ncas-radar-data-standard Documentation

Release 1.0.0

National Centre for Atmospheric Science

Oct 01, 2023

CONTENTS

Documentation for ncas-radar-data-standard.

This document should be read in conjunction with the documentation for CfRadial Version-1.4

CHAPTER ONE

OVERVIEW

NCAS currently operates a variety of radars as part of its programme of observational science and facility provision. These range in capability, some incorporating scanning antennas, and some operating in fixed, zenith pointing mode. This document outlines the specification of a data standard developed for NCAS radar data. It forms part of a wider standardisation of data formats for NCAS observational data.

1.1 FAIR Principles

The guiding principles for the NCAS Radar Data Standard are those of FAIR (**F**indable **A**ccessible **I**nteroperable and **R**eusable) data management [1].

A detailed description of the FAIR principles may be found at https://go-fair.org/fair-principles/. For ease of reference these are reproduced in part below. The FAIR principles of data management and stewardship aim to make data:

Findable

The first step in (re)using 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:

- (Meta)data should be assigned a globally unique and persistent identifier.
- Data should be described with rich metadata.
- Metadata should clearly and explicitly include the identifier of the data they describe.
- (Meta)data should be registered or indexed in a searchable resource.

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 should be open, free, and universally implementable.
- The protocol should allow for an authentication and authorization procedure, where necessary.
- Metadata should be accessible, even when the data are no longer available

Interoperable

The data usually need to be integrated with other data. In addition, the data need to interoperate with applications or workflows for analysis, storage, and processing.

This means:

- (Meta)data should use a formal, accessible, shared, and broadly applicable language for knowledge representation.
- (Meta)data should use vocabularies that follow FAIR principles.
- (Meta)data should include qualified references to other (meta)data.

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:

- (Meta)data should be richly described with a plurality of accurate and relevant attributes.
- (Meta)data should be released with a clear and accessible data usage license.
- (Meta)data should be associated with detailed provenance.
- (Meta)data should meet domain-relevant community standards.

1.2 NetCDF

NCAS radar data products are provided in the **netCDF** format. NetCDF (Network Common Data Form) is a set of software libraries and platform independent data formats which are designed to support the creation, access, and sharing of array-oriented scientific data. NetCDF is designed to be

- **Self-describing.** A netCDF file includes information about the data it contains (i.e. metadata).
- **Portable.** A netCDF file can be accessed by computers with different ways of storing integers, characters, and floating-point numbers.
- **Scalable.** A small subset of a large dataset may be accessed efficiently.
- **Appendable.** Data may be appended to a properly structured netCDF file without copying the dataset or redefining its structure.
- **Shareable.** One writer and many readers may simultaneously access the same netCDF file.
- **Archivable.** Access to all earlier forms of netCDF data will be supported by current and future versions of the software.

NetCDF is in common use, and is almost ubiquitous within the Earth sciences. Unidata (https://unidata.ucar.edu) provide and maintain software libraries for accessing netCDF data using C, C++, Java, and FORTRAN. Third-party libraries (which are generally bindings or wrappers to the Unidata libraries) are available for Python, IDL, MATLAB, R, Ruby, and Perl, among others.

NetCDF files generally consist of four components:

- **Attributes:** Attributes are metadata that can be attached to the netCDF file itself (called global attributes), to variables, and to groups (variable attributes and group attributes, respectively). Attributes may be textual or numeric; numeric attributes may be arrays.
- **Groups:** Groups (available since netCDF4) provide a method to encapsulate related *dimensions, variables,* and *attributes*. They can be thought of as somewhat analogous to directories in a filesystem.
- **Dimensions:** Dimensions specify the size of a single axis of a variable within a netCDF file. Common dimensions for geophysical data include time, latitude, and longitude, though they do not need to correspond to physical dimensions. There is no practical limit to the number of dimensions which may be defined in a netCDF file.

- **Variables:** Variables are either scalar (single values pertaining the whole data set) or named *n*-dimensional arrays (thus associated with *n dimensions*) of a specified data type. Variables may have zero or more *attributes*, which act as metadata to describe the contents of the variable.

Radar data are typically recorded as time-stamped rays, each providing values of an observable at a set of distances (ranges) from the radar. Such so-called field variables are thus organised into 2-dimensional arrays, along *time* and *range* dimensions.

