Binding Regulations for Storing Data as netCDF Files

Gisbert Breitbach, Beate Geyer, Ulrike Kleeberg, Elke Meyer, Reiner Onken
Helmholtz-Zentrum hereon GmbH

Version 1.4

May 2021

Table of Contents

1. Introduction

Files are written and used in netCDF format at Hereon in several institutes. Files originating from Hereon should be written in uniform nomenclature. This promotes and facilitates exchange and dissemination. There are specifications in regard to naming both attributes as well as variables (naming conventions). In addition, there are specifications regarding which attributes should be added to certain variables. Not all specifications are mandatory and, in certain cases, adjustments are also necessary.

This information essentially refers to the agreements within the netCDF *Climate and Forecast (CF) Metadata Conventions*[1] *(hereafter referred to as the CF Conventions).* The relevant binding regulations are summarised and expanded in the following document.

In addition to the CF Conventions, there are other agreements that are only relevant to parts of Hereon. When dealing with observation data intended for CMEMS[2] , the rules, for example, for the OceanSITES Format Reference[3] should be followed. These are not, however, used internally at Hereon. In addition, the SeaDataNet netCDF conventions[4], which are themselves based on the CF Conventions, may be relevant to other users.

After saving the data according to the specified regulations, the files should be checked using the "CF Conventions Compliance Checker for NetCDF Format"[5]. In addition, a second person should review whether the special Hereon regulations have been followed.

After successful checks, measurement data can be saved in the COSYNA data area (contact: Gisbert Breitbach) and model data can, for example, be saved in the DKRZ's CERA database with a DOI designation.

[1] http://cfconventions.org/index.html
[2] http://marine.copernicus.eu
[3] http://www.oceansites.org/docs/oceansites_data_format_reference_manual.pdf
[4] https://www.seadatanet.org/content/download/637/3338/file/SDN_D85_WP8_netCDF_files_examples.zip
[5] http://puma.nerc.ac.uk/cgi-bin/cf-checker.pl

2. General Specifications

Provided here is general information that has nothing directly to do with the netCDF format itself but should, however, be taken into consideration.

2.1. File Names

- NetCDF files are designated with the extension .nc
- The file name begins with a letter or a letter sequence. The letter sequence should allow users to draw conclusions about the type of data found in the file.
- Customary sequences used thus far (e.g., “ctd” for CTD data, “sf” for ScanFish data) will be retained. Specification should still be designated for other devices.
- When dealing with model data, either the model name with the version designation or the ExperimentID should be used in the first segment of the filename.
- Spaces, umlauts, “ß”, special characters (except underscore “_”, dash “-” and periods) are not to be used in file names.
- File names are encoded with the starting date (and possibly the end date) of the data contained in the file according to the format: YYYYMMDDHH. The date provided complies with the scope of data. A yearly file usually only contains the numerical year listed as YYYY. If appropriate, minutes can be added in the “mm” format and seconds can be added as “SS”. Unresolved timescales can be omitted. Additional information can be added after the date and after “_” or “.”, such as variable designations, region, statistical processing abbreviations.

Example 1
ctd200904171324.nc for a CTD profile that began on April 17th, 2009 at 13:24.

Example 2
WAVE2016080312_gb.nc as wave model calculation output for 12:00 on 03.08.2016 in the German Bight (gb)

Example 3
cD3_0025_ERAi.1948-2015.T_2M.DB.mm.nc as a time series from COSMO-CLM simulation output for 2m air temperature reduced to the German Bight and *monthly mean*.

2.2. Naming Conventions

- *Variables, dimensions and attribute names* begin with a letter and consist of letters, numbers and underscores.
- Spaces, umlauts, ß, special characters (except underscore “_”) are not allowed.
- Upper and lower case letters are relevant. See Example 3.

Names must not be differentiated through case sensitive designations alone.
Variable names should adhere as much as possible to international standards. For observable marine quantities, the BODC’s[6] P09 database is, for example, available.

[6] http://vocab.nerc.ac.uk/collection/P09/current/

Example 1
pressure, longitude, DRYT, air_pressure, FlugHoehe are valid *names*. Of the listed variable names, only DRYT (dry bulb temperature) is included in P09.

Example 2
geogr. Breite, air pressure, Flughöhe, $velocity are invalid *names*.

Example 3
Variable names such as PRESSURE and pressure are not to be used together in the same netCDF file because they can only be differentiated in terms of case sensitivity. This also applies to *dimensions* and *attribute names*.

