Data formats standards and metadata

Discussion led by Bobby Candey

Space Physics Data Facility (SPDF)
<https://spdf.gsfc.nasa.gov>
Heliophysics Science Division (Code 670)
NASA Goddard Space Flight Center

Third IHDEA meeting 2019 Oct.17

Formats in NASA Space Science

- Standard formats
  - **FITS** used in astronomy and solar physics [FITS and WCS metadata]
  - **HDF** in Earth sciences [HDF-EOS hdfeos.org metadata]
  - **netCDF** in atmosphere [Climate and Forecast cfconventions.org] and ITM [ISTP/SPDF metadata]
  - **CDF** in the rest of Heliophysics [ISTP/SPDF Guidelines metadata]

- PDS added **CDF-A** as standard format (PDS-3, PDS-4, JPEG): CDF with ISTP/SPDF Guidelines and two SPASE attributes, but no compression or sparse variables
- ICON/GOLD metadata uses the ISTP/SPDF guidelines in netCDFs, netCDF4 Classic model with no groups or user-defined variable types, time is unlimited dimension

- SPDF has converters between CDF, CDFML, netCDF, HDF, FITS, and to PDS-3

Why metadata conventions

- Leverage standardized self-describing data formats, metadata for datasets and parameters, time conventions, and dataset and filenaming conventions to enable effective data analysis and browsing using generic easy-to-use software and web services
- Restricting metadata representations limits the number of equivalent possibilities with which software must deal, and thus fosters interoperability
- Conventions standardize ways to name things, represent relationships, and locate data in space and time
- Enables developing applications with powerful extraction, regridding, analysis, visualization, and processing capabilities
- Abstracts general data models to represent data semantics.
- Embody provider’s experience and capture the meaning in data and make data semantics accessible to humans as well as programs
- Provide higher-level abstractions such as coordinate systems, standard names for physical quantities for comparing different data, and distinguish variables

Some standards and conventions

- **SPASE** <http://www.spase-group.org> dataset descriptions for easy searching
- **Heliophysics Data Portal** <https://heliophysicsdata.sci.gsfc.nasa.gov>
- **ISTP/IACG/SPDF Guidelines** for global and variable attributes <https://spdf.gsfc.nasa.gov/sp_use_of_cdf.html>
  - SKTeditor metadata creation tool <https://spdf.gsfc.nasa.gov/skteditor>
  - Defining additional standard attributes: Cluster, THEMIS, RBSP (PRBEM), MMS, etc.
- **Dataset naming and file naming** recommendations <http://www.tsds.org/Recommended_file_and_data_collection_naming_practices> and filenaming templates <http://tsds.org/uri_templates> $Y/data_$Y_j_idx.cdf
- **CDF** <https://cdf.sci.gsfc.nasa.gov> scientific data format (including its new Python library <https://github.com/MAVENSDC/cdflib>)
  - Time variable types <https://cdf.sci.gsfc.nasa.gov/html/leapseconds_requirements.htm>
- **netCDF** <https://www.unidata.ucar.edu/software/netcdf/>
- **FITS** <https://fits.gsfc.nasa.gov/>
- **UDunits** www.unidata.ucar.edu/software/udunits/
- Tools enabled by standards: CDAWeb and CDAWlib IDL library, Autoplot <http://autoplot.org>, SPEDAS <http://spedas.org> IDL library

ISTP/SPDF Guidelines Structure and Metadata Concepts

- **ISTP/IACG Guidelines (mid1990s) and subsequent extensions by SPDF define a limiting set of implementation standards for CDFs**
  - Include general file naming conventions
  - Data is time-ordered and time-identified; times vary by record
  - Set of required and suggested metadata (details on next slide)
  - Variable attributes can point to other variables by name and carry arguments
    - Attributes thus carry information about relationships among variables
    - Variables can carry metadata (e.g. labels for dimensional variables)
  - Terminology: "Skeleton" CDF is a CDF with structure and metadata defined but no data, so it can be used as a template from which to build a data file

- **CDAWeb additional concepts: "Master" CDFs and "Virtual" Variables**
  - "Master" CDF is the use of a "skeleton" CDF to insert supplemental or updated metadata for CDFs as a dataset
  - "Virtual" variables are computed variables, using specialized CDF attributes to link defined variables and routines within CDAWeb/CDAWlib

- **Concepts above directly/easily map to data in netCDF**

ISTP/SPDF Metadata Elements

- **Variable attributes required for automated processing:**
  - Catdesc for longer variable description
  - Depend_0 points to time variables
  - Depend_1, 2, 3 point to variables that describe other dimensions
  - Fieldnam short variable name for plots
  - Fillval values indicating missing or bad data
  - Lablaxis/Labl_ptr for axis and column titles
  - Units/Unit_ptr
  - Validmin/max for valid data range
- **CDF Time variable types**
  - CDF_TIME_TT2000 nanoseconds from J2000 in Terrestrial Time in 8 byte integer handles leap seconds and is well-defined; UTC conversion requires up-to-date leap second table (last value stored in CDF header as a check)
  - EPOCH milliseconds from 0AD in 8byte float; usually UTC but not leap seconds
  - EPOCH16 picoseconds from 0AD in two 8byte float; usually UTC but not leap seconds
- **ISTP/SPDF Guidelines online at**

https://spdf.gsfc.nasa.gov/sp_use_of_cdf.html

CDF in More Detail

- **Software distribution APIs: C, C#, VisualBasic, Java, Perl, Fortran**
  - Stable, fully functional
    - Built-in compression capability and transparent decompression
    - CDF includes an internal checksum to ensure integrity
    - CDFconvert utility to optimize internal layout
    - Multiple standard format translators
  - Utilities for modifying CDFs and to/from regular text or XML (CDFML) files
  - Support libraries for IDL and MATLAB (included in their distributions)
  - Additional CDAWlib distribution includes rich set of IDL procedures
- **3 additional independent implementations for reading/writing CDFs**
  - Bryan Harter’s pure Python github.com/MAVENSDC/cdflib
  - Mark Taylor’s pure Java JCDF library (CDF read only)
    - Used by TOPCAT and STILTS. See www.star.bristol.ac.uk/~mbt/jcdf
  - Nand Lal’s pure Java CDFJ library (now included in SPDF’s CDF distribution)

