Concise Binary Object Representation (CBOR) Tags for Time, Duration, and
                                Period
                    draft-bormann-cbor-time-tag-04

Abstract

   The Concise Binary Object Representation (CBOR, RFC 8949) is a data
   format whose design goals include the possibility of extremely small
   code size, fairly small message size, and extensibility without the
   need for version negotiation.

   In CBOR, one point of extensibility is the definition of CBOR tags.
   RFC 8949 defines two tags for time: CBOR tag 0 (RFC3339 time as a
   string) and tag 1 (Posix time as int or float). Since then,
   additional requirements have become known. The present document
   defines a CBOR tag for time that allows a more elaborate
   representation of time, as well as related CBOR tags for duration and
   time period. It is intended as the reference document for the IANA
   registration of the CBOR tags defined.

Note to Readers

   Version -00 of the present draft opened up the possibilities provided
   by extended representations of time in CBOR. Version -01
   consolidated this draft to non-speculative content, the normative
   parts of which are believed will stay unchanged during further
   development of the draft. This version is provided to aid the
   registration of the CBOR tag immediately needed. Versions -02 and
   -03 made use of the IANA allocations registered and made other
   editorial updates. Further versions will re-introduce some of the
   material from -00, but in a more concrete form.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering
Task Force (IETF). Note that other groups may also distribute
working documents as Internet-Drafts. The list of current Internet-
Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months
and may be updated, replaced, or obsoleted by other documents at any
time. It is inappropriate to use Internet-Drafts as reference
material or to cite them other than as "work in progress."

This Internet-Draft will expire on 26 August 2021.

Copyright Notice

Table of Contents

1. Introduction

   The Concise Binary Object Representation (CBOR, [RFC8949]) provides
   for the interchange of structured data without a requirement for a
   pre-agreed schema. RFC 8949 defines a basic set of data types, as
   well as a tagging mechanism that enables extending the set of data
   types supported via an IANA registry.

   (TBD: Expand on text from abstract here.)

1.1. Terminology

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in RFC 2119 [RFC2119].

   The term "byte" is used in its now customary sense as a synonym for
   "octet". Where bit arithmetic is explained, this document uses the
   notation familiar from the programming language C (including C++14's
   0bnnn binary literals), except that the operator "**" stands for
   exponentiation.

2. Objectives

   For the time tag, the present specification addresses the following
   objectives that go beyond the original tags 0 and 1:

   * Additional resolution for epoch-based time (as in tag 1). CBOR
      tag 1 only provides for integer and up to binary64 floating point
      representation of times, limiting resolution to approximately
      microseconds at the time of writing (and progressively becoming
      worse over time).

   * Indication of time scale. Tags 0 and 1 are for UTC; however, some
      interchanges are better performed on TAI. Other time scales may
      be registered once they become relevant (e.g., one of the proposed
      successors to UTC that might no longer use leap seconds, or a
      scale based on smeared leap seconds).

   Not currently addressed, but possibly covered by the definition of
   additional map keys for the map inside the tag:

* Direct representation of natural platform time formats. Some
  platforms use epoch-based time formats that require some
  computation to convert them into the representations allowed by
  tag 1; these computations can also lose precision and cause
  ambiguities. (TBD: The present specification does not take a
  position on whether tag 1 can be "fixed" to include, e.g., Decimal
  or BigFloat representations. It does define how to use these with
  the extended time format.)

* Additional indication of intents about the interpretation of the
  time given, in particular for future times. Intents might include
  information about time zones, daylight savings times, etc.

Additional tags are defined for durations and periods.

3. Time Format

An extended time is indicated by CBOR tag 1001, which tags a map data
item (CBOR major type 5). The map may contain integer (major types 0
and 1) or text string (major type 3) keys, with the value type
determined by each specific key. Implementations MUST ignore key/
value types they do not understand for negative integer and text
string values of the key. Not understanding key/value for unsigned
keys is an error.