3. Information on Entire Data Set (global attributes)

The global attributes store descriptive information pertaining to the entire dataset.
A distinction is made between mandatory attributes (Chapter 3.1) and optional attributes (Chapter 3.2). The optional attributes are necessary to accommodate the different data sources/disciplines.

Attribute names can generally be freely chosen, but there are *attribute names* that are already reserved and can only be used for specified purposes. These are as follows:

```
add_offset, ancillary_variables, axis, bounds, calendar, cell_measures,
cell_methods, climatology, comment, compress, Conventions, coordinates,
_FillValue, flag_masks, flag_meanings, flag_values, formula_terms, grid_mapping,
history, institution, leap_month, leap_year, long_name, missing_value,
month_lengths, positive, references, scale_factor, source,
standard_error_multiplier, standard_name, title, units, valid_max, valid_min,
valid_range.
```

For the meaning of the attribute names listed above, please see Appendix A in *netCDF Climate and Forecast (CF) Metadata Conventions*[7].

3.1. **Mandatory Global Attributes**

Global attributes are the netCDF file's metadata. Only information that applies to the file as a whole is to be included here.

Attribute	Description/Format	Specification/Example
Conventions	- fixed -	"CF-1.8"
institution	- fixed -	"Helmholtz-Zentrum Hereon, Institute of Coastal Systems, Germany"
title	Any text	
source	Represents the source and method used to produce the original data. For data generated by models, the model name and version should be listed here	"cosmo_100614_4.14_clm2 int2lm_120824_1.20_clm1", "remote sensing", "in-situ"
creation_date	Information that pertains to the creation	"2017-06-06T15:00:29Z"

[7] http://cfconventions.org/cf-conventions/cf-conventions.html#attribute-appendix

Attribute	Description/Format	Specification/Example
	date of the netCDF file	"2017-06-06T15:00:29+02:00"
originator	Creator of the data set	"John Smith"
contact	Person, email or URL	"http://coastmod.hereon.de"
crs	coordinate reference system: WGS84 is the default	"EPSG:4326"

3.2. Optional Global Attributes

Optional global attributes are not relevant at the Hereon for all netCDF files, but agreements can be made as necessary in individual groups. If the described attribute is relevant, then this attribute should be used consistently.
The `comment` attribute should be used in particular to describe the file contents.
The `carrier` attribute describes the carrier on which a platform is installed temporarily—for example, RV Prandtl.
In COSYNA the `platform` attribute is mandatory and is usually specified. An example of `platform` would be "Ferrybox on RV Polarstern".
The origin of a netCDF file is described in `lineage`. This is usually an original file in another format.
The individual sequence of processing should be described in `processingSteps`.
`StartTime` and `StopTime` denote the first and last time values and facilitate quick and easy access to these values—for example via OPeNDAP.
Should the file be disseminated externally or published, the user rights and obligations are outlined in the `distribution_statement`. The content should be coordinated within the respective institute division.
In `history` one can directly record what was done to the file to reach its current state. Tools like the CDOs or NCOs document the workflow in the `history` automatically. One can also refer to an external file or URL here.
PI should designate the scientist responsible for the entire project to which the file belongs.
The following attributes are available for describing geographical location: `geospatial_lon_min/_max`, `geospatial_lat_min/_max` or `Bbox`. They are, for example, necessary for metadata catalogue inquiries.

Attribute	Description/Format	Example
comment	A short description of the file contents; essentially an abstract.	
campaign	Campaign ID (for observations)	"belawatt071"
experiment_id	Experiment ID (for models)	"cD3_011_ERAi"
carrier	Any text.	"RV Prandtl"
platform	Any text. The specified name is required for COSYNA platforms	"Scanfish01"
profileType		"DownCast"
lineage	Origin of the file can be written as any text (i.e., no formatting rules)	
processingSteps	Any text; example format: "{original data}{Plausibility check using defaults}{5-point median filter}{0.1-dbar block median}"	

