White Paper

Guide to Archiving CDF Files in PDS4

Revision 7, July 24, 2018

Todd King, Joseph Mafi

Overview

The Common Data Format (CDF) is widely used in the Heliophysics domain and for data obtained from ground based observatories. It is less commonly used in the planetary domain. However, many international space agencies (i.e. JAXA) and crossover projects (i.e. MAVEN) have adopted CDF as the preferred data format because they have existing infrastructure that relies on the format and due to the broad software support for the format. These groups are seeking or are required to archive their data with the Planetary Data System (PDS). Since CDF is a general and flexible format, files can be constructed in a variety of ways. Adopting a set of "best practices" for constructing a CDF will enable archiving of the data with PDS and reliable generation of alternate forms (i.e. ASCII Tables) of the data. This white paper describes the best practices and the rational for them.

CDF Specification

Common Data Format (CDF) [1] is a self-describing data format for the storage of scalar and multidimensional data in a platform- and discipline-independent way. The format supports built-in data compression (RLE, gZIP, Huffman). It is has both library and toolkit support on the most commonly used platforms and programming languages. Compressed data is transparently uncompressed when read with the provided libraries and tools. The current release of CDF is Version 3.4 (February 28, 2012) [2].

A CDF file can contain both data and metadata. Data are stored in a CDF file as variables, with metadata stored as attributes. However, its common practice to store some data related metadata in variables. It is possible to assign an attribute to a variable to help in differentiating between data and metadata (See section "Records and Sparse Data"). Typically all variables are stored in a single file, but the CDF specification does allow each variable to be stored in a separate file (multi-file CDF).

It is possible to write the values of variables to a CDF file incrementally. Each write will create a new set of metadata and data. This will cause the data for a variable to spread out (or fragmented) in the CDF. In addition it is possible to mark variables as deleted (unused) without removing the data. Once data is marked as "unused" the metadata describing it is no longer available. A variable may also be "virtual" with the values being determined through a calculation of a formula. The formula is expressed as a text string and is defined by individual projects (or data producers). There are no functions defined in the CDF specification.

Attributes

In a CDF file there are two types of attributes: global and variable. Global attributes describe aspects of the overall CDF and variable attributes describe features of a variable. An attribute has a name and value. An attribute name must start with a letter and can otherwise contain letters, numbers and the underscore character (no other special characters allowed). An attribute name is case-sensitive. A value can be an array and of any allowed CDF data type.

In the CDF specification there are no pre-defined or required attributes. However, commonly used tools and infrastructure (i.e. CDAWeb) expect certain attributes to be defined. The International Solar-

Terrestrial Physics (ISTP) Program [4] defined a set of global and variable attributes which is endorsed by the Inter-Agency Consultative Group (IACG) [5] and has been widely adopted. There has also been an effort to define an archive suitable version of CDF, referred to as CDF-A, which supports a richer set of metadata based on the Space Physics Archive Search and Extract (SPASE) information model [6].

Global Attributes

The ISTP/IACG guidelines define the following global attributes:

Project	The name of the project.
Discipline	The science discipline and subdiscipline. (enumeration)
Data_type	ISTP defined exchangeable data product type. (enumeration)
Descriptor	The name of the instrument or sensor that collected the data.
Data_version	Project assigned version for the data.
Instrument_type	The ISTP defined instrument type. Multi-valued. (enumeration)
Logical_file_id	The name of the CDF file using the ISTP naming convention.
Logical_source	Source_name, data_type, and descriptor information.
Logical_source_description	Full words associated with the Logical_source.
Mission_group	The assigned name of the mission or project. (enumeration)
PI_name	First initial and last name of the PI.
PI_affiliation	A recognizable abbreviation of the PI affiliation.
Source_name	The mission or investigation that contains the sensors
TEXT	Description of the experiment

ISTP/IACG also defines the following recommended (optional) attributes:

Generated_by	The generating data center/group.
Generation_date	Date stamp for the creation of the file.
HTTP_LINK	The URL for the PI or Co-I web site holding on-line data.
LINK_TEXT	Text describing on-line data available at PI or Co-I web sites.
LINK_TITLE	The title of the web site holding on-line data available at PI or Co-I web sites.
MODS	History of modifications made to the CDF data set.
Parents	The parent CDF(S) for files of derived and merged data sets.
Rules_of_use	Citability and PI access restrictions. This may point to a World Wide Web page specifying the rules of use.
Skeleton_version	The skeleton file version number.
Software_version	The version of the software that generated the CDF.
Time_resolution	The time resolution of the file.
TITLE	A title for the data set.
Validate	Written by software for automatic validation of features.

CDF-A defines the following required attributes:

spase_DatasetResourceID	The SPASE ResourceID assigned of the NumericalData resource the data file is part of.

