

How to convert NetCDF to CSV

Last modified on Oct 04, 2023 10:32

- [Introduction](#)
- [First option : Python Script](#)
- [Second option : Panoply](#)
- [Third option : Windows users](#)
- [Other solutions](#)
- [Related articles](#)

Introduction

This article explains how to extract data from a 3 dimension NetCDF file using different options and save the output as a CSV (comma separated variables) file.

You are expected to have installed python 2.7 or later, and the [CDS API](#) on a Linux machine before you continue.

First option : Python Script

The first option is to use a python script (below). The script allows you to covert data from NetCDF in two different ways, as explained in the workflow below:

- Retrieve data with the CDS API and store as a netCDF4 file in the working directory.
- Extract the variable from the NetCDF file and get the dimensions (i.e. time, latitudes and longitudes)
- Extract each time as a 2D pandas DataFrame and write it to the CSV file
- Write the data as a table with 4 columns: time, latitude, longitude, value

Python source code example for ERA5 single level data

```

#!/usr/bin/python3

import cdsapi
import netCDF4
from netCDF4 import num2date
import numpy as np
import os
import pandas as pd

# Retrieve data and store as netCDF4 file
c = cdsapi.Client()
file_location = './t2m.nc'
c.retrieve(
    'reanalysis-era5-single-levels',
    {
        'product_type':'reanalysis',
        'variable':'2m_temperature',  # 't2m'
        'year':'2019',
        'month':'06',
        'day':[
            '24','25'
        ],
        'time':[
            '00:00','06:00','12:00',
            '18:00'
        ],
        'format':'netcdf'
    },
    file_location)

# Open netCDF4 file
f = netCDF4.Dataset(file_location)

# Extract variable
t2m = f.variables['t2m']

# Get dimensions assuming 3D: time, latitude, longitude
time_dim, lat_dim, lon_dim = t2m.get_dims()
time_var = f.variables[time_dim.name]
times = num2date(time_var[:], time_var.units)
latitudes = f.variables[lat_dim.name][:]
longitudes = f.variables[lon_dim.name][:]

output_dir = './'

# ====== METHOD 1 ======
# Extract each time as a 2D pandas DataFrame and write it to CSV
# ======
os.makedirs(output_dir, exist_ok=True)
for i, t in enumerate(times):
    filename = os.path.join(output_dir, f'{t.isoformat()}.csv')
    print(f'Writing time {t} to {filename}')
    df = pd.DataFrame(t2m[i, :, :], index=latitudes, columns=longitudes)
    df.to_csv(filename)
print('Done')

# ====== METHOD 2 ======
# Write data as a table with 4 columns: time, latitude, longitude, value
# ======
filename = os.path.join(output_dir, 'table.csv')
print(f'Writing data in tabular form to {filename} (this may take some time)...')
times_grid, latitudes_grid, longitudes_grid = [
    x.flatten() for x in np.meshgrid(times, latitudes, longitudes, indexing='ij')]
df = pd.DataFrame({
    'time': [t.isoformat() for t in times_grid],
    'latitude': latitudes_grid,
    'longitude': longitudes_grid,
    't2m': t2m[:,].flatten()})
df.to_csv(filename, index=False)
print('Done')

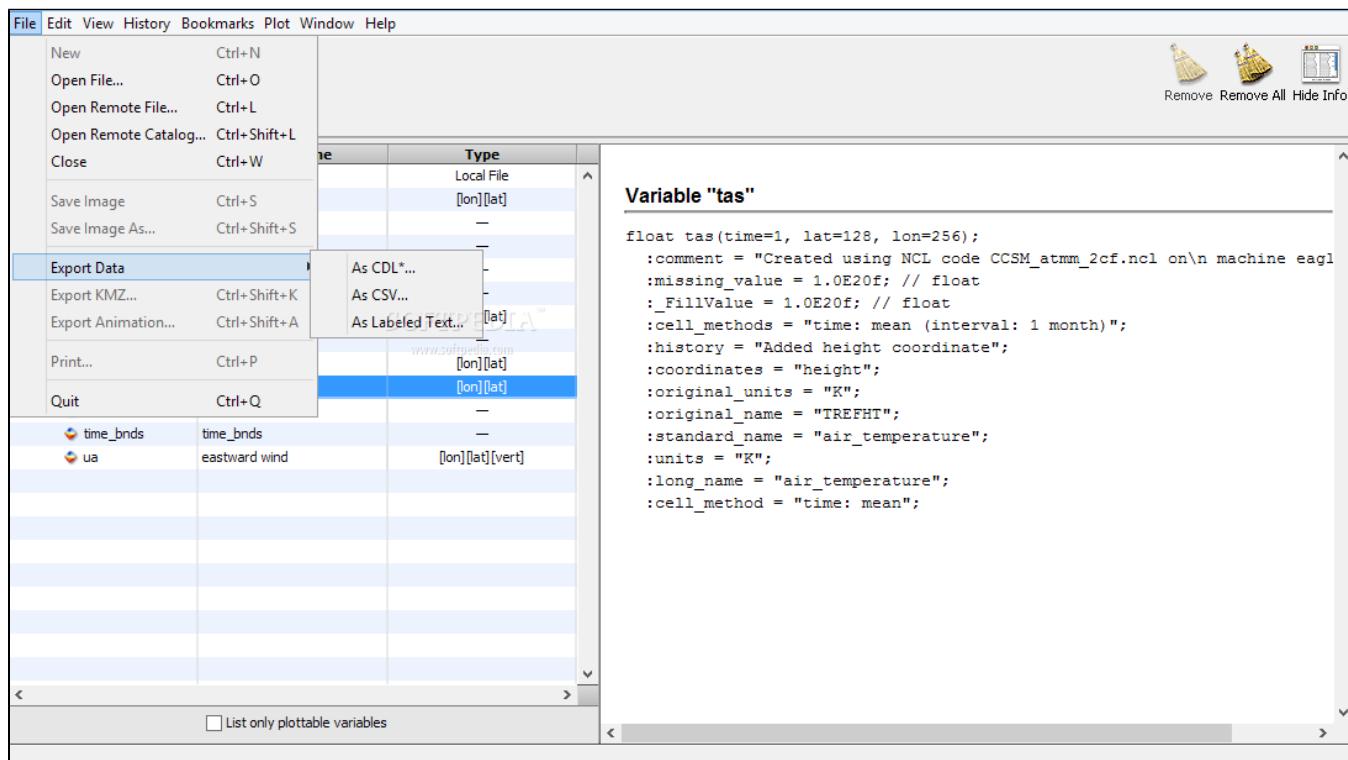
```



