The Generic Earth Observation Metadata Standard (GEOMS)

Version 1.0
March 21, 2011

Authors
Christian Retscher, UMBC, NASA/GSFC
Martine De Mazière, BIRA-IASB
Yasjka Meijer, ESA/ESRIN
Aasmund F. Vik, NILU
Ian Boyd, NIWA
Sander Niemeijer, S[&]T
Robert M. Koopman, ESA/ESRIN
Bojan Bojkov, ESA/ESRIN

Contributors
Terje Krognes, NILU
Rolf Von Kuhlmann, ESA/ESRIN
Thorsten Fehr, ESA/ESRIN
Geoff Buswell, Logica UK Limited

Document Profile Information

Title:	The Generic Earth Observation Metadata Standard (GEOMS)
Issue	v1.0
Issue date:	March 21, 2011
Status:	Final
Audience:	Data providers
Reference(s):	Generic metadata guidelines on atmospheric and oceanographic datasets for the Envisat Calibration and Validation Project [1]. Addendum to the "Generic metadata guidelines on atmospheric and oceanographic datasets for the Envisat Calibration and Validation Project" as implemented by the Aura Validation Data Center (AVDC) [2].
Filename:	geoms-1.0.pdf
Distribution:	Public
Author(s):	Christian Retscher (UMBC, NASA/GSFC, USA, Christian.Retscher@nasa.gov) Martine De Mazière (BIRA-IASB, Belgium, Martine.DeMaziere@bira-iasb.oma.be) Yasjka Meijer (ESA/ESRIN, Italy, Yasjka.Meijer@esa.int) Aasmund F. Vik (NILU, Norway, afv@nilu.no) Ian Boyd (NIWA, New Zealand, i.boyd@niwa.com) Sander Niemeijer (S[&]T, The Netherlands, niemeijer@stcorp.nl) Robert M. Koopman (ESA/ESRIN, Italy, Rob.Koopman@esa.int) Bojan Bojkov (ESA/ESRIN, Italy, Bojan.Bojkov@esa.int)
Contributor(s):	Terje Krognes (NILU, Norway, tk@nilu.no) Rolf Von Kuhlmann (ESA/ESRIN, Italy, Rolf.Von.Kuhlmann@esa.int) Thorsten Fehr (ESA/ESRIN, Italy, Thorsten.Fehr@esa.int) Geoff Buswell (Logica UK Limited, UK, geoff.busswell@logica.com)

Table of Contents

Preface

This document outlines the metadata and data structure requirements developed to facilitate the use of geophysical datasets by improving their portability and accessibility, and by making their contents self-describing. This approach was originally selected to deal with atmospheric [3] and oceanographic datasets, but has been recently expanded to support all measurements from Earth observation instruments.

The Generic Earth Observation Metadata Standard (GEOMS) metadata and data structure requirements described in this document may be applied to any project where data are to be exchanged. More information on GEOMS which complements this document can be found at http://avdc.gsfc.nasa.gov/GEOMS/.

The first application for earlier versions of this document was the definition of metadata requirements for the calibration and validation activities of the Advanced Along Track Scanning Radiometer (AATSR), the Global Ozone Monitoring by Occultation of Stars (GOMOS), the MEdium Resolution Imaging Spectrometer (MERIS), the Michelson Interferometer for Passive Atmospheric Sounding (MIPAS) and the SCanning Imaging Absorption SpectroMeter for Atmospheric CHartographY (SCIAMACHY) sensors flying on the European Space Agency (ESA) Envisat platform [4]. These metadata requirements are as well applied to the validation program of the National Aeronautics and Space Administration (NASA) Earth Observing System (EOS) Aura mission [5] carrying the High Resolution Dynamic Limb Sounder (HIRDLS), the Microwave Limb Sounder (MLS), the Ozone Monitoring Experiment (OMI) and the Tropospheric Emission Spectrometer (TES) atmospheric instruments. The scope of this document has been broadened to NASA A-train satellite instruments, to data from the Network for the Detection of Atmospheric Composition Change (NDACC) and additional ESA Earth observation missions. The latter are supported by the Generic Environment for Calibration/Validation Analysis (GECA) project.

1. Introduction

For geophysical validation, independent observations are performed by a large number of *in-situ*, remote sensing, and satellite instruments for comparison with satellite based geophysical data products. To enhance the usability of the diverse correlative datasets collected for the EOS-Aura validation program and the Envisat calibration and validation campaign (Cal/Val) [6], metadata definitions, covering a broad range of instrument types and geophysical parameters have been established. In support of these efforts, relational databases have been designed to store the metadata and to allow extensive quality assurance (QA) and quality control (QC) of the submitted files, while enabling easy data mining and retrieval of selected datasets. This development was initiated in 1998 through the European Commission (EC) project COSE, Compilation of atmospheric Observations in support of Satellite measurements over Europe [7], and extended in collaboration with ESA, NASA, principal investigators (PI) of the Envisat and Aura validation campaign, and selected PIs from NDACC, for the implementation of a uniform data exchange standard.

These GEOMS guidelines describe the standard metadata definitions adopted for the correlative, experimental and model data archived for the EOS-Aura validation program, the Envisat calibration and validation campaign, data from NDACC, and the GECA project, which supports existing and future ESA calibration and validation programs. The definitions have been carefully chosen to allow applicability to other scientific endeavors.

The document is structured as follows. It begins with an overview of the definitions of the metadata standard, where we outline some terminology, structure and general guidelines (cf. Chapter 2). The next Chapter 3 describes metadata parameter formatting issues with details on character sets, special characters, capitalization, used numerical types and date/time formatting. The two core structure elements of this metadata standard, the global attributes (GA) and the variable attributes (VA), are discussed in detail in Chapters 4 and 5, respectively, Chapter 6 introduces the implementation of GEOMS and the used hierarchical data formats. GEOMS attribute values, templates and examples are found in Tables in Chapter 7. In order to support the user when dealing with conversion of e.g. date/time we give code examples in Appendix A.

2. Definitions

The multidisciplinary exchange of data in programs such as the EOS-Aura validation program, the Envisat calibration and validation campaigns, NDACC or GECA, depend heavily on good definitions for data. Freedom of choice would let different end-users describe identical datasets in very different terms, thus hindering effective data usage. To avoid this, there is a necessity to describe the contents of the data files unambiguously. Therefore, consistent sets of descriptors characterizing the data are defined. These descriptors, the so-called metadata elements, are each allowed a limited set of predefined values. The following Chapters will introduce generic metadata definitions for correlative datasets as they are used in Envisat and Aura validation as well as for NDACC and GECA. They will further provide guidelines for the introduction of metadata for future cooperating data centers (DC).

2.1. Terminology

Metadata: Data about data, e.g. information that describes, characterizes and indexes the scientific data.

Attribute: A single metadata element. This document will define the attribute names, as well as their allowed values (or metadata entries).

Attribute field: An attribute field is an element in a multi-field attribute entry. For example, a person's name attribute consists of both family name and first name attribute fields.

Parameter: A geophysical entity that is measured or computed.

Dataset: A set of one or more parameter data reported in coincident time and space. A time-series of such datasets is also referred to as a dataset.

Variable: When a parameter is reported in a dataset, it will be represented by one or more variables of the dataset. The variables will generally be identified by two fields, namely variable name and variable mode, and possibly a third field called variable descriptor.

Variable name: The **mandatory** primary name of the geophysical entity (parameter) reported in the dataset, from an observation or estimation by a measurement or model calculation, for example pressure.

Variable mode: The **optional** mode field generally indicates “how” the parameter was measured/calculated to obtain the current parameter data, for example, absorption vs. emission.

Variable descriptor: The **optional** descriptor field is used to distinguish the primary variable from a related variable that provides “additional information” about the geophysical parameter in question, such as its uncertainty, its minimum, a flag, etc.

Data source: An instrument or a model from which the reported data originate.

Data location: The location name associated with sampled or modeled data. The position may be varying with time, e.g., when data are taken from a moving platform such as a plane, balloon, ship, drifting buoy, or satellite.

Data dependence: Parameters or variables that are provided as a function of another parameter or variable (for example temperature as a function of time) are considered dependent parameters. The parameter(s) on which the dependent parameter depends (time in the example) are the independent parameter(s). The number of independent parameters determines the dimensionality of the grid on which the dependent parameter is reported.

2.2. File Structure

Data and associated metadata are combined into data files. Sufficient metadata must be available in each file (as specified in Chapters 4 and 5) to make the data useful to the end-user and to meet the requirements outlined here for proper data indexing. End-users must be able to use any data file adhering to the outlined standard without having to search for relevant metadata in other sources.

Metadata are divided into 1) one set of global attributes, which pertain to the identification of the file itself and to the dataset contained in the file as a whole, and 2) an ensemble of sets of variable attributes, each set pertaining to one single variable within the dataset. There is one set of variable attributes per variable in the data file.

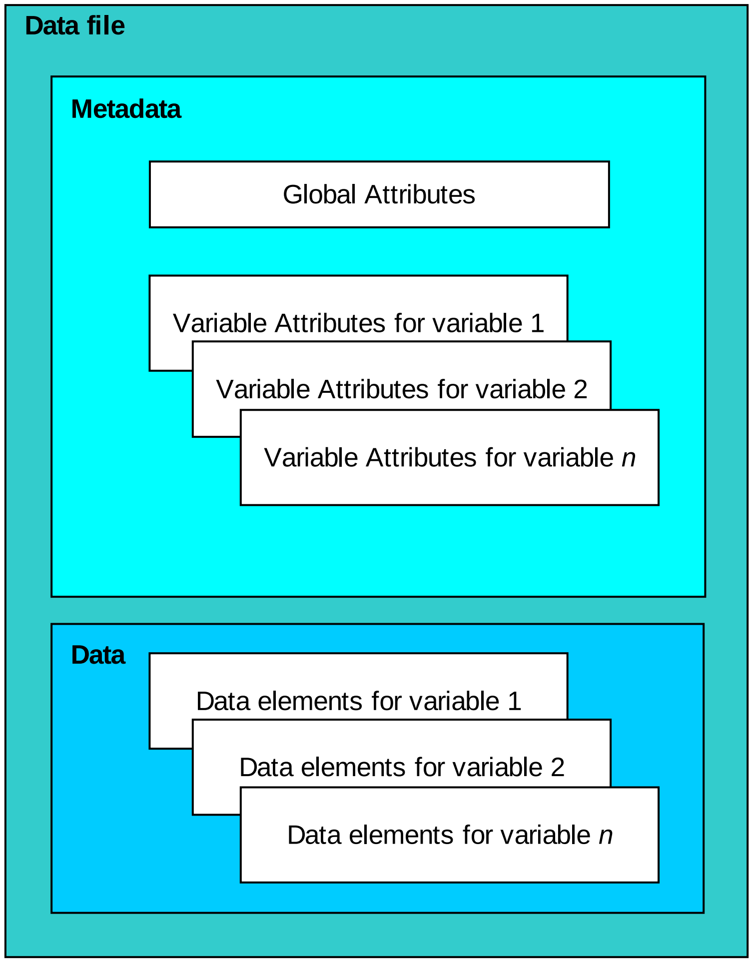

Figure 2.1 Overview of the data file structure.

