Pacific Northwest
NATIONAL LABORATORY

PNNL-36159

Wind Data Hub: Data Standards Version 2.1

August 2024

Chitra Sivaraman
Bridget Thrasher
Michael Kieburtz
Kenneth Burk
Maxwell Levin
James McVey
Ekman Kaur

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-36159

0BWind Data Hub: Data Standards Version 2.1

August 2024

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Acronyms and Abbreviations

A2e	Atmosphere to Electrons Initiative
AGL	above ground level
ARM	Atmospheric Radiation Measurement Climate Research Facility
CF	Climate and Forecast metadata conventions
MSL	mean sea level
NetCDF	Network Common Data Format
QC	quality control
UTC	Coordinated Universal Time
VAP	value-added product
WDH	Wind Data Hub

Contents

Acronyms and Abbreviations ii
1.0 Introduction 1
2.0 Dataset Contents 1
2.1 Time span 1
2.2 Raw data 1
2.3 Processed data 1
3.0 File Naming Conventions 2
3.1 Requirements 2
3.2 Suggested filename format 2
3.2.1 Instrument data files 2
3.2.2 Model output files 3
3.3 Data tiers and levels 3
3.4 Start date 4
3.5 Start time 4
3.6 Optional attributes 4
3.7 File type 5
4.0 Time Fields 5
5.0 Location Fields 5
5.1 Sensor height 6
6.0 Field Names 7
6.1 Field name descriptors 8
6.1.1 Prefix qualifiers 8
6.1.2 Measurement qualifiers 8
6.1.3 Subcategory qualifiers 9
6.1.4 Quantity qualifiers 9
6.1.5 Coordinate dimensions 10
6.1.6 Field name attributes 10
6.1.7 Standard field attribute names 11
6.1.8 Other possible attributes (not all inclusive) 11
7.0 NetCDF Global Attributes 11
7.1 Required and recommended global attributes 12
7.2 Other recommended global attributes 14
8.0 ASCII Header Requirements 15
9.0 Example Data Files 15
9.1 Example NetCDF data file 15
9.2 Example csv data file 19
10.0 Plots 23

11.0 References 23

1.0 Introduction

This living document proposes required and recommended standards adapted from the Atmospheric Radiation Measurement (ARM) Climate Research Facility Data Standards[1] and supersedes the previous version (PNNL-28552) titled “Atmosphere to Electrons (A2e) Data File Standards Version 1.1.” The standards described herein will enable development of automated analysis and discovery tools for Wind Data Hub (WDH) data and will facilitate development of future capabilities for delivering data on demand that can be tailored explicitly to user needs. Where feasible, the standards follow the Climate and Forecast (CF) convention. Using CF standards will increase the usability of the data by the broader scientific community. A full description of the CF convention can be found at cfconventions.org.

Benefits of adhering to these standards include:

- consistency across datastreams,
- code reuse by using consistent formats,
- simple and consistent software able to read all standardized files,
- files both human and machine readable as much as possible, and
- self-describing data files containing metadata describing how the data were collected.

2.0 Dataset Contents

2.1 Time span

To keep file sizes manageable and to keep content convenient for end users, the data within a single file must cover a time span no longer than an individual calendar year. If it is expected that end users would be interested in data covering time spans shorter than one year (e.g., specific seasons or months), then files should include data only from a single time span of either month, day, or hour as appropriate.

2.2 Raw data

Preserve all raw data in the event there is a need to reprocess due to changes in calibration or configuration of the instrument. Raw files must contain the instrument software version. For binary and ASCII files, a document containing the data format description must be attached to the metadata record.

2.3 Processed data

Processed data should follow the following guidelines:

1. If data are filtered or removed, ancillary variables or attributes should indicate why.
2. If data are averaged, ancillary variables or attributes should include the averaging interval and the averaging bin width.

[1] http://www.arm.gov/publications/programdocs/doe-sc-arm-14-010.pdf.

3. Quality check fields must indicate how they should be interpreted.
4. All measurements should include minimum value, maximum value and UDUNITS-compliant units.
5. Files must be greater than zero length.

3.0 File Naming Conventions

To facilitate the management of data stored in Atmosphere to Electrons (A2e), a standard file-naming format has been established for observational data and model output. It is highly recommended that the PI consult with the WDH before finalizing a filename.

3.1 Requirements

Filenames must adhere to the following requirements:

1. Filenames must be unique within a project.
2. Filenames must adhere to standards that can be parsed.
3. Filenames must be less than 255 characters.
4. Filenames must consist of alphanumeric characters with the addition of "-," "_," and ".".
5. Filenames should be all lower case.

