CORONAL DIAGNOSTIC SPECTROMETER

SoHO

CDS SOFTWARE NOTE No. 14

Version 7 19 June 1995

CDS Time Conversion Software

W. Thompson
Applied Research Corporation
NASA Goddard Space Flight Center
Laboratory for Astronomy and Solar Physics
Code 682.1
Greenbelt, MD 20771, USA

William.T.Thompson.1@gsfc.nasa.gov
pal::thompson

1 Time formats

There are two kinds of time used within the CDS project:

1.1 TAI—International Atomic Time

This is the time used by the SoHO spacecraft, and is defined as the number of standard seconds since 0h on 1 January 1958. In the CDS software, TAI time is always expressed as a double precision floating point number.

A distinction needs to be made between spacecraft time and geocentric time. The latter includes a correction for the difference in light arrival time between the spacecraft and the Earth. It has been discussed at SoHO SOWG meetings that all times should be geocentric except when explicitly labelled as spacecraft time. The correction would be based on the spacecraft ephemeris supplied by the FOT. However, the software to calculate this correction has not been written yet.

1.2 UTC—Coordinated Universal Time

This is the time standard on which civil time is based. The main distinction between UTC and TAI, at least since 1 January 1972, is that occasionally a "leap second" is inserted into the UTC time to keep it in sync with the rotation of the earth. (Before 1972 the situation was more complicated.) TAI time has no leap seconds. Therefore, in order to convert between the two kinds of time, one needs to know when leap seconds were added to the UTC time. This information is maintained within the file "leap_seconds.dat" in the directory given by the environment variable TIME_CONV.

Some operating systems, such as VMS and MacOS, keep track only of local time, not UTC. In such cases, one can store the difference in hours (local-UTC) in the file "local_diff.dat" in the same TIME_CONV directory as "leap_seconds.dat" above. For example, for U.S. Eastern Standard Time, this file would contain the value -5.

If the computer is running on GMT rather than local time, one can signify this by adding a second line to the "local_diff.dat" file with the letters "GMT". For example, if one is on the east coast of the United States during winter, and the computer is set up to use GMT, then the file would contain the lines

```
-5
GMT
```

When daylight saving time is in effect this would be changed to reflect this.

It is not necessary for "leap_seconds.dat" and "local_diff.dat" to be in the same directory. One can define TIME_CONV to be a set of directories, using the same format one would use for IDL_PATH. That way, one can have a single "leap_seconds.dat" file which is common among a number of computers (e.g. through NFS or mirror), and a separate "local_diff.dat" file for each computer.

Note that if time differences between observations are needed to an accuracy of a second and the observations are separated by more than a day then the UTC times should be converted to TAI before differencing in case there is an intervening leap-second.

There are three formats that the CDS software uses for UTC, all of which are calendar-based. These are:

Internal: Referred to as "INT" in any routine names. A structure containing the following elements as longword integers:

MJD	The Modified Julian Day (MJD) number. This is defined as the ordinary Julian Day (JD) number minus 2400000.5. The ".5" represents the fact that MJD numbers begin at midnight, whereas JD numbers begin at noon.
TIME	The time of day, in milliseconds since the beginning of the day.

External: Referred to as "EXT" in any routine names. A structure containing the elements, YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, and MILLISECOND as shortword integers.

String: Referred to as "STR" in any routine names. A calendar date in ASCII string format. This format is subdivided into the following subcategories.

CCSDS: A string variable containing the calendar date in the format recommended by the Consultative Committee for Space Data Systems (ISO 8601), e.g.

"1988-01-18T17:20:43.123Z"

It should be noted that the "Z" delimiter at the end of the string explicitly denotes that the time is in coordinated universal time.

ECS: A variation on the CCSDS format used by the EOF Core System. The "T" and "Z" separators are eliminated, and slashes are used instead of dashes in the date, e.g.

"1988/01/18 17:20:43.123"

VMS: Similar to that used by the VMS operating system, this format uses a three-character abbreviation for the month, and rearranges the day and the year, e.g.

"18-JAN-1988 17:20:43.123"

STIME: Based on !STIME in IDL, this format is the same as the VMS format, except that the time is only given to 0.01 second accuracy, e.g.

"18-JAN-1988 17:20:43.12"

Other string types may be added in the future.

