Dataset Interoperability Recommendations for Earth Science

Status of this RFC

This RFC provides information to the NASA Earth Science community. This RFC does not specify an Earth Science Data Systems (ESDS) standard. Distribution of this memo is unlimited.

Change Explanation

Changes from Version 1.2:
Add row describing "coordinates" to end of first table in section 2.2.

Changes from Version 1.1:
Change category from "Technical Note" to "Suggested Practice"

Changes from Version 1.0:
Recommendation 2.2: Punctuation change in summary sentence.
Recommendation 2.8: Re-word summary sentence to better reflect the recommendation.
Recommendation 2.12: Removed the term "non-model" in the title and summary sentence to better reflect the recommendation and to correct an earlier editing oversight.

Copyright Notice

Abstract

This document contains a series of recommendations made by the NASA Earth Science Data System (ESDS) Dataset Interoperability Working Group (DIWG) that are meant to increase and enhance the interoperability of Earth Science data product files. The DIWG recommendations herein embody best practices to reduce and bridge gaps between geoscience dataset formats widely used at NASA and elsewhere and to help ensure that Earth science datasets smoothly interoperate with each other regardless of their origin.

Table of Contents

1 Introduction

The Earth Science Data System Working Groups (ESDSWG) is a NASA organization established under the auspices of NASA Headquarters in 2004. The chartered role of the ESDSWG focuses on the exploration and development of recommendations derived from pertinent community insights of NASA's heterogeneous and distributed Earth science data systems.

The purpose of the Dataset Interoperability Working Group (DIWG) is to formulate, deliberate and make recommendations to help ensure that Earth Science datasets smoothly interoperate with each other regardless of their origin. The DIWG recommendations herein embody best practices to reduce and bridge gaps between geoscience dataset formats widely used at NASA and elsewhere, and to improve dataset compliance, discoverability, extensibility with relevant metadata conventions.

This document contains twelve specific recommendations made by the DIWG.

2 Recommendations

2.1 Maximize HDF5/netCDF4 interoperability via API accessibility

We recommend that Earth Science data product files in HDF5 be designed to maximize netCDF4 interoperability by making such HDF5 files accessible from the netCDF4 API to the extent that this is possible.

Background: NASA data products based on Earth Science observations are typically in HDF, trending to HDF5 [1], while Earth Science modelers generally prefer to produce data in netCDF, trending to netCDF4 [2]. It is not possible to make HDF4 files look exactly like netCDF3 files. On the other hand, netCDF4 is built on HDF5 (netCDF4 is essentially a subset of HDF5), and so it is possible to construct HDF5 files that are accessible from the netCDF4 API, which is a tremendous opportunity for interoperability. While using the netCDF4 API ensures this, the recommendation also provides guidance for those using the HDF5 API to ensure netCDF4 interoperability.

Recommendation Details:

A generic HDF5 format should be used in conjunction with netCDF4

-HDF5 files can be made nearly indistinguishable from netCDF4 files by adding dimension scales to the HDF5 files in a way that mimics netCDF shared dimensions [3].

-HDF4 has limitations that create difficulty making it netCDF-compliant. For this reason, we discourage use of HDF4 for new NASA Earth Science Data products and recommend use of the more flexible HDF5.

-HDF5 features that are not supported in netCDF4, either completely or at all, include:

- Fixed size string arrays
- Compound data types
- Datasets with more than one unlimited dimension
- Long double data types
- Soft links

These features should be avoided to enable interoperability between HDF5 and netCDF4.

-The Unidata Common Data Model (CDM) [4] attempts to merge netCDF, OPeNDAP and HDF5 data models. Use of these data model specifications will substantially increase interoperability and minimize data processing time and effort.

-If product developers prefer to use HDF-EOS5 [5], then their HDF-EOS5 products should be made netCDF4-compatible and CF-compliant.

Interoperability between NASA produced HDF5 products and netCDF4 APIs should be validated:

-The netCDF4 compatibility of HDF5 files can be tested by ncdump [6], ncks [7], the JPL Web-based Metadata Compliance Checker (MCC) [8], or HDF Product Designer (HPD) [9]. Note that the enhanced mode/group hierarchy may not be fully supported by some tools.