3.2 Suggested filename format

3.2.1 Instrument data files

The suggested filename format for instrument data is:

<location>.<instrument_code>.<instance>.<data_level>.<start_date>.<start_time>.<optional_attributes>.<file_type>

- location – code that indicates a particular site within a project
- instrument_code – based on a list of accepted instrument types for a project
- instance – from a list of codes for each instrument_code that differentiates instruments (e.g., z01, z02, etc.)
- data_level – valid values of 00, 01, a0, a1, b1, etc. (see Section 3.3)
- start_date – YYYYMMDD (see Section 3.4)
- start_time – HHMMSS (see Section 3.5)
- optional_attributes – any number of attributes necessary to ensure uniqueness of files for a particular set (e.g., diag, field1) (see Section 3.6)
- file_type – a standard file type (e.g., .nc, .csv) (see Section 3.7)

Examples:

- sa5.sonic.z01.00.20150315.000000.wind_dir.csv
- sa5.sonic.z01.00.20150315.000000.wind_speed.csv
- sa5.sonic.z02.00.20150315.000000.nc

3.2.2 Model output files

The suggested filename format for model output is as follows and can depend on the project needs:

<scale>.<institution>.<codebase>.<location>.<domain>.<start_date>.<start_time>.<file_type>

- scale - mesoscale or microscale
- institution - institution that completed the model run
- codebase - used for the model run (e.g., WRF or HRRR)
- location - area of the model run (e.g., WFIP2 region or TTU Tower)
- domain - model domain number
- start_date - YYYYMMDD (see Section 3.4)
- start_time - HHMMSS (see Section 3.5)
- file_type - a standard file type extension (e.g., .nc, .csv) (See Section 3.6)

Examples:

mesoscale.nrel.wrf.wfip2.d02.20161122.005000.nc

mesoscale.pnnl.hrrr.wfip2.d01.20150923.121015.nc

3.3 Data tiers and levels

Data tiers are designated as either 0, a, b, c, d, or i.

- **0:** raw data
- **a:** converted or processed data
- **b:** quality-controlled data
- **c:** derived or calculated data
- **d:** model output
- **i:** plots

Data levels are based on the level of processing, with the lowest level of data being designated “00” data. Each subsequent data level has minimum requirements, and a data level is not increased until all the requirements of that level, as well as those of all data levels below it, have been met. A data level notation will consist of its data tier designation followed by a number.

- **00**: primary raw data streams collected directly from an instrument.

- **01** to **99**: redundant raw data stream, sneakernet data (transfer of data files by physically moving removable media), or external data that may consist of higher-order products but require further processing to conform to standards.
- **a0**: converted or processed raw data and metadata that have been reviewed and standardized.
- **a1**: a0-level data that have been additionally consolidated or reformatted.
- **b0**: intermediate quality-controlled data and metadata that have been reviewed and standardized.
- **b1**: b0-level data that have been additionally consolidated or reformatted.
- **c0**: derived or calculated data from one or more measured or modeled datasets and metadata that have been reviewed and standardized.
- **c1**: c0-level data that have been additionally consolidated or reformatted.
- **d1 to d9**: model output representing a certain domain.
- **i0:** used for plots generated from existing datasets.

3.4 Start date

The start date is the UTC (Coordinated Universal Time) date in YYYYMMDD format, consisting of exactly eight characters that indicate the start date of the first data point in the file. Single-digit month and day values are padded with a “0.” For example, February 4, 2012, is expressed as “20120204.”

3.5 Start time

The start time is the UTC time in HHMMSS format, consisting of exactly six characters that indicate the start time of the first data point in the file. Single-digit values are padded with a “0.” Sub-second times are truncated to the integer of the second’s value. For example, 5:00:19.57 UTC is expressed as “050019.” The time sample may not exceed 23:59:59. Numbers of hours greater than or equal to 24 or numbers of minutes or seconds greater than 60 will cause problems with time conversion programs.

3.6 Optional attributes

Optional attributes are necessary to indicate unique files within a project if two files are generated from the same instrument and location.

