Community Oriented Research Exchange for Atmospheric Composition (CORE-AC) Requirements Document

Beth, Sean, Megan, Crystal, Amy Jo, Brian, Michael, Morgan, Ali, Matt, Gao

Version: Jan 2026

Status: Draft

Table of contents

1 Introduction

1.1 Objectives

The overarching goal of the Community Oriented Research Exchange for Atmospheric Composition (CORE-AC) is to document a set of requirements that will increase the Findability Accessibility Interoperability and Reusability (FAIRness) of NASA atmospheric composition focused field campaign data products. More specifically, this document describes metadata and formatting requirements tailored to the atmospheric composition research community that will facilitate compliance of CORE-AC-compliant data with Climate and Forecast conventions, in order to improve the usability of NASA atmospheric composition field campaign data products by the Earth science modeling community and the Earth science research community at large.

1.2 Background and Context

There are currently two predominant data format standards for suborbital, atmospheric composition data. The **ICARTT format** developed by the International Consortium for Atmospheric Research on Transport and Transformation (ICARTT) field campaign science team is used primarily for reporting data acquired through in situ measurements. Remotely sensed data, on both satellite and suborbital platforms are typically reported in Hierarchical Data Format **HDF5** or Network Common Data Form **NetCDF4**. Both standards have limitations that the **Community Oriented Research Exchange for Atmosopheric Composition (CORE-AC)** requirements seek to address.

Prior to 2010, when the ICARTT format was adopted as a NASA standard, there was no standardized format for atmospheric composition measurements taken by field campaigns. ICARTT was considered easy to use and included metadata that aided in data discovery. However, multi-dimensional arrays, which are needed for reporting aerosol number size distributions, or vertical profiles of aerosol backscatter, for example) are unsupported in ICARTT. Metadata requirements lack adequate guidelines or shared semantics and, as a result, data integration generally is accomplished only by writing custom code. Additionally, ICARTT is ASCII-based, meaning that large volumes of data yield exceedingly large files that are difficult to store. Concerns about cloud storage costs and data accessibility indicate that the atmospheric composition research community would be better served by adoption of the HDF and NetCDF formats.

As already noted, HDF and NetCDF are the accepted standard for scientists working with remote sensing instruments in both satellite missions and airborne field campaigns. However, use of these formats has not resulted in broadly interoperable data products. The structure of dimensions, variables, and attributes can vary substantially from one data product to another, as can the quality and content of metadata. The lack of any guidelines for how to design and annotate these file components creates significant difficulties in almost every stage of the data product lifecycle, from data ingestion and publication, to end-user analysis. In particular, files are often incompatible with common tools and libraries because, lacking adequate and consistent metadata, they are not truly self-describing. This is

especially frustrating to the modeling community, which relies on integrated data from multiple instruments and campaigns to assess and improve Earth System Models (ESMs).

To address these challenges, **Measurements of Aerosols, Clouds and their Interactions for ESMs (MACIE)**, an interagency working group, was established in 2021. Co-led by **Dr. Jeffrey Reid (NRL)** and **Dr. Jeffrey Stehr (DOE)**, with members from NASA, NOAA, NSF NCAR, and university partners, one of MACIE's primary tasks was to make field campaign data more usable for the modeling community. After reviewing existing HDF/NetCDF profiles, including Climate and Forecast (CF) conventions and Generic Earth Observation Metadata Standard (GEOMS), the MACIE group determined that a new set of requirements was needed that would be applicable to both in-situ and remote sensing instruments, guided by modeling needs, and sensitive to the operational nature of diverse instruments. This effort led to the development of the Community Oriented Research Exchange for Atmospheric Composition (CORE-AC) Requirements.

The CORE-AC requirements outlined in this document represent *basic* requirements for HDF and/or netCDF files for the atmospheric composition research community. Additional global, group, and variable attributes may be needed to make specific datasets truly self-describing, and CORE-AC requirements may not address specific needs of other scientific disciplines. CORE-AC is compatible with the GEOMS approach, but with two key distinctions: CORE-AC is **capable of handling data from multiple instruments in a single file** and is **compatible with CF-aware tools**.

