Abstract

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

// The present version (-05) includes a few changes that are intended
// for discussion at IETF 114. In particular, the introduction of
// the critical-flag exposes the fact that some RFC 3339
// implementations assign different semantics to the time zone
// offsets Z and +00:00; we may want to consider ways to cope with
// this apparently common deviation. Also, the name of the format is
// still up for suggestions that improve upon the current choice.

About This Document

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

Status of This Memo

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

* 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 3 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 [JAVA Zealand], 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. 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.
2.1. 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 case-sensitive. 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

2.2. 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 3.1)

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.

2.3. 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 3.1).

IXDTF strings such as:

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

are internally inconsistent, as Europe/Paris does not use a time zone offset of 0 (which is indicated in the Z, an abbreviation for +00:00), nor 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:

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

Similar with:

2022-07-08T00:14:07Z[knort=blargel]

However,
all have an internal inconsistency or an unrecognized suffix key/value, 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 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. Syntax Extensions to RFC 3339

3.1. 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 / "." / "_"

Figure 1: 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.

3.2.  Examples

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

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

Figure 2: RFC 3339 date-time with time zone offset

Figure 2 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 3: Adding a time zone name
Figure 3 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 4: Projecting to the Hebrew calendar

Figure 4 represents the exact same instant, but it informs calendar-aware implementations (see Section 4) that they should project it to the Hebrew calendar.

1996-12-19T16:39:57-08:00[foo=bar][baz=bat]

Figure 5: Adding experimental tags

Figure 5, based on Figure 2, 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.

4. 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].

5. 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 2.2. Initial contents of the registry are specified in Table 1.
### 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.

#### 6. Security Considerations
TBD

#### 7. References

7.1. Normative References

7.2. Informative References


Acknowledgements

Richard Gibson provided some editorial improvements.

Contributors

Justin Grant
Email: justingrant.ietf.public@gmail.com

Your Name Here

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