Workgroup:	Calendaring Extensions Working Group
Internet-Draft:	draft-ryzokuken-datetime-extended-02
Published:	16 June 2021
Intended Status:	Standards Track
Expires:	18 December 2021
Author:	U. Sharma
	Igalia, S.L.

Date and Time on the Internet: Timestamps with additional information

Abstract

This document defines an extension to the timestamp format defined in [RFC3339] for representing additional information including a time zone.

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 18 December 2021.

Copyright Notice

Table of Contents

1. Introduction

Dates and times are used in a very diverse set of internet applications, all the way from server-side logging to calendaring and scheduling.

Each distinct instant in time can be represented in a descriptive text format using a timestamp, and [ISO8601] standardizes a widely-adopted timestamp format, which forms the basis of [RFC3339]. However, this format doesn't allow timestamps to contain any additional relevant information, which means that any contextual information related to a given timestamp needs to be either handled separately or attached to it in a non-standard manner.

This is already a pressing issue for applications that handle each instant with an associated time zone name, to take into account things like DST transitions. Most of these applications attach the timezone to the timestamp in a non-standard format, atleast one of which is fairly well-adopted. Furthermore, applications might want to attach even more information to the timestamp, including but not limited to the calendar system it must be represented in.

This document seeks to standardize an extension syntax for timestamps as specified in [RFC3339] that has the following properties:

- The extension suffix should be completely optional, making existing [RFC3339] timestamps compatible with this format.
- The format should be compatible to the pre-existing popular syntax for attaching time zone names to timestamps.
- The format should allow a generalized way to attach any additional information to the timestamp.

2. Definitions

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 [RFC2119].

UTC Coordinated Universal Time as maintained since 1988 by the Bureau International des Poids et Mesures (BIPM) in conjunction with leap seconds as announced by the International Earth Rotation and Reference Frames Service [IERS]. From 1972 through 1987 UTC was maintained entirely by Bureau International de l'Heure (BIH). Before 1972 UTC was not generally recognized and civil time was determined by individual jurisdictions using different techniques for attempting to follow Universal Time based on measuring the rotation of the earth.

ABNF Augmented Backus-Naur Form, a format used to represent permissible strings in a protocol or language, as defined in [RFC2234].

Internet Date/Time Format The date/time format defined in section 4 of this document.

Timestamp This term is used in this document to refer to an unambiguous representation of some instant in time.

Z A suffix which, when applied to a time, denotes a UTC offset of 00:00; often spoken "Zulu" from the ICAO phonetic alphabet representation of the letter "Z".

Time Zone A time zone that is a included in the Time Zone Database (often called `tz` or `zoneinfo`) maintained by IANA.

For more information about time scales, see Appendix E of [RFC1305], Section 3 of [ISO8601], and the appropriate ITU documents [ITU-R-TF].

3. Local Time

3.1. Coordinated Universal Time (UTC)

Because the daylight saving rules for local time zones are so convoluted and can change based on local law at unpredictable times, true interoperability is best achieved by using Coordinated Universal Time (UTC). This specification by itself does not cater to local time zone rules. However, certain implementations may be expected to. For these situations, a timestamp may additionally include a local time zone that the implementations can take into account.

3.2. Local Offsets

The offset between local time and UTC is often useful information. For example, in electronic mail (RFC2822, [RFC2822]) the local offset provides a useful heuristic to determine the probability of a prompt response. Attempts to label local offsets with alphabetic strings have resulted in poor interoperability in the past [RFC1123]. As a result, RFC2822 [RFC2822] has made numeric offsets mandatory.

Numeric offsets are calculated as "local time minus UTC". So the equivalent time in UTC can be determined by subtracting the offset from the local time. For example, `18:50:00-04:00` is the same time as `22:50:00Z`. (This example shows negative offsets handled by adding the absolute value of the offset.)