CORE-AC incorporates relevant global and variable attributes from CF conventions for defining coordinates, units, and variable relationships; Attribute Convention for Data Discovery (ACDD) recommendations; and established GEOMS guidelines, as well as adding some requirements specific to atmospheric composition data. The need to combine and extend requirements in this way reflects the reality that generic standards like CF and ACDD are not sufficient by themselves to make data truly findable and (re)usable for research. Non-CF and non-ACDD attributes are needed to describe field campaigns, instruments, and data variables sufficiently to enable researchers to find all of the relevant data (recall), and to quickly filter them to only those that are most suitable (precision).

CORE-AC mandates use of Atmospheric Composition Variable Standard Names Convention (ACVSNC) for naming variables, a convention that has been in use for several years within the NASA atmospheric composition research community. ACVSNC names provide significantly more detailed descriptions of variables than CF standard names. ACVSNC names, for example, include information about the measurement object, the property that was measured, and the conditions under which the measurement was made. The need to include this type of information was determined by documenting use cases derived from descriptions by atmospheric composition researchers of their actual data needs and research processes.

CORE-AC requirements have been tested and fine-tuned through feedback from multiple field campaigns. They have been shown to be effective in supporting data discovery and other common use cases such as real-time production of data merge products, for both remote sensing and in situ data products. Moreover, ACVSN names have been generally accepted by instrument and modeling teams. This document outlines a set of requirements that are designed to:

~~1.2 HDF5 vs. NetCDF4~~

~~HDF5 acts as the underlying storage container, while NetCDF4 defines the data model and interface implemented on top of it. Although HDF5 provides the physical file structure capable of holding diverse data types, NetCDF4 enforces the specific rules—such as Dimensions, Coordinates, and Variables—required for scientific software interpretation.~~

2 File Structure

2.1 Format

All files must be valid HDF5 (.hdf) or NetCDF4 (.nc) files that strictly adhere to the NetCDF4 data model constraints outlined in this document, to ensure compatibility with standard analysis tools such as Panoply, IDL, Python (xarray), and Matlab.

2.2 Data Model

Root: All **Data Product** variables and **Coordinate** variables may reside under the root.
Subgroups: May be used to organize intermediate variables or ancillary data.
Variable Types: All variables must be classified into one of the following types via the VarType attribute:

- Dimension
- Coordinate
- Data Product
- Flag
- Other

---

3 Global Attributes

These attributes describe the dataset as a whole and are useful for data discovery.

Table 2.1 Global Attributes

Attribute	Requirement	Description / Rationale
ACVSNC_standard_name_URL	Mandatory	URL linking to the Standard Name description, for example: "https://www-air.larc.nasa.gov/missions/etc/AtmosphericCompositionVariableStandardNames.pdf
ACVSNC_standard_name_version	Mandatory	Version of the Standard Name table used.
Conventions	Mandatory	Conventions followed, including version. For example, CF version 1.9, typically represented as "CF-1.9".
format	Mandatory	Format of the files that make up the data product, such as

Attribute	Requirement	Description / Rationale
		"HDF5" or NetCDF4
history	Mandatory	Description of data revision history.
IdentifierProductDOI	Conditional	Data DOI (required for publication quality data).
institution	Mandatory	PI affiliation. Comma-separated if multiple.
keywords	Mandatory	High-level description using GCMD Keywords.
PI_contact	Mandatory	Email of Principal Investigator.
PI_name	Mandatory	Name of Principal Investigator.
ProcessingLevel	Mandatory	Data processing level (e.g., L1C, L2).
project	Mandatory	Campaign or Mission acronym (e.g., MOOSE).
references	Optional	Citation of journal publication or readme filename.
source	Mandatory	Source of data (e.g., instrument acronym, model).
summary	Optional	Brief description of file content (project, platform, instrument).
title	Mandatory	One-line description of the data product.
VersionID	Mandatory	Revision number (must match revision in filename, e.g., R0, R1).
data_processing_date	Optional	Date when data was processed (YYYY-MM-DD HH:MM:SS).
data_processing_note	Optional	Identifier for algorithm/software or assumptions.
data_product_groups	Mandatory	Subgroups holding data variables (blank if all data variables in root).
data_use_guideline	Mandatory	Instructions on citation and co-authorship.
file_originator	Mandatory	Name of person who created/submitted the file.
file_originator_contact	Mandatory	Email of the file originator.