Below is a minimal example in Python of accessing a 2-dimensional field variable, *DBZH*, along with its *units* attribute and a global *title* attribute from a netCDF file. Note that the netCDF library, *netCDF4*, is not included as part of the Python standard library, but may be installed using your system package manager, pip, or conda.

```python
from netCDF4 import Dataset

with Dataset('some_radar_file.nc', 'r') as nc:
    title = nc.title
    DBZH_units = nc['DBZH'].units
    DBZH_data = nc['DBZH'][:]
```

This provides a simple means of inspecting the content of a radar field variable. However, it takes no account of the spatial geometry. To do this the user would need to read in additional variables describing the range and the azimuth and elevation angles of the radar antenna. To assist in standardising the way this is handled, the NCAS Radar Data Standard draws on the CfRadial initiative (https://github.com/NCAR/CfRadial.git), and currently uses CfRadial Version-1.4 as a base convention.

CfRadial has been developed as a CF-compliant netCDF format for radar and lidar moments data in radial (i.e. polar) coordinates. The intention is that the format should, as far as possible, comply with the CF (Climate and Forecast) conventions (http://cfconventions.org/) for gridded data. However, the current convention does not support radial radar/lidar data. Therefore, extensions to the conventions are being proposed by the developers of the CfRadial.

CfRadial is already supported by a number of community tools developed for reading, visualizing, and analysing radar data. These tools and software environments include the Python ARM Radar Toolkit (Py-ART), the Lidar and Radar Open Software Environment (LROSE), and wradlib, an open source library for weather radar data processing. Hence, there is a strong motivation for aligning NCAS radar data with CfRadial in order to be able to allow users to employ some of the most widely used software tools.

CHAPTER

TWO

FILENAME CONVENTIONS

Each radar filename is constructed as follows:

`<instrument_name>_<platform_name>_<date>-[<time>]_<scan_name>_[<option1>_<option2>_<option3>]_v<version`
`nc`

Items in square brackets may or may not be present, depending on the data. For example, if the data represent an entire day, the `<time>` will not be included. The `<scan_name>` is the name of the scan strategy employed. If the file contains a single sweep (see below) this should indicate the type of this sweep, e.g. `rhi`, `ppi`. If there are multiple sweeps in the file the `<scan_name>` will represent the strategy used, e.g. `vol` for a volume scan. More specific names for the scan strategy may be used such as `hsrhi` [2] for "hemispheric sky RHI", which is a volume scan comprising a sequence of horizon-to-horizon RHI scans equally spaced in azimuth.

For NCAS instruments, the `<instrument_name>` follows a controlled vocabulary.

CHAPTER

THREE

NETCDF CONVENTIONS

The NCAS-Radar convention has at its heart the CfRadial format, and its subconventions. CfRadial is a comprehensive specification, and we only discuss required elements here. For more complex use, e.g. for pulsed radar systems with complicated pulsing schemes, the user should consult the full CfRadial documentation, which may be found on Github at `https://github.com/NCAR/CfRadial`.

Version 1.0 of the NCAS-Radar standard is based on CfRadial-1.4.

3.1 Sub-conventions

The CfRadial-1.4 standard describes a number of sub-conventions related to instrument parameters and calibration. The following sub-conventions are **obligatory** within the NCAS Radar 1.0 standard:

- instrument_parameters
- radar_parameters
- radar_calibration

NCAS Radar format files need to comply with the requirements set out for these sub-conventions within the CfRadial Version-1.4 documentation.

Strict variable and attribute names for non-field variables

In CfRadial a 'field' variable stores such quantities as radar moments, derived quantities, quality control measures, etc. These variables store the fundamental scientific data associated with the instrument. By contrast, metadata variables store the dimensional information such as *time*, *range*, *azimuth* and *elevation*, and other metadata such as calibration and radar characteristics.

CfRadial requires strict adherance to naming conventions for dimensions and for metadata variables. The NCAS-Radar convention inherits this requirement. For details see the CfRadial documentation on Github. A summary of metadata variables may be found in the following *table*

This strictness requirement only applies to non-field metadata variables. The field variables will be handled as usual in CF, where the standard name is the definitive guide to the contents of the field. Suggested standard names for radar variables not yet supported by CF have been proposed in the CfRadial documentation. For convenience a summary of field variables relevant to radar instruments is reproduced in the following *table*

CHAPTER

FOUR

OVERVIEW OF DATA CONTENT

The data fields containing observables from a radar instrument, i.e. the moments of the Doppler velocity spectrum are produced over a time or angular interval at a sequence of ranges increasing radially away from the instrument. The term "ray" is used to refer to a set of range gates at a given time or angle. In most cases the spacing between range gates is constant along a ray, but this is not compulsory.

