Stream: Internet Engineering Task Force (IETF)
RFC: 9557
Updates: 3339
Category: Standards Track
Published: April 2024
ISSN: 2070-1721
Authors: U. Sharma, *Igalia, S.L.*; C. Bormann, *Universität Bremen TZI*

RFC 9557
Date and Time on the Internet: Timestamps with Additional Information

Abstract

This document defines an extension to the timestamp format defined in RFC 3339 for representing additional information, including a time zone.

It updates RFC 3339 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".

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/rfc9557.

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. [ISO8601-1:2019] standardizes a widely adopted timestamp format, an earlier version of which [ISO8601:1988] formed the basis of the Internet Date/Time Format [RFC3339]. However, this format allows timestamps to contain very little additional relevant information. 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 a pressing issue for applications that handle each such instant in time with an associated time zone name in order to take into account events such as daylight saving time transitions. Many 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.

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.

1.1. Scope

[RFC3339] defines a syntax for timestamps to represent date and time in the Internet. The present document defines an extension syntax that achieves 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 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 (such as a political decision to enact or rescind daylight saving time) affect the instant in time represented by the timestamp;
- "floating time", i.e., a local time without information describing the UTC offset or time zone in which it should be interpreted; or
- the use of timescales different from UTC, such as International Atomic Time (TAI).

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

- political decisions, as discussed above,
- updates to time zone definitions being applied at different times by timestamp producers and receivers, or
- errors in programs producing and consuming timestamps.

While the information available in an IXDTF string is not generally sufficient to resolve an 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, related specifications in the future might define syntax and semantics of strings similar to those described in [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 Systems Service [IERS]. From 1972 through 1987, UTC was maintained entirely by the 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 (Greenwich Mean Time), an earlier timescale for which UTC was designed to be a useful successor.

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.

IXDTF: Internet Extended Date/Time Format, the date/time format defined in Section 4 of this document.

Timestamp: An unambiguous representation of a particular 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 the city of New York, NY, USA in the wintertime in 2023 was 5 hours behind UTC, so its UTC offset was `-05:00`.

Z: A suffix that, when applied to a time, denotes a UTC offset of `00:00`; often pronounced "Zulu" from the ICAO phonetic alphabet representation of the letter "Z". (The definition is from Section 2 of [RFC3339]; see the International Civil Aviation Organization (ICAO) document [ICAO-PA] for the phonetic alphabet mentioned.)

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, California", 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].

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: the additional information conveyed by using that time zone name is to change with any rule changes 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 [RFC3339] timestamp, for example:

```
2022-07-08T00:14:07+08:45[+08:45]
```

An offset in the suffix that does not repeat the offset of the timestamp is inconsistent (see Section 3.4).

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 that requires a time zone suffix 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). However, 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 time zone `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]. (Note: [RFC1305] was obsoleted by [RFC5905], which no longer contains the Appendix E referenced here.)

2. Updating RFC 3339

2.1. Background

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 that is described in Section 3.3 of [RFC5322] and introduced earlier in Section 3.3 of [RFC2822]. This email header convention is in actual use, while its adaptation into [RFC3339] was always compromised by the fact that [ISO8601:2000] and later versions do not actually allow `-00:00`.

Implementations that needed to express the semantics of `-00:00` therefore tended to use `Z` instead.

2.2. Update to RFC 3339

This specification updates Section 4.3 of [RFC3339], aligning it with the actual practice of interpreting the offset `Z` to mean the same as `-00:00`: "the time in UTC is known, but the offset to local time is unknown".

Section 4.3 of [RFC3339] is revised to read as follows:

> If the time in UTC is known, but the offset to local time is unknown, this can be represented with an offset of "Z". (The original version of this specification provided `-00:00` for this purpose, which is not allowed by [ISO8601:2000] and therefore is less interoperable; Section 3.3 of [RFC5322] describes a related convention for email, which does not have this problem). This differs semantically from an offset of `+00:00`, which implies that UTC is the preferred reference point for the specified time.

2.3. Notes

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.

Also note that the fact that [ISO8601:2000] and later do not allow `-00:00` as a local offset reduces the level of interoperability that can be achieved in using this feature; however, the present specification does not formally deprecate this syntax. With the update to [RFC3339], the local offset `Z` should now 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 the IXDTF format, which extends [RFC3339] for use in Internet protocols.