Attribute	Requirement	Description / Rationale
flight_number_day	Conditional	Sequence number for multiple flights in a day (L1, L2).
flight_start_date	Mandatory	UTC date of takeoff (YYYYMMDD).
geospatial_lat_max	Optional	Max sampling latitude (degrees north).
geospatial_lat_min	Optional	Min sampling latitude (degrees north).
geospatial_lon_max	Optional	Max sampling longitude (degrees east).
geospatial_lon_min	Optional	Min sampling longitude (degrees east).
geospatial_vertical_max	Optional	Max vertical extent (with unit).
geospatial_vertical_min	Optional	Min vertical extent (with unit).
last_modified_date	Mandatory	Date of last modification (YYYY-MM-DD).
License	Conditional	data license if not CC0
measurement_platform	Mandatory	Description of platform (include location for ground sites).
platform_identifier	Mandatory	Platform ID (must match filename; e.g., tail number).
platform_type	Optional	WMO Facility Type (e.g., AirMobile).
source_description	Optional	Brief description of measurement/instrument with URL.
time_coverage_end	Mandatory	UTC end time (YYYY-MM-DD HH:MM:SS).
time_coverage_resolution	Optional	Data reporting frequency (e.g., 1 s or irregular).
time_coverage_start	Mandatory	UTC start time (YYYY-MM-DD HH:MM:SS).
unit_convention	Optional	http://codes.wmo.int/wmdr/unit

2. Global Attribute Text Descriptions (Sorted)

- ~~**ACVSNC_standard_name_URL (Mandatory):** URL linking to the Standard Name description, for example: "https://www-air.larc.nasa.gov/missions/etc/AtmosphericCompositionVariableStandardNames.pdf~~".

- ~~**ACVSNC_standard_name_version (Mandatory):** Version of the Standard Name used.~~
- ~~**Conventions (Mandatory):** conventions followed, including CF convention version, typically represented as "CF-1.9".~~
- ~~**Format (Mandatory):** format the file was created in, such as "HDF5" or "netCDF4…".~~
- **History (Mandatory):** A brief description of the data revision history associated with the revision identifier, similar to ICARTT RevisionNotes.
- **IdentifierProductDOI (Conditional):** data DOI, which is required for publication-quality data.
- **Institution (Mandatory):** PI affiliation, such as "NASA Langley Research Center"; please use a comma to separate affiliations if there is more than one and list them in the same order as the Provider list.
- **Keywords (Mandatory):** high-level description of the data products using the best match GCMD keywords (e.g., "NO2 and CH2O column densities").
- **PI_contact (Mandatory):** PI contact e-mail; please use a comma to separate emails if there is more than one.
- **PI_name (Mandatory):** PI name formatted as "first name last name"; please use a comma to separate names if there is more than one.
- **ProcessingLevel (Mandatory):** data processing level (e.g., "L1C", "L2"); if the file contains more than one product level, use a comma to separate them.
- **Project (Mandatory):** project acronym, for example, "MOOSE".
- **References (Optional):** description of the data, such as a citation of a journal publication or a readme filename (similar to ICARTT DATA_INFO).
- **Source (Mandatory):** source of the data, such as an instrument acronym, name, or model (e.g., "RSP").
- **Summary (Optional):** A brief description of the file including project, platform, instrument, and data product identifier (e.g., "GCAS NO2 column measurements for 20220611 flight").
- **Title (Mandatory):** Title of the file providing a one-line description of the data product, such as "MOOSE GCAS Measurements".
- **VersionID (Mandatory):** same as the revision number from ICARTT (e.g., "RA", "RB", "RC" for preliminary or “field” data or "R0", "R1", "R2" for publication-quality or “final” data); this must match the identifier in the filename.
- **data_processing_date (Optional):** date indicating when the data was processed, formatted as YYYY-MM-DD HH:MM:SS.
- **data_processing_note (Optional):** identifier for any data processing algorithm, software, or major assumptions utilized.
- **data_product_groups (Mandatory):** names of subgroups that hold the data product variables, separated by a comma (e.g., "Observation_Data"); leave this blank if all data are located under the root.
- **data_use_guideline (Mandatory):** Equivalent to "STIPULATIONS_ON _USE" in ICARTT. It outlines guidelines for responsible scientific use, for instance: "For responsible scientific use of the data