2 Procedures

The following procedures are used to convert between the different time formats:

OBT2TAI	Converts the 6-byte SoHO on-board time (OBT/LOBT) to TAI format.
TAI2UTC	Converts TAI times to any one of the CDS UTC formats.
UTC2TAI	Converts any one of the CDS UTC formats to TAI.
INT2UTC	Converts internal time to either external or CCSDS format.
UTC2INT	Converts either external or CCSDS time to internal format.
STR2UTC	Converts string time to either internal or external format.
UTC2STR	Converts either internal or external time to string format.

In addition, there are also the following routines:

UTC2DOW	Calculates the day of the week from any of the UTC formats.
UTC2DOY	Calculates the day of the year from any of the UTC formats.
GET_UTC	Gets the current UTC date/time from the system clock in any one of the CDS UTC formats.
CHECK_INT_TIME	Checks the internal time for consistency.
CDS2JD	Calculate full Julian day equivalent of CDS date/time.
LOCAL_DIFF	Returns the difference in hours between local and UTC time.

The following are essentially internal routines:

DATE2MJD	Converts dates to MJD numbers.
MJD2DATE	Converts MJD numbers to dates.
GET_LEAP_SEC	Gets the MJD numbers for days with leap seconds.

The following are used in the UTPLOT programs and take no account of leap seconds:

UTC2SEC	Converts CDS UTC time format to seconds since MJD=0.
SEC2UTC	Converts seconds since MJD=0 to CDS UTC format.

Note that where a routine uses a particular form of time, any of the specified formats for that time may be used as input to the routine–the code can distinguish the formats. Thus, in the routine STR2UTC the input variable may be any recognized string format, including CCSDS or ECS, plus some variations on these. In UTC2TAI the input can be any of the UTC formats mentioned above (INT, EXT, STR).

If there is a choice on output, then this is controlled by keywords. For instance, when using INT2UTC the keywords /EXTERNAL, /CCSDS or /ECS would be used to determine the format of the UTC output.

Appendices

A Notes on string formats

The two supported string date/time formats, CCSDS and ECS, are very similar in the way that they represent the date. Each uses a four digit year, followed by a numerical month, and then a day of the month. There are certainly other ways to represent dates. For example, some common representations are:

January 18, 1988
18-JAN-1988
01/18/88
18/01/88

There are a number of reasons to use the CCSDS/ECS style scheme in preference to any of the above variations. Some of these reasons are as follows:

- In the United States it is popular to use a month-day-year format, while in Europe a day-month-year format is more common. Thus, a date such as 01/02/95 could be interpreted as either January 2, 1995 or as 1 February 1995. Using instead a year-month-day format eliminates this source of confusion.
- Starting the date off with a full four-character year makes it self-evident what the date convention is. A date such as 01/02/03 could be interpreted as January 2, 2003, as 1 February 2003, or as 3 February 2001.
- Using numerical months instead of character months, together with a year-month-day convention, means that dates sorted by date are also sorted alphabetically, and vice-versa. This can be useful in software development.

The CDS string parsing routine STR2UTC is more flexible in the kinds of strings that it accepts than UTC2STR is in formatting them. For example, it can accept strings such as "88/01/18". However, it is very conservative in how it attempts to guess what order is used for the year, month, and day. Normally, it assumes that the date is in the format year-month-day. Automatic detection of the two other possibilities, day-month-year or month-day-year, is supported only if the month is given as a character string. For example, the string "18-Jan-1988" is automatically recognized. Both the VMS and STIME formats satisfy the above conditions, so they have been added as officially supported formats. In other cases, the day-month-year and month-day-year conventions are supported through the keywords /DMY and /MDY.

Note: Due to popular demand, it was decided that the STR2UTC routine should assume that when the month is given as a character string, then the year is the *last* parameter in the date, and not the first as before. Thus, the software will now automatically recognize dates of the form "18-Jan-88" (which it didn't before), *but will no longer automatically support dates such as "88-Jan-18"*. This is a significant departure from the previous behavior. There are two ways that this can be overcome—i.e. two ways that dates with a character month and beginning with a year can be supported:

1. Give the year with all four digits.
2. Use the keyword /YMD.

Generally speaking, it is safest to always use a four digit year.

The following table illustrates various date formats which are supported. It is not meant to be an exhaustive list.

1995-02-15	
1995/02/15	
95/02/15	
1995-046	(Day-of-year variation)
15-Feb-95	
1995-February-15	
15-FEB-1995	
02/15/95	(Only with /MDY keyword)
15/02/95	(Only with /DMY keyword)
95-Feb-15	(Only with /YMD keyword)