
Subject: NCDF_ATTCOPY and typecasting
Posted by alexzcervantes@gmail.com on Fri, 18 Aug 2006 17:26:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello All,

I am newb to NetCDF and IDL so bear with me :)

Here's my situation:

I am reading in a data array of type FLOAT, and creating a new NetCDF file and writing the data array to the new file in type BYTE. I can do this fine. However, when I try to do a `ncdf_attcopy` on the variable attributes from the float array to my new byte array, it fails. I do know that when doing a `varput`, the datatype of the array and its attributes need to be the same.

Does anyone know any way around this where I can maybe do a typecast from float to byte using `ncdf_attcopy`?

Thanks in advance,
Alex

Subject: Re: NCDF_ATTCOPY and typecasting
Posted by [David Fanning](#) on Fri, 18 Aug 2006 20:34:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

`alexzcervantes@gmail.com` writes:

```
> Here is a chunk of my code:
>
> if(natts gt 0) then begin
>   FOR index = 0L, natts - 1L DO BEGIN
>     ;---Load the attribute name in attribute name array
>     maf_attname = ncdf_attname(mafncid, mafvarid, index)
>     ncdf_attget, mafncid, mafvarid, maf_attname, attval
>     ncdf_attput, cdfid, l2pvarid, maf_attname, attval, /byte
>   ENDFOR
> endif
>
> Everything seems to work except when I have to copy over strings. When
> I typecast a string to byte, I get output like this:
>
> solar_irradiance:_FillValue = -1b ;
> solar_irradiance:scale_factor = 1b ;
> solar_irradiance:add_offset = 0b ;
> solar_irradiance:valid_min = 0b ;
```

```
> solar_irradiance:valid_max = 0b ;
> solar_irradiance:long_name = 115b, 111b, 108b, 97b, 114b, 32b, 105b,
> 114b, 114b, 97b, 100b, 105b, 97b, 110b, 99b, 101b ;
```

So, what do you think is wrong?

```
IDL> long_name = [115b, 111b, 108b, 97b, 114b, 32b, 105b, 114b, $
                  114b, 97b, 100b, 105b, 97b, 110b, 99b, 101b]
IDL> print, string(long_name)
solar irradiance
```

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. (Opata Indian saying, meaning "Perhaps thou speakest truth.")

Subject: Re: NCDF_ATTCOPY and typecasting

Posted by alexzcervantes@gmail.com on Mon, 21 Aug 2006 16:40:17 GMT

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

Doh! I guess a better question would be, when using `ncdf_attget`, is there a way to determine the type of the attribute you are reading? I am thinking I need special 'if' statement to check if the attribute is of type string, then write it out as string if it is so.

-Alex

David Fanning wrote:

```
> alexzcervantes@gmail.com writes:
```

```
>
```

```
>> Here is a chunk of my code:
```

```
>>
```

```
>> if(natts gt 0) then begin
```

```
>>   FOR index = 0L, natts - 1L DO BEGIN
```

```
>>     ;----Load the attribute name in attribute name array
```

```
>>     maf_attname = ncdf_attname(mafncid, mafvarid, index)
```

```
>>     ncdf_attget, mafncid, mafvarid, maf_attname, attval
```

```
>>     ncdf_attput, cdfid, l2pvarid, maf_attname, attval, /byte
```

```
>>   ENDFOR
```

```
>> endif
```

```
>>
```

```
>> Everything seems to work except when I have to copy over strings. When
```

```
>> I typecast a string to byte, I get output like this:
>>
>> solar_irradiance:_FillValue = -1b ;
>> solar_irradiance:scale_factor = 1b ;
>> solar_irradiance:add_offset = 0b ;
>> solar_irradiance:valid_min = 0b ;
>> solar_irradiance:valid_max = 0b ;
>> solar_irradiance:long_name = 115b, 111b, 108b, 97b, 114b, 32b, 105b,
>> 114b, 114b, 97b, 100b, 105b, 97b, 110b, 99b, 101b ;
>
> So, what do you think is wrong?
>
> IDL> long_name = [115b, 111b, 108b, 97b, 114b, 32b, 105b, 114b, $
>                  114b, 97b, 100b, 105b, 97b, 110b, 99b, 101b]
> IDL> print, string(long_name)
>    solar irradiance
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui. (Opata Indian saying, meaning "Perhaps thou
> speakest truth.")
```

Subject: Re: NCDF_ATTCOPY and typecasting
Posted by [K. Bowman](#) on Mon, 21 Aug 2006 17:08:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <1156178416.968784.228920@74g2000cwt.googlegroups.com>,
"alexzcervantes@gmail.com" <alexzcervantes@gmail.com> wrote:

```
> Doh! I guess a better question would be, when using ncdf_attget, is
> there a way to determine the type of the attribute you are reading? I
> am thinking I need special 'if' statement to check if the attribute is
> of type string, then write it out as string if it is so.
```

That would be NCDF_ATTINQ.

Cheers, Ken Bowman

Subject: Re: NCDF_ATTCOPY and typecasting

Okay, after doing more researching, I read that ncdump translates byte data into readable string whenever it can. How come it won't do it in my case?

I keep getting:

```
solar_irradiance:long_name = 115b, 111b, 108b, 97b, 114b, 32b, 105b,
114b, 114b, 97b, 100b, 105b, 97b, 110b, 99b, 101b ;
```

-Alex

David Fanning wrote:

> alexzcervantes@gmail.com writes:

>

>> Here is a chunk of my code:

>>

>> if(natts gt 0) then begin

>> FOR index = 0L, natts - 1L DO BEGIN

>> ;----Load the attribute name in attribute name array

>> maf_attname = ncdf_attname(mafncid, mafvarid, index)

>> ncdf_attget, mafncid, mafvarid, maf_attname, attval

>> ncdf_attput, cdfid, l2pvarid, maf_attname, attval, /byte

>> ENDFOR

>> endif

>>

>> Everything seems to work except when I have to copy over strings. When

>> I typecast a string to byte, I get output like this:

>>

>> solar_irradiance:_FillValue = -1b ;

>> solar_irradiance:scale_factor = 1b ;

>> solar_irradiance:add_offset = 0b ;

>> solar_irradiance:valid_min = 0b ;

>> solar_irradiance:valid_max = 0b ;

>> solar_irradiance:long_name = 115b, 111b, 108b, 97b, 114b, 32b, 105b,

>> 114b, 114b, 97b, 100b, 105b, 97b, 110b, 99b, 101b ;

>

> So, what do you think is wrong?

>

> IDL> long_name = [115b, 111b, 108b, 97b, 114b, 32b, 105b, 114b, \$

> 114b, 97b, 100b, 105b, 97b, 110b, 99b, 101b]

> IDL> print, string(long_name)

> solar irradiance

>

> Cheers,

>

> David

>

> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
> Sepore ma de ni thui. (Opata Indian saying, meaning "Perhaps thou
> speakest truth.")

Subject: Re: NCDF_ATTCOPY and typecasting
Posted by [David Fanning](#) on Mon, 21 Aug 2006 17:30:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

alexzcervantes@gmail.com writes:

> Okay, after doing more researching, I read that ncdump translates byte
> data into readable string whenever it can. How come it won't do it in
> my case?

Don't know. Maybe this is one of the scenarios in which it can't. :-)

I wouldn't worry too much about it. It is quite common to store strings as byte arrays. Presumably the user can figure out he needs a string and cast the byte array to that.

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Sepore ma de ni thui. (Opata Indian saying, meaning "Perhaps thou
speakest truth.")
