Common Calendar Introduction and Scope

Local Time Timestamp System

Brooks Harris Version 14 2025-02-02

Common Calendar Timestamp System (CCT) is a timekeeping framework that provides a uniform set of local timescales. The proposal is made up of nine tightly coupled specifications including three types of timestamp may be expressed as a binary format for efficient machine interchange or as a human readable character format.

1. **Common Calendar Date and Time Terms and Definitions** – Comprehensive collection of terms and definitions to clarify use of UTC and local time and provide common lexicon for the Common Calendar specifications.
2. **Common Calendar TAI-UTC API** – Data types and operations for automated access to dynamic IERS TAI-UTC (leap-second) information.
3. **Common Calendar Time Zone API** – Data types and operations for automated access to dynamic IANA Time Zone Database (TzDb) information.
4. **Common Calendar YMDhms API** – Data types and algorithms for conversion between seconds and YMDhms representation with support for leap-seconds.
5. **Common Calendar Conventional timestamp (CCTC)**
6. **Common Calendar Media timestamp (CCTM)**
7. **Common Calendar Enhanced timestamp (CCTE)**
8. **Common Calendar Timestamp API** - User level timestamp interface and manipulation functionality.
9. **Geostamp** - Combines Common Calendar Timestamps with geographic coordinates for traceability provenance and use in 4-GIS applications,

It is hoped Common Calendar may provide a starting point for formal standardization that finds its way to international acceptance. Comments, ideas, and help improving these specifications are welcomed.

Table of Contents

Notation

“YMDhms” is shorthand for year-month-day hour:minute:second representation.

ISO 8601 representation is supplemented with suffixes (UTC) and (TAI), for example 1970-01-01 00:00:10 (TAI) = 1970-01-01T00:00:00 (UTC).

“UTC1970” is shorthand for 1970-01-01 00:00:10 (TAI) = 1970-01-01T00:00:00 (UTC).

1 Introduction

Common Calendar has been developed to resolve the ambiguities and inconsistencies in current timekeeping specifications to realize a practical engineering solution for accurate local timekeeping.

Common Calendar defines a set of local timescales together with technical specifications for interoperable timestamps and associated calculation formulas. By carefully defining the environment and parameters it is possible to create a self-contained environment that is both reverse compatible to existing timekeeping technologies and provides a set of deterministic local timescales.

The integrity of the design has been verified by the CCT c/c++ reference implementation.

The Common Calendar specifications are intended to be interpreted as generic APIs suitable for implementation on many platforms. The specifications are written largely in terms of c/c++, adopting the c/c++ language syntax for data types and methods to avoid potential ambiguity. CCT "(for "Common Calendar Timescales") is a c/c++ reference implementation of the Common Calendar specifications. The specifications cite the CCT code directly to show unambiguous logic and facilitate implementation.

The author hopes Common Calendar may provide a starting point for discussion, refinement, and formal standardization that may find its way to adoption.

2 Common Calendar

Common Calendar presents a comprehensive design to overcome the ambiguities of interpretation of standards and the indeterminate counting methods of discontinuities.

There are two sets of dynamic metadata that must be available to provide accurate civil timekeeping:

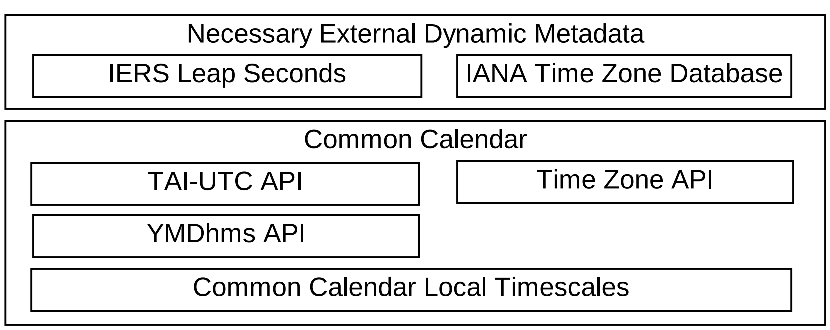
- International Earth Rotation and Reference Systems Service (IERS) leap-seconds
- IANA Time Zone Database Time Zone, local UTC-offset, and Daylight Saving Time (DST) rules

The introduction of leap-seconds is unpredictable because of the many factors that affect earth's rotation. Use of time zones, UTC-offsets and Daylight Saving Time rules are subject to political and cultural change.