2.2 Include Basic CF Attributes

We recommend that, at minimum, the following basic Climate and Forecast (CF) Convention attributes be included in future NASA Earth Science data products where applicable.

Background: The Climate and Forecast (CF) Conventions [10] are widely employed guidelines for Earth Science data and metadata storage. Included in the conventions is a comprehensive list of metadata attributes that are available for use by dataset producers. Because the list of metadata attributes is so extensive, dataset producers are constantly struggling with which metadata attributes to attach to a variable.

Recommendation Details:

The following CF Convention attributes should be included in future NASA Earth Science data products where applicable – as they are necessary to describe any interoperable data file.

Attribute Name	Short Definition	Example	Notes
Conventions	CF version	CF-1.6	
units	A string that represents the units of measurement.	Kelvin	A variable with no units attribute is assumed to be unitless.
long_name	A descriptive name that indicates a variable's content.	Sea Surface Temperature	
standard_name*	A standard name that references a description of a variable's content in the standard name table of CF conventions.	latitude	*standard_name may not be applicable to many NASA HDF products, as some physical variable names of NASA satellites are not in the CF standard name table [11].
_FillValue	A value used to represent missing or undefined data.	-9999.0	Include _FillValue only if variable has missing values. Also _FillValue data type must equal the variable data type.
valid_min	Smallest valid value of a variable.	0	If applicable, must have either (valid_min, valid_max) or valid_range. According to the CF convention, the valid_min attribute should have the same type as the data variable. Thus, if the data variable is packed with scale_factor and add_offset, valid_min is also stored in packed form and must be unpacked: valid_min_value = scale_factor x valid_min + add_offset
valid_max	Largest valid value of a variable.	1	If applicable, must have either (valid_min, valid_max) or valid_range. According to the CF convention, the valid_max attribute should have the same type as the data variable. Thus, if the data variable is packed with scale_factor and add_offset, valid_max is also stored in packed form and must be unpacked: valid_max_value = scale_factor x valid_max + add_offset
valid_range	Smallest and largest valid values of a variable.	(0,1)	If applicable, must have either (valid_min, valid_max) or valid_range. According to the CF convention, the valid_range attribute should have the same type as the data

			variable. Thus, if the data variable is packed with scale_factor and add_offset valid_max is also stored in packed form and must be unpacked: valid_min_value = scale_factor x valid_range[0] + add_offset valid_max_value = scale_factor x valid_range[1] + add_offset
scale_factor	If present for a variable, the data are to be multiplied by this factor after the data are read by an application.	0.1	The unpacked final value is, per netCDF convention: Final_data_value = scale_factor x Raw_data_value + add_offset See also Recommendation 2.6 When to Employ Packing Attributes.
add_offset	If present for a variable, this number is to be added to the data after it is read by an application. If both scale_factor and add_offset attributes are present, the data are first scaled before the offset is added.	300	The unpacked final value is, per netCDF convention: Final_data_value = scale_factor x Raw_data_value + add_offset See also Recommendation 2.6 When to Employ Packing Attributes.
coordinates	Identifies a variable's coordinates.	"time latitude longitude"	Recommended when using multidimensional coordinate variables or a one-dimensional coordinate variable with a name that differs from its dimension's name. This helps geospatial tools identify spatio-temporal coordinates unambiguously for a variable. Makes the "bounds" attribute more useful. See also Recommendation 2.3 Use CF "bounds" attributes.

Because CF has special requirements for dimensional units and some coordinate values, when applicable, the **units** attribute values listed below should be used

Coordinate Variable	Unit Value	Examples
latitude	degrees_north	
longitude	degrees_east	
pressure	Pa or hPa	

height (depth)	meter (m) or kilometer (km)	
time	Seconds, minutes, hours, days, etc., since a specific starting point in time, often (but not always) representing a canonical time (e.g., 1 Jan 1970, TAI93, start of mission, etc.). Time is in ISO-8601 format.	seconds since 1992-10-08T15:15:42.5-6:00 days since 1970-01-01T00:00:00Z

2.3 Use CF “bounds” attributes

We recommend that spatio-temporal and other coordinate boundaries be specified by adding CF "bounds" attributes.

