NDACC DATA VERSIONING SYSTEM

Version: 2.3, February 19, 2020

Authors:

I. Boyd, UMass
A.-M. Fjaeraa, NILU
J. Hannigan, NCAR
B. Langerock, BIRA
T. Leblanc, NASA-JPL
E. Mahieu, ULC
G. Nedoluha NRL
A. Niemeijer, STCORP
I. Petropavlovskikh, NOAA-ESRL
R. Querel, NIWA
A. Thompson, NASA-GSFC
J. Wild, UMD-ESSIC, NOAA/NCEP

Doc Changes:

April 13, 2019, v1.0: Initial draft

April 14, 2019, v1.1: Revised to expand on specific syntax for algorithm version number

August 30, 2019, v1.2: Minor revisions before release to the NDACC SC

January 10, 2020, v1.3: Minor revisions before final discussion of version 1

January 29, 2020, v2.0: Major revisions following discussions of Jan 27, 2020 telecon

January 30, 2020, v2.1: Minor corrections to wording

February 13, 2020, v2.2: Rewording of Section 3 paragraph 3, and new Section 3 paragraph 11

February 19, 2020, v2.3: Minor changes in the wording

1. Introduction:

This document describes the data versioning system designed to archive multiple NDACC data versions in HDF format using GEOMS-compliant standards. It does not address the archiving of NDACC data files in Ames format.

The purpose of the NDACC data versioning system is to provide a framework that allows the simultaneous archival of multiple data versions, data originating from a single NDACC instrument, but processed differently.

Different data processing is typically needed when 1) a newly re-processed version is made available by the NDACC PI, 2) it is deemed suitable and valuable to provide one or more centrally-processed and/or in-house-processed data products, and 3) it is deemed suitable and valuable to provide data products obtained using different data processing parameters (e.g., different time averaging).

This data versioning system is designed so that multiple versions are readily available to the data user, yet minimizing the confusion that could arise from having to choose among multiple products or datasets.

2. Previous NDACC Data Versioning System:

The NDACC data files in HDF format follow the GEOMS standards. These standards prescribe the inclusion of several global attributes, whose values are automatically passed into the filename. It is the case for the DATA_SOURCE, DATA_LOCATION, and DATA_FILE_VERSION attributes.

The previous GEOMS-compliant NDACC data versioning system was provided solely by the DATA_FILE_VERSION global attribute. This attribute is not directly tied to the algorithm, and had to be incremented with any change in the data in the file, including changes in algorithm resulting in a full reprocessing of the dataset, an archival of NDACC-certified data to replace Rapid Delivery (RD) data, or simply an error in a single or small number of files that had to be corrected. Compliant with GEOMS standards, the latest data file had to have the higher version number, but a complete dataset could have different version numbers in its constituent files.

Example:

groundbased_lidar.o3_nasa.jpl003_table.mountain.ca_20170522t031737z_20170522t051742z_001.hdf

The above HDF file has a DATA_VERSION attribute filled with a value of 001. Any file replacing it had to have a version number greater than 001.

This versioning system, which relied on a single field, has been replaced by a more sophisticated system, as described below.

3. New NDACC Data Versioning System:

The NDACC data versioning system now makes use of the GEOMS-compliant DATA_SOURCE attribute, which is transferred to the HDF file name just before the instrument location.

By default, this attribute consists of two fields separated by underscore and describing the instrument type with species, plus the instrument identifier.

Examples:

lidar.o3_nasa.jpl003 (taken from the full file name used in the previous section)

ftir.ch4_ncar001

mwr.o3_umass002

In the following, we describe how the DATA_SOURCE global attribute can be used to take advantage of the multi-version functionality of the new NDACC Data Versioning System.

1. To delineate datasets from a single instrument, but having undergone different processing, an optional third field can be added to the DATA_SOURCE attribute. The addition of the new field is optional, i.e., fully backward-compatible with all files archived following the previous versioning system. No renaming of currently archived datasets is required

2. The added field can contain one or more strings of letters and numbers (A-Z, 0-9), optionally separated by dots “.” Although not mandatory, it is recommended to include one or more numbers to accommodate a sense of data processing version number.

3. The concatenation of all numbers and dots whose purpose is to express a data processing version number is referred to as a “data processing version number”. The concatenation of all characters (letters and numbers) that are not part of a data processing version number is referred to as a “data processing keyword”. Multiple data processing keywords may be concatenated using dots. The order in which some of these keywords are concatenated may be constrained. The data processing keywords and the data processing version number are both optional items. DATA_SOURCE can be extended by one, neither or both items. If both items are present, the data processing version number should always be placed after the last data

processing keyword, with a dot separator between them, and no letters are allowed after that separator.