Attribute	Description/Format	Example
processingSoftware	Name und version. If using several programs, names should be separated by ";"	"GALATON V0.97"
StartTime		"2007-03-13T07:35:10Z"
StopTime		"2007-03-13T07:40:28Z"
Distribution_ statement	Coordinated text, Example A3	
history	This should be a sort of logbook. This logbook can be an external file. According to CF conventions, it is a list of the applications that have modified the original data.	
references	E.g., DOI	
contact_email	Email address (Hereon)	
StartLatitude	Float	54.7457933333333
StopLatitude		54.7451966666667
StartLongitude	Relevant for transects	8.30397666666667
StopLongitude		8.30919166666667
geographicName		"hoernumtief"
station		"KormoranHoernum"
castID		"CTD001DW"
PI	Standard first and last name	"Person"
operator	Standard first and last name	"Volker Schmidt"
contributor	Standard first and last name	"Anna Mayr, Tim Schmalbach"
forcing		"COSMO CLM 5.0"
forcingResolution		"0.11 degrees"
boundaryCondition	Any text	
modelNest		"CD16"
geospatial_lon_min	Data in decimal degrees (float) • Latitude units = degrees north • Longitude units = degrees east	5.3457
geospatial_lon_max		10.1234
geospatial_lat_min		53.1234
geospatial_lat_max		56.1234
Bbox	alternative to geospatial_l*. Sequence: lonMin, latMin, lonMax, latMax	5.3,53.1,10.1,56.1
imo_platform_code	International Maritime Organisation platform code	"9144263"
ices_platform_code	ICES platform code	"58LY"
licence	Licence of the submitted data	

3.3. **Dimensions**

The data is stored as multi-dimensional fields in netCDF files. Decisive structural elements are the *dimensions.* They are used to define the internal coordinate structure of the data sets (*variables*). They contain information on the spatial dimensions (nx,ny,nz) as well as the time dimension (time). These are explained under *variables* (see Section 4).

- A *variable* can have any number of *dimensions*.
- All *dimensions* must have different *names*.

- A restriction to four *dimensions* is recommended.
- Dimensions of a variable should be arranged in the relative order of date/time, height/depth, latitude, longitude (T, Z, Y, X). If this is not possible, (e.g., for spectra), other reasonable sequences should be created.

Additional *dimensions* must be to the left of (T,Z,Y,X).
If time series are stored in netCDF files, then one must keep in mind that the series can be continued by adding new times. The time variable dimension should be "UNLIMITED" to ensure expansion of the time series without difficulties. It is then referred to as the *record dimension.*

4. Variables und Associated Attributes

In principle, the variable name in the netCDF file can be freely chosen (subject to the notes in Chapter 2.2), but the variable must be described using the corresponding *attributes* (see table in 4.2). Many communities have established standards for these names, such as the often mentioned P09 database[8].
Both mandatory and optional attributes exist.

4.1. Data Types

Only the netCDF *data types* char, byte, short, float, real, double are allowed.

4.2. Mandatory Attributes

Information regarding the unit (`units`) as well as the definition (`standard_name`[9] and/or `long_name`) must be provided for each variable. In rare cases, it does not make sense to request a `standard_name`. Only then is the *attribute* `long_name` mandatory and the `long_name` should be constructed like a standard name.

Attribute	Description/Format	Example
units	"string", use SI unit if possible This also applies to dimensionless units, then use given format: "1", "1e±2", "1e±3", "1e±6", ... "1e±24"	"m" "1"
standard_name[9]	The *standard_name* stems from **http://cfconventions.org/standard-names.html**. In the case of *ancillary variables, it is mandatory to* use the *Standard Name Modifiers*[10] (e.g.,	"sea_water_salinity"

[8] http://seadatanet.maris2.nl/v_bodc_vocab_v2/browse.asp?order=conceptid&formname=search&screen=0&lib=p09&v1_0=conceptid%2Cpreflabel%2Caltlabel%2Cdefinition%2Cmodified&v2_0=0&v0_1=&v1_1=conceptid&v2_1=3&v0_2=&v1_2=preflabel&v2_2=3&v0_3=&v1_3=altlabel&v2_3=3&v0_4=&v1_4=modified&v2_4=9&v0_5=&v1_5=modified&v2_5=10&v1_6=&v2_6=&v1_7=&v2_7=
[9] If this does not already exist in the cf-conventions table, it should be requested. Beate Geyer can provide further information on standard names and how to apply for them and https://github.com/cf-convention/discuss/.
[10] http://cfconventions.org/Data/cf-conventions/cf-conventions-1.6/build/cf-conventions.html#standard-name-modifiers

Attribute	Description/Format	Example
	status_flag).	
long_name	Any text If no `standard_name` exists, then the `long_name` should be constructed like a standard name.	"Salinity from Seabird CTD"

Example: (dimensionless value, `standard_name` exists, `long_name` chosen freely)

```
double Salinity(levels) ;
        Salinity:units = " 1e-3 " ;
        Salinity:standard_name = "sea_water_salinity" ;
        Salinity:long_name = "salinity from SeaBird CTD" ;
```

If the fitting *Standard Name Modifier* does not exist in the table, it must be requested with cf-metadata similar to the procedure for the *standard_name* (see standard_name[9]).

