
Subject: Re: read SCIAMACHY NO2 hdf files

Posted by [James Kuyper](#) on Mon, 02 Oct 2006 15:24:27 GMT

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

Kostasc wrote:

> hello all, first of all i would like to tell u that i have no idea
> about programming in idl. What i 'd like to ask is if it is possible to
> get from a hdf file the NO2 values for particular longitudes and
> latitudes, i need them so as to compare with ground based measurements.
> In the site i downloaded the hdf files they give the following code.
> Could please someone tell me how to export the values in an ASII file?

> pro read_scia_no2

> ;-----
> ; Read data from GOME HDF file between two times "date1" and "date2"
> ; and stores in structure "no2"
> ;
> ;"maxorbits" maximum number of orbits in a file
> ;"maxpix" maximum number of pixels in an orbit
> ;"nplevs" number of pressure levels in TEMIS SCIA data
> ;
> ; Folkert Boersma, KNMI, February 2004
> ;-----

> maxorbits = 20

> maxpix = 5000

> nplev = 35

> file = 'no2track20030417.hdf'

> ; Define structure for GOME data

> no2 = { \$

> norbits : 0, \$

> npix : intarr(maxorbits), \$

> a_lev : make_array(nplev,/float,value=-999),\$

> b_lev : make_array(nplev,/float,value=-999),\$

> date : make_array(maxorbits,maxpix,/string,value=-999), \$

> time : make_array(maxorbits,maxpix,/string,value=-999), \$

> lon : make_array(maxorbits,maxpix,/float,value=-999.9), \$

> lat : make_array(maxorbits,maxpix,/float,value=-999.9), \$

> vcd : make_array(maxorbits,maxpix,/float,value=-999.9), \$

> sigvcd : make_array(maxorbits,maxpix,/float,value=-999.9), \$

> vcdtrop : make_array(maxorbits,maxpix,/float,value=-999.9), \$

> sigvcdt : make_array(maxorbits,maxpix,/float,value=-999.9), \$

> vcdstrat : make_array(maxorbits,maxpix,/float,value=-999.9), \$

> sigvcds : make_array(maxorbits,maxpix,/float,value=-999.9), \$

> fltrop : make_array(maxorbits,maxpix,/int,value=-999), \$

```

> psurf    : make_array(maxorbits,maxpix,/float,value=-999.9), $
> sigvcdak : make_array(maxorbits,maxpix,/float,value=-999.9), $
> sigvcdtak : make_array(maxorbits,maxpix,/float,value=-999.9), $
> kernel   : make_array(maxorbits,maxpix,nplev,/float,value=-999),$
> ghostcol : make_array(maxorbits,maxpix,/float,value=-999.9), $
> sza      : make_array(maxorbits,maxpix,/float,value=-999.9),$
> vza      : make_array(maxorbits,maxpix,/float,value=-999.9),$
> raa      : make_array(maxorbits,maxpix,/float,value=-999.9),$
> ssc      : make_array(maxorbits,maxpix,/int,value=-999), $
> loncorn  : make_array(maxorbits,maxpix,4,/float,value=-999.),$
> latcorn  : make_array(maxorbits,maxpix,4,/float,value=-999.),$
>   scd     : make_array(maxorbits,maxpix,/float,value=-999.),$
> amf      : make_array(maxorbits,maxpix,/float,value=-999.),$
> amftrop  : make_array(maxorbits,maxpix,/float,value=-999.),$
> amfgeo   : make_array(maxorbits,maxpix,/float,value=-999.),$
> scdstr   : make_array(maxorbits,maxpix,/float,value=-999.),$
> clfrac   : make_array(maxorbits,maxpix,/float,value=-999.),$
> cltpres  : make_array(maxorbits,maxpix,/float,value=-999.),$
> albclr   : make_array(maxorbits,maxpix,/float,value=-999.),$
> crfrac   : make_array(maxorbits,maxpix,/float,value=-999.),$
> ltropo   : make_array(maxorbits,maxpix,/int,value=-999)}
>
>
> id = hdf_sd_start(file,/read)
> ; The HDF_SD_FILEINFO procedure retrieves the number of datasets and
> ; global attributes in an HDF file.
> hdf_sd_fileinfo,id,datasets,attributes
> help,datasets,attributes
>
> iret = 0
> for i=0,attributes-1 do begin
>   hdf_sd_attrinfo,id,i,name = name, data = d
>   command = name+'=d'
>   iret = execute(command)
>   print, 'Retrieved Attribute:',name,' ',d
> endfor
> hdf_sd_end,id
>
> ; Open file and initialize vdata reading
> file_id=hdf_open(file,/read)
> vd_id = -1
> vd_handle = -1
> end_of_file=0
>
> vds = hdf_vd_lone(file_id)
> nvds = n_elements(vds)
> if (nvds eq 0) then begin
>   print, 'ERROR: No vdatas found in file'

