Common Calendar Introduction and Scope

Local Time Timestamp System

Brooks Harris Version 13 2024-05-20

Abstract

A new and comprehensive approach is needed to achieve uniform civil timekeeping across the world. We propose a timekeeping framework called Common Calendar which provides a uniform matrix of local timescales. The proposal is made up of eight specifications:

1. *Common Calendar Date and Time Related Terms and Definitions* provides a comprehensive glossary for the set of specifications, collecting terms from many sources to clarify the use of the UTC (Coordinated Universal Time) standards in general and application to Common Calendar in particular.
2. *Common Calendar TAI-UTC API* (Application Programming Interface) provides mechanisms for automatic acquisition of TAI-UTC (leap-second) history, announcement, and expiration metadata to fill the obvious missing link between UTC time dissemination and the TAI (International Atomic Time) timescale.
3. *Common Calendar Time Zone API* describes the CCT interface to IANA Time Zone Database (TzDb) metadata and Windows CLDR time zones.
4. *Common Calendar YMDhms API* details the calculations necessary to perform conversion between seconds and accurate UTC compliant YMDhms representation.
5. *Common Calendar Binary Format* defines a binary data format for compact carriage of local timescale date, time and metadata.
6. *Common Calendar Character Format* provides a comprehensive YMDhms character representation. It augments ISO[1] 8601[2] recommendations with leap-second, UTC-offset and DST metadata to provide symmetrical reflection of CBF binary data in character form.
7. *Common Calendar Timestamp API* provides user-level timestamp interface and manipulation functionality.
8. *Geostamp* combines Common Calendar Timestamp with geographic coordinates.

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

[1] ISO: International Organization for Standardization
[2] ISO 8601 2004-12-01, Data elements and interchange formats - Information interchange - Representation of dates and times

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 Foreword

Timekeeping technologies are good enough for everyday use but flaws remain that prevent accurate representation of date and time in all cases. Inconsistent treatment of the discontinuities introduced by leap-seconds and Daylight Saving Time lead to inconveniences, inaccuracies, errors, and threat of system failure. A new and comprehensive approach is needed to achieve reliable uniform timing across the world.

Many topics of standardization contribute to incompatible system implementation:

- Varied terminology through the literature and the complexity of the UTC standards risks misinterpretation.
- The absence of standardized formatting and dissemination protocols of TAI-UTC information is an obvious missing link that thwarts automated accounting of leap-seconds.
- Time zones and Daylight Saving Time rules are not well defined and subject to political policy that may introduce unexpected changes.
- The history of timing specifications has led to systems with mismatched origins and counting methods.
- No standardized algorithm for conversion between seconds and YMDhms representation with native support for leap-seconds has been adopted.
- No standardized compact binary data type conveying the necessary and sufficient information for consistent local date and time representation has been proposed.
- Specifications for YMDhms character representation are varied and incomplete.

There are several issues with traditional timekeeping that prevent accurate timestamps in all cases.

Traditional timekeeping systems:

- Operate on 86400 second calendar days, while UTC may have 86401 or 86399 second days. This discrepancy underlies all difficulties with traditional systems. It cannot be resolved within the context of the traditional time format because there is no way to represent a positive leap-second. This results in at least one indeterminate timestamp value at or near the leap-second.
- Complex and arguably vague specifications of UTC and mismatched or incomplete specifications of traditional systems contribute to misunderstanding and implementation errors and omissions.
- Specifications of local time are not well defined, and widely deployed systems use different schemes (Windows v.s. Tz Database, for example).
- The common practice of introducing leap-seconds in the YMDhms encoding of local time simultaneous with introduction at UTC generates a discontinuity in midday of the local YMDhms sequence. This creates a complex asymmetrical matrix of local YMDhms encodings.
- No binary timestamp format has the necessary and sufficient information to fully and deterministically describe local time.
- YMDhms character encoding formats, often relying on ISO 8601 guidelines, have incomplete information to deterministically represent local time.

These factors combine to a witches brew of complexity and controversy with resulting inaccuracies amongst systems. Despite their imperfections, none of these conventions or practices can be changed without risk of upsetting vast numbers of devices and applications.

- It's been announced by the BIPM that the definitions of UTC and treatment of leap-seconds may be substantially changed, but probably not until 2034. This leaves the mismatch between 86400 second day systems and UTC with leap-seconds in place for the foreseeable future, certainly a decade. In any event UTC must be supported to interpret existing timestamps correctly.
- The systems and applications using the 86400 second day cannot be suddenly replaced. The huge scale of deployment of these systems makes any near-term upgrades entirely infeasible.