Numeric offsets may differ from UTC by any number of seconds, or even a fraction of seconds. This can be easily represented by including an optional seconds value in the offset, which may further optionally include a fraction of seconds behind a decimal point, for example `+12:34:56.789`. This is especially useful in the case of certain historical time zones.

3.3. Unknown Local Offset Convention

If the time in UTC is known, but the offset to local time is unknown, this can be represented with an offset of "-00:00". This differs semantically from an offset of "Z" or "+00:00", which imply that UTC is the preferred reference point for the specified time. RFC2822 [RFC2822] describes a similar convention for email.

3.4. Unqualified Local Time

A number of devices currently connected to the Internet run their internal clocks in local time and are unaware of UTC. While the Internet does have a tradition of accepting reality when creating specifications, this should not be done at the expense of interoperability. Since interpretation of an unqualified local time zone will fail in approximately 23/24 of the globe, the interoperability problems of unqualified local time are deemed unacceptable for the Internet. Systems that are configured with a local time, are unaware of the corresponding UTC offset, and depend on time synchronization with other Internet systems, MUST use a mechanism that ensures correct synchronization with UTC. Some suitable mechanisms are:

- Use Network Time Protocol [RFC1305] to obtain the time in UTC.
- Use another host in the same local time zone as a gateway to the Internet. This host MUST correct unqualified local times that are transmitted to other hosts.
- Prompt the user for the local time zone and daylight saving rule settings.

4. Date and Time format

This section discusses desirable qualities of date and time formats and defines a format that extends the profile of ISO 8601 for use in Internet protocols.

4.1. Ordering

If date and time components are ordered from least precise to most precise, then a useful property is achieved. Assuming that the time zones of the dates and times are the same (e.g., all in UTC), expressed using the same string (e.g., all "Z" or all "+00:00"), all times have the same number of fractional second digits, and they all have the same suffix (or none), then the date and time strings may be sorted as strings (e.g., using the `strcmp()` function in C) and a time-ordered sequence will result. The presence of optional punctuation would violate this characteristic.

4.2. Human Readability

Human readability has proved to be a valuable feature of Internet protocols. Human readable protocols greatly reduce the costs of debugging since telnet often suffices as a test client and network analyzers need not be modified with knowledge of the protocol. On the other hand, human readability sometimes results in interoperability problems. For example, the date format "10/11/1996" is completely unsuitable for global interchange because it is interpreted differently in different countries. In addition, the date format in (RFC822) has resulted in interoperability problems when people assumed any text string was permitted and translated the three letter abbreviations to other languages or substituted date formats which were easier to generate (e.g. the format used by the C function `ctime`). For this reason, a balance must be struck between human readability and interoperability.

Because no date and time format is readable according to the conventions of all countries, Internet clients SHOULD be prepared to transform dates into a display format suitable for the locality. This may include translating UTC to local time as well as converting from the Gregorian calendar to the viewer's preferred calendar.

4.3. Rarely Used Options

A format which includes rarely used options is likely to cause interoperability problems. This is because rarely used options are less likely to be used in alpha or beta testing, so bugs in parsing are less likely to be discovered. Rarely used options should be made mandatory or omitted for the sake of interoperability whenever possible.

4.4. Redundant Information

If a date/time format includes redundant information, that introduces the possibility that the redundant information will not correlate. For example, including the day of the week in a date/time format introduces the possibility that the day of week is incorrect but the date is correct, or vice versa. Since it is not difficult to compute the day of week from a date (see Appendix A), the day of week should not be included in a date/time format.

4.5. Simplicity

The complete set of date and time formats specified in ISO 8601 [ISO8601] is quite complex in an attempt to provide multiple representations and partial representations. Internet protocols have somewhat different requirements and simplicity has proved to be an important characteristic. In addition, Internet protocols usually need complete specification of data in order to achieve true interoperability. Therefore, the complete grammar for ISO 8601 is deemed too complex for most Internet protocols.

The following section defines a format that in an extension of a profile of ISO 8601 for use on the Internet. It is a conformant subset of the ISO 8601 extended format with additional information optionally suffixed. Simplicity is achieved by making most fields and punctuation mandatory.

