

photolysis-frequencies

Document Version: 1.0

Vocabulary Version: 2.0

Last Revised Date: September 2020

Revision Due Date: September 2021

Repository Link: Not yet available

Revision History.

Date	Version	Comments	Revised by	Contact
10/09/2020	v1.0	Initial document generation	B.Brooks	barbara.brooks@ncas.ac.uk

Table of contents:

1. Introduction
2. Data Processing Levels
3. Data Types
4. Nomenclature
5. Data Quality Flags
6. File Naming
7. Common
 - 7.1. Metadata (Global Attributes)
 - 7.2. Dimensions
 - 7.3. Variables
8. Data Product Specific
 - 8.1. File Name Variants
 - 8.2. Metadata (Global Attributes)
 - 8.3. Dimensions
 - 8.4. Variables
- 9.

1. Introduction

The data product developed for NCAS that this document supports are designed following the FAIR (Findable Accessible Interoperable Reusable) principles of data management: “FAIR Guiding Principles for scientific data management and stewardship.” Wilkinson et al., 2016: <https://doi.org/10.1038/sdata.2016.18>.

There are now substantial drivers for data providers to implement the FAIR principles. These include:

1. The G20 group of nations: At the 2016 Hangzhou summit, the G20 leaders issued a statement endorsing the application of FAIR principles to research.
2. Increased emphasis on data and its reusability at policy level in national government and science funding bodies.
3. Science publishers have a growing requirement for data DOI's and data traceability.
4. The science community is expected to demonstrate data dissemination, usage, and deliver impact statements.

The FAIR principles of data management and stewardship aim to make data:

Findable:

- The first step in use and reuse data is to find them.
- Metadata and data should be easy to find for both humans and computers.
- Machine-readable metadata are essential for automatic discovery of datasets and services.
- This means:
 - Metadata are assigned a globally unique and eternally persistent identifier.
 - Data are described with rich metadata.
 - Metadata are registered or indexed in a searchable resource.
 - Metadata specify the data identifier.

Accessible:

- Once the user finds the required data, they need to know how they can be accessed, possibly including authentication and authorisation.
- This means:
 - Metadata are retrievable by their identifier using a standardised communications protocol.
 - The protocol is open, free, and universally implementable.
 - The protocol allows for an authentication and authorization procedure, where necessary.
 - Metadata are accessible, even when the data are no longer available

Interoperable:

- Data usually needs to be integrated with other data.
- Data needs to interoperate with applications or workflows for analysis, storage, and processing.
- This means:
 - Metadata uses a formal, accessible, shared, and broadly applicable language for knowledge representation.
 - Metadata use vocabularies that follow FAIR principles.
 - Metadata include qualified references to other metadata.

Reusable:

- The ultimate goal of FAIR is to optimise the use and reuse of data.
- To achieve this, metadata and data should be well-described so that they can be replicated and/or combined in different settings.

- This means:
 - Metadata have a plurality of accurate and relevant attributes.
 - Metadata are released with a clear and accessible data usage license.
 - Metadata are associated with their provenance.
 - Metadata meet domain-relevant community standards.

NCAS (<https://www.ncas.ac.uk/>) is putting its own house in order and applying the FAIR principles to the data it generates. Working with the data scientists at CEDA (<https://www.ceda.ac.uk/>), the UK archive for atmospheric and earth observation data, the NCAS-Observation team have:

- Defined data products for NCAS instrumentation.
- Uniquely named all the NCAS instrumentation.
- Developed an integrated repository structure for software and supporting documentation.
- Introduced a controlled vocabulary for instruments and data products.
- Introduced file standards that utilise the controlled vocabulary and are NetCDF4 - classic compliant.

The team is also working to produce various software tools which enable easy access to the data in these files. These tools are open-source and can be accessed via the NCAS-Observations website (<https://sites.google.com/ncas.ac.uk/ncasobservations/home>). The site also provides access to supporting information.

This document details what a user should expect to find in one of our data files. The first section of this document details the common file components and how they should be used: this includes how file names are constructed and the use of quality control flags. The second section details a common set of file level metadata (global and variable attributes and array dimensions) that appear in all files. i.e. irrespective of the data product. The final section details the **data product specific** metadata.

Note that:

1. Files will never include metadata (dimensions, global or variable attributes) that are not detailed in the data product supporting document.
2. If data for a **common** variable is not available then, these variables will be included but will contain only their designated `_FillValue`.
3. If data for a **data product specific** variable is not available then, rather than pad the file with variables containing only the `_FillValue`, that variable will be omitted.