sets provided, data users are strongly encouraged to carefully study the file headers and directly consult with the instrument PIs. Please acknowledge the data source and offer co-authorship to relevant instrument PIs when appropriate".

- **file_originator (Mandatory):** Person who created and submitted the file, such as "John Smith"; this can be the same as the Principal Investigator (PI).
- **file_originator_contact (Mandatory):** email address for the person who created and submitted the file, which can be the same as the PI.
- **flight_number_day (Conditional):** sequence flight number of the day, used to distinguish multiple flights within a single day, such as "L1" or "L2".
- **flight_start_date (Mandatory):** UTC date when the flight took off, formatted in YYYYMMDD (e.g., "20220601").
- **geospatial_lat_max (Optional):** maximum sampling latitude in degrees north, with the value and unit separated by a space.
- **geospatial_lat_min (Optional):** minimum sampling latitude in degrees north, with the value and unit separated by a space.
- **geospatial_lon_max (Optional):** maximum sampling longitude in degrees east, with the value and unit separated by a space.
- **geospatial_lon_min (Optional):** minimum sampling longitude in degrees east, with the value and unit separated by a space.
- **geospatial_vertical_max (Optional):** maximum sampling altitude (e.g., "5 km"), specifying MSL or AGL where applicable, with the value and unit separated by a space; note that this is not applicable for column measurements.
- **geospatial_vertical_min (Optional):** minimum sampling altitude (e.g., "0.3 km"), specifying MSL or AGL where applicable, with the value and unit separated by a space; note that this is not applicable for column measurements.
- **last_modified_date (Mandatory):** Equivalent to the data revision or processing date, formatted in YYYYMMDD (e.g., "2022-03-01").
- **license (Conditional):** data license if other than CC0.
- **measurement_platform (Mandatory):** Description of the platform, including the location for ground sites (e.g., "NASA King Air N528NA").
- **platform_identifier (Mandatory):** Platform identifier, such as "KingAir"; this must match the one in the filename and should use the tail number or PID when available.
- **platform_type (Mandatory):** Keywords to increase interoperability, such as "AirMobile".
- **source_description (Optional):** a brief description of the measurement and/or instrument, including a URL and last modified date if applicable (e.g., "https://data.giss.nasa.gov/rsp_air/").
- **time_coverage_end (Mandatory):** UTC data end date and time, formatted as YYYY-MM-DD HH:MM:SS.
- **time_coverage_resolution (Optional):** Data reporting frequency, for instance "1 s", "10 s", or "irregular" if not reported in a regular interval.

- **time_coverage_start (Mandatory):** UTC data start date and time, formatted as YYYY-MM-DD HH:MM:SS.
- **unit_convention (Optional):** Unit convention; the use of WMO WIGOS codes for measurement units is recommended (e.g., "http://codes.wmo.int/wmdr/unit").