```

```

> stop
> endif
>
> ; Loop over vdatas
> for i=0,attributes+nvds-1 do begin
>
>     vd_id = hdf_vd_getid(file_id,vd_id)
>     vd_handle=hdf_vd_attach(file_id,vd_id,/read)
>
>
>     hdf_vd_get,vd_handle,nfields=nf,name=vd_name,count=count,fields=fields
>
>
>     ; Read pressure fields a_lev and b_lev into variables
>     if (strmid(vd_name,0,4) eq 'pres' ) then begin
>
>         for j=0,nf-1 do begin
>             hdf_vd_getinfo,vd_handle,j,name=fieldname,size=size,type=type
>
>             name = strcompress(fieldname,/remove_all)
>             iret = execute('nread=hdf_vd_read(vd_handle,'+$
>                 name+',fields="'+fieldname+'")')
>             if (iret ne 1) then begin
>                 print,'Error dataset',i
>                 stop
>             endif
>
>         endfor
>         no2.a_lev(0:nplev-1) = transpose(a_lev)
>         no2.b_lev(0:nplev-1) = transpose(b_lev)
>     endif
>
>     if (strmid(vd_name,0,4) eq 'NO2_' ) then begin
>         if (vd_name ne 'start_time' and vd_name ne 'end_time') then
> begin
>         track_date = long64(strmid(vd_name,strlen(vd_name)-8,8))
>         track_date = track_date * 1000 + 20000000000000
>         endif
>
>         ; Check whether orbit fits in data structure
>         if (no2.norbits ge maxorbits) then begin
>             print,no2.norbits
>             print, 'ERROR: more orbits in file than', maxorbits
>             stop
>         endif
>         if (count gt maxpix) then begin
>             print, 'ERROR: orbit', no2.norbits+1, ' has more pixels (', $
>                 count, ' ) than', maxpix

```

```

>     stop
> endif
>
> ; Read fields into variables
> for j=0,nf-1 do begin
>     hdf_vd_getinfo,vd_handle,j,name=fieldname,size=size,type=type
>     name = strcompress(fieldname,/remove_all)
>     iret = execute('nread=hdf_vd_read(vd_handle,'+$
>         name+',fields="'+fieldname+'"')')
>     if (iret ne 1) then begin
>         print,'Error dataset',i
>         stop
>     endif
>
> endfor
>
> ; Add variables to structure
> no2.date(no2.norbits,0:count-1) = date
> no2.time(no2.norbits,0:count-1) = time
> no2.lon(no2.norbits,0:count-1) = lon
> no2.lat(no2.norbits,0:count-1) = lat
> no2.vcd(no2.norbits,0:count-1) = vcd
> no2.sigvcd(no2.norbits,0:count-1) = sigvcd
> no2.vcdtrop(no2.norbits,0:count-1) = vcdtrop
> no2.sigvcddt(no2.norbits,0:count-1) = sigvcddt
> no2.vcdstrat(no2.norbits,0:count-1) = vcdstrat
> no2.sigvcdds(no2.norbits,0:count-1) = sigvcdds
> no2.fltrop(no2.norbits,0:count-1) = fltrop
> no2.psurf(no2.norbits,0:count-1) = psurf
> no2.sigvcdak(no2.norbits,0:count-1) = sigvcdak
> no2.sigvcddtak(no2.norbits,0:count-1) = sigvcddtak
> no2.kernel(no2.norbits,0:count-1,*) = transpose(kernel)
> no2.ghostcol(no2.norbits,0:count-1) = ghostcol
> no2.npix(no2.norbits) = count
>
>
> endif
>
> if (strmid(vd_name,0,4) eq 'GEO_') then begin
>
>     ; Check whether orbit fits in data structure
>     if (no2.norbits ge maxorbits) then begin
>         print, 'ERROR: more orbits in file than', maxorbits
>         stop
>     endif
>     if (count gt maxpix) then begin
>         print, 'ERROR: orbit', no2.norbits+1, ' has more pixels
> (, $

