Stream: Internet Engineering Task Force (IETF)
RFC: 9581
Category: Standards Track
Published: August 2024
ISSN: 2070-1721
Authors: C. Bormann, *Universität Bremen TZI*; B. Gamari, *Well-Typed*; H. Birkholz, *Fraunhofer SIT*

RFC 9581
Concise Binary Object Representation (CBOR) Tags for Time, Duration, and Period

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 (RFC 3339 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. This document is intended as the reference document for the IANA registration of the CBOR tags defined.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9581.

Copyright Notice

Table of Contents

1. Introduction 3
   1.1. Terminology 3
2. Objectives 4
3. Time Format 4
   3.1. Key 1 6
   3.2. Keys 4 and 5 6
   3.3. Keys -3, -6, -9, -12, -15, and -18 6
   3.4. Keys -1, -13, and 13: Timescale 7
   3.5. Clock Quality 8
      3.5.1. ClockClass (Key -2) 8
      3.5.2. ClockAccuracy (Key -4) 8
      3.5.3. OffsetScaledLogVariance (Key -5) 9
      3.5.4. Uncertainty (Key -7) 9
      3.5.5. Guarantee (Key -8) 9
   3.6. Keys -10, 10: Time Zone Hint 10
   3.7. Keys -11, 11: IXDTF Suffix Information 10
4. Duration Format 11
5. Period Format 12
6. CDDL Type Names 13
7. IANA Considerations 13
   7.1. CBOR Tags 13
   7.2. Timescales Registry 13
   7.3. Time Tag Map Keys Registry 14
8. Security Considerations 15

9. References 16
9.1. Normative References 16
9.2. Informative References 17
Appendix A. Collected CDDL 18
Acknowledgements 21
Authors' Addresses 21

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 for CBOR tags (see [IANA.cbor-tags] and Section 9.2 of [RFC8949]).

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 durations and time periods. This document is intended as the reference document for the IANA registration of the CBOR tags defined.

1.1. Terminology

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The term "byte" is used in its now customary sense as a synonym for "octet".

Superscript notation denotes exponentiation. For example, 2 to the power of 64 is notated: 2^{64}. In the plain-text rendition of this specification, superscript notation is not available and exponentiation is therefore rendered by the surrogate notation seen here in the plain-text rendition.

CBOR diagnostic notation is defined in Section 8 of [RFC8949] and Appendix G of [RFC8610]. A machine-processable model of the data structures defined in this specification is provided throughout the text using the Concise Data Definition Language (CDDL) [RFC8610]; Appendix A provides the collected model information.

Several time-related terms, such as UTC and International Atomic Time (TAI), are discussed in [RFC9557], which may be a useful companion document beyond its direct use in Sections 3.6 and 3.7.

2. Objectives

For the time tag, the present specification addresses the following objectives that go beyond the original tags 0 and 1 (defined in Sections 3.4.1 and 3.4.2 of [RFC8949]):

- Additional resolution for epoch-based time (as in tag 1). CBOR tag 1 only provides for representation of time as an integer and as up to a binary64 floating-point value [IEEE754], which limits the resolution to approximately microseconds at the time of writing (progressively becoming worse over time).
- Indication of timescale. Tags 0 and 1 are defined for UTC; however, some interchanges are better performed on TAI. Other timescales 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).

By incorporating a way to transport [RFC9557] suffix information (see Sections 3.6 and 3.7), additional indications of intents about the interpretation of the time given can be provided; in particular, for instances of time that, at the time they are being described, are in the future. Intents might include information about time zones, daylight saving times, preferred calendar representations, etc.

Semantics not covered by this document can be added by registering additional map keys for the map that is the content of the tag (see `etime-detailed` in Figure 1), the specification for which is referenced by the registry entry (see Section 3).

For example, map keys could be registered for direct representations 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. (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 representations with the extended time format.)

Additional tags are defined for durations and periods.