A data file consists of data from one single data source, auxiliary data (such as related meteorological data), and metadata that describe the data. For a dependent parameter, the independent parameter on which it depends must be available in the dataset.

For the EOS-Aura, Envisat validation, NDACC and GECA activities, the files must meet the metadata standard presented in this document, as well as being formulated as Hierarchical Data Format (HDF) type files [8]. Some data centers support as well the network Common Data Form (netCDF) [9]. HDF and netCDF implementations are described in Chapter 6.

2.3. General Guidelines

The purpose of these comprehensive guidelines is to reduce the difficulties and ambiguities in dealing with the complexity inherent with data exchange. The majority of typical errors will be detected at a data center (e.g. the Envisat Validation Data Center [10]) before the dataset is indexed and added to a file tree. This form of quality assurance and quality control constitutes an improvement in the management of data compared to a system that is not supported by clearly defined nomenclature and formulations. The resulting unambiguity in reporting datasets will improve their scientific use and enable users to validate datasets more effectively. In this metadata standard the following rules are defined to optimize the effective exchange of data and efficient data management.

- The unambiguous identification of the various parameters is of great importance to data exchange. The associated variable name attribute `VAR_NAME` (cf. Sections 4.2.6 and 5.1.1), consisting of the **variable name**, and possibly the **variable mode** and/or the **variable descriptor**, assigns a unique name to each variable that refers to the same underlying geophysical entity. For this reason, the **variable name** must contain a basic description in physical terms of the geophysical quantity that it represents, for example `PRESSURE`. The **variable mode** emphasizes those aspects of the reported quantity related to the method of estimation that may affect comparisons with other estimates from differing measurement methods or calculations. The third entry, the **variable descriptor**, is used to construct a related variable that contains derived and/or additional information about the main variable, for example: the error or uncertainty associated with the pressure data.

- The **variable mode** or the **variable descriptor** should not be used to distinguish estimation methods that are characterized by the use of specific but potentially different values of a reference or calibration quantity. Typical examples are reference wavelength or pre-defined depths. If the variable is reported for several **reference values** under the same conditions of variable mode, then the variable should be reported as a dependent variable, dependent on the reference values; these reference values become the independent parameters. If the variable is reported for only one reference value, then this reference value is reported as a constant independent variable, and there is no need to make the reported parameter depend on this single reference parameter.

- Time, latitude, and longitude position parameters are **mandatory**. Geolocation must be specified in terms of date, time, latitude, longitude and possibly a vertical coordinate. Such a coordinate can be altitude, depth, pressure or geopotential height. If no altitude parameter is measured, the altitude position of the instrument should be given, if it makes sense for the observation. Preferably the geolocation must describe the effective date, time and location of the geophysical quantity that is reported. The geolocation of the sampling instrument may also be provided as additional information to the actual location of the reported geophysical quantity.

- GEOMS describes many **mandatory** and **optional** attributes. Optional attributes may be omitted when indicated in the “Req” (Requirements) column in Tables 7.1 to 7.4 in Chapter 7. Some attributes are **optional**, but may be **mandatory** when data is reported compliant to a GEOMS template (GEOMS-TE; cf. Section 2.4). Files must contain all mandatory attributes, may contain optional attributes, but may also contain non-standard attributes. The usage of non-standard attributes is not a violation to this standard, but it is preferred to use attributes described within this document.

- Default or missing entries must be set to the appropriate fill value described by the attribute

VAR_FILL_VALUE in Section 5.1.11. With the supported GEOMS data formats (cf. Chapter 6) empty data entries are not possible.

- GEOMS provides several rules and guidelines regarding the dimensions of variables. Each dimension of a variable can either be associated with an axis (such as time, latitude, longitude, height/depth, etc.) or can be an independent dimension. Each axis comes with a variable, the axis variable, that contains the values along that axis. When referring to a dimension that corresponds with an axis then in GEOMS the name of the axis variable will be used.

  A variable that has more than one dimension shall follow the so-called "C convention" for **dimension ordering** when describing information regarding its dimensions. With the C convention the last dimension (writing from left to right) is fastest running when enumerating all elements, compared to the Fortran convention, where the first dimension is fastest running. Note that different file access libraries may have different conventions with regard to how they deal with array ordering in their function interfaces. What is important is that a multidimensional variable is stored with the proper dimensions as fastest/slowest running. This means that storing a DATETIME;ALTITUDE variable as a DATETIME × ALTITUDE array with C ordering convention or storing it as an ALTITUDE × DATETIME array with Fortran ordering convention is both correct (since in both cases the DATETIME dimension will be the slowest running). However, attributes for a variable should, as mentioned before, still only follow the C ordering convention. So in both cases the VAR_DEPEND attribute should equal DATETIME;ALTITUDE. Note, IDL natively follows the Fortran convention, but when using the IDL HDF library command by default output data is transposed.

  For many file formats there is the concept of an "appendable dimension", which is the dimension that grows when more data is added. Data in GEOMS should be structured in such a way that the slowest running dimension is the appendable dimension. In most cases this is the dimension that corresponds to the DATETIME axis. Overall, the following order of dimensions (from slowest to fastest running) should be used: DATETIME, LATITUDE, LONGITUDE, ALTITUDE (or PRESSURE, ALTITUDE.GPH, etc.), WAVELENGTH (or WAVENUMBER), and then any other dimensions. Any independent dimension should always come last (e.g. they should be the fastest running dimension). Except for appendable dimensions, any axis that only consists of one element should not be used as a dimension. For instance, if there is only one LATITUDE and LONGITUDE value applicable for the whole file, then variables should not have a LATITUDE or LONGITUDE dimension.

  Axis variables are in most cases one-dimensional. However, when an axis such as a height grid differs per time value the axis variable can be provided as a two-dimensional variable. The first dimension of such an axis variable should then map to the appendable dimension (e.g. DATETIME) and the second dimension should map to its own axis. See Section 5.1.5 for some examples.

2.4. Template Guidelines

Some GEOMS compliant measurements **must be reported** according to guidelines as set in GEOMS templates (cf. Chapter 7). In general, a GEOMS template states, which metadata attributes and data variables must be reported. The general guidelines set above and throughout this document are **valid for all templates** and all rules **must be followed**. Templates cannot be more loose on general rules, but only more **restrictive** in the usage of metadata attributes and data variables. Templates may e.g. rule the usage of generally **optional**, but **mandatory** variable attributes if used in the GEOMS-TE context.

3. Metadata Element Formatting Issues

3.1. Character set, Special characters, and Capitalization

- All metadata entries use the printable ISO 646/US ASCII [11] character set.
- All attribute names and values must be used as outlined in the following Sections and in Tables 7.1 to 7.12.
- Free format attribute values must abide by the character set restriction set above, but non-printable characters horizontal tab (HT), line feed (LF), and carriage return (CR) are allowed.
- In multi-field attribute values the semicolon character ";" is reserved for specific purposes. No white space is allowed before and after this symbol.
- All attribute names are set in upper case and are used case sensitive.
- All attribute values which are outlined in the following Sections and have specific format requirements, as well as all attribute values which are defined in Tables 7.1 to 7.12 are used case sensitive.

3.2. Numerical Types

The currently accepted binary representations (or numerical types) are found at `VAR_DATA_TYPE` in Table 7.4. The used floating-point representations follow the IEEE 754 standard for floating-point arithmetic. For more details on the supported data formats please see Chapter 6.

3.3. Date and time formats

There are two accepted date/time formats used in these guidelines: a numerical format (`MJD2K`) for data reporting of the variable attribute `DATETIME` and any other variable attributes which have `VAR_DATA_TYPE = MJD2K`; and a string format, ISO 8601 [12], for file name construction and use in global metadata attributes `DATA_START_DATE`, `DATA_STOP_DATE`, and `FILE_GENERATION_DATE`. For reasons of **compatibility to UDUNITS** [13], `MJD2K` is used as a short form for MJD2000.

3.3.1. Date/Time - MJD2K

The Modified Julian Date, `MJD2K`, used throughout this document is defined as follows:

`MJD2K` is 0.000000 on January 1, 2000 at 00:00:00 UTC

Special care must be given to the formatting of `MJD2K` by reporting the appropriate number of significant digits after the comma to represent the actual time resolution:

for minutes [min.]	…	*minimum 3 decimals*
for seconds [s]	…	*minimum 4 decimals*
for milliseconds [ms]	…	*minimum 7 decimals*

3.3.2. Date/Time - ISO 8601

The UTC `DATETIME` representation in ISO 8601 [12] long format is:

YYYYMMDDThhmmssZ

where

YYYY	*is the numeric year*
MM	*is the numeric month*
DD	*is the numeric day*
hh	*is the numeric hour*
mm	*is the numeric minute*
ss	*is the numeric second*
T	*is the time delimiter*
Z	*indicates the Universal Time (UTC)*

3.3.3. Use of leap seconds

Note that the `MJD2K` format that is used to store time information in variables is unable to represent leap seconds. For instance, at the end of 2005 a leap second was introduced with associated UTC time value “2005-12-31T23:59:60”. For `MJD2K` values, the introduction of a leap second will for GEOMS be formally captured by an epoch shift. This means that the datetime offset (e.g. what is considered the time at start of January 1, 2000) should then be different for the `MJD2K` values for UTC “2005-12-31T23:59:59” and UTC “2006-01-01T00:00:00”. This additional difference of 1 second in the epoch would then accommodate for the fact that the official time interval between these two UTC datetime values is not 1 but 2 seconds. This approach is straightforward, since conversion between `MJD2K` binary and UTC string representation will in most cases be just as if there were no leap seconds. The only exception is the datetime value at a leap second introduction itself (e.g. “2005-12-31T23:59:60”). Routines that convert from a UTC string to an `MJD2K` value will have to support this exceptional case where the amount of seconds can be 60. In this case the value should be the same as that given for the next second. For example, both “2005-12-31T23:59:60” and “2006-01-01T00:00:00” will result in an `MJD2K` value of 189388800.0. Formally these values are different, since they are related to different epochs. However, since the associated epoch is not tracked within GEOMS, this information is lost and data users (DU) should thus be aware that this ambiguity exists. If an application has no other information to determine the proper epoch when converting an `MJD2K` value back to a UTC string, it should assume the most recent epoch. For our example this means that 189388800.0 would convert to the UTC string “2006-01-01T00:00:00”.