Data fields are typically stored as 2-D arrays, with dimnesions **(time,range)**. This is typical for current NCAS radars where each ray has the same number of gates. CfRadial does allow for the more general case where rays have a variable number of gates. For details see the CfRadial documentation.

In addition to the **time** and **range** dimensions, CfRadial introduces a third "pseudo"-dimension, which allows the field data to be subdivided into so-called "sweeps". For example, a single constant elevation PPI scan constitutes an example of a sweep, and a typical NetCDF data file will have a **volume** that comprises one or more such sweeps. The convention uses start and stop indices to identify which rays belong to a given sweep. Also, some rays may contain data collected during the transition between sweeps, and these are indicated using an "antenna_transition" flag.

4.1 Attributes required by CfRadial-1.4

As the NCAS-Radar-1.0 convention uses CfRadial-1.4 as its basis, all global attributes required by the latter must be included. The following global attributes are required by CfRadial-1.4:

Conventions

A space-delineated list of the conventions (and sub-conventions) that are followed by the dataset. As NCAS-Radar-1.0 uses version 1.4 of the CfRadial standard, this should be included explicitly. Sub-conventions such as "radar_parameters" are inherited from CfRadial-1.4. NCAS-Radar-1.0 does not have a separate set of sub-conventions.

Example

`NCAS-Radar-1.0 CfRadial-1.4 instrument_parameters radar_parameters radar_calibration`

title

This is a short description of the file contents.

Example

`Moments from the NCAS Mobile X-band Radar unit 1 at Sandwith, UK`

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)`

references
References that describe the data or the methods used to produce it. For example, this may be a paper describing the instrument.

Example
`https://doi.org/10.5194/amt-11-6481-2018`

source
This is a descriptor that uniquely identifies the source of the original data.

Example
`NCAS Mobile X-band Radar unit 1`

history
This is free form text that gives the history of the data from collection to the present version. A time-stamped new line should be appended to describe each processing step.

comment
This is free form text and is used to provide the user with any additional information that may be of use.

instrument_name
This should be filled with the unique NCAS instrument name

Example
`ncas-mobile-x-band-radar-1`

4.2 Attributes required by NCAS-Radar-1.0 that are optional in CfRadial-1.4

The following global attributes are optional within CfRadial-1.4, but are required by NCAS-Radar-1.0.

platform_is_mobile

Example
`false`

4.3 Additional attributes required by NCAS-Radar-1.0

The following global attributes are required by NCAS-Radar-1.0 but are not part of the CfRadial-1.4 convention:

instrument_manufacturer
The name of the instrument manufacturer

Example
`Meteorologische Messtechnik (Metek) GmbH`

instrument_model
The instrument model name

Example
`MIRA-35`

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_pid
This is a unique persistent identifier (PID) for the instrument, for example registered on the Handle.Net registry.

Example
`https://hdl.handle.net/21.12132/3.191564170f8a4686`

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

Example
`radar-camra-rec`

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 metadata.

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 they were produced.

Example
`A. Person`

creator_email
The contact email for the person who created the file. It is, however, recognized that people move institution, and that 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. Other PIDs may be used, but ORCID is the preferred option.

Example
`https://orcid.org/0000-0000-0000-0000`

processing_software_url
To go from the Level 0 data produced by the source to the files that are to be 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 deposited in the repository.

Example
`https://github.com/name1/name2/`

processing_software_version
This is the version of the processing software.

Example
`v1.3`

product_version
Over time, the discovery of errors, introduction of new processing algorithms or the refinement of calibration

values may mean that the data need to be reissued. Three levels of revision are indicated in the format `v<n>.<m>.<p>`, where `n` is a major revision (e.g. application of a new processing algorithm), `m` is a minor revision, and `p` is a patch (e.g. correction of typographical errors). The reason for a the revision should always be detailed in the history field.

Example
`v2.1.1`

processing_level
This indicates the level of quality control that has been applied to the data. See the "Data Processing Levels" section for a full discussion. Options: `1`, `2`, or `3`

last_revised_date
This is the date of production of the data file. 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
`Microbiology-Ocean-Cloud-Coupling in the High Arctic (MOCCHA)`

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 or other persistent identifier of the 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 Open Government 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 these 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 the site or mobile platform where 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`

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

time_coverage_start
This is the time value of the first ray of data in the file. The time is UTC and in ISO format. Note that CfRadial-1.4 also incorporates this as a global string variable. Including it here as a global attribute aligns with usage in data files from other NCAS instruments.

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

time_coverage_end
This is the time value of the last ray of data in the file. The time is UTC and in ISO format. Note that CfRadial-1.4 also incorporates this as a global string variable. Including it here as a global attribute aligns with usage in data files from other NCAS instruments.

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

geospatial_bounds
This field defines the latitude and longitude bounds associated with the file. For a vertically pointing radar on a stationary platform this is just the latitude and longitude of the point of deployment (as signed decimals). Otherwise it is the bounding box, i.e. a rectangle enclosing the extent of the data resource described in latitude and longitude.

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

platform_altitude
This is the altitude above the geoid of the platform at the location where the instrument is deployed (i.e. the orthometric height), using the WGS84 ellipsoid and EGM2008 geoid model. For a land-based deployment this is the orthometric height of the local ground level. For a mobile platform this is the altitude at the start of the data volume. Note that the altitude of the instrument is given in the variable *altitude* and may be offset from the platform altitude.

location_keywords
These are words with geographical relevance that aid data discovery.

Example
`cumbria, sandwith`

4.4 Special case of a stationary vertically pointing radar

Vertically pointing radars on a stationary platform produce a series of profile features at the same horizontal position with monotonically increasing times. As such the data products conform to a CF-compliant feature type, and we should add the following global attribute (and value).

featureType:
`timeSeriesProfile`

This attribute should be omitted for other radar configurations (e.g. scanning radars on a stationary platform, or radars on a mobile platform).

CHAPTER

FIVE

DIMENSIONS

As mentioned above, the naming of these dimensions must adhere strictly to the CfRadial-1.4 requirements.

Dimension name	Description
time	The number of rays. This dimension is optionally unlimited.
range	The number of range bins.
sweep	The number of sweeps.
string_length[1]	Length of char type variables.

[1] Any number of 'string_length' dimensions may be created and used. For example, you may declare the dimensions 'string_length', 'string_length_short' and 'string_length_long', and use them appropriately for strings of various lengths. These are only used to indicate the length of the strings actually stored, and have no effect on other parts of the format.

CHAPTER

SIX

GLOBAL VARIABLES

Variables named in **bold** in the following table are required by Cf-Radial-1.4 and NCAS-Radar-1.0. Others are optional.

Variable name	Type	Dimension	Comments
volume_number	int	none	Volume numbers are sequential, relative to some arbitrary start time, and may wrap.
platform_type	char	(string_length)	Options are: *"fixed"*, *"vehicle"*, *"ship"*, *"aircraft"*, *"aircraft_fore"*, *"aircraft_aft"*, *"aircraft_tail"*, *"aircraft_belly"*, *"aircraft_roof"*, *"aircraft_nose"*, *"satellite_orbit"*, *"satellite_geostat"*. Assumed *"fixed"* if missing.
time_coverage_start	char	(string_length)	UTC time of first ray in file. Resolution is integer seconds. The ''time(time)" variable is computed relative to this time unless time_reference is defined. Format is yyyy-mm-ddTHH:MM:SSZ
time_coverage_end	char	(string_length)	UTC time of last ray in file. Resolution is integer seconds.
time_reference	char	(string_length)	UTC time reference. Resolution is integer seconds. If defined, the time(time) variable is computed relative to this time instead of relative to **time_coverage_start**.

CHAPTER

SEVEN

COORDINATE VARIABLES

Variables in the following table are required by Cf-Radial-1.4 and NCAS-Radar-1.0.

Name	Data type	Dimension
time	double	(time)
range	float	(range) or (sweep,range)

7.1 Attributes for the time coordinate variable

Attribute name	Type	Value
standard_name	string	"time"
long_name	string	"time_in_seconds_since_volume_start" or "time_since_time_reference"
units	string	"seconds since yyyy-mm-ddTHH:MM:SSZ", where the actual reference time values are used.
calendar	string	Defaults to "gregorian" if missing.

7.2 Attributes for the range coordinate variable

Attribute name	Type	Comments
standard_name	string	"projection_range_coordinate"
long_name	string	e.g. "range_to_measurement_volume" or "range_to_middle_of_each_range_gate"
units	string	"metres" or "meters"
spacing_is_constant	string	"true" or "false"
meters_to_center_of_first_gate	float or float(sweep)	Start range
meters_between_gates	float or float(sweep)	Gate spacing. Required if spacing_is_constant is "true".
axis	string	"radial_range_coordinate"

CHAPTER

EIGHT

LOCATION VARIABLES

Name	Data type	Dimension	Comments
latitude	double	none or (time)	Latitude of the instrument
longitude	double	none or (time)	Longitude of the instrument
altitude	double	none or (time)	Altitude of the instrument above the geoid (i.e. the orthometric height), using the WGS84 ellipsoid and EGM2008 geoid model[2]. For a scanning radar this is the altitude of the centre of rotation of the antenna.

[2] This definition is more specific than that given in the CfRadial-1.4 specification and aligns with that used in CfRadial-2.1.

CHAPTER

NINE

SWEEP VARIABLES

Sweep variables are always required, even if the volume only contains a single sweep.

Name	Data type	Dimension	Units	Comments
sweep_number	int	(sweep)		The number of the sweep in the volume scan, starting at 0.
sweep_mode	char	(sweep,string_length)		Options are "sector", "coplane", "rhi", "vertical_pointing", "idle", "azimuth_surveillance", "elevation_surveillance", "sunscan", "pointing", "manual_ppi", "manual_rhi"
fixed_angle	float	(sweep)	degree	Target angle for the sweep.
sweep_start_ray_index	int	(sweep)		Index of the first ray in sweep relative to the start of volume, 0-based.
sweep_end_ray_index	int	(sweep)		Index of the last ray in sweep relaitve to the start of the volume. 0-based.

MOMENTS FIELD DATA VARIABLES

Handling of moments field variables in NCAS Radar 1.0 follows that documented for CfRadial-1.4. Most commonly data from NCAS radars will have a fixed number of range gates per ray, and the field variables will be 2-dimensional arrays with the dimensions *time* and *range*. For the special case of variable numbers of gates per ray see CfRadial Version-1.4 documentation for more details.

The field data will be stored using one of the following:

Type	Byte width
byte	1
short	2
int	4
float	4
double	8

The netCDF variable name is interpreted as the short name for the field.

The following attributes are required for field variables:

Attribute name	Type	Convention	Description
long_name	string	CF	Long name describing the field.
standard_name or **proposed_standard_name**	string	CF	Proposed CF standard name for the field
units	string	CF	Units for the field
_FillValue	same type as field data	CF	Indicates data are missing at this range bin.
coordinates	string	CF	See note below

Use of coordinates attribute

The *"coordinates"* attribute lists the variables needed to compute the location of a data point in space. For stationary platforms it should be set to *"elevation azimuth range"*. For moving platforms it should be *"elevation azimuth range heading roll pitch rotation tilt"*

10.1 Quality control

In CfRadial-1.4 a field variable may make use of more than one reserved value to indicate a variety of conditions. For example, with radar data, you may wish to indicate that the beam is blocked for a given gate, and that no echo will ever be detected at that gate. That provides more information than just using *_FillValue*. The *flag_values* and *flag_meanings* attributes can be used in this case, which specifies the associated quality-control field variable.

Although CfRadial-1.4 allows the assignment of *flag_values* directly to a moment field, this is **not** the preferred approach in NCAS-Radar. Instead, quality control for a field variable is specified through one or more associated "quality control fields", which are specified by the *ancillary_variables* attribute.

Quality control fields may be constructed using sets of *flag_values* together with associated *flag_meanings*. For example, we might use a quality control field named *qc_flag* as follows:

```
ubyte qc_flag(time, range) ;
qc_flag:is_quality_field = "true" ;
qc_flag:qualified_variables = "dBZH vel" ;
qc_flag:long_name = "Quality control flag" ;
qc_flag:flag_values = 0UB, 1UB, 2UB, 4UB, 255UB ;
qc_flag:flag_meanings = "not_used good_data bad_data data_in_blind_range no_qc_performed
↪" ;
```

Note the use of the *is_quality_field* attribute to indicate that this is a quality control field. This is important as it defaults to "false" if not present.

A quality control field uses the attribute *qualified_variables* (in this example variables with the short names *dBZH* and *vel*) to specify (as a space delimited list) which field variables it qualifies.

Instead of a list of *flag_values*, we also have the option of specifying quality control using a flag_mask field. This is an integer-type field variable where each element is constructed using a bit-wise OR to combine conditions. In this case the *flag_masks* and *flag_meanings* attributes are used to indicate the valid values and meanings.

A given field variable may be associated with more than one quality control field. For example, in addition to a quality control flag we may have an associated quality control field to specify the uncertainty in the field variable. Such a field would be of the same type as the field variable it qualifies.

CHAPTER

ELEVEN

NAMING OF VARIABLES

11.1 Table of field variables and proposed standard names

The attribute `standard_name` should only be used where it has already been accepted as a standard name in the CF convention. Otherwise, the attribute `proposed_standard_name` shoud be used.

Standard name	Short name	Units	Aready in CF?
equivalent_reflectivity_factor	DBZ	dBZ	yes
linear_equivalent_reflectivity_factor	Z	Z	no
radial_velocity_of_scatterers_away_from_instrument	VEL	m s-1	yes
doppler_spectrum_width	WIDTH	m s-1	no
log_differential_reflectivity_hv	ZDR	dB	no
log_linear_depolarization_ratio_hv	LDR	dB	no
log_linear_depolarization_ratio_h	LDRH	dB	no
log_linear_depolarization_ratio_v	LDRV	dB	no
differential_phase_hv	PHIDP	degree	no
specific_differential_phase_hv	KDP	degree km-1	no
cross_polar_differential_phase	PHIHX	degree	no
cross_correlation_ratio_hv	RHOHV		no
co_to_cross_polar_correlation_ratio_h	RHOXH		no
co_to_cross_polar_correlation_ratio_v	RHOXV		no
log_power	DBM	dBm	no
log_power_co_polar_h	DBMHC	dBm	no
log_power_cross_polar_h	DBMHX	dBm	no
log_power_co_polar_v	DB-MVC	dBm	no
log_power_cross_polar_v	DB-MVX	dBm	no
linear_power	PWR	mW	no
linear_power_co_polar_h	PWRHC	mW	no
linear_power_cross_polar_h	PWRHX	mW	no
linear_power_co_polar_v	PWRVC	mW	no
linear_power_cross_polar_v	PWRVX	mW	no
signal_to_noise_ratio	SNR	dB	no
signal_to_noise_ratio_co_polar_h	SNRHC	dB	no
signal_to_noise_ratio_cross_polar_h	SNRHX	dB	no
signal_to_noise_ratio_co_polar_v	SNRVC	dB	no
signal_to_noise_ratio_cross_polar_v	SNRVX	dB	no

continues on next page

Table 1 – continued from previous page

Standard name	Short name	Units	Aready in CF?
normalized_coherent_power	NCP		no
corrected_equivalent_reflectivity_factor	DBZc	dBZ	no
corrected_radial_velocity_of_scatterers_away_from_instrument	VELc	m s-1	no
corrected_log_differential_reflectivity_hv	ZDRc	dB	no
radar_estimated_rain_rate	RRR	mm h-1	no
rain_rate	RR	kg m-2 s-1	yes
radar_echo_classification	REC	legend	no