3. Variable Attributes & Dimensionality

3.1 Coordinate Variables

Coordinate variables map the data to physical space and time.

- **time (Mandatory):**
  - **Units:** seconds since YYYY-MM-DD HH:MM:SS.
  - **Long Name:** Must specify the **Time Standard** (UTC) and **Anchor Point** (Start/Mid/End/Instant).
    - *Example:* UTC mid time of sampling interval
- **lat / lon (Mandatory):**
  - **Units:** degrees_north / degrees_east.
- **altitude (Mandatory for Airborne/Profile):**
  - **Units:** WMO standard (e.g., m, km).
  - **Long Name:** Must specify reference (MSL vs AGL).

3.2 Data Product Variables (1D vs 2D)

One dimentional data variables (x) store the scientific measurements. Their dimensionality depends on the measurement technique.

Case A: 1D Variables (Time Series / Point Data)

Used for in-situ measurements where there is exactly **one value per timestamp**.

- **Structure:** double x(time) or double x(along_track)
- **Description:** The variable is a simple array aligned one-to-one with the primary dimension.

Case B: 2D Variables (Swaths or Profiles)

Used when the instrument captures multiple values per timestamp (e.g., a vertical profile or a horizontal scan).

- **Structure:** The variable depends on the primary dimension **AND** a secondary dimension.
  - *Examples:* double x(time, altbin) or double x(along_track, cross_track).
- **Requirement:** The secondary dimension (altbin, cross_track, etc.) must be defined as a dimension variable in the file.

3.3 Common Attributes & Vocabulary

The official CF Standard Name table covers less than 30% of the specific atmospheric composition species measured in field campaigns. Therefore, strict adherence to CF names would result in generic,

unusable metadata. Furthermore, the ACVSNC standard names provide more accurate variable descriptions.

The following attributes are required to ensure data is properly described:

Attribute	Requirement	Explanation
long_name	Mandatory	More detailed description and may be used for plot labels.
units	Mandatory	WIGOS unit code, adopted as a NASA standard.
_FillValue	Conditional	If used, value for missing data (e.g., -9999.0 or NaN).
coordinates	Optional	List of coordinate variables (e.g., time lat lon).
ACVSNC_standard_name	Mandatory	A standardized identifier used as a tag for data discovery and (re)usability. Reference: ACVSNC Variable Table.
standard_name	Optional	CF standard name If a matching can be found
bounds	Conditional	Defines the intervals for coordinate variables (e.g., start/stop time, swath footprint). Required if the measurement represents an interval.

3.4 Flag Variables

- **Variable Name:** Must contain the string "flag" in the variable name.
- **Attributes:**
  - flag_values: Comma-separated integers (e.g., 0, 1, 2).
  - flag_meanings: Space-separated descriptive words or phrases for each flag value (e.g., good suspect bad). **Use underscores** for multi-word phrases.
  - flag_masks: A list of values defining bit field masks that are bitwise ANDed with the data variable to isolate and identify independent boolean conditions or status flags.

---

4. Defined Data Types

The following data types are **strictly required** to ensure consistency across products.

Type Name	Bit Depth	Usage Definition
double	64-bit float	**Required** for all **Data Product** variables and **Coordinate** variables (time, lat, lon). Provides high precision for physical quantities.
int	32-bit integer	**Required** for standard **Flag** variables and discrete counts.
long	64-bit integer	**Reserved** for **Flag** variables that require large bitmasks (values exceeding 32-bit limits).

5. Compliance Checker

For a standard to be viable, it requires adequate maintenance and implementation support. A key component of this support is a dedicated tool that allows users to verify their degree of compliance—an approach that proved critical to the success of the ICARTT format standards. Developed in parallel with the profile revisions, this checker provides users with immediate feedback on standard adherence and pinpoints specific errors.

