Subject: H5D_Read on compound type (GDL)
Posted by rjp23 on Thu, 24 Apr 2014 13:58:03 GMT
View Forum Message <> Reply to Message

I was hoping someone could help with this.

I'm converting some code from IDL to GDL. In IDL I normally use h5_parse but as that's not supported in GDL I'm trying to use the H5 procedures individually:

```
file_id = H5F_OPEN(file)
data_id = H5D_OPEN(file_id, '/path/to/variable/')
variable = H5D_READ(data_id)
H5D_CLOSE, data_id
```

This works fine for almost all the variables I need to read in.

However, it fails on the "time" variable as this is a "H5T_compound" type:

```
Dataset: 'Time'
H5T_COMPOUND (12 bytes)
10 elements
```

When trying this I get:

```
% H5D_READ: Read failed
```

Is there another way I should be trying to read this in?

Cheers

Rob

Subject: Re: H5D_Read on compound type (GDL)
Posted by markb77 on Thu, 24 Apr 2014 17:42:08 GMT
View Forum Message <> Reply to Message

I tested this on IDL 8.3, and it looks like you're reading it correctly. For a compound variable type, the command

```
var = h5d_read(data_id)
```

should simply read in the entire variable. I tested this by creating a compound variable in a new hdf5 file using HDFView. I was then able to read that variable back into IDL using the same command sequence you used.

Mark

Subject: Re: H5D_Read on compound type (GDL)
Posted by rjp23 on Thu, 24 Apr 2014 18:20:11 GMT
View Forum Message <> Reply to Message

Hi Mark,

Thanks for testing it. I can get it to work fine in IDL but GDL fails. I've submitted a GDL bug support but that doesn't help for now.

I wondered if there was another way to read it in that avoided using h5d_read? I'm hoping to avoid a messy solution like using python to read it in and writing that to a file that GDL can then read but I'm not sure what other options I have.

Any help appreciated.

Cheers

Rob

Subject: Re: H5D_Read on compound type (GDL)
Posted by kramers541 on Thu, 24 Apr 2014 20:54:07 GMT
View Forum Message <> Reply to Message

H5D_read is a low-level routine, I'm afraid... no way around that.