CDF-A defines the following optional attributes:

spase_DatasetResource	The SPASE XML description of the dataset that corresponds to the SPASE ResourceID
spase_GranuleResourceID	The Granule ResourceID assigned to the data file.

spase_GranuleResource	The SPASE XML description of the dataset that corresponds to the SPASE Granule ResourceID

Variable Names

CDF allows variable names to be composed from the "ASCII Character Set". While the ASCII Character set includes non-printable characters, it appears that the intention was printable characters in the ASCII character set. The ISTP/IACG specification restricts CDF Variable names to contain only letters, numbers and the underscore. ISTP/IACG further specifies that a variable name must begin with a letter.

Variable Attributes

CDF tools require (expect) the following variable attributes

FORMAT	A Fortran or C format specification that is used when displaying a variable value.
VALIDMIN	The minimum valid value for a variable.
VALIDMAX	The maximum valid value for a variable.
FILLVAL	The value used for missing or invalid variable values.
MONOTON	The monotonicity of a variable: INCREASE (strictly increasing values), DECREASE (strictly decreasing values), or FALSE (not monotonic).
SCALEMIN	The minimum value for scaling a variable when graphically displaying its values.
SCALEMAX	The maximum value for scaling a variable when graphically displaying its values. In the description of each CDF toolkit program, the special attributes that may affect that program's operation are defined. Note that most of the CDF toolkit programs can be instructed to ignore these special attributes.

In addition, the ISTP/IACG requires the following attributes:

CATDESC	Approximately 80-character string which is a textual description of the variable
DEPEND_i	Ties a dimensional data variable to a support_data variable. Contains the name of the variable. "i" is replaced with the index of the dimension. For ISTP the DEPEND_0 must be defined and have the value 'Epoch'.
DISPLAY_TYPE	what type of plot to make (i.e. time_series, spectrogram, stack_plot,image)
FIELDNAM	A description of the variable (up to 30 characters).
FORM_PTR	A variable which stores the name of the variable that contains the FORMAT string. To be used only if FORMAT is not present.
LABLAXIS/LABL_PTR_i	The label for the y-axis of a plot or to provide a heading for a data listing. If labeling a variable with dimensions use the LABL_PTR_i form with "i" replaced with the index of the dimension.
UNITS/UNIT_PTR	The units of the variable. In the UNIT_PTR form it contains the name of the variable which stores the UNITS string. To be used only if UNITS is not present.
VAR_TYPE	Identifies a variable as either data, support_data, metadata or ignore_data.

ISTP/IACG also defines the following recommended (optional) attributes:

AVG_TYPE	Identifies the technique used for averaging the data.
DELTA_PLUS_VAR	The positive uncertainty in (range of) the original variable's value.
DELTA_MINUS_VAR	The negative uncertainty in (range of) the original variable's value.

DERIVN	The derivation of the variable, possibly including a function/algorithm name or journal reference.
DICT_KEY	The ISTP/IACG dictionary keyword that describes the variable.
MONOTON	Whether the variable is monotonically increasing or monotonically decreasing. A value of INCREASE indicates strictly increasing values, DECREASE strictly decreasing values, or FALSE indicates not monotonic.
SCALETYP	Whether the variable should have a linear or a log scale as a default.
SCAL_PTR	The name of the variable containing SCALETYP for multidimensional data.
sig_digits	the number of significant digits or other measure of data accuracy
SI_conversion	The conversion factor to SI units.
VAR_NOTES	Ancillary information about the variable.
V_PARENT	The "attached" variable which stores the parent variable(s) of a derived variable.

Records and Sparse Data

A CDF can have one or more records. Each record is a set of variable arrays. The variable arrays in each record are generally related to each other in some way (i.e. time), but this not required in the CDF specification. For time varying data ISTP/IACG requires the time value associated with each record be stored in the variable with the name 'Epoch' and is attached to all time varying data variables via DEPEND_0. Furthermore, 'EPOCH' should be the first variable in each CDF data file. The ISTP/IACG guidelines also recommend the following names and purpose for variables:

Quality Flag	Quality or status flag for each record.
Time_PB5	Alternate representation of time in the format YEAR (4 digit), DAY OF YEAR (note: January 1 is Day 1), and MSEC OF DAY (elapsed ms).
Post Gap Flag	An indication of the reason for a gap. Appears in the record following the gap.

CDF supports all the common data types (single byte; character; 1, 2, 4, and 8 byte integers; 4 and 8 bytes floating point; and special 8 and 16 byte time). ISTP/IACG limits data values to integer and real, with character data allowed for metadata or support data (for example, labels).