All the data files have the same file-level metadata irrespective of the instrument source. The file-level metadata components in each file are:

- Global attributes: including the metadata standards followed in the file, change log (history) and other useful, general information (e.g. licence, authors etc):
- Dimensions: information about the data array sizes. All data have a dimension of time
- Variables: attributes of each data variable included in the file, e.g. name(s), units

Comment and feedback on the data file content, structure and supporting document is always welcome and appreciated. Please contact: barbara.brooks@ncas.ac.uk

2. Data Processing Levels

The NCAS-AMF standard defines the following data product levels and are used to indicate what post-processing has been performed. That is what additional processing has been

performed over-and-above that done automatically by the instrument at the time of the original measurement. There are 4 processing levels defined as follows:

- **Data Product Level 0**
 - Data is in the format native to the instrument - no processing has been applied.
- **Data Product Level 1**
 - Post processing to parse data from native format to archive format - basic QC applied.
- **Data Product Level 2**
 - Post processing to parse data from native format to archive format - basic QC applied and additional processing for motion correction, flow distortion, statistical appraisal, calibration.
- **Data Product Level 3**
 - A data product that is derived from a data product of level 1 or 2 or their combination.

The data to be found in these files will **always** be level 1 or above. Level 0 data is kept in deep storage and is only accessible through contact with the file author,

3. Data types

All data conform to defined data types. Depending on the software used to interrogate the files data types may be given a different name to that used here. To be precise:

Python3 name	Definition	Range
byte	8-bit unsigned integer	0 to 255
int32	32-bit signed integer	-2,147,483,648 to +2,147,483,647
int64	64-bit signed integer	-9,223,372,036,854,775,808 to +9,223,372,036,854,775,807
float32	32-bit Single-precision floating-point	-3.4E+38 to +3.4E+38
float64	64-bit Double-precision floating-point	-1.7E+308 to +1.7E+308

Note the data type is not given as an explicit variable attribute.

4. Nomenclature

The language and field name structuring used throughout this document and the files themselves, follows the CF conventions. In particular attributes and variable names are:

- All lowercase.
- Where the attribute or variable name comprises multiple parts these parts are separated by an underscore “_”.

Full details of the CF conventions used can be found at (<http://cfconventions.org/>).

All variables are defined in terms of their appropriate dimensions, the ordering of which is always temporal, spatial, other and the data type, float64, int32 for example: as pointed out in section 3 the data type definition is implicit - no type attribute is given in file and is provided here for reference.

The **standard_name** attribute of a variable refers to the name that has been accepted, for that variable, into the CF conventions and not every variable has a standard_name. A useful tool for finding what standard_names have been accepted is

<http://cfconventions.org/Data/cf-standard-names/27/build/cf-standard-name-table.html>.

Where a standard_name is not available then this attribute is omitted from the variable definition.

Even if a **standard_name** is not available then the **long_name** always is. There is no rule as to what form this should take but this attribute is used to provide a meaningful description of the variable.

The **units** attribute provides the units the data is provided in. When there is a standard_name defined then an associated units attribute is also defined. Some variables have no units and in these instances the **units** attribute is given a value "1" (string not numeric).

Where a variable may have missing or padding values then these would be replaced by a NaN (Not-a-Number) placeholder. This is not allowed under the CF convention and a numeric value has to be used. The value of this numeric value is indicated by the attribute **_FillValue**.

To improve automated visualisation the attributes **valid_min** and **valid_max** are used. These are the maximum and minimum values, in a given file, for that variable; the range excludes the value assigned to **_FillValue** attribute. This is a numeric value and is of the same data type as the data.

Where a variable is a directly measured quantity then the attribute **cell_methods** is used. This indicates what the data represent. All the data is a time series of some sort so each data point may be a mean of a series measurements, the standard deviation of a series of measurements or it may be a point measurement. The cell_methods attribute would indicate this with "**time: mean**", "**time: standard_deviation**", and "time: point" respectively. Although some variables have a spatial dimension and in principle the cell-method can be used to indicate what has been done in the spatial dimension that is not utilised in this standard.

The attribute **coordinate** is included to aid usability and given the value "**latitude longitude**". This attribute indicates what is used as the spatial coordinate reference grid.

These are the basic attributes you will find and where others arise they will be described locally.

5. Data Quality Flags

The data provided will have had some level of processing performed upon: be that instrument or post processing averaging, motion correction, or the variable may be derived from such core variables. These concepts were introduced in section 3. The quality of the data is provided via the Data Quality Control Flag. This flag is a mask and represents the

provider's considered opinion. Data users can apply the mask to the data or not - it is the user's choice. By taking this approach, the data provided is of greatest versatility.