4.6. Informative

The format should allow implementations to specify additional important information in addition to the bare timestamp. This is done by allowing implementations to include an informative suffix at the end with as many tags as required, each with a key and value separated by an equals sign. The value can be a hyphen delimited list of multiple values.

In case a key is repeated or conflicted, the implementations should give precedence to whichever value is positioned first.

4.7. Namespaced

Since the suffix can include all sorts of additional information, different standards bodies/ organizations need a way to identify which part adheres to their standards. For this, all information needs to be namespaced. Each key is therefore divided into two hyphen-separated sections: the namespace and the key. For example, the calendar as defined by the Unicode consortium could be included as `u-ca=<value>`.

All single-character namespaces are reserved for BCP47 extensions recorded in the BCP47 extensions registry. For these namespaces:

- Case differences are ignored.
- The namespace is restricted to single alphanum, corresponding to extension singletons ('x' can be used for a private use extension).
- In addition, for CLDR extensions:
  - There must be a `namespace-key` and it is restricted to 2 `alphanum` characters.
  - A `suffix-value` is limited to `3*8alphanum`.

Multi-character namespaces can be registered specifically for use in this format. They are assigned by IANA using the "IETF Review" policy defined by [RFC5226]. This policy requires the development of an RFC, which SHALL define the name, purpose, processes, and procedures for maintaining the subtags. The maintaining or registering authority, including name, contact email, discussion list email, and URL location of the registry, MUST be indicated clearly in the RFC. The RFC MUST specify or include each of the following:

- The specification MUST reference the specific version or revision of this document that governs its creation and MUST reference this section of this document.
- The specification and all keys defined by the specification MUST follow the ABNF and other rules for the formation of keys as defined in this document. In particular, it MUST specify that case is not significant and that keys MUST NOT exceed eight characters in length.
- The specification MUST specify a canonical representation.
- The specification of valid keys MUST be available over the Internet and at no cost.
- The specification MUST be in the public domain or available via a royalty-free license acceptable to the IETF and specified in the RFC.
- The specification MUST be versioned, and each version of the specification MUST be numbered, dated, and stable.
- The specification MUST be stable. That is, namespace keys, once defined by a specification, MUST NOT be retracted or change in meaning in any substantial way.

- The specification MUST include, in a separate section, the registration form reproduced in this section (below) to be used in registering the namespace upon publication as an RFC.
- IANA MUST be informed of changes to the contact information and URL for the specification.

IANA will maintain a registry of allocated multi-character namespaces. This registry MUST use the record-jar format described by the ABNF in [RFC5646]. Upon publication of a namespace as an RFC, the maintaining authority defined in the RFC MUST forward this registration form to <mailto:iesg@ietf.org>, who MUST forward the request to <mailto:iana@iana.org>. The maintaining authority of the namespace MUST maintain the accuracy of the record by sending an updated full copy of the record to <mailto:iana@iana.org> with the subject line "TIMESTAMP FORMAT NAMESPACE UPDATE" whenever content changes. Only the 'Comments', 'Contact_Email', 'Mailing_List', and 'URL' fields MAY be modified in these updates.

Failure to maintain this record, maintain the corresponding registry, or meet other conditions imposed by this section of this document MAY be appealed to the IESG [RFC2028] under the same rules as other IETF decisions (see [RFC2026]) and MAY result in the authority to maintain the extension being withdrawn or reassigned by the IESG.

```
%%
Identifier:
Description:
Comments:
Added:
RFC:
Authority:
Contact_Email:
Mailing_List:
URL:
%%
```

Figure 1: Format of Records in the Timestamp Format Namespace Registry

'Identifier' contains the multi-character sequence assigned to the namespace. The Internet-Draft submitted to define the namespace SHOULD specify which sequence to use, although the IESG MAY change the assignment when approving the RFC.

'Description' contains the name and description of the namespace.