4. Global Attributes

To organize the global attributes, they have been grouped into three categories, namely **originator attributes** (cf. Section 4.1), **dataset attributes** (cf. Section 4.2), and **file attributes** (cf. Section 4.3). Each one of the global attributes appears once in each data file. All attributes marked with an "**x**" in the "Req" (Requirement) column in Tables 7.1 to 7.3 are **mandatory** and must have an entry. For the ones marked with an "**o**", the attribute is **optional** and may be omitted. Hereinafter, all attributes are mandatory unless mentioned otherwise explicitly.

4.1. Global Originator Attributes

The **global originator attribute** entries describe the ownership of the data found in a given file. Remember that all entries for person names are case-sensitive.

4.1.1. PI_NAME

The global attribute `PI_NAME` contains the data's Principal Investigator's (PI) Name. The PI has the main scientific and/or institutional responsibility for the given data. In general, the PI attributes refer to the instrument/model PI exclusively. In some instances, where no single Principle Investigator exists, the `PI_NAME` and `PI_AFFILIATION` (cf. Section 4.1.2) must hold the name of the entity responsible for the instrument/model.

Type:	STRING
Format:	Family name;Given name
Entry:	Two semicolon-separated fields
Example:	`PI_NAME = Retscher;Christian`

4.1.2. PI_AFFILIATION

The global attribute `PI_AFFILIATION` contains the Principal Investigator's **official** affiliation name preferably in English and affiliation acronym (cf. Table 7.1).

Type:	STRING
Format:	Affiliation name;Affiliation acronym
Entry:	Two semicolon-separated fields
Example:	`PI_AFFILIATION = NASA Goddard Space Flight Center;NASA.GSFC`

4.1.3. PI_ADDRESS

The global attribute `PI_ADDRESS` contains the Principal Investigator's **official** mailing address. The country name must be an official short name entry in English as listed in ISO 3166-1 [14].

Type:	STRING
Format:	Address;Postal code and locality;Country name

Entry: Three semicolon-separated fields
Example: `PI_ADDRESS = NASA GSFC Code 613.3; Greenbelt, MD 20771;UNITED STATES`

4.1.4. PI_EMAIL

The global attribute `PI_EMAIL` is the Principal Investigator’s e-mail address.

Type: STRING
Format: Follow RFC2822
Entry: Single field
Example: `PI_EMAIL = christian.retscher@nasa.gov`

4.1.5. DO_NAME

The global attribute `DO_NAME` contains the Data Originator’s (DO) Name. The DO is the person that generated and quality controlled the data. Where no single Data Originator (DO) exists, the `DO_NAME` and `DO_AFFILIATION` (cf. Section 4.1.6) will hold the name of the entity responsible for the instrument, while the `DO_ADDRESS` (cf. Section 4.1.7) and `DO_EMAIL` (cf. Section 4.1.8) will contain the appropriate contact information. The DO may or may not be the same person as the PI.

Type; STRING
Format: Family name;Given name
Entry: Two semicolon-separated fields
Example 1: `DO_NAME = Ofstad;Thor`
Example 2: `DO_NAME = National Oceanic and Atmospheric Administration;NOAA`

4.1.6. DO_AFFILIATION

The global attribute `DO_AFFILIATION` contains the Data Originator’s **official** affiliation and acronym. The `DO_AFFILIATION` may differ from the `PI_AFFILIATION` (cf. Table 7.1).

Type: STRING
Format: Affiliation name;Affiliation acronym
Entry: Two semicolon-separated fields
Example 1: `DO_AFFILIATION = Norwegian Institute for Air Research;NILU`
Example 2: `DO_AFFILIATION = National Oceanic and Atmospheric Administration;NOAA`

4.1.7. DO_ADDRESS

The global attribute `DO_ADDRESS` contains the Data Originator’s mailing address. The country name must be an official short name entry in English as listed in ISO 3166-1 [14]. The `DO_ADDRESS` may differ from the `PI_ADDRESS`.

Type: STRING
Format: Address;Postal code and locality;Country name
Entry: Three semicolon-separated fields
Example 1: `DO_ADDRESS = P.O. Box 100;N-2027 Kjeller;NORWAY`
Example 2: `DO_ADDRESS = 14th Street and Constitution Avenue NW, Room 6217; Washington, D.C. 20230;UNITED STATES`

4.1.8. DO_EMAIL

The global attribute `DO_EMAIL` contains the Data Originator's e-mail address. The `DO_EMAIL` may differ from the `PI_EMAIL`.

Type: STRING
Format: Follow RFC2822
Entry: Single field
Example 1: `DO_EMAIL = thor.ofstad@nilu.no`
Example 2: `DO_EMAIL = answers@noaa.gov`

4.1.9. DS_NAME

The global attribute `DS_NAME` contains the Data Submitter's (DS) name. The Data Submitter is the person that submitted the data to the data center.

Type; STRING
Format: Family name;Given name
Entry: Two semicolon-separated fields
Example: `DS_NAME = De Maziere;Martine`

4.1.10. DS_AFFILIATION

The global attribute `DS_AFFILIATION` contains the Data Submitter's **official** affiliation and acronym. The `DS_AFFILIATION` may differ from the `PI_AFFILIATION` and `DO_AFFILIATION` (cf. Table 7.1).

Type: STRING
Format: Affiliation name;Affiliation acronym
Entry: Two semicolon-separated fields
Example: `DS_AFFILIATION = Belgian Institute for Space Aeronomy;BIRA.IASB`

4.1.11. DS_ADDRESS

The global attribute `DS_ADDRESS` contains the Data Submitter's mailing address. The country name must be an official short name entry in English as listed in ISO 3166-1 [14]. The `DS_ADDRESS` may differ from the `PI_ADDRESS` and `DO_ADDRESS`.

Type: STRING

Format: Address;Postal code and locality;Country name
Entry: Three semicolon-separated fields
Example: `DS_ADDRESS = Ringlaan 3;B-1180 Brussels;BELGIUM`

4.1.12. DS_EMAIL

The global attribute `DS_EMAIL` contains the Data Submitter's e-mail address. The `DO_EMAIL` may differ from the `PI_EMAIL` and the `DO_EMAIL`.

Type: STRING
Format: Follow RFC2822
Entry: Single field
Example: `DS_EMAIL = Martine.DeMaziere@bira-iasb.oma.be`

4.2. Global Dataset Attributes

The **global dataset attributes** provide a general description of the data contained in the given file, as well as guidelines for the use and/or publication of these data. These attributes include among others, the type and identity of the instrument or model that was used to generate the reported data, information on the geolocation of the data, the field of research, and a list of the data variables included in the file.

4.2.1. DATA_DESCRIPTION

The **optional** global attribute `DATA_DESCRIPTION` contains a brief sentence summarizing the file's data content.

Type: STRING
Format: Descriptive text, free format
Entry: Single field
Example: `DATA_DESCRIPTION = Weekly NILU ozonesonde launch from Orland, Norway`

4.2.2. DATA_DISCIPLINE

The global attribute `DATA_DISCIPLINE` describes the field of research to which the data in the file belongs and the data acquisition method. An entry consists of three fields: Discipline field, Acquisition method, and Acquisition platform. For a list of allowed variables please refer to Table 7.2.

Type: STRING
Entry: Three semicolon-separated fields
Format: Discipline;Acquisition method;Acquisition platform
Example: `DATA_DISCIPLINE = ATMOSPHERIC.CHEMISTRY;INSITU;BALLOON`

4.2.3. DATA_GROUP

The global attribute DATA_GROUP has a two-field entry, specifying the origin of the data (experimental, model, or a combination of both) and the spatial characteristics of the data. The spatial characteristics indicate the dimensionality of the spatial grid of the dataset for a single data element, in addition to whether the spatial grid moves in space with time. The dimensionality that is expressed in DATA_GROUP by SCALAR (0D), PROFILE (1D), and FIELD (2D or more) refers to the spatial dimensionality of the target in the dataset only. This is best explained by considering the example of an airborne LIDAR system: At a given point in time, this LIDAR system provides measurements at a single latitude and longitude location but for multiple altitudes. With time, this 1-dimensional spatial grid (fixed latitude and longitude, vector of altitudes) is moving in latitude and longitude. The two-field entry for this example thus becomes EXPERIMENTAL;PROFILE.MOVING. For a list of allowed variables please refer to Table 7.2.

Type:	STRING
Entry:	Two semicolon-separated fields
Format:	Data origin;spatial characteristics
Example 1:	DATA_GROUP = EXPERIMENTAL;SCALAR.STATIONARY *for a time-series of groundbased column measurements*
Example 2:	DATA_GROUP = MODEL;FIELD.STATIONARY *for a 3-D model output on a fixed spatial grid*

4.2.4. DATA_LOCATION

The global attribute DATA_LOCATION contains the identification of the location of the reported geophysical quantities. In general, DATA_LOCATION identifies the fixed location of an instrument, such as a station name; or of a moving platform such as a plane, a ship, a satellite, from which the data originated. For some cases, e.g. when model results are reported, DATA_LOCATION refers to the spatial data coverage. For a list of allowed variables please refer to Table 7.2.

Data measured by **satellite instruments** are treated differently than other moving platforms. In case the data location is not known, the following reserved words shall be used. In case a full orbit is reported, "**orbit**" shall be used. In case only parts of an orbit is reported, then "**subset**" shall be used. Note, the DATA_LOCATION appears in the file name of the reported data.