A file containing just one data quality flag will contain the variable **qc_flag**. Where a file contains more than one data quality flag variable the data quality flag named is structured as:

qc_flag_<name>

- qc_flag_temperature
- qc_flag_relative_humidity
- qc_flag_pressure
- qc_flag_wind
- qc_flag_radiation
- qc_flag_precipitation

Flag variables are always of data type **byte** and are defined such that they have the same dimensions as the variables they are associated with: there is a flag value associated with **every** data point. They all follow a standard structure with the following attributes:

units

Definition: Units of a variable's content. Where a variable is unit less the value 1 is used.

Example: 1

long_name

Definition: Long descriptive name which is often used for labelling plots

Example: Data Quality flag: Temperature

flag_values

Definition: Values the data flag can have

Example: 0b, 1b, 2b, 3b

flag_meanings

Definition: How the flag should be interpreted

Example:

not_used
good_data
suspect_data_unspecified_instrument_performance_issues_contact_data_originator_for_more_information
Suspect_data_time_stamp_error

To reflect the fact that what affects data quality can vary, the flag_values and flag_meanings are not rigidly tied down. That is they may vary on a file-by-file basis. What does not vary is the structure and the usage: the qc_flag variable is structured and used so that for every flag_value there is a corresponding flag_meaning. In this standard we use an integer value in the range 0 to n (being of data type byte the maximum value of n is 255):

- 0 is reserved for future use and is **not used**
- 1 is **always** good data.

Consider the variable air_temperature which has data:

-20 -3 -2 -1 -2 -3 -2 -1 0 -1 0 2 3 4 2 3 20 4 3 2

While qc_flag_temperature has data:

3 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 3 2 1 1

The flag_values attribute is "0b, 1b, 2b, 3b" and the flag_meanings attribute gives:

not_used
good_data

suspect_data_unspecified_instrument_performance_issues_contact_data_originator_for_more_information
Bad_data_value_outside_instrument_measurement_range

If the user wanted only to see “good” data (indicated by a qc_flag value of 1) all they would need to do would be to:

1. Make a copy of the variable data array
2. Set the value of the elements in the duplicate data array that correspond to elements on the qc_flag that have a value not equal to 1 to NaN.

This will result in the temporary data variable looking like:

NaN -3 NaN -1 -2 -3 -2 -1 0 -1 0 2 3 NaN 2 3 NaN NaN 3 2

If the user wanted to accept “suspect” data in addition to “good” data (indicated by a qc_flag value of 1 and 2) all they would need to do would be to:

1. Make a copy of the variable data array
2. Set the value of the elements in the duplicate data array that correspond to elements on the qc_flag that have a value not equal to 1 or 2 to NaN.

This will result in the temporary data variable looking like:

NaN -3 -2 -1 -2 -3 -2 -1 0 -1 0 2 3 4 2 3 NaN 4 3 2

6. File Naming

File names follow a defined structure and are built up of unique components. In building the file name components are separated by an underscore “_” while individual elements within a component are separated by a hyphen “-”. The basic structure is as follows and note the use of lower case:

<instrument-name>_<platform-name>_<YYYY><MM><DD>-<HH><mm><SS>_<data-product>_<option-1>_<option-2>_<option-3>_v<version>.nc

where

- Instrument name: name of an instrument as registered with CEDA. The Instrument name is unique and in the CEDA archive it is linked to an instrument record that lists relevant information such as serial number. This leads to improved archive searching and hence data visibility and data traceability. To discuss registering an instrument with CEDA contact: data.management@cede.ac.uk
 - Example: ncas-aws-1
- Platform name: where or on what (in the case of ships and aircraft) was the instrument deployed
 - Example: knust, oden, ral, faam
- Date & Time: YYYYMMDD-HHmmSS
 - Example:
 - for a file containing up to 1 year's (YYYY) worth of data: 2016,
 - for a file containing up to 1 month's (MM) worth of data: 201604,
 - for a file containing up to 1 day's (DD) worth of data: 20160401,
 - for a file containing up to 1 hour's (HH) worth of data: 20160401-09,
 - for a file containing up to 1 minute's (mm) worth of data: 20160401-0950,
 - for a file starting at a specific time - for example launch time of soundings: 20160401-095059.
- Data product: name of the defined data product

- Example: surface-met
- option1, option2, option3: these are optional extras providing more information to the user. Where this is applicable the file name variant will be discussed in detail in section 8.1.
 - Example:
 - option1: cas, cis, fixed, ppi, rhi, user-1, user-2, user-3, user-4, user-5, winds-ppi , high-range-mode , low-range-mode
 - option 2: co, cr, 10mins, 15mins
 - Option 3: standard or advanced
- version: version of the data set
 - Example: n.m: n - major revision integer, m - minor revision integer

Examples of compliant filenames are:

ncas-aws-1_cao_20171225_surface-met_v1.3.nc

ncas-lidar-dop-3_cao_20191201_aerosol-backscatter-radial-winds_fixed_co_standard_v1.0.nc

7. Common

The following three sections provide details of the global attributes, dimensions, and variables that you should expect to find in any NCAS_AMF compliant file. These will be present irrespective of the source instrument with variation a result of being a land, air, or sea deployment.

