Conventions for the standardization of NetCDF files

Sponsored by the "Cooperative Climate Research Virtual Data Center", a NOAA/university cooperative for the sharing and distribution of global atmospheric and oceanographic research data sets

10 February 1995 – version 1.0
13 March 1995 – minor editorial repairs

University participants:

- CIRES (University of Colorado Joint Institute, Boulder CO)
- JISAO (University of Washington Joint Institute, Seattle WA)

NOAA participants:

- CDC, Asheville NC
- NCDC, Boulder CO
- PMEL, Seattle WA

Relationship to other Standards:

This standard is a set of conventions adopted in order to promote the interchange and sharing of files created with the netCDF Application Programmer Interface (API). This standard is based upon version 2.3 of netCDF. Documentation of the netCDF API may be found in the *"NetCDF Users' Guide, Version 2.3*, April 1993 available from URL *http://www.unidata.ucar.edu/packages/netcdf/* or via anonymous ftp at ftp.unidata.ucar.edu. All conventions named in that document will be adhered to in this standard unless noted to the contrary.

This standard also refers to version 1.7.1 of the udunits standard supported by Unidata. The udunits package is available via anonymous ftp at ftp.unidata.ucar.edu. Included in the udunits package is a file, udunits.dat, which lists collections of unit names. The names given therein and their plural forms will be regarded as acceptable unit names for this standard with the following additions and deletions:

- "degrees" – deleted
- "level", "layer", "sigma_level" – added

The unit "degrees" creates ambiguities when attempting to differentiate longitude and latitude coordinate variables; files must use "degrees_east" for units of longitude and "degrees_north" for units of latitude or the alternative "spellings" of those names listed in the sections on longitude and latitude coordinates below. The dimensionless units "level", "layer", and "sigma_level" are sometimes needed when representing numerical model outputs.

The udunits package also supports linear transformation of all units through the syntax "scale_factor unit_name@offset", for example, "0.0005 degC@40000". This syntax, however, is not supported by this standard.

Conventions:

File Name:
NetCDF files should have the file name extension ".nc".

Coordinate Variables:
1−dimensional netCDF variables whose dimension names are identical to their variable names are regarded as "coordinate variables" (axes of the underlying grid structure of other variables defined on this dimension).

Global attributes:
Although not mandatory the attribute "history" is recommended to record the evolution of the data contained within a netCDF file. Applications which process netCDF data can append their information to the history attribute.

Data Variable attributes:

- **long_name −** a long descriptive name (title). This could be used for labelling plots, for example. If a variable has no long_name attribute assigned, the variable name will be used as a default.
- **scale_factor −** If present for a variable, the data are to be multiplied by this factor after the data are read by the application that accesses the data. (see further discussion under the add_offset attribute)
- **add_offset −** If present for a variable, this number is to be added to the data after it is read by the application that accesses the data. If both scale_factor and add_offset attributes are present, the data are first scaled before the offset is added. The attributes scale_factor and add_offset can be used together to provide simple data compression to store low−resolution floating−point data as small integers in a netCDF file. When scaled data are written, the application should first subtract the offset and then divide by the scale factor. The NOAA cooperative standard is more restrictive than the netCDF Users Guide with respect to the use of the scale_factor and add_offset attributes; ambiguities and precision problems related to data type conversions are resolved by these restrictions. If the scale_factor and add_offset attributes are of the same data type as the associated variable no restrictions apply; the unpacked data is assumed to be of the same data type as the packed data. However, if the scale_factor and add_offset attributes are of a different data type than the associated variable (containing the packed data) then in files adhering to this standard the associated variable may only be of type byte, short, or long. The attributes scale_factor and add_offset (which must match in data type) must be of type float or double. The data type of the attributes should match the intended type of the unpacked data. (It is not advised to

unpack a long into a float as there is a potential precision loss.)

