
Subject: Re: NetCDF tools

Posted by [Kenneth P. Bowman](#) on Thu, 31 Mar 2011 18:48:48 GMT

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

In article

<ff693c57-2325-432f-bb05-705bd303932a@q12g2000prb.googlegroups.com>,
Ed Hyer <ejhyer@gmail.com> wrote:

> Hi guys,
>
> I'm migrating some ugly FORTRAN binary blobs used by some applications
> into shiny new NetCDF files, and I realize I've never worked with
> NetCDF in IDL before.
> My first stop, as usual, was at Coyote, and it looks like the Coyote
> NetCDF object can be used to build whatever edifice I need.
>
> But before I get to building, I wanted to inquire of the assembled: is
> there a NetCDF equivalent to Liam Gumley's fantastic SDS_READ program
> for HDF files?
> If the answer is no, I may just be tempted to port that program to
> NetCDF, because I have become so used to it.
>
> What would be absolutely fantastic is a file that vacuums up all the
> attributes and data in a NetCDF file and returns a structure with
> everything, but I don't think that could be done without lots of
> EXECUTE(), which makes the baby Cthulhu cry.
>
> --Edward H.

This function will read a variable and its attributes and return
a data structure.

It is slightly specialized for my use. I like to include the
number of elements in the 'n' field, but that can potentially
conflict with an existing attribute name.

To get all the variables in a file you would need to make a
wrapper function that will get the list of variables and
loop over them.

I have never had much use for a completely general-purpose
routine that reads an entire file, because, among other things,
the files are often larger than system memory. Adding the
capability to read just parts of a file will make the program
much more complex. At that point I find it easier to just
write the code to get what I want.

Ken Bowman

FUNCTION NCDF_READ_VAR, id, name, _EXTRA = extra

```
;  
;+  
; Name:  
;   NCDF_READ_VAR  
; Purpose:  
;   This function reads a variable and all of its attributes from an open  
;   netCDF file and returns the variable values and attributes in a structure.  
; Category:  
;   NCDF utility.  
; Calling sequence:  
;   data = NCDF_READ_VAR(id, name)  
; Input:  
;   id   : netCDF file id of the input file. The file must already be open.  
;   name : variable id or name of the variable to be read. The variable must  
;         exist in the file.  
; Output:  
;   Structure containing requested variable values and all attributes.  
; Keywords:  
;   If present, the keywords COUNT, OFFSET, and STRIDE are passed to NCDF_VARGET.  
; Author:  
;   K. Bowman. 2004-05-22.  
;-
```

COMPILE_OPT IDL2 ;Set compile options

```
IF (N_PARAMS() NE 2) THEN $ ;Check parameters  
  MESSAGE, 'You must specify a file id and variable id or name.'
```

var_info = NCDF_VARINQ(id, name) ;Get variable info

```
NCDF_VARGET, id, name, values, _EXTRA = extra ;Read data  
IF (var_info.datatype EQ 'CHAR') THEN values = STRING(values) ;If CHAR, convert to  
STRING
```

IF (var_info.natts GT 0) THEN BEGIN ;Read attributes

```
  att_name = NCDF_ATTNAME(id, var_info.name, 0) ;Get first attribute name  
  att_info = NCDF_ATTINQ(id, var_info.name, att_name) ;Get first attribute info  
  NCDF_ATTGET, id, name, att_name, att ;Read first attribute value  
  IF (att_info.datatype EQ 'CHAR') THEN att = STRING(att) ;Convert CHAR to STRING  
  atts = CREATE_STRUCT(att_name, att) ;Create attribute structure
```

```
  FOR i = 1, var_info.natts - 1 DO BEGIN ;For each attribute  
    att_name = NCDF_ATTNAME(id, var_info.name, i) ;Get attribute name
```

```

    att_info = NCDF_ATTINQ(id, var_info.name, att_name)           ;Get attribute info
    NCDF_ATTGET, id, name, att_name, att                         ;Read attribute value
    IF (att_info.datatype EQ 'CHAR') THEN att = STRING(att)      ;Convert CHAR to STRING
    IF (att_name NE 'n') THEN atts = CREATE_STRUCT(att, att_name, att) ;Add attribute to
structure, skip attributes named 'n'
    ENDFOR

    data = CREATE_STRUCT('name', var_info.name, $              ;Create structure
        'values', values, $
        'n', N_ELEMENTS(values), $
        atts)

ENDIF ELSE BEGIN

    data = CREATE_STRUCT('name', var_info.name, $              ;Create structure
        'values', values, $
        'n', N_ELEMENTS(values))

ENDELSE

RETURN, data                                           ;Return data structure

END

```
