# Python codes generate\_days\_netcdf\_metsensors and generate\_days\_local netcdf\_metsensors

New Python codes have been written to produce netCDF files from our meteorological sensors, raingauges and broadband radiometers. This has been necessary for several reasons: since April 2020 we are required to use a new format for our netCDF files as defined by AMOF (using convention (CF-1.6, NCAS-AMF-1.0) rather than our existing format as used in cfarr\*.nc files), we have been in the process of transferring as many met. sensors as possible to Campbell Scientific CR1000X dataloggers (so the format of our raw data files has changed) and we have a desire to move our data processing from Matlab to Python.

This document describes how to use the codes. A separate document, cao\_metinstruments\_netcdf\_production.docx, describes their structure and rationale in more detail.

You are unlikely to need to use generate\_days\_local netcdf\_metsensors.py. It was an interim code that was used during the transition period between Matlab and Python codes and old and new netCDF file formats.

## Quick start guide to producing netCDF files from met. data

1. Use PuTTY to start a terminal window connected to Wilma.
2. Log in to Wilma using usual username and password.
3. **ssh python**

This transfers you from Wilma to the “python” computer.

1. Enter your password again
2. Enter the command **conda activate chil\_3\_8**
3. Connect to the directory where the code is located using

**cd /home/chilbolton\_software/python/ncas\_python/metsensors**

1. Enter **./generate\_days\_netcdf\_metsensors.py -s *yyyymmdd* -e *yyyymmdd* –x 2.** This will generate files for Chilbolton drop-counting and tipping bucket gauges.
2. Enter **./generate\_days\_netcdf\_metsensors.py -s *yyyymmdd* -e *yyyymmdd* –x 1** once the previous step has ended. This will generate files for the Chilbolton weighing raingauge.
3. If you want to generate Sparsholt files, enter **./generate\_days\_netcdf\_metsensors.py -s *yyyymmdd* -e *yyyymmdd* –x 5** once the previous step has ended.
4. Other “-x” options are listed in the next section. The crontab for the “range” user on acaconda gives an indication of current operational datasets.
5. You can see the results of your corrections at the pages linked from <https://gate.chobs.rl.ac.uk/amof-netCDF/>. Points that are unreliable are now left in the plot but marked with a red circle, rather than removing them. The raingauge names have changed to meet NCAS requirements. They correspond to the previous names as follows:

|  |  |  |
| --- | --- | --- |
| **New NCAS name** | **Previous name (and correction file name)** | **Description** |
| ncas-rain-gauge-1 | rg001dc\_ch | Chilbolton turf-wall drop-counting gauge, drop\_count\_a |
| ncas-rain-gauge-2 | rg006dc\_ch | Chilbolton standard drop-counting gauge, drop\_count\_b |
| ncas-rain-gauge-3 | rg008dc\_ch | Chilbolton low rate drop-counting gauge, drop\_count\_c |
| ncas-rain-gauge-4 | pldc\_ch | Chilbolton weighing raingauge |
| ncas-rain-gauge-5 | rg004tb\_ch | Chilbolton tipping bucket raingauge |
| ncas-rain-gauge-6 | rg002tb\_sp | Sparsholt tipping bucket raingauge |
| ncas-rain-gauge-7 | rg001dc\_sp | Sparsholt drop-counting raingauge. |
| ncas-rain-gauge-9 | rg009dc\_ch | Chilbolton standard drop-counting gauge adjacent to flux compound |

## Datasets which can be processed

Both codes call a range of functions depending on which met. sensor is to be processed. The purpose of each code is as shown in the table below, together with the meaning of each sensor selection.