Type:	STRING
Entry:	Single field
Format:	Refer to Table 7.2. If the name consists of two or more words, they are separated with periods ".", blanks (white space characters) must not occur in the names.
Example 1:	DATA_LOCATION = KIRUNA
Example 2:	DATA_LOCATION = ORBIT
Example 3:	DATA_LOCATION = SUBSET

4.2.5. DATA_SOURCE

The global attribute DATA_SOURCE consists of two underscore separated fields. The first field describes the type of instrument or numerical model that created the data. The type may consist of several dot-separated words. The main constituent reported in the file should extend the DATA_SOURCE, e.g. LIDAR.O3, or

FTIR.NO2. The second field is the acronym of the institute/organization (cf. affiliation acronyms link in Table 7.1 above) that owns the instrument/model. This institute/organization may differ from the affiliations of the PI, the DO and the DS, but it is usually the PI's or DO's. Concatenated to the institute/organization acronym is a unique numeric identifier of the instrument.

The instrument identification system is **mandatory** for all instruments, including single or disposable instruments. Each organization must assure that different instruments of the same type have unique identifiers, even if they are operated in different locations. For example, if NASA owns two LIDAR instruments (LIDAR.O3), the entire entry for the DATA_SOURCE attribute for NASA's second instrument would become: LIDAR.O3_NASA.GSFC002. This instrument identification system allows each organization to create a **unique identifier** for each instrument, without conflicting with other organizations instrument identifiers. Different organizations may operate several instruments of the same type at the same location without conflicting instrument identification. The instruments may be used at different locations at different times, while the instrument history remains traceable. When an instrument is taken out of service, the identifier **must not be reused** for another instrument. A particular case exists for instruments that are used as "consumables", for example weather balloons that are often lost after the sounding flight. In such cases a unique identifier may be useless. The identifier 000 is therefore reserved for the non-unique case. A laboratory may re-use this particular identifier any number of times.

For satellite data from the original data release, DATA_SOURCE shall be reported with an instrument name and a platform name. e.g. SCIAMACHY.ENVISAT, or TOMS.EARTHPROBE. The main constituent reported in the file should extend the DATA_SOURCE, e.g. SCIAMACHY.ENVISAT.O3 or TOMS.EARTHPROBE.O3. The second part of DATA_SOURCE refers to the satellite operating agency and an instrument identification number, e.g. SCIAMACHY.ENVISAT.NO2_ESA001. For data which were derived from the original data release, were reprocessed or filtered by algorithms developed by third parties, the second part of DATA_SOURCE shall refer to the third party's acronym and an instrument identification number, e.g F3C.FM1_UGRAZ.WEGC001. Note, the DATA_SOURCE appears in the file name of the reported data.

Actually used DATA_SOURCEs: Brand names shall not be included in the DATA_SOURCE. A distinction of instruments from different manufactures can be made in the second field by numbering. Alternatively a more detailed description can be given in the VAR_DESCRIPTION field. For a list of allowed variables please refer to Table 7.2.

Type:	STRING
Entry:	Two underscore "_" concatenated fields
Format:	Instrument or model type (from Table 7.2) "_" institute/organization acronym (from Table 7.1) concatenated with a unique three-digit identifier (for example 001, 007, 111, or 000 for disposable equipment).
Example 1:	DATA_SOURCE = SONDE.O3_NILU000
Example 2:	DATA_SOURCE = FTIR.CH4_NILU001
Example 3:	DATA_SOURCE = LIDAR.O3_NASA.GSFC002
Example 4:	DATA_SOURCE = SCIAMACHY.ENVISAT.NO2_ESA001

4.2.6. DATA_VARIABLES

The global attribute DATA_VARIABLES lists all data variables reported in the current data file. The list is a succession of fields in the `DATA_VARIABLES` entry; in other words, the entry consists of one field per variable. Generally, a field consists of the **variable name**, the **variable mode** and the **variable descriptor**, separated by underscores.

Type:	STRING
Entry:	Arbitrary number of semicolon-separated fields (each field constructed according to the format below)
Format:	Variable name[_Variable mode][_Variable descriptor]
Example 1:	`DATA_VARIABLES = DATETIME;LATITUDE;LONGITUDE;ALTITUDE.GPH; O3.PARTIAL.PRESSURE_INSITU;PRESSURE_INSITU; TEMPERATURE_INSITU;HUMIDITY.RELATIVE_INSITU; INTERNAL.BOX.TEMPERATURE_INSITU`
Example 2:	`DATA_VARIABLES = DATETIME;LATITUDE.INSTRUMENT; LONGITUDE.INSTRUMENT;ALTITUDE.INSTRUMENT;ALTITUDE; H2O.COLUMN_ABSORPTION.SOLAR`

where `ALTITUDE` *is an independent parameter and* `H2O.COLUMN_ABSORPTION.SOLAR` *is dependent on* `DATETIME` *and* `ALTITUDE`.

The variables `DATETIME`, `ALTITUDE(.INSTRUMENT)` or any other vertical coordinate, `LATITUDE(.INSTRUMENT)` and `LONGITUDE(.INSTRUMENT)` variables are always **mode-less** and **descriptor-less**. All other variables can have an **optional mode** and an **optional descriptor** as appropriate.

With some exceptions (cf. Section 4.2.6.4) variable names first refer to a geophysical entity followed by a physical property.

4.2.6.1. Variable name

The **variable name** is a basic identification of the data parameter described in the dataset, e.g., the geophysical entity that is reported. The name includes the chemical or physical identification of the reported entity. A typical example of a variable name is the column amount of ozone:

```
O3.COLUMN
```

4.2.6.2. Variable mode

The **variable mode** provides the information on the measurement/calculation method that can lead to differences when comparing to results obtained with other methods to observe/calculate the same quantity. It is **optional** in the construction of every field in the `DATA_VARIABLES` entry. Examples related to ozone observations compliant with Table 7.2 are:

```
O3.COLUMN.SLANT_ABSORPTION.SOLAR
O3.COLUMN_ABSORPTION.SOLAR
```

4.2.6.3. Variable descriptor

The **variable descriptor** is added only to construct variables that provide additional information about a primary variable or that describe some particular property of that variable. For instance for the ozone examples above, we may add additional variables as follow:

```
O3.COLUMN.SLANT_ABSORPTION.SOLAR_UNCERTAINTY.STDEV
O3.COLUMN_ABSORPTION.SOLAR_UNCERTAINTY.STDEV
```

The descriptor is not intended to distinguish subsets of a dataset. Such dataset distinctions should be made by providing additional dependent or independent parameters, as outlined in the following examples:

Example 1:

The ozone columns obtained by SAOZ measurements are traditionally distinguished in two subsets: measurements at dawn and measurements at dusk. The solar azimuth angle is the relevant parameter distinguishing these measurements and should be provided together with every measurement of the ozone column.

Example 2:

Irradiance/radiance measurements are often performed at specific wavelengths. Wavelength should therefore be an independent parameter if values at more than one wavelength are reported.

Example 3:

For an instrument capable of measuring in alternating modes, for example zenith sky vs. direct sun, the data taken in these modes should be reported as distinct variables constructed with different variable modes.

```
O3.COLUMN.SLANT_ZENITH
O3.COLUMN.SLANT_ABSORPTION.SOLAR
```

4.2.6.4. Special variables

`PRESSURE`, `TEMPERATURE`, `SURFACE.TEMPERATURE`, `DENSITY`, and `HUMIDITY`:

Special rules apply for variable names **referring to air**. The variable name to be used does not explicitly carry the `AIR` keyword, but implicitly refers to air, e.g. `TEMPERATURE` instead of `AIR.TEMPERATURE`. Note, `SURFACE.TEMPERATURE` refers the the air temperature above the surface. In case another medium than `AIR` is measured, the medium or identifier will be on the first position, e.g. `INTERNAL.BOX.TEMPERATURE`, `ICE.TEMPERATURE`, or `LAND.SURFACE.TEMPERATURE`. The same approach is valid for gases mixed in air, e.g. `CH4.MIXING.RATIO`. The only variable still carrying the `AIR` keyword is `AIR.MASS`.

`DATETIME`, `LATITUDE`, `LONGITUDE`, `ALTITUDE[.GPH]`, `PRESSURE`, `DEPTH`:

These variable names refer to effective geolocation (mid-time/point) of a reported parameter.

4.2.6.5. Geolocation

Every data file **must** contain a specification of geolocation in **three or four dimensions** (cf. Section 2.3), specifying when and where the reported data is sampled. In other words, the `DATETIME` variable, as well as latitude and longitude are **mandatory** with a vertical geolocation variable only **mandatory** if the measurement has been taken in the corresponding dimension. The vertical geolocation should be expressed as `ALTITUDE` or `DEPTH`. If `ALTITUDE` is not available, acceptable substitutes are `PRESSURE` and `ALTITUDE.GPH` (geopotential height), etc. If no altitude parameter is measured, the altitude of the instrument position should be given, if it makes sense for the observation, e.g. `ALTITUDE.INSTRUMENT`.

The geolocation variables must comply with stringent naming criteria. Acceptable combinations of geolocation variables are:

- `DATETIME;LATITUDE;LONGITUDE;[ALTITUDE|ALTITUDE.GPH|PRESSURE]`
- `DATETIME;LATITUDE;LONGITUDE;DEPTH`
- `DATETIME;LATITUDE.INSTRUMENT;LONGITUDE.INSTRUMENT;`
  `ALTITUDE.INSTRUMENT;[LATITUDE;LONGITUDE;`
  `[ALTITUDE|ALTITUDE.GPH|PRESSURE]]`

4.2.7. DATA_START_DATE

The global attribute `DATA_START_DATE` specifies the earliest date/time found in the data file. The `DATA_START_DATE` format is the ISO 8601 data/time described in Sections 3.3.1 and 3.3.2. Fractions of a second shall be rounded down to the nearest second.

Type:	STRING
Entry:	Single field
Format:	ISO 8601 date/time
Example:	`DATA_START_DATE = 20010124T110000Z`

4.2.8. DATA_STOP_DATE

The global attribute `DATA_STOP_DATE` specifies the latest date/time found in the data file. The `DATA_STOP_DATE` format is the ISO 8601 data/time described in Sections 3.3.1 and 3.3.2. Fractions of a second shall be rounded up to the nearest second.