4. Data processing keywords are free-form and may have different purposes. Some may refer to the name of a processor, others to a type of processing. In all cases, each data processing keyword is uniquely defined and all subsequent usage of this keyword must strictly adhere to its definition.

5. The introduction of a new data processing keyword (and if applicable, its associated data processing version number) is managed at the NDACC Instrument Working Group level. A new keyword may be proposed individually by NDACC PIs or may be proposed collectively by one or several Instrument Working Group. In all cases, a newly-proposed keyword must be approved by the NDACC Instrument Working Groups.

6. Before an Instrument Working Group approves a new keyword, all the other NDACC Instrument Working Groups must be consulted to ensure that the use of this new keyword is suitable, and strictly adheres to its definition, across the network.

7. After approval by all NDACC Instrument Working Groups and prior to data archival, the new keyword must be submitted to the NDACC Data Handling Facility (DHF) Manager for final approval.

8. Newly-accepted keywords must be added, with its purpose fully described, to the "List of Approved NDACC Data Versioning System Keywords", stored and maintained at two separate locations: a metadata document centrally accessible on the NDACC DHF, and a metadata document centrally accessible on the NDACC Instrument Working Group websites. This list is also included in Appendix A-1 of this document.

9. Each keyword (and associated version number) used by an NDACC PI must be fully deciphered in the instrument Meta data file associated with the PI's dataset. No data file using a new keyword shall be archived until the new keyword is defined in the PI's metadata file

10. Although there are no technical limitations in the drafting of new keywords, it is recommended to choose keywords that intuitively reflect the nature of the data product, and to choose keywords of limited length (i.e., avoid long, complicated and poorly descriptive keywords). It is the NDACC Instrument Working Groups' duty to make sure that the above recommendations are followed.

11. If a concatenation of keywords is proposed, then the concatenation order must be reviewed and approved by the Instrument Working Groups and the NDACC Data Handling Facility (DHF) Manager.

4. Examples of Keywords and their Compliance:

Example 1:

An instrument working group wishes to have its members archive a dataset using a standardized algorithm in addition to in-house processing. The corresponding DATA_SOURCE attribute values could be:

ftir.ch4_ncar001 (continuation of in house algorithm, no new field added)

ftir.ch4_ncar001_std (new added field is keyword "std" that refers to a standardized algorithm)

Example 2:

A team wishes to provide data with daily integration and weekly integration. The corresponding DATA_SOURCE attribute values could be:

mwr.o3_umass002_daily.01 (daily integration, version 01)
mwr.o3_umass002_weekly.01 (weekly integration, version 01)

Example 3:

A team wishes to delineate a dataset that has been homogenized:
sonde.o3_niwa000 (continuation of current dataset, no new field added)
sonde.o3_niwa000_homogn.1.0 (keyword “homogn” describes homogenized dataset)

Example 4a:

A team wants to inform of a data processing version upgrade:
lidar.o3_cnrs.latmos001_algo1.1 (data processor name algo1, processing version number 1)
lidar.o3_cnrs.latmos001_algo1.2 (data processor name algo1, processing version number 2)
or
lidar.o3_cnrs.latmos001_1.1 (no specifics on data processor, processing version number 1.1)
lidar.o3_cnrs.latmos001_1.2 (no specifics on data processor, processing version number 1.2)

Example 4b:

A team wants to clearly establish algorithm reprocessing:
lidar.o3_cnrs.latmos001_algo1.1 (old processor name is algo1, processing version number 1)
lidar.o3_cnrs.latmos001_algo2.1 (new processor name is algo2, processing version number 1)
or
lidar.o3_cnrs.latmos001_1.1 (no data processor name, only processing version number 1.1)
lidar.o3_cnrs.latmos001_2.1 (no data processor name, only processing version number 2.1)

Examples 4a and 4b show that a version change may have multiple meanings. Sections 5 and 6 address the increased complexity and resulting potential ambiguity associated with a change of version number.

