

# Python routines to convert L1 files in netCDF format

12 March 2018

In the frame of the FRM4DOAS project, it has been decided that the standard format for L1 radiance spectra to be submitted to the centralised processing system is netCDF. The description of groups and variables corresponding to this format is provided in the L1\_format\_20180312\_v1.pdf document. BIRA-IASB provided Python routines to convert in netCDF files in the ASCII format used for spectra measured during the CINDI-2 campaign and files in the MFC BIRA binary format (daily files generated by MFC4QDOAS\_Converter from original MFC binary or standard format). A description of these routines is given below.

## Function call

```
python3 -c "from netCDF_L1 import *; \
netCDF_L1_Convert(cfgFile, \
                 inFile, \
                 ncPath, \
                 refFile='', \
                 slfFile='', \
                 tpFile='', \
                 clbFile='', \
                 file_version='001', \
                 slfWavelength=0., \
                 file_format='ASCII', \
                 commentChar='#', \
                 solarAzimuthCorrection=0., \
                 viewingAzimuthCorrection=0.);"
```

with :

cfgFile	ini file with some (key,value) to use as file attributes
inpFile	name of the input file (native format)
ncPath	name of the output path
refFile	optional, reference file for QDOAS calibration as keydata
slfFile	optional, slit function file provided by the data submitter as keydata
tpFile	optional, temperature/profile file
clbFile	optional, wavelength calibration file
file_version	optional, file version on three digits
slfWavelength	optional, for 1D slit function, the wavelength at which the slit function has been measured
file_format	native file format ('ASCII' or 'MFC-BIRA')
commentChar	optional, character to use for comment in clbFile, refFile, slfFile
solarAzimuthCorrection	optional, azimuth correction to apply to solar azimuth angle
viewingAzimuthCorrection	optional, azimuth correction to apply to viewing azimuth angle

Some comments about the parameters :

- Configuration file contains information that are used as netCDF file attributes but also to build the output file name (see example below)
- The format of “refile”, “slfFile”, “clbFile” is ASCII (it is the same format as the ones provided for CINDI-2)
- tpFile : is loaded by the `__init__` function of “netCDF\_L1\_tp” class (could be modified according to the format of your file). The time vector has to cover the measurements times.
- commentChar : usually ‘#’ or ‘;’ depending on the user preference
- solarAzimuthCorrection, viewingAzimuthCorrection : profiling tools usually work with relative azimuth. A correction could be applied if solar and viewing azimuth are not given with the same convention or to suit requirements (for example, azimuth requested within 0..360 range).

`newAzimuth=oldAzimuth+correction.`

## ***Example of config file***

[instrument]

```
instr_number=004
instr_channel=1
instr_type=maxdoas
```

[data\_submitter]

```
institution=BIRA.IASB
pi_name=Michel Van Roozendael
pi_email=michel.vanroozendael@aeronomie.be
do_name=Caroline Fayt
do_email=caroline.fayt@aeronomie.be
ds_name=Francois Hendrick
ds_email=francois.hendrick@aeronomie.be
```

[location]

```
station_name=CABAUW
instr_latitude=51.97
instr_longitude=4.93
instr_altitude_asl=20.
instr_altitude_agl=5.
```

## ***Example of call***

```
python3 -c "from netCDF_L1 import *; \
netCDF_L1_Convert('./Example/cindi2_004_1_asc2netcdf.ini', \
                  './Example/bira_maxdoas_4_vis_spectra_cindi2_20160924_v1.asc', \
                  './Example', \
                  refFile='./Example/bira_maxdoas_4_vis_spectra_cindi2_20160924_v1.ref', \
                  slfFile='./Example/bira_maxdoas_4_vis_slfct_cindi2_20160924_v1.asc', \
                  file_format='ASCII', \
                  commentChar='#')"
```

## **Conversion from other file formats**

The netCDF\_L1\_Convert function (netCDF\_L1.py) currently supports three formats for L1 spectra :

- ASCII : ASCII format used to submit CINDI-2 spectra
- CCDEEV : format used at BIRA-IASB
- MFC\_BIRA : format generated by the BIRA\_IASB MFC4QDOAS\_Converter tool

To add a new format, a function netCDF\_L1from<new format> has to be created. This function should call your function that reads the spectra in the native format and make the equivalence between the data sets and the field of the class “netCDF\_L1\_records” that contains the description of the netCDF file.

Refer to netCDF\_L1\_LoadfromASCII or netCDF\_L1\_LoadfromMFC\_BIRA to see how to proceed.

Then in netCDF\_L1\_Convert function, the following tests have to be completed with a call to netCDF\_L1from<new format> :

```
if (file_format=='CCDEEV'):
    L1_data=netCDF_L1_LoadfromCCDEEV(inpFile,clbFile);
elif (file_format=='ASCII'):
    L1_data=netCDF_L1_LoadfromASCII(inpFile);
elif (file_format=='<new format>'):
    L1_data=netCDF_L1_Loadfrom<new format>(inpFile,<options>);
elif (file_format=='MFC_BIRA'):
    L1_data=netCDF_L1_LoadfromMFC_BIRA(inpFile,clbFile,commentChar=commentChar);
else:
    print(['+functionName+'] Unknown format '+file_format+' !!!);
```