Workgroup:
Serialising Extended Data About Times and
Events
Internet-Draft:
draft-ietf-sedate-datetime-extended-06
Updates: 3339 (if approved)
Published: 16 September 2022
Intended Status: Standards Track
Expires: 20 March 2023
Authors: U. Sharma C. Bormann
Igalia, S.L. Universität Bremen TZI

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.

It updates RFC3339 in the specific interpretation of the local offset Z, which is no longer understood to "imply that UTC is the preferred reference point for the specified time"; see Section 2.

The present version (-06) reflects the discussions at IETF 114. In particular, RFC 3339 is now updated with respect to the semantics of time zone offset Z.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-sedate-datetime-extended/.

Discussion of this document takes place on the Serialising Extended Data About Times and Events (SEDATE) Working Group mailing list (mailto:sedate@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/sedate/. Subscribe at https://www.ietf.org/mailman/listinfo/sedate/.

Source for this draft and an issue tracker can be found at https://github.com/ietf-wg-sedate/draft-ietf-sedate-datetime-extended.

Status of This Memo

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 20 March 2023.

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 [ISO8601:1988] standardizes a widely-adopted timestamp format, which forms the basis of the Internet Date/Time Format [RFC3339]. However, this format only allows timestamps to contain very little additional relevant information, which means that, beyond 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 events such as daylight saving time transitions. Most of these applications attach the time zone to the timestamp in a non-standard format, at least one of which is fairly well-adopted [JAVAZDT]. Furthermore, applications might want to attach even more information to the timestamp, including but not limited to the calendar system in which it should be represented.

1.1. Scope

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

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

We refer to this format as the Internet Extended Date/Time Format (IXDTF).

This document does not address extensions to the format where the semantic result is no longer a fixed timestamp that is referenced to a (past or future) UTC time. For instance, it does not address:

- Future time given as a local time in some specified time zone, where changes to the definition of that time zone (e.g., a

political decision to enact or rescind daylight saving time) affect the instant in time corresponding with the timestamp.

* "Floating time", i.e., a local time without information describing the UTC offset or time zone in which it should be interpreted.

* The use of timescales different from UTC, such as TAI.

However, additional information augmenting a fixed timestamp may be sufficient to detect an inconsistency between intention and the actual information in the timestamp, e.g., between the UTC offset and time zone name in the timestamp. For instance, such an inconsistency might arise because of:

* political decisions as discussed above, or

* updates to time zone definitions being applied at different times by timestamp producers and receivers, or

* errors in the applications producing and consuming such a timestamp.

While the information available is not generally sufficient to resolve the inconsistency, it may be used to initiate some out of band processing to obtain sufficient information for such a resolution.

In order to address some of the requirements implied here, future related specifications might define syntax and semantics of strings similar to [RFC3339]. Note that the extension syntax defined in the present document is designed in such a way that it can be useful for such specifications as well.

1.2. Definitions

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.

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.

UTC is often mistakenly referred to as GMT, an earlier timescale UTC was designed to be a useful successor for.

ABNF: Augmented Backus-Naur Form, a format used to represent permissible strings in a protocol or language, as defined in [RFC5234]. The rules defined in Appendix B of [RFC5234] are imported implicitly.

Internet Extended Date/Time Format (IXDTF): The date/time format defined in Section 4 of this document.

Timestamp: An unambiguous representation of some instant in time.

UTC Offset: Difference between a given local time and UTC, usually given in negative or positive hours and minutes. For example, local time in New York in the wintertime is 5 hours behind UTC, so its UTC offset is "-05:00".

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". (Definition from Section 2 of [RFC3339].)

Time Zone: A set of rules representing the relationship of local time to UTC for a particular place or region. Mathematically, a time zone can be thought of as a function that maps timestamps to UTC offsets. Time zones can deterministically convert a timestamp to local time. They can also be used in the reverse direction to convert local time to a timestamp, with the caveat that some local times may have zero or multiple possible timestamps due to nearby Daylight Saving Time changes or other changes to the UTC offset of that time zone. Unlike the UTC offset of a timestamp which makes no claims about the UTC offset of other related timestamps (and which is therefore unsuitable for performing local-time operations such as "one day later"), a time zone also defines how to derive new timestamps based on differences in local time. For example, to calculate "one day later than this timestamp in San Francisco", a time zone is required because the UTC offset of local time in San Francisco can change from one day to the next.

IANA Time Zone: A named time zone that is included in the Time Zone Database (often called tz or zoneinfo) maintained by IANA [TZDB] [BCP175]. Most IANA time zones are named for the largest city in a particular region that shares the same time zone rules, e.g. Europe/Paris or Asia/Tokyo [TZDB-NAMING]. Special IANA time zones such as Etc/GMT+10 can be used to represent timestamps outside

country boundaries, e.g. a buoy in the middle of the Pacific Ocean (note that the Etc/GMT+10 time zone has a constant UTC Offset of -10:00 [sic!]).