Examples:

- sa2.ecor.bao1_300nw.00.20150101.000000.wind_raw.csv
- sa2.ecor.bao1_300nw.00.20150101.000000.humid.csv

3.7 File type

The file type should be in a format that is documented and can be understood by available software tools. Examples include the following types:

- nc: NetCDF data format
- csv: Comma Separated values (CSV) file type
- hdf: Hierarchical Data Format data format (limited to satellite data)
- png: Portable Network Graphics (PNG) data format, recommended for drawings, sketches, and data plots
- jpg: Joint Photographic Expert Group (JPEG) data format, recommended for photographs

4.0 Time Fields

Time in NetCDF files is indicated in UTC and represented as “seconds since January 1, 1970 00:00:00,” which also is known as “epoch time.” For example, an epoch time of 1 is “Thursday January 1, 1970 00:00:01 UTC.” An epoch time of 992794875 is “Sunday June 17, 2001 16:21:15 UTC.” The time dimension must be set to UNLIMITED.

Time in a csv file is indicated with year, day of year, and time.

5.0 Location Fields

The instrument location is described using latitude, longitude, and altitude fields.

6. The requirements for latitude are units of degrees north and a field name of “lat”.
7. The requirements for longitude are units of degrees east and a field name of “lon”.
8. The recommended units of altitude are meters above mean sea level (MSL), and the required field name is “alt”.
9. The altitude measurement references the altitude of ground level to MSL.
10. The instrument height above ground level (AGL) is defined with the sensor_height attribute. Refer to Sensor Height (Section 5.1) for a complete explanation.
11. The use of standard_name attributes of “latitude”, “longitude”, and “altitude” is recommended.
12. The lat, lon, and alt fields can be dimensioned by time for mobile platforms when needed.

Example:

float lat;

lat:long_name = “North latitude”

lat:units = “degree_N”

lat:standard_name = “latitude”

lat:valid_min = -90.f

lat:valid_max = 90.f

float lon;

lon:long_name = “East longitude”

lon:units = “degree_E”

lon:standard_name = “longitude”

lon:valid_min = -180.f

lon:valid_max = 180.f

float alt;

alt:long_name = “Altitude above mean sea level”

alt:units = “m”

alt:standard_name = “altitude.”

5.1 Sensor height

If the declaration of the height of an instrument is desired, it is declared with an optional sensor_height attribute. If all sensors are at the same height for a datastream, a global attribute may be used. If different fields represent data at different heights, each field indicates the sensor height with the sensor_height attribute. The presence of a sensor_height field attribute supersedes the global attribute. To determine the height of the sensor above MSL, add a sensor_height value to the alt field value.

The sensor_height attribute format is written in the following order:

Numerical value, a UDUNITS-compliant unit, and “AGL,” all separated with a single space character. A negative value represents a measurement taken below ground level. The value is the height of the sensor AGL.

Example:

float wind_speed (time)

wind_speed:long_name = “Mean wind speed”

wind_speed:units = “m/s”

wind_speed;missing_value = -9999.f

wind_speed:sensor_height = “10.5 m AGL.”

6.0 Field Names

A field name should convey a basic understanding of the associated data. File space is not an issue, so cryptic field names that are typically only understood by the person who chose the name should not be used.

- The first character must be a letter. In accordance with NetCDF requirements, only letters, numbers, or underscores are allowed. Uppercase letters should be used sparingly.
- The field name is constructed by joining the names to the qualifiers using underscores (_).
- Field names should be concise. Be reasonable when selecting field names.
- Abbreviations are recommended only when needed for limiting excessively long variable names, for following previous conventions, or for clarity.
- To comply with archive database storage requirements, field name lengths must not exceed 255 characters.
- Use of single-character names is not recommended.
- The singular form of field names and dimensions are recommended (i.e., “temperature,” not “temperatures”).
- Greek letters are not allowed in NetCDF. In addition, it is strongly recommended that the spelled forms of Greek letters, formula symbols, or units be avoided.

Again, field names should be as concise as possible. For example, “temperature” fully spelled out is recommended unless the full field name becomes unreasonably long. The field name “atmospheric_temperature” is more descriptive of the measurement than temperature alone. A field labeled “temperature” could describe air temperature, instrument temperature, derived temperature, etc.

Name hierarchy is used for field differentiation within the same file. If a conflict arises, then the following hierarchy is used:

1. [super prefix]; e.g., qc, aqc, be, source
2. [prefix]; e.g., interpolated, calibrated, instantaneous
3. [measurement]; e.g., vapor_pressure, pressure, temperature
4. [subcategory]; e.g., head, air, upwelling, shortwave, hemisphere
5. [medium]; e.g., earth, satellite, sea, atmosphere
6. [height/depth]; e.g., 10m, 2cm, 5km
7. [enumeration]; e.g., e, w, n, s, a, b, 1, 2
8. [source name]; e.g., smos, met,
9. [algorithm]; e.g., fibonacci, wrf
10. [quantity]; e.g., mean, standard deviation, maximum, summation

Examples of field names using hierarchy are as follows:

- qc_atmospheric_temperature_10m
- soil_temperature_swats
- wind_speed_5m
- relative_humidity
- qc_vapor_pressure_aeri_std
- rain_rate_attenuation_csapr
- source_absorption_coefficient_405nm
- qc_log_backscatter_xpol_std

6.1 Field name descriptors

Some abbreviations are common and will be used often.