Common Calendar addresses these topics by defining APIs to provide access to these two sets of dynamic factors. TAI-UTC API and Time Zone API aggregate the dynamic metadata of leap-seconds, time zones, local UTC-offset, and DST rules necessary to fully describe local date and time.

System clocks typically operate on a seconds basis in some form and conversion between seconds and YMDhms values are fundamental to most implementations. The YMDhms API defines algorithms for these calculations with native support for leap-seconds as interfaced to the TAI-UTC API.

Together these APIs collect all the necessary raw information into normalized data representations and algorithms to represent local time in YMDhms form. Common Calendar Local Timescales define a matrix of local timescales consistent with these APIs, with counting methods, rules, and policies to enable deterministic civil timekeeping.



3 Three Timestamp Types

Common Calendar provides three types of timestamps: Conventional, Media and Enhanced. Each may be expressed as a binary format or as a human readable character format for interfacing and interchange.

The binary formats provide compact binary data to facilitate fast transfer, efficient interchange, and economical storage. The binary design is intended for binary systems, protocols, and languages, such as embedded systems, time dissemination protocols, and c/c++, while offering compatible expression on other platforms and formats such as Java and XML

The character formats are ASCII based machine-readable representation of date and time in a YMDhms form to impart familiar meaning to human users.

The binary formats act in concert with the character formats to provide comprehensive description of local date and time with symmetrical conversion between two representations.

Each of the types include a Timestamp API to provide convenient user-level access to the Common Calendar capabilities. A selection of key operations illustrate the nature of the Timestamp API:

- SetYMDhmsd - Populate a binary format as UTC accurate local date and time-of-day from YMDhmsd user input values and parameters with input value validation
- SetFrom1970Seconds - Populate a binary format as UTC accurate local date and time-of-day from seconds and decimal fraction values and user input parameters
- SetIntervalFromSecondsFrac - Populate a binary format as an interval (duration) from SecondsFrac_st user input values and parameters
- AddUnits - Add units to binary format values.
- Difference - Compute the difference, the duration of the time interval, between two time points (A and B).
- SetCcfFromCCbf - Construct a character format string from binary data and metadata
- ParseCcfSetCCbf - Parse character string and populate a binary format
- Calendar Operations - CCT Calendar provides month and year calendar representation

4 Common Calendar Conventional Timestamp (CCTC)

CCTC presents a standardized form of traditional timestamps. CCTC can be used in most cases where conventional POSIX-time based timestamps are used such as operating systems, file systems, network time, databases and desktops. The binary format is made up of seconds-since-1970 and decimal-fractions-of-seconds together with time zone metadata. An optional extension can support labeling of video and audio media. The character format reflects the binary data in human readable form:

- CCTC Binary Format (CBFC) provides a compact binary data form suitable for fast transfer, efficient interchange, and economical storage.
- CCTC Character Format (CCFC) is an ascii format in YMDhms form.
  (The example below shows a timestamp with millisecond resolution.)

The two formats are symmetrically convertible to one another.

CCTC Binary Format (CBFC)

Seconds-Since-1970 (POSIX time_t in seconds) **Decimal Fractions of Seconds**	**UTC-offset** (POSIX gmtoff)	**Time Zone** (TzDb time zone identity)

CCTC Character Format (CCFC)

`D2024-03-10T03:00:00m999U-06Zamerica/denverV2024aMuX`

See Common Calendar Conventional Timestamp specification.
See CCT\CCTCLib\CCTC.h, CCT\\include\CCbfC.h, CCT\CCTCLib\CCbfC.cpp, class CCbfC
See CCT\\include\CCcfC.h, CCT\CCTCLib\CCcfC.cpp, class CCcfC
See CCT\\include\CCctC.h, CCT\CCTCLib\CCctC.cpp, class CCctC

5 Common Calendar Media Timestamp (CCTM)

CCTM is optimized for labeling video and audio media. The binary format has a compound counter composed of the date as day-number and time-of-day as seconds-since-start-of-day which is an efficient method of representing SMPTE timecode. CCTM includes complete media metadata and supports frame accurate timecode as per the SMPTE ST12-2 UTC Aligned Timecode algorithms. It supports many video and audio rates.