Recommendation Details:

The CF conventions are widely employed guidelines for Earth Science data and metadata storage. The purpose of the CF conventions is to require conforming datasets to contain sufficient metadata that they are self-describing in the following ways: Each variable in the file has an associated description of what it represents, including physical units if appropriate; and each value can be located in space (relative to Earth-based coordinates) and time. Thus, adhering to CF guidelines will increase completeness, consistency, and interoperability of conforming datasets.

CF conventions state: “When gridded data does not represent the point values of a field but instead represents some characteristic of the field within cells of finite ‘volume,’ a complete description of the variable should include metadata that describes the domain or extent of each cell, and the characteristic of the field that the cell values represent.”

Bounds are implemented by adding a “bounds" attribute to each applicable coordinate dimension, and the attribute specifies the name of the variable that contains the edges of the respective coordinate [12].

Example: Data representative of a time interval (rather than a specific time) might annotate the “time" coordinate with a “bounds" attribute with value “time_bounds.” The “time_bounds” variable would be a multi-dimensions array of the intervals for each value of “Time.”

Similar conventions apply to spatial and other coordinates.

2.4 Verify CF compliance

We recommend that CF compliance of NASA-distributed HDF/netCDF files be verified.

Background: The Climate and Forecast (CF) Conventions are widely employed guidelines for Earth Science data and metadata storage, which help tool developers find and/or interpret data values, coordinates, units, and measurements in data files. Thus, it is increasingly important to adhere to CF conventions in order to benefit from analysis tools, web services, and middleware that exploit them.

Recommendation Details:

We recommend using (freely-available) checkers, such as cfchecker [13], NCO [14], the JPL Web-based Metadata Compliance Checker (MCC) [8], or HDF Product Designer (HPD) [9], to understand and mitigate the CF compliance of files employing either the classic or enhanced data models.

2.5 Distinguish clearly between HDF and netCDF packing conventions

We recommend that datasets with non-netCDF packing be clearly distinguished from datasets that use the netCDF packing convention.

Background: Earth Science observers and modelers often employ a technique called “packing” (a.k.a. “scaling’) to make their product files smaller. "Packed" datasets must be correctly "unpacked" before they can be used properly. Confusingly, non-netCDF (e.g., HDF4_CAL [15]) and netCDF algorithms both store their parameters in attributes with the same or similar names – and unpacking one algorithm with the other will result in incorrect conversions. Many netCDF-based tools are equally unaware of the non-netCDF (e.g., HDF_CAL) packing cases and so interpret all readable data using the netCDF convention. Unfortunately, few users are aware that their datasets may be packed, and fewer know the details of the packing algorithm employed. This is an interoperability issue because it hampers data analysis performed on heterogeneous systems.

Recommendation Details:

-One widely used HDF4 "packing" convention uses the following "unpacking" equation: **unpacked = scale_factor x (packed - add_offset)**. We shall refer to this convention as "non-netCDF" .

-The standard netCDF "packing" convention uses the following "unpacking" equation: **unpacked = scale_factor x packed + add_offset**.

-To disambiguate the various packing conventions we recommend that two new attributes be included in NASA Earth Science data products, especially if something other than the netCDF convention is used:

- :packing_convention="non-netCDF"

- :packing_convention_description="unpacked = scale_factor x (packed - add_offset)"

or

- :packing_convention="netCDF"
- :packing_convention_description="unpacked = scale_factor x packed + add_offset"

-Future packing implementations should use scale_factor and add_offset only if these adhere to the netCDF packing convention.

2.6 When to employ packing attributes

We recommend that packing attributes (i.e., scale_factor and add_offset) be employed only when data are packed as integers.

Recommendation Details:

Packing refers to a lossy means of data compression that typically works by converting floating point data to an integer representation that requires fewer bytes for storage. The packing attributes “scale_factor” and “add_offset” are the netCDF (and CF) standard names for the parameters of the packing and unpacking algorithms. If “scale_factor” is 1.0 and “add_offset” is 0.0, the packed value and the unpacked value are identical, although their datatype (float or integer) may differ. Unfortunately, many datasets annotate floating point variables with the attributes, apparently for completeness, even though the variables have not been packed and remain as floating point values. Incorporating packing attributes on data that have not been packed is a misuse of the packing standard and it should be avoided. Data analysis software that encounters packing attributes on data that are not packed is liable to be confused and perform in unexpected ways. Packed data must be represented as integers, and only integer types should have packing attributes.