7.1. Metadata (Global Attributes)

The attribute field is **always a string** and no attribute is ever left blank: “Not applicable” or “data not available” is inserted. With the exception of the first letter in Conventions, lower case is always used.

Conventions

This indicates the conventions\ vocabularies that are being followed in the generation of the data file. The core convention is that used for CF naming, that is CF-1.6. The NCAS Data Project has developed a controlled vocabulary (using CF-16) but specific to the needs of the instrumentation and this is NCAS-AMF-1.0. It is version controlled and develops as we refine and expand the data products available

source

This is a descriptor which uniquely identifies the instrument providing the data

Example: NCAS Automatic Weather Station unit 1

instrument_manufacturer

The name of the instrument manufacturer

Example: Davis Instruments

instrument_model

The instrument model name

Example: Vantage Pro 2 Wireless Station

instrument_serial_number

The instrument serial number which is registered to the instrument name used in the file name and linked to the “source”

Example: 63270V

instrument_software

If known this is the name of the software running on the instrument that actually controls and makes the measurement.

Example: R2CH

instrument_software_version

Manufacturers often update instrument software and subtle changes in this code can result in changes in the quality of the data provided. To be able to trace any such effect the version of software running is embedded in the meta data.

Example: v2.08.11

creator_name

This is the name of the person who generated the file. This is the person to contact if there are any questions about the data presented and how it was produced.

Example: A. Person

creator_email

The contact email for the person who created the file. People move and this may not always be valid.

Example: A.Person@aplace.ac.uk

creator_url

The ORCID URL of the person who created the file is something that goes with them and unlike email using this to trace the creator has a greater chance of success.

Example: <https://orcid.org/0000-0000-0000-0000>

institution

This is the name of the institution employing the creator. This is added to help users of the data track down the creator if they need to.

Example: National Centre for Atmospheric Science (NCAS)

processing_software_url

To go from the level 0 data produced by the source to the files archived requires the creator to do some sort of data processing. This processing may involve various levels of QC and data formatting so that it meets the archive standard. Where this code is developed by the creator it is deposited on an open repository - usually GitHub and this is the url to that code. The use of a repository means that the code is version controlled and the exact version used to create the file is accessible.

This only applies to creator developed code - no manufacturer proprietary software is ever deposited to the repository

Example: <https://github.com/name1/name2/>

processing_software_version

This is the version of the processing software.

Example: v1.3

calibration_sensitivity

All sensors report what they measure to some level of sensitivity and this may change for the same instrument over time - this is why calibration is important. The value or values given here are the sensitivities from the last calibration

Example: Calibrated to manufacturers standard:

Temperature(0.1K),RH(2%),Pressure(0.01hPa),Wind Speed(0.1ms⁻¹),Wind Direction(22degrees),Accumulated Rain(0.01mm),Radiation(0.1 Wm⁻²)

calibration_certification_date

Calibrations are often accompanied by a certificate that is dated and if not there is always a date as to when the calibration for the data provided was done. The date provided here is either when the calibration was done and if there is a certificate the date of this

Example: 01/01/2016

calibration_certification_url

When the calibration is accompanied by a certificate a copy of the certificate is put into a GitHub repository. This is the URL to that repository.

Example: <https://github.com/name1/name2/>

sampling_interval

This is the time interval between successive samples in the file. Where there has been substantial drop out we do not pack the file with null data rather we allow there to be a jump in what is otherwise a monotonically increasing time series.

Example: 10 minutes

averaging_interval

This is the time over which each measurement is averaged. For example samples appear in the file at 10 minute intervals. The sample saved is the average of all the data taken in the preceding 10 minutes. In this case the averaging interval is 10 minutes. In some cases it may be a spot measurement that is taken in which case this would be indicated as 0 - spot measurement

Example: 10 minutes (0 - spot used to indicate spot measurement)

product_version

Over time, errors or new calibrations means that the data may need to be reissued: it is the same data but just a different version. The version number is part of the file name and should match this value. Major revisions occur when a new calibration or processing method is applied while minor revisions occur to correct typos. The reason for a the revision is detailed in the history field

Example: v <n.m> n - major revision, m - minor revision

processing_level

This indicated the amount of quality control that has been put into the data. See the "Data Processing Levels" section for a full discussion.