- CCTM Binary Format (CBFM) provides a compact binary data form suitable for fast transfer, efficient interchange, and economical storage.
- CCTM Character Format (CCFM) is machine readable ascii format in YMDhms form.
  (The example below shows the classic NTSC 30000/1001 video rate with drop-frame count mode.)

The two formats are symmetrically convertible to one another.

CCTM Binary Format (CBFM)

Local Date	**Time-of-day**	**UTC-offset**
Day Number since 1970 Leap-seconds Time Zone Identity	Integer unit count from start-of-day Rate - video, audio, or clock Video and/or audio metadata	(POSIX gmtoff)

CCTM Character Format (CCFM)

`D2024-10-05T12:00;00,29U-06Zamerica/denverV2024aMuX`

See Common Calendar Media Timestamp specification.
See CCT\ \include\CBFM.h, CCT\\include\CCbfM.h, CCT\CCTMLib\CCbfM.cpp, class CCbfM
See CCT\ \include\CCFM.h, CCT\\include\CCcfM.h, CCT\CCTMLib\CCcfM.cpp, class CCcfM
See CCT\\include\CCctM.h, CCT\CCTMLib\CCctM.cpp, class CCctM

6 Common Calendar Enhanced Timestamp (CCTE)

The CCTE “Enhanced” timestamp includes extended metadata to represent all aspects of TzDb data which are typically not included in timestamps used in common practice. This includes UTC-offset transitions (STDOFF), DST transitions (dstoff), POSIX abbreviated name transitions and leap-second transitions.

CCTE is designed to facilitate calculation of time marks within the current day without need for access to external metadata for systems that may require inexpensive timestamp receivers. It is also helpful in development because it reflects all the metadata thus revealing any inaccuracies in algorithms and other formats. It may form the basis of a "forensic tracing" application, useful for "legal time" traceability. Like CCTM it can support video and audio media labeling.

- CCTE Binary Format (CBFE) provides a compact binary data form suitable for fast transfer, efficient interchange, and economical storage.
- CCTE Character Format (CCFE) is machine readable ascii format in YMDhms form with comprehensive metadata.

The two formats are symmetrically convertible to one another.

CCTE Binary Format (CBFE)

Local Date	**Time-of-day**	**Daylight Saving**
Day Number since 1970 Leap-seconds Leap-second transitions Time Zone Identity UTC-offset (TzDb STDOFF) UTC-offset transitions	Integer unit count from start-of-day Rate - video, audio, or clock Video and/or audio metadata	DST Bias (dstoff) DST transitions **POSIX Abbreviated name** Abbreviated name transitions

CCTE Character Format (CCFE)

`D2024-03-10T03:00:00U-06Zamerica/denverAmdtV2024aL27S01t01a02cMuX`

See Common Calendar Enhanced Timestamp specification.
See CCT\ \include\CBFE.h, CCT\\include\CCbfE.h, CCT\CCTELib\CCbfE.cpp, class CCbfE
See CCT\ \include\CCFE.h, CCT\\include\CCcfE.h, CCT\CCTELib\CCcfE.cpp, class CCcfE
See CCT\\include\CCctE.h, CCT\CCTELib\CCctE.cpp, class CCctM

7 Intervals and Timers

While the primary objective of Common Calendar is support of UTC accurate local date and time-of-day timestamps and most of the specifications and implementation details are devoted to that purpose, it also provides means of representation of other useful configurations of timestamps:

- time interval less than 24 hours (< 86400 seconds)
- time interval 24 hours or greater (>= 86400 seconds)
- time point less than 24 hours (< 86400 seconds) with an abstract zero origin having no relation to any date; a timer.
- time point equal or greater the 24 hours (>= 86400 seconds) with an abstract zero origin having no relation to any date; a timer.
- time point with UTC accurate local date with the time portion having no relation to the date, that is; the time portion (hms) is not time-of-day of the date but rather a time point less than 24 hours (< 86400 seconds) with an abstract zero origin having no relation to the date; a timer starting sometime during that date.