'Comments' is an OPTIONAL field and MAY contain a broader description of the namespace.

'Added' contains the date the namespace's RFC was published in the "date-full" format specified in Figure 2. For example: 2004-06-28 represents June 28, 2004, in the Gregorian calendar.

'RFC' contains the RFC number assigned to the namespace.

'Authority' contains the name of the maintaining authority for the namespace.

'Contact_Email' contains the email address used to contact the maintaining authority.

'Mailing_List' contains the URL or subscription email address of the mailing list used by the maintaining authority.

'URL' contains the URL of the registry for this namespace.

The determination of whether an Internet-Draft meets the above conditions and the decision to grant or withhold such authority rests solely with the IESG and is subject to the normal review and appeals process associated with the RFC process.

4.8. Internet Date/Time Format

The following extension of a profile of [ISO8601] dates SHOULD be used in new protocols on the Internet. This is specified using the syntax description notation defined in [RFC2234].

```
alphanum       = ALPHA / DIGIT

date-year      = 4DIGIT / ("+" / "-") 6DIGIT
date-month     = 2DIGIT ; 01-12
date-mday      = 2DIGIT ; 01-28, 01-29, 01-30, 01-31 based on month/year
date-full      = date-year "-" date-month "-" date-mday

time-hour      = 2DIGIT ; 00-23
time-minute    = 2DIGIT ; 00-59
time-second    = 2DIGIT ; 00-58, 00-59, 00-60 based on leap second rules
time-secfrac   = "." 1*DIGIT
time-partial   = time-hour ":" time-minute ":" time-second [time-secfrac]
time-numoffset = ("+" / "-") time-partial
time-offset    = ("Z" / "z") / time-numoffset
time-full      = time-partial time-offset

time-zone-char = ALPHA / "." / "_"
time-zone-part = time-zone-char *13(time-zone-char / DIGIT / "-" /
"+") ; but not "." or ".."
time-zone-name = time-zone-part *("/" time-zone-part)
time-zone      = "[" time-zone-name "]"

namespace      = 1*alphanum
namespace-key  = 1*alphanum
suffix-key     = namespace ["-" namespace-key]

suffix-value   = 1*alphanum
suffix-values  = suffix-value *("-" suffix-value)
suffix-tag     = "[" suffix-key "=" suffix-values "]"
suffix         = [timezone] *suffix-tag

date-time      = date-full ("T" / "t") time-full suffix
```

Figure 2

> NOTE 1: Per [RFC2234] and ISO8601, the "T" and "Z" characters in this syntax may alternatively be lower case "t" or "z" respectively. Because this date/time format may be used in some environments or contexts that distinguish between the upper- and lower-case letters 'A'-'Z' and 'a'-'z' (e.g. XML), applications that generate this format SHOULD use upper case letters.

> NOTE 2: ISO 8601 defines date and time separated by "T". Applications using this syntax may choose, for the sake of readability, to specify a full-date and full-time separated by (say) a space character.

4.9. Restrictions

The grammar element date-mday represents the day number within the current month. The maximum value varies based on the month and year as follows:

Month Number	Month/Year	Maximum value of date-mday
01	January	31
02	February, normal	28
02	February, leap year	29
03	March	31
04	April	30
05	May	31
06	June	30
07	July	31
08	August	31
09	September	30
10	October	31
11	November	30
12	December	31

Table 1: Days in each month

Appendix B contains sample C code to determine if a year is a leap year.

The grammar element time-second may have the value "60" at the end of months in which a leap second occurs - to date: June (XXXX-06-30T23:59:60Z) or December (XXXX-12-31T23:59:60Z); see Appendix C for a table of leap seconds. It is also possible for a leap second to be subtracted, at which times the maximum value of time-second is "58". At all other times the maximum value of time-second is "59". Further, in time zones other than "Z", the leap second point is shifted by the zone offset (so it happens at the same instant around the globe).