A Useful Tool: Create/Edit a Skeleton file compliant to ISTP/SPDF standard

- SKTEditor is a Java, web-start application
  - Guide designers to good choices consistent with ISTP/SPDF guidelines
  - Create new CDF ..or... check/correct then modify an existing skeleton file
  - Now supports netCDF

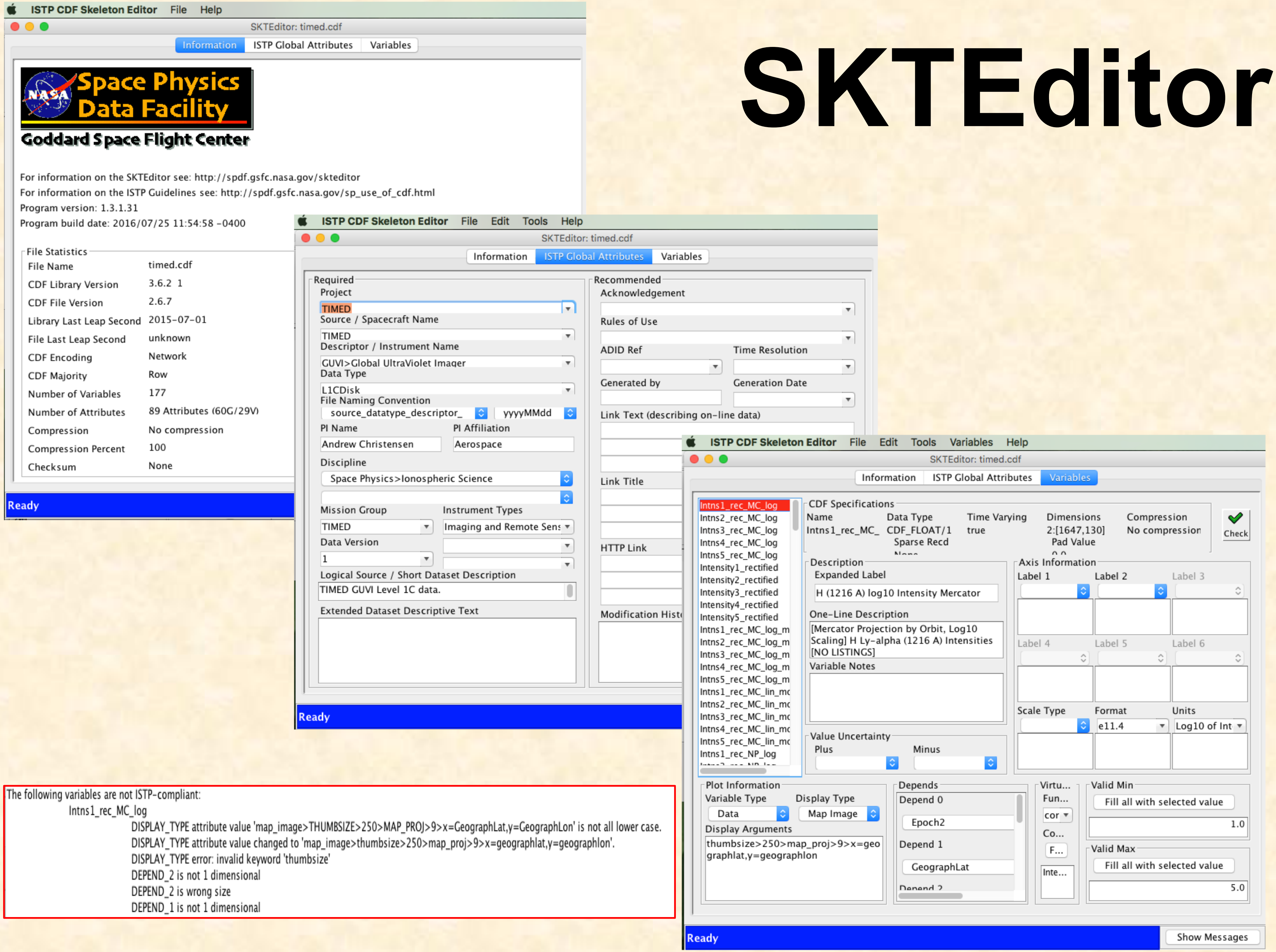
    »

- Guided by the interface flow, add or edit
  - Scalar and higher-dimensional variables, multiple time variables
    - Times as cdf_epoch or as cdf_time_tt2000
  - Variable attributes (descriptions, labels, units, display_type)
  - Global attributes and file naming
  - Virtual variables (functions in CDAWlib, compute values on-the-fly)

    »

- Checking and validation functions
  - Against ISTP/SPDF standards
  - For PRBEM, MMS or other specified project compliance reporting

SKTEditor



ISTP CDF Skeleton Editor File Help
SKTEditor: timed.cdf
Information
ISTP Global Attributes
Variables
Space Physics Data Facility
Goddard Space Flight Center

For information on the SKTEditor see: http://spdf.gsfc.nasa.gov/skteditor
For information on the ISTP Guidelines see: http://spdf.gsfc.nasa.gov/sp_use_of_cdf.html
Program version: 1.3.1.31
Program build date: 2016/07/25 11:54:58 -0400

File Statistics

File Name	timed.cdf
CDF Library Version	3.6.2 1
CDF File Version	2.6.7
Library Last Leap Second	2015-07-01
File Last Leap Second	unknown
CDF Encoding	Network
CDF Majority	Row
Number of Variables	177
Number of Attributes	89 Attributes (60G/29V)
Compression	No compression
Compression Percent	100
Checksum	None

Ready

Required

Project: TIMED
Source / Spacecraft Name: TIMED
Descriptor / Instrument Name: GUVI>Global UltraViolet Imager
Data Type: L1CDisk
File Naming Convention: source_datatype_descriptor_ yyyyMMdd
PI Name: Andrew Christensen
PI Affiliation: Aerospace
Discipline: Space Physics>Ionospheric Science
Mission Group: TIMED
Instrument Types: Imaging and Remote Sens
Data Version: 1
Logical Source / Short Dataset Description: TIMED GUVI Level 1C data.
Extended Dataset Descriptive Text

Recommended

Acknowledgement
Rules of Use
ADID Ref
Time Resolution
Generated by
Generation Date
Link Text (describing on-line data)
Link Title
HTTP Link
Modification Hist