3. Time Format

An extended time is indicated by CBOR tag 1001, the content of which is 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. For negative integer keys and text string values of the key, implementations **MUST** ignore key/value pairs they do not understand; these keys are "elective", as the extended time as a whole is still usable without the information they carry if an implementation elects not to implement them. Conversely, implementations **MUST** signal an error when encountering key/value pairs that use unsigned integer keys they do not understand or implement (these are either "base time" or "critical", see below).

The map **MUST** contain exactly one unsigned integer key that specifies the "base time" and **MAY** also contain one or more negative integer or text-string keys, which may encode supplementary information.

Supplementary information **MAY** also be provided by additional unsigned integer keys that are explicitly defined to provide supplementary information (we say these keys are defined to be "critical"); as these are required to be understood, there can be no confusion with base time keys.

Negative integer and text string keys always supply supplementary information (they are "elective", and this will not be explicitly stated below).

Supplementary information may include:

- a higher precision time offset to be added to the base time,
- a reference timescale and epoch different from the default UTC and 1970-01-01, and
- information about clock quality parameters, such as source, accuracy, and uncertainty.

Additional keys can be defined by registering them in the "Time Tag Map Keys" registry (Section 7.3). Registered keys may, for instance, add intent information such as time zone, daylight saving time, and/or possibly positioning coordinates to express information that would indicate a local time.

This document does not define supplementary text keys. A number of both unsigned and negative-integer keys are defined in the following subsections.

Figure 1 provides a formal definition of tag 1001 in CDDL.

```
Etime = #6.1001(etime-detailed)

etime-framework = {
  uint => any ; at least one base time
  * (nint/text) => any ; elective supplementary information
  * uint => any ; critical supplementary information
}

etime-detailed = ({
  $$ETIME-BASETIME
  ClockQuality-group
  * $$ETIME-ELECTIVE
  * $$ETIME-CRITICAL
  * ((nint/text) .feature "etime-elective-extension") => any
  * (uint .feature "etime-critical-extension") => any
}) .within etime-framework
```

Figure 1: CDDL Definition of Tag 1001

3.1. Key 1

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

```
$$ETIME-BASETIME //= (1: ~time)
```

3.2. Keys 4 and 5

Keys 4 and 5 indicate a base time value and 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, e.g., to achieve higher resolution or to avoid rounding errors.

```
$$ETIME-BASETIME //= (4: ~decfrac)
$$ETIME-BASETIME //= (5: ~bigfloat)
```

3.3. Keys -3, -6, -9, -12, -15, and -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). Each extended time data item **MUST NOT** contain more than one of these keys. These additional fractions are added to a base time in seconds [SI-SECOND] indicated by 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: Keys for Decimally Scaled Fractions

```
$$ETIME-ELECTIVE //= (-3: uint)
$$ETIME-ELECTIVE //= (-6: uint)
$$ETIME-ELECTIVE //= (-9: uint)
$$ETIME-ELECTIVE //= (-12: uint)
$$ETIME-ELECTIVE //= (-15: uint)
$$ETIME-ELECTIVE //= (-18: uint)
```

Note that these keys have been provided to facilitate representing pairs of the form second/decimal fraction of a second, as found for instance in C `timespec` (Section 7.27.1 of [C]). When ingesting a timestamp with one of these keys into a type provided by the target platform, care has to be taken to meet its invariants. For example, for C `timespec`, the fractional part `tv_nsec` needs to be between 0 inclusive and 10^9 exclusive, which can be achieved by also adjusting the base time appropriately.

3.4. Keys -1, -13, and 13: Timescale

Keys -1, -13, and 13 are used to indicate a timescale, where key 13 is critical. Keys -1 and -13 have identical semantics (both are assigned because key -1 was chosen first and then, when key 13 was added, it appeared desirable to have a negative equivalent). Each extended time data item **MUST NOT** contain more than one of these keys.