2.7 Mapping between ACDD and ISO

We recommend use of existing mapping between ACDD and ISO developed by ESIP.

Background: The ESIP Community supports a vast array of systems that are accessed and utilized by a diverse group of users. Historically, groups within the community have approached metadata differently in order to effectively describe their data. As a result, similar dialects have emerged to address specific user requirements. The multi-dialect approach described above hinders interoperability— as it results in different terminology being used to describe the same concepts. By clearly depicting fundamental documentation needs and concepts and mapping to them in the different dialects, confusion is minimized and interoperability is facilitated. Thus, demonstrating connections between dialects increases discoverability, accessibility, and reusability of data via consistent, compatible metadata.

ESIP is furthering the interoperability effort by describing the connections between fundamental concepts in dialects used throughout the ESIP Community – such that effective communication is maintained even when different metadata models are employed.

Recommendation Details:

-The Attribute Conventions for Data Discovery (ACDD) [16] identify important data discovery concepts and provide standard names for those concepts often used in netCDF files.

-These concepts also exist in ISO 19115 [17] and the standard mapping is implemented as part of the ncISO tool that is available as a stand-alone tool and as part of THREDDS [18].

-The mapping between ACDD and ISO developed by ESIP should be used to translate ACDD attributes to ISO-compliant XML.

We recommend continuing to endorse and support the ESIP effort toward Interoperability. Specifically, ESDIS should stay involved in the ESIP Documentation Cluster where these concepts will be implemented as standard objects (i.e. with groups) in the near future.

Group Structures in HDF5 and netCDF4 Files

In an HDF5 or netCDF4 file:

- A ***group*** contains a collection of objects, such as datasets, attributes, and/or other groups. Note that in an Earth Science data file, groups are usually defined by the type of data they contain; e.g., ancillary data, geolocation data, instrument data, science data, or metadata.
- A ***group structure*** is a group that has a specific structure defined by its dimensions (or lack thereof in the case of a metadata group). Below is a selection of common group structure types that appear in Earth Science data products. Note that each group structure is defined by its typical dimensions.

  1. **Swath Structure**: Time or along-track dimension, cross-swath dimension (parameter and vertical dimension are optional).
  2. **Nadir Profile Structure**: Time dimension, vertical dimension.
  3. **Grid Structure**: Time dimension, X dimension, Y dimension (parameter and vertical dimension are optional).
  4. **Zonal Structure**: Time dimension, latitude dimension (parameter and vertical dimension are optional).

5. **Metadata Structure** (e.g., ISO Metadata): No dimensions.

Earth Science data products usually include data-driven group structures, yet the group structures are highly individualized, as there are no general or comprehensive guidelines directing how these group structures should be assembled. This disparity introduces an interoperability issue that could be significantly minimized if some useful commonality were adopted for the group structures contained in Earth Science data products. Many of the following recommendations are designed to help standardize the formation of Grid Structures in Earth Science data products, thus increasing their interoperability and facilitating ease of use.

2.8 Make HDF5 files netCDF4-Compatible and CF-compliant within Groups

We recommend that all HDF5 Earth Science product files be made netCDF4-compatible and CF-compliant within groups.

Recommendation Details:

Compatibility with netCDF4

Unlike netCDF3, netCDF4 is based on HDF5 and thus allows for the creation of group structures. Therefore, it makes sense to create group structures in netCDF4 directly, or within HDF5 products that can be read through the netCDF4 API. This can be achieved by adding dimension datasets and dimension scales that follow the netCDF data model to the HDF5 products.

Example: A dimension named "Time":

1. When setting up the definitions, a dataset called "Time" is created at the root level with the required values.
2. The dataset at the root level is turned into a dimension scale via the H5DSset_scale function. This allows datasets below the root level to attach to the dimension dataset via the H5DSattach_scale function.

