

# IDL read routine for DOMINO HDF4 data file

Folkert Boersma, KNMI, 21 april 2008

The following IDL program reads a DOMINO HDF4 data file and stores the information in a structure `lv2`. Because of the large number of pixels observed by OMI in one day ( $1644 \times 60 \times 15 \approx 1.5 \times 10^6$ ), quite a number of less relevant entries has been commented out. For DOMINO NO<sub>2</sub> data, the most relevant data fields are the location (pixel center and pixel corners), and the tropospheric NO<sub>2</sub> column (`vcdtrop`), as well as the `fltrop` (0: cloud radiance fraction <50%, -1: >50%) and the viewing zenith angle (`vza`). The latter is especially relevant since OMI pixels increase significantly in size going from nadir ( $24 \times 13 \text{ km}^2$ ) to the edges of the swath ( $135 \times 26 \text{ km}^2$ ). By simply uncommenting a line, a user can add more data fields to the structure `lv2`, at the expense of read-in speed.

## IDL sample code to read the DOMINO NO<sub>2</sub> fields

The code below (download as ascii-file `read_omi_no2.pro` from [www.temis.nl](http://www.temis.nl)) is stand-alone, and does not need any other routines. Note that a user needs to specify a filename and path himself. The filepath in the software below is just for illustration.

```
pro read_omi_no2

;-----
; Read data from OMI HDF file and store in structure lv2.
; "maxorbits" is the maximum number of OMI orbits in one day.
; "maxpixels" is the maximum number of pixels observed by OMI in one day.
;
; Folkert Boersma and Ronald van der A, KNMI, 21 April 2008
;-----

file = '/nobackup/users/boersma/omi/coll3/2004/11/no2track20041112.hdf'

print, 'Reading OMI data from ', file

maxorbits = 15
maxpix    = 15000000

norbits=0

; Define structure for OMI data
lv2 = { $
  npix      : 0L, $
  ;date     : make_array(maxpix,/int,value=-999), $
  ;time     : make_array(maxpix,/int,value=-999), $
  lon       : make_array(maxpix,/float,value=-999.9), $
  lat       : make_array(maxpix,/float,value=-999.9), $
  ;vcd      : make_array(maxpix,/float,value=-999.9), $
  ;sigvcd   : make_array(maxpix,/float,value=-999.9), $
  vcdtrop   : make_array(maxpix,/float,value=-999.9), $
  sigvcddt  : make_array(maxpix,/float,value=-999.9), $
  ;vcdstrat : make_array(maxpix,/float,value=-999.9), $
  ;sigvcdds : make_array(maxpix,/float,value=-999.9), $
  fltrop    : make_array(maxpix,/int,value=-999.9), $
  ;psurf    : make_array(maxpix,/int,value=-999.9), $
  ;sigvcdak : make_array(maxpix,/float,value=-999.9), $
  ;sigvcctak : make_array(maxpix,/float,value=-999.9), $
  ;kernel   : make_array(maxpix,nplev,/float,value=-999),$
  ;sza      : make_array(maxpix,/float,value=-999.9),$
  vza       : make_array(maxpix,/float,value=-999.9),$
  ;raa      : make_array(maxpix,/float,value=-999.9),$
  ;ssc      : make_array(maxpix,/int,value=-999.9), $
  loncorn   : make_array(maxpix,4,/float,value=-999),$
  latcorn   : make_array(maxpix,4,/float,value=-999)}
  ;scd      : make_array(maxpix,/float,value=-999),$
  ;amf      : make_array(maxpix,/float,value=-999),$
  ;amftrop  : make_array(maxpix,/float,value=-999),$
  ;amfgeo   : make_array(maxpix,/float,value=-999),$
  ;scdstr   : make_array(maxpix,/float,value=-999),$
```

```

;clfrac      : make_array(maxpix,/float,value=-999),$
;cltpres     : make_array(maxpix,/float,value=-999),$
;albc1r      : make_array(maxpix,/float,value=-999),$
;crfrac      : make_array(maxpix,/float,value=-999),$
;ltropo      : make_array(maxpix,/int,value=-999)}

; Open file and initialize vdata reading
; Extracting the number of attributes:
file_id=hdf_open(file,/read)
sd_id=hdf_sd_start( file,/read)
hdf_sd_fileinfo,sd_id,nr_dataset,nr_attributes
hdf_sd_end,sd_id
hdf_close,file_id

; Open file
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
istart = 0

for i=0,nr_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
    print,'nfields ',nf,' name ',vd_name,' count ',count

    ;if (fields eq
'date,time,lon,lat,vcd,sigvcd,vcdtrop,sigvcdt,vcdstrat,sigvcds,fltrop,psurf,sigvcdak,sigvcdtak
,kernel,ghostool') then begin
        if (strpos(fields,'date,time') eq 0) then begin
            if (vd_name ne 'start_time' and vd_name ne 'end_time' and vd_name ne 'track_number')
then begin
                track_date = long64(strmid(vd_name,strlen(vd_name)-10,10))
                track_date = track_date * 100 + 200000000000000
            endif

            ; Check time range
            ; if (track_date ge date1 and track_date le date2) then begin
            print, count, ' pixels in orbit', norbits+1

            ; Check whether orbit fits in data structure
            if (norbits ge 16) then begin
                print, 'ERROR: more orbits in file than 16'
                stop
            endif
            iend=istart+count
            if (iend gt maxpix) then begin
                print, 'ERROR: day-file contains more pixels (', iend, ' ) 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,'readscia: error dataset',i
                    stop
                endif
            endfor
        endif
    endif
endfor