3.1. Format of Extended Information

The format allows applications to specify additional important information in addition to a bare [RFC3339] 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 lowercase only. Values are case-sensitive unless otherwise specified.

See Section 3.3 for the handling of inconsistent information in a suffix.

3.2. Registering Keys for Extended Information Tags

Suffix tag keys are registered by supplying the information specified in this section. This information is modeled after that specified for the "Media Types" 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, 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. Optional Generation and Elective vs. Critical Consumption

For the IXDTF format, suffix tags are always *optional*. They can be added or left out as desired by the generator of the string. (An application might require the presence of specific suffix tags, though.)

Without further indication, suffix tags are also *elective*. The recipient is free to ignore any suffix tag included in an IXDTF string. 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 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).

For example, IXDTF strings such as:

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

are internally inconsistent (see Section 3.4), because `Europe/Paris` did not use a time zone offset of `+01:00` in July 2022. However, the time zone hint given in the suffix tag is elective, 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
```

> 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. Using the Time Zone Database rules for `Europe/Paris` in the summer of 2022, it is equivalent to:
>
> ```
> 2022-07-08T02:14:07+02:00[Europe/Paris]
> ```

Similarly, an unknown suffix may be entirely ignored:

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

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

In contrast to this elective use of a suffix tag,

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

each have an internal inconsistency or an unrecognized suffix key/value that is marked as critical, so a recipient **MUST** treat these IXDTF strings as erroneous. This means that the application **MUST** reject the data or perform some other error handling, such as asking the user how to resolve the inconsistency (see Section 3.4).

Note that applications **MAY** also perform additional processing on inconsistent or unrecognized elective suffix tags, such as asking the user how to resolve the inconsistency. While they are not required to do so with elective suffix tags, they are required to reject or perform some other error handling when encountering inconsistent or unrecognized suffix tags marked as critical.

An application that encounters duplicate use of a suffix key in elective suffixes and does not want to perform additional processing on this inconsistency **MUST** choose the first suffix that has that key, that is,

```
2022-07-08T00:14:07Z[u-ca=chinese][u-ca=japanese]
2022-07-08T00:14:07Z[u-ca=chinese]
```

are then treated the same.

3.4. Inconsistent `time-offset` and Time Zone Information

An [RFC3339] 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 an inconsistency between `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, where the first case is one for which the application **MUST** act on the inconsistency (the time zone suffix is marked critical) and the second case is one for which it **MAY** act on the inconsistency (the time zone suffix is elective).

```
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 by using `Z` instead of a numeric time zone offset. 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 calculate the local offset and local time using the rules of the time zone suffix, such as `Europe/London` in the example in Figure 2, which like Figure 1 has a case with a time zone suffix marked critical (i.e., the intention is that the application must understand the time zone information) and one marked elective, which then only is provided as additional information.

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

Figure 2: No Inconsistency in IXDTF Timestamps

Note that `-00:00` may be used instead of `Z` because they have the same meaning according to Section 2, but `-00:00` is not allowed by [ISO8601:2000] and later so `Z` is preferred.