The map must contain exactly one unsigned integer key, which
specifies the "base time", and may also contain one or more negative
integer or text-string keys, which may encode supplementary
information such as:

* a higher precision time offset to be added to the base time,

* a reference time scale and epoch different from the default UTC
  and 1970-01-01

* information about clock quality parameters, such as source,
  accuracy, and uncertainty

Future keys may add:

* intent information such as timezone and daylight savings time,
  and/or possibly positioning coordinates, to express information
  that would indicate a local time.

While this document does not define supplementary text keys, a number
of unsigned and negative-integer keys are defined below.

3.1. Key 1

   Key 1 indicates a value that is exactly like the data item that would
   be tagged by CBOR tag 1 (Posix time [TIME_T] as int or float). The
   time value indicated by the value under this key can be further
   modified by other keys.

3.2. Keys 4 and 5

   Keys 4 and 5 are like key 1, except that the data item is an array as
   defined for CBOR tag 4 or 5, respectively. This can be used to
   include a Decimal or Bigfloat epoch-based float [TIME_T] in an
   extended time.

3.3. Keys -3, -6, -9, -12, -15, -18

   The keys -3, -6, -9, -12, -15 and -18 indicate additional decimal
   fractions by giving an unsigned integer (major type 0) and scaling
   this with the scale factor 1e-3, 1e-6, 1e-9, 1e-12, 1e-15, and 1e-18,
   respectively (see Table 1). More than one of these keys MUST NOT be
   present in one extended time data item. These additional fractions
   are added to a base time in seconds [SI-SECOND] indicated by a Key 1,
   which then MUST also be present and MUST have an integer value.

Key	meaning	example usage
-3	milliseconds	Java time
-6	microseconds	(old) UNIX time
-9	nanoseconds	(new) UNIX time
-12	picoseconds	Haskell time
-15	femtoseconds	(future)
-18	attoseconds	(future)

                     Table 1: Key for decimally scaled
                                Fractions

3.4. Key -1: Time Scale

   Key -1 is used to indicate a time scale. The value 0 indicates UTC,
   with the POSIX epoch [TIME_T]; the value 1 indicates TAI, with the
   PTP (Precision Time Protocol) epoch [IEEE1588-2008].

If key -1 is not present, time scale value 0 is implied. Additional
values can be registered in the (TBD define name for time scale
registry); values MUST be integers or text strings.

(Note that there should be no time scales "GPS" or "NTP" -- instead,
the time should be converted to TAI or UTC using a single addition or
subtraction.)

$$t_{utc} = t_{ntp} - 2208988800$$

$$t_{tai} = t_{gps} + 315964819$$

           Figure 1: Converting Common Offset Time Scales

3.5. Clock Quality

A number of keys are defined to indicate the quality of clock that
was used to determine the point in time.

The first three are analogous to "clock-quality-grouping" in
[RFC8575], which is in turn based on the definitions in
[IEEE1588-2008]; two more are specific to this document.

```
ClockQuality-group = (
  ? ClockClass => uint .size 1 ; PTP/RFC8575
  ? ClockAccuracy => uint .size 1 ; PTP/RFC8575
  ? OffsetScaledLogVariance => uint .size 2 ; PTP/RFC8575
  ? Uncertainty => ~time/~duration
  ? Guarantee => ~time/~duration
)
ClockClass = -2
ClockAccuracy = -4
OffsetScaledLogVariance = -5
Uncertainty = -7
Guarantee = -8
```

3.5.1. ClockClass (Key -2)

Key -2 (ClockClass) can be used to indicate the clock class as per
Table 5 of [IEEE1588-2008]. It is defined as a one-byte integer as
that is the ranged defined there.