The value 0 indicates UTC, with the POSIX epoch [TIME_T]; the value 1 indicates TAI, with the Precision Time Protocol (PTP) epoch (1 January 1970 00:00:00 TAI, see [IEEE1588-2019] or [IEEE1588-2008]).

```
$$ETIME-ELECTIVE //= (-1 => $ETIME-TIMESCALE)
$$ETIME-ELECTIVE //= (-13 => $ETIME-TIMESCALE)
$$ETIME-CRITICAL //= (13 => $ETIME-TIMESCALE)

$ETIME-TIMESCALE /= &(etime-utc: 0)
$ETIME-TIMESCALE /= &(etime-tai: 1)
```

If none of the keys are present, the default timescale value 0 is implied.

Timescale values **MUST** be unsigned integers or text strings; text strings are provided for experimentation and **MUST NOT** be used between parties that are not both part of the experiment. Additional unsigned integer values can be registered in the "Timescales" registry (Section 7.2). (Note that there should be no timescales "GPS" or "NTP" [RFC5905] -- 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 2: Converting Common Offset Timescales

> Editor's note: This initial set of timescales was deliberately chosen to be frugal, as the specification of the tag provides an extension point where additional timescales can be registered at any time. Registrations are clearly needed for earth-referenced timescales (such as UT1 and TT), as well as possibly for specific realizations of abstract timescales (such as TAI(USNO), the specific realization obtained at the United States Naval Observatory, which is more accurate as a constant offset basis for GPS times). While the registration process itself is trivial, these registrations need to be made based on a solid specification of their actual definition.

3.5. Clock Quality

A number of keys are defined to indicate the quality of the 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]; the last two are specific to this document.

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

3.5.1. ClockClass (Key -2)

Key -2 (ClockClass) can be used to indicate the clock class as per [RFC8575] (which is based on Table 5 in Section 7.6.2.4 of [IEEE1588-2008]; Table 4 in Section 7.6.2.5 of [IEEE1588-2019] has updated language). It is defined as a one-byte unsigned integer as that is the range defined in IEEE 1588.

3.5.2. ClockAccuracy (Key -4)

Key -4 (ClockAccuracy) can be used to indicate the clock accuracy as per [RFC8575] (which is based on Table 6 in Section 7.6.2.5 of [IEEE1588-2008]; additional values have been defined in Table 5 in Section 7.6.2.6 of [IEEE1588-2019]). It is defined as a one-byte unsigned integer as that is the range defined there. The range between 23 and 47 is a slightly distorted logarithmic scale from 1 ps to 1 s in [IEEE1588-2019] (in [IEEE1588-2008], the range was a subset of that, 32 to 47 for 25 ns to 1 s) -- see Figure 3; the number 254 is the value to be used if an unknown accuracy needs to be expressed.

$$enum_{acc} \approx 48 + \lfloor 2 \cdot log_{10} \frac{acc}{\mathrm{s}} - \varepsilon \rfloor$$

Figure 3: 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 [IEEE1588-2019] and the same section in [IEEE1588-2008].

3.5.4. Uncertainty (Key -7)

Key -7 (Uncertainty) can be used to represent a known uncertainty of measurement 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 expanded uncertainty for k = 2 (Section 6.2.1 of [GUM]) in seconds.

Note that the additional information that can be meaningfully provided with the duration that represents an uncertainty is limited, e.g., it is not customary to provide an uncertainty for a duration representing an uncertainty. Implementations are free to reduce the information contained in an uncertainty (which is already elective) to the information they can process.

For example, a timestamp that is given to a resolution of 10^{-6} seconds (microseconds) but only has an uncertainty of 10^{-3} seconds (milliseconds) could be expressed by one of the extended time tags in Figure 4 (note the slight rounding error in the third case, which is probably inconsequential for an uncertainty value):

```
1001({1: 1697724754, -6: 873294, -7: {1: 0, -6: 1000}}),
1001({1: 1697724754, -6: 873294, -7: {1: 0, -3: 1}}),
1001({1: 1697724754, -6: 873294, -7: {1: 0.001}})
```

Figure 4: Examples Using Uncertainty

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.

Note that the additional information that can be meaningfully provided with the duration that represents a guarantee is limited, e.g., it is not meaningful to provide a guarantee of accuracy for the duration representing a guarantee of accuracy. Implementations are free to reduce a guarantee (which is already elective) to the information they can process.