Leap seconds cannot be predicted far into the future. The International Earth Rotation Service publishes bulletins (IERS) that announce leap seconds with a few weeks' warning. Applications should not generate timestamps involving inserted leap seconds until after the leap seconds are announced.

The maximum value of a time-second grammar element inside a time-offset grammar element is always "59".

Although ISO 8601 permits the hour to be "24", this extension of a profile of ISO 8601 only allows values between "00" and "23" for the hour in order to reduce confusion.

4.10. Examples

Here are some examples of Internet date/time format.

```
1985-04-12T23:20:50.52Z
```

Figure 3

This represents 20 minutes and 50.52 seconds after the 23rd hour of April 12th, 1985 in UTC.

```
+001985-04-12T23:20:50.52Z
```

Figure 4

This represents the same instant as the previous example but with the expanded 6-digit year format.

```
1996-12-19T16:39:57-08:00
```

Figure 5

This represents 39 minutes and 57 seconds after the 16th hour of December 19th, 1996 with an offset of -08:00 from UTC. Note that this is equivalent to 1996-12-20T00:39:57Z in UTC.

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

Figure 6

This represents the exact same instant as the previous example but additionally specifies the human time zone associated with it ("Pacific Time") for time-zone-aware implementations to take into account.

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

Figure 7

This represents the exact same instant but it informs calendar-aware implementations that they should project it to the Hebrew calendar.

```
1990-12-31T23:59:60Z
```

Figure 8

This represents the leap second inserted at the end of 1990.

```
1990-12-31T15:59:60-08:00
```

Figure 9

This represents the same leap second in Pacific Standard Time, 8 hours behind UTC.

```
1937-01-01T12:00:27.87+00:19:32.130
```

Figure 10

This represents the same instant of time as noon, January 1, 1937, Netherlands time. Standard time in the Netherlands was exactly 19 minutes and 32.13 seconds ahead of UTC by law from 1909-05-01 through 1937-06-30.

```
1937-01-01T12:00:27.87+00:19:32.130[u-ca=gregory]
```

Figure 11

This represents the exact same instant as the previous example but additionally specifies explicit use of the Gregorian calendar.

```
1937-01-01T12:00:27.87+00:19:32.130[u-ca=islamic-civil]
```

Figure 12

Since there's not a single agreed-upon way to deal with dates in the Islamic calendar, it provides another value to disambiguate between the different interpretations.

```
1937-01-01T12:00:27.87+00:19:32.130[x-foo=bar][x-baz=bat]
```

Figure 13

This timestamp utilizes the private use namespace to declare two additional pieces of information in the suffix that can be interpreted by any compatible implementations and ignored otherwise.

5. Normative references

[RFC2822] Resnick, P., Ed., "Internet Message Format", IETF RFC 2822, IETF RFC 2822, DOI 10.17487/RFC2822, April 2001, <https://www.rfc-editor.org/info/rfc2822>.

[RFC2234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", IETF RFC 2234, IETF RFC 2234, DOI 10.17487/RFC2234, November 1997, <https://www.rfc-editor.org/info/rfc2234>.

[RFC1123] Braden, R., Ed., "Requirements for Internet Hosts — Application and Support", IETF RFC 1123, IETF RFC 1123, DOI 10.17487/RFC1123, October 1989, <https://www.rfc-editor.org/info/rfc1123>.

[RFC1305] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation and Analysis", IETF RFC 1305, IETF RFC 1305, DOI 10.17487/RFC1305, March 1992, <https://www.rfc-editor.org/info/rfc1305>.

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

[RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", IETF RFC 5646, IETF RFC 5646, DOI 10.17487/RFC5646, September 2009, <https://www.rfc-editor.org/info/rfc5646>.

[RFC2026] Bradner, S., "The Internet Standards Process — Revision 3", IETF RFC 2026, IETF RFC 2026, DOI 10.17487/RFC2026, October 1996, <https://www.rfc-editor.org/info/rfc2026>.

