

# Read and Write Data

ncgen and ncdump to create/export NetCDF and CDL

Thanks to all contributors:

Eduardo Damasio Da Costa

# ncdump and ncgen

Used together, **ncdump** and **ncgen** can accomplish simple netCDF manipulations with little or no programming.

**ncdump** and **ncgen** are reverse operations:



# CDL (network Common Data form Language)

CDL provides a means for NetCDF  $\leftrightarrow$  ASCII translation. It is a simple text-based language that can be directly converted to NetCDF.

It has two main utilities:

1. `ncdump` (from NetCDF to ASCII)
  - to get overview of NetCDF file contents
2. `ncgen` (from ASCII to NetCDF)
  - to create NetCDF files

# ncdump

For quick glimpse at file's "metadata"

- To view header info:

```
ncdump -h myfile.nc
```

- To view header info plus values of coordinate variables:

```
ncdump -c myfile.nc
```

- To output data values in ascii form:

```
ncdump -v varname myfile.nc
```

# An easy way to view NetCDF: CDL

CDL is a human-readable notation for netCDF objects and data.

```
netcdf example { // example of CDL notation
dimensions:
    lon = 3 ;
    lat = 8 ;
variables:
    float rh(lon, lat) ;
        rh:units = "percent" ;
        rh:long_name = "Relative humidity" ;
// global attributes
    :title = "Simple example, lacks some conventions" ;
data:
    rh =
        2, 3, 5, 7, 11, 13, 17, 19,
        23, 29, 31, 37, 41, 43, 47,
        53, 59, 61, 67, 71, 73, 79, 83, 89 ;
}
```

Let's zoom in...



```
netcdf example { // example of CDL notation
dimensions:
    lon = 3 ;
    lat = 8 ;

variables:
    float rh(lon, lat) ;
        rh:units = "percent" ;
        rh:long_name = "Relative humidity" ;

    // global attributes
        :title = "Simple example, lacks some
conventions" ;

data:
    rh =
        2, 3, 5, 7, 11, 13, 17, 19,
        23, 29, 31, 37, 41, 43, 47,
        53, 59, 61, 67, 71, 73, 79, 83, 89 ;
}
```



```
netcdf example { // example of CDL notation
```

```
dimensions:
```

```
    lon = 3 ;
```

```
    lat = 8 ;
```

```
variables:
```

```
    float rh(lon, lat) ;
```

```
        rh:units = "percent" ;
```

```
        rh:long_name = "Relative h
```

```
// global attributes
```

```
    :title = "Simple example,  
conventions" ;
```

```
data:
```

```
    rh =
```

```
        2, 3, 5, 7, 11, 13, 17, 19,
```

```
        23, 29, 31, 37, 41, 43, 47,
```

```
        53, 59, 61, 67, 71, 73, 79, 85, 89 ;
```

```
}
```

This example specifies a netCDF dataset with:

- 2 dimensions (**lon** and **lat**)
- 1 variable (**rh**)
- 2 variable attributes (**units** and **long\_name**)
- 1 global attribute (**title**), and 24 data values for the variable.



# NCL example: explained

- Only one variable, but multiple variables are allowed.
- CDL comments follow "//" symbols. They are not part of netCDF data and will not be recorded in a transformation.
- You can use the **ncdump** utility to get the CDL form of a binary netCDF file.
- You can use the **ncgen** utility to generate a binary netCDF file from CDL.
- This simple example neglects some recommended best practices for netCDF data.



# Writing a NetCDF file with ncgen

ncgen can be used to:

- Generate a NetCDF file

```
ncgen -o mydata.nc mydata.cdl
```

- Generate code that will produce a NetCDF file

- Fortran:

```
ncgen -f mydata.cdl > mydata.f
```

- C:

```
ncgen -c mydata.cdl > mydata.c
```

# Further information

- NetCDF:

<https://www.unidata.ucar.edu/software/netcdf/>

- CDL:

<https://www.unidata.ucar.edu/software/netcdf/netcdf/CDL-Syntax.html>

- Best practices to writing NetCDF files:

<https://www.unidata.ucar.edu/software/netcdf/docs/BestPractices.html>