Options: 1, 2, or 3

last_revised_date

This is the date that this data file was processed. The time is UTC and is given in ISO format.

Example: 2013-06-06T12:00:00

project

This is the full name and associated acronym of the project and should match that on official funding documents.

Example: Dynamics-aerosol-chemistry-cloud interactions in West Africa.
(DACCIWA)

project_principal_investigator

The name of the project Principal Investigator

Example: B. Person

project_principal_investigator_email

Contact email for project PI

Example: B.Person@someplace.com

project_principal_investigator_url

ORCID URL of PI.

Example: <https://orcid.org/0000-0000-0000-0000>

licence

The UK Government Licensing Framework (UKGLF) provides a policy and legal overview of the arrangements for licensing the use and re-use of public sector information, both in central government and the wider public sector. It sets out best practice, standardises the licensing principles for government information, mandates the Open Government Licence (OGL) as the default licence for Crown bodies and recommends OGL for other public sector bodies.

Example: Data usage licence - UK Government Open Licence agreement:
<http://www.nationalarchives.gov.uk/doc/open-government-licence>

acknowledgement

Obtaining and producing these data represents a substantial amount of effort and investment of resources. It is expected that users of this data acknowledge this by following the request directive given in this field.

Example: Acknowledgement of NCAS as the data provider is required whenever and wherever these data are used

platform

The platform is where or on what the instrument was deployed for example if it was deployed at Christmas Island then the value in this field would be "christmas island". If the instrument was deployed on a ship called Oden then the value in this field would be "oden"

platform_type

This indicates the type of platform the instrument was deployed on and the value in this field is either stationary_platform or moving_platform.

deployment_mode

Instruments can be deployed either on land, sea or air. The value in this field indicates which.

title

This is the title that describes the data in the file. This is a descriptor that is used for searching.

Example: Time series of Temperature, Pressure, Relative Humidity, Wind Speed, Wind Direction, Down Welling Radiation from the NCAS Davis Automatic Weather Station.

featureType

This is a CF term that describes the type of data and the field will be either timeSeries, timeSeriesProfile, or trajectory. Type timeSeries is associated with a time series of data that has no vertical dimension: for example a temperature measurement at a specific height. Type timeSeriesProfile is associated with a time series of vertical profiles: for example data from wind profiler. The type trajectory is used when the data is time series that is also positional and altitude dependent: for example that from a radiosonde

time_coverage_start

This is the time value of the first data point in the file. The time is UTC and in ISO format.

Example: 2013-02-01T00:00:00

time_coverage_end

This is the time value of the last data point in the file. The time is UTC and in ISO format.

Example: 2013-03-31T23:59:59

geospatial_bounds

This field defines the latitude and longitude range associated with the file. For a stationary_platform this is just the latitude longitude part (signed decimal). For moving_platform it is the bounding box: : top left corner, bottom right corner presented as : latitude longitude, latitude longitude (signed decimal)

Example:

Point deployment: -111.29N 40.26E

Bounding box: -111.29N 40.26E, -110.29N 41.26E

platform_altitude

This is the altitude, wrt WGS84, of the ground at the point of deployment. All instrument deployment heights are given with respect to this. Where altitude is a variable this is given with respect to WGS84 and not respect to the local ground.

Example: 263m or 263 m

location_keywords

These are words with geographical relevance that aid archival searching.

Example: africa, ghana, kumasi, knust

amf_vocabularies_release

This is the url to the version controlled vocabulary used in defining the data file.

Example: https://github.com/ncasuk/AMF_CVs/releases

history

This is freeform text that gives the history of the data from collection to the present version. Where there have been data revisions the data and reason for that revision is included.

Example: Data collected June - July 2016, processed & QC'd Oct 2016

comment

This is free form text and is used to provide the user with any additional information that may be of use: for example the height of an instrument above ground, the local magnet variance, the doppler +ve direction.

Example: All AWS sensors mounted on a 2m mast - sensors are at a height of +2m with respect to platform_height. Positive indicates upward direction.

7.2. Dimensions

All variables are defined in terms of dimensions. For example consider a temperature measurement that was taken every minute for 1 hour. The temperature variable is defined to have the dimension of time.

Instruments can be deployed on both stationary and moving platforms and the dimensions need to be able to reflect this. For stationary deployments the latitude and longitude are single valued and hence have a length of 1. For a moving platform these dimensions will be equal in length to time.