Note that the rules defined for a named IANA time zone can change over time. The use of a named IANA time zone implies that the intent is for the rules that are current at the time of interpretation to apply, i.e., the additional information conveyed by using that time zone name is to change with the changed rules as recorded in the IANA time zone database.

Offset Time Zone: A time zone defined by a specific UTC offset, e.g. +08:45 and serialized using as its name the same numeric UTC offset format used in an RFC 3339 timestamp. Although serialization with offset time zones is supported in this document for backwards compatibility with java.time.ZonedDateTime [JAVAZDT], use of offset time zones is strongly discouraged. In particular, programs **MUST NOT** copy the UTC offset from a timestamp into an offset time zone in order to satisfy another program which requires a time zone annotation in its input. Doing this will improperly assert that the UTC offset of timestamps in that location will never change, which can result in incorrect calculations in programs that add, subtract, or otherwise derive new timestamps from the one provided. For example, 2020-01-01T00:00+01:00[Europe/Paris] will let programs add six months to the timestamp while adjusting for Summer Time (Daylight Saving Time). But the same calculation applied to 2020-01-01T00:00+01:00[+01:00] will produce an incorrect result that will be off by one hour in the timezone Europe/Paris.

CLDR: Common locale data repository [CLDR], a project of the Unicode Consortium to provide locale data to applications.

For more information about timescales, see Appendix E of [RFC1305], Section 3 of [ISO8601:1988], and the appropriate ITU documents [ITU-R-TF.460-6].

2. Updating RFC 3339

Section 4.3 of [RFC3339] states that an offset given as Z or +00:00 implies that "UTC is the preferred reference point for the specified time". The offset -00:00 is provided as a way to express that "the time in UTC is known, but the offset to local time is unknown".

This convention mirrors a similar convention for date/time information in email headers, described in Section 3.3 of [RFC5322] and introduced earlier in Section 3.3 of [RFC2822]. The latter convention is in actual use, while the former always was handicapped by the fact that [ISO8601:1988] does not actually allow -00:00.

Implementations that needed to express the semantics of -00:00 therefore tended to use Z as a "neutral" offset instead.

This specification updates RFC3339, aligning it with the actual practice of interpreting the local offset Z: this is no longer understood to "imply that UTC is the preferred reference point for the specified time".

Note that the semantics of the local offset +00:00 is not updated; this retains the implication that UTC is the preferred reference point for the specified time.

Note also that the fact that [ISO8601:1988] does not allow -00:00 as a local offset reduces the level of interoperability that can be achieved in using this feature; the present specification however does not formally deprecate this syntax. For the intents and purposes of the present specification, the local offset Z can be used in its place.

3. Internet Extended Date/Time format (IXDTF)

This section discusses desirable qualities of formats for the timestamp extension suffix and defines such a format that extends [RFC3339] for use in Internet protocols.

3.1. Informative

The format is intended to allow implementations to specify additional important information in addition to the bare timestamp.

This is done by defining *tags*, each with a *key* and a *value* separated by an equals sign. The value of a tag can be one or more items delimited by hyphen/minus signs.

Applications can build an informative timestamp *suffix* using any number of these tags.

Keys are lower-case only. Values are case-sensitive unless otherwise specified.

When a suffix contains a repeated key or otherwise conflicting tags, implementations **MUST** give precedence to whichever value is positioned first. I'd rather place a MUST NOT for this case, first. This definitely needs to be expanded into some general text about error handling.--- cabo

3.2. Registered

Actual suffix tag keys are registered by supplying the information specified in this section. This information is modeled after that

specified for the media type registry [RFC6838]; if in doubt, the provisions of this registry should be applied analogously.

Key Identifier: The key (conforming to suffix-key in Section 4.1)

Registration status: "Provisional" or "Permanent"

Description: A very brief description of the key.

Change controller: Who is in control of evolving the specification governing values for this key. This information can include email addresses of contact points and discussion lists, and references to relevant web pages (URLs).

Reference: A reference. For permanent tag keys, this includes a full specification. For provisional tag keys, there is an expectation that some information is available even if that does not amount to a full specification; in this case, the registrant is expected to improve this information over time.

Key names that start with an underscore are intended for experiments in controlled environments and cannot be registered; such keys **MUST NOT** be used for interchange and **MUST** be rejected by implementations not specifically configured to take part in such an experiment. See [BCP178] for a discussion about the danger of experimental keys leaking out to general production and why that **MUST** be prevented.

3.3. Optionally Critical

For the format defined here, suffix tags are always *optional*: They can be added or left out as desired by the generator of the string in Internet Extended Date/Time Format (IXDTF). (An application might require the presence of specific suffix tags, though.)