- The practice of introducing leap-seconds in midday of local time simultaneous with UTC cannot be withdrawn without upsetting the many systems designed to expect it.
- Definitions of local time will remain subject to change by local custom or political influence.
- The practice of "leap-second smear" used to avoid possible system disruption caused by the mismatch between 86400 second day systems and UTC will probably continue and expand, further compromising timestamp accuracy.

If a world-wide accurate and deterministic timekeeping system is to emerge it will only be realized through a long process of adoption and migration to some proposed and widely accepted plan. That plan must include an underlying design capable of supporting legacy timekeeping systems while providing for the complete and deterministic representation of local time that accounts for both UTC leap-seconds and local time UTC Offset and Daylight Saving rules. Any plan must be backed with rigorous technical specifications, attract wide adoption, and ultimately be standardized by international standards bodies.

The author hopes Common Calendar may provide a starting point for discussion, refinement, and formal standardization that may find its way to international acceptance. 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 takes the form of defining 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. This includes a counting mode that eliminates midday YMDhms discontinuities as described in *Common Calendar Local Timescales.*

The integrity of the design has been verified 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 "(or "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

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:

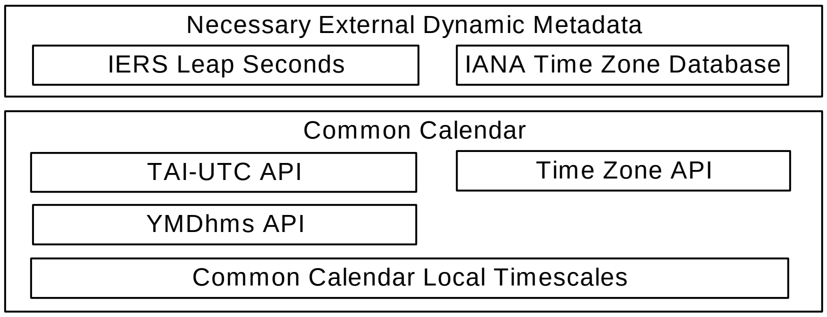
- Leap-seconds
- Time Zone, local UTC-offset, and Daylight Saving Time (DST) rules

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.



Common Calendar provides two formats of data representation for interfacing and interchange:

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

The two formats are symmetrically convertible to one another.

Common Calendar Binary Format (CBF)

Time-of-day	**Local Date**	**Daylight Savings**
Rate	Day Number since UTC1970	DST Bias
Integer count	Leap -seconds	DST State
	Time Zone Name	DST Change Day
	UTC Offset	DST Change Time
	Leap Second metadata	DST Count Mode
	Leap Second Count Mode	

Common Calendar Character Format (CCF)

```
D2015-06-30T19:59:60U-04Zamerica/new_yorkAedtV2021aL25*S01cMuX
```

Common Calendar Specifications present eight tightly coupled specifications which together overcome the ambiguities of standards interpretation and indeterminate accounting of YMDhms discontinuities:

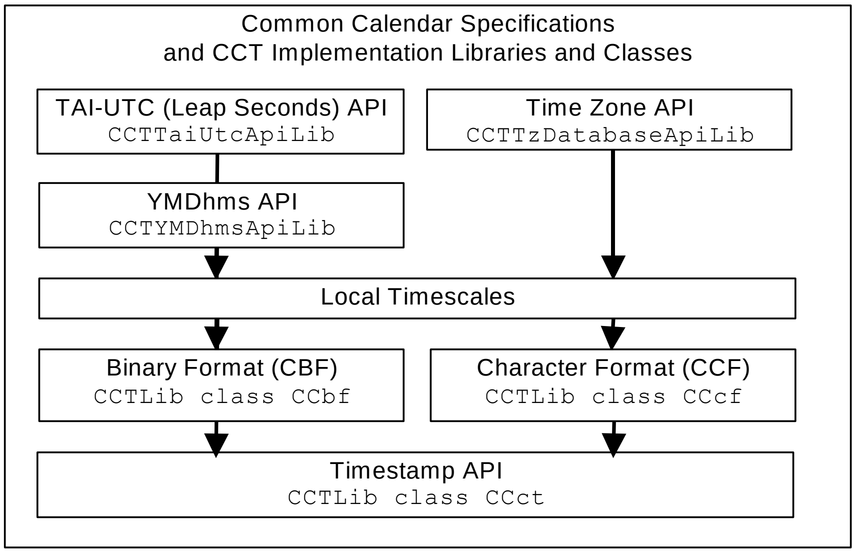
1. **Common Calendar Date and Time Terms and Definitions** – Comprehensive collection of terms and definitions to clarify use of UTC and local time and provide common lexicon for the Common Calendar specifications.
2. **Common Calendar TAI-UTC API** – Data types and operations for automated access to dynamic IERS TAI-UTC information.
3. **Common Calendar Time Zone API** – Data types and operations for automated access to dynamic IANA Time Zone Database information.
4. **Common Calendar YMDhms API** – Data types and algorithms for conversion between seconds and YMDhms representation with native support of leap-seconds.
5. **Common Calendar Binary Format (CBF)** – Compact binary data format to facilitate fast transfer, efficient interchange, and economical storage.
6. **Common Calendar Character Forma**t **(CCF)** - Ascii character based machine-readable representation of date and time in a YMDhms form to impart familiar meaning to human users.
7. **Common Calendar Timestamp API** - User level timestamp interface and manipulation functionality.