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Purpose** | **Option** | **Instruments processed** |
| generate\_days\_netcdf\_metsensors | Generates new format ncas\*.nc netCDF files, suitable for archiving at CEDA from 1st April 2020 onwards | 1 | Pluvio weighing raingauge |
| 2 | Chilbolton met. sensors, drop-counting and tipping bucket raingauges |
| 3 | Chilbolton raingauges from format5 files (use before 20200728) |
| 4 | Chilbolton broadband radiometers from format5 files (use up to and including 12/10/2021). |
| 5 | Sparsholt drop-counting and tipping bucket raingauge |
| 6 | Chilbolton RD-80 disdrometer |
| 7 | Sparsholt RD-80 disdrometer (before 20190624) |
|  |  | 8 | Chilbolton PWS100 present weather sensor |
|  |  | 9 | Chilbolton broadband radiometers (use from 22/11/2021) |
|  |  | 10 | Chilbolton net flux radiometer from Datataker files |
|  |  | 11 | Chilbolton sonic anemometer from Datataker files |
|  |  |  |  |
| generate\_days\_local\_netcdf\_metsensors | Generates original format cfarr\*.nc netCDF files, as were archived at CEDA up to 31st March 2020. | 1 | Met., rain and broadband radiometer plots for Weather Web |
| 2 | Chilbolton met\_sensors, raingauge and multiple\_raingauges files |
| 3 | Sparsholt multiple-raingauges files |
| 4 | Sparsholt single raingauge files |
| 5 | Chilbolton radiometer-vis and radiometer-ir files |

## Viewing netCDF files

NetCDF files are not viewable using a text file editor. Instead use the “ncdump” command at the command line. You will see a text depiction of the header lines (or “metadata”) followed by the data lines in the file.

Example:

ncdump /data/amof-netCDF/ncas-sonic-5/ ncas-sonic-5\_cao\_*yyyymmdd*\_mean-winds\_v1.0.nc | less

## Comments on the 2 codes

Generate\_days\_netcdf\_metsensors.py is the usual code to use for producing netCDF files from new data.

Generate\_days\_netcdf\_metsensors.py options 2 and 5 read data from the Campbell datalogger files which are regularly updated in /data/range/mirror\_grape\_loggernet. Each datalogger writes all the values that it records to 1 comma-separated file (although they could be separated by instrument in future if it’s felt to be helpful).

The remaining options in generate\_days\_netcdf\_metsensors.py read data from other loggers and computers. NetCDF files from the Pluvio weighing raingauge are only produced using generate\_days\_netcdf\_metsensors.py – there are no old-format cfarr\*.nc files for this instrument.

Generate\_days\_local\_netcdf\_metsensors.py produces files in the old Chilbolton netCDF format. It should become less necessary to use it with time. It was written so that we could easily view quicklooks of data at <https://gate.chobs.rl.ac.uk/netCDF/> (internal to Chilbolton) and <https://www.chobs.rl.ac.uk/graphs/> (externally visible). A new quicklook interface has been developed, <https://gate.chobs.rl.ac.uk/amof-netCDF/> (internal to Chilbolton).

Generate\_days\_local\_netcdf\_metsensors.py depends on an original cfarr\*.nc file having already been generated in the relevant subdirectory under /data/netCDF/files. These cfarr\*.nc files are generated using Matlab as described in… Where met. instruments have been moved to being logged by the Campbell Scientific dataloggers from the old Microlink system these files have no data. This code takes those files and modifies them with data recorded using the Campbell Scientific dataloggers. Having done that, it is possible to generate quicklooks of the data using another Matlab code.

Generate\_days\_local\_netcdf\_metsensors.py option 1 modifies the cfarr\*.nc files which are used to generate images for the Chilbolton Weather Web. These images are generated automatically using a separate process which is not documented to the best of my knowledge.

## Procedure

The command line input uses the same format for both programs. Generate\_days\_netcdf\_metsensors.py will be used in the following instructions.

Text shown in bold is a command to be typed.

1. Use PuTTY to start a terminal window connected to Wilma.
2. Log in to Wilma using usual username and password.
3. **ssh python**

This transfers you from Wilma to the “python” computer.

1. Enter your password again
2. Connect to the directory where the code is located using

**cd /home/chilbolton\_software/python/ncas\_python/met\_sensors**

1. Enter **./generate\_days\_netcdf\_metsensors.py -s *yyyymmdd* -e *yyyymmdd* –x *option***

Substitute the start (s) and end (e) date you want to process for yyyymmdd e.g. 20210425. If you only want to process 1 day they can both be the same. The option number comes from the table above. To process raingauge data after corrections have been generated using raingauge\_click\_plots.py, use option 2 (it will also recreate Chilbolton met. sensor files such as temperature and wind speed).

1. The program takes a few seconds to run – as it does text will appear on the screen. No plots are displayed. Add “successful” comment to text output. When it finishes the command prompt will be visible again.
2. Go to <https://gate.chobs.rl.ac.uk/amof-netCDF/> to see the new files that have been produced. Any data marked as suspect or bad in the corrections files will be marked with a red circle.