6.1.1 Prefix qualifiers

- inst = instantaneous
- fgp = fraction of good points
- be = best estimate
- qc = quality control
- aqc = ancillary QC or alternate QC
- inter = interpolated

6.1.2 Measurement qualifiers

- temp = temperature
- snr = signal to noise ratio
- lat = latitude
- lon = longitude
- alt = altitude
- navg = number of points averaged
- aod = aerosol optical depth
- aot = aerosol optical thickness (aod is preferred to aot)
- precip = precipitation
- rh = relative humidity
- wspd = wind speed
- wdir = wind direction

6.1.3 Subcategory qualifiers

- low = lower
- high = higher
- up = upwelling or coming from below
- down = downwelling or coming from above
- long = longwave
- short = shortwave
- pol = polarization
- hemisp = hemispheric
- ref = reference
- ir = infrared
- vis = visible
- uv = ultraviolet
- coef = coefficient
- scat = scattering
- aux = auxiliary
- rot = rotational
- copol = co-polarization
- xpol = cross-polarization
- depol = depolarization
- diff = delta or difference
- anc = ancillary

6.1.4 Quantity qualifiers

- std = standard deviation
- mean = arithmetic mean
- avg = arithmetic average (mean is preferable to average when the two are used interchangeably)
- mode = arithmetic mode
- med = arithmetic median
- var = variance
- sum = summation

- min = minimum
- max = maximum
- stderr = standard error
- log = logarithm
- ln = natural logarithm

6.1.5 Coordinate dimensions

If a coordinate dimension is used, a variable with the same name as the dimension is added with a long_name and a units attribute. Examples of coordinate dimensions are bin, height, range, and depth. The name of the dimension should clearly articulate the values. Using singular names is recommended but not abbreviations. The long_name attribute should be as concise as possible in describing what the values represent.

Example:

dimensions:

time = UNLIMITED; // (1440 currently)

range = 1999;

variables:

float range(range);

range:long_name = “Distance from transceiver to center of corresponding bin” ;

range:units = “km”;

A coordinate variable may not have a missing_value, _FillValue, or NaN value and must be monotonically increasing or decreasing. Data files containing dimensions that fail these regulations will be sent to Instrument Mentor or Translator for review.

6.1.6 Field name attributes

Long_name and units are required field name attributes.

- long_name: The long_name must be unique regarding the other fields in the same NetCDF file. Be as clear and concise as possible.
- units: UDUNITS-compliant units are highly recommended.

  The missing_value and standard_name attributes are also required in certain circumstances.
- missing_value: If the data field uses a specific value to represent no data, a missing_value attribute must be declared. There is no required value, but the recommended value is -9999. The value must be a scalar and of the same type as the corresponding data values. The

value must be outside the valid data range. Do not include a missing_value attribute with coordinate fields.

- standard_name: This is required for primary fields and should be set to a name found in the CF Standard Names table (https://cfconventions.org/Data/cf-standard-names/current/build/cf-standard-name-table.html) when available.

6.1.7 Standard field attribute names

- valid_min (the minimum valid value of the field, with all field data of lower value considered bad)
- valid_max (the maximum valid value of the field, with all field data of higher value considered bad)
- valid_delta
- qc_min
- qc_max
- resolution
- comment
- comment_<#> (used for multiple distinct comments within a single field)
- precision
- accuracy
- uncertainty

6.1.8 Other possible attributes (not all inclusive)

- valid_range
- actual_wavelength
- correction
- filter_wavelength
- sensor_height

7.0 NetCDF Global Attributes

All global attributes must have a value. If a value is unknown at the time the file is created, the attribute must clearly indicate that no known value exists. A standard value of “unknown” or -9999 set to the proper data type is recommended (127 for type byte). Recommended attributes may be omitted if the value is expected to be unknown. If required attributes must be written but a value is not expected to exist, use “N/A.”

7.1 Required and recommended global attributes

The order of global attributes is not a requirement, but the order listed in this document is recommended. These global attribute recommendations are applicable to NetCDF files. See Section 9.1 for an example NetCDF file header.

