Fall 2011 AGU Poster SM21A-2001

Common Data Format (CDF) New Time Variable CDF_TIME_TT2000

<http://cdf.gsfc.nasa.gov>

Robert.M.Candey@nasa.gov, Nand Lal, Michael H. Liu, Robert E. McGuire

NASA Goddard Space Flight Center, Heliospheric Physics Laboratory, Code 672, Greenbelt MD 20771 USA

Abstract

The Common Data Format (CDF.gsfc.nasa.gov, a project within the Space Physics Data Facility, SPDF.gsfc.nasa.gov) team has defined a new CDF data type, CDF_TIME_TT2000, to provide a standardized and well-defined time scheme for science data, so that data can be accurately compared across missions and used in the future (without misunderstanding and without requiring multiple software routines for conversion). CDF_TIME_TT2000 is defined as an 8-byte signed integer with a fixed time base of J2000 (Julian date 2451545.0 Terrestrial Time (TT) or 2000 January 1, 12h TT), in units and resolution of nanoseconds, and scaled in Terrestrial Time on the rotating Earth Geoid. Routines to convert between UTC and CDF_TIME_TT2000 are provided. This improves on the existing time variable types, CDF_EPOCH and CDF_EPOCH16, since they have undefined time systems (although usually assumed to be in UTC), and times during leap seconds are conflated with the next second. Support for CDF_EPOCH variables will continue, including optional new metadata attributes available for better describing the EPOCH times as used in each dataset. The upcoming CDF version 3.4 will also improve performance and security, with internal compression routines changed to packages with appropriate software licenses.

Common Data Format (CDF)

Self-describing data format for scalar and multidimensional data in a platform- and discipline-independent fashion

Transparent access to data and metatda through Application Programming Interfaces (APIs) - no low level I/O's required

Built-in support for data compression (RLE, gZIP, Huffman) and automatic data uncompression

Large file support: > 2GBytes

Broad platform support: Windows 95/98/NT/2000/XP/Vista, Linux, Solaris, Mac OS X, QNX, Cygwin, DEC Alpha OSF/1, OpenVMS, DECstation Ultrix, HP 9000 series HP-UX, MinGW, IBM RS6000 series AIX, NeXT Mach, SGI Iris, Power series, Indigo IRIX, Sun SunOS, Solaris
[DEC OSF/1, IRIX, SunOS, HP-UX and IBM AIX only to 2.6]

Many Languages: C, Fortran, Perl, C#, Java, IDL, Matlab

Adopted by many missions, including THEMIS, RBSP and MMS for all public data.

Basis for CDAWeb and supported by Autoplot and other analysis tools.

Problem

Nominally, existing CDF_Epoch time scheme is continuous Gregorian time from 0AD with no leap seconds or defined coordinate system. In practice, it generally holds UTC times, but leap seconds are overloaded onto the first second of the next day. Fortunately, conversion between Epoch time and UTC hours:minutes:seconds is otherwise reversible.

People assume they know what is meant by a specific time measurement, when in fact time scales and units have varied greatly over the years and even now still come in many flavors. It's usually also not obvious whether times refer to the beginning, middle, or end of the data measurement.

Mission data generally have ill-defined and ill-documented time variables, sometimes including leap seconds and sometimes using only UTC at start of mission with no leap seconds afterwards.

Science data should have well-defined times to enable more accurate cross-comparison between missions and as documentation for future archive use.

Using a standard time variable greatly improves ease and accuracy of cross-comparison of data with other missions and development of general data analysis tools (CDAWeb, Autoplot, etc.).

What does adding CDF_TIME_TT2000 mean to me?

- More accurate times
- Easier cross-mission comparison and analysis
- Requires using newer version of CDF and updating software but also provides better performance and security

Time Scales

Time lacks a single standard definition, unlike most metric units. Even ignoring many local times and summer time shifts, time scales include:

Universal Time (UT): based on rotation of Earth (mean solar day)

Ephemeris Time (ET): based on revolution of Earth in orbit around Sun

Atomic Time (AT): based on the quantum mechanics of the atom

UT1: true measure of the Earth's rotation, corrected for polar motion and used in celestial navigation

UT2: UT1 corrected for the seasonal variation due to irregularities in the Earth's rotation

International Atomic Time (TAI): time scale maintained by the BIPM

Coordinated Universal Time (UTC): atomic time scale at rate of TAI but kept within 0.9 s of UT1 by leap seconds as decided by International Earth Rotation Service (IERS) (current civil time)

Terrestrial Dynamical Time (TDT) or Terrestrial Time (TT): like ET with origin on the Earth geoid, (TAI + 32.184 s)

GPS Time (GPST): origin at midnight of 1980 January 5/6 so TAI is constantly ahead of GPS Time by 19 s, and UTC currently behind by 15s

EME1950: seconds since 1950 Jan 1 00:00:00 in Earth Mean Equator.

