PNNL-28552

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by Battelle Since 1965

Atmosphere to Electrons (A2e) Data File Standards Version 1.1

March 2019

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

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;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <http://www.ntis.gov/about/form.aspx>
Online ordering: http://www.ntis.gov


This document was printed on recycled paper.
(8/2010)

PNNL-28552

Atmosphere to Electrons (A2e) Data File Standards Version 1.1

March 2019

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

Pacific Northwest National Laboratory
Richland, Washington 99352

Summary

The standards described in this document will enable development of automated analysis and discovery tools for Atmosphere to Electrons data and will facilitate development of future capabilities for delivering data on demand that can be tailored explicitly to user needs. This living document proposes standards that include required and recommended standards adapted from the Atmospheric Radiation Measurement Climate Research Facility Data Standards.[1]

Summary of guidelines

Raw data

Preserve all raw data in case there is a need to reprocess due to changes in calibration or configuration of the instrument.

Raw files must contain instrument software version

For Binary and ASCII files, a document containing the data format description should be included.

Processed data

If data is filtered or removed, it is best to include ancillary variables to indicate why data was filtered.

If data is averaged or interpolated, ancillary variables should include averaging interval and averaging bin width.

Quality check fields should indicate how they should be interpreted.

All measurements should include minimum value, maximum value and units (udunit compliant)

Files must be greater than zero length.

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

Acronyms and Abbreviations

A2e	Atmosphere to Electrons Initiative
AGL	above ground level
ARM	Atmospheric Radiation Measurement Climate Research Facility
CF	calendar field
MSL	mean sea level
NetCDF	Network Common Data Form
QC	quality control
UTC	Coordinated Universal Time
VAP	value-added product

Contents

1.0 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 DAP before finalizing a filename.

1.1 Requirements

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

The filename standard for observational data is:

<instrument_code>.<instance>.<data_level>.<startdate>.<starttime>.<optional-attributes>.<file-type>

- 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 (IE z01, z02,etc)
- data_level – valid values of 00, 01,a0,a1,b1,c1 (refer to Section 1.2)
- startdate – YYYYMMDD (see Section 1.3)
- starttime – HHMMSS (see Section 1.4)
- optional – any number of attributes necessary to ensure uniqueness of files for a particular set (i.e., diag, field1, etc.)
- file-type – a standard file extension that is documented to be used and understood by an available software tool.

Examples:

- ecor.z01.00.20150315.000000.wind_dir.csv
- ecor.z01.00.20150315.000000.wind_speed.csv
- ecor.z02.00.20150315.000000.nc

The filename standard for MMC model output is as follows and can depend on the project needs.

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

- scale - mesoscale or microscale

- institution - institution that completed the 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
- startdate - YYYYMMDD (see Section 1.3)
- starttime - HHMMSS (see Section 1.4
- filetype - a standard file extension that is documented to be used and understood by an available software tool. (e.g., .nc)

Examples:

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

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

1.2 Data Levels

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 of the requirements of that level, as well as those of all data levels below it, have been met. A data level will consist of one lowercase letter followed by one number (except for raw data).

- **00**: raw data – primary raw data stream collected directly from instrument.
- **01** to **99**: raw data – redundant 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**: raw data converted to higher-level data products. Not intended for distribution to data users.
- **a1**: calibration factors applied and converted to geophysical units.
- **b0**: intermediate quality-controlled datastream – this data level always is used as input to higher-level data products. Not intended for distribution to data users.
- **b1**: quality control (QC) checks applied to, at least, one measurement and stored in an accompanying QC field meeting QC standards listed in this document.
- **b2** to **b9**: further processing on b1 level data that does not merit c-level classification. For example, additional QC test or different parameters used in processing. A description of the process must be described in the NetCDF (Network Common Data Form) header, instrument handbook, or technical paper available to data users.
- **c0**: intermediate VAP; data level always used as input to a higher-level value-added product (VAP). Not intended for distribution to data users.

- **c1**: derived or calculated VAP using one or more measured or modeled data as input. For external data, .c1-level data may contain gridded model data, satellite data, or other data that have had algorithms applied by an external source.
- **d01 to d09**: model output representing a certain domain.
- **d02**: model output representing a certain domain.

1.3 Startdate

(YYYYMMDD) is the UTC (Coordinated Universal Time) date in year, month, day-of-month format, consisting of exactly eight characters and indicating 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."

1.4 Starttime

(HHMMSS) is the UTC time in hour, minute, second 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.

1.5 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:

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

1.6 File_type

- nc: NetCDF data format.
- asc: ASCII data format.
- 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.
- mpg: Moving Picture Expert Group (MPEG) format. Recommended for movie format.

- pdf: Portable Document Format (PDF) file type. Recommended for formatted and graphics-rich documents.

2.0 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 means “Thursday January 1, 1970 00:00:01 UTC.” An epoch time of 992794875 is “Sunday June 17, 2001 16:21:15 UTC.”

Time in an ASCII file is indicated with year, day of year, and time of year.

.

3.0 Location Fields

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

1. The required unit of latitude is degrees north, and the field name is lat.
2. The required unit of longitude is degrees east, and the field name is lon.
3. The recommended unit of altitude is meters above mean sea level (MSL), and the required field name is alt.
4. The altitude measurement references the altitude of ground level to MSL.
5. The instrument height above ground level (AGL) is defined with the sensor_height attribute. Refer to Sensor Height (Section 3.1) for a complete explanation.
6. Use of standard_name attributes is recommended.
7. 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.”

3.1 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 sensor_height value to the alt field value.