Trajectory

time

CF Standard Name: time

length: <n>

units: seconds since 1970-01-01 00:00:00

Stationary Platform

time

CF Standard Name: time

length: <n>

units: seconds since 1970-01-01 00:00:00

latitude

CF Standard Name: latitude

length: 1

units: degrees_north

longitude

CF Standard Name: longitude

length: 1

units: degrees_east

Moving Platform

time

CF Standard Name: time

length: <n>

units: seconds since 1970-01-01 00:00:00

latitude

CF Standard Name: latitude
length: <n>
units: degrees_north
longitude
CF Standard Name: longitude
length: <n>
units: degrees_east

7.3. Variables

All data files produced using the NCAS–AMF standard include the following variables. Where a variable has no units this is indicated by assigning a value of “1” to the units attribute and where no CF standard_name has been registered for the variable this attribute is omitted.

To reiterate the attribute type is included in the documentation to inform the user as to what the data type should be. In the data files themselves the variables have no type attribute as this is implicit at the time the file is written.

These variables will be present in the data file.

time

Time is always UTC and is given as the number of seconds that has elapsed since 00:00 hours on the 1st January 1970. It is also often known as EPOCH time and is the common representation of time in the UNIX community. The variable is a float64 to allow for time bases that are subunits of 1 second. The calendar type is standard and this indicates the use of the Julian calendar over the Gregorian. The axis attribute is used by analysis software and is incorporated to aid usability.

dimensions: time
type: float64
units: seconds since 1970-01-01 00:00:00
standard_name: time
long_name: Time (seconds since 1970-01-01)
axis: T
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
calendar: standard

day_of_year

Also known as decimal day it is a number that runs from 1.0 (midnight on 1st January) to 365 (366 for leap years).

dimensions: time
type: float32
units: 1
long_name: Day of Year
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file

year

dimensions: time
type: int32
units: 1
long_name: Year
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file

month

dimensions: time
type: int32
units: 1
long_name: Month
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file

day

dimensions: time
type: int32
units: 1
long_name: Day
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file

hour

dimensions: time
type: int32
units: 1
long_name: Hour
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file

minute

dimensions: time
type: int32
units: 1
long_name: Minute
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file

second

dimensions: time
type: float32
units: 1
long_name: Second
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file

Additional Common Variables - Stationary Platform

latitude

For the stationary platform this is single valued and is presented in this form to aid archiving and subsequent searching.

dimensions: latitude
type: float32
units: degrees_north
standard_name: latitude
long_name: Latitude

longitude

For the stationary platform this is single valued and is presented in this form to aid archiving and subsequent searching.

dimensions: longitude
type: float32
units: degrees_east

standard_name: longitude
long_name: Longitude

Additional Common Variables - Moving Platform - Sea

latitude

For moving platforms the length of the variable is the same as time and the axis value of Y is incorporated to aid subsequent usage. The cell_method attribute indicates averaging that has taken place to produce the value. This is given only for the time axis: value can be point, mean, standard_deviation.

dimensions: latitude
type: float32
units: degrees_north
axis: Y
standard_name: latitude
long_name: Latitude
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: point

longitude

For moving platforms the length of the variable is the same as time and the axis value of Y is incorporated to aid subsequent usage. The cell_method attribute indicates averaging that has taken place to produce the value. This is given only for the time axis: value can be point, mean, standard_deviation.

dimensions: longitude
type: float32
units: degrees_east
axis: X
standard_name: longitude
long_name: Longitude
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: point

platform_course

This describes the motion and orientation of the vehicle from which observations were. The platform course is the direction in which the platform is travelling (not necessarily the same as the direction in which it is pointing)

dimensions: time
type: float32
units: degree
standard_name: platform_course
long_name: Direction in which the platform is travelling
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: mean
coordinate: latitude longitude

platform_orientation

The platform orientation is the direction in which the "front" or longitudinal axis of the platform is pointing.

dimensions: time
type: float32
units: degree
standard_name: platform_orientation
long_name: Direction in which the platform is pointing
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: mean
coordinate: latitude longitude

platform_speed_wrt_ground

The platform speed with respect to ground is relative to the solid Earth beneath it, i.e. the sea floor for a ship. It is often called the "ground speed" of the platform

dimensions: time
type: float32
units: m s-1
standard_name: platform_speed_wrt_ground
long_name: Platform speed with respect to ground
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: mean
coordinate: latitude longitude

instrument_pitch_angle

This is the pitch angle of the instrument: depending on exactly how the instrument was mounted to the platform the motion of the instrument may be independent of the platform motion. In aviation terms the pitch axis has its origin at the center of gravity and is directed to the right, parallel to a line drawn from wingtip to wingtip. Motion about this axis is called pitch. A positive pitching motion raises the nose of the aircraft and lowers the tail.