1. command_line

This attribute records the command line used to run the process. If the command is run multiple times to generate the individual file, list the command used to generate the initial file. If a single command line is not used to generate the file, list the parameters that need to be set to create the file.

Example: command_line = “langley -d 20130116 -p mfrsr -f sgp.E13”

2. process_version (for processed files)

This attribute records the version of the process used to produce the data.

Example: process_version = “ingest-met-4.10-0.el5”

3. input_datastreams (for processed data only, required with conditions)

This attribute records the itemized list of input datastreams available at runtime, process versions, and filename date ranges. It may be omitted if the source attribute or source fields are used to describe input datastreams. The datastream, version, and date range are separated by a space-colon-space (“ : ”). The individual datastream entries are separated by a space-semicolon-new line-space (“ ;\n ”). If multiple files exist for a single date but not all files are used, the individual ranges used should be itemized as separate entries. The separator between dates in a given date-time range is a hyphen (“yyyymmdd.hhmmss-yyyymmdd.hhmmss”). If the time period spans a single date, no hyphen or end date should be included, and the date range is a single date-time (“yyyymmdd.hhmmss”).

Example: input_datastreams = “sgpsondewnpnC1.a1 : 6.1 : 20010208.232700-20010210.053400 ;\n sgpmwrlosC1.b1 : 1.17 : 20010209.000000 ;\n sgp1twrmrC1.c1: Release_1_4 : 20010209.000000 ;\n sgparscl1clothC1.c1 : Release_2_9 : 20010209.000000”

4. field_campaign_short_name

This attribute contains the field campaign acronym.

Example: field_campaign_short_name = “XPIA”

5. field_campaign_long_name

This attribute contains the full name of the field campaign.

Example: field_campaign_long_name = “Experimental Planetary Boundary Layer Instrumentation Assessment”

6. instrument_id

This attribute records the identifier of the instrument.

7. instrument_name

This attribute records the full name of the instrument.

8. instance_id

This attribute records the instance of the instrument if multiple sensors are in the same location.

9. data_level

This attribute records the data level.

Example: data_level = “a1”

10. datastream

This attribute records the identifier of the datastream.

Example: datastream = “sgpmfrsrE32.b1”

11. serial_number (ingest only, required with stipulation)

This attribute records the serial number of the instrument(s) used to collect data. This is only required if the serial number is expected to be known at runtime and is capable of changing. If multiple instruments exist, specify the instrument. Otherwise, use only a serial number. Individual serial number entries are separated by a space-semicolon-new line space (“ ;\n ”). Instrument descriptors are separated from the serial number with a colon-space (“: ”).

Example 1: serial_number = “54321DT.”

Example 2: serial_number = “PIR1-DIR: 31312F3 ;\n PIR2-DIR: 30167F3 ;\n Diffuse PSP: 33271F3 ;\n NIP: 31876E6 ;\n PSP-DS: 33703F3 ;\n SKY-IR: 1845.”

12. sampling_interval

This attribute records the expected sampling interval. If the instrument sampling interval is different, it should be noted in the instrument documentation. The format is the interval time and a UDUNITS-compliant descriptor separated by a single space character.

Example: sampling_interval = “5 minutes”

13. averaging_interval (if the data is averaged)

This attribute records the time interval of the averaging period recorded in seconds.

Example: averaging_interval = “60 seconds”

14. averaging_bin_width

This attribute records the width of the bin over which averaging was performed.

15. sensor_height

This attribute records the height of all sensors AGL. If multiple sensors exist at different heights, use a field-level attribute. See Sensor Height (Section 3.1) for format details. If sensor_height is defined at the field level for all relevant fields, a global attribute should not be defined.

16. history

This attribute records the username, machine name, and date in UDUNITS or ISO 8601 format. If the file is modified, the original value is retained and new information is appended to the attribute value with statements separated by a space-semicolon-new line-space (“ ;\n ”).

Example: history = “created by user dsmgr on machine ruby at 1-Jan-2007,2:43:02”

For model results, the following global attributes are highly recommended:

17. model_run_hours

This attribute records the number of hours run by the model.

18. model_grid_size

This attribute records the model grid size in kilometers.

19. num_vertical_layers

This attribute records the number of vertical layers represented in the model.

7.2 Other recommended global attributes

1. title

This attribute contains a succinct English language description of what is in the dataset. The value would be similar to a publication title.

Example: title = “Planetary boundary layer height”

2. institution

This attribute specifies which institution was responsible for producing the original data.

Example: institution = “United States Department of Energy - Atmospheric Radiation Measurement (ARM) program”

3. description

This attribute records a longer English language description of the data.