4.3. Optional Attributes

Optional attributes should supplement the variable description in a meaningful way.

Attribute	Description/Format	Example
_FillValue	Default grid value becomes transparent in many visualisations	-32768.f Note: do not use NaN in ncWMS2 application
comment	Any text	"3 out of 4 HPBL estimates agree with the given range"
coordinates	Using data in projections other than WGS84; the variables “ion” and “lat” must be contained in the file	"lon lat" or "lon lat height_10m"
grid_mapping	Indicates the projection used	"rotated_pole"
cell_methods	For example, in the case of non-instantaneous data, the aggregation type is to be specified. This requires the *attribute* `time_bnds` for the time variable in which the aggregation start and end times are specified.	"time: mean"

Example for using the attribute `cell_methods`:

```
dimensions:
        time = UNLIMITED ; // (744 currently)
        bnds = 2 ;
        rlon = 234 ;
        rlat = 228 ;
variables:
    double time(time) ;
        time:standard_name = "time" ;
        time:long_name = "time" ;
        time:units = "seconds since 1948-01-01 00:00:00" ;
        time:bounds = "time_bnds" ;
```

```
double time_bnds(time, bnds) ;
        time_bnds:long_name = "time bounds" ;
        time_bnds:units = "seconds since 1948-01-01 00:00:00" ;
float TOT_PREC(time, rlat, rlon) ;
        TOT_PREC:standard_name = "precipitation_amount" ;
        TOT_PREC:long_name = "total precipitation amount" ;
        TOT_PREC:units = "kg m-2" ;
        TOT_PREC:cell_methods = "time: sum" ;
```

4.4. Handling Missing Data

- Missing values in the data set are indicated only by the `_FillValue` *attribute*.

The *value* `_FillValue` can be set to NaN or NaN.f (with float). In some instances, other values have been agreed upon for naming the missing values due to technical reasons (1.0E20 is a frequently used standard). Using an actual value has the advantage that in visualizations such as ncWMS, the pixels with `_FillValue` are displayed transparently. Files with a `_FillValue` = NaN cannot be used for ncWMS2 because ncWMS2 no longer supports this (see Section. 4.1).

4.5. Coordinates

Coordinates are the variables corresponding to a dimension. While the correct information on the coordinates is immensely important, the explanations provided here are general. Different examples on coordinate information from various Hereon datasets, however, can be found in the Appendix.
The dimensions assigned to degrees of longitude and latitude are to be assigned indices from west to east and south to north in ascending order.

4.5.1. Latitude

- The variable name is "lat" and the format is float.
- The *attribute* `units` is degrees_north[11].
- The *attribute* `standard_name` is "latitude".
- The *attribute* `axis` (optional) contains the *value* `Y`.
- For transformed grids (e.g., coordinate systems with rotated poles) only degrees is used as the `units`.

Example:

```
dimensions:
        lon = 69 ;
        lat = 64 ;
float lat(lat);
        lat:units = "degrees_north" ;
        lat:standard_name = "latitude" ;
```

4.5.2. Longitude

- The variable name is "lon" and the format is float.

[11] http://cfconventions.org/cf-conventions/cf-conventions.html#_independent_latitude_longitude_vertical_and_time_axes

- The *attribute* `units` is degree_east.[12]
- The *attribute* `standard_name` is "longitude".
- The *attribute* `axis` (optional) contains the *value* X.
- For transformed grids (e.g., coordinate systems with rotated poles) only degrees is used as the `units`.

Example:
```
dimensions:
      lon = 69 ;
      lat = 64 ;
float lon(lon);
          lon:units = "degrees_east" ;
          lon:standard_name = "longitude" ;
```

4.5.3. Height and Depth (vertical coordinates)

- The `positive` *attribute* is mandatory as there is often disagreement concerning the direction of the vertical coordinate.
- The direction of ascending coordinate values must be provided with the `positive` ***attribute*** and the possible *values* up or down.

General Example:
```
axis_name:units = "m" ;
axis_name:positive = "down" ;
```

Example 1:
```
double Pressure(levels) ;
          Pressure:units = "dbar" ;
          Pressure:standard_name = "sea_water_pressure" ;
          Pressure:positive = "down" ;
```

The coordinate origin is at the sea surface in this case; pressure increases with depth.