In a single-file CDF a variable can be specified as having sparse records. When using sparse records, a value is physically stored only when the value changes. When the data are read the "virtual" records can either be filled with the defined pad value or with the last known (physical) value. If a variable does have sparse records the internal (CDF binary record) sparse records attribute must be set to either PAD_SPARSERECORDS or PREV_SPARSERECORDS.

In a CDF file, variables which maintain the same value from record to record do not have to be physically stored in a CDF file. A repeating value may be stored once in the CDF with metadata indicating how the variable is to be replicated by software. If the variable has the internal (CDF) record variance attribute of "VARY" then a value may change from record to record.

Physical Layout

A CDF file is a sequentially written set of alternating blocks of metadata and data with a block of global metadata at the beginning of the file and another at end of the file. The structure, encoding and storage

order for a block of data is defined in the preceding block of metadata. The CDF specification refers to these blocks as "records".

Other Features

Starting with Version 3.2 an MD5 checksum can be stored at the end of the CDF file following the last record of the CDF file. It is calculated on the CDF content of the file can only be used to ensure the integrity of a CDF content. The MD5 Checksum is not included in the internal value for the length of the CDF. The MD5 Checksum is calculated on the CDF content (all bytes up to, but not including the MD5 checksum appended to the end of the file).

Prior to version 3.0 a CDF file could not be larger than 2G bytes. Starting with version 3.0 this limit is removed. Version 3.1 and later is backward compatible with version 2.7.2 and earlier.

PDS4 Archive Requirements

The Planetary Data System (PDS) prefers transparent, non-proprietary formats when archiving data. Data should be in a form which can be read using the accompanying plain text metadata. Once read, the data should be usable without additional processing. The PDS4 information model supports describing a variety of storage structures which includes arrays and tabular data.

Requirements for Archivable CDF files

To ensure data in a CDF file will be in an archivable form

1) Create CDF compliant with version 3.4 or later.
2) Use single file CDF.
3) No compression (file or variable).
4) No fragmented variables (all data for a variable must be contiguous in the file).
5) Use only "zVariables" (also recommended by the CDF standard)

To aid in the generation of PDS metadata it is advisable to include

1) CDF Tool compliant metadata.
2) ISTP/IACG compliant metadata.

Labeling CDF files with PDS4

If the recommendations are followed it is possible to create a PDS4 label to describe the contents of the CDF file as containing multiple arrays. Each variable can be described as an appropriately dimensioned array. Much of metadata contained in the CDF file can be replicated in the appropriate elements in the PDS4 label.

Tools and Techniques

Generating compliant CDF files

Creating a CDF file which is compliant with the structural PDS archive requirements can be achieved using the "cdfconvert" tool which is part of the CDF software distribution [4]. The command and options are:

```
cdfconvert  {src.cdf} {dest.cdf} -single  -network -sparseness vars:srecords.no \
        -compression vars:none -zmode 2
```

This will convert {src.cdf} (which can be a multiple file CDF) and write to {dest.cdf} a copy that is written as a single file with network encodings, no sparseness, no compression and that contains only zVariables.

Using igpp-docgen to create a PDS4 label

The igpp-docgen [8] is an application which defines an environment that can process Apache Velocity[5] templates. It can read CDF files (amount other formats) and make information about the content available for use through the Apache Velocity Template Language [6]

```
docgen -f cdf cdf:cdf:cdf/thg_l2_mag_and_20121226_v01.cdf cdf/pds4label.vm
```

Converting Tables to CDF

Tools exist to convert PDS3 labeled table data into CDF files. PDS-CDF converters are available at http://mgmt.pds.nasa.gov/converters.html

Appendix A

CDF Tools and Libraries.

CDF is supported on the following platforms:

- DEC Alpha/OSF1 & OpenVMS
- DECstation/ULTRIX & VMS
- HP 9000 series/HP-UX
- PC Windows NT/2000/XP/Vista/Windows 7, Linux, Solaris, Cygwin, MinGW & QN X
- IBM RS600 series/AIX
- Macintosh OS X 10.3 or a later version
- NeXT/Mach
- SGI Iris, Power series and Indigo/IRIX
- Sun/SunOS & SOLARIS

CDF libraries are available for the following programming languages:

- C
- C#
- Fortran
- Java
- Perl

Support Format Transforms

- MakeCDF (reads flat data sets, in both binary and text)
- CDF-to-netCDF (Only supporting netCDF V3.*)
- CDF-to-FITS
- CDF-to-ASCII (Text dump of a CDF file)
- CDF-to-CDF Skeleton table
- CDF-to-CDFML (XML representation of CDF)
- CDFML-to-CDF
- netCDF-to-CDF (Only supporting netCDF V3.*)
- FITS-to-CDF
- HDF4-to-CDF
- HDF5-to-CDF (HDF5 in text dump to CDF. To be provided upon request)

Supported Analsyis Environments:

IDL:

MATLAB

Tools

CDFedit: Allows the display and/or modification contents of a CDF.