Ready

Intns1_rec_MC_log
Intns2_rec_MC_log
Intns3_rec_MC_log
Intns4_rec_MC_log
Intns5_rec_MC_log
Intensity1_rectified
Intensity2_rectified
Intensity3_rectified
Intensity4_rectified
Intensity5_rectified
Intns1_rec_MC_log_m
Intns2_rec_MC_log_m
Intns3_rec_MC_log_m
Intns4_rec_MC_log_m
Intns5_rec_MC_log_m
Intns1_rec_MC_lin_mc
Intns2_rec_MC_lin_mc
Intns3_rec_MC_lin_mc
Intns4_rec_MC_lin_mc
Intns5_rec_MC_lin_mc
Intns1_rec_NP_log

CDF Specifications

Name	Data Type	Time Varying	Dimensions	Compression
Intns1_rec_MC_	CDF_FLOAT/1	true	2:[1647,130]	No compression
	Sparse Recd		Pad Value	

Description
Expanded Label: H (1216 A) log10 Intensity Mercator
One-Line Description: [Mercator Projection by Orbit, Log10 Scaling] H Ly-alpha (1216 A) Intensities [NO LISTINGS]
Variable Notes

Axis Information
Label 1 Label 2 Label 3
Label 4 Label 5 Label 6
Scale Type
Format: e11.4
Units: Log10 of Int

Value Uncertainty
Plus Minus

Plot Information
Variable Type: Data
Display Type: Map Image
Display Arguments: thumbsize>250>map_proj>9>x=geographlat,y=geographlon

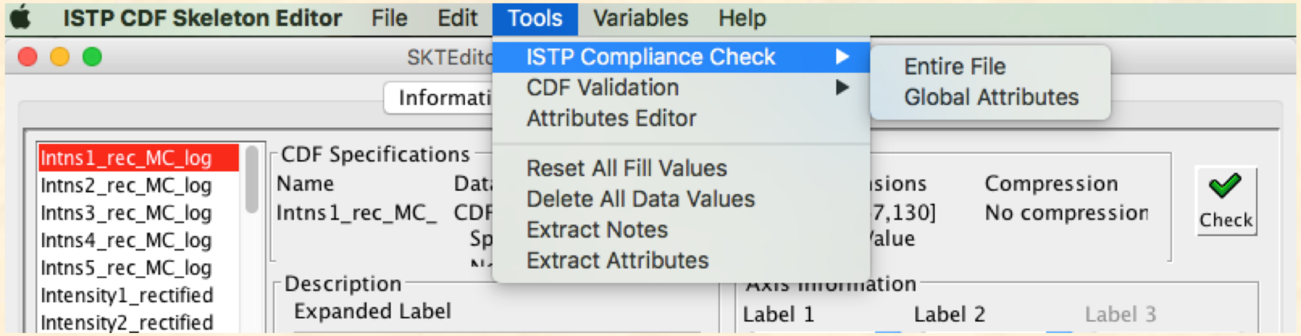
Depends
Depend 0: Epoch2
Depend 1: GeographLat
Depend 2

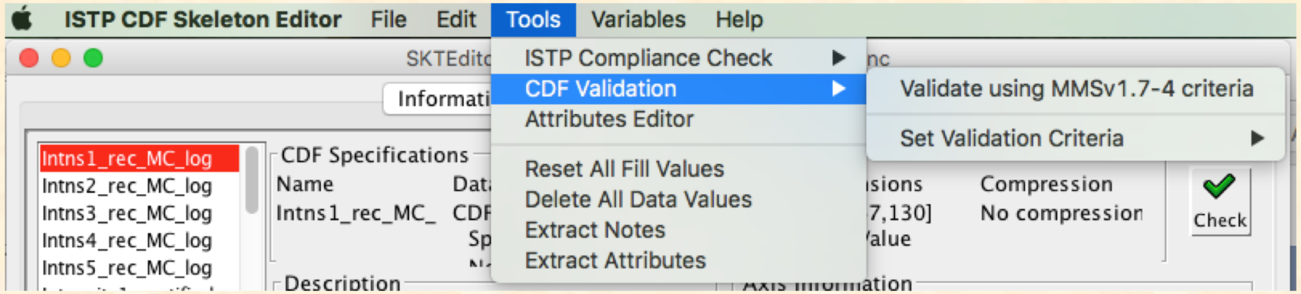
Virtu...
Fun... cor
Co... F...
Inte...

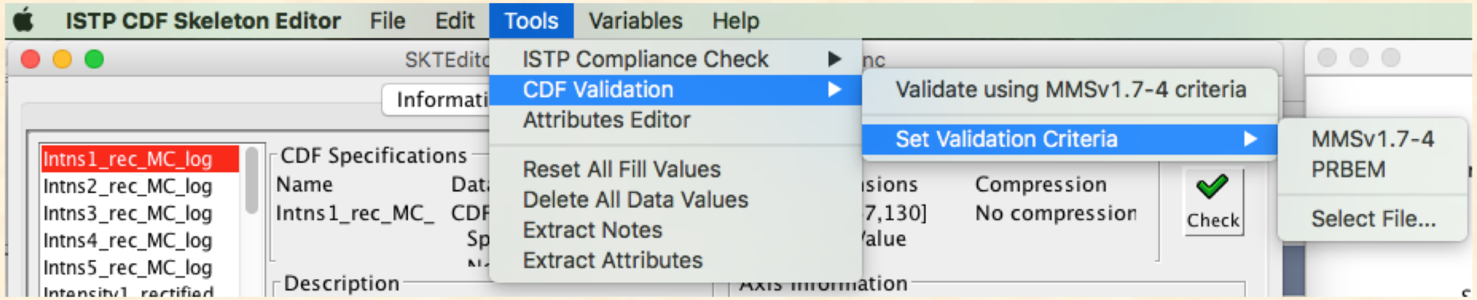
Valid Min: Fill all with selected value 1.0
Valid Max: Fill all with selected value 5.0

