

30 June 2025 | MessageFormat Working Group Teleconference

Attendees:

- Addison Phillips - Unicode (APP) - chair
- Mihai Niță \- Google (MIH)
- Tim Chevalier \- Igalia (TIM)
- Eemeli Aro \- Mozilla (EAO)
- Richard Gibson \- OpenJSF (RGN)
- Shane Carr \- Google (SFC)

Previous Attendees:

- Addison Phillips \- Unicode (APP) - chair
- Mihai Niță \- Google (MIH)
- Tim Chevalier \- Igalia (TIM)
- Eemeli Aro \- Mozilla (EAO)
- Richard Gibson \- OpenJSF (RGN)
- Shane Carr \- Google (SFC)

Scribe: RGN

Previous Scribe: MIH

Topic: Info Share, Project Planning

Topic: PR Review

Timeboxed review of items ready for merge.

PR	Description	Recommendation
#1084	Fix contradiction in markup resolution	Discuss, Reject
#1083	Include :date, :datetime, and :time with minimal options	Discuss

#1081	Clarifications to resolved value section	Discuss
#1080	Implement the simplified pattern select mechanism	Discuss
#1078	Define time zone values and conversions	Discuss
#1077	Include :datetime, :date, and :time with style options only	Discuss, Merge
#1076	Make expErrors use an array (and only an array)	Discuss, Merge
#1068	Design document for percent formatting	Merge
#1067	Semantic skeletons design	Discuss

Topic: Clarifications to 'Resolved Values' section (#1081)

MIH: I'm not happy about the "unwrap" name, but don't have anything better.

APP: Hearing no objections, I'm merging it.

Topic: Define time zone values and conversions (#1078)

EAO: We should discuss the name of the option value that is currently "local" but means "use the timezone from the argument" (i.e., floating time).

EAO: Minimizing confusion could look like "input", or maybe "argument".

APP: One challenge is that everyone has named it something else. HTML has named it "local", so I wouldn't object to that (but I don't like it).

EAO: I don't think readers actually understand "local".

RGN: I'm trying to get a handle on this.

EAO: Consider JavaScript Temporal.ZonedDateTime. Formatting that with e.g. :date should support a timeZone option that reads from the input value. When the input doesn't have a sense of its own time zone, behavior is either an error or implementation-defined.

SFC: I have thoughts to share, but they're too detailed to communicate real-time in this call.

Topic: Make expErrors use an array (and only an array) (#1076)

EAO: I want to check against my implementation before approving.

Topic: Resolve Values and Simplified Pattern Selection (#1077 and #1083)

We have a pair of PRs implementing this.

MIH: I'm not comfortable with the direction of #1083. We've had styles for a long time and semantic skeletons on the way... inventing something in a rush does not sit well with me.

APP: This seems like an attempt to split the difference. Semantic skeletons are well enough understood and this takes us in that direction as a mix.

SFC: I support the general direction of a minimal set of options to represent "style" behavior, but this PR seems more broad than necessary.

APP: What would you leave in vs. out?

EAO: This PR uses fields like semantic skeletons to differentiate inclusion vs. exclusion of weekday. Removing that would reduce scope a lot.

SFC: I agree with EAO, dropping support for `:date style=full` would greatly reduce scope. Likewise dropping a way to configure time precision, but I don't think we're currently ready to lock in a particular shape.

EAO: We could introduce `:date`, `:time`, and `:datetime` as required with some options either absent or marked as draft.

APP: The latter seems likely to see adoption which we would later break. Can't we either identify a minimal set or just actually solve the problem?

APP: I like this in general, but not that time and date fields behave differently.

SFC: I have a thread on this, and agree that internal alignment would be better than strict adherence to UTS #35.

SFC: Date formatting has the concept of an optional era, to be displayed only outside of a particular range (e.g., Gregorian displays era before 1500 CE). It was previously `eraDisplay`, but is now lumped in with `yearStyle` along with century elision.

SFC: There's something similar in time formatting—for example, 12:58pm to 12:59pm to 1pm to 1:01pm. We think of minutes and seconds as precision for a single time-of-day field. An older draft of semantic skeletons included time fields. But a "timeFields" option could express something similar.