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], and `ALPHA` and `DIGIT` are imported from Appendix B.1 of [RFC5234].

```
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)
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.

The ABNF definition of `time-zone-part` matches "." and "..", which are both explicitly excluded (see the note below on `time-zone-part`).

`time-zone-name` is intended to be the name of an IANA Time Zone. As a generator and a recipient may be using different revisions of the Time Zone Database, recipients may not be aware of such an IANA Time Zone name and should treat such a situation as any other inconsistency.

> Note: At the time of writing, the length of each `time-zone-part` is limited to a maximum of 14 characters by the rules in [TZDB-NAMING]. One platform is known to enforce this limit, and a time zone name on another platform is known to exceed this limit. As the `time-zone-name` will ultimately have to be looked up in the local database, which therefore has control over the length, the `time-zone-part` production in Figure 3 is deliberately permissive.

4.2. Examples

This section contains 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 19, 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 in time as the previous example but additionally specifies the human time zone associated with it ("Pacific Time") for time-zone-aware applications 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 in time, but it informs calendar-aware applications (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 the set of values defined for the Unicode Calendar Identifier [TR35]. [CLDR-LINKS] provides links to the most recent information about [CLDR], both stable and specific development stages.

6. IANA Considerations

IANA has created a registry called "Timestamp Suffix Tag Keys" in a new registry group titled "Internet Date/Time Format". Each entry in the registry shall consist of the information described in Section 3.2. The 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	IETF	Section 5 of RFC 9557

Table 1: Initial Contents of Timestamp Suffix Tag Keys Registry

The registration policy [BCP26] is "Specification Required" for permanent entries and "Expert Review" for provisional ones. In the second case, the experts are instructed to ascertain that a basic specification does exist, even if not complete or published yet.

The experts are also instructed to be frugal in the allocation of key identifiers that are suggestive of generally applicable semantics, keeping them in reserve for suffix keys that are likely to enjoy wide use and can make good use of the key identifier's conciseness. If the experts become aware of key identifiers that are deployed and in use, they may also initiate a registration on their own if they deem such a registration can avert potential future collisions.

7. Security Considerations

7.1. Excessive Disclosure

The ability to include various pieces of ancillary information with a timestamp might lead to excessive disclosure. An example for possibly excessive disclosure is given in Section 7 of [RFC3339]. Similarly, divulging information about the calendar system or the language of choice may provide more information about the originator of a timestamp than the data minimization principle would permit [DATA-MINIMIZATION]. More generally speaking, generators of IXDTF timestamps need to consider whether information to be added to the timestamp is appropriate to divulge to the recipients of this information and need to err on the side of minimizing such disclosure if the set of recipients is not under control of the originator.

7.2. Data Format Implementation Vulnerabilities

As usual when extending the syntax of a data format, this can lead to new vulnerabilities in implementations parsing and processing the format. No considerations are known for the IXDTF syntax that would pose concerns that are out of the ordinary.

7.3. Operating with Inconsistent Data

Information provided in the various parts of an IXDTF string may be inconsistent in interesting ways, both with the extensions defined in this specification (for instance, see Section 3.4) and with future extensions still to be defined. Where consistent interpretation between multiple actors is required for security purposes (e.g., where timestamps are embedded as parameters in access control information), only extensions that have a well-understood and shared resolution of such inconsistent data can be employed.

8. References

8.1. Normative References

[BCP175] Best Current Practice 175, <https://www.rfc-editor.org/info/bcp175>.
At the time of writing, this BCP comprises the following:

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] Best Current Practice 178, <https://www.rfc-editor.org/info/bcp178>.
At the time of writing, this BCP comprises the following:

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>.

[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>.

[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>.

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

[CLDR-LINKS] Unicode CLDR, "Stable Links Info", <https://cldr.unicode.org/stable-links-info>.

[DATA-MINIMIZATION] Arkko, J., "Emphasizing data minimization among protocol participants", Work in Progress, Internet-Draft, draft-arkko-iab-data-minimization-principle-05, 10 July 2023, <https://datatracker.ietf.org/doc/html/draft-arkko-iab-data-minimization-principle-05>.

[ICAO-PA] International Civil Aviation Organization, "Annex 10 to the Convention on International Civil Aviation: Aeronautical Telecommunications; Volume II Communication Procedures including those with PANS status", 7th ed., July 2016, <https://store.icao.int/annex-10-aeronautical-telecommunications-volume-ii-communication-procedures-including-those-with-pans-status>.

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

[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>.

[ISO8601:2000] ISO, "Data elements and interchange formats -- Information interchange -- Representation of dates and times", ISO 8601:2000, December 2000, <https://www.iso.org/standard/26780.html>.

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

[JAVAZDT] Oracle, "Class 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>.

[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>.

[TR35] Davis, M., Ed., "Unicode Technical Standard #35: Unicode Locale Data Markup Language (LDML)", <https://www.unicode.org/reports/tr35/#UnicodeCalendarIdentifier>.

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

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

Acknowledgements

This specification benefits from work prepared by ECMA TC39, specifically in the Temporal proposal.

Richard Gibson and Justin Grant provided editorial improvements. The SEDATE WG Chairs Mark McFadden and Bron Gondwana, the latter also in his role as CALEXT WG Chair, helped set up the structures needed to navigate the multi-SDO environment. John Klensin critically accompanied the development of this specification, which led to significant improvements. The authors would also like to especially thank Francesca Palombini for her AD review and for her overall guidance during the development process.

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