```

```

; Add variables to structure
;lv2.date(istart:iend-1) = date
;lv2.time(istart:iend-1) = time
lv2.lon(istart:iend-1) = lon
lv2.lat(istart:iend-1) = lat
;lv2.vcd(istart:iend-1) = vcd
;lv2.sigvcd(istart:iend-1) = sigvcd
lv2.vcdtrop(istart:iend-1) = vcdtrop
lv2.sigvcdt(istart:iend-1) = sigvcdt
;lv2.vcdstrat(istart:iend-1) = vcdstrat
;lv2.sigvcds(istart:iend-1) = sigvcds
lv2.fltrop(istart:iend-1) = fltrop
;lv2.psurf(istart:iend-1) = psurf
;lv2.sigvcdak(istart:iend-1) = sigvcdak
;lv2.sigvcdtak(istart:iend-1) = sigvcdtak
;lv2.kernel(istart:iend-1,*) = transpose(kernel)
endif

; if ( fields eq 'sza,vza,raa,ssc,loncorn,latcorn,pixelnr,imagenr' ) then begin
if ( strpos(fields,'sza,vza') eq 0 ) then begin

; 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,'readscia: error dataset',i
stop
endif
endif

endfor

; Add variables to structure
;lv2.sza(istart:iend-1) = sza
lv2.vza(istart:iend-1) = vza
;lv2.raa(istart:iend-1) = raa
;lv2.ssc(istart:iend-1) = ssc
lv2.loncorn(istart:iend-1,*) = transpose(loncorn)
lv2.latcorn(istart:iend-1,*) = transpose(latcorn)

endif

; if (fields eq
'scd,scdorig,amf,amftrop,amfgeo,scdstr,clfrac,cltpres,albclr,crfrac,ltropo,fcvcdtr') then
begin
if (strpos(fields,'scd') eq 0 ) then begin

; 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,'readscia: error dataset',i
stop
endif
endif

endfor

; Add variables to structure
;lv2.scd(istart:iend-1) = scd
;lv2.amf(istart:iend-1) = amf
;lv2.amftrop(istart:iend-1) = amftrop
;lv2.amfgeo(istart:iend-1) = amfgeo
;lv2.scdstr(istart:iend-1) = scdstr
;lv2.clfrac(istart:iend-1) = clfrac
;lv2.cltpres(istart:iend-1) = cltpres
;lv2.albclr(istart:iend-1) = albclr
;lv2.crfrac(istart:iend-1) = crfrac
;lv2.ltropo(istart:iend-1) = ltropo
norbits = norbits+1
istart = iend
print,istart,iend

```

```
        lv2.npix = iend
    endif

    hdf_vd_detach, vd_handle

endfor

hdf_close, file_id

; Print information on data
print, ' ', lv2.npix, ' pixels'

end
```