Please note that the netCDF file could contain an additional dimension when there is a mix of ERA5 and ERA5T data. For more details please have a look at the following link: [ERA5 CDS requests which return a mixture of ERA5 and ERA5T data](#)

Second option : Panoply

A second option is to convert the data using the NASA 'Panoply' software. User can find the option under File Export data As CSV. The data are saved in the file maintaining the structure of the lon/lat matrix, but different times are divided by an empty row.



Third option : Windows users

A third option to convert the data from NetCDF to CSV, for Windows users, is download and install [netcdf4excel](#). The plug-in opens directly NetCDF files in Microsoft Excel maintaining conventions for the NetCDF variables. Please see the link for more details: <http://netcdf4excel.github.io/>.

Other solutions

For Unix users, there are others options provided by some common NetCDF software packages. Please the links for more details:

- ncdump <http://manpages.ubuntu.com/manpages/xenial/man1/ncdump-hdf.1.html>
- cdo <https://code.mpimet.mpg.de/projects/cdo/wiki/Tutorial>
- nccks <https://www.unidata.ucar.edu/support/help/MailArchives/netcdf/msg08405.html>

This document has been produced in the context of the Copernicus Atmosphere Monitoring Service (CAMS) and Copernicus Climate Change Service (C3S).

The activities leading to these results have been contracted by the European Centre for Medium-Range Weather Forecasts, operator of CAMS and C3S on behalf of the European Union (Delegation Agreement signed on 11/11/2014 and Contribution Agreement signed on 22/07/2021). All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose.

The users thereof use the information at their sole risk and liability. For the avoidance of all doubt, the European Commission and the European Centre for Medium-Range Weather Forecasts have no liability in respect of this document, which is merely representing the author's view.

Related articles

- [What data and charts are available through CAMS \(Copernicus Atmosphere Monitoring Service\)?](#)
- [What data and maps are available through C3S \(Copernicus Climate Change Service\)?](#)
- [CAMS User Support Journey](#)
- [C3S User Support Journey](#)
- [Transformation or regridding of ECMWF Reanalyses data](#)