dimensions: time
type: float32
units: degree
long_name: Instrument Pitch Angle
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: mean
coordinate: latitude longitude

instrument_pitch_rate

This indicates the rate at which the pitch angle is changing

dimensions: time
type: float32
units: degree s-1
long_name: Instrument Pitch Angle Rate of Change
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: mean
coordinate: latitude longitude

instrument_pitch_minimum

This is the minimum pitch angle measured during the sample averaging period.

dimensions: time

type: float32

units: degree

long_name: Instrument Pitch Angle Minimum

_FillValue: -1e+20

valid_min: smallest valid value of this variable in the file

valid_max: largest valid value of this variable in the file

cell_method: time: minimum

coordinate: latitude longitude

instrument_pitch_maximum

This is the maximum pitch angle measured during the sample averaging period.

dimensions: time

type: float32

units: degree

long_name: Instrument Pitch Angle Maximum

_FillValue: -1e+20

valid_min: smallest valid value of this variable in the file

valid_max: largest valid value of this variable in the file

cell_method: time: maximum

coordinate: latitude longitude

instrument_pitch_standard_deviation

This is the standard deviation of the pitch angle sample in the sample averaging period and indicates the “steadiness of the motion around this axis.

dimensions: time

type: float32

units: degree

long_name: Instrument Pitch Angle Standard Deviation

_FillValue: -1e+20

valid_min: smallest valid value of this variable in the file

valid_max: largest valid value of this variable in the file

cell_method: time: standard_deviation

coordinate: latitude longitude

instrument_roll_angle

This is the roll angle of the instrument: depending on exactly how the instrument was mounted to the platform the motion of the instrument may be independent of the platform motion. In aviation terms the roll axis has its origin at the center of gravity and is directed forward, parallel to the fuselage reference line. Motion about this axis is called roll. A positive rolling motion lifts the left wing and lowers the right wing.

dimensions: time

type: float32

units: degree

long_name: Instrument Roll Angle

_FillValue: -1e+20

valid_min: smallest valid value of this variable in the file

valid_max: largest valid value of this variable in the file

cell_method: time: mean

coordinate: latitude longitude

instrument_roll_rate

This indicates the rate at which the pitch angle is changing

dimensions: time

type: float32
units: degree s-1
long_name: Instrument Roll Angle Rate of Change
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: mean
coordinate: latitude longitude

instrument_roll_minimum

This is the minimum roll angle measured during the sample averaging period.

dimensions: time
type: float32
units: degree
long_name: Instrument Roll Angle Minimum
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: minimum
coordinate: latitude longitude

instrument_roll_maximum

This is the maximum roll angle measured during the sample averaging period.

dimensions: time
type: float32
units: degree
long_name: Instrument Roll Angle Maximum
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: maximum
coordinate: latitude longitude

instrument_roll_standard_deviation

This is the standard deviation of the pitch angle sample in the sample averaging period and indicates the “steadiness of the motion around this axis.

dimensions: time
type: float32
units: degree
long_name: Instrument Roll Angle Standard Deviation
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: standard_deviation
coordinate: latitude longitude

instrument_yaw_angle

This is the yaw angle of the instrument: depending on exactly how the instrument was mounted to the platform the motion of the instrument may be independent of the platform motion. In aviation terms the yaw axis has its origin at the center of gravity and is directed towards the bottom of the aircraft, perpendicular to the wings and to the fuselage reference line. A positive yawing motion moves the nose of the aircraft to the right.

dimensions: time
type: float32

units: degree
long_name: Instrument Yaw Angle
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: mean
coordinate: latitude longitude

instrument_yaw_rate

This indicates the rate at which the pitch angle is changing

dimensions: time
type: float32
units: degree s-1
long_name: Instrument Yaw Angle Rate of Change
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: mean
coordinate: latitude longitude

instrument_yaw_minimum

This is the minimum yaw angle measured during the sample averaging period.

dimensions: time
type: float32
units: degree
long_name: Instrument Yaw Angle Minimum
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: minimum
coordinate: latitude longitude

instrument_yaw_maximum

This is the maximum yaw angle measured during the sample averaging period.

dimensions: time
type: float32
units: degree
long_name: Instrument Yaw Angle Maximum
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: maximum
coordinate: latitude longitude

instrument_yaw_standard_deviation

This is the standard deviation of the pitch angle sample in the sample averaging period and indicates the “steadiness of the motion around this axis.

