Common Calendar GeoStamp

Geostamp format for Legal Time Traceability and 4D GIS Applications

Brooks Harris Version 2 2024-04-25

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).

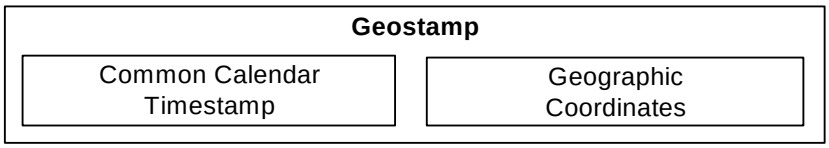
"UTC1970" is shorthand for 1970-01-01 00:00:10 (TAI) = 1970-01-01T00:00:00 (UTC).

1 Introduction

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 of 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.



Example in Common Calendar Character Format (CCF) with default tzdb time zone location:

```
D1972-06-29T23:59:59U-04Zamerica/new_yorkAedtV2021aL0Ss+01cMuCzt4200.850000g0.383333X
```

For an overview of Common Calendar in general please see
Common Calendar Introduction and Scope

2 Scope

This interoperability standard specifies a geographic coordinates structure to be used in conjunction with the Common Calendar Binary Format (CBF).

3 Normative References

Common Calendar Date and Time Terms and Definitions

Common Calendar TAI-UTC API

Common Calendar YMDhms API

Common Calendar Time Zone API

Common Calendar Binary Format

Common Calendar Character Format

National Marine Electronics Association
NMEA 0183 Interface Standard
GGA Global Positioning System Fix Data. Time,
Position and fix related data for a GPS receiver $GPGGA
https://www.nmea.org/nmea-0183.html

4 Geostamp

A Geostamp includes a Common Calendar timestamp together with geographic coordinates.

See Common Calendar Binary Format.
See Common Calendar Character Format.

CCT carries coordinates in the form specified by National Marine Electronics Association (NMEA), NMEA 0183 Interface Standard, GPGGA., GGA Global Positioning System Fix Data. Time, Position and fix related data for a GPS receiver.

The NMEA data is translated into the CBFLocation_st structure in the CCT binary CBF format. The CBF data is reflected in the CCF character format in the Location Element field.

The presence of the CBFLocation_st structure in the CCT timestamp is indicated by the `CBFLocalDate_st::m_TZDTimeZoneID_st::m_bCBFLocationExt` flag.

4.1.1 Geographic Coordinates - CBFLocation_st

The CBFLocation_st struct carries coordinates in a compact form consistent with data as specified by the NMEA 0183 Interface Standard, GPGGA, GGA Global Positioning System Fix Data. Time, Position and fix related data for a GPS receiver.

```
typedef struct CBFLocation_st // 14 bytes
{
unsigned long m_i21Lat_uMin:21;  // micro-minutes -100000 to 100000 range
                                 //[((2^21)/2)-1  =  1048575 MAX]
unsigned long m_i9Lat_Deg:9;     // degrees -90 to 90 range, negative is South
                                 // [((2^9) / 2) - 1 = 255 MAX]
unsigned long m_Pad1:2;

unsigned long m_i21Lng_uMin:21;  // micro-minutes -100000 to 100000 range
                                 // [((2^21)/2)-1  =  1048575 MAX]
unsigned long m_i9Lng_Deg:9;     // degrees -180 to 180 range, negative is
                                 // West [((2^9) / 2) - 1 = 255 MAX]
unsigned long m_Pad2:2;

unsigned long m_i19Alt_cm:19;    // +- 20000 meter x 100 centimeter range
                                 // [((2^19) / 2) - 1 = 262143 MAX]
unsigned long m_i7Lng_Min:7;     // minutes 0 to 60 range [((2^7) / 2) - 1 =
                                 // 63 MAX]
unsigned long m_Pad3:6;

unsigned char m_i7Lat_Min:7;     // minutes 0 to 60 range [((2^7) / 2) - 1 =
                                 // 63 MAX]
unsigned char m_bSourceIsExtern:1; // flag is external location, otherwise is
                                   // Tz Database location
unsigned char m_bIsValidLat:1;   // flag Latitude value valid
unsigned char m_bIsValidLng:1;   // flag Longitude value valid
unsigned char m_bIsValidAlt:1;   // flag Altitude value valid
unsigned char m_Pad4:5;
} CBFLocation_st;
```

See TzDatabaseAPI.h, CBFLocation_st
See CCct.h, CCct.cpp class CCct
int CCct::SetLocation(char* psLatitude, char* psLongitude, char* psAltitude);