The tool features two modes of operation:

1. **Generic Mode:** Designed for HDF/netCDF files, this mode validates the completeness of global and variable attributes. It also verifies that data and time variables possess the correct dimensionality.
2. **Instrument-Specific Mode:** Like the GEOMS file checker, this mode performs a rigorous validation using a configuration file developed in consultation with instrument scientists. It checks attribute content and variable names against pre-determined values defined in the configuration.

Further technical details regarding the checker are provided in **Appendix A**.

Appendix A: Compliance Checker Technical Description

A.1 Overview

The Compliance Checker is a Python-based utility designed to validate HDF5/NetCDF files against the Requirements Profile outlined in this document. The tool is built using the h5py library rather than the standard netCDF4 library. This design choice was made to maximize robustness; h5py is more tolerant of formatting errors, allowing the tool to open and diagnose files that might otherwise cause standard NetCDF readers to fail immediately.

A.2 Objectives

The tool serves two primary functions:

1. **Verification:** Ensures files adhere to the mandatory global and variable attributes defined in the Profile.
2. **Diagnostics:** Provides specific error messages and warnings to guide data producers in correcting non-compliant files.

A.3 Modes of Operation

The checker operates in two distinct modes, determined by the configuration provided at runtime.

1. Instrument-Specific Mode

This mode is triggered when a specific **Table of Attributes and Variables (TAV)** configuration file is configured within the Checker and the user running the checker specifies that their file should be validated against that configuration file. The Checkker performs a rigorous validation of the file content against strict, pre-determined values.

- **Attribute Content:** Verifies that specific attributes (e.g., source, platform_identifier) match the exact values defined in the TAV.
- **Variable Names:** Confirms that all variables listed in the configuration file are present in the data file.
- **Flag Validation:** Checks for the presence and length of flag_values and flag_meanings for status variables.

2. Generic Mode

This mode applies general consistency checks for HDF5/NetCDF files without requiring a strict instrument-specific configuration.

- **Global Completeness:** Checks for the presence of mandatory global attributes (e.g., PI_name, data_use_guideline) utilizing a generic TAV.
- **Time Dimension:** Verifies that the time variable exists, is a dimension scale, and uses correct units (must start with "seconds since").
- **Dimensionality:** Scans all variables to ensure they are properly attached to dimension scales (e.g., time, altitude).

A.4 Key Validation Logic

A.4.1 Global Attribute Validation

The tool iterates through required global attributes, applying different checks based on the attribute type:

- **Text Check:** Ensures attributes are not blank and contain required keywords.
- **Lat/Lon Check:** Verifies that geospatial bounds (e.g., geospatial_lat_min) contain numeric values separated from units by a space (e.g., -45.0 degrees_north).
- **Group Check:** Confirms the existence of required HDF5 groups (subfolders) defined in the data_product_groups attribute.

A.4.2 Variable and Dimension Validation

- **Dimension Scales:** The tool verifies that every data variable is attached to a Dimension Scale. This is critical for distinguishing "Data Product" variables from "Coordinate" variables.
- **Standard Names:** Enforces the use of ACVSNC_standard_name for research-grade identification.
- **Time Formatting:**
    - **Units:** Must contain "seconds since".
    - **Long Name:** Must contain a valid anchor point keyword (start, stop, mid, or instant) to describe the sampling interval.
- **NaN & Fill Values:** If a variable contains NaN values, the tool enforces the presence of a _FillValue attribute.

A.5 Dependencies

The code requires the following Python libraries:

- h5py (HDF5 file I/O)

- numpy (Numerical operations)
- datetime (Date string validation)
- time
- json
- re
- os
- sys
- unicodedata
- pathlib

A.6 Error Reporting

The tool outputs diagnostic messages directly to the console:

- **Error:** Indicates a critical failure where the file violates a mandatory requirement (e.g., "Error: no dimension scale attached").
- **Warning:** file violates an optional or conditional requirement; but does not strictly violate the file structure. This Indicates a potential issue that may affect usability (e.g., "Warning: long_name description may be insufficient").