- **_FillValue –** If a scalar attribute with this name is defined for a variable and is of the same type as the variable, it will be subsequently used as the fill value for that variable. The purpose of this attribute is to save the applications programmer the work of prefilling the data and also to eliminate the duplicate writes that result from netCDF filling in missing data with its default fill value, only to be immediately overwritten by the programmer's preferred value. This value is considered to be a special value that indicates missing data, and is returned when reading values that were not written. The missing value should be outside the range specified by valid_range (if used) for a variable. It is not necessary to define your own _FillValue attribute for a variable if the default fill value for the type of the variable is adequate.
- **missing_value –** missing_value is a conventional name for a missing value that will not be treated in any special way by the library, as the _FillValue attribute is. It is also useful when it is necessary to distinguish between two kinds of missing values. The netCDF data type of the missing_value attribute should match the netCDF data type of the data variable that it describes. In cases where the data variable is packed via the scale_value attribute this implies that the missing_value flag is likewise packed. The same holds for the _FillValue attribute. The NOAA cooperative standard does not endorse any particular interpretation of the distinction between missing_value and _FillValue.

Units attribute:
A character array that specifies the units used for the variable's data. Where possible the units attribute should be formatted as per the recommendations in the Unidata udunits package.

Other attributes:
A file will normally contain many attributes that are not standardized in this profile. Those attributes do not represent a violation of this standard in any way. Application programs should ignore attributes that they do not recognize.

Variable names:
Variable names should begin with a letter and be composed of letters, digits, and underscores. It is recommended that variable names be case–insensitive implying that the same case–insensitive name should not be used for multiple variables within a single file.

Rectilinear coordinate systems, only:
The space/time locations of points within the netCDF variables should be the simple ordered tuples formed by associating values from their coordinate axes. Thus, for example, curvilinear coordinate systems in which the coordinate locations must be inferred from other non–coordinate variables or from an equation are not standardized by this netCDF profile.

Number of dimensions:
All netCDF variables will be defined on either one, two ,three, or four dimensions

(the nature of the data will dictate the natural encoding). Where it makes sense single point locations should be encoded as coordinate variables, for example, the latitude and longitude positions of a vertical profile are natural candidates for single point latitude and longitude coordinate variables.
If it is necessary to create a netCDF file with more than 4 dimensions it is recommended that the additional dimension(s) be added "to the left" of the space and time dimensions as represented in CDL. For example

```
        float my_variable(param_value,time,height,lat,lon);
```

would be the recommended representation of a fifth, parameter value, coordinate.

Coordinate variable names:
The names of coordinate variables are not standardized by these conventions (since data sets may in general contain multiple coordinate variables of the same orientation). Coordinate variable names should follow the same general naming rules (above) as other netCDF variables.

Order of dimensions:
If any or all of the dimensions of a variable have the interpretations of "date or time" (a.k.a. "T"), "height or depth" (a.k.a. "Z"), "latitude" (a.k.a. "Y"), or "longitude" (a.k.a. "X") then those dimensions should appear in the relative order T, then Z, then Y, then X in the CDL definition corresponding to the file.

Data type:
The data type of coordinate and non−coordinate variables is unrestricted (byte, short, long, float, and double are all acceptable data types). Although not forbidden by this standard the data type "char", which is functionally identical to "byte", is not recommended as netCDF has reserved the option to modify its behavior in future versions.

Coordinate value ordering:
The coordinate values of a coordinate variable must be either monotonically increasing or monotonically decreasing. However, the coordinate values need not be evenly spaced. Missing values are not permitted in coordinate variables.

Coordinate Variable Attributes:
If a coordinate variable contains longitude, latitude, depth, elevation, date, or time values then the units attribute is mandatory; it is used to determine the orientation of the coordinate variable. The long_name attribute is optional but may be used to enhance clarity and the self−describing nature of the netCDF file. Since coordinate variables may not contain missing values the attributes _FillValue and missing_value may not be used with coordinate variables.

Time or date dimension:
Coordinate variables representing time must always explicitly include the units attribute; there is no default value. A time coordinate variable will be identifiable by its units, alone. The units attribute will be of character type with the string formatted as per the recommendations in the Unidata udunits package version 1.7.1. The

following excerpt from the udunits documentation explains the time unit encoding by example:

```
The specification:

seconds since 1992-10-8 15:15:42.5 -6:00

indicates seconds since October 8th, 1992 at 3 hours, 15
minutes and 42.5 seconds in the afternoon in the time
zone which is six hours to the west of Coordinated
Universal Time (i.e. Mountain Daylight Time). The time
zone specification can also be written without a colon
using one or two-digits (indicating hours) or three or
four digits (indicating hours and minutes).
```

The acceptable units for time are listed in the file udunits.dat. The most commonly used of these strings (and their abbreviations) includes day (d), hour (hr, h), minute (min), second (sec, s), year (yr). Plural forms are also acceptable. The date string may include date alone; date and time; or date, time, and time zone.
It is recommended that the unit "year" not be used as a unit of time. Year is an ambiguous unit as years are of varying length. Udunits defines a year as exactly 365 days.

