Subject: Re: HDF5 - Group or Dataset?
Posted by James Kuyper on Wed, 05 Oct 2005 14:56:59 GMT
View Forum Message <> Reply to Message

Peter Albert wrote:
> Hi everybody,
>
> I am again troubled by how IDL deals with HDF5. The problem is that I'd
> like to check whether a given name is the name of a dataset within an
> HDF5 file or the name of a group. Before IDL 6.1 a statement like
>
> n_members = h5g_get_nmembers(file_id, "/aaa/bbb")
>
> nicely returned 0 if /aaa/bbb was a dataset and the number of the
> group's members (> 0) otherwise.

What if it's a group that happens for some reason to have no members?
That's not a good way to test for distinguishing groups from
non-groups.

> Now, with IDL6.1, I get the error message
>
> % H5G_GET_NMEMBERS: unable to open group: (67108874, "/aaa/bbb") if I
> am actually looking at a dataset.
>
> Well, I looked through all those H5* routines in the IDL documentation,
> but did not find something like h5g_is_group. I would really like to
> solve this using error catching, so I would highly appreciate if
> someone has an idea.

Well, you can use H5G_OPEN(); if it succeeds, the name you provided is
the name of a group. Of course, the disadvantage is that you'll have to
call H5G_CLOSE().

Subject: Re: HDF5 - Group or Dataset?
Posted by peter.albert@gmx.de on Thu, 06 Oct 2005 06:31:05 GMT
View Forum Message <> Reply to Message

> What if it's a group that happens for some reason to have no members?
> That's not a good way to test for distinguishing groups from
> non-groups.

Well, that's right, of course.

> Well, you can use H5G_OPEN(); if it succeeds, the name you provided is
> the name of a group. Of course, the disadvantage is that you'll have to
> call H5G_CLOSE().

If you try to open a dataset with H5G_OPEN(), IDL also drops an error message. And waiting for errors to occur is also not a nice way for solving this.

Well, I thought that there is no easy way, thus I decided to write a small routine which I named H5O_is_group(). It recursively checks if the parent object of the object in question is a group and then loops over all its members. If one of the member names equals the name of the given object, it uses h5g_get_objinfo() for distinguishing groups from non_groups.

Regards,

Peter

```
+
; NAME: h5o_is_group
;
;
;
; PURPOSE: Check, whether a given object is a group or
;          a dataset within an HDF5 file
;
;
;
; CALLING SEQUENCE: result = h5o_is_group(filename, name)
;
;
;
; INPUTS: filename_or_id: Filename of a HDF5 file or file_id of opened
;                         HDF file
;         name: Name of an object
;
;
;
; OUTPUTS: result is 1 if name is the name of a group, 0 otherwise
;
;
; MODIFICATION HISTORY:
; written by Peter Albert, 05.10.2005
;
;-

function h5o_is_group, filename_or_id, name
```

```
  if name eq "/" then return, 1

; Find the base level of the given name

  elements = strsplit(name, "/", /extract, count = c)
  base_level = c ge 2 $
     ? "/" + strjoin(elements[0:c-2], "/") $
     : "/"

; Recursively check, if base level is a valid group, if not, return 0

  if h5o_is_group(filename_or_id, base_level) then begin

; Check whether the HDF file is already opened

     file_id = size(filename_or_id, /type) eq 7 $
        ? h5f_open(filename_or_id) $
        : filename_or_id

; Loop over all members of the base level

     n = h5g_get_nmembers(file_id, base_level)
     is_group = 0
     for i = 0, n-1 do begin
        member_name = h5g_get_member_name(file_id, base_level, i)

; If we found the given object, then check whether it is a group

        if member_name eq elements[c-1] then begin
           group_id = h5g_open(file_id, base_level)
           is_group = (h5g_get_objinfo( $
                                 group_id, $
                                 member_name $
                               ) $
                    ).type eq "GROUP"
           h5g_close, group_id
        endif
     endfor

; Close HDF file, if necessary

     if size(filename_or_id, /type) eq 7 then h5f_close, file_id
     return, is_group

; If the upper level is no valid group, we can't actually check,
; but the given name is for sure neither a valid group or something.

  endif else return, 0
```

end