Type:	STRING
Entry:	Single field
Format:	ISO 8601 date/time
Example:	`DATA_STOP_DATE = 20010124T110500Z`

4.2.9. DATA_FILE_VERSION

The global attribute `DATA_FILE_VERSION` specifies the version of the data. It is not associated with a scientific algorithm or a processing algorithm, the attribute entry specifies an arbitrary version of the file, beginning with `001` (with leading zeroes). With each update the data file version shall be incremented by 1.

Type:	STRING
Entry:	Single field

Format: [ddd]
Example 1: `DATA_FILE_VERSION = 003`

4.2.10. DATA_MODIFICATIONS

The **optional** global attribute `DATA_MODIFICATIONS` is intended to describe the data modification history associated with `DATA_FILE_VERSION` found in the data file. Detail of the information is up to the discretion of the data originator.

Type: STRING
Entry: Single field
Format: Free format
Example: `DATA_MODIFICATIONS = Version 002, uses the pump correction table of Komhyr (1986); Version 003, box temperature correction.`

4.2.11. DATA_CAVEATS

The **optional** global attribute `DATA_CAVEATS` refers to potential issues with the data in the current data file and shall inform the user to use this data with caution.

Type: STRING
Entry: Single field
Format: Free format
Example 1: `DATA_CAVEATS = Possible problems with the telemetry.`
Example 2: `DATA_CAVEATS = This is near real-time data, final revised data will be available within 3 months.`

4.2.12. DATA_RULES_OF_USE

The **optional** global attribute `DATA_RULES_OF_USE` entry is the PI's guidelines for the data usage. This entry is usually guided through a specific project data protocol.

Type: STRING
Entry: Single field
Format: Free format
Example: `DATA_RULES_OF_USE = Refer to Envisat Cal/Val data protocol, for more information contact nadirteam@nilu.no.`

4.2.13. DATA_ACKNOWLEDGEMENT

The **optional** global attribute `DATA_ACKNOWLEDGEMENT` specifies the PI's "desired" acknowledgment of the data when used in publications, presentations, etc.

Type: STRING
Entry: Single field

Format: Free format
Example: `DATA_ACKNOWLEDGEMENT = We thank the NILU ozonesonde team for providing us with the revised ozonesonde data from Orland.`

4.2.14. DATA_QUALITY

The global attribute `DATA_QUALITY` specifies information on quality of the data. This attribute is **mandatory** for datasets which follow a **GEOMS template** description, otherwise the use is **optional**. Currently GEOMS does not regulate the content of `DATA_QUALITY`, thus the data quality classification is to be defined by the data provider.

Type: STRING
Entry: Single field
Format: Free format
Example: `DATA_QUALITY = Daily HBR cell measurements analyzed with Linefit v8. Reference paper Senten et al., ACP 8, 3483-3508, 2008. NDACC qualification in progress.`

4.2.15. DATA_TEMPLATE

The global attribute `DATA_TEMPLATE` specifies information on applicable templates for reported data. This attribute is **mandatory** for datasets which follow a **GEOMS template** description, otherwise the use is **optional**.

Type: STRING
Entry: Single field
Format: Refer to Table 7.2.
Example: `DATA_TEMPLATE = GEOMS-TE-MWR-001`

4.2.16. DATA_PROCESSOR

The **optional** global attribute `DATA_PROCESSOR` specifies information on the data processor and the data processor version in a single element field. `DATA_PROCESSOR` should only be used for concise information on the name and version, while `DATA_DESCRIPTION` can be used to explain more details on the data processor.

Type: STRING
Entry: Single field
Format: [data processor name][data processor version]
Example: `DATA_PROCESSOR = GeoFitv1.0`
`DATA_DESCRIPTION = GeoFitv1.0, MIPAS special modes processor.`

4.3. Global File Attributes

The **global file attributes** provide identification, metadata parameter, formatting and archiving information on the data file. These attributes include the file name and generation date, the file access information, and the version of the metadata used.

4.3.1. FILE_NAME

The global attribute `FILE_NAME` contains the current data file name. This entry must be identical to the file name in the data archive. The `FILE_NAME` entry is always set in lower case, even if the field entries from which it is built are capitalized. Some data centers support HDF4, HDF5, and netCDF datasets. Correlative datasets in HDF4 format must have extension "`.hdf`", files in HDF5 format must have the extension "`.h5`" while files in netCDF format must have extension "`.nc`" (cf. Chapter 6).

Type:	STRING
Entry:	**Lower case**, underscore separated + `.[hdf|h5|nc]`
Format:	The `FILE_NAME` entry is constructed using seven underscore separated global attribute entries + the correct file extension (`.[hdf|h5|nc]`): The `DATA_DISCIPLINE_03` acquisition platform entry from Table 7.2. The `DATA_SOURCE` type entry from Table 7.2. The `AFFILIATION` acronym entry from Table 7.1 and a three-digit identifier. The `DATA_LOCATION` entry from Table 7.2. The `DATA_START_DATE` entry from Section 4.2.7 (ISO 8601 format). The `DATA_STOP_DATE` entry from Section 4.2.8 (ISO 8601 format). The `DATA_FILE_VERSION` entry from Section 4.2.9. The `.[hdf|h5|nc]` file extension (referring to the HDF4, HDF5, and netCDF file formats).
Example:	groundbased_ftir.hno3_ncar001_thule_20080305t151349z_20080824t221536z_001.hdf

4.3.2. FILE_GENERATION_DATE

The global attribute `FILE_GENERATION_DATE` specifies the date/time of generation of the current file. The `FILE_GENERATION_DATE` format is the ISO 8601 date/time described in Section 3.3.2.

Type:	STRING
Entry:	Single field
Format:	ISO 8601 date/time
Example:	`FILE_GENERATION_DATE = 20020420T112923Z`

4.3.3. FILE_ACCESS

The global attribute `FILE_ACCESS` has a multi-field character string entry referring to the file project association in the data archive. `FILE_ACCESS` is used to define the file's access rights through data center interfaces. For a list of allowed variables please refer to Table 7.3.

Type:	STRING
Entry:	One or more semicolon-separated fields
Format:	project1;project2;project3;…
Example:	`FILE_ACCESS = AVDC;EVDC;NDACC`

4.3.4. FILE_PROJECT_ID

The **optional** global attribute `FILE_PROJECT_ID` is an additional multi-field string entry to the `FILE_ACCESS` attribute in Section 4.3.3 defining custom project identification codes. These project identification codes are defined by and specific to campaigns or individual projects that have set them. For example, for the Envisat validation only, one Envisat Cal/Val `FILE_PROJECT_ID` field is **mandatory**, namely the Announcement of Opportunity (AO) responsible for providing the data.

Type: STRING
Entry: Empty, one or more semicolon-separated fields
Format: [id1;id2;id3;…]
Example: `FILE_PROJECT_ID = AOID126`
for Envisat Cal/Val AO 126

4.3.5. FILE_ASSOCIATION

The **optional** global attribute `FILE_ASSOCIATION` has a multi-field character string entry defining the file's other associations or funding programs such as National Programs, networks, special campaigns, or funding programs. This variable **shall not regulate** access to files on data center's servers. `FILE_ACCESS` shall be used instead.

Type: STRING
Entry: Single field or multiple semicolon-separated fields
Format: Free format
Example: `FILE_ASSOCIATION = Norwegian Research Council`

4.3.6. FILE_META_VERSION

The global attribute `FILE_META_VERSION` indicates the version of the metadata definitions used in the data file and the tool name used to generate the current HDF or netCDF data file. For a list of allowed variables please refer to Table 7.3.

Type: STRING
Entry: Two semicolon-separated fields
Format: nnRddd;tool name
Example: `FILE_META_VERSION = 04R001;IDLCR8HDF`

4.3.7. FILE_DOI

The global attribute `FILE_DOI` indicates the use of a digital object identifier. The field is currently mandatory, but with empty variable value. A specification of the DOI syntax will be given in a future version of GEOMS.

Type: STRING

5. Variable Attributes

Unlike the global attributes, the variable attributes are the specific metadata for a single variable. Each variable listed in the global attribute DATA_VARIABLES (cf. Section 4.2.6.) **must** be accompanied by a complete set of associated variable attributes. The variable attributes are called the **variable description attributes** (cf. Section 5.1) whose names begin with "VAR_". Tables 7.4 gives an overview of all variable attributes. All attributes marked with an "**x**" in the "Req" (Requirement) column in Table 7.4 are **mandatory** and must have an entry. For the ones marked with an "**o**", the attribute is **optional** and may be omitted. Hereinafter, all attributes are mandatory unless mentioned otherwise explicitly.

5.1. Variable Description Attributes

5.1.1. VAR_NAME

The variable attribute VAR_NAME identifies the data parameter. The entry must be one of the fields reported in the global attribute DATA_VARIABLES in Section 4.2.6.

Type:	STRING
Entry:	Single field constructed according to the format described in Section 4.2.6.
Format:	Variable name[_Variable mode][_Variable descriptor]
Example:	`VAR_NAME = O3.PARTIAL.PRESSURE_INSITU`

5.1.2. VAR_DESCRIPTION

The variable attribute VAR_DESCRIPTION provides a brief description of the variable. This entry should clearly identify the variable's meaning, either explicitly, or by reference to a readily accessible reference document.

Type:	STRING
Entry:	Single field
Format:	Free format
Example 1:	`VAR_DESCRIPTION = In-situ ozone partial pressure measured by ECC ozonesondes.`
Example 2:	`VAR_DESCRIPTION = ECC ozone measurements as described by Komhyr (1964, 1969).`

5.1.3. VAR_NOTES

The **optional** variable attribute VAR_NOTES conveys any additional information pertinent to the reported variable.

Type:	STRING
Entry:	Single
Format:	Free format

Example 1: `VAR_NOTES = Applied Komhyr (1986) ozonesonde pump correction table.`
Example 2: `VAR_NOTES = Retrieval SFIT2; spectroscopic database HITRAN2K.`

5.1.4. VAR_SIZE

The variable attribute `VAR_SIZE` has a multiple field entry containing the sizes of each dimension of the variable. In the following example 2, the dependent variable is reported in four independent dimensions (time, x, y, z) on a grid of 10*2*3*4 nodes. For `VAR_DATA_TYPE = STRING` (cf. Section 5.1.6) the string length does not count as a dimension.