```

```

>         count, ' ) than', maxpix
>         stop
>     endif
>
> ; Read fields into variables
>     for j=0,nf-1 do begin
>
> hdf_vd_getinfo,vd_handle,j,name=fieldname,size=size,type=type
>         name = strcompress(fieldname,/remove_all)
>         iret = execute('nread=hdf_vd_read(vd_handle,'+$
>             name+',fields="'+fieldname+'"')
>         if (iret ne 1) then begin
>             print,'Error dataset',i
>             stop
>         endif
>     endfor
>
> ; Add variables to structure
>     no2.sza(no2.norbits,0:count-1) = sza
>     no2.vza(no2.norbits,0:count-1) = vza
>     no2.raa(no2.norbits,0:count-1) = raa
>     no2.ssc(no2.norbits,0:count-1) = ssc
>     no2.loncorn(no2.norbits,0:count-1,*) = transpose(loncorn)
>     no2.latcorn(no2.norbits,0:count-1,*) = transpose(latcorn)
>
>
> endif
>
> if (strmid(vd_name,0,4) eq 'ANC_') then begin
>
> ; Check whether orbit fits in data structure
> if (no2.norbits ge maxorbits) then begin
>     print, 'ERROR: more orbits in file than', maxorbits
>     stop
> endif
> if (count gt maxpix) then begin
>     print, 'ERROR: orbit', no2.norbits+1, ' has more pixels
> (, $
>         count, ' ) than', maxpix
>         stop
>     endif
>
> ; Read fields into variables
>     for j=0,nf-1 do begin
>
> hdf_vd_getinfo,vd_handle,j,name=fieldname,size=size,type=type
>         name = strcompress(fieldname,/remove_all)
>         iret = execute('nread=hdf_vd_read(vd_handle,'+$

```

```

>         name+',fields="'+fieldname+'"')
>     if (iret ne 1) then begin
>         print,'Error dataset',i
>         stop
>     endif
>
> endfor
>
> ; Add variables to structure
> no2.scd(no2.norbits,0:count-1) = scd
> no2.amf(no2.norbits,0:count-1) = amf
> no2.amftrop(no2.norbits,0:count-1) = amftrop
> no2.amfgeo(no2.norbits,0:count-1) = amfgeo
> no2.scdstr(no2.norbits,0:count-1) = scdstr
> no2.clfrac(no2.norbits,0:count-1) = clfrac
> no2.cltpres(no2.norbits,0:count-1) = cltpres
> no2.albclr(no2.norbits,0:count-1) = albclr
> no2.crfrac(no2.norbits,0:count-1) = crfrac
> no2.ltropo(no2.norbits,0:count-1) = ltropo
> no2.norbits = no2.norbits+1
>
>
> endif
>
> hdf_vd_detach, vd_handle
>
> endfor
>
> hdf_close, file_id
>
> end

```

To write in ASCII format you just need three tools: OPENW, to open the file for writing, PRINTF to write data to that file in ASCII format, and CLOSE to close the file. If you're saying "it can't be that simple", you're right. The problem is that you've got to decide what you want to write, in what order, and with what format strings. In this case, you could try starting with free-formatted output. The command

PRINTF, outfile, no2

will dump out all of the data you've retrieved. However, I can virtually guarantee you'll want to replace the free formatting with explicit formatting. Look in the online help in the section "Using Explicitly Formatted Input/Output". There is, unfortunately, an awful lot of complicated and tricky stuff you'll need to understand to make good use of those features.

Note: for some purposes, just using the ncdump utility that comes with HDF may be quicker and easier than writing a program specialized to read this particular file type.