dimensions: time
type: float32
units: degree
long_name: Instrument Yaw Angle Standard Deviation
_FillValue: -1e+20
valid_min: smallest valid value of this variable in the file
valid_max: largest valid value of this variable in the file
cell_method: time: standard_deviation

coordinate: latitude longitude

Additional Common Variables - Moving Platform - Air

In addition to the variables introduced in Additional Common Variables - Moving Platform - Sea, data from aircraft deployments have the following additional variable.

altitude

Altitude is the (geometric) height above the geoid, which is the reference geopotential surface. The geoid used is WGS84. The axis value of Z is incorporated to aid subsequent usage. The cell_method attribute indicates averaging that has taken place to produce the value. This is given only for the time axis: value can be point, mean, standard_deviation.

dimensions: time

type: float32

units: m

standard_name: altitude

long_name: Geometric height above geoid (WGS84).

_FillValue: -1e+20

axis: Z

valid_min: smallest valid value of this variable in the file

valid_max: largest valid value of this variable in the file

cell_method: time: mean

coordinate: latitude longitude

8. Data Product Specific

The **photolysis-frequencies** data product is associated with the ncas-filtered-radiometer-1 instrument.

The variables in this data product are:

- Photolysis Frequencies (j(NO2))
- Photolysis Frequencies (j(O1D))
- Photolysis Frequencies (j(HCHO))

Where a given variable has been accepted by the CF community (indicated by the presence of a standard_name) it can be found in the standard name tables:

<http://cfconventions.org/Data/cf-standard-names/27/build/cf-standard-name-table.html>

8.1. File Name Variants

There are no file naming variants associated with this data product: options 1-3 in the file name structure are not used.

8.2. Metadata (Global Attributes)

There are no additional metadata associated with this data product.

8.3. Dimensions

There are no additional dimensions associated with this data product.

8.4. Variables

In addition to the variables introduced in section 7 this data product has the additional product specific variables.

Note:

1. Where a variable has no units this is indicated by assigning a value of "1" to the "units" attribute.
2. If data for a **data product specific** variable is not available then, rather than pad the file with variables containing only the _FillValue, that variable will be omitted.
3. Where no CF standard_name has been registered for the variable this attribute is omitted.

photolysis_frequencies_jno2

A photolysis reaction involves the breaking of a chemical bond in a molecule by an incident photon. The photolysis frequency is molecule specific and is the frequency required to initiate a reaction involving that species.

dimensions: time

type: float32

units: s-1

long_name: Photolysis Frequencies (j(NO2))

_FillValue: -1.00E+20

valid_min: smallest valid value of this variable in the file

valid_max: largest valid value of this variable in the file

cell_methods: time: mean or time: point

coordinate: latitude longitude

photolysis_frequencies_jo1d

A photolysis reaction involves the breaking of a chemical bond in a molecule by an incident photon. The photolysis frequency is molecule specific and is the frequency required to initiate a reaction involving that species.

dimensions: time

type: float32

units: s-1

long_name: Photolysis Frequencies (j(O(1))(D))

_FillValue: -1.00E+20

valid_min: smallest valid value of this variable in the file

valid_max: largest valid value of this variable in the file

cell_methods: time: mean or time: point

coordinate: latitude longitude

photolysis_frequencies_jhcho

A photolysis reaction involves the breaking of a chemical bond in a molecule by an incident photon. The photolysis frequency is molecule specific and is the frequency required to initiate a reaction involving that species.

dimensions: time

type: float32

units: s-1

long_name: Photolysis Frequencies (j(HCHO))

_FillValue: -1.00E+20

valid_min: smallest valid value of this variable in the file

valid_max: largest valid value of this variable in the file

cell_methods: time: mean or time: point

coordinate: latitude longitude

photolysis_frequencies_jhono

A photolysis reaction involves the breaking of a chemical bond in a molecule by an incident photon. The photolysis frequency is molecule specific and is the frequency required to initiate a reaction involving that species.

dimensions: time

type: float32

units: s-1

long_name: Photolysis Frequencies (j(HONO))

_FillValue: -1.00E+20

valid_min: smallest valid value of this variable in the file

valid_max: largest valid value of this variable in the file

cell_methods: time: mean or time: point

coordinate: latitude longitude

qc_flag

This control flag indicates the quality of the measurement. The number of values will be at least 2: 0 or 1, and no more than 256. The attribute flag_values is a string giving a list of the values used in context of this qc_flag: 0 is never used, 1 always indicates good data. The attribute flag_meanings indicates what each value means and how a data point with this flag value should be interpreted. There is a flag value and meaning for every data point and how this flag mask can be applied can be found in section 5 of this document.