CF-Compliant Within Groups

The CF conventions are widely employed guidelines for Earth Science data and metadata storage. The purpose of the CF conventions is to require conforming datasets to contain sufficient metadata that they are self-describing in the following ways: Each variable in the file has an associated description of what it represents, including physical units if appropriate; and each value can be located in space (relative to Earth-based coordinates) and time. Thus, adhering to CF guidelines will increase completeness, consistency, and interoperability of conforming datasets.

Currently CF only applies to "flat files" with a single group, and not to files with multiple groups or hierarchical structures that typify modern NASA satellite datasets. Until CF is extended to apply to multiple groups, such NASA datasets can be most CF-compliant by following CF within each group.

To achieve the maximum CF-compliance within each group, we recommend the following:

1. Datasets in the group have the required CF attributes.

2. For the cases when horizontal space coordinates can be described with one-dimensional latitude and longitude arrays the following is recommend:

   1) Dimension datasets are located at the appropriate level for the group.

   2) Dimension datasets for the group have the appropriate CF attributes (e.g., no fill value).

   3) The appropriate dimension scales have been implemented for the group.

3. For the cases when horizontal space coordinates can be described only with two-dimensional latitude and longitude arrays, the CF "coordinate" attribute that consists of the coordinate dataset names must be used for each corresponding dataset. To avoid ambiguity and to take advantage of some popular CF tools (Panoply [19], etc.) that already support HDF5/netCDF-4 files with multiple groups, the following is recommended to handle the coordinates:

   1) For coordinate datasets that are not latitude and longitude, follow 1)-3) under 2. above.

   2) Make all dimensions associated with the 2-D latitude/longitude arrays pure netCDF dimensions by defining the NAME attribute of the dimension dataset to be "This is a netCDF dimension but not a netCDF variable."

   3) If only one pair of 2-D latitude/longitude arrays is needed for the whole file, then define them at the root level. This can make more CF tools visualize the physical HDF5 datasets.

   4) Use the absolute path of HDF5 datasets that store latitude and longitude for the coordinate attributes. For example, consider an HDF5 dataset "temperature" that is under the group g2, and whose parent group is g1 (i.e., float /g1/g2/temperature[Dim1][Dim2]). The two-dimensional latitude and longitude fields that describe this temperature field is under the group g1 (i.e., /g1/latitude[Dim1][Dim2], /g1/longitude[Dim1][Dim2]). One should define a coordinates attribute coordinates="/g1/latitude /g1/longitude".

2.9 Include time dimension in grid structured data

We recommend that datasets in grid structures include a Time dimension, even if Time is degenerate (i.e., includes only one value) for the cases when the entire grid has one time range or time stamp.

Recommendation Details:

A Time dimension is required for a single grid product file that contains many time intervals of daily, weekly, or monthly averages. In contrast, grid product files that are distributed as daily, weekly, or monthly granules that have one time range or stamp for the entire grid could be defined without a Time dimension because each file records the specific time interval being provided in both the file name and file-level attributes. We nevertheless recommend that a Time dimension be defined and used in all data fields that vary in time, regardless of whether multiple time slices are stored in the file. More specifically, we recommend that Time be defined as a record dimension, not a fixed-length dimension. This allows downstream users to more easily and efficiently aggregate data across separate files because HDF5 and netCDF4 storage geometries and APIs are designed to more easily extend record dimensions than fixed-length dimensions.

Unfortunately not all software understands multi-dimensional datasets with Time as one dimension. In such situations, the NCO ncwa [20] operator can be used to eliminate the Time dimension from an entire file with a single command.

2.10 Order dimensions to facilitate readability of grid structure datasets

We recommend that the dimensions in grid structure datasets be ordered in a manner that facilitates readability for the anticipated end users.

Recommendation Details:

The order of the dimensions in a group structure should be carefully considered, since this can have significant impact on the ease with which an end user can read the data in the group structure. While tools such as NCO's ncpdq [21] can re-order dataset dimensions, thereby permitting dataset designers to test the effects of their ordering choices against common access patterns, re-ordering large datasets can be time consuming and is best avoided.