These configurations allow intervals and timers to be represented by the same basic binary and character formats and operations used by the full UTC accurate form.

8 Common Calendar Date and Time Terms and Definitions

Any standard requires a comprehensive terms and definition specification. This is difficult in timekeeping because so many terms have been used throughout the literature. The specifications of UTC itself are distributed through several standards organizations using various nomenclature leading to some confusion and implementation inconsistency.

To address this challenge Common Calendar has collected terminology in the *Date and Time Terms and Definitions* document. It consolidates the descriptions of TAI and UTC, collecting the many details of the dispersed international standards and clarifying the chain of provenance amongst those specifications. The basic timekeeping definitions are drawn from ISO 8601, its underlying IEC specifications, the *BIPM Brochure* (Bureau International des Poids et Mesures) The International System of Units (SI)) and the *BIPM International vocabulary of metrology* (VIM).

See *6 Common Calendar Date and Time Terms and Definitions*

9 Metadata: Leap-seconds, Time Zones and Daylight-Saving Time

There are two sets of dynamic metadata that must be available to provide the necessary and sufficient information for accurate civil timekeeping:

- TAI-UTC (leap-seconds) values
- Time Zone and Daylight Saving Time rules

The introduction of leap-seconds is unpredictable because of the many factors that affect earth's rotation. Time zones and observation of Daylight Saving Time are politically and culturally selected and subject to change. This necessitates timely access to updated information for both factors.

This unavoidable unpredictability limits the range of possible prediction. No timekeeping system can accurately calculate a future date-time beyond either the expiration date of the current leap-second information or the current valid time zone and DST rules, which ever is earlier. Calculations in the past require access to the full history of leap-seconds, time zone changes, and DST occurrences to represent local time across the full extent of the timescale.

Common Calendar addresses these topics by defining APIs to provide access to these two dynamic factors. TAI-UTC API and Time Zone API aggregate the dynamic metadata of leap-seconds, zones, and DST rules necessary to fully describe local date and time.

10 Common Calendar TAI-UTC API - Leap-second Metadata

TAI-UTC, the integral second difference between TAI and UTC, is at the foundation of modern civil timekeeping. Obtaining and maintaining this information is critical to accurate timekeeping but no formal mechanism for automatic access to this metadata has been adopted. The Common Calendar TAI-UTC API specification addresses this obvious missing link in timekeeping technologies.

See *6 Common Calendar TAI-UTC API*
See `CCT\CCTTaiUtcApiLib`

11 Common Calendar Time Zone API - Time Zone Metadata

"Local time", often called "civil time", refers to time scales established by law or custom in some jurisdiction at some geographic location for legal, commercial, and social purposes[1]. Not all systems fully support accurate and deterministic representation of local time. Common Calendar Time Zone API supports local time based on data supplied by IANA Time Zone Database (TzDb).

See *Common Calendar Time Zone API*
See `CCT\CCTTzDatabaseApiLib`

12 Common Calendar YMDhms API – Seconds to YMDhms Conversion

Common Calendar specifies the YMDhms API. It specifies data types and operations to perform the conversion between seconds and YMDhms with leap-second compensation using the TAI-UTC API for access to the required dynamic TAI-UTC leap-seconds information.

See *Common Calendar YMDhms API*
See `CCT\CCTTaiUtcApiLib`

13 Geostamp

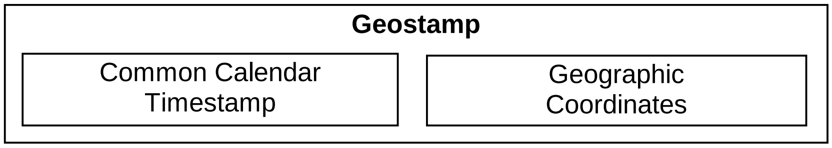
The Common Calendar Timestamp (CCT) specification has been extended to include geographic coordinates to create a Geostamp. The Geostamp specification was developed in collaboration with Son Voba at Microsoft and Sync-n-Scale to support "tractability provenance".