3.5.2. ClockAccuracy (Key -4)

   Key -4 (ClockAccuracy) can be used to indicate the clock accuracy as
   per Table 6 of [IEEE1588-2008]. It is defined as a one-byte integer
   as that is the ranged defined there. The range between 32 and 47 is
   a slightly distorted logarithmic scale from 25 ns to 1 s (see
   Figure 2); the number 254 is the value to be used if an unknown
   accuracy needs to be expressed.

$$\mathrm{enum}_{acc} \approx 48 + \left\lfloor 2 \cdot \log_{10}\{acc \text{ over } \mathrm{s}\} - \epsilon \right\rfloor$$

        Figure 2: Approximate conversion from accuracy to accuracy
                            enumeration value

3.5.3. OffsetScaledLogVariance (Key -5)

   Key -5 (OffsetScaledLogVariance) can be used to represent the
   variance exhibited by the clock when it has lost its synchronization
   with an external reference clock. The details for the computation of
   this characteristic are defined in Section 7.6.3 of [IEEE1588-2008].

3.5.4. Uncertainty (Key -7)

   Key -7 (Uncertainty) can be used to represent a known measurement
   uncertainty for the clock, as a numeric value in seconds or as a
   duration (Section 4).

   For this document, uncertainty is defined as in Section 2.2.3 of
   [GUM]: "parameter, associated with the result of a measurement, that
   characterizes the dispersion of the values that could reasonably be
   attributed to the measurand". More specifically, the value for this
   key represents the extended uncertainty for k = 2, in seconds.

3.5.5. Guarantee (Key -8)

   Key -8 (Guarantee) can be used to represent a stated guarantee for
   the accuracy of the point in time, as a numeric value in seconds or
   as a duration (Section 4) representing the maximum allowed deviation
   from the true value.

   While such a guarantee is unattainable in theory, existing standards
   such as [RFC3161] stipulate the representation of such guarantees,
   and therefore this format provides a way to represent them as well;
   the time value given is nominally guaranteed to not deviate from the
   actual time by more than the value of the guarantee, in seconds.

4. Duration Format

   A duration is the length of an interval of time. Durations in this
   format are given in SI seconds, possibly adjusted for conventional
   corrections of the time scale given (e.g., leap seconds).

   Except for using Tag 1002 instead of 1001, durations are structurally
   identical to time values. Semantically, they do not measure the time
   elapsed from a given epoch, but from the start to the end of (an
   otherwise unspecified) interval of time.

   In combination with an epoch identified in the context, a duration
   can also be used to express an absolute time.

   | (TBD: Clearly, ISO8601 durations are rather different; we do
   | not want to use these.)

5. Period Format

   A period is a specific interval of time, specified as either two
   times giving the start and the end of that interval, or as one of
   these two plus a duration.

   They are given as an array of unwrapped time and duration elements,
   tagged with Tag 1003:

```
Period = #6.1003([
  start: ~Time / null
  end: ~Time / null
  ? duration: ~Duration / null
])
```

   If the third array element is not given, the duration element is
   null. Exactly two out of the three elements must be non-null, this
   can be clumsily expressed in CDDL as:

```
Period = #6.1003([
  (start: ~Time,
   ((end: ~Time,
     ? duration: null) //
    (end: null,
     duration: ~Duration))) //
  (start: null,
   end: ~Time,
   duration: ~Duration)
])
```

> (Issue: should start/end be given the two-element treatment, or
> start/duration?)

6. CDDL typenames

   For the use with the CBOR Data Definition Language, CDDL [RFC8610],
   the type names defined in Figure 3 are recommended:

   etime = #6.1001({* (int/tstr) => any})
   duration = #6.1002({* (int/tstr) => any})
   period = #6.1003([~etime/null, ~etime/null, ~duration/null])

                  Figure 3: Recommended type names for CDDL

7. IANA Considerations

   In the registry [IANA.cbor-tags], IANA has allocated the tags in
   Table 2 from the FCFS space, with the present document as the
   specification reference.