3.6. Keys -10, 10: Time Zone Hint

Keys -10 and 10 supply supplementary information, where key 10 is critical.

They can be used to provide a hint about the time zone that would best fit for displaying the time given to humans, using a text string in the format defined for `time-zone-name` or `time-numoffset` in [RFC9557]. Key -10 is equivalent to providing this information as an elective hint, while key 10 provides this information as critical (i.e., it **MUST** be used when interpreting the entry with this key).

Keys -10 and 10 **MUST NOT** both be present.

```
$$ETIME-ELECTIVE //= (-10: time-zone-info)
$$ETIME-CRITICAL //= (10: time-zone-info)

time-zone-info = tstr .abnf
                 ("time-zone-name / time-numoffset" .det IXDTFtz)
IXDTFtz = '
   time-hour       = 2DIGIT  ; 00-23
   time-minute     = 2DIGIT  ; 00-59
   time-numoffset  = ("+" / "-") time-hour ":" time-minute

   time-zone-initial = ALPHA / "." / "_"
   time-zone-char    = time-zone-initial / DIGIT / "-" / "+"
   time-zone-part    = time-zone-initial *time-zone-char
                       ; but not "." or ".."
   time-zone-name    = time-zone-part *("/" time-zone-part)
   ALPHA             =  %x41-5A / %x61-7A   ; A-Z / a-z
   DIGIT             =  %x30-39 ; 0-9
' ; extracted from [RFC9557] and [RFC3339]
```

3.7. Keys -11, 11: IXDTF Suffix Information

Keys -11 and 11 supply supplementary information, where key 11 is critical.

Similar to keys -10 and 10, keys -11 (elective) and 11 (critical) can be used to provide additional information in the style of Internet Extended Date/Time Format (IXDTF) suffixes, such as the calendar that would best fit for displaying the time given to humans. The key's value is a map that has IXDTF `suffix-key` names as keys and corresponding suffix values as values, specifically:

```
$$ETIME-ELECTIVE //= (-11: suffix-info-map)
$$ETIME-CRITICAL //= (11: suffix-info-map)

suffix-info-map = { * suffix-key => suffix-values }
suffix-key = tstr .abnf ("suffix-key" .det IXDTF)
suffix-values = one-or-more<suffix-value>
one-or-more<T> = T / [ 2* T ]
suffix-value = tstr .abnf ("suffix-value" .det IXDTF)

IXDTF = '
   key-initial       = lcalpha / "_"
   key-char          = key-initial / DIGIT / "-"
   suffix-key        = key-initial *key-char

   suffix-value      = 1*alphanum
   alphanum          = ALPHA / DIGIT
   lcalpha           =  %x61-7A
   ALPHA             =  %x41-5A / %x61-7A   ; A-Z / a-z
   DIGIT             =  %x30-39 ; 0-9
' ; extracted from [RFC9557]
```

When keys -11 and 11 are both present, the two maps **MUST NOT** have entries with the same map keys.

Figure 4 of [RFC9557] gives an example for an extended date-time with both time zone and suffix information:

```
1996-12-19T16:39:57-08:00[America/Los_Angeles][u-ca=hebrew]
```

A time tag that is approximating this example, in CBOR diagnostic notation, would be:

```
/ 1996-12-19T16:39:57-08:00[America//Los_Angeles][u-ca=hebrew] /
1001({ 1: 851042397,
     -10: "America/Los_Angeles",
     -11: { "u-ca": "hebrew" }
})
```

Note that both -10 and -11 are using negative keys and therefore provide elective information, as in the IXDTF form given in the comment. Also note that, in this example, the time numeric offset (`-08:00`) is lost in translating from the [RFC3339] information in the IXDTF into a POSIX time that can be included under key 1 in a time tag.

4. Duration Format

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

Except for using tag 1002 instead of 1001, durations are structurally identical to time values.

```
Duration = #6.1002(etime-detailed)
```

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.