The filenames and their corresponding sensors are as shown in the table at the bottom of this section.

1. You should have a command prompt at the start of the line, as at the start. Return to command 6 to process other days.
2. When you have finished work, enter **exit** to leave the Python machine, then **exit** again to leave Wilma. Or you can stay on the Python machine to run other programs.

|  |  |
| --- | --- |
| **Instrument name and ID** | **NCAS netCDF filename** |
| HMP155A temperature/relative humidity sensor oatnew\_ch, rhnew\_ch | ncas-temperature-rh-1\_cao\_*yyyymmdd*\_surface-met\_v1.0.nc |
| Vector Instruments cup and vane anemometer ws\_ch, wd\_ch | ncas-anemometer-2\_cao\_*yyyymmdd*\_surface-met\_v1.0.nc |
| Vaisala PTB110 pressure sensor QFE\_ch | ncas-pressure-1\_cao\_*yyyymmdd*\_surface-met\_v1.0.nc |
| Turf wall drop-counting raingauge rg001dc\_ch | ncas-rain-gauge-1\_cao\_*yyyymmdd*\_precipitation\_v1.0.nc |
| Standard drop-counting raingauge rg006dc\_ch | ncas-rain-gauge-2\_cao\_*yyyymmdd*\_precipitation\_v1.0.nc |
| Low rate (200 mm collector) drop-counting raingauge rg008dc\_ch | ncas-rain-gauge-3\_cao\_*yyyymmdd*\_precipitation\_v1.0.nc |
| Pluvio weighing raingauge pldc\_ch | ncas-rain-gauge-4\_cao\_*yyyymmdd*\_precipitation\_v1.0.nc |
| Tipping bucket raingauge rg004tb\_ch | ncas-rain-gauge-5\_cao\_*yyyymmdd*\_precipitation\_v1.0.nc |
| Sparsholt tipping bucket raingauge rg002tb\_sp | ncas-rain-gauge-6\_cao-sparsholt\_*yyyymmdd*\_precipitation\_v1.0.nc |
| Sparsholt drop-counting raingauge rg001dc\_sp | ncas-rain-gauge-7\_cao-sparsholt\_*yyyymmdd*\_precipitation\_v1.0.nc |
| Chilbolton RD-80 disdrometer ch\_distrom | ncas-disdrometer-1\_cao\_*yyyymmdd*\_precipitation\_v1.0.nc  …. |
| Sparsholt RD-80 disdrometer sp\_distrom | ncas-disdrometer-2\_cao-sparsholt\_*yyyymmdd*\_precipitation\_v1.0.nc |
| Downwelling pyranometer (visible radiation) pyrCM21\_ch | ncas-radiometer-1\_cao\_*yyyymmdd*\_radiation\_v1.0.nc |
| Downwelling diffuse radiation only pyranometer pyr\_CMP21\_ch | ncas-radiometer-2\_cao\_*yyyymmdd*\_radiation\_v1.0.nc |
| Downwelling pyrgeometer (IR radiation) pyrCG4\_ch | ncas-radiometer-3\_cao\_*yyyymmdd*\_radiation\_v1.0.nc |
| Direct visible radiation pyrCP1\_ch | ncas-radiometer-4\_cao\_*yyyymmdd*\_radiation\_v1.0.nc |
| Flux compound drop-counting raingauge rg009dc\_ch (not yet being collected for AMOF submission, but may be – ncas filename not confirmed) | ncas-rain-gauge-9\_cao\_*yyyymmdd*\_precipitation\_v1.0.nc |
| PWS100 present weather sensor pws100, pws100\_met, pws100\_rain | ncas-present-weather-1\_cao\_*yyyymmdd*\_present-weather\_v1.0.nc |
| Flux compound net flux radiometer, cnr4\_dsw, cnr4\_dlw, cnr4\_usw, cnr4\_ulw, cnr4\_T, ncas filename not confirmed | ncas-radiometer-5\_cao\_*yyyymmdd*\_radiation\_v1.0.nc |
| Flux compound sonic anemometer, sonic | ncas-sonic-5\_cao\_*yyyymmdd*\_mean-winds\_v1.0.nc |

The instrument ID gives the code by which the instrument might be better known (particularly for raingauges) and the name of the corrections file in /data/netCDF /corrections where periods of time with suspect or bad data are recorded.