Example: description = “XPIA hourly averaged QC controlled product, derived from scanning lidar”

4. references

This attribute records the published or web-based references that describe the dataset or the methods used to produce it.

Example: references = “http://www.arm.gov/data/vaps/armbe/armbecldrad”

8.0 ASCII Header Requirements

All ASCII files should be in csv (comma-separated values) format and adhere to the following requirements:

1. The global and variable attributes must be listed in the header section.
2. The first row of the header must contain the number of rows in the header.
3. The last row of the header must contain a comma-separated list of the variable names.
4. Strings of characters containing commas must be enclosed in quotation marks, e.g., "location_meaning=Block Island, RI" and "qc_wind_speed_stbd:flag_masks=[1, 2, 4]"
5. Variable attributes must be entered on separate rows and formatted as <variable_name>:<attribute_name>=<attribute_value>

See Section 9.2 for an example csv file.

9.0 Example Data Files

The following are examples of NetCDF and csv data files:

9.1 Example NetCDF data file

data file name = sgptempprofile10sC1.c1.20130101.010203.nc

Dimensions:

time = UNLIMITED ; // (14400 currently)

bound = 2

height = 100

Variables:

int base_time

base_time:string = “01-Jan-2013,00:00:00 GMT”

base_time:long_name = “Base time in Epoch”

base_time:units = “seconds since 1970-1-1 0:00:00 0:00”

base_time:ancillary_variables = “time_offset”

double time_offset (time)

time_offset:long_name = “Time offset from base_time”

time_offset:units = “seconds since 2013-01-01 00:00:00 0:00”

time_offset:ancillary_variables = “base_time”

time_offset:bounds = “time_bounds”

double time (time)

time:long_name = “Time offset from midnight”

time:units = “seconds since 2013-01-01 00:00:00 0:00”

time:standard_name = “time”

time:bounds = “time_bounds”

double time_bounds (time, bound)

time_bounds:long_name = “Time cell bounds”

time_bounds:units = “seconds”

float height (height)

height:long_name = “Center of height bin”

height:units = “m”

height:standard_name = “height”

height:bounds = “height_bounds”

float height_bounds(height, bounds)

height_bounds:long_name = “Height bin bounds”

height_bounds:units = “m”

float atmospheric_temperature(time, height)

atmospheric_temperature:long_name = “Atmospheric temperature”

atmospheric_temperature:units = “degC”

atmospheric_temperature:missing_value = -9999.f

atmospheric_temperature:standard_name = “air_temperature”

atmospheric_temperature:cell_methods = “time:mean height:mean”

atmospheric_temperature:ancillary_variables = “qc_atmospheric_temperature

source_atmospheric_temperature instrument_status”

int qc_atmospheric_temperature(time, height)

qc_atmospheric_temperature:long_name = “Quality check results on field: Atmospheric temperature”

qc_atmospheric_temperature:units = “1”

qc_atmospheric_temperature:flag_method = “bit”

qc_atmospheric_temperature:comment = “A QC bit set anywhere along the profile will result in the bit being set.”

qc_atmospheric_temperature:bit_1_description = “Value is equal to missing_value”

qc_atmospheric_temperature:bit_1_assessment = “Bad”

qc_atmospheric_temperature:bit_2_description = “The instrument detected a hardware failure”

qc_atmospheric_temperature:bit_2_assessment = “Bad”

qc_atmospheric_temperature:bit_3_description = “Values greater than two standard deviations of historical distribution”

qc_atmospheric_temperature:bit_3_assessment = “Indeterminate”

int source_atmospheric_temperature (time)

source_atmospheric_temperature:long_name = “Source for field: Atmospheric temperature”

source_atmospheric_temperature:units = “1”

source_atmospheric_temperature:description = “This field contains bit-packed integer values, where each bit represents a source of the data. Non-zero bits indicate the source used in the description for those bits. A value of 0 (no bits set) indicates no source.”