Therefore, producers should design products in which the initial dimension order makes it easy for the most typical end users to read. This order is not necessarily the same as what may be simplest to produce.

The Aura File Format Guidelines [22] suggests that the dimensions in HDF-EOS files be ordered based on how rapidly the dimensions change, with the most rapidly changing dimension located first in Fortran arrays, e.g., (XDim,YDim,time), and last in C arrays, e.g., (time,YDim,XDim). Adhering to this method when using either the Fortran or C API results in identical ordering of values and efficient storage orders on disk in the HDF or netCDF product files. The COARDS conventions [23] suggest that dimensions in addition to space and time be added "to the left" of the space and time dimensions.

2.11 Consider "balanced" chunking for 3-D datasets in grid structures

We recommend that "balanced" chunking be considered for three-dimensional datasets in grid structures.

Recommendation Details:

If a dataset is exceptionally large, it is often more useful to break it up into manageable parts. This process is known as "chunking" and is used on data in datasets that are part of a grid structure. Exactly how the data chunking is done can greatly affect performance for the end user. Because the precise access pattern employed by the end user is usually unknown until the distributor analyzes sufficient requests to discern a pattern, it is difficult to determine the most effective way to chunk.

However, there are some common ways of applying chunking that have been broadly successful. Among them is a method known as "balanced chunking," which is chunking that balances access speeds for time-series and geographic cross-sections, the two most-common geometries of requested data.

For example, Unidata has an algorithm for balanced chunking [24]. This and other chunking algorithms are implemented in NCO's ncks [7] and described in greater detail in the NCO documentation. The HDF Group also provides some guidelines ([25] and [26]).

2.12 Include datum attributes for data in grid structures

We recommend that Horizontal and Vertical (as necessary) Datum attributes be included for data in grid structures.

Recommendation Details:

Locations on Earth are specified using coordinates, which are tuples of numbers that describe the horizontal and vertical distances from a fixed point, thereby pinpointing a particular place on the map at some level of precision. But knowing the coordinates is very different from being able to interpret them.

For example, the following set of coordinates marks a specific location the Earth:

(36.8, -121.5, 2000.0)

When those numbers are used in conjunction with a Datum [27], a reference point (or set of points) against which geospatial measurements are made, the numbers can be used to identify the location by their relative position to fixed points:

Coordinate System Type: Geographic
Units: Degrees, Degrees, Meters
Horizontal datum: WGS84
Vertical datum: EGM96

In addition, having reference information also helps in converting the coordinates to other representations. Thus, the capability of accurately interpreting coordinate information relies on the presence of Datum attributes. This recommendation ensures the Datum may be found in the data product file, so that users need not search for it in external sources such as READMEs.

3 References

1. HDF5 Data Model, File Format and Library – HDF5 1.6, https://earthdata.nasa.gov/standards/hdf5

2. netCDF-4/HDF5 File Format, https://earthdata.nasa.gov/standards/netcdf-4hdf5-file-format

3. HDF5 Dimension Scales - Part 2, http://www.unidata.ucar.edu/blogs/developer/en/entry/dimension_scale2

4. Unidata's Common Data Model Version 4, http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/CDM/

5. HDF-EOS5 Data Model, File Format and Library, https://earthdata.nasa.gov/standards/hdf-eos5

6. ncdump utility, http://www.unidata.ucar.edu/software/netcdf/docs/netcdf_utilities_guide.html#ncdump_guide

7. ncks netCDF Kitchen Sink, http://nco.sourceforge.net/nco.html#ncks-netCDF-Kitchen-Sink

8. JPL PODAAC, Metadata Compliance Checker, http://podaac-uat.jpl.nasa.gov/mcc/

9. HDF Product Designer, https://wiki.earthdata.nasa.gov/display/HPD/HDF+Product+Designer

10. netCDF Climate and Forecast (CF) Metadata Conventions, http://cfconventions.org/cf-conventions/v1.6.0/cf-conventions.html

11. netCDF Climate and Forecast (CF) Metadata Conventions - Standard Name, http://cfconventions.org/cf-conventions/v1.6.0/cf-conventions.html#standard-name

12. netCDF Climate and Forecast (CF) Metadata Conventions – Cell Boundaries, http://cfconventions.org/cf-conventions/v1.6.0/cf-conventions.html#cell-boundaries