SFC: We could discuss it and settle things. This is very much green fields, with the only prior art being what landed in UTS #35 and ICU4X.

APP: So you would not object if we changed “precision” to...

SFC: I’m saying that design space is open.

APP: I really do want shorthands, because this will otherwise be awkwardly verbose, especially in comparison to alternatives.

SFC: I agree, and have some ideas. For example, support `dateFields=year-month-day` and `dateFields=YMD`.

EAO: We have a general approach with going to semantic skeletons, of which this is a subset.

MIH: if you’re not comfortable with this particular subset, what would you like to see?

MIH: This did not seem to me like a proper subset.

SFC: I think that the PR is an attempt to put in writing one of the approaches for how semantic skeletons could appear.

MIH: Who has the power to decide amongst all of these approaches? I’m assuming they are not mutually compatible.

SFC: CLDR did not tread into how to express the options in string form. This group is in a position to do so.

EAO: We ought not assign a higher level of stability to these than that of their base definition in UTS #35.

SFC: UTS #35 doesn’t have an API. It contains spec enumerations, but does not require that wrapping APIs use its terms. CLDR might add text describing such details, but currently does not.

EAO: IIUC, the time precision approach does not support a tabular breakdown.

SFC: Date and time fields should all be compatible with each other, but e.g. “Monday at 45 [minutes]” is nonsensical. But `formatToParts` or other APIs beyond semantic skeletons could support that kind of use case.

APP: We [MessageFormat] don’t specify `formatToParts`. We actually need to think about that superset for e.g. standalone field formatting. But a Timestamp value doesn’t have an hour field, and would need some kind of formatting to even expose that. The danger is that we reinvent picture strings.

MIH: I don't think formatToParts would solve this. Formatting individual fields can be done with skeletons today. If it's not bad i18n, we should support it.

EAO: I think we are pointing at a use case for a function that extracts formatted fields.

SFC: Semantic skeletons already support standalone fields, but not all fields—and in particular not minute or era. But there is a path to adding them.

APP: Maybe we add an advanced formatter in the future?

MIH: I don't like adding things piecemeal as people request them. Functionality baked into e.g. Android OS sticks around for years.

SFC: We are exclusively driven by use cases. There must be a demonstrable benefit for every feature to justify adding it. Limited scope that excludes a larger theoretical set of use cases supports a number of valuable attributes.

MIH: ICU is used by Windows, Mac, Android, etc., which are long-lived and have slow release cycles.

EAO: IIUC, MIH is advocating for withholding a semantic skeleton approach until the future has been identified and characterized.

MIH: I would like semantic skeletons to cover everything I can do now with classic skeletons that is not "wrong".

EAO: Is there an agreed-upon definition of "not wrong"?

MIH: No, I don't think many things are wrong other than "January at 5pm" etc.

SFC: If this group wants to fully close the gap between classic vs. semantic skeletons before adopting the latter, I think that is achievable.

Topic: Percent Formatting (#1068)

Not having achieved consensus previously, let's discuss the options or proceed to a ballot. In the previous calls we reintroduced ``style=percent``. Some delegates do not want ``:integer`` to include ``style=percent``. It's unclear if ``:unit unit=percent`` has consensus support or not. We have not yet established a clear split-vs-lump policy or criteria for functions, so ``:percent`` might still be a possibility.

Topic: Semantic Skeletons

Let's discuss next steps in completing the design.

<https://github.com/unicode-org/message-format-wg/pull/1067>

Topic: Issue review

<https://github.com/unicode-org/message-format-wg/issues>

Currently we have 26 open (was 26 last time).

- 18 are tagged for 48
- 3 are tagged "Future"
- 10 are **Preview-Feedback**
- 1 is tagged **Feedback**
- 2 are **resolve-candidate** and proposed for close.
- 2 are **Agenda+** and proposed for discussion (see below)
- 1 is **PR-Needed** and needs a pull request
- 0 are ballots

Issue	Description	Recommendation
#866	CLDR semantic datetime skeleton spec is nearly ready and MF2 should use it	Discuss
#978	Interoperability concerns and normative-optional features	Discuss