source_atmospheric_temperature:flag_method = “bit”

source_atmospheric_temperature:bit_1_description = “sgpsondewnpnC1.b1:tdry”

source_atmospheric_temperature:bit_2_description = “sgpaeriprofC1.c1:temperature”

source_atmospheric_temperature:bit_3_description = “sgp1290rwpC1.c1:temp”

source_atmospheric_temperature:bit_4_description = “conwarfX1.a1:atmos_temp”

int instrument_status(time)

instrument_status:long_name = “Instrument status”

instrument_status:units = “1”

instrument_status:missing_value = -9999

instrument_status:flag_masks = 1, 2, 4, 8

instrument_status:flag_meanings = “power_failure hardware_fault software_fault maintenance_mode”

float lat

lat:long_name = “North latitude”

lat:units = “degree_N”

lat:standard_name = “latitude”

lat:missing_value = -9999

lat:valid_min = -90.f

lat:valid_max = 90.f

float lon

lon:long_name = “East longitude”

lon:units = “degree_E”

lon:standard_name = “longitude”

lon:missing_value = -9999.f

lon:valid_min = -180.f

lon:valid_max = 180.f

float alt

alt:long_name = “Altitude above mean sea level”

alt:units = “m”

alt:standard_name = “altitude”

alt:missing_value = -9999.f

// global attributes:

:command_line = “tempprofile -d 20130101 -f sgp.C1”

:process_version = “ingest-met-4.10-0.el5”

:dod_version = “tempprofile-b1-2.0”

:input_datastreams = “sgpsondewnpnC1.b1 : 6.1 : 20130101 ;\n sgpaeriprofC1.c1 : 1.1 : 20130101.000000 ;\n sgp1290rwpC1.c1: Release_1_4 : 20130101.000000 ;\n conwarfX1.a1 : Release_2_9 : 20130101.000000”

:site_id = “sgp”

:platform_id = “tempprofile”

:facility_id = “C1”

:data_level = “c1”

:location_description = “Southern Great Plains (SGP), Lamont, OK (C1)

:datastream = “sgptempprofileC1.c1”

:title = “Lidar best estimate of atmospheric temperature profile”

:institution = “United States Department of Energy – Pacific Northwest National Laboratory”

:description = “Best estimate of atmospheric temperature profile over Boulder, CO”

:references = “http://www.anything.gov”

:history = “created by user jelliot on machine ruby at 1-Jan-2007,2:43:02.”

9.2 Example csv data file

header=74

title=Metocean Buoy Pipeline

description=Ingest of metocean data from an AXYS Technologies buoy

location_id=oahu

dataset_name=buoy_z07

qualifier=metocean

temporal=10m

data_level=a1

datastream=oahu.buoy_z07-metocean-10m.a1

history=created by user dsmgr on machine ruby at 2023-11-16T15:35:25.798004

code_version=38ce32ca17bd2d8f4d5388208946ff7a66829621

"location_meaning=Oahu, HI"

instrument_manufacturer=AXYS Technologies Inc.

instrument_name=WindSentinel

wind_speed_stbd:units=m/s

wind_speed_stbd:long_name=Horizontal Wind Speed

"wind_speed_stbd:comment=Surface wind speed, anemometer located on starboard buoy mast"

wind_speed_stbd:valid_max=50.0

wind_speed_stbd:ancillary_variables=qc_wind_speed_stbd

wind_direction_stbd:units=degrees

wind_direction_stbd:long_name=Horizontal Wind Direction

"wind_direction_stbd:comment=Surface wind direction, wind vane located on starboard buoy mast"

wind_direction_stbd:valid_max=360.0

wind_direction_stbd:ancillary_variables=wind_direction_stbd

solar_radiation:units=W/m^2

solar_radiation:long_name=Solar Radiation

solar_radiation:ancillary_variables=qc_solar_radiation

relative_humidity:units=%

relative_humidity:long_name=Relative Humidity

relative_humidity:valid_max=100.0

relative_humidity:ancillary_variables=qc_relative_humidity

latitude:units=degree_N

latitude:long_name=Latitude

latitude:ancillary_variables=qc_latitude

longitude:units=degree_E

longitude:long_name=Longitude

longitude:ancillary_variables=qc_longitude

qc_wind_speed_stbd:units=1

qc_wind_speed_stbd:long_name=Quality check results on field: Horizontal Wind Speed

"qc_wind_speed_stbd:flag_masks=[1, 2, 4]"

"qc_wind_speed_stbd:flag_meanings=[Value is equal to _FillValue or NaN, Value is less than the valid_min., Value is greater than the valid_max.]"

"qc_wind_speed_stbd:flag_assessments=[Bad, Bad, Bad]"

qc_wind_speed_stbd:standard_name=quality_flag

qc_wind_direction_stbd:units=1

qc_wind_direction_stbd:long_name=Quality check results on field: Horizontal Wind Direction

"qc_wind_direction_stbd:flag_masks=[1, 2, 4]"

"qc_wind_direction_stbd:flag_meanings=[Value is equal to _FillValue or NaN, Value is less than the valid_min., Value is greater than the valid_max.]"