13. CF-Convention Compliance Checker for netCDF Format,
http://puma.nerc.ac.uk/cgi-bin/cf-checker.pl

14. Checking CF-compliance,
http://nco.sf.net/nco.html#cf-compliance

15. HDF User's Guide, Section 3.10.6 Calibration Attributes,
http://www.hdfgroup.org/release4/doc/UsrGuide_html/UG_PDF.pdf

16. Attribute Conventions Dataset Discovery,
http://wiki.esipfed.org/index.php/Category:Attribute_Conventions_Dataset_Discovery

17. ISO 19115-1:2014, Geographic information -- Metadata -- Part 1: Fundamentals,
http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=53798

18. THREDDS Data Server 4.6,
http://www.unidata.ucar.edu/software/thredds/current/tds/TDS.html

19. Panoply,
http://www.giss.nasa.gov/tools/panoply

20. ncwa netCDF Weighted Averager,
http://nco.sourceforge.net/nco.html#ncwa-netCDF-Weighted-Averager

21. ncpdq netCDF Permute Dimensions Quickly,
http://nco.sourceforge.net/nco.html#ncpdq-netCDF-Permute-Dimensions-Quickly

22. Aura File Format for Satellite Atmospheric Chemistry Data,
https://earthdata.nasa.gov/standards/aura-file-format-satellite-atmospheric-chemistry-data

23. COARDS Conventions
http://ferret.wrc.noaa.gov/noaa_coop/coop_cdf_profile.html

24. Chunking Data: Choosing Shapes,
http://www.unidata.ucar.edu/blogs/developer/en/entry/chunking_data_choosing_shapes

25. netCDF4 Performance Report, Section 5.1.2,
https://www.hdfgroup.org/pubs/papers/2008-06_netcdf4_perf_report.pdf

26. Chunking in HDF5,
https://www.hdfgroup.org/HDF5/doc/Advanced/Chunking/index.html

27. Spatial References, Horizontal and Vertical Datum,
http://docs.osgearth.org/en/latest/user/spatialreference.html#horizontal-datum

4 Authors

2013 and 2014 Dataset Interoperability Working Group

Charles S. Zender and Peter J.T. Leonard, Technical Chairs, et al.

Charles S. Zender
University of California, Irvine
3200 Croul Hall
Irvine, CA 92697-3100

Peter J.T. Leonard
ADNET Systems, Inc.
7515 Mission Drive, Suite A100
Lanham, MD 20706

Edited by ESDIS Standards Office staff
eso-staff@lists.nasa.gov

Appendix A - Glossary

ACDD - Attribute Convention for Dataset Discovery

API - Application Program Interface

CDM - Unidata Common Data Model

CF - Climate and Forecast Metadata Conventions

COARDS - Cooperative Ocean/Atmospheric Research Data Service

DIWG - Dataset Interoperability Working Group

ESDIS - Earth Science Data and Information System

ESDS - Earth Science Data Systems

ESDSWG - Earth Science Data System Working Groups

ESIP - Federation of Earth Science Information Partners

ESO - ESDIS Standards Office

HDF - Hierarchical Data Format

HDF4 - Hierarchical Data Format, version 4

HDF5 - Hierarchical Data Format, version 5

HDF-EOS - Hierarchical Data Format - Earth Observing System

HDF-EOS5 - Hierarchical Data Format - Earth Observing System, version 5

HPD – HDF Product Designer

ISO - International Organization for Standardization

JPL - Jet Propulsion Laboratory

MCC - Metadata Compliance Checker

NASA - National Aeronautics and Space Administration

ncks - netCDF Kitchen Sink

NCO - netCDF Operator

ncpdq - netCDF Permute Dimensions Quickly

ncwa - netCDF Weighted Averager

netCDF - Network Common Data Form

netCDF3 - Network Common Data Form, version 3

netCDF4 - Network Common Data Form, version 4

OPenDAP - Open-source Project for a Network Data Access Protocol

RFC - Request For Comments

THREDDS - Thematic Real-time Environmental Distributed Data Services

XML - Extensible Markup Language