J2000: seconds since 2000 Jan 1 12:00:00 in Earth Mean Equator

Julian Date (JD): days and fractions since noon January 1, 4714 BC

POSIX or "Unix time": seconds since UTC January 1, 1970, not counting leap seconds. The signed 32bit implementation will overflow on 2038-01-19T3:14:08Z. Unix time often counts then uncounts the leap seconds or slews the computer clock for a bit.

CDF_EPOCH: time from 01-Jan-0000 00:00:00.000, no leap seconds, so 19s behind GPS, 34s behind TAI

Barycentric Dynamical Time (TDB): origin at center of the solar system (often used for ephemeris)

Barycentric Coordinate Time (TCB) at the geocenter

More Time Scales at <http://www.ucolick.org/~sla/leapsecs/timescales.html>

CDF_TIME_TT2000

Storage format = 8-byte signed integer

Time_Base = J2000 (Julian 2451545.0, 2000 Jan. 1, 12h TT)

Resolution = nanoseconds over ±280 years

Time_Scale = Terrestrial Time (TT)

Units = nanoseconds

Reference_Position = rotating Earth Geoid

FILLVAL/Pad = 0x8000000000000000

= 1707-09-22T12:13:15.145224192

Given a current list of leap seconds, conversion between TT and UTC is straightforward:

TT = TAI + 32.184s; TT = UTC + ΔAT + 32.184s,

where ΔAT is the sum of the leap seconds since 1960; for 2009 for example, ΔAT = 34s

Accurate conversions from other time scales may require that data providers use the JPL NAIF SPICE or similar library, but data users will easily be able to convert to UTC time.

Older CDF Time Variables still supported:

CDF_EPOCH (8-byte float) as milliseconds from 0AD in an undetermined coordinate system

CDF_EPOCH16 (two 8-byte floats) as seconds from 0AD and picoseconds within that second.

Recommended Attributes for Time Variables

Time_Base: 0AD, 1900, 1970 (POSIX), J2000, 4714 BC (Julian)

Time_Scale: TT (same as TDT, used by CDF_TIME_TT2000), TAI (same as IAT, TT-32.184s), UTC (includes leap seconds), TDB (same as SPICE ET), EME1950 [default: UTC]

Reference_Position [optional]: Topocenter (local), Geocenter , rotating Earth geoid (used by CDF_TIME_TT2000)

Leap_Seconds_Included [required for UTC only]

"[1961JAN01+1.42282s,1961AUG01-0.05s,1962JAN01+0.47304s,1963NOV01+0.1s, 1964JAN01+1.29427s,1964APR01+0.1s,1964SEP01+0.1s,1965JAN01+0.1s,1965MAR01+0.1s, 1965JUL01+0.1s,1965SEP01+0.1s,1966JAN01+0.47304s,1968FEB01-0.1s,1972JAN01+5.78683s, 1972JUL01+1s,1973JAN01+1s,1974JAN01+1s,1975JAN01+1s,1976JAN01+1s,1977JAN01+1s, 1978JAN01+1s,1979JAN01+1s,1980JAN01+1s,1981JUL01+1s,1982JUL01+1s,1983JUL01+1s, 1985JUL01+1s,1988JAN01+1s,1990JAN01+1s,1991JAN01+1s,1992JUL01+1s,1993JUL01+1s, 1994JUL01+1s,1996JAN01+1s,1997JUL01+1s,1999JAN01+1s,2006JAN01+1s,2009JAN01+1s]"

Required to account for time scales that don't have all 34 (in 2009) leap seconds and for the clocks in various countries that started using leap seconds at different times.

Units [optional]: SI measurement unit: s, ms, ns, ps

Resolution [optional]: using ISO8601 relative time format, 1s

Absolute_Error [optional]: Absolute or systematic error in same units as Units attribute.

Relative_Error [optional]: Relative or random error in same units as Units attribute to specify the accuracy of the time stamps relative to each other. This is usually much smaller than Absolute_Error.

Bin_Location [optional]: relative position of time stamp to the data measurement bin, with 0.0 at beginning of time bin and 1.0 at end. Default is 0.5 for the time at the center of the data measurement. Since clock readings are usually truncated, the real value may be closer to 0.0.