A Geostamp consists of geographic coordinates and a CCT timestamp. Geostamps are technically accurate, making them suitable for general and legal purposes where time recording is used for tracking and auditing and a wide range of spatial-temporal geographic information systems (4D GIS) applications in machine learning, artificial intelligence, data analytics and blockchain distributed ledgers.

---

[1] There are generally two types of time zone in use: civil (land) and nautical. Civil time zones are usually designated as a time offset from the UTC applicable to some territory on land. Nautical time zones are specified by longitude for purposes of navigation at sea. Common Calendar is concerned only with civil time and does not address nautical time zones. (see *Date and Time Terms and Definitions, 11 Civil Time (Local Time*)

Like CCT, Geostamps can be formed in either a binary or character format. The binary format supports efficient machine interoperability while the character format is human readable making their meaning accessible to those less familiar with the intricacies of timekeeping and geographic representations.



14 Acknowledgments

The Science of Time Symposium provided an informative and stimulating event for the presentation of Common Calendar that further informed the work. The author wants to thank all the participants for their comments and inspiration.

Thanks to the many contributors on the LEAPSECS discussion list.

Thanks to the contributors to IANA Time Zone Database.

Thanks to all participants at SMPTE and AES.

Special thanks to Judah Levine, of NIST, for his most valuable guidance.

Thanks to my collaborator Stephen Scott for his expertise in SMPTE timecode.

15 References

Common Calendar Timestamp System (CCT)
common-calendar.org

Date and Time Terms and Definitions
https://common-calendar.org/Common_Calendar_Specification/Common_Calendar_Date_and_Time_Terms_and_Definitions_V38_2023_02_24.pdf

BIPM The International System of Units (SI) 8th edition 2006 (commonly called the SI Brochure)
http://www.bipm.org/en/publications/si-brochure/

BIPM JCGM 200:2012, International vocabulary of metrology – Basic and general concepts and associated terms (VIM)
http://www.bipm.org/en/publications/guides/vim.html

ISO 8601 2004-12-01, Data elements and interchange formats — Information interchange — Representation of dates and times
http://www.iso.org/iso/iso8601

ISO 8601-1:2019, Date and time, Representations for information interchange – Part 1: Basic rules
https://www.iso.org/standard/70907.html

ISO 8601-2:2019, Date and time, Representations for information interchange – Part 2: Extensions
https://www.iso.org/standard/70908.html

Recommendation ITU-R TF.460-6 (02/02), Standard-Frequency and Time-Signal Emissions
http://www.itu.int/rec/R-REC-TF.460/en

Bureau International des Poids et Mesures (BIPM)
https://www.bipm.org/en/

International Earth Rotation and Reference Systems Service (IERS), IERS Earth Rotation Center
https://www.iers.org/IERS/EN/Home/home_node.html

Society of Motion Picture and Television Engineers (SMPTE)
https://www.smpte.org/

SMPTE ST 12-1:2014, Time and Control Code
http://ieeexplore.ieee.org/document/7291029/

Conversion between SMPTE hh:mm:ss:ff Time Code and Frames
http://edlmax.com/SMPTETimeCodeConversion.htm

Accurate x/1001 Rate ST12 Time-of-day Calculation
https://edlmax.com/WSTS_2024/Accurate_1001_Time_Of_Day_Calculation_BEH_V5_2024-04-08.pdf

ST 2059-2:2015 - SMPTE Standard - SMPTE Profile for Use of IEEE-1588 Precision Time Protocol in Professional Broadcast Applications
http://ieeexplore.ieee.org/document/7291608/

ST 2059-1:2015 - SMPTE Standard - Generation and Alignment of Interface Signals to the SMPTE Epoch
http://ieeexplore.ieee.org/document/7291827/

1588-2019 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems
https://ieeexplore.ieee.org/document/9120376

LEAPSECS -- Leap Second Discussion List
https://pairlist6.pair.net/mailman/listinfo/leapsecs

The Science of Time Symposium
https://sot2016.cfa.harvard.edu/