GetData, thisSdsID, olddata
newdata = olddata+500
```

instead of how I was creating my new data. And it worked!

I went back to my original program. Here's some stats about the "old"
and "new" data (the second line produced by "help, L1B_emiss, /stru"):

```
olddata:
Max: 9931, Min: 2969, Mean: 5842.88
L1B_EMISS       UINT      = Array[1354, 2030, 2]

newdata:
Max: 9931, Min: 0, Mean: 718.132
L1B_EMISS       UINT      = Array[1354, 2030, 2]
```

The dimensionality, size and type of the arrays is identical. No NaNs
are present (otherwise I couldn't have calculated the mean). But
still, I get the "slice" error! Could the HDF interface possibly be
complaining about the presence of zeros in the new data set?!?!

I also tried creating blank int arrays of the same dimensions. Still
no go.

Any more ideas?! I'm at a loss!

Thanks,
Alex.

Subject: Re: HDF_SD_ADDDATA problem
Posted by jameskuyper on Wed, 30 Apr 2008 11:13:39 GMT

View Forum Message <> Reply to Message

adfraser@utas.edu.au wrote:
> Thanks again, James, for another interesting reply.
>
> I rewrote the above code to use only HDF-EOS calls instead of HDF_SD
> calls. You were right - the operation still fails (though instead of
> an error message, I now get a "-1" returned when using

> EOS_SW_WRITEFIELD, indicating a failure to write). Again, the new code
> works perfectly on the un-subsetted files.
>
> Thanks for checking out the compression stuff too. You're right - very
> odd that the compressed data is the stuff that can be written too
> properly. A red herring, like you say.
>
> So, you are able to write back to the subsetted file (only) when
> writing back something the same size as the full SDS? I wonder why I'm
> unable to do this.
>
> If it's alright with you, can I contact you via email and send you one
> of my subsetted files I'm having trouble with?

You can use my work e-mail, james.r.kuyper@nasa.gov, since this is definitely work-related. However, I don't think I need your subsetted file; I can duplicate the problem with the files I already possess.

Subject: Re: HDF_SD_ADDDATA problem
Posted by jameskuyper on Wed, 30 Apr 2008 11:20:11 GMT
View Forum Message <> Reply to Message

adfraser@utas.edu.au wrote:
> Just an update: I ran the original code, line by line from the IDL
> command line, but substituted in your two lines
>
> HDF_SD_GetData, thisSdsID, olddata
> newdata = olddata+500
>
> instead of how I was creating my new data. And it worked!
>
> I went back to my original program. Here's some stats about the "old"
> and "new" data (the second line produced by "help, L1B_emiss, /stru"):
>
> olddata:
> Max: 9931, Min: 2969, Mean: 5842.88
> L1B_EMISS   UINT   = Array[1354, 2030, 2]
>
> newdata:
> Max: 9931, Min: 0, Mean: 718.132
> L1B_EMISS   UINT   = Array[1354, 2030, 2]
>
>
> The dimensionality, size and type of the arrays is identical. No NaNs
> are present (otherwise I couldn't have calculated the mean). But
> still, I get the "slice" error! Could the HDF interface possibly be
> complaining about the presence of zeros in the new data set?!?!

>
> I also tried creating blank int arrays of the same dimensions. Still
> no go.

That is bizarre!

The only idea I've been able to come up with is to see if I can duplicate the problem using the HDF C libraries, rather than working through IDL. If so, then it's an HDF issue, and I'll check with the HDF help desk. If not, it's an issue with the IDL wrappers for the HDF functions, and I'll have to talk to IITVS.

Subject: Re: HDF_SD_ADDDATA problem
Posted by adfraser on Fri, 02 May 2008 02:05:51 GMT
View Forum Message <> Reply to Message

Hello again,

Good news: I've discovered a workaround! Inexplicably, if you use getdata to get the data and adddata to add back the data BETWEEN THE ONE SET OF HDF_SD_START and HDF_SD_END calls, then this seems to "unlock" the SDS, meaning you can then use the above code (HDFeditSDS) to write to the data as normal (by normal, I mean as if it were an unsubsetted MODIS L1B file).

Here's the code:

```
=======================================
FUNCTION HDFunlockSDS, filename, sdsname

sdfileid=hdf_sd_start(filename, /rdwr)

index = HDF_SD_NAMETOINDEX(sdFileID, sdsname)

thisSdsID = HDF_SD_SELECT(sdFileID, index)

HDF_SD_GETDATA, thisSdsID, sds

HDF_SD_AddData, thisSdsID, sds

HDF_SD_ENDACCESS, thisSdsID

hdf_sd_end, sdfileid

return, sds

END
```

=======================================

Any thoughts, James or anyone else? I have no idea why this works, but it does!

Cheers,
Alex.