Without further indication, they are also *elective*: Even if included in the IXDTF string, the recipient is free to ignore the suffix tag. Reasons might include that the recipient does not implement (or know about) the specific suffix key, or that it does recognize the key but cannot act on the value provided.

A suffix tag may also indicate that it is *critical*: The recipient is advised that it **MUST NOT** act on the Internet Extended Date/Time Format (IXDTF) string unless it can process the suffix tag as specified. A critical suffix tag is indicated by following its opening bracket with an exclamation mark (see critical-flag in Section 4.1).

IXDTF strings such as:

```
2022-07-08T00:14:07+01:00[Europe/Paris]
```

are internally inconsistent (see Section 3.4), as Europe/Paris did not use a time zone offset of +01:00 in July 2022. The time zone hint given in the suffix tag is elective, though, so the recipient is not required to act on the inconsistency; it can treat the Internet Date/Time Format string as if it were:

```
2022-07-08T00:14:07+01:00
```

Similar with:

```
2022-07-08T00:14:07+01:00[knort=blargel]
```

(assuming that the recipient does not understand the suffix key knort).

Note that as per Section 2 (see also Section 3.4), the IXDTF string:

2022-07-08T00:14:07Z[Europe/Paris]

does not exhibit such an inconsistency, as the local offset of Z does not imply a specific preferred time zone of interpretation. With the knowledge of how time zone offsets applied to Europe/Paris in the summer of 2022, it is equivalent to:

2022-07-08T02:14:07+02:00[Europe/Paris]

In contrast to this elective use of a suffix tag,

```
2022-07-08T00:14:07+01:00[!Europe/Paris]
2022-07-08T00:14:07Z[!knort=blargel]
```

each have an internal inconsistency or an unrecognized suffix key/value that are marked as critical, so a recipient **MUST** treat the IXDTF string as erroneous.

Note that this does not mean that an application is disallowed to perform additional processing on inconsistent or unrecognized elective suffix tags, e.g., asking the user how to resolve the inconsistency. It means it is not required to do so with elective suffix tags, but is required to reject or perform some other error handling when encountering inconsistent or unrecognized suffix tags marked as critical.

3.4. Inconsistent time-offset/Time-Zone Information

An RFC 3339 timestamp can contain a time-offset value that indicates the offset between local time and UTC (see Section 4 of [RFC3339], noting that Section 2 of the present specification updates Section 4.3 of [RFC3339]).

The information given in such a time-offset value can be inconsistent with the information provided in a time zone suffix for an IXDTF timestamp.

For example, a calendar application could store an IXDTF string representing a far-future meeting in a particular time zone. If that time zone's definition is subsequently changed to abolish Daylight Saving Time, IXDTF strings that were originally consistent may now be inconsistent.

In case of inconsistent time-offset and time zone suffix, if the critical flag is used on the time zone suffix, an application **MUST** act on the inconsistency. If the critical flag is not used, it **MAY** act on the inconsistency. Acting on the inconsistency may involve rejecting the timestamp, or resolving the inconsistency via additional information such as user input and/or programmed behavior.

For example, the IXDTF timestamps in Figure 1 represent 00:14:07 UTC, indicating a local time with a time-offset of +00:00. However, because Europe/London used offset +01:00 in July 2022, the timestamps are inconsistent:

```
2022-07-08T00:14:07+00:00[!Europe/London]
2022-07-08T00:14:07+00:00[Europe/London]
```

Figure 1: Inconsistent IXDTF timestamps

As per Section 4.3 of [RFC3339] as updated by Section 2, IXDTF timestamps may also forego indicating local time information in their [RFC3339] part. The IXDTF timestamps in Figure 2 (which represent the same instant in time as the strings in Figure 1) are not inconsistent because they do not assert any particular local time nor local offset in their [RFC3339] part. Instead, applications that receive these strings can base their local offset and local time calculations on the time zone suffix given, i.e., using the Europe/London time zone rules.

```
2022-07-08T00:14:07Z[!Europe/London]
2022-07-08T00:14:07Z[Europe/London]
2022-07-08T00:14:07-00:00[!Europe/London]
2022-07-08T00:14:07-00:00[Europe/London]
```

Figure 2: No inconsistency in IXDTF timestamps

4. Syntax Extensions to RFC 3339

4.1. ABNF

The following rules extend the ABNF syntax defined in [RFC3339] in order to allow the inclusion of an optional suffix.

The Internet Extended Date/Time Format (IXDTF) is described by the rule date-time-ext.

date-time and time-numoffset are imported from Section 5.6 of [RFC3339], ALPHA and DIGIT from Appendix B.1 of [RFC5234].