Type: INTEGER(s)
Entry: n semicolon-separated fields, one field per dimension
Format: Integer1;Integer2;Integer3;…
Example 1: `VAR_SIZE= 2376`
Example 2: `VAR_SIZE= 10;2;3;4`

5.1.5. VAR_DEPEND

The variable attribute `VAR_DEPEND` lists all independent variables on which the current variable depends. The number of independent variables listed must be equal to the number of semicolon-separated fields in `VAR_SIZE`, and the order in which the variables are listed must correspond to the order in which their sizes are given in `VAR_SIZE`.

Type: STRING
Entry: n semicolon-separated fields, with n equal to number of semicolon-separated fields in `VAR_SIZE`
Format: Appropriate subset of the `DATA_VARIABLES` (cf. Section 4.2.6) entry
Example 1: `VAR_DEPEND = DATETIME`
Example 2: `VAR_DEPEND = DATETIME;LONGITUDE;LATITUDE;ALTITUDE`
Example 3: `VAR_DEPEND = CONSTANT`

Constants must have the following attributes:

```
VAR_DEPEND = CONSTANT
VAR_SIZE = 1
```

An **axis variable** (`DATETIME`, `LATITUDE`, `LONGITUDE`, and e.g. `ALTITUDE`), which is by definition independent, must use its own variable name as a dependent variable:

```
VAR_NAME = DATETIME
VAR_DEPEND = DATETIME
VAR_SIZE = n
```

Independent variables, which are **not axis variables** must have the following attributes:

```
VAR_DEPEND = INDEPENDENT
VAR_SIZE = n
```

Multiple occurrences of a variable in `VAR_DEPEND` are allowed.

When using older versions of HDF4 (prior to HDF4.2r2) some additional attention is drawn on the length of the name parameter saved with an SDS and its reference through `VAR_DEPEND`. Please see full discussion in Section 6.1.2.

Four different cases for `VAR_DEPEND` values for an `ALTITUDE` variable. Only in the last two cases the `ALTITUDE` variable is an **axis variable**:
All measurements are taken at a constant single height:

```
VAR_DEPEND = CONSTANT
```

Measurements are taken at a single height and this height differs over time:

```
VAR_DEPEND = DATETIME
```

Measurements have a height axis and the height grid is constant within the file:

```
VAR_DEPEND = ALTITUDE
```

Measurements have a height axis and the height grid differs over time:

```
VAR_DEPEND = DATETIME;ALTITUDE
```

5.1.6. VAR_DATA_TYPE

The variable attribute `VAR_DATA_TYPE` specifies the binary representation (“numerical” type) of the data associated with the variable. The used floating-point representations follow the IEEE 754 standard for floating-point arithmetic.

Type:	STRING
Entry:	Single
Format:	Refer to Table 7.4
Example:	`VAR_DATA_TYPE = REAL`

5.1.7. VAR_UNITS

The variable attribute `VAR_UNITS` specifies the **actual** units in which the variable data are reported in the current data file. The allowed units are **case sensitive** combinations of unit prefixes and units given in Tables 7.4 and 7.5. The unit definitions are compliant to the Unidata **UDUNITS** package [13] and have been extended by **GEOMS aliases** where appropriate (cf. Table 7.5). The prefix of a unit is concatenated with the unit, multiple units (multiplicative) are separated by spaces. Powers of units (integer) are concatenated with the unit; for example centimeter squared becomes `cm2`. For example, if the total column of NO_2 is equal to 2.7×10^{15} molecules per square centimeter, then you can report it either as:

`2.7` *with* `VAR_UNITS = Pmolec cm-2`

or

`2.7E15` *with* `VAR_UNITS = molec cm-2`

String data types must have a corresponding attribute value of an empty string:

```
VAR_DATA_TYPE = STRING
VAR_UNITS = [empty]
```

Dimensionless variables which do not report a unit must have an attribute value of 1, e.g.:

```
VAR_NAME = UV.INDEX
VAR_UNITS = 1
```

Type: STRING
Entry: single field, case sensitive
Format: Combination of entries in Tables 7.4 and 7.5 (brackets cannot be used)
Example 1: `VAR_UNITS = mPa`
Example 2: `VAR_UNITS = Pmolec cm-2`
Example 3: `VAR_UNITS = molec cm-2`

5.1.8. VAR_SI_CONVERSION

The variable attribute `VAR_SI_CONVERSION` provides the formula used to convert the data specified in `VAR_UNITS` to the equivalent data in the corresponding "base units"; the latter are identified in Table 7.5. The units conversion is compliant to the Unidata **UDUNITS** package [13] and has been extended by **GEOMS aliases** where appropriate. This attribute aims at facilitating calculations and comparisons by automated tools, for example, for comparison of data of the same geophysical entity but reported in different units.

The conversion formula is specified as follows:

`VAR_UNITS` = Offset + Conversion factor × Base unit

Divisions and multiplications of units should be factored out such as to have the shortest possible string of base units, in which any base unit appears at most once. For example, for the units [`nm m-2`] the attribute `VAR_SI_CONVERSION = 0;1.0E-9;m-1`.

String data types must have a corresponding attribute value of an empty string:

```
VAR_DATA_TYPE = STRING
VAR_SI_CONVERSION = [empty]
```

Dimensionless variables use base unit 1, e.g.:

```
VAR_UNITS = ppmv
VAR_SI_CONVERSION = 0;1E-6;1
```

Type: STRING
Entry: Three semicolon-separated fields
Format: Offset;Conversion factor;Base unit
Example 1: `VAR_SI_CONVERSION = 0;1E-3;kg m-1 s-2` *for* `VAR_UNITS = mPa`
Example 2: `VAR_SI_CONVERSION = 273.15;1;K` *for* `VAR_UNITS = Celsius`
Example 3: `VAR_SI_CONVERSION = 0;1.0E-9;m-1` *for* `VAR_UNITS = nm m-2`

5.1.9. VAR_VALID_MIN

The variable attribute `VAR_VALID_MIN` indicates the valid minimum or detection limit of the data variable. The value of `VAR_VALID_MIN` must be reported in the units specified in `VAR_UNITS`.

String data types must have a corresponding attribute value of an empty string:

VAR_DATA_TYPE = STRING
VAR_VALID_MIN = [empty]

Type:	Equal to the ones specified in VAR_DATA_TYPE (cf. Section 5.1.6)
Entry:	Single
Format:	Consistent with the variable's data type and formatting
Example:	VAR_VALID_MIN = 0.0

5.1.10. VAR_VALID_MAX

The variable attribute VAR_VALID_MAX indicates the valid maximum or saturation limit of the data variable. The value of VAR_VALID_MAX must be reported in the units specified in VAR_UNITS.

String data types must have a corresponding attribute value of an empty string:
VAR_DATA_TYPE = STRING
VAR_VALID_MIN = [empty]

Type:	Equal to the ones specified in VAR_DATA_TYPE (cf. Section 5.1.6)
Entry:	Single
Format:	Consistent with the variable's data type and formatting
Example:	VAR_VALID_MAX = 35.0

5.1.11. VAR_FILL_VALUE

The variable attribute VAR_FILL_VALUE is a number or an empty string value inserted as a substitute data element, if a data element of a variable is reported as a **default** or **missing** value. For variables with numeric VAR_DATA_TYPE (cf. Section 5.1.6), the VAR_FILL_VALUE can take any value allowed by the numeric type. If VAR_FILL_VALUE is set to a value **within the range** spanned by VAR_VALID_MIN in Section 5.1.9 and VAR_VALID_MAX in Section 5.1.10 it is considered a **default** value. If VAR_FILL_VALUE is set **outside the range** spanned by VAR_VALID_MIN and VAR_VALID_MAX it is considered a **missing** value.

String data types must have a corresponding attribute value of an empty string:
VAR_DATA_TYPE = STRING
VAR_FILL_VALUE = [empty]

Type:	Equal to the ones specified in VAR_DATA_TYPE (cf. Section 5.1.6)
Entry:	Single
Format:	Consistent with the variable's VAR_DATA_TYPE (cf. Section 5.1.6)
Example 1:	VAR_VALID_MIN = 0.00 *and* VAR_VALID_MAX = 35.00 *then* VAR_FILL_VALUE = 99.99 *is acceptable*
Example 2:	VAR_VALID_MIN = 0.000 *and* VAR_VALID_MAX = 2.600E5 *then* VAR_FILL_VALUE = -5.000E6 *is acceptable*
Example 3:	VAR_DATA_TYPE = STRING *then* VAR_FILL_VALUE = [empty] *is acceptable*

6. Implementation

In recent years, the Hierarchical Data Format [8] has become the *de-facto* satellite data exchange format for the ESA and the NASA Earth observation missions. HDF was originally developed by National Center for Supercomputing Application (NCSA) and is currently supported by the non-profit *The HDF Group*. Next to HDF, another hierarchical data format, netCDF, is extensively used in Earth observation. The netCDF format is maintained by Unidata [9]. To facilitate the validation of space borne measurements by correlative data from heterogeneous sources, the use of a common file format becomes a necessity. The metadata guidelines defined in this document are implemented using the HDF4, HDF5 or netCDF file formats, but are not limited to these formats. Special care must be given to the submission of files to GEOMS supporting data centers. Some data centers may not have native support to all discussed data formats.

6.1. HDF4

The recommended metadata implementation using the HDF4 file format uses the HDF's SDS (Scientific Data Structure) data model. The SDS allow for the efficient implementation of the global attributes from Chapter 4 and up to 64 variables of rank 8 with their corresponding variable attributes from Chapter 5.

6.1.1. Rules for generating GEOMS HDF4 files

- Files must have the file extension "`.hdf`".
- Only the use of the SD interface is allowed.
- Each dataset has a unique identifier in the file.
- Global attributes should be stored as SD file attributes.
- Variable attributes should be stored as attributes to the SDS.
- Storing dimension names is not allowed.
- HDF4 allows for reporting of pre-defined attributes: `long_name`, `units`, `format`, `coordsys`, `valid_range`, `_FillValue`, `scale_factor`, `scale_factor_err`, `add_offset`, `add_offset_err`, and `calibrated_nt`.
    - No scaling and/or off-setting of datasets is permitted therefore the last five listed attributes must not be reported.
    - The reporting of remaining pre-defined attributes is optional. The first four attributes are written using the SDsetdatastrs procedure; `valid_range` is written using the SDsetrange procedure; `_FillValue` is written using the SDsetfillvalue procedure.
    - If units, valid_range, and/or _FillValue are reported the values must exactly match the respective variable attribute values `VAR_UNITS`, `VAR_VALID_MIN`, `VAR_VALID_MAX` and `VAR_FILL_VALUE`.

