Subject: HDF SDS data attributes
Posted by William Clodius on Mon, 18 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

My team has been using HDF files to store data from a multi-spectral imaging sensor. The images are typically stored as band sequential arrays in SDS data sets. Ideally we would like to attach various information to the SDS data sets such as band names, x, y coordinate mappings, band wavelengths, etc. Some of thsi information can be readilly implemented using IDL's HDF interface but the dimension specific information such as band name and band wavelength doe not seem to be easilly implemented in IDL. The C and Fortran interfaces to HDF appear to let you define attributes associated with specific dimensions of an SDS, but not the IDL interface. Does anyone know of a workaround?

Subject: Re: HDF SDS data attributes
Posted by Liam E. Gumley on Tue, 19 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

William B. Clodius <wclodius@lanl.gov> wrote in message news:39C7A0A6.F610F1DE@lanl.gov...
>> Let me see if I understand your problem. You have a dimension named
>> 'Band_Number', which is used for image SDS arrays. However you'd also like
>> to record the actual band numbers which correspond to the 'Band_Number'
>> dimension.
>
> Not quite, we would like to record the band names, which for historical
> reasons are not always in a logical sequence. In explanation, most of
> the names are single characters in alphabetical order increasing with
> wavelength, A (blue), B (green), C (red), ..., but one (low signal) band
> is duplicated twice as a means of improving S/N (a form of time delayed
> integration) and after much of the design work was done and publicized
> an additional band was added at an intermediate wavelength without
> renaming the other bands. As a result various users of our data
> currently get confused as to whether the data are stored in alphabetical
> or wavelength order. This confusion was made worse by the implementors
> providing wrapper routines that by default returns the data in a
> different order from how they are actually stored. (If I remember
> correctly, this was sort of the fault of the rest of the team that
> decided how data was to be stored and accessed at different meetings
> without checking as to whether they were being consistent.) Soon we may
> have the question as to whether both versions of the duplicated band are
> present or only one combined image.
>
> It would also be useful for our users if the data included the center
> wavelengths and band widths for each band and directly associated that

> with the array dimension.

It seems to me you need several SDS arrays for this purpose. Let's say you have 10 bands:

```
dimensions:
  Band_Number = 10

 variables:
  long Band_Number(Band_Number)
  float Band_Center(Band_Number)
  float Band_Width(Band_Number)
  char Band_Name

 data:
  Band_Number = 1, 2, 3, 3, 4, 5, 6, 7, 8, 9
    long_name: 'Band Numerical Identifier'
  Band_Center = 0.40, 0.45, 0.50, 0.50, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85
    long_name: 'Band Central Wavelength'
    units: 'microns'
  Band_Width = 0.05, 0.05, 0.05, 0.05, 0.05, 0.02, 0.02, 0.02, 0.05, 0.05
    long_name: 'Band Width'
    units: 'microns'
  Band_Name = 'A, B, C, C, D, E, F, G, H, I'
```

Band_Name is a comma-delimited string because I don't believe HDF supports arrays of strings.

As you said, you can only have one coordinate variable (aka dimension scale) per dimension. I prefer the multiple-array approach over having multiple non-standard attributes for a single coordinate variable (i.e. Band_Number with attributes Band_Center, Band_Width, Band_Name).

Cheers,
Liam.
http://cimss.ssec.wisc.edu/~gumley

Subject: Re: HDF SDS data attributes
Posted by William Clodius on Tue, 19 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Liam Gumley wrote:
>
> William B. Clodius <wclodius@lanl.gov> wrote in message
> news:39C6A8DC.6C0F3BA1@lanl.gov...
>> My team has been using HDF files to store data from a multi-spectral
>> imaging sensor. The images are typically stored as band sequential

>> arrays in SDS data sets. Ideally we would like to attach various
>> information to the SDS data sets such as band names, x, y coordinate
>> mappings, band wavelengths, etc. Some of thsi information can be
>> readilly implemented using IDL's HDF interface but the dimension
>> specific information such as band name and band wavelength doe not seem
>> to be easilly implemented in IDL. The C and Fortran interfaces to HDF
>> appear to let you define attributes associated with specific dimensions
>> of an SDS, but not the IDL interface. Does anyone know of a workaround?

