
Subject: Re: Testing if HDF5 file is corrupt

Posted by [Michael Galloy](#) on Mon, 07 Jun 2010 16:22:41 GMT

[View Forum Message](#) <> [Reply to Message](#)

On 6/7/10 9:38 AM, rjp23@le.ac.uk wrote:

```
> I'm dealing with a large amount of satellite data in HDF5 format.
>
> I use the h5_parse command to read the data in and this all works fine
> apart from in a few instances one of the files seems to be corrupt and
> I get:
>
> IDL> test=h5_parse('examplefilename10000', /read_data)
> % Compiled module: H5_PARSE.
> % H5F_OPEN: unable to open file: Object Name:"examplefilename10000"
> % Error occurred at: H5_PARSE      553 /usr/local/itt/idl70/lib/
> h5_parse.pro
> %          $MAIN$
>
> I tried using H5F_IS_HDF5 but this returns 1 for all the files,
> regardless of the above. Is there another way to test that the file is
> readable by h5_parse?
>
> My code processes of lots of files in a batch and if the file can't
> be opened by h5_parse, I just want to skip that file.
```

If HDF5_IS_HDF5 isn't catching it, then I guess I would just make a wrapper around H5_PARSE that uses CATCH to catch this error, like:

```
function mg_h5_parse, filename, error=error
  compile_opt strictarr

  catch, error
  if (error eq 0L) then begin
    catch, /cancel
    return, -1
  endif

  return, h5_parse(filename, /read_data)
end
```

I didn't have an invalid .h5 file, so I didn't test this code at all (seeing as that its over 10 lines long, there are probably at least 2-3 errors in it). But it's not supposed to crash and should have a non-zero valid for the ERROR keyword if it couldn't parse the given .h5 file. So then you should be able to do something like:

```
for i = 0L, nfilenames - 1L do begin
  result = mg_h5_parse(filename[i], error=error)
```

```
if (error eq 0L) then begin
    ; process result how you see fit
endif
endfor
```

Mike

--

www.michaelgalloy.com
Research Mathematician
Tech-X Corporation