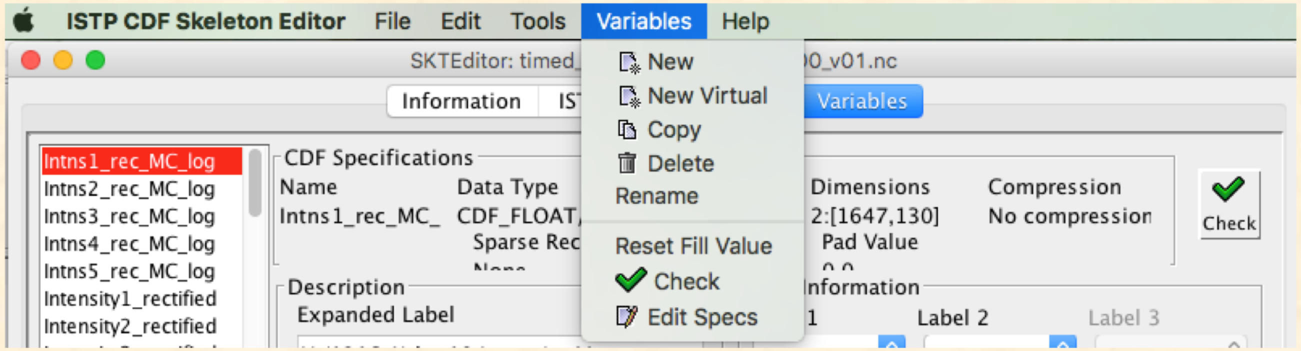
Ready

```
The following variables are not ISTP-compliant:
        Intns1_rec_MC_log
                DISPLAY_TYPE attribute value 'map_image>THUMBSIZE>250>MAP_PROJ>9>x=GeographLat,y=GeographLon' is not all lower case.
                DISPLAY_TYPE attribute value changed to 'map_image>thumbsize>250>map_proj>9>x=geographlat,y=geographlon'.
                DISPLAY_TYPE error: invalid keyword 'thumbsize'
                DEPEND_2 is not 1 dimensional
                DEPEND_2 is wrong size
                DEPEND_1 is not 1 dimensional
```


ISTP CDF Skeleton Editor
File
Edit
Tools
Variables
Help
SKTEdito
Informati
ISTP Compliance Check
CDF Validation
Attributes Editor
Reset All Fill Values
Delete All Data Values
Extract Notes
Extract Attributes
Entire File
Global Attributes
Intns1_rec_MC_log
Intns2_rec_MC_log
Intns3_rec_MC_log
Intns4_rec_MC_log
Intns5_rec_MC_log
Intensity1_rectified
Intensity2_rectified
CDF Specifications
Name
Intns1_rec_MC_
Compression
No compression
Check
Description
Expanded Label
Label 1
Label 2
Label 3


ISTP CDF Skeleton Editor
File
Edit
Tools
Variables
Help
ISTP Compliance Check
CDF Validation
Attributes Editor
Reset All Fill Values
Delete All Data Values
Extract Notes
Extract Attributes
Validate using MMSv1.7-4 criteria
Set Validation Criteria
Intns1_rec_MC_log
Intns2_rec_MC_log
Intns3_rec_MC_log
Intns4_rec_MC_log
Intns5_rec_MC_log
CDF Specifications
Compression
No compression
Check
Description


ISTP CDF Skeleton Editor
File
Edit
Tools
Variables
Help
ISTP Compliance Check
CDF Validation
Attributes Editor
Reset All Fill Values
Delete All Data Values
Extract Notes
Extract Attributes
Validate using MMSv1.7-4 criteria
Set Validation Criteria
MMSv1.7-4
PRBEM
Select File...
Intns1_rec_MC_log
Intns2_rec_MC_log
Intns3_rec_MC_log
Intns4_rec_MC_log
Intns5_rec_MC_log
CDF Specifications
Compression
No compression
Check
Description


ISTP CDF Skeleton Editor
File
Edit
Tools
Variables
Help
SKTEditor: timed_
)0_v01.nc
Information
IS
Variables
New
New Virtual
Copy
Delete
Rename
Reset Fill Value
Check
Edit Specs
Intns1_rec_MC_log
Intns2_rec_MC_log
Intns3_rec_MC_log
Intns4_rec_MC_log
Intns5_rec_MC_log
Intensity1_rectified
Intensity2_rectified
CDF Specifications
Name
Data Type
Intns1_rec_MC_
CDF_FLOAT
Sparse Rec
Dimensions
2:[1647,130]
Pad Value
Compression
No compression
Check
Description
Expanded Label
Information
Label 2
Label 3

Creating an ISTP/IACG Skeleton CDF: Understand the Data to be Loaded

- What are the key data quantities
  - What is their definition/meaning?
  - How are they going to be named?
    - N.B. MMS parameter naming convention: scId_instrumentID_paramName
      -
- Understand (at the dataset level)
  - Dimensionality and dependencies
  - Variance with time and dimension
    - ISTP/SPDF conventions allow >1 time variable in a file
    - Carry slowly-varying data as variables rather than in attributes
- General rule is to capture relationships in the structure
  - Otherwise capture relationships in variable attributes
  - Want relationships to be logically-structured and machine-readable
    - Available for more general-purpose codes to exploit
- Let CDF deal with mechanics of efficient data storage
  - Once more: lay out data by what’s science logical and useful
    - E.g. methods to handle slowly-varying data include setting “sparse=sRecords.PREV”

The MakeCDF Program

- Highly relevant to teams with an internal format like csv
- Engine+Data+FileFormatDescription+Skeleton CDF = Output CDF
  - Capable of supporting both ASCII and BINARY files, multiple time formats
  - Can automatically skip header records (or be manually controlled)
  - (Optional) arguments allow handling complex inputs; e.g. sub-records
- File Format Description (FFD) or Translation File defines
  - Mapping of data values to named CDF variables
  - Data format of input data
- Example Input (specifically Geotail CPI plasma data made into csv)

```
2007,304,0,1,10,102,0.4400E+03,0.91000E+02,0.18230E+03,-.43958E+03,-
.17655E+02,-.76791E+01,0.1050000E+06,0.78900E+01,0.3055E+01
2007,304,0,2,48,842,0.4420E+03,0.90900E+02,0.18180E+03,-.44173E+03,-
.13882E+02,-.69426E+01,0.1230000E+06,0.78000E+01,0.3048E+01
2007,304,0,4,24,590,0.4370E+03,0.90600E+02,0.18040E+03,-.43697E+03,-
.30506E+01,-.45762E+01,0.1310000E+06,0.58800E+01,0.2246E+01
```