[RFC2028] Hovey, R. and S. Bradner, "The Organizations Involved in the IETF Standards Process", IETF RFC 2028, IETF RFC 2028, DOI 10.17487/RFC2028, October 1996, <https://www.rfc-editor.org/info/rfc2028>.

6. Bibliography

[ISO8601] ISO, "Data elements and interchange formats", ISO 8601:1988, June 1988, <https://www.iso.org/standard/15903.html>.

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

[ITU-R-TF] "", ITU-R TF.460-6.

[ZELLER] "Zeller, Chr. Kalender-Formeln. Acta Math. 9 (1887), 131—136. doi:10.1007/BF02406733".

[IERS] "International Earth Rotation Service Bulletins".

Appendix A. Day of the Week

The following is a sample C subroutine loosely based on Zeller's Congruence [ZELLER] which may be used to obtain the day of the week for dates on or after 0000-03-01:

```
char *day_of_week(int day, int month, int year)
{
    int cent;
    char *dayofweek[] = {
        "Sunday", "Monday", "Tuesday", "Wednesday",
        "Thursday", "Friday", "Saturday"
    };

    /* adjust months so February is the last one */
    month -= 2;
    if (month < 1) {
        month += 12;
        --year;
    }
    /* split by century */
    cent = year / 100;
    year %= 100;
    return (dayofweek[((26 * month - 2) / 10 + day + year
                    + year / 4 + cent / 4 + 5 * cent) % 7]);
}
```

Figure 14

Appendix B. Leap Years

Here is a sample C subroutine to calculate if a year is a leap year:

```
/* This returns non-zero if year is a leap year.  Must use 4 digit
    year.
*/
int leap_year(int year)
{
    return (year % 4 == 0 && (year % 100 != 0 || year % 400 == 0));
}
```

Figure 15

Appendix C. Leap Seconds

In 1970 CCIR Recommendation 460 produced international agreement that starting on 1972-01-01 radio broadcast time signals should provide SI seconds with occasional leaps of 1 SI second as necessary to agree with Universal Time. The time scale in radio broadcasts became known as UTC, and the current version of that recommendation is [ITU-R-TF]. Since 1988 IERS has the responsibility for announcing when leap seconds will be introduced into UTC. Further information about leap seconds can be found at the US Navy Oceanography Portal. In particular, it notes that:

> The decision to introduce a leap second in UTC is the responsibility of the International Earth Rotation Service [IERS]. According to the CCIR Recommendation, first preference is given to the opportunities at the end of December and June, and second preference to those at the end of March and September.

When required, insertion of a leap second occurs as an extra second at the end of a day in UTC, represented by a timestamp of the form YYYY-MM-DDT23:59:60Z. A leap second occurs simultaneously in all time zones, so that time zone relationships are not affected. See section Section 4.10 for some examples of leap second times.

The following table is an excerpt from the table maintained by the IERS. The source data are located at the Earth Orientation Parameters Product Centre at Observatoire de Paris.

For dates after the initial adjustment on 1972-01-01 this table shows the date of the leap second, and the difference between the time scale TAI (which is not adjusted by leap seconds) and UTC after that leap second.

UTC Date	TAI - UTC After Leap Second
1972-06-30	11
1972-12-31	12
1973-12-31	13
1974-12-31	14
1975-12-31	15
1976-12-31	16
1977-12-31	17
1978-12-31	18
1979-12-31	19
1981-06-30	20
1982-06-30	21
1983-06-30	22
1985-06-30	23
1987-12-31	24
1989-12-31	25
1990-12-31	26
1992-06-30	27
1993-06-30	28
1994-06-30	29
1995-12-31	30
1997-06-30	31
1998-12-31	32
2005-12-31	33
2008-12-31	34

UTC Date	TAI - UTC After Leap Second
2012-06-30	35
2015-06-30	36
2016-12-31	37

Table 2: Historic leap seconds

Author's Address

Ujjwal Sharma
Igalia, S.L.
Email: ryzokuken@igalia.com