11.2 Table of metadata variables with strict names and suggested long names

Variable name	Long name	Units
altitude_agl	altitude_above_ground_level	meters or metres
altitude_correction	altitude_correction	meters or metres
altitude	altitude	meters or metres
antenna_transition	antenna_is_in_transition_between_sweeps	
azimuth_correction	azimuth_angle_correction	degrees
azimuth	ray_azimuth_angle	degrees
drift_correction	platform_drift_angle_correction	degrees
drift	platform_drift_angle	degrees
eastward_velocity_correction	platform_eastward_velocity_correction	m s-1
eastward_velocity	platform_eastward_velocity	m s-1
eastward_wind	eastward_wind	m s-1
elevation_correction	ray_elevation_angle_correction	degrees
time_coverage_end	data_volume_end_time_utc	seconds
fixed_angle	target_fixed_angle	degrees
follow_mode	follow_mode_for_scan_strategy	
frequency	radiation_frequency	s-1
heading_change_rate	platform_heading_angle_rate_of_change	degrees
heading_correction	platform_heading_angle_correction	degrees
heading	platform_heading_angle	degrees
instrument_name	name_of_instrument	
instrument_type	type_of_instrument	
latitude_correction	latitude_correction	degrees
latitude	latitude	degrees_north
longitude	longitude	degrees_east
northward_velocity_correction	platform_northward_velocity_correction	m s-1
northward_velocity	platform_northward_velocity	m s-1
northward_wind	northward_wind	m s-1
nyquist_velocity	unambiguous_doppler_velocity	m s-1
n_samples	number_of_samples_used_to_compute_moments	
pitch_change_rate	platform_pitch_angle_rate_of_change	degree s-1
pitch_correction	platform_pitch_angle_correction	degrees
pitch	platform_pitch_angle	degrees
platform_is_mobile	platform_is_mobile	

continues on next page

Table 2 – continued from previous page

Variable name	Long name	Units
platform_type	platform_type	
polarization_mode	transmit_receive_polarization_mode	
prt_mode	transmit_pulse_mode	
pressure_altitude_correction	pressure_altitude_correction	meters or metres
primary_axis	primary_axis_of_rotation	
prt	pulse_repetition_time	seconds
prt_ratio	multiple_pulse_repetition_frequency_ratio	
pulse_width	transmitter_pulse_width	seconds
radar_antenna_gain_h	nominal_radar_antenna_gain_h_channel	dB
radar_antenna_gain_v	nominal_radar_antenna_gain_v_channel	dB
radar_beam_width_h	half_power_radar_beam_width_h_channel	degrees
radar_beam_width_v	half_power_radar_beam_width_v_channel	degrees
radar_receiver_bandwidth	radar_receiver_bandwidth	s-1
radar_measured_transmit_power_h	radar_measured_transmit_power_h_channel	dBm
radar_measured_transmit_power_v	radar_measured_transmit_power_v_channel	dBm
range_correction	range_to_center_of_measurement_volume_correction	meters or metres
range	projection_range_coordinate	meters or metres
roll_correction	platform_roll_angle_correction	degrees
roll	platform_roll_angle	degrees
rotation_correction	ray_rotation_angle_relative_to_platform_correction	degrees
rotation	ray_rotation_angle_relative_to_platform	degrees
r_calib_antenna_gain_h	calibrated_radar_antenna_gain_h_channel	dB
r_calib_antenna_gain_v	calibrated_radar_antenna_gain_v_channel	dB
r_calib_base_dbz_1km_hc	radar_reflectivity_at_1km_at_zero_snr_h_co_polar_channel	dBZ
r_calib_base_dbz_1km_hx	radar_reflectivity_at_1km_at_zero_snr_h_cross_polar_channel	dBZ
r_calib_base_dbz_1km_vc	radar_reflectivity_at_1km_at_zero_snr_v_co_polar_channel	dBZ
r_calib_base_dbz_1km_vx	radar_reflectivity_at_1km_at_zero_snr_v_cross_polar_channel	dBZ
r_calib_coupler_forward_loss_h	radar_calibration_coupler_forward_loss_h_channel	dB
r_calib_coupler_forward_loss_v	radar_calibration_coupler_forward_loss_v_channel	dB
r_calib_index	calibration_data_array_index_per_ray	
r_calib_ldr_correction_h	calibrated_radar_ldr_correction_h_channel	dB
r_calib_ldr_correction_v	calibrated_radar_ldr_correction_v_channel	dB
r_calib_noise_hc	calibrated_radar_receiver_noise_h_co_polar_channel	dBm
r_calib_noise_hx	calibrated_radar_receiver_noise_h_cross_polar_channel	dBm
r_calib_noise_vc	calibrated_radar_receiver_noise_v_co_polar_channel	dBm
r_calib_noise_vx	calibrated_radar_receiver_noise_v_cross_polar_channel	dBm
r_calib_noise_source_power_h	radar_calibration_noise_source_power_h_channel	dBm
r_calib_noise_source_power_v	radar_calibration_noise_source_power_v_channel	dBm
r_calib_power_measure_loss_h	radar_calibration_power_measurement_loss_h_channel	dB
r_calib_power_measure_loss_v	radar_calibration_power_measurement_loss_v_channel	dB
r_calib_pulse_width	radar_calibration_pulse_width	seconds
r_calib_radar_constant_h	calibrated_radar_constant_h_channel	(m mW-1)dB
r_calib_radar_constant_v	calibrated_radar_constant_v_channel	(m mW-1)dB
r_calib_receiver_gain_hc	calibrated_radar_receiver_gain_h_co_polar_channel	dB
r_calib_receiver_gain_hx	calibrated_radar_receiver_gain_h_cross_polar_channel	dB
r_calib_receiver_gain_vc	calibrated_radar_receiver_gain_v_co_polar_channel	dB
r_calib_receiver_gain_vx	calibrated_radar_receiver_gain_v_cross_polar_channel	dB
r_calib_receiver_mismatch_loss	radar_calibration_receiver_mismatch_loss	dB
r_calib_receiver_slope_hc	calibrated_radar_receiver_slope_h_co_polar_channel	