Example 2:
```
double Altimeter(levels) ;
          Altimeter:units = "m" ;
          Altimeter:positive = "up" ;
          Altimeter: standard_name = "height_above_sea_floor" ;
          Altimeter:long_name = "distance to bottom" ;
```

The coordinate origin is at the seafloor in this case. The value increases closer to the sea surface.

Example 3:
```
double Depth(levels) ;
         Depth:units = "m" ;
         Depth:positive = "down" ;
         Depth:long_name = "water depth relative to sea surface" ;
         Depth:standard_name = "depth" ;
```

The coordinate origin is at the sea surface in this case.

[12] http://cfconventions.org/cf-conventions/cf-conventions.html#_independent_latitude_longitude_vertical_and_time_axes

Example 4:

```
double time(time) ;
        time:standard_name = "time" ;
        time:long_name = "time" ;
        time:units = "seconds since 1979-01-01 00:00:00" ;
float rlon(rlon) ;
        rlon:standard_name = "grid_longitude" ;
        rlon:long_name = "rotated longitude" ;
        rlon:units = "degrees" ;
rlat(rlat) ;
        rlat:standard_name = "grid_latitude" ;
        rlat:long_name = "rotated latitude" ;
        rlat:units = "degrees" ;
float height_2m ;
        height_2m:standard_name = "height" ;
        height_2m:long_name = "height above the surface" ;
        height_2m:units = "m" ;
        height_2m:positive = "up" ;
float T_2M(time, rlat, rlon) ;
        T_2M:standard_name = "air_temperature" ;
        T_2M:long_name = "2m temperature" ;
        T_2M:units = "K" ;
        T_2M:coordinates = "lon lat height_2m" ;
```

Although the temperature is given at 2m above ground, there is no separate *dimension* for it.

4.5.4. Time Coordinate

The `units` *value* is, for example:

- "days since YYYY-MM-DD HH:mm:SS" or
- suitable formulations[13]

There is no time zone indicator! UTC is therefore automatically set as the time coordinate.
It is most preferable to use *double* as the *type,* otherwise there is a risk of unresolved seconds.

The `calendar` *attribute* should be set to proleptic-gregorian or standard. In simplified cases, such as calculations without a leap year or only months with thirty days, the corresponding calendar attribute is taken into account: "noleap" and/or "360_day".

```
dimensions:
        time = UNLIMITED ; // (744 currently)
variables:
double time(time) ;
        time:standard_name = "time" ;
        time:long_name = "time" ;
        time:calendar = "proleptic_gregorian" ;
```

4.6. **Quality Flags**

Quality flags are usually only used for observation data. The ultimate goal in regard to quality is error estimation and indication. This could then also apply to model output.
The following OceanSites regulations[14] are mandatory for data intended for the Copernicus Marine Environment Monitoring Service and are managed there as well.

[13] Further information: http://cfconventions.org/cf-conventions/cf-conventions.html#time-coordinate
[14] http://www.oceansites.org/docs/oceansites_data_format_reference_manual.pdf

If possible, observation data should always be denoted with *quality flags*. For each data point, these *quality flags* indicate whether the data point has undergone quality control and the specific result of this quality control. The *quality flag* does not provide any information whether the data point is close to reality.
In the netCDF file, the *quality flag* is an additional variable that has the same name as the variable for which the quality specification applies, but with an appended string "_QC" (OceanSITES) or "_qc" (CF conventions, see the *Flags* chapter[15]). "QC" stands for *quality control* for the respective parameter. Example names are TIME_QC, POSITION_QC or TEMP_QC. For these variables' attributes, the following applies:

```
conventions = "OceanSITES reference table 2" ;
long_name = "quality flag" ;
_FillValue = -128b ;
valid_min = 0b ;
valid_max = 9b ;
flag_values = "0, 1, 2, 3, 4, 5, 7, 8, 9" ;
flag_meanings = "no_qc_performed good_data probably_good_data
bad_data_that_are_potentially_correctable bad_data value_changed not_used nominal_value
interpolated_value missing_value" ;
```

The data variable refers to the *Quality Flag* by means of the `ancillary_variable` attribute.