I should note that I have the July 10, 1996 documentation for HDF (4.0?)
and that I apparently misread part of it in posting my message. Perhaps
because the following examples had dim_id defined I was under the
impression that SDsetattr could take the dimension as an argument.
Further review of the documentation suggests that I spaced out. I should
also note that I am not one of the ones in charge of generating the HDF
files, although I have provided significant input on the file structure.
t?)
A significant nuisance is that our users often use ENVI to access data,
and while much of our file structure could be nominally improved by
relying on VDATAs ENVI has (almost?) no understanding of VDATAs. As a
result our files are much flatter than my ideal. ENVI also seems to make
it awkward to access vector HDF data. It is wonderful for two or three
dimensional data though, and 99.9% of the time thats what the users are
only interested in.

>
> Let me see if I understand your problem. You have a dimension named
> 'Band_Number', which is used for image SDS arrays. However you'd also like
> to record the actual band numbers which correspond to the 'Band_Number'
> dimension.

Not quite, we would like to record the band names, which for historical
reasons are not always in a logical sequence. In explanation, most of
the names are single characters in alphabetical order increasing with
wavelength, A (blue), B (green), C (red), ..., but one (low signal) band
is duplicated twice as a means of improving S/N (a form of time delayed
integration) and after much of the design work was done and publicized
an additional band was added at an intermediate wavelength without
renaming the other bands. As a result various users of our data
currently get confused as to whether the data are stored in alphabetical
or wavelength order. This confusion was made worse by the implementors
providing wrapper routines that by default returns the data in a
different order from how they are actually stored. (If I remember
correctly, this was sort of the fault of the rest of the team that
decided how data was to be stored and accessed at different meetings
without checking as to whether they were being consistent.) Soon we may
have the question as to whether both versions of the duplicated band are
present or only one combined image.

It would also be useful for our users if the data included the center wavelengths and band widths for each band and directly associated that with the array dimension.

>
> In the HDF world, you can create an SDS known as a 'dimension scale', which
> assigns values to a dimension (in the netCDF world, this type of variable is
> known as a coordinate variable). Dimension scales encode the 'values' of a
> dimension, and attributes may be added as well (e.g. 'units'. For example,
> if you stored atmospheric profiles using a 'Pressure_Level' dimension, you
> might want to also store the pressure values at each level. In this case you
> would create a dimension scale named 'Pressure_Level' which contained the
> pressure values, e.g.
>
> float Pressure(Pressure) ;
> Pressure:long_name = "Pressure Levels for Atmospheric Profiles" ;
> Pressure:units = "hPa" ;
>
> Here's an example from IDL 5.3 for Windows. The trick when creating the
> dimension scale is to use the desired dimension name as the SDS name:

The problems I saw with dimension scale were

1. You can only have one scale per dimension, i.e., band wavelength or width or name, but not all three.

2. It wasn't clear that the dimension scale could be a vector of character strings. From my limited experience users remember names better than wavelengths.

>
> ;-----
> PRO TESTHDF
>
> ;- Create the file
> hdfid = hdf_sd_start('test.hdf', /create)
>
> ;- Create the profile SDS
> varid = hdf_sd_create(hdfid, 'Profile', [10], /float)
> dimid = hdf_sd_dimgetid(varid, 0)
> hdf_sd_dimset, dimid, name='Pressure'
> hdf_sd_adddata, varid, findgen(10)
> hdf_sd_endaccess, varid
>
> ;- Check for coordinate variable
> index = hdf_sd_nametoindex(hdfid, 'Profile')
> varid = hdf_sd_select(hdfid, index)

```
>  print, hdf_sd_iscoordvar(varid)
>  hdf_sd_endaccess, varid
>
>  ;- Create the pressure coordinate variable (aka dimension scale)
>  varid = hdf_sd_create(hdfid, 'Pressure', [10], /float)
>  dimid = hdf_sd_dimgetid(varid, 0)
>  hdf_sd_dimset, dimid, name='Pressure'
>  hdf_sd_adddata, varid, (findgen(10) + 1.0) * 100.0
>  hdf_sd_endaccess, varid
>
>  ;- Check for coordinate variable
>  index = hdf_sd_nametoindex(hdfid, 'Pressure')
>  varid = hdf_sd_select(hdfid, index)
>  print, hdf_sd_iscoordvar(varid)
>  hdf_sd_endaccess, varid
>
>  ;- Close the file
>  hdf_sd_end, hdfid
>
>  END
>  ;-----
>
>  You can add attributes to the dimension scale as desired. When this
>  procedure is executed, it correctly identifies the second variable
>  'Pressure' as a coordinate variable (aka dimension scale).
>
>  IDL> testhdf
>             0
>             1
>
>  Here are the contents of the file as seen by the HDF version of ncdump:
>
>  netcdf test {
>  dimensions:
>    Pressure = 10 ;
>
>  variables:
>    float Profile(Pressure) ;
>    float Pressure(Pressure) ;
>
>  data:
>
>    Profile = 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9  ;
>
>    Pressure = 100 , 200 , 300 , 400 , 500 , 600 , 700 , 800 , 900 , 1000  ;
>  }
>
>  For more information, see the HDF documentation at
```