Major CDAWlib routines

- READ_MYCDF - The function READ_MYCDF reads from one to many variables from one to many CDF files (in one dataset), and returns all data and metadata for these variables in a single structure
- PLOTMASTER - This function accepts from 1 to 10 structures of the type returned by READ_MYCDF, determines the plot type for each variable in each of the structures, and plots each (to either an X window or GIF file). Returns a 0 if plotting was successful, and a -1 if unsuccessful.
- LIST_MYSTRUCT - Given a "data structure" read with read_mycdf, LIST_mystruct generates an ascii listing of the data
- WRITE_MYCDF - This function accepts from 1 to 10 structures of the type returned by READ_MYCDF, produces a cdf file for each structure.
  - Each have many keywords, please see the code for those

SPDF Services to Support netCDF

- **Use the ISTP/SPDF metadata/structure guidelines and “master” files**
- **Support ingest/distribution of data through CDAWeb**
  - CDAWeb system extended to read/write data in netCDF using same IDL structures used for CDFs
  - Enables access through existing webservices APIs
  - SKTEditor tool extended to read/write netCDF
- **GOLD and ICON agreed to try to follow ISTP/SPDF metadata and structure standards in producing netCDF4 data products**
  - SPDF further enhanced CDF <-> netCDF conversion software
  - SPDF created an IDL script to address a specific netCDF structuring issue
- **Expect to leverage this new netCDF capability for other datasets**
  - New high-resolution GOES science data from NOAA (including 16 and 17)
  - Improved support for older Heliophysics missions (mainly ITM) that used netCDF but without ISTP/SPDF metadata and structure standards

netCDF Issues

- No predefined time variable types
  - Time not always the unlimited dimension
  - CDAWeb adds CDF_TIME_TT2000 virtual variables for netCDF datasets, computed from various time schemes (base time, time units)
- CDAWeb adds missing Fillval, Validmin/max, Var_type, depend_0, and other attributes
- netCDF to CDF converter adds attributes to store version, dimensions, sizes, compression, chunking, and string (not character) information
- CDF to netCDF converter converts time variables to binary or encoded string forms
- Compression requires careful block size determination
- Supports only netCDF4 Classic model with no groups or user-defined variable types

Some Recent CDF updates

- Improved CDFML format
- Added ISO-8601 time outputs to utilities
- Added leap second header to flag outdated leap second table
- Improved temporary file and directory handling
- Added new modular CDFread C-based functions
- Allowed Null-terminating string for variable data and attribute entries
- Allowed multiple strings for variable attribute entry
- Added support for ARM architecture
- Added Itanium IA64 on OpenVMS
- Added pure Java package, cdfj.jar, for CDF read/write
- And miscellaneous bug fixes and performance tweaks

Upcoming Activities

- CDF
  - ongoing maintenance, performance improvements
  - CDF beginners guide
  - Python library: add WCS time conversions
  - Adapt netCDF command line tools like NCO.sf.net for CDFs for operations on files
- ISTP/SPDF Guidelines
  - Will soon add SPASE and DOI global attributes to CDAWeb datasets via Master CDFs when available and expose in CDAWeb interface
  - Better document Guidelines on Github with mission-specific metadata as well, but want to keep flexible for interactions with missions and enabling framework for CDAWeb services
- Rewrite SKTeditor in Javascript or similar and include SPASE fields
- Changes are driven by active archiving needs and new technology

https://spdf.sci.gsfc.nasa.gov/pub/catalogs/all.xml

<dataset ID="ac_h2_cris_cdaweb" nssdc_ID="(None) " serviceprovider_ID="AC_H2_CRIS" timerange_start ="1997-08-27 00:00:00"timerange_stop="2018-10-03 23:00:00">

- <access filenaming="ac_h2_cris_%Y%m%d_%Q.cdf" protocol="ftp" subdividedby="%Y" timerange_start= "1997-08-27 00:00:00"timerange_stop="2018-10-03 23:00:00"> <URL>ftp://cdaweb.gsfc.nasa.gov/pub/data/ace/cris/level_2_cdaweb/cris_h2</URL> </access>
- <other_info><link URL="http://www.srl.caltech.edu/ACE/ASC/level2/index.html" title="The ACE Science Center Level 2 Data website"type="documentation"> Release notes and other info available at </link> </other_info>
- <observatory ID="AC" nssdc_ID="None" serviceprovider_ID="AC">
- <description short="Advanced Composition Explorer"/> </observatory>
- <instrument ID="CRIS" nssdc_ID="None" serviceprovider_ID="CRIS">
- <description short="ACE Cosmic Ray Isotope Spectrometer"/> </instrument>
- <data_producer affiliation="California Institute of Technology" name="E. C. Stone" title="None"/>
- <mission_group ID="ACE" serviceprovider_ID="ACE"> <description short="ACE"/> </mission_group>
- <instrument_type ID="Particles (space)" serviceprovider_ID="Particles (space)"/>
- <description short="ACE/CRIS Cosmic Ray Isotope Spectrometer 1-Hour Level 2 Data - E. C. Stone (California Institute of Technology)"/>
- <mastercdf ID="https://cdaweb.gsfc.nasa.gov/pub/software/cdawlib/0MASTERS/ac_h2_cris_00000000_v01.cdf"serviceprovider_ID="https://cdaweb.gsfc.nasa.gov/pub/software/cdawlib/0MASTERS/ac_h2_cris_00000000_v01.cdf"/> </dataset>

Backup slides

SPDF Services

- **Archive** for non-solar NASA Heliophysics science data and many other missions
- **CDAWeb** browse, correlations and display, simple interface
- **SSCWeb** orbit/ground track data/displays and conjunction queries, 4D viewer
- **OMNI Database / OMNIweb-Plus** (baseline solar wind data at Earth)
- **Heliophysics Data Portal (HDP)** SPASE-based inventory of public Heliophysics-relevant data
- **CDF** self-describing scientific data format
- **SKTeditor** for creating and testing **ISTP/SPDF Guidelines** metadata (CDF/netCDF)
- **Master** CDF/netCDF concept uses file with no data to add/over-ride metadata in datasets
- **Web services** for CDF/netCDF data in CDAWeb, SSC orbits, OMNIweb, HDP; use REST versions, many language examples
  - https://cdaweb.sci.gsfc.nasa.gov/WebServices/REST/ (same for SSCweb)
- SPDF cited in a third of JGR Blue articles