CDFexport: Write the contents of a CDF to the terminal screen, a text file, or another CDF.

CDFconvert: Change format, version, encoding, compression, sparseness and checksum.

CDFcompare: Displays the differences between two CDFs

CDFstats: Produces a statistical report on a CDF's variable data.

CDFinquire: Displays the version of the CDF distribution being used.

CDFdir: Display a directory listing of a CDF's files.

CDFmerge: Merge two or more CDF files into a single file.

CDFdump: Display or extract the contents of a CDF file to a screen (default) or text file.

CDFirsdump: displays the statistics of CDF Internal Records (IRs).

CDFvalidate: optionally performs sanity checks on data in the CDF files.

CDFleapsecondsinfo: Displays the information of the leap seconds table that the CDF uses.

SkeletonTable: create an ASCII text file containing information about a CDF.

SkeletonCDF: Make a fully structured CDF, by reading a structured information in a text file.

Data Types

Integer Data Types

CDF_BYTE 1-byte, signed integer.
CDF_INT1 1-byte, signed integer.
CDF_UINT1 1-byte, unsigned integer.
CDF_INT2 2-byte, signed integer.
CDF_UINT2 2-byte, unsigned integer.
CDF_INT4 4-byte, signed integer.
CDF_UINT4 4-byte, unsigned integer.
CDF_INT8 8-byte, signed integer.

Floating Point Data Types

CDF_REAL4 & CDF_FLOAT 4-byte, single-precision floating-point.
CDF_REAL8 & CDF_DOUBLE 8-byte, double-precision floating-point.

Character Data Types (Limited to ASCII set of characters)

CDF_CHAR 1-byte, character.
CDF_UCHAR 1-byte, unsigned character.

EPOCH Data Types (milliseconds since 01-Jan-0000 00:00:00.000)

CDF_EPOCH 8-byte, double precision floating point.
CDF_EPOCH16 two 8-byte, double precision floating point.

TT2000 Data Types (milliseconds since 2000-01-01T12:00:00.000000000, aka J2000. w/ leap seconds)

CDF_TIME_TT2000 8-byte, signed integer

Encoding

Run-Length Encoding, Huffman, Adaptive Huffman, GZIP

Special Attributes

FORMAT	A Fortran or C format specification that is used when displaying a variable value.
VALIDMIN	The minimum valid value for a variable.
VALIDMAX	The maximum valid value for a variable.
FILLVAL	The value used for missing or invalid variable values.
MONOTON	The monotonicity of a variable: INCREASE (strictly increasing values), DECREASE (strictly decreasing values), or FALSE (not monotonic). Monotonicity only applies to NRV variables that vary along one dimension and RV variables that vary along no dimensions.

SCALEMIN	The minimum value for scaling a variable when graphically displaying its values.
SCALEMAX	The maximum value for scaling a variable when graphically displaying its values.

In the description of each CDF toolkit program, the special attributes that may affect that program's operation are defined. Note that most of the CDF toolkit programs can be instructed to ignore these special attributes.

References

[1] CDF Internal Format Description; Version 3.4, February 28, 2012; Space Physics Data Facility; NASA / Goddard Space Flight Center. http://cdaweb.gsfc.nasa.gov/pub/software/cdf/doc/cdf34/cdf34ifd.pdf

[2] CDF User's Guide; Version 3.4, February 28, 2012; Space Physics Data Facility; NASA / Goddard Space Flight Center. http://cdaweb.gsfc.nasa.gov/pub/software/cdf/doc/cdf34/cdf34ug.pdf

[3] ISTP/IACG guidelines for CDF, http://spdf.gsfc.nasa.gov/istp_guide/istp_guide.html

[4] International Solar Terrestrial Physics (ISTP), http://pwg.gsfc.nasa.gov/istp/

[5] Inter-Agency Consultative Group (IACG), http://www.iacg.org/

[6] Space Physics Archive Search and Extract (SPASE), http://www.spase-group.org/

[7] COSPAR Panel on Radiation Belt Environment Modeling (PRBEM) http://craterre.onecert.fr/prbem/Standard_File_Format.pdf

[8] igpp-docgen tool, http://release.igpp.ucla.edu/igpp/docgen/

Revision History

July 24, 2018: Revision 7. Add “Tools and Techniques” section, changed title to “Guide to Archiving CDF Files in PDS4”

March 5, 2014: Revision 6.

January 30, 2014: Revision 5. Updated list of the requirements for archivable CDF to match refined requirements in presentations.

July 13, 2013: Revision 4.