Example 5:

A team wishes to link a data processor to a publication:
uvvis.dobson_noaa.esrl061_re2017 (refers to R. Evans et al., 2017)

Example 6:

Use of numerical strings when archiving an updated version
1. ftir.ch4_ncar003_proc01.01.01 replaced later by ftir.ch4_ncar003_proc01.01.02 is allowed
2. ftir.ch4_ncar003_proc01.01.02 replaced later by ftir.ch4_ncar003_proc01.01.01 is forbidden
3. ftir.ch4_ncar003_proc02.01.01 replaced later by ftir.ch4_ncar003_proc01.01.01 is allowed but not recommended)

In the above examples, the numerical strings located after the first dot represent the algorithm version number, and therefore must be incremented positively for each version update occurring forward in time (lines 1 and 2). On the other hand, the numerical string located before the first

dot is part of the data processing keyword, and therefore treated just like any alphabetical character. Although technically allowed, using proc01 after proc02 is not recommended as it creates ambiguity on the meaning of the number itself (line 3).

Example 7:

A bad example:

uvvis.dobson_noaa.esrl061_WD001RE2017BPO3XSuniformT001

Technically allowed, but too long, somewhat overstretched, i.e., not recommended.

Example 8:

Improper use of common keywords

ftir.ch4_ncar001_glass (new keyword “glass” is the same as lidar keyword shown below)

lidar.o3_cnrs.latmos001_glass (new keyword “glass” is the same as ftir keyword shown above)

Let’s assume that the FTIR and Lidar Working Groups are proposing, independently, to use a keyword called “glass” because they just liked the name. The methods used to process the lidar and FTIR data under that keyword are completely separate and completely different (physically and conceptually). This keyword therefore cannot be used by both Instrument Working Groups. Whoever submits this keyword first will be allowed to use the keyword. Whoever submits this keyword second will be denied the use of the keyword. First come, first served!

Example 9:

Proper use of common keywords

ftir.ch4_ncar001_weekly.01.01 (keyword “weekly” is the same as mwr keyword shown below)

mwr.o3_umass002_weekly.01.01 (keyword “weekly” is the same as ftir keyword shown above)

Let’s assume that the FTIR and MW Working Groups are independently proposing to use the keyword “weekly” to archive data with a 1-week granularity. The processors used to process the MWR and FTIR data are physically different, but conceptually, the purpose of the keyword is fulfilled (i.e., weekly average or weekly integration). This keyword therefore can be used by both Instrument Working Groups.

5. New Role for DATA_FILE_VERSION Attribute:

With the previous NDACC data versioning system, each incremented number strictly corresponded to the archive/release of a more recent version (see section 2). The higher the number, the more recent the data file version. Because there was no distinction between algorithm versions, this field could reflect a change in anything pertaining to the production of this file, i.e., data processor change, but also re-processing with parameter changes, technical bugs, etc.

With the new NDACC data versioning system, the DATA_SOURCE attribute is used to describe an algorithm or a product change. The DATA_FILE_VERSION attribute has therefore been re-purposed to simply reflect a sub-version number change within a given algorithm or product version. It is used for small/minor changes, for example to replace a data file that was corrupted, or to replace a data file created with a bug in the algorithm.

When there are only minor algorithm differences in a dataset, the delineation between a “file version change” and “data processing change” can be ambiguous. In order to improve the delineation, it is recommended to follow this simple rule of thumb: If the changes leading to a re-archive were planned, with the purpose of optimizing a dataset, then DATA_SOURCE should

be used. If the changes leading to a re-archive resulted from a-posterior identification of an (unexpected) issue, then DATA_FILE_VERSION should be used. It is equivalent to say that in the former case, “the right answer is refined”, while in the latter case, “the wrong answer is corrected”.

Example 1:

Initial file:

groundbased_lidar.o3_nasa.jpl003_algo.1_table.mountain.ca_20170522t031737z_20170522t051742z_001.hdf

If a bug is found in the above file, the old file is replaced by:

groundbased_lidar.o3_nasa.jpl003_algo.1_table.mountain.ca_20170522t031737z_20170522t051742z_002.hdf

Example 2:

Initial file:

groundbased_lidar.o3_nasa.jpl003_algo.1_table.mountain.ca_20170522t031737z_20170522t051742z_001.hdf

A different data processing version is used, the new file is:

groundbased_lidar.o3_nasa.jpl003_algo.2_table.mountain.ca_20170522t031737z_20170522t051742z_001.hdf

Important note: In example 1, the new file always replaces the old file, and the old file is always removed from the database. In example 2, the new file may either replace the old file, or it may be added to the database as a separate dataset, or as a continuing dataset (see next section).

