

# SEDATE

March 21st, (Monday), 12:00–13:00 UTC  
(13:00–14:00 CET, 05:00–06:00 PDT)

## Proposed structure of discussion:

- status
- confirmation of recent directions taken on the draft
  - meaning of referencing a named IANA time zone (#6)
  - X-Dash (RFC6648) danger (#2)
- need work (#4, #9, #17, #18)
- editorial (#14)

# Status: 9 issues open on github

- 2 are out of scope now.
  - Floating Times and future time (i.e., we stick within the RFC3339 envelope).
- 1 marked philosophical (#5), → no direct text changes
- 1 solved in -02 but possibly needs more discussion (#2)
- 1 is marked editorial (#14)
- 4 need work (#4, #9, #17, #18)

# **draft-ietf-sedate-datetime-extended-03**

draft-ietf-sedate-datetime-extended-03, 2022-03-07

Main new item:

— #16: Get rid of namespaces

# draft-ietf-sedate-datetime-extended-04

draft-ietf-sedate-datetime-extended-04, 2022-03-21

Main new item:

- #15 (close #12): Support for [+08:00]  
(static offsets in the position of timezones, as in Java)
- #6: meaning of named IANA time zone (#6)  
(closed with new text, but had no discussion so far)

# meaning of named IANA time zone (#6)

commit bf12f13:

*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.*

Discuss.

# X-Dash (#2)

*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.*

# #17: How to handle conflicts between timestamp offset vs. bracketed timezone

(Continuation of closed #10)

"invitation to regrettable error"?

Or exactly the information needed to detect and solve a problem?

Maybe just strengthen the language:  
offset will always win in case of a conflict?  
(or will it?)



## #18: **MUST** understand (Elective/Critical)

- -04: all tags are completely ~~optional~~ **elective**
  - can be safely ignored by any implementation
    - that may not understand
    - that deems it irrelevant for its use case

Should we include syntax to mark a certain tag **critical**?

- implementation that doesn't understand/implement **MUST** throw an error

1996-12-19T16:39:57-08:00[America/Los\_Angeles][u-ca=hebrew]

(**Elective**, just a hint, ignore option if not understood)

1996-12-19T16:39:57-08:00[America/Los\_Angeles][u-ca!hebrew]

(**Critical**, must understand/must reject date if not understood)

2022-12-19T16:39:57!-08:00[!America/Los\_Angeles][!u-ca=hebrew]

(Aargh.)

## #9: We need to name the baby

Internet Date/Time Format

- of little definitional sharpness
  - but then, [RFC3339](#) was sharp but meh
- overpretentious

[extended RFC3339](#)?

Any proposals that are even worse?

## #4: Need at least one extension defined → good examples

Proposal: actually define u-ca (calendar hint)

This will bring up issues such as:

- alphanum vs. ASCII-- vs. Unicode in an extension tag..
- labels (→ alphanum)
- values (→ \_\_\_\_\_)

## #14 (editorial)

Convention for sequencing ABNF productions:

- RFC 3339: bottom-up
- Maybe better to order top-down
  - Entry points no longer buried