**GOES-16 AOD File Format and IDL Reader**

**6/26/2018**

The GOES-16 aerosol optical depth (AOD) is **beta** product, which has not been fully validated, and therefore is not recommended for scientific use.

The GOES-16 aerosol optical depth (AOD) files are in NetCDF4 format with naming convention as:

“**OR\_ABI-L2-AODC-M3\_G16\_s*YYYYDDDHHMMSST\_eYYYYDDDHHMMSST\_cYYYYDDDHHMMSST.nc***”

Where **AODC** is for CONUS. The corresponding string for full disk is **AODF**. **M3** means satellite scan mode 3. It can also be **M4** for scan mode 4. ‘***YYYY***’ is four-digit year, ‘***DDD***’ is three-digit day of the year, ‘***HH***’ is two-digit hour, ‘***MM***’ is two-digit minute, ‘***SS***’ is two-digit second, ‘***T***’ is the tenth of second. ‘***sYYYYDDDHHMMSST***’ is the granule starting time, ‘***eYYYYDDDHHMMSST***’ is the ending time, and ‘***cYYYYDDDHHMMSST***’ is the file creation time. One example the file name:

OR\_ABI-L2-AODC-M3\_G16\_s20172471757154\_e20172471759527\_c20172471805345.nc

The GOES-16 AOD data has a spatial resolution of 2 km. It has a temporal resolution of 5 minutes for CONUS and 15 minutes for full disk. The file contains AOD at 550 nm field and quality flag field (DQF), as shown in Table 1. The dimension of the field is xsize=2500, ysize=1500 for CONUS, and xsize=ysize=5424 for full disk. The file also contains many metadata fields, whose descriptions can be found in Table 5.10.6-1 and Table 5.10.6-2 of GOES-R Product Definition and Users’ Guide (<http://www.goes-r.gov/products/docs/PUG-L2+-vol5.pdf>).

Table 1. AOD file data fields

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Dimension** |
| AOD | output | Retrieved aerosol optical depth at 550 nm | pixel (xsize, ysize) |
| DQF | output | Quality flag. 0: good. 1: bad (for data before 4/5/2018)  Quality flag. 0: high, 1: medium, 2: low, 3: not retrieved. (for data after 4/5/2018)  Data on 4/5/2018 contain both versions and also some wrong data. | pixel(xsize, ysize) |

**IDL code to read AOD**

; call command in idl: read\_aod,fname,aod,dqf

; where fname is file name

; aod and dqf are output data fields

;

function read\_abi\_var,filename,var\_name, scaled

print, filename

id = ncdf\_open(filename)

v\_id = ncdf\_varid(id, var\_name)

if (v\_id eq -1) then begin

print, var\_name + " does not exist"

return, -1

endif

long\_name = ' '

;ncdf\_attget,id, v\_id, 'long\_name', long\_name

print, "Getting " + string(var\_name)

ncdf\_varget,id, v\_id, unscaled ; stored data

test = NCDF\_VARINQ(id, v\_id)

; now we need to see if 'scale\_factor' is any of the attributes

num\_attr = test.Natts

data\_type = test.datatype

if (scaled eq 1) then begin

ncdf\_attget,id, v\_id, 'scale\_factor', scale\_factor

ncdf\_attget,id, v\_id, 'add\_offset', add\_offset

ncdf\_attget,id, v\_id, '\_FillValue', Fill\_value

temp =unscaled

unscaled = (temp \* scale\_factor) + add\_offset

index = where(temp eq Fill\_value, count)

if (count gt 0) then unscaled(index) = -999.0

endif

ncdf\_close, id

return, unscaled

end

pro read\_g16\_aod,infname,aod,dqf

file\_id = NCDF\_OPEN(infname,/NOWRITE)

data\_id = NCDF\_VARID(file\_id, 'AOD')

NCDF\_VARGET, file\_id, data\_id, newData

NCDF\_ATTGET, file\_id, data\_id, 'valid\_range', valid\_range

NCDF\_ATTGET, file\_id, data\_id, '\_FillValue', FillValue

NCDF\_ATTGET, file\_id, data\_id, 'scale\_factor', scale

NCDF\_ATTGET, file\_id, data\_id, 'add\_offset', offset

ncdf\_close,file\_id

fillIdx = WHERE(newData EQ FillValue, c2)

;fixed the issue for unsigned integer by idl reading as signed integer

newData = UINT(newData)

valid\_range = UINT(valid\_range)

print,'valid\_range,',valid\_range

newData = scale[0] \* newData + offset[0]

aod = FLOAT(newData)

IF (c2 GT 0) THEN aod[fillIdx] = -9999.

dqf=read\_abi\_var(infname,'DQF',0)

end

**GOES-16 Conversion Tool**

# Motivation

Most GOES-16 Level 2 (L2) products are gridded into a fixed grid, but the coordinates are scaled into integer radian angles rather than a map projection. While this helps reduce file size, it requires a work around to display the files in netCDF viewers.

# Description

A python program reads in GOES-16 L2 AOD and ADP aerosol product netCDF files and converts fixed grid radian coordinates to their corresponding latitude longitude projection. The result is then re-saved in a more easily viewed NetCDF file. This tool can also be used to batch convert a large number of individual files and compile them into a single file time series. This permits for easier viewing and animation in common NetCDF viewing/plotting tools.

Converting files occurs in three simple steps:

1. Download and place files that you wish to display in the input folder.
2. Run the python script to convert the files.
3. Check the output file for the results.

## Getting Started

### Prerequisites

* GOES-16 aerosol data that you wish to display
* Download the netCDF conversion tool

## Running

### Inputs

Before doing anything, you need to have ABI L2 NetCDF files. Place the files that you wish to convert into the input folder.

Note: If more than one file of the same product and region are placed in the input folder, they will be combined into a single file.

### Run abi\_convert.exe

Double click to convert all files in the input directory. See the "Creating a standalone executable file" section for details.

### Output

Finished files will be saved in as ABI-[product name][region]-converted.nc

### Caveats

* The program will convert ALL files in the input directory of each product/region type.
* Make sure files are sequential.
* Output files WILL OVERWRITE existing files with the same name.

## Useful NetCDF viewing tools

* [Panoply](https://www.giss.nasa.gov/tools/panoply/) - Developed by Dr. Robert Schmunk, NASA/GISS. Great general purpose tool for display NetCDF data. Has a large number of map projections.
* [NOAA's Weather and Climate Toolkit](https://www.ncdc.noaa.gov/wct)Useful for plotting data over the continental United States.

## More Information

* GOES-R Series input data can be downloaded from NOAA's [Comprehensive Large Array-data Stewardship System (CLASS)](https://www.class.noaa.gov/).
* Additional information on GOES-16 products can be found in the [Algorithm Theoretical Basis Documents (ATBD)](http://www.goes-r.gov/resources/docs.html).