Tag	Data Item	Semantics
1001	map	[RFCthis] extended time
1002	map	[RFCthis] duration
1003	array	[RFCthis] period

                       Table 2: Values for Tags

   IANA is requested to change the "Data Item" column for Tag 1003 from
   "map" to "array".

> (TBD: Add registry for time scales. Add registry for map keys
> and allocation policies for additional keys.)

8. Security Considerations

   The security considerations of RFC 8949 apply; the tags introduced
   here are not expected to raise security considerations beyond those.

   Time, of course, has significant security considerations; these
   include the exploitation of ambiguities where time is security
   relevant (e.g., for freshness or in a validity span) or the
   disclosure of characteristics of the emitting system (e.g., time
   zone, or clock resolution and wall clock offset).

9. References

9.1. Normative References

   [GUM]   Joint Committee for Guides in Metrology, "Evaluation of
                  measurement data Guide to the expression of uncertainty
                  in measurement", JCGM 100:2008, September 2008,
                  <https://www.bipm.org/en/publications/guides/gum.html>.

   [IANA.cbor-tags]
                  IANA, "Concise Binary Object Representation (CBOR) Tags",
                  <http://www.iana.org/assignments/cbor-tags>.

   [IEEE1588-2008]
                  IEEE, "1588-2008 - IEEE Standard for a Precision Clock
                  Synchronization Protocol for Networked Measurement and
                  Control Systems", July 2008,
                  <http://standards.ieee.org/findstds/
                  standard/1588-2008.html>.

   [RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
                  Requirement Levels", BCP 14, RFC 2119,
                  DOI 10.17487/RFC2119, March 1997,
                  <https://www.rfc-editor.org/info/rfc2119>.

   [RFC8610]   Birkholz, H., Vigano, C., and C. Bormann, "Concise Data
                  Definition Language (CDDL): A Notational Convention to
                  Express Concise Binary Object Representation (CBOR) and
                  JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610,
                  June 2019, <https://www.rfc-editor.org/info/rfc8610>.

   [RFC8949]   Bormann, C. and P. Hoffman, "Concise Binary Object
                  Representation (CBOR)", STD 94, RFC 8949,
                  DOI 10.17487/RFC8949, December 2020,
                  <https://www.rfc-editor.org/info/rfc8949>.

   [SI-SECOND]
                  International Organization for Standardization (ISO),
                  "Quantities and units Part 3: Space and time",
                  ISO 80000-3, 1 March 2006.

   [TIME_T]   The Open Group Base Specifications, "Vol. 1: Base
                  Definitions, Issue 7", Section 4.15 'Seconds Since the
                  Epoch', IEEE Std 1003.1-2008, 2016 Edition, 2016,
                  <http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/
                  V1_chap04.html#tag_04_16>.

9.2. Informative References

   [RFC3161] Adams, C., Cain, P., Pinkas, D., and R. Zuccherato,
              "Internet X.509 Public Key Infrastructure Time-Stamp
              Protocol (TSP)", RFC 3161, DOI 10.17487/RFC3161, August
              2001, <https://www.rfc-editor.org/info/rfc3161>.

   [RFC8575] Jiang, Y., Ed., Liu, X., Xu, J., and R. Cummings, Ed.,
              "YANG Data Model for the Precision Time Protocol (PTP)",
              RFC 8575, DOI 10.17487/RFC8575, May 2019,
              <https://www.rfc-editor.org/info/rfc8575>.

Acknowledgements

Authors' Addresses

   Carsten Bormann
   Universität Bremen TZI
   Postfach 330440
   D-28359 Bremen
   Germany

   Phone: +49-421-218-63921
   Email: cabo@tzi.org

   Ben Gamari
   Well-Typed
   117 Middle Rd.
   Portsmouth, NH 03801
   United States

   Email: ben@well-typed.com

   Henk Birkholz
   Fraunhofer Institute for Secure Information Technology
   Rheinstrasse 75
   64295 Darmstadt
   Germany

   Email: henk.birkholz@sit.fraunhofer.de