Example:

```
float PSAL(TIME, DEPTH) ;
        PSAL:units = "1e-3" ;
        PSAL:standard_name = "sea_water_practical_salinity" ;
        PSAL:_FillValue = -999.f ;
        PSAL:DM_indicator = "R" ;
        PSAL:ancillary_variables = "PSAL_QC" ;
        PSAL:valid_min = 0 ;
        PSAL:valid_max = 40 ;
        PSAL:long_name = "Practical salinity" ;
        PSAL:cell_methods = "TIME:point DEPTH:point LATITUDE:point LONGITUDE:point" ;
byte PSAL_QC(TIME, DEPTH) ;
        PSAL_QC:flag_values = "0, 1, 2, 3, 4, 5, 7, 8, 9" ;
        PSAL_QC:flag_meanings = "no_qc_performed good_data probably_good_data
bad_data_that_are_potentially_correctable bad_data value_changed not_used nominal_value
interpolated_value missing_value" ;
        PSAL_QC:_FillValue = -128b ;
        PSAL_QC:long_name = "quality flag" ;
        PSAL_QC:conventions = "OceanSITES reference table 2" ;
        PSAL_QC:valid_min = 0b ;
        PSAL_QC:valid_max = 9b ;


byte current_speed_qc(time, depth, lat, lon) ;
   current_speed_qc:long_name = "Current Speed Quality" ;
   current_speed_qc:standard_name = "status_flag" ;
   current_speed_qc:_FillValue = -128b ;
   current_speed_qc:valid_range = 0b, 2b ;
   current_speed_qc:flag_values = 0b, 1b, 2b ;
   current_speed_qc:flag_meanings = "quality_good sensor_nonfunctional
                                outside_valid_range" ;
```

[15] http://cfconventions.org/cf-conventions/cf-conventions.html#flags

Appendix: Sample Collection on Coordinate Information

Example A1: In this example, the *variable* WaterLevel has the dimensions time, latitude and longitude. Here the *coordinate* variables are `time`, `longitude` and `latitude`, with *names* that are identical to the *dimension names*. That is, it is a three-dimensional grid and/or a time series of surfaces.

```
dimensions:
        time = UNLIMITED ; // (177 currently)
        longitude = 69 ;
        latitude = 64 ;
variables:
        double time(time) ;
                time:units = "days since 1970-01-01 00:00:00" ;
                time:standard_name = "time" ;
        double longitude(longitude) ;
                longitude:units = "degrees_east" ;
                longitude:standard_name = "longitude" ;
        double latitude(latitude) ;
                latitude:units = "degrees_north" ;
                latitude:standard_name = "latitude" ;
        double WaterLevel(time, latitude, longitude) ;
                WaterLevel:units = "m" ;
               WaterLevel:standard_name=” sea_surface_height_above_sea_level”
```

Example A2: The dataset contains a trajectory—for example, glider data.

```
dimensions:
        time = UNLIMITED ; // (177 currently)
variables:
        double time(time) ;
                time:units = "days since 1970-01-01 00:00:00" ;
                time:standard_name = "time" ;
        double longitude(time) ;
                longitude:units = "degrees_east" ;
                longitude:standard_name = "longitude" ;
        double latitude(time) ;
                latitude:units = "degrees_north" ;
                latitude:standard_name = "latitude" ;
       double Pressure (time) ;
                Pressure:units = "dbar" ;
                Pressure:positive = "down"
                Pressure:standard_name = "sea_water_pressure" ;
     double Salinity(time) ;
                Salinity:units = "1" ;
                Salinity:standard_name = "sea_water_salinity" ;
```

Example A3: Example of a COSYNA Distribution Statement.

Distribution_statement:
"COSYNA provides free public access to observational and model data. For near-real time data only limited quality control is performed. Accessible data are subject to change due to delayed processing and quality control. For all COSYNA data no warranty is made, expressed or implied, regarding the accuracy or validity of the data, or regarding the suitability of the data for any particular application. Guidelines for data use and publication: the right of priority use is reserved. If you intend to use COSYNA data please consult the COSYNA coordinator. The following conditions apply to any use of COSYNA data. The user agrees to acknowledge the provision of the data referring to the data originator and to include the following statement: \"The data were provided by the COSYNA system operated by Helmholtz-Zentrum hereon GmbH\"; to offer co-authorship to the data originator(s) if the provided COSYNA data are a principal component of the publication; not to use any data commercially without prior approval of the data owner(s). In addition, the data policies of the respective third party institution(s) must be followed in regard to any type of third party data provided by COSYNA."