> http://hdf.ncsa.uiuc.edu/UG41r3_html/SDS_SD.fm7.html#40381

Thanks for the reference.
>
> Cheers,
> Liam.
> http://cimss.ssec.wisc.edu/~gumley

Subject: Re: HDF SDS data attributes
Posted by Liam E. Gumley on Tue, 19 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

William B. Clodius <wclodius@lanl.gov> wrote in message
news:39C6A8DC.6C0F3BA1@lanl.gov...
> My team has been using HDF files to store data from a multi-spectral
> imaging sensor. The images are typically stored as band sequential
> arrays in SDS data sets. Ideally we would like to attach various
> information to the SDS data sets such as band names, x, y coordinate
> mappings, band wavelengths, etc. Some of thsi information can be
> readilly implemented using IDL's HDF interface but the dimension
> specific information such as band name and band wavelength doe not seem
> to be easilly implemented in IDL. The C and Fortran interfaces to HDF
> appear to let you define attributes associated with specific dimensions
> of an SDS, but not the IDL interface. Does anyone know of a workaround?

Let me see if I understand your problem. You have a dimension named 'Band_Number', which is used for image SDS arrays. However you'd also like to record the actual band numbers which correspond to the 'Band_Number' dimension.

In the HDF world, you can create an SDS known as a 'dimension scale', which assigns values to a dimension (in the netCDF world, this type of variable is known as a coordinate variable). Dimension scales encode the 'values' of a dimension, and attributes may be added as well (e.g. 'units'. For example, if you stored atmospheric profiles using a 'Pressure_Level' dimension, you might want to also store the pressure values at each level. In this case you would create a dimension scale named 'Pressure_Level' which contained the pressure values, e.g.

```
float Pressure(Pressure) ;
   Pressure:long_name = "Pressure Levels for Atmospheric Profiles" ;
   Pressure:units = "hPa" ;
```

Here's an example from IDL 5.3 for Windows. The trick when creating the dimension scale is to use the desired dimension name as the SDS name:

```
;-----
```

```
PRO TESTHDF

;- Create the file
hdfid = hdf_sd_start('test.hdf', /create)

;- Create the profile SDS
varid = hdf_sd_create(hdfid, 'Profile', [10], /float)
dimid = hdf_sd_dimgetid(varid, 0)
hdf_sd_dimset, dimid, name='Pressure'
hdf_sd_adddata, varid, findgen(10)
hdf_sd_endaccess, varid

;- Check for coordinate variable
index = hdf_sd_nametoindex(hdfid, 'Profile')
varid = hdf_sd_select(hdfid, index)
print, hdf_sd_iscoordvar(varid)
hdf_sd_endaccess, varid

;- Create the pressure coordinate variable (aka dimension scale)
varid = hdf_sd_create(hdfid, 'Pressure', [10], /float)
dimid = hdf_sd_dimgetid(varid, 0)
hdf_sd_dimset, dimid, name='Pressure'
hdf_sd_adddata, varid, (findgen(10) + 1.0) * 100.0
hdf_sd_endaccess, varid

;- Check for coordinate variable
index = hdf_sd_nametoindex(hdfid, 'Pressure')
varid = hdf_sd_select(hdfid, index)
print, hdf_sd_iscoordvar(varid)
hdf_sd_endaccess, varid

;- Close the file
hdf_sd_end, hdfid

END
;-----
```

You can add attributes to the dimension scale as desired. When this procedure is executed, it correctly identifies the second variable 'Pressure' as a coordinate variable (aka dimension scale).

```
IDL> testhdf
       0
       1
```

Here are the contents of the file as seen by the HDF version of ncdump:

```
netcdf test {
```

dimensions:
 Pressure = 10 ;

variables:
 float Profile(Pressure) ;
 float Pressure(Pressure) ;

data:

 Profile = 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 ;

 Pressure = 100 , 200 , 300 , 400 , 500 , 600 , 700 , 800 , 900 , 1000 ;
}

For more information, see the HDF documentation at
http://hdf.ncsa.uiuc.edu/UG41r3_html/SDS_SD.fm7.html#40381

Cheers,
Liam.
http://cimss.ssec.wisc.edu/~gumley