6. Use of Caution When Providing and/or Accessing Multiple Versions:

When archiving multiple data versions, the NDACC PI must understand all the implications this might have, including the potential for users’ confusion, and the potential for data misuse (wrong version). Uncompromising, highest standards must therefore be used to document each archived version. The documentation must be easily accessible from the data itself, and must describe a specific version in the context of the other archived versions (including consequences on product quality and accuracy). It is also recommended that each NDACC-certified data version be traced to a peer review publication. Each type of product should be highlighted and fully described in the applicable metafiles.

Below is an example illustrating the increased complexity associated with multiple datasets:

Up to year 2000, a PI uses the same algorithm called “algo1” to analyze his data. As a result, the following data files are archived for measurement years 1990 to 2000:

groundbased_lidar.o3_nasa.jpl003_algo1.1_table.mountain.ca_***_001.hdf

After year 2000, the PI uses a new data processor named “algo2”. For some unknown reasons, the PI has no longer access to the old raw data, so he will use the new data processor only for the most recent years. As a result, the following data files for measurement years 2000 to 2010 are archived:

groundbased_lidar.o3_nasa.jpl003_algo2.1_table.mountain.ca_***_001.hdf

In this situation, the PI must clearly document the fact that both types of files belong to the same, continuous dataset (1990-2000 use algo1 and 2000-2010 use algo2).

The above situation is not to be confused with the following situation:

From years 1990 to 2010, a PI uses algorithm “algo1” to analyze his data. As a result, the following data files are archived for measurement years 1990 to 2010:

groundbased_lidar.o3_nasa.jpl003_algo1.1_table.mountain.ca_***_001.hdf

Then in 2010, this PI uses a new algorithm called “algo2”. He decides to reanalyze the full dataset (1990-2010), but sees a good reason (e.g., consistency) to keep the old version. The following data files are therefore archived for measurement years 1990 to 2010:

groundbased_lidar.o3_nasa.jpl003_algo1.1_table.mountain.ca_***_001.hdf and

groundbased_lidar.o3_nasa.jpl003_algo2.1_table.mountain.ca_***_001.hdf

In this case, the PI must clearly document the fact that these are two separate versions, and he/she must indicate which version is better suited for specific applications (trends, validation etc.)

7. Distinguishing NDACC and non-NDACC Data on the NDACC DHF:

Currently, the NDACC DHF provides access to NDACC-certified datasets, either archived in HDF or Ames format, as well as non-NDACC datasets, including “RD” (Rapid Delivery) and “MUSICA” datasets, available in HDF only. The distinction between “NDACC” and non-NDACC is made by ways of keywords in the GEOMS-compliant global attribute DATA_QUALITY.

Because this is a separate problem, the NDACC Data Versioning System described in this document does not discuss the identifier used in the DATA_QUALITY attribute to direct the RD and MUSICA data in their proper repository. Multiple data products can therefore be found in either the NDACC-certified data repository (as long as they are certified following the NDACC protocols), or in the non-NDACC data repositories. However, if a specific product (as defined by the DATA_SOURCE newly-added field) is found in both the non-NDACC and NDACC-certified data repositories, then the DATA_FILE_VERSION attribute of the file in the NDACC-certified repository must always contain a higher value than that of the file in the non-NDACC repository.

We are aware that the presence of NDACC-certified and non-NDACC datasets on the NDACC DHF can cause confusion to data users. This issue is currently being investigated, and the present document will be updated as changes in data organization and other aspects occur.

8. NDACC Data Processing Keyword list:

Appendices A-1 and A-2 contain the list of “Approved” and “Proposed” NDACC Data Processing Keywords. This list is maintained and updated on a regular basis at the NDACC DHF and Instrument Working Group level.

A system of validity inclusion and exclusion criteria was designed to ensure proper use of the keywords within NDACC and within other networks. The criteria are based upon the values of DATA_ACCESS (i.e., the Data Center), DATA_SOURCE, PI_NAME, ORGANIZATION, and DATA_LOCATION. These criteria are compiled and listed together with each approved keyword. The full list, including exclusion and inclusion criteria, is maintained and updated on a regular basis at the NDACC DHF and Instrument Working Group level.

9. Moving Forward with the NDACC Data Versioning System

This versioning system will be refined as needed to ensure that it remains useful, effective, and straightforward. Feedback from NDACC PIs, data users, data providers, and database managers is essential and will always be welcome.