"qc_wind_direction_stbd:flag_assessments=[Bad, Bad, Bad]"

qc_wind_direction_stbd:standard_name=quality_flag

qc_solar_radiation:units=1

qc_solar_radiation:long_name=Quality check results on field: Solar Radiation

"qc_solar_radiation:flag_masks=[1, 2, 4]"

"qc_solar_radiation:flag_meanings=[Value is equal to _FillValue or NaN, Value is less than the valid_min., Value is greater than the valid_max.]"

"qc_solar_radiation:flag_assessments=[Bad, Bad, Bad]"

qc_solar_radiation:standard_name=quality_flag

qc_relative_humidity:units=1

qc_relative_humidity:long_name=Quality check results on field: Relative Humidity

"qc_relative_humidity:flag_masks=[1, 2, 4]"

"qc_relative_humidity:flag_meanings=[Value is equal to _FillValue or NaN, Value is less than the valid_min., Value is greater than the valid_max.]"

"qc_relative_humidity:flag_assessments=[Bad, Bad, Bad]"

qc_relative_humidity:standard_name=quality_flag

qc_latitude:units=1

qc_latitude:long_name=Quality check results on field: Latitude

"qc_latitude:flag_masks=[1, 2, 4]"

"qc_latitude:flag_meanings=[Value is equal to _FillValue or NaN, Value is less than the valid_min., Value is greater than the valid_max.]"

"qc_latitude:flag_assessments=[Bad, Bad, Bad]"

qc_latitude:standard_name=quality_flag

qc_longitude:units=1

qc_longitude:long_name=Quality check results on field: Longitude

"qc_longitude:flag_masks=[1, 2, 4]"

"qc_longitude:flag_meanings=[Value is equal to _FillValue or NaN, Value is less than the valid_min., Value is greater than the valid_max.]"

"qc_longitude:flag_assessments=[Bad, Bad, Bad]"

qc_longitude:standard_name=quality_flag

time,wind_speed_stbd,wind_direction_stbd,solar_radiation,relative_humidity,latitude,longitude,qc_wind_speed_stbd,qc_wind_direction_stbd,qc_solar_radiation,qc_relative_humidity,qc_latitude,qc_longitude

11/17/22 00:10,2.049,162,812.058,52.616,21.293467,-157.859483,0,0,0,0,0,0

11/17/22 00:20,2.75,146,941.182,53.345,21.29355,-157.859467,0,0,0,0,0,0

11/17/22 00:30,3.41,160,831.263,52.05,21.29355,-157.859483,0,0,0,0,0,0

11/17/22 00:40,4.15,107,907.458,51.819,21.2935,-157.859483,0,0,0,0,0,0

11/17/22 00:50,1.679,105,407.876,53.898,21.293517,-157.859483,0,0,0,0,0,0

11/17/22 01:00,3.15,156,575.918,54.492,21.293483,-157.859517,0,0,0,0,0,0

11/17/22 01:10,5.57,103,791.369,53.184,21.293517,-157.85955,0,0,0,0,0,0

11/17/22 01:20,5.23,131,744.359,51.846,21.293533,-157.859533,0,0,0,0,0,0

10.0 Plots

Plots should adhere to the following guidelines whenever possible:

1. Label each plot with a short, self-explanatory title.
2. Indicate the name of the dataset containing the data in the plot.
3. Include the date and time of creation.
4. Format text in a sans-serif font between eight and fourteen points.
5. Plot the dependent variable on the vertical axis and the independent variable on the horizontal axis.
6. Label both axes with descriptive variable names and units.
7. Divide the axes into 4-10 units with a range that makes full use of the plot area.
8. Label divisions with a meaningful numbering system.
9. Use zero as the origin of both axes. (If the scaling of any axis prohibits the inclusion of the origin, a broken axis should indicate the offset.)
10. Use a single linear scale whenever possible, or a common scale if a single scale is not possible.
11. Add gridlines on major divisions, ideally white gridlines on a gray background.
12. Show different sets of data by using different plotting symbols that are defined in a legend.
13. Avoid conveying information only through color or use color palettes that are colorblind accessible.

11.0 References

DOE/SC-ARM-14-010. 2014. *ARM Data File Standards Version 1.0*. Prepared by ARM Standards Committee for the Atmospheric Radiation Measurement Climate Research Facility and U.S. Department of Energy, Richland, Washington.

PNNL-28552. 2019. *Atmosphere to Electrons (A2e) Data File Standards Version 1.1*. Prepared by A2e Standards Committee for the U.S. Department of Energy, Richland, Washington.

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov