
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 comments is an outdated comment, sorry for misleading you.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-routines134.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.