8. **Geostamp** - Combines Common Calendar Timestamp with geographic coordinates for traceability provenance and use in 4-GIS applications,

While the primary objective of Common Calendar is support of UTC accurate local date and time-of-day 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. These specifications have been verified in the CCT c/c++ reference implementation. The layers of the specifications have been carefully isolated to separate libraries and classes, illustrated in the following graphic.



3 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 Related 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*

4 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 Daylights 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.

5 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`

6 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[3]. 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`

7 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`

8 Common Calendar Binary Format

Common Calendar Binary Format (CBF) provides a compact binary data format to facilitate fast transfer, efficient interchange, and economical storage. The CBF design is intended for binary systems, protocols, and languages, such as embedded systems, time dissemination, and c/c++ , while offering compatible expression on other platforms and formats such as Java and XML.

CBF acts in concert with Common Calendar Character Format (CCF) to provide comprehensive description of local date and time with symmetrical conversion between the binary and character based YMDhms representations.

See *Common Calendar Binary Format*
See `CCT\CCTLib\CBF.h, CCbf.h, CCbf.cpp, class CCbf`

9 Common Calendar Character Format

Common Calendar Character Format (CCF) is a character based machine-readable representation of date and time in a YMDhms form to impart familiar meaning to human users. It is complete and

[3] 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)*)

comprehensive to provide accurate and deterministic representation and symmetrical conversion to binary CBF.

See *Common Calendar Character Format*
See `CCT\CCTLib\CCF.h, CCcf.h, CCcf.cpp, class CCcf`

10 Common Calendar Timestamp API

Common Calendar as a whole provides specifications, data formats, and manipulation algorithms for deterministic UTC accurate local timekeeping. Common Calendar Timestamp API provides convenient user-level access to the Common Calendar capabilities.

Common Calendar specifies two timestamp formats, Common Calendar Binary Format (CBF) and Common Calendar Character Format (CCF), that are symmetrically convertible to one another.

The CBF TimeStamp API operations provide means of populating CBF instances for the supported purposes and their manipulating values and metadata. All timekeeping operations act on the binary CBF while the YMDhms form is reflected by a corresponding CCF.

A selection of key operations illustrate the nature of the Timestamp API:

- SetYMDhmsd - Populate a CBF as UTC accurate local date and time-of-day from YMDhmsd user input values and parameters with input value validation
- SetFrom1970Seconds - Populate a CBF as UTC accurate local date and time-of-day from seconds and decimal fraction values and user input parameters
- SetIntervalFromSecondsFrac - Populate a CBF as an interval (duration) from SecondsFrac_st user input values and parameters
- Set24HourTimepointFromSecondsFrac - Populate a CBF as a time-point from SecondsFrac_st user input values and parameters.
- AddUnits - Add units to CBF values.
- Difference - Compute the difference, the duration of the time interval, between two time points (A and B).
- GetCbf1970Seconds - Get total CBF seconds and fractions of seconds
- SetCcfFromCCbf - Construct a CCF string from CBF binary data and metadata
- ParseCcfSetCCbf - Parse CCF string and populate a CBF
- Calendar Operations - CCT Calendar provides month and year calendar representation

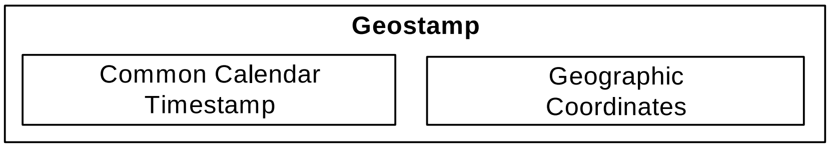
See *12 Common Calendar Timestamp API*
See CCT\CCTLib\CCct.h, CCct.cpp

11 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.

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.



12 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 many contributors to IANA Time Zone Database.

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

13 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://hpiers.obspm.fr/eop-pc/index.php

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

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/