6.1.2. HDF4.2r1 and prior

Earlier versions of HDF4 (4.2r1 and prior) have a 63-character limitation on the length of the name parameter saved with the SDS. Any name longer than this is truncated. For AVDC a work around in the **idlcr8 suite** was achieved by reading/writing the `VAR_NAME` attribute rather than the SDS name parameter.

Further, earlier versions of HDF (4.2r1 and prior) contain a bug described as follows in the HDF4.2r2 release notes:
"When a dimension has the same name as an SDS, depending on the order in which they were created, either the SDS or the dimension will be corrupted if certain operations occur, such as a `Sdsetdimscale` *or* `Sdsetattr` *call to the dimension."*

NASA's and ESA's data centers' HDF generation software such as **idlcr8hdf** and **asc2hdf** use the operation HDF_SD_DIMSET (or Fortran equivalent) to save information on any non `INDEPENDENT/CONSTANT` `VAR_DEPEND` value(s). By definition the `VAR_DEPEND` value(s) always reference a previously declared SDS. Consequently the given dimension parameter has the same name as an already existing SDS. Because of a bug prior to HDF4.2r2, this has meant that the dimension information has not been stored in a GEOMS HDF4 file as it should have been. This did not corrupt the HDF file in a way to make it unreadable, as the `VAR_DEPEND` values always referenced a previously declared (and saved) variable name, so the SDS remained uncorrupted. By upgrading to the HDF4.2r3 library the layout of the HDF file changed as the dimension variable name information is now referenced correctly in the file. Referencing `VAR_DEPEND` values as dimension parameters is discontinued. AVDC's HDF5 files already work this way.

6.2. HDF5

The HDF5 type file is conceptually related to HDF4, but it uses a completely new format and library [8]. HDF5 has a simpler data model that includes two primary objects: datasets and groups. For GEOMS, global attributes and datasets are written to the root group and the variable attributes are associated with the datasets. HDF5 does not have pre-defined SDS attributes. The HDF Group provides utilities that can convert HDF4 files to HDF5, and *vice versa*, but they are generic in nature.

6.2.1. Rules for generating GEOMS HDF5 files

- Files must have the file extension "`.h5`".
- VGroups are not allowed.
- Links are not allowed.
- Global attributes should be stored as file attributes.
- Variable attributes should be stored as attributes to the dataset.
- Integer data should be stored using the `H5T_INTEGER` class.

- Floating point data should be stored as either H5T_NATIVE_FLOAT or H5T_NATIVE_DOUBLE.

- String data should be stored using the H5T_STRING class. Only fixed sized strings are allowed for attributes and dataset values.

- No other types are allowed for datasets (e.g. no H5T_ENUM, H5T_COMPOUND, H5T_TIME, H5T_BITFIELD, H5T_OPAQUE, H5T_REFERENCE, H5T_ARRAY, H5T_VLEN, H5T_NO_CLASS, or H5T_NCLASSES).

6.3. NetCDF

The GEOMS standard can be applied to the netCDF classic, netCDF 64-bit offset, and netCDF-4 versions of the netCDF data format. When using netCDF-4, data should be stored using the netCDF4 Classic Model Format [15].

6.3.1. Rules for generating GEOMS netCDF files

- Files must have the file extension ".nc".

- 64-bit offset mode is only to be enabled for files where this is required due to the file size.

- GEOMS global attributes map to netCDF global attributes.

- GEOMS variables and variable attributes map to netCDF variables and variable attributes.

- The use of add_offset and scale_factor is not allowed.

- The name of a dimension is based on the entry in VAR_DEPEND for that specific dimension. This means that:
  - For a dimension with an axis variable, the dimension name is equal to name of the variable (not VAR_NAME, but the actual variable name, in case they differ) for which the value of its VAR_NAME attribute equals the entry in VAR_DEPEND.
  - For an independent dimension, the dimension name starts with INDEPENDENT (which may be lower or uppercase) and is followed by a string of choice in order to distinguish between independent dimensions of different length. A recommended approach is to use an underscore followed by the length of the dimension as a postfix. For example: INDEPENDENT_12 for an independent dimension of length 12.

7. Attribute values, templates and examples

The following list of Tables gives an overview of GEOMS attribute values (GEOMS-AV), templates (GEOMS-TE) and examples (GEOMS-EX) to this metadata standard document. GEOMS-AV, GEOMS-TE and GEOMS-EX reflect the current agreed set of vocabulary, templates and examples, which complement the metadata standard.

Not all variables have links to online resources, as detailed information is either hidden to the users and needs verification by the data centers, or fields have free format and thus do not have a controlled vocabulary. Note, some variables are **mandatory** "**x**" and some are **optional** "**o**", where the latter can be omitted completely in the reported file. Some **mandatory** variables allow free formatting, where even empty entries are acceptable. A variable attribute may generally be **optional**, but the use may be **mandatory** within the context of **GEOMS templates** "**o/x**". In general, a GEOMS template states, which metadata attributes and data variables **must be reported**.

Files which are submitted to a GEOMS compliant data center go through a quality assurance process and will be tested against this metadata standard.

Table 7.1 Global originator attributes in GEOMS-AV.

Global Attributes: Originator Attributes		
Name	***Req***	***Reference***
PI_NAME	x	cf. Section 4.1.1
PI_AFFILIATION	x	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=AFFILIATION
PI_ADDRESS	x	cf. Section 4.1.3
PI_EMAIL	x	cf. Section 4.1.4
DO_NAME	x	cf. Section 4.1.5
DO_AFFILIATION	x	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=AFFILIATION
DO_ADDRESS	x	cf. Section 4.1.7
DO_EMAIL	x	cf. Section 4.1.8
DS_NAME	x	cf. Section 4.1.9
DS_AFFILIATION	x	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=AFFILIATION
DS_ADDRESS	x	cf. Section 4.1.11
DS_EMAIL	x	cf. Section 4.1.12

Table 7.2 Global dataset attributes in GEOMS-AV.

Global Attributes: Dataset Attributes		
Name	***Req***	***Reference***
DATA_DESCRIPTION	o	cf. Section 4.2.1
DATA_DISCIPLINE	x	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=DATA_DISCIPLINE_01 http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=DATA_DISCIPLINE_02 http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=DATA_DISCIPLINE_03
DATA_GROUP	x	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=DATA_GROUP_01 http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=DATA_GROUP_02
DATA_LOCATION	x	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=DATA_LOCATION
DATA_SOURCE	x	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=DATA_SOURCE
DATA_VARIABLES	x	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=DATA_VARIABLES_01 http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=DATA_VARIABLES_02 http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=DATA_VARIABLES_03
DATA_START_DATE	x	cf. Section 4.2.7
DATA_STOP_DATE	x	cf. Section 4.2.8
DATA_FILE_VERSION	x	cf. Section 4.2.9
DATA_MODIFICATIONS	o	cf. Section 4.2.10
DATA_CAVEATS	o	cf. Section 4.2.11
DATA_RULES_OF_USE	o	cf. Section 4.2.12
DATA_ACKNOWLEDGEMENT	o	cf. Section 4.2.13
DATA_QUALITY	o/x	cf. Section 4.2.14
DATA_TEMPLATE	o/x	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE
DATA_PROCESSOR	o	cf. Section 4.2.16

Table 7.3 Global file attributes in GEOMS-AV.

Global Attributes: File Attributes		
Name	***Req***	***Reference***
FILE_NAME	x	cf. Section 4.3.1
FILE_GENERATION_DATE	x	cf. Section 4.3.2
FILE_ACCESS	x	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=FILE_ACCESS
FILE_PROJECT_ID	x	cf. Section 4.3.4
FILE_ASSOCIATION	o	cf. Section 4.3.5
FILE_META_VERSION	x	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=FILE_META_VERSION
FILE_DOI	x	cf. Section 4.3.7

Table 7.4 Variable description attributes in GEOMS-AV.

Variable Attributes: Variable Description Attributes		
Name	***Req***	***Reference***
VAR_NAME	x	cf. Section 5.1.1
VAR_DESCRIPTION	x	cf. Section 5.1.2
VAR_NOTES	o	cf. Section 5.1.3
VAR_SIZE	x	cf. Section 5.1.4
VAR_DEPEND	x	cf. Section 5.1.5
VAR_DATA_TYPE	x	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=VAR_DATA_TYPE
VAR_UNITS	x	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=VAR_UNITS
VAR_SI_CONVERSION	x	cf. Section 5.1.8
VAR_VALID_MIN	x	cf. Section 5.1.9
VAR_VALID_MAX	x	cf. Section 5.1.10
VAR_FILL_VALUE	x	cf. Section 5.1.11

Table 7.5 Additional extensions to GEOMS-AV.

Additional references	
Name	***Reference***
UNITS_PREFIX	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=UNITS_PREFIX
UDUNITS alias	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-UDUNITS

Table 7.6 Examples of GEOMS.

Example references	
Name	***Reference***
Global Originator Attributes	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-EX-GOA
Global Dataset Attributes	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-EX-GDA
Global File Attributes	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-EX-GFA
Variable Description Attributes	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-EX-VDA

Table 7.7 NDACC working group on balloon data templates (GEOMS-TE-O3SONDE) and examples (GEOMS-EX-O3SONDE).

NDACC WG Balloon template references	
Name	***Reference***
DATA_SOURCE	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-O3SONDE-DS
Common template	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-O3SONDE
Global Attributes Example	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-O3SONDE-GA
Variable Attributes Example	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-O3SONDE-VA

Table 7.8 NDACC working group on LIDAR data templates (GEOMS-TE-LIDAR) and examples (GEOMS-EX-LIDAR).

NDACC WG LIDAR template references	
Name	***Reference***
DATA_SOURCE	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-LIDAR-DS
Common template	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-LIDAR
Ozone	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-LIDAR-Ozone-VA
Temperature	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-LIDAR-Temperature-VA
Water Vapor Raman	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-LIDAR-WaterVaporRaman-VA
Water Vapor DIAL	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-LIDAR-WaterVaporDIAL-VA
Aerosol	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-LIDAR-Aerosol-VA
Global Attributes Example	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-EX-LIDAR-GA
Variable Attributes Example	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-EX-LIDAR-VA

Table 7.9 NDACC working group on MWR data templates (GEOMS-TE-MWR) and examples (GEOMS-EX-MWR).