Example A4: Output from COSMO-CLM.

```
dimensions:
        time = UNLIMITED ; // (2919 currently)
        rlat = 170 ;
        rlon = 148 ;
        nb2 = 2 ;
variables:
        float TOT_PREC(time, rlat, rlon) ;
                TOT_PREC:standard_name = "precipitation_amount" ;
                TOT_PREC:long_name = "total precipitation amount" ;
                TOT_PREC:units = "kg m-2" ;
                TOT_PREC:grid_mapping = "rotated_pole" ;
                TOT_PREC:coordinates = "lon lat" ;
                TOT_PREC:cell_methods = "time: sum" ;
        float lat(rlat, rlon) ;
                lat:standard_name = "latitude" ;
                lat:long_name = "latitude" ;
                lat:units = "degrees_north" ;
        float lon(rlat, rlon) ;
                lon:standard_name = "longitude" ;
                lon:long_name = "longitude" ;
                lon:units = "degrees_east" ;
        float rlat(rlat) ;
                rlat:standard_name = "grid_latitude" ;
                rlat:long_name = "rotated latitude" ;
                rlat:units = "degrees" ;
        float rlon(rlon) ;
                rlon:standard_name = "grid_longitude" ;
                rlon:long_name = "rotated longitude" ;
                rlon:units = "degrees" ;
        char rotated_pole ;
                rotated_pole:long_name = "coordinates of the rotated North Pole" ;
                rotated_pole:grid_mapping_name = "rotated_latitude_longitude" ;
                rotated_pole:grid_north_pole_latitude = 54.2f ;
                rotated_pole:grid_north_pole_longitude = -57.7f ;
        double time(time) ;
                time:standard_name = "time" ;
                time:long_name = "time" ;
                time:units = "seconds since 1948-01-01 00:00:00" ;
                time:calendar = "proleptic_gregorian" ;
                time:bounds = "time_bnds" ;
        double time_bnds(time, nb2) ;
                time_bnds:units = "seconds since 1948-01-01 00:00:00" ;

// global attributes:
                :title = "COSMO-CLM simulation for Bohai, Yellow and East China Sea" ;
                :institution = "Helmholtz-Zentrum Hereon, Institute of Coastal Systems, Germany" ;
                :source = "cosmo_100614_4.14_clm2 int2lm_120824_1.20_clm1" ;
```

:project_id = "-" ;
:experiment_id = "BH_LONG_ERAin" ;
:realization = 1 ;
:Conventions = "CF-1.6" ;
:conventionsURL = "http://cfconventions.org/" ;
:contact = "http://coastmod.hereon.de" ;
:references = "doi:10.1002/2015JD024177 and http://www.clm-community.eu/" ;
:creation_date = "2013-09-13T18:45:45" ;
:originator = "Delei Li" ;
:crs = "WGS84" ;
:geospatial_lon_min = 117.f ;
:geospatial_lat_min = 30.7f ;
:geospatial_lat_max = 41.3f ;
:geospatial_lon_max = 128.f ;

Example A5:

netcdf DWD_Radiosondes_05839.20080101_20081231 {
dimensions:
obs = UNLIMITED ; // (957166 currently)
profile = 774 ;
variables:
float profile(profile) ;
profile:cf_role = "profile_id" ;
int parentIndex(obs) ;
parentIndex:long_name = "index of profile" ;
parentIndex:instance_dimension = "profile" ;
double time(obs) ;
time:standard_name = "time" ;
time:units = "hours since 1970-01-01 00:00:00" ;
float lat(profile) ;
lat:standard_name = "latitude" ;
lat:long_name = "latitude" ;
lat:units = "degree_north" ;
float lon(profile) ;
lon:standard_name = "longitude" ;
lon:long_name = "longitude" ;
lon:units = "degree_east" ;
float alt(profile) ;
alt:standard_name = "altitude" ;
alt:long_name = "height above geoid" ;
alt:units = "m" ;
alt:coordinates = "lon lat" ;
float PS(profile) ;
PS:standard_name = "surface_air_pressure" ;
PS:long_name = "ground air pressure" ;
PS:units = "hPa" ;
PS:coordinates = "startTime lon lat" ;
PS:_FillValue = -1.e+20f ;
float T_2M(profile) ;
T_2M:standard_name = "air_temperature" ;
T_2M:long_name = "ground air temperature" ;
T_2M:units = "degree_C" ;
T_2M:coordinates = "startTime lon lat" ;
T_2M:_FillValue = -1.e+20f ;
float RELHUM_2M(profile) ;
RELHUM_2M:standard_name = "relative_humidity" ;
RELHUM_2M:long_name = "ground relative humidity" ;
RELHUM_2M:units = "%" ;
RELHUM_2M:coordinates = "startTime lon lat" ;
RELHUM_2M:_FillValue = -1.e+20f ;
float reportingTime(profile) ;
reportingTime:standard_name = "time" ;
reportingTime:long_name = "reporting time" ;
reportingTime:units = "hours since 1970-01-01 00:00:00" ;
reportingTime:_FillValue = -1.e+20f ;
float startTime(profile) ;
startTime:standard_name = "time" ;
startTime:long_name = "start of sounding" ;

```
        startTime:units = "hours since 1970-01-01 00:00:00" ;
        startTime:_FillValue = -1.e+20f ;
    float CAPE(profile) ;
        CAPE:standard_name = "atmosphere_convective_available_potential_energy" ;
        CAPE:long_name = "convective available potential energy" ;
        CAPE:units = "J kg-1" ;
        CAPE:coordinates = "reportingTime lon lat" ;
        CAPE:_FillValue = -1.e+20f ;
    float ae_lat(obs) ;
        ae_lat:standard_name = "latitude" ;
        ae_lat:long_name = "sounding latitude" ;
        ae_lat:units = "degree_north" ;
        ae_lat:coordinates = "time" ;
        ae_lat:_FillValue = -1.e+20f ;
    float ae_lon(obs) ;
        ae_lon:standard_name = "longitude" ;
        ae_lon:long_name = "sounding longitude" ;
        ae_lon:units = "degree_east" ;
        ae_lon:coordinates = "time" ;
        ae_lon:_FillValue = -1.e+20f ;
    float FIH(obs) ;
        FIH:standard_name = "geopotential_height" ;
        FIH:long_name = "geopotential height" ;
        FIH:units = "m" ;
        FIH:coordinates = "time ae_lon ae_lat" ;
        FIH:_FillValue = -1.e+20f ;
    float P(obs) ;
        P:standard_name = "air_pressure" ;
        P:long_name = "air pressure" ;
        P:units = "hPa" ;
        P:coordinates = "time ae_lon ae_lat" ;
        P:_FillValue = -1.e+20f ;
    float T(obs) ;
        T:standard_name = "air_temperature" ;
        T:long_name = "air temperature" ;
        T:units = "degree_C" ;
        T:coordinates = "time ae_lon ae_lat FIH" ;
        T:_FillValue = -1.e+20f ;
    float RELHUM(obs) ;
        RELHUM:standard_name = "relative_humidity" ;
        RELHUM:long_name = "relative humidity" ;
        RELHUM:units = "%" ;
        RELHUM:coordinates = "time ae_lon ae_lat FIH" ;
        RELHUM:_FillValue = -1.e+20f ;
    float TD(obs) ;
        TD:standard_name = "dew_point_temperature" ;
        TD:long_name = "dew point temperature" ;
        TD:units = "degree_C" ;
        TD:coordinates = "time ae_lon ae_lat FIH" ;
        TD:_FillValue = -1.e+20f ;
    float WSS(obs) ;
        WSS:standard_name = "wind_speed" ;
        WSS:long_name = "wind speed" ;
        WSS:units = "m s-1" ;
        WSS:coordinates = "time ae_lon ae_lat FIH" ;
        WSS:_FillValue = -1.e+20f ;
    float WDIR(obs) ;
        WDIR:standard_name = "wind_from_direction" ;
        WDIR:long_name = "wind direction" ;
        WDIR:units = "degree" ;
        WDIR:coordinates = "time ae_lon ae_lat FIH" ;
        WDIR:_FillValue = -1.e+20f ;

// global attributes:
        :institution = "DWD" ;
        :station = "Emden" ;
        :station_id = "05839" ;
        :source = "measurements: radiosonde ascents" ;
        :instrument = "Radiosonde RS92-SGP" ;
```

:instrument_type = "MSGRT" ;
:featureType = "profile" ;
:Conventions = "CF-1.7" ;
:conventionsURL = "http://www.unidata.ucar.edu/packages/netcdf/conventions.html" ;
:download_site =
"ftp://opendata.dwd.de/climate_environment/CDC/observations_germany/radiosondes/high_resolution/historical" ;
:licence = "Geodatennutzungsverordnung, GeoNutzV,
ftp://opendata.dwd.de/climate_environment/CDC/Terms_of_use.pdf" ;
:creation_date = "transformation to netCDF:2019-11-08 13:52:52 +0100" ;