Without such context, durations are subject to some uncertainties underlying the timescale used. For example, for durations intended as a determinant of future time periods, there is some uncertainty of what irregularities (such as leap seconds and timescale corrections) will be exhibited by the timescale in that period. For durations as measurements of past periods, abstracting the period to a duration loses some detail about timescale irregularities. For many applications, these uncertainties are acceptable and thus the use of durations is appropriate.

> Note that the durations defined in [ISO8601:1988] and [ISO8601-1:2019] are rather different from the ones defined in the present specification; there is no intention to support ISO 8601 durations here.

5. Period Format

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

This is represented as an array of unwrapped time and duration elements, tagged with tag 1003, one of:

- a start and end time, in which case the tag content is an array of two unwrapped extended time elements, or
- a start time with duration or an end time with duration. The tag content is an array of 3 elements: the first two as above but either the start or end time **MUST** be set to null and the third one then is an unwrapped duration.

A simple CDDL definition that does not capture all the constraints is:

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

Exactly two out of the three elements must be present and non-null; this can be somewhat more verbosely expressed in CDDL as:

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

6. CDDL Type Names

When detailed validation is not needed, the type names defined in Figure 5 are recommended:

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

Figure 5: Recommended Type Names for CDDL

7. IANA Considerations

7.1. CBOR Tags

In the "CBOR Tags" registry [IANA.cbor-tags], IANA has allocated the tags in Table 2.

Tag	Data Item	Semantics	Reference
1001	map	extended time	[RFC9581, Section 3]
1002	map	duration	[RFC9581, Section 4]
1003	array	period	[RFC9581, Section 5]

Table 2: Values for Tags

IANA has updated the "Data Item" column for tag 1003 from "map" to "array".

7.2. Timescales Registry

Per this specification, IANA has created a new "Timescales" registry within the "Concise Binary Object Representation (CBOR) Tags" registry group [IANA.cbor-tags]. The registration procedure requires both "Expert Review" and "RFC Required" (Sections 4.5 and 4.7 of RFC 8126 [BCP26], respectively).

Each entry needs to provide a timescale name (a sequence of uppercase ASCII characters and digits, where a digit may not occur at the start: `[A-Z][A-Z0-9]*`), a value (CBOR unsigned integer, uint, 0..18446744073709551615), a brief description of the semantics, and a specification reference (RFC). The initial contents are shown in Table 3.

Timescale	Value	Semantics	Reference
UTC	0	UTC with POSIX Epoch	[RFC9581]
TAI	1	TAI with PTP Epoch	[RFC9581]

Table 3: Initial Content of Timescale Registry

7.3. Time Tag Map Keys Registry

Per this specification, IANA has created a new "Time Tag Map Keys" registry within the "Concise Binary Object Representation (CBOR) Tags" registry group [IANA.cbor-tags]. The registration procedure is "Specification Required" (Section 4.6 of RFC 8126 [BCP26]).

The designated expert is requested to assign the key values with the shortest encodings (1+0 and 1+1 encoding) to registrations that are likely to enjoy wide use and can benefit from short encodings.

Each entry needs to provide a map key value (CBOR integer, int, -18446744073709551616..18446744073709551615), a brief description of the semantics, and a specification reference. Note that negative integers indicate an elective key, while unsigned integers indicate a key that either provides a base time or is critical. The designated expert is requested to discuss with the registrant whether or not it is desirable to register a pair of an elective and a critical key for the same information, where the elective key value is the negative of the critical key (similar to how for example -11 and 11 have been assigned in Table 4). For the unsigned integers as keys, the choice of base time or critical needs to be indicated in the brief semantics description. (Elective map keys may be explicitly marked as such in the description, e.g., to distinguish them from critical keys.)

The initial contents are shown in Table 4.