CDF_TIME_TT2000 Routines

C/Fortran:

CDF_TT2000_to_UTC_parts, CDF_TT2000_from_UTC_parts

CDF_TT2000_to_UTC_EPOCH, CDF_TT2000_from_UTC_EPOCH

CDF_TT2000_to_UTC_string, CDF_TT2000_from_UTC_string

cdfleapsecondsinfo shows leap second table

IDL:

CDF_ENCODE_TT2000 to/from a time string

CDF_PARSE_TT2000 parses complex time strings

CDF_TT2000 between CDF_TIME_TT2000 and EPOCH

CDF_EPOCH_COMPARE compares time variables

CDF_EPOCH_DIFF returns difference between two time variables in nanoseconds

CDF_EPOCH_TOJULDAYS converts time variable to Julian Day in IDL style

CDF_LEAPSECONDS_INFO shows leap second table

Matlab:

cdftt2000 construct TT2000 data object (similar to cdfepoch object) from UTC string or number (of nanoseconds)

computett2000 compute UTC date/time to TT2000 time(s)

breakdowntt2000 break down TT2000 time(s) to UTC date/time

encodett2000 encode TT2000 time(s) into UTC date/time string(s)

parsett2000 parse UTC date/time string(s) to TT2000 time(s)

tt2000todatenum convert TT2000, UTC numbers or strings to MATLAB's datenum

cdfleapsecondsinfo shows leap second table

Leap Second Table

Current list of leap seconds is stored in 3 places:

- Array in the source code, updated with new CDF versions
- Separate text file: CDFLeapSeconds.txt
- <http://cdf.gsfc.nasa.gov/html/CDFLeapSeconds.txt>

Users can use their own list of leap seconds by editing the text file or point to a separate file (perhaps mission-specific where leap seconds are not continued to be added), by using the environment variable CDF_LEAPSECONDSTABLE

List updated in new versions of CDF and can be updated by editing the text file for new leap seconds.

Mission Handling of Leap Seconds

Usually unclear what exact time system is used on each mission and software package, nor how leap seconds are handled.

Times written into CDFs are usually assumed to be UTC (and thus have leap seconds), but this assumption is often not valid.

Some Heliophysics missions convert a continuous spacecraft counter on the ground and some regularly upload corrections to onboard time in UTC including leap seconds.

Some missions update their spacecraft clocks over a period of time around the leap seconds (as much as days off), and some slew the spacecraft clock over time, and some increment at once but not always at the time of the leap second.

These complications are too confusing to document in a standard way inside the CDFs but should be covered in mission documentation.

For better accuracy, the Leap_Seconds_Included list should have the times when each leap second was fully added.

ACE, GPS, SOHO, C/NOFS and other missions don't add leap seconds onboard, while WMAP, Stereo, Wind, Geotail, Polar, TRACE, and RXTE do.

For instance, Wind has leap seconds adjusted in ten 100ms steps over 2 hours to minimize impact on attitude sensors and stored commands.

Geotail divides data files at leap seconds; so one file ends at XX:59:60 and the other file starts at XX:00:00.

References

Requirements for Handling Leap Seconds in CDF
<http://cdf.gsfc.nasa.gov/html/leapseconds_requirements.html>

JPL NAIF SPICE time conversion routines
<http://naif.jpl.nasa.gov/naif/>

IERS Conventions
<http://www.iers.org/nn_11216/IERS/EN/Publications/TechnicalNotes/>

IAU Time Commission 31
<http://www.atnf.csiro.au/iau-comm31/activities.php>

USNO
<http://aa.usno.navy.mil/publications/docs/Circular_179.php>

Space-Time Coordinate Metadata for the Virtual Observatory
<http://www.ivoa.net/Documents/latest/STC.html>

Standish, E. M., "Time scales in the JPL and CfA ephemerides", Astronomy and Astrophysics, v.336, p.381-384 (1998)

Fukushima, T., "Time ephemeris", Astronomy and Astrophysics, vol. 294, no. 3, p. 895-906, 02/1995

<http://tycho.usno.navy.mil/systime.html>

<http://tycho.usno.navy.mil/leapsec.html>

<http://www.cv.nrao.edu/~rfisher/Ephemerides/times.html>

<http://www.stjarnhimlen.se/comp/time.html>

<http://sspg1.bnsc.rl.ac.uk/Share/Coordinates/leap_secs_in_sp.htm>

<http://www.ucolick.org/~sla/leapsecs>

<http://leapsecond.com/java/gpsclock.htm>

<http://hea-www.harvard.edu/~arots/time/time_tutorial.html>

<http://www.merlyn.demon.co.uk/leapsecs.htm>