SPDF Data Access

- All data (not just CDFs and netCDFs) through FTP and HTTP spdf.gsfc.nasa.gov/pub/
- https://spdf.sci.gsfc.nasa.gov/pub/catalogs/all.xml and SPASE records
- CDAWeb data browser for plots, lists (text, CSV, JSON), CDFs, audio
- Web Services in REST and SOAP cdaweb.gsfc.nasa.gov/WebServices/
- In IDL cdaweb.gsfc.nasa.gov/WebServices/REST/CdasIdlLibrary.html using CDAWlib IDL library routines spdf.gsfc.nasa.gov/CDAWlib.html
- Within Autoplot autoplot.org/help#CDAWeb
- HAPI interface to CDAWeb holdings cdaweb.gsfc.nasa.gov/hapi

- Get a CDF file containing the variables Magnitude and BGSEc data from the AC_H2_MFI dataset in the time range of 2009-06-01T00:00:00 to 2009-06-03T00:00:00: https://cdaweb.gsfc.nasa.gov/WS/cdasr/1/dataviews/sp_phys/datasets/AC_H2_MFI/data/20090601T000000Z,20090603T000000Z/Magnitude,BGSEc?format=cdf

Infrastructure for the Heliophysics Data Environment

- **Heliophysics Data Portal (HDP)**
  - HDP is a world-wide inventory of public Heliophysics-relevant data
  - SPDF also uses HDP as our high-level dataset inventory
- **CDF (Common Data Format) and SPDF Metadata Guidelines**
  - **Self-describing** data format for storing/using scalar and multi-dimensional data in a platform- and discipline-independent fashion.
  - **Self-documenting** through use of global and variable “attributes”, both to the meaning/use of data and dependencies among variables
  - **Associated** ISTP/SPDF structuring and metadata **guidelines** are critical to Heliophysics usability and are applicable beyond data in CDF
- **APIs to SPDF system capabilities and data**
  - External software and services can leverage SPDF data/services (such as AMDA, Autoplot, IDL, Python libraries)

Basic Definitions: What is CDF?

- Common Data Format (CDF)
  - Self-describing data format for the storage and manipulation of scalar and multidimensional data in a platform- and discipline-independent fashion
  - Actual data format which CDF utilizes is intended to be completely transparent to the user and accessible through a consistent set of interface routines
    - Programmers are not burdened with performing low level I/O's to physically format and un-format data files
    - Built-in compression capability and transparent decompression
    - Library core is pointer logic that maps to/from block data implementation
    - CDF includes an internal checksum to ensure integrity
- Software distribution includes C, Java, Perl and Fortran APIs
  - High-level toolkit of utilities for creating, browsing and modifying CDF data to/from a regular text or XML files
  - Support libraries for IDL and MatLab (included in their distributions)
  - Additional CDAWlib distribution includes rich set of IDL procedures

SPDF adding netCDF support

- As a Heliophysics Final Archive
  - Work closely with ICON/GOLD teams, and the relevant Heliophysics Virtual Observatory, to understand planned and actual data products
  - SPDF archival formats are NOT restricted to CDF nor is metadata restricted to follow ISTP/SPDF Guidelines but Guidelines important to services
- Help support creation and use of ICON/GOLD metadata to the ISTP/SPDF guidelines in the netCDF format
  - Use of data/metadata standard generally aids science within mission
  - Use of data/metadata standard generally enables easier community access
  - Use of data/metadata standard enables use of data in generic tools
  - SPDF has extended its SKTEditor tool to read/write netCDF
- Support ingest/distribution of data through CDAWeb
  - Requires SPDF to extend CDAWeb system to read/write data in netCDF
  - Will enable access through existing webservices APIs
  - Multiple ways SPDF can ingest data

Presently Low Priority Directions

- CDF libraries, tools and wrappers
  - Add SWIG.org to support GDL, Octave, etc.; support Excel, WebWinds
  - Get Opendap working with latest CDF versions
  - Add groups, parent-child relationship [complicates generic software]
  - Layer CDF API on HDF-5 as netCDF did [pure Python CDF library better than heavy HDF library]
  - Streaming CDFs
  - Parallel or in-memory compression for higher performance
  - Support UDunits www.unidata.ucar.edu/software/udunits/

- ISTP/SPDF Guidelines
  - Port SKTeditor to Javascript
  - Add naming spaces to attribute names, SPDF_*

- CDFlib (IDL)
  - Add naming spaces for our routines, SPDF_*

Recommended Steps to Put Data into CDF

1. Define and create the CDF structure to receive the data
   - – Create/edit skeleton CDF
     - Use SKTEditor
     - Use Skeletontable and SkeletonCDF programs
   - – Science-driven design
     - »

2. Use one of multiple technical options to add data
   - – In IDL, e.g. use IDLmakeCDF procedures
   - – Use makeCDF tool
   - – Direct writes to CDF using CDF library

N.B. For data in another standard self-describing format, SPDF's translation software can convert data files into CDF

- – E.g. netCDF, HDF, FITS

Examples of CDF Variables

- A simple scalar (e.g. B magnitude) is
  - Dimension 0 and record-varying
    - By convention, time dependence is captured as record variance
- Vector B might be
  - 1 time-dependent/record-varying variable of dimension 1 and size 3
  - OR it could be (not recommended) 3 time-dependent scalars (dimension 0)
- Flux at 10 energies should be
  - 1 time-dependent variable of dimension 1 and size 10
  - Plus an attribute pointing to another variable with numerical values for these 10 energies (even if they don’t vary in time)
  - AND/OR an attribute pointing to another variable with e.g. energy band (time-independent) identifications (for labeling)
- Flux at 10/20 energies and 16/8 pitch angles should be
  - 1 time-dependent variable of dimension 2 and sizes (20,16) (i.e. max dims)

ISTP/SPDF Metadata Guidelines

- ISTP/IACG Guidelines (mid1990s) and subsequent extensions by SPDF define a set of implementation standards for CDFs
  - General file naming conventions
  - Data is time ordered and time-identified; times vary by record
  - Set of required and suggested metadata
  - Variable attributes can point to other variables by name
  - Attributes thus carry information about relationships among variables
  - Variables can carry metadata (e.g. labels for dimensional variables)
- CDAWeb uses the additional concepts of "Skeleton" and "Master" CDFs
  - "Skeleton" CDF is a CDF with structure and metadata defined but no data, so it can be used as a template from which to build a data file
  - "Master" CDF is the use of a "skeleton" CDF to insert supplemental or updated metadata for CDFs in a dataset
- CDAWeb display service (CDAWlib software using IDL) and services are keyed to data and master CDFs that follow the above Guidelines
- Concepts above directly/easily map and can fully apply to data in netCDF
- Plan to systematize Guidelines but keep flexible for missions and CDAWeb