The sensor_height attribute format is written in the following order:

Numerical value, calendar field (CF) udunit compliant unit, “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.”

4.0 Field Names

A field name should convey basic understanding of the associated data. File space is not an issue, so cryptic field names that typically are 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 should be used except in cases where their use is needed to avoid excessively long field names, to follow previous conventions, or to provide 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 NetCDF4. Also, 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]; for example, qc, aqc, be, source
2. [prefix]; for example, interpolated, calibrated, instantaneous
3. [measurement]; for example, vapor_pressure, pressure, temperature
4. [subcategory]; for example, head, air, upwelling, shortwave, hemisphere
5. [medium]; for example, earth, satellite, sea, atmosphere
6. [height/depth]; for example, 10m, 2cm, 5km
7. [enumeration]; for example, e, w, n, s, a, b, 1, 2
8. [source name]; for example, smos, met,
9. [algorithm]; for example, fibonacci, wrf
10. [quantity]; for example, mean, standard deviation, maximum, summation.

Example of field names using hierarchy are listed 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.

4.1 Field Name Descriptors

Some abbreviations are common and will be used often.

4.1.1 Prefix Qualifier

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

4.1.2 Measurement Qualifier

- 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.

4.1.3 Subcategory Qualifier

- 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.

4.1.4 Quantity Qualifier

- 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.

4.1.5 Coordinate Dimensions

If a coordinate dimension is used, a variable with the same name as the dimension is added with a long_name and units attribute. Examples of coordinate dimensions are: bin, height, range, or 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.

4.1.6 Field Name Attributes

Long_name and units are required field name attributes.

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

- 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. Do not include with coordinate fields. The value must be a scalar and the same type as the corresponding data values. The value must be outside the valid data range.
- standard_name: Required if a primary field and the standard name exist in the CF table.
- min_value: The minimum value for the variable below that considered bad.
- max_value: The maximum value for the variable above that considered bad.

4.1.7 Standard Field Attribute Names

- valid_min
- valid_max
- valid_delta
- qc_min
- qc_max
- resolution
- comment
- comment_<#> (used for multiple distinct comments within a single field)
- precision
- accuracy
- uncertainty.

4.1.8 Other Possible Attributes (Not All Inclusive)

- valid_range
- actual_wavelength
- corrections
- filter_wavelength
- sensor_height.

5.0 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.”

5.1 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 for NetCDF files.

1. **command_line**

Definition: Records 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 file)**

Definition: 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)**

Definition: Records the itemized list of input datastreams available at runtime, process versions, and filename date ranges. 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**

Definition: Field campaign acronym.

5. **field_campaign_long_name**

Example: site = “Experimental Planetary Boundary Layer Instrumentation Assessment.”

6. **instrument_id**
7. **instrument_name**
8. **instance_id**

Definition: The instance of the instrument if multiple sensors are located in the same location.

data_level

Definition: Records data level.

Example: data_level = ".00,a1,.b1.c1."

9. **datastream**[SC2]

Definition: Datastream identifier.

Example: datastream = "sgpmfrsrE32.b1."

10. **serial_number (ingest only, required with stipulation)**

Definition: Records serial number of instrument(s) used to collect data. 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 (": "). Type is recommended to be character.

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."

11. **sampling_interval**

Definition: Records expected sampling interval. If the instrument sampling interval is different, it should be noted in the instrument documentation. Format is the interval time and compliant *udunit* descriptor separated by a single space character.

12. **averaging_interval (if the data is averaged)**

Definition: Records the time interval of the averaging period recorded in seconds.

Example: 60 seconds.

13. **averaging_bin_width**

Definition: Records if the averaging was done to the beginning, middle, or end.

14. **sensor_height**

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

15. **history**

Definition: Records the user name, machine name, and the date in CF *udunit* 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, these global attributes are highly recommended:

16. **model_run_hours**

The number of hours run by the model.

17. **model_grid_size**

The number in kms of the model grid size.

18. **num_vertical_layers**

The number of vertical layers represented in the model.

5.2 Other Recommended Global Attributes

1. **title**

Definition: A succinct English language description of what is in the data set. The value would be similar to a publication title.

Example: “Planeteary Boundary layer height.”

2. **institution**

Definition: Specifies where the original data were produced. If provided, the value exactly matches the value listed here. Exceptions will be allowed on a case-by-case basis.

3. **description**

Definition: Longer English language description of the data.

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

4. **references**

Definition: Published or web-based references that describe the data or methods used to produce it.

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

6.0 Example ASCII and NetCDF Data Files

The following depicts examples of ASCII and NetCDF data files:

6.1 Example ASCII Data File

- number of header lines to skip
- number of items to expect per data line
- type of delimiter

index,time_stamp,doy,toy,d_n,lat,lon,alt,h_flux,h_radiation

6769,20141420000,142, 142.00000,0,71.2803,156.6095,7.0000,3.9500,3.2000

6770,20141420030,142, 142.02084,0,71.2803,156.6095,7.0000,3.9500,3.2000

6771,20141420100,142, 142.04167,0,71.2803,156.6095,7.0000,3.9500,3.2000

6772,20141420130,142, 142.06250,0,71.2803,156.6095,7.0000,3.9500,3.2000

6773,20141420200,142, 142.08334,0,71.2803,156.6095,7.0000,3.9500,3.2000

6.2 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 = "unitless"

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 = "unitless"

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 = “unitless”

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”

:Conventions =“ARM_Convention-1.0 CF-1.6”

: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.”

7.0 Reference

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.

.

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by **Battelle** *Since 1965*

U.S. DEPARTMENT OF
ENERGY

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)

www.pnnl.gov