Value	Semantics	Reference
-18	attoseconds	[RFC9581]
-15	femtoseconds	[RFC9581]
-13	timescale (elective)	[RFC9581]
-12	picoseconds	[RFC9581]
-11	IXDTF Suffix Information (elective)	[RFC9581], [RFC9557]

Value	Semantics	Reference
-10	IXDTF Time Zone Hint (elective)	[RFC9581], [RFC9557]
-9	nanoseconds	[RFC9581]
-8	Guarantee	[RFC9581]
-7	Uncertainty	[RFC9581]
-6	microseconds	[RFC9581]
-5	Offset-Scaled Log Variance	[RFC9581]
-4	Clock Accuracy	[RFC9581]
-3	milliseconds	[RFC9581]
-2	Clock Class	[RFC9581]
-1	timescale (elective) legacy	[RFC9581]
1	base time value as in CBOR tag 1	[RFC8949], [RFC9581]
4	base time value as in CBOR tag 4	[RFC8949], [RFC9581]
5	base time value as in CBOR tag 5	[RFC8949], [RFC9581]
10	IXDTF Time Zone Hint (critical)	[RFC9557], [RFC9581]
11	IXDTF Suffix Information (critical)	[RFC9557], [RFC9581]
13	timescale (critical)	[RFC9581]

Table 4: Initial Content of Time Tag Map Keys Registry

8. Security Considerations

The security considerations of [RFC8949] 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).

A more detailed discussion of security considerations emanating from using a representation of time that allows the inclusion of complex and possibly inconsistent information is available in "Security Considerations" (Section 7 of [RFC9557]).

9. References

9.1. Normative References

[BCP26] Best Current Practice 26, <https://www.rfc-editor.org/info/bcp26>.
At the time of writing, this BCP comprises the following:

Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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", <https://www.iana.org/assignments/cbor-tags>.

[IEEE1588-2008] IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE 1588-2008, July 2008, <https://standards.ieee.org/ieee/1588/4355/>. Often called PTP v2, as it replaced the earlier 2002 version of this standard by a non-backwards compatible protocol.

[IEEE1588-2019] IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE 1588-2019, June 2020, <https://standards.ieee.org/ieee/1588/6825/>. Often called PTP v2.1, as it has been designed so it can be used in a way that is fully backwards compatible to IEEE1588-2008.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[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>.

[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>.

[RFC9557] Sharma, U. and C. Bormann, "Date and Time on the Internet: Timestamps with Additional Information", RFC 9557, DOI 10.17487/RFC9557, April 2024, <https://www.rfc-editor.org/info/rfc9557>.

[SI-SECOND] ISO, "Quantities and units -- Part 3: Space and time", ISO 80000-3:2019, October 2019, <https://www.iso.org/standard/64974.html>.

[TIME_T] IEEE, "The Open Group Base Specifications Issue 7", Section 4.16 Seconds Since the Epoch, IEEE Std 1003.1-2017, 2018, <http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap04.html#tag_04_16>.

9.2. Informative References

[C] ISO, "Information technology -- Programming languages -- C", Fourth Edition, ISO/IEC 9899:2018, June 2018, <https://www.iso.org/standard/74528.html>. Contents available via <https://www.open-std.org/jtc1/sc22/wg14/www/docs/n2310.pdf>

[IEEE754] IEEE, "IEEE Standard for Floating-Point Arithmetic", IEEE 754-2019, DOI 10.1109/IEEESTD.2019.8766229, July 2019, <https://ieeexplore.ieee.org/document/8766229>.

[ISO8601-1:2019] ISO, "Date and time -- Representations for information interchange -- Part 1: Basic rules", ISO 8601-1:2019, February 2019, <https://www.iso.org/standard/70907.html>.

[ISO8601:1988] ISO, "Data elements and interchange formats -- Information interchange -- Representation of dates and times", ISO 8601:1988, June 1988, <https://www.iso.org/standard/15903.html>. Also available from <https://nvlpubs.nist.gov/nistpubs/Legacy/FIPS/fipspub4-1-1991.pdf>.

[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>.