A time coordinate variable is identifiable from its units string, alone. The udunits routines utScan and utIsTime can be used to make this determination. (*Note that at the time of this writing the author of this draft profile had not tested these routines personally.)

Climatological time:
Coordinate variables representing climatological time (an axis of 12 months, 4 seasons, etc. that is located in no particular year) should be encoded like other time axes but with the added restriction that they be encoded to begin in the year 0000. (Note – at the time of this writing this encoding has not been tested with the udunits package.)

Vertical (height or depth) dimension:
Coordinate variables representing height or depth must always explicitly include the units attribute; there is no default value for the units attribute. The units attribute will be of character type.

The acceptable units for vertical (depth or height) coordinate variables are

- units of pressure as listed in the file udunits.dat. For vertical axes the most commonly used of these include include bar, millibar, decibar, and atmosphere (atm).
- units of length as listed in the file udunits.dat. For vertical axes the most commonly used of these include meter (metre, m), centimeter (cm), decimeter (dm), and feet (ft).
- the (dimensionless) units "level", "layer", or "sigma_level"
- other units listed in the file udunits.dat that may under certain circumstances

reference vertical position such as units of density or temperature.

Plural forms are also acceptable.

The direction of positive, whether up or down, cannot in all cases be inferred from the units. The direction of positive is useful for applications displaying the data. For this reason the new attribute *positive* is defined in this standard. The inclusion of the *positive* attribute is required by this standard if the vertical axis units are not a valid unit of pressure (a determination which can be made using the udunits routine, utScan) −− otherwise its inclusion is optional. The *positive* attribute may have the value "up" or "down" (case insensitive).

For example, if an oceanographic netCDF file encodes the depth of the surface as 0 and the depth of 1000 meters as 1000 then the axis would use attributes as follows:

```
axis_name:units="meters";
axis_name:positive="down";
```

If, on the other hand, the depth of 1000 meters were represented as −1000 then the value of the *positive* attribute would have been "up". If the units attribute value is a valid pressure unit the default value of the *positive* attribute is "down".

A vertical coordinate variable will be identifiable by

- units of pressure; or
- the presence of the *positive* attribute with a value of "up" or "down" (case insensitive).

Latitude dimension:
Coordinate variables representing latitudes must always explicitly include the units attribute; there is no default value for the units attribute. The units attribute will be of character type with the string formatted as per the recommendations in the Unidata udunits package.

The recommended unit of latitude is "degrees_north". Also acceptable are "degree_north", "degree_N", and "degrees_N".

A latitude coordinate variable is identifiable from its units string, alone. The udunits routines utScan can be used to make this determination. (Note that at the time of this writing the author of this draft profile had not tested these routines personally.)

Longitude dimension:
Coordinate variables representing longitudes must always explicitly include the units attribute; there is no default value for the units attribute. The units attribute will be of character type with the string formatted as per the recommendations in the Unidata udunits package.

The recommended unit of longitude is "degrees_east" (eastward positive). Also acceptable are "degree_east", "degree_E", and "degrees_E". The unit "degrees_west" (westward positive) is not recommended because it implies a negative conversion factor from degrees_east.

Longitudes may be represented modulo 360. Thus, for example, −180, 180, and 540 are all valid representations of the International Dateline and 0 and 360 are both valid representations of the Prime Meridian. Note, however, that the sequence of numerical longitude values stored in the netCDF file must be monotonic in a non−modulo sense.

A longitude coordinate variable is identifiable from its units string, alone. The udunits routines utScan can be used to make this determination. (Note that at the time of this writing the author of this draft profile had not tested these routines personally.)

The **example CDL file** which follows contains variable and coordinate definitions (output of "ncdump −h") that illustrate most of the points discussed above. The actual data values are omitted.

The CDL file contains the followind elements:

- a 1−dimensional (vertical axis) atmospheric variable at a point
- a 1−dimensional (vertical axis) variable globally averaged in longitude
- a 2−dimensional packed variable on a latitude/longitude grid
- a 3−dimensional variable on a lat/long/climatological time grid
- two 4−dimensional variables on grids that are staggered in lat/long

---

hankin@pmel.noaa.gov