NDACC WG MWR template references	
Name	***Reference***
DATA_SOURCE	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-MWR-DS
Common template	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-MWR
Global Attributes Example	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-EX-MWR-GA
Variable Attributes Example	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-EX-MWR-VA

Table 7.10 NDACC working group on FTIR data templates (GEOMS-TE-FTIR) and examples (GEOMS-EX-FTIR).

NDACC WG FTIR template references	
Name	***Reference***
DATA_SOURCE	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-FTIR-DS
Common template	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-FTIR
Global Attributes Example	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-EX-FTIR-GA
Variable Attributes Example	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-EX-FTIR-VA

Table 7.11 Radio Occultation data templates (GEOMS-TE-RO) and examples (GEOMS-EX-RO).

RO references	
Name	***Reference***
DATA_SOURCE	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-RO-DS
Common template	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-RO
Global Attributes Example	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-RO-GA
Variable Attributes Example	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-RO-VA

Table 7.12 UVVIS DOAS (GEOMS-TE-UVVIS.DOAS) and examples (GEOMS-EX-UVVIS.DOAS).

RO references	
Name	***Reference***
DATA_SOURCE	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-UVVIS.DOAS-DS
Common template	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-UVVIS.DOAS
Global Attributes Example	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-UVVIS.DOAS-GA
Variable Attributes Example	http://avdc.gsfc.nasa.gov/GEOMS.php?GEOMS=GEOMS-TE-UVVIS.DOAS-VA

A. Code Examples

A. Calculation of MJD2K

For a given `YYYY`, `MM`, `DD`, `hh`, `mm`, `ss`, `[and ms]`:

Step 1: Calculate the Julian date

```
IF ( MM > 2 ) THEN
     y = DOUBLE(YYYY)
     m = DOUBLE(MM - 3)
     d = DOUBLE(DD)
ELSE BEGIN
     y = DOUBLE(YYYY - 1)
     m = DOUBLE(MM + 9)
     d = DOUBLE(DD)
ENDELSE

 j = INTEGER(365.25×(y+4712.0)) + INTEGER(30.6×m+0.5) + 58.5 + d
```

Step 2: Check for Julian or Gregorian calendar

```
IF ( j < 2299159.5 ) THEN           if Julian calendar
     jd = j
ELSE                                if Gregorian calendar
     gn = 38.0 - INTEGER(3.0×INTEGER(49.0 + y/100.0)/4.0)
     jd = j + gn
ENDELSE
```

Step 3: Calculate day fraction

for [s] resolution:

```
df = (hh×3600.0 + mm×60.0 + ss)/86400.0
```

for [ms] resolution:

$df = (hh\times3.6\times10^{6} + mm\times6.0\times10^{4} + ss\times10^{3} + ms)/8.64\times10^{7}$

Step 4: Calculate MJD2000

```
mjd2000 = jd + df – 2451544.5
```

Example:

2002/04/20 at 11:29:23UTC   MJD2000 = 840.478738

For reasons of **compatibility to UDUNITS** [13], `MJD2K` is used as a short form for `MJD2000` in GEOMS data variables.

B. Acknowledgments

Acknowledgments go to Mr. Bart Quaghebeur and Dr. Daniel Heynderickx (BIRA-IASB) for demonstrating the usefulness of a modern data format; to Ms. Françoise Pinsard and Dr. Phillippe Keckut of CNRS/SA, Dr. K.-H. Fricke of the Bonn University, Drs. H. Siegel and T. Ohde of the Institut für Ostseeforschung Warnemünde, and Dr. Thierry Leblanc of JPL for their patience and valuable contributions to these guidelines. We thank Michael Yan of NASA/GSFC for his reviews of this document.

The community participating in the first and second Envisat Cal/Val rehearsals as well as participants of various Aura validation campaigns have provided very valuable feedback on a multitude of technical and scientific matters, without which this document could not have been completed.

Funding has been provided by EC COSE (ENV4-CT98-0750), the ESA Envisat Cal/Val database project (ESTEC Contract 14419/00/NL/SF), by the NASA contract NAS5-01008, by the NASA GEST Work Activity 3657, and by the GECA project.

C. Acronyms

AATSR	Advanced Along Track Scanning Radiometer
AO	Announcement of Opportunity
ASCII	American Standard Code for Information Interchange
Aura	NASA's Earth Observing System Aura satellite
AVDC	Aura Validation Data Center
AVK	Averaging Kernel
BIRA	Belgisch Instituut voor Ruimte-Aëronomie
CEOS	Committee on Earth Observation Satellites
COSE	Compilation of atmospheric Observations in Support of Satellite measurements over Europe
DC	Data Center
DO	Data Originator
DS	Data Submitter
DU	Data User
DIAL	Differential Absorption LIDAR
DOI	Digital Object Identifier
EC	The European Commission
Envisat	ESA's Environmental Satellite
EOS	Earth Observing System
ESA	European Space Agency
EVDC	Envisat Validation Data Center
FTIR	Fourier Transform InfraRed
GECA	Generic Environment for Calibration/Validation Analysis
GEOMS	Generic Earth Observation Metadata Standard
GEOMS-AV	GEOMS Attribute Value
GEOMS-EX	GEOMS Example
GEOMS-TE	GEOMS Template
GA	Global Attribute
GOMOS	Global Ozone Monitoring by Occultation of Stars
GSFC	Goddard Space Flight Center
HDF	Hierarchical Data Format
HIRDLS	High Resolution Dynamic Limb Sounder

IASB	Institut d'Aéronomie Spatiale de Belgique
ISO	International Organization for Standardization
LIDAR	LIght Detection And Ranging
MERIS	MEdium Resolution Imaging Spectrometer
MIPAS	Michelson Interferometer for Passive Atmospheric Sounding
MJD2000	Modified Julian Date 2000
MJD2K	Modified Julian Date 2000 short form
MLS	Microwave Limb Sounder
MWR	Microwave Radiometer
NADIR	NILU's Atmospheric Database for Interactive Retrieval
NASA	National Aeronautical and Space Administration
NCSA	National Center for Supercomputing Applications
NDACC	Network for the Detection of Atmospheric Composition Change
netCDF	network Common Data Form
NILU	Norwegian Institute for Air Research
NIWA	National Institute of Water and Atmospheric Research
OMI	Ozone Monitoring Instrument
PI	Principle Investigator
QA	Quality Assurance
QC	Quality Control
SCIAMACHY	SCanning Imaging Absorption SpectroMeter for Atmospheric CHartographY
SDS	HDF4 Scientific Data Sets
TES	Tropospheric Emission Spectrometer
VA	Variable Attribute

References

[1] Bojkov, B.R., De Mazière, M., and Koopman, R.M., Generic metadata guidelines on atmospheric and oceanographic datasets for the Envisat Calibration and Validation Project, (2002): http://avdc.gsfc.nasa.gov/PDF/GEOMS/Envisat_metadata_guideline_ver01r001_20020423.pdf.

[2] Bojkov, B.R., Boyd, I., De Mazière, M., and Koopman, R.M., Addendum to the "Generic metadata guidelines on atmospheric and oceanographic datasets for the Envisat Calibration and Validation Project" as implemented by the Aura Validation Data Center (AVDC), (2006): http://avdc.gsfc.nasa.gov/PDF/GEOMS/AVDC_file_format_addendum_20060831.pdf.

[3] De Mazière, M., Van Roozendael, M., Bojkov, B.R., de la Noë, J., Mahieu, E., and Neuber, R., "Archiving of atmospheric data: Data formats and database", in IRS 2000: Current Problems in Atmospheric Radiation, W. L. Smith and Yu. M. Timofeyev (Eds.). A. Deepak Publishing, Hampton, Virginia. pp. 1019 – 1022, (2002).

[4] ESA, "Envisat - Caring for the Earth", European Space Agency, Paris, France, (2001).

[5] Schoeberl, M.R., Douglass, A.R., Hilsenrath, E., Bhartia, P.K., Beer, R., Waters, J.W., Gunson, M.R., Froidevaux, L., Gille, J.C., Barnett, J.J., Levelt, P.F., and DeCola, P., "Overview of the EOS aura mission", in Geoscience and Remote Sensing, IEEE Transactions on, Volume 44, Issue 5, pp. 1066 – 1074, (2006), doi:10.1109/TGRS.2005.861950.

[6] Bojkov, B.R., Koopman, R.M., and De Mazière, M., "The Envisat Cal/Val Data Centre", presented at the NDSC 2001 Symposium – Celebrating 10 years of atmospheric research. September 24 – 27, 2001, Arcachon, France.

[7] De Mazière, M., "Final Report of the EC-COSE Project (contract ENV4-CT98-0750)", BIRA-IASB, Brussels, Belgium, (2001).

[8] The HDF Group – HDF home page: http://www.hdfgroup.org/.

[9] The Unidata netCDF data format: http://www.unidata.ucar.edu/software/netcdf/.

[10] Koopman, R.M., Bojkov, B.R. and De Mazière, M, "The ENVISAT Cal/Val Data center", in Proceedings of the 2002 IEEE Geoscience and Remote Sensing Symposium (IGARSS) and the 24th Canadian Symposium on Remote Sensing, Toronto, Ontario. Published on CD-ROM, (2002).

[11] ISO, "ISO 7-bit coded character set for information interchange", ISO/IEC 646:1991, International Organization for Standardization (ISO), Geneva, Switzerland, (1991).

[12] ISO, "Representation of Dates and Times", ISO 8601:1988, International Organization for Standardization (ISO), Geneva, Switzerland, (1988).

[13] The Unidata UDUNITS Package: http://www.unidata.ucar.edu/software/udunits/.

[14] ISO, "English country names and code elements – ISO 3166-1", International Organization for

Standardization (ISO), Geneva, Switzerland: http://www.iso.org/iso/list-en1-semic-3.txt.

[15] The NetCDF-4 Classic Model Format: http://www.unidata.ucar.edu/software/netcdf/docs/netcdf/NetCDF_002d4-Classic-Model-Format.html.