Key Metadata Guidelines

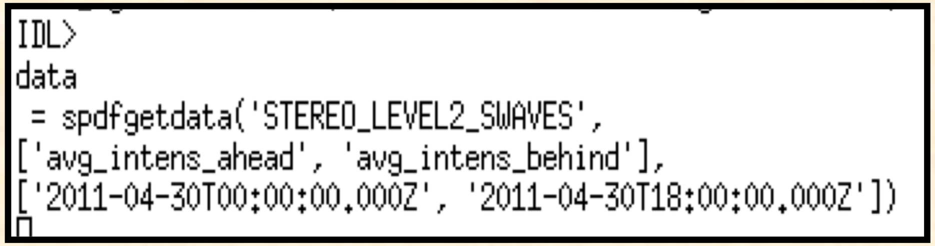
- ISTP/SPDF Guidelines spdf.gsfc.nasa.gov/sp_use_of_cdf.html
- Variable attributes required for automated processing:
  - Catdesc for longer variable description
  - Depend_0 points to time variables
  - Depend_1, 2, 3 point to variables that describe other dimensions
  - Fieldnam short variable name for plots
  - Fillval values indicating missing or bad data
  - Lablaxis/Labl_ptr for axis and column titles
  - Units/Unit_ptr
  - Validmin/max for valid data range
- Will soon add SPASE and DOI global attributes to CDAWeb datasets
- CDF Time variable types
  - CDF_TIME_TT2000 nanoseconds from J2000 in Terrestrial Time in 8 byte integer handles leap seconds and is well-defined; UTC conversion requires up-to-date leap second table (last value stored in CDF header as a check)
  - EPOCH milliseconds from 0AD in 8byte float; usually UTC but not leap seconds
  - EPOCH16 picoseconds from 0AD in two 8byte float; usually UTC but not leap seconds

Reading CDFs

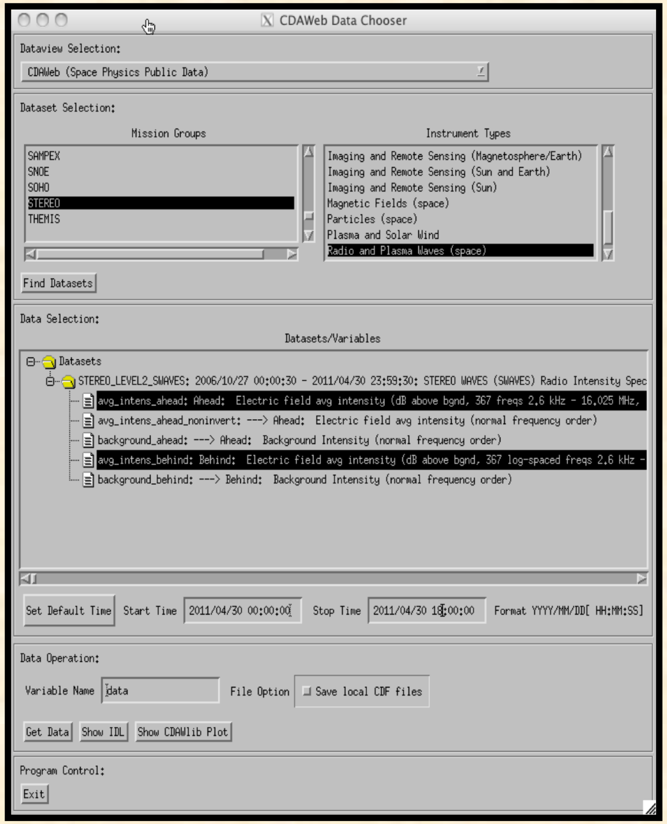
- Data from CDAWeb can be readily displayed/subsetted in CDAWeb
- Use SPDF/CDF Data Translation software to FITS, HDF, netCDF, PDS3, CDFML
- Use CDF toolkit functions (CDFexport) to make ASCII or XML
- Use Autoplot.org
- Project CDFs are handled by various project software routines (e.g. SPEDAS)
- In IDL: for CDFs written to the ISTP/IACG/SPDF Guidelines
  - Value-added CDAWlib functions (that underlie CDAWeb)
  - Value-added display program (CDFX) built on CDAWlib
- Use IDL or MatLab to read/write and manipulate data
  - Using CDF supplied functions included in IDL or MatLab distributions
- Python library and recently-developed generic read routines
- Write custom C, Fortran, Perl, C# or Java programs using CDF APIs

“Fill My (IDL) Array” with Data from CDAWeb

- Load specific CDAWeb data into an IDL structure using
  - @compile_cdaweb
  - spdfgetdata
- GUI to select/load/display data from CDAWeb in IDL
  - spdfcdawebchooser



```
IDL>
data
 = spdfgetdata('STEREO_LEVEL2_SWAVES',
['avg_intens_ahead', 'avg_intens_behind'],
['2011-04-30T00:00:00.000Z', '2011-04-30T18:00:00.000Z'])
```



Dataview Selection:
CDAWeb (Space Physics Public Data)

Dataset Selection:

Mission Groups: SAMPEX, SNOE, SOHO, STEREO (selected), THEMIS

Instrument Types: Imaging and Remote Sensing (Magnetosphere/Earth), Imaging and Remote Sensing (Sun and Earth), Imaging and Remote Sensing (Sun), Magnetic Fields (space), Particles (space), Plasma and Solar Wind, Radio and Plasma Waves (space) (selected)

Data Selection:

Datasets/Variables

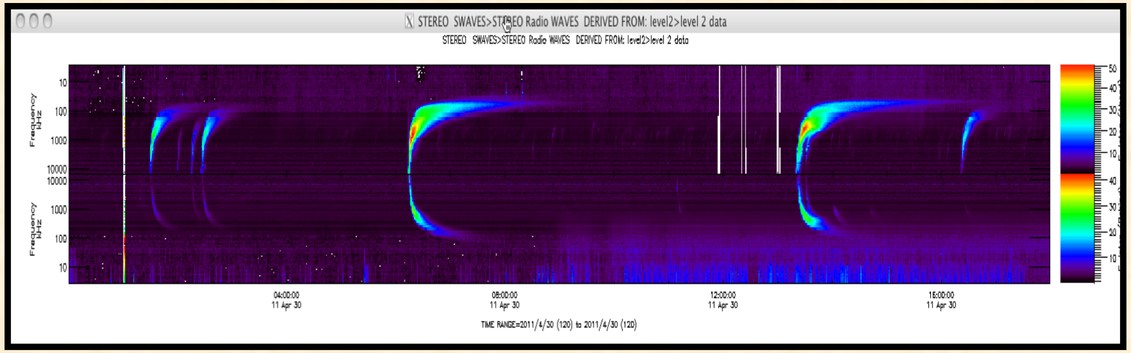
```
Datasets
  STEREO_LEVEL2_SWAVES: 2006/10/27 00:00:30 - 2011/04/30 23:59:30: STEREO WAVES (SWAVES) Radio Intensity Spec
    avg_intens_ahead: Ahead:  Electric field avg intensity (dB above bgnd, 367 freqs 2.6 kHz - 16.025 MHz,
    avg_intens_ahead_noninvert: ---> Ahead:  Electric field avg intensity (normal frequency order)
    background_ahead: ---> Ahead:  Background Intensity (normal frequency order)
    avg_intens_behind: Behind:  Electric field avg intensity (dB above bgnd, 367 log-spaced freqs 2.6 kHz -
    background_behind: ---> Behind:  Background Intensity (normal frequency order)
```

Start Time: 2011/04/30 00:00:00 Stop Time: 2011/04/30 18:00:00 Format YYYY/MM/DD[HH:MM:SS]

Data Operation:

Variable Name: data File Option: □ Save local CDF files

Program Control:



I Have A Data File in CDF: Now What?

- Data from CDAWeb can be readily displayed/subsetted in CDAWeb

»

- Use SPDF/CDF Data Translation software
- Use CDF toolkit functions (CDFExport) to make ASCII or XML
- Use Autoplot (www.autoplot.org)
- Project CDFs are handled by various project s/w routines (e.g. TDAS)
- In IDL: for CDFs written to the ISTP/IACG/SPDF Guidelines
  - Value-added CDAWlib functions (that underlie CDAWeb)
  - Value-added display program (CDFX) built on CDAWlib
- Use IDL or MatLab to read/write and manipulate data
  - Using CDF supplied functions included in IDL or MatLab distributions
- Write custom C, Fortran, Perl, C# or Java programs using CDF APIs

Directly Read Data from CDAWeb into IDL

```
timename='jul_day'    ;name of time variable -- Julian days
start_time = '1998-06-10T00:00:00.0Z'  ;start time
stop_time  = '1998-06-10T23:59:59.0Z'  ;stop time
dt_sec=10.0 ;sec -- bin size in seconds

dataset_id='WI_H0_MFI'    ; CDAWeb dataset ID
vars=[ 'B3F1=Bmag3', 'B3GSE=Bx3,By3,Bz3']   ; CDAWeb variable names with locally assigned names
cdaweb_get_bin, dataset_id,vars,start_time,
                              stop_time,dt_sec,time_name=timename
dataset_id='WI_PM_3DP'
vars=[ 'P_DENS=np3', 'P_VELS=Vxp3,Vyp3,Vzp3',  'P_TEMP=Tp3', 'A_DENS=na3',  $
               'A_VELS=Vxa3,Vya3,Vza3', 'A_TEMP=Ta3']
cdaweb_get_bin, dataset_id,vars,start_time,
           stop_time,dt_sec,time_name=timename,/autobad
```

No more writing code for every dataset.

The Internet functions as a local, easy to use hard drive.

“HAPI” will generalize this to accessing “everything.”

VSO does the same for Solar Data

Notes on CDF and netCDF

- CDF and netCDF come from a common heritage
  - CDF started in 1984 on Modcomp computer and converted to C in 1990; netCDF development started in 1988
  - Self-describing data formats for the storage & manipulation of scalar and multidimensional data in a platform- and discipline-independent fashion
  - Actual data layout utilized is intended to be transparent to the user and accessible through a consistent set of interface routines
    - Noting the underlying layout is described and directly accessible also
  - Interface routines and underlying implementation are different
- Common concepts
  - Variables generally carry data
  - Data can be scalar or multi-dimensional
  - Attributes generally carry metadata (i.e. information about data)
    - Global (file level) attributes
    - Variable level attributes
- PDS defined CDF-A as a version of CDF with ISTP/SPDF Guidelines and 2 SPASE attributes, but no compression or sparse variables
- Converters between CDF, CDFML, netCDF, HDF, FITS

Two CDF Concepts

- Variables generally carry data
  - Variables can vary/not vary with record (typically time) and 0 or more dimensions
  - Variables will also sometimes carry metadata (e.g. labels for dimensional variables)

»

- Attributes generally carry metadata (i.e. information about data)
  - Two levels of attributes
    - Global (file level) attributes
    - Variable level attributes
- Variable attributes can point to other variables
  - Can thus carry information about relationships among variables
  - Can thus use variables to carry metadata (e.g. labels for dimensional variables)
- Some standard attributes are defined in CDF library,
  - Additional standard attributes defined in the ISTP/SPDF Guidelines
  - Projects or communities can/have defined additional standard attributes
    - E.g. Cluster, THEMIS, RBSP (PRBEM extensions)

Notes on CDF and netCDF

- CDF and netCDF come from a common heritage in PLDS CDF
  - Self-describing data formats for the storage & manipulation of scalar and multidimensional data in a platform- and discipline-independent fashion
  - Actual data layout utilized is intended to be transparent to the user and accessible through a consistent set of interface routines

- Common concepts
  - Variables generally carry data
  - Data can be scalar or multi-dimensional
  - Attributes generally carry metadata (i.e. information about data)
    - Global (file level) attributes
    - Variable level attributes

- SPDF has well-tested converter between CDF and netCDF
  - Also ability to output CDF attributes and data in XML (CDFML)
  - Also nominal converters to/form CDF to HDF, FITS, and PDS-3