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <https://www.rfc-editor.org/info/rfc3339>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

Appendix A. Collected CDDL

This appendix collects the CDDL rules spread over the document into one convenient place.

```
Etime = #6.1001(etime-detailed)

etime-framework = {
  uint => any ; at least one base time
  * (nint/text) => any ; elective supplementary information
  * uint => any ; critical supplementary information
}

etime-detailed = ({
  $$ETIME-BASETIME
  ClockQuality-group
  * $$ETIME-ELECTIVE
  * $$ETIME-CRITICAL
  * ((nint/text) .feature "etime-elective-extension") => any
  * (uint .feature "etime-critical-extension") => any
}) .within etime-framework


$$ETIME-BASETIME //= (1: ~time)


$$ETIME-BASETIME //= (4: ~decfrac)
$$ETIME-BASETIME //= (5: ~bigfloat)


$$ETIME-ELECTIVE //= (-3: uint)
$$ETIME-ELECTIVE //= (-6: uint)
$$ETIME-ELECTIVE //= (-9: uint)
$$ETIME-ELECTIVE //= (-12: uint)
$$ETIME-ELECTIVE //= (-15: uint)
$$ETIME-ELECTIVE //= (-18: uint)


$$ETIME-ELECTIVE //= (-1 => $ETIME-TIMESCALE)
$$ETIME-ELECTIVE //= (-13 => $ETIME-TIMESCALE)
$$ETIME-CRITICAL //= (13 => $ETIME-TIMESCALE)

$ETIME-TIMESCALE /= &(etime-utc: 0)
$ETIME-TIMESCALE /= &(etime-tai: 1)


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


$$ETIME-ELECTIVE //= (-10: time-zone-info)
$$ETIME-CRITICAL //= (10: time-zone-info)

time-zone-info = tstr .abnf
                 ("time-zone-name / time-numoffset" .det IXDTFtz)
IXDTFtz = '
   time-hour       = 2DIGIT  ; 00-23
```

```
   time-minute     = 2DIGIT  ; 00-59
   time-numoffset  = ("+" / "-") time-hour ":" time-minute

   time-zone-initial = ALPHA / "." / "_"
   time-zone-char    = time-zone-initial / DIGIT / "-" / "+"
   time-zone-part    = time-zone-initial *time-zone-char
                     ; but not "." or ".."
   time-zone-name    = time-zone-part *("/" time-zone-part)
   ALPHA             =  %x41-5A / %x61-7A   ; A-Z / a-z
   DIGIT             =  %x30-39 ; 0-9
' ; extracted from [RFC9557] and [RFC3339]


$$ETIME-ELECTIVE //= (-11: suffix-info-map)
$$ETIME-CRITICAL //= (11: suffix-info-map)

suffix-info-map = { * suffix-key => suffix-values }
suffix-key = tstr .abnf ("suffix-key" .det IXDTF)
suffix-values = one-or-more<suffix-value>
one-or-more<T> = T / [ 2* T ]
suffix-value = tstr .abnf ("suffix-value" .det IXDTF)

IXDTF = '
   key-initial       = lcalpha / "_"
   key-char          = key-initial / DIGIT / "-"
   suffix-key        = key-initial *key-char

   suffix-value      = 1*alphanum
   alphanum          = ALPHA / DIGIT
   lcalpha           =  %x61-7A
   ALPHA             =  %x41-5A / %x61-7A   ; A-Z / a-z
   DIGIT             =  %x30-39 ; 0-9
' ; extracted from [RFC9557]


Duration = #6.1002(etime-detailed)


simple-Period = #6.1003([
  start: ~Etime / null
  end: ~Etime / null
  ? duration: ~Duration
])


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


etime = #6.1001({* (int/tstr) => any})
```

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

Figure 6: Collected CDDL Rules from This Specification

Acknowledgements

The authors would like to acknowledge the many comments from members of the CBOR WG, Francesca Palombini for her AD review, Thomas Fossati and Qin Wu for their directorate reviews, and Rohan Mahy for one more review late in the process.

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 of America
Email: ben@well-typed.com

Henk Birkholz
Fraunhofer Institute for Secure Information Technology
Rheinstrasse 75
64295 Darmstadt
Germany
Email: henk.birkholz@ietf.contact