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

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

suffix-value      = 1*alphanum
suffix-values     = suffix-value *("-" suffix-value)
suffix-tag        = "[" critical-flag
                        suffix-key "=" suffix-values "]"
suffix            = [time-zone] *suffix-tag

date-time-ext     = date-time suffix

critical-flag     = [ "!" ]

alphanum          = ALPHA / DIGIT
lcalpha           = %x61-7A
```

Figure 3: ABNF grammar of extensions to RFC 3339

Note that a time-zone is syntactically similar to a suffix-tag, but does not include an equals sign. This special case is only available for time zone tags.

4.2. Examples

Here are some examples of Internet Extended Date/Time Format (IXDTF).

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

Figure 4: RFC 3339 date-time with time zone offset

Figure 4 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 the same instant in time as 1996-12-20T00:39:57Z, expressed in UTC.

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

Figure 5: Adding a time zone name

Figure 5 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 6: Projecting to the Hebrew calendar

Figure 6 represents the exact same instant, but it informs calendar-aware implementations (see Section 5) that they should project it to the Hebrew calendar.

```
1996-12-19T16:39:57-08:00[_foo=bar][_baz=bat]
```

Figure 7: Adding experimental tags

Figure 7, based on Figure 4, utilizes keys identified as experimental by a leading underscore to declare two additional pieces of information in the suffix; these can be interpreted by implementations that take part in the controlled experiment making use of these tag keys.

5. The u-ca Suffix Key: Calendar Awareness

Out of the possible suffix keys, the suffix key u-ca is allocated to indicate the calendar in which the date/time is preferably presented.

A calendar is a set of rules defining how dates are counted and consumed by implementations. The set of suffix values allowed for this suffix key is as defined by the [CLDR] data for [TR35]. At the time of writing, this information is collected in [CLDR-CALENDAR].

6. IANA Considerations

RFC Editor: please replace RFCthis with the RFC number of this RFC and remove this note.

IANA is requested to establish a registry called "Timestamp Suffix Tag Keys". Each entry in the registry shall consist of the information described in Section 3.2. Initial contents of the registry are specified in Table 1.

Key Identifier	Registration status	Description:	Change controller	Reference
u-ca	Permanent	Preferred Calendar for Presentation	IESG	Section 5 of RFCthis

Table 1: Initial Content of Timestamp Suffix Tag Keys registry

The registration policy [RFC8126] is "Specification Required" for permanent entries, and "Expert Review" for provisional ones. In the second case, the expert is instructed to ascertain that a basic specification does exist, even if not complete or published yet.

7. Security Considerations

TBD

8. References

8.1. Normative References

[BCP175] Lear, E. and P. Eggert, "Procedures for Maintaining the Time Zone Database", BCP 175, RFC 6557, DOI 10.17487/RFC6557, February 2012, <https://www.rfc-editor.org/info/rfc6557>.

[BCP178] Saint-Andre, P., Crocker, D., and M. Nottingham, "Deprecating the "X-" Prefix and Similar Constructs in Application Protocols", BCP 178, RFC 6648, DOI 10.17487/RFC6648, June 2012, <https://www.rfc-editor.org/info/rfc6648>.

[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>.

[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>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[RFC8126] 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>.

[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>.

8.2. Informative References

[CLDR] "Unicode CLDR Project", <https://cldr.unicode.org>.

[CLDR-CALENDAR] "cldr/common/bcp47/calendar.xml", <https://github.com/unicode-org/cldr/blob/main/common/bcp47/calendar.xml>.

[IERS] "International Earth Rotation Service Bulletins", <https://www.iers.org/IERS/EN/Publications/Bulletins/bulletins.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>.

[ITU-R-TF.460-6] "ITU-R TF.460-6. Standard-frequency and time-signal emissions", February 2002, <https://www.itu.int/rec/R-REC-TF.460/en>.

[JAVAZDT] "Java SE 8, java.time.format, DateTimeFormatter: ISO_ZONED_DATE_TIME", <https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#ISO_ZONED_DATE_TIME>.

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

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

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <https://www.rfc-editor.org/info/rfc5322>.

[TR35] "Unicode Technical Standard", <https://unicode-org.github.io/cldr/ldml/tr35-dates.html#Supplemental_Calendar_Data>.

[TZDB] "Sources for time zone and daylight saving time data", <https://data.iana.org/time-zones/tz-link.html>.

[TZDB-NAMING] "Theory and pragmatics of the tz code and data", <https://data.iana.org/time-zones/theory.html>.

Acknowledgements

Richard Gibson provided some editorial improvements.

Contributors

Justin Grant

Email: justingrant.ietf.public@gmail.com

Authors' Addresses

Ujjwal Sharma
Igalia, S.L.
Bugallal Marchesi, 22, 1º
15008 A Coruña
Spain

Email: ryzokuken@igalia.com

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

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