continues on next page

Table 2 – continued from previous page

Variable name	Long name	Units
r_calib_receiver_slope_hx	calibrated_radar_receiver_slope_h_cross_polar_channel	
r_calib_receiver_slope_vc	calibrated_radar_receiver_slope_v_co_polar_channel	
r_calib_receiver_slope_vx	calibrated_radar_receiver_slope_v_cross_polar_channel	
r_calib_sun_power_hc	calibrated_radar_sun_power_h_co_polar_channel	dBm
r_calib_sun_power_hx	calibrated_radar_sun_power_h_cross_polar_channel	dBm
r_calib_sun_power_vc	calibrated_radar_sun_power_v_co_polar_channel	dBm
r_calib_sun_power_vx	calibrated_radar_sun_power_v_cross_polar_channel	dBm
r_calib_system_phidp	calibrated_radar_system_phidp	degrees
r_calib_test_power_h	radar_calibration_test_power_h_channel	dBm
r_calib_test_power_v	radar_calibration_test_power_v_channel	dBm
r_calib_time	radar_calibration_time_utc	
r_calib_two_way_radome_loss_h	radar_calibration_two_way_radome_loss_h_channel	dB
r_calib_two_way_radome_loss_v	radar_calibration_two_way_radome_loss_v_channel	dB
r_calib_two_way_waveguide_loss_h	radar_calibration_two_way_waveguide_loss_h_channel	dB
r_calib_two_way_waveguide_loss_v	radar_calibration_two_way_waveguide_loss_v_channel	dB
r_calib_xmit_power_h	calibrated_radar_xmit_power_h_channel	dBm
r_calib_xmit_power_v	calibrated_radar_xmit_power_v_channel	dBm
r_calib_zdr_correction	calibrated_radar_zdr_correction	dB
scan_name	name_of_antenna_scan_strategy	
scan_rate	antenna_angle_scan_rate	degree s-1
site_name	name_of_instrument_site	
spacing_is_constant	spacing_between_range_gates_is_constant	
sweep_end_ray_index	index_of_last_ray_in_sweep	
sweep_mode	scan_mode_for_sweep	
sweep_number	sweep_index_number_0_based	
sweep_start_ray_index	index_of_first_ray_in_sweep	
sweep_unambiguous_range	unambiguous_range_for_sweep	meters or metres
tilt_correction	ray_tilt_angle_relative_to_platform_correction	degrees
tilt	ray_tilt_angle_relative_to_platform	degrees
time	time	seconds
time_coverage_start	data_volume_start_time_utc	
unambiguous_range	unambiguous_range	meters or metres
vertical_velocity_correction	platform_vertical_velocity_correction	m s-1
vertical_velocity	platform_vertical_velocity	m s-1
vertical_wind	upward_air_velocity	m s-1
volume_number	data_volume_index_number	

CHAPTER

TWELVE

REFERENCES

BIBLIOGRAPHY

[1] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A.C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, mar 2016. doi:10.1038/sdata.2016.18.

[2] Pavlos Kollias, Nitin Bharadwaj, Kevin Widener, Ieng Jo, and Karen Johnson. Scanning ARM cloud radars. part i: operational sampling strategies. *Journal of Atmospheric and Oceanic Technology*, 31(3):569–582, mar 2014. doi:10.1175/JTECH-D-13-00044.1.