

ICU4X and TC39 Liaison Update to UTC #187

April 22, 2026

ICU4X Update

ICU4X 2.2 Release Highlights



Lightweight, portable, and secure i18n



Data Updates

CLDR 48.2 & TZDB 2026a

- Full support for Unicode data standards.
- **Japanese Calendar:** Pre-Meiji eras deprecated per CLDR-11400.
- **Hijri:** Move Reingold simulation out of default calendars



Core i18n

Date Arithmetic

New APIs for addition, subtraction, and duration. Shipping in Chrome Temporal.

Typed Months

- Type-safe Month enum replaces codes.
- Optimized `until()` math by up to 75%.



UTC Relevant

Property Support

Numeric_Type, Joining_Group, and Indic_Conjunct_Break.

Experimental

- **ML Segmentation:** RAdaBoost (CN) and CNN (TH).
- **Kotlin Bindings:** First-class FFI support added.

UCD Flag Refactoring

Standardizing data sourcing and versioning across ICU4X components

Current Flag Matrix

`--ucd-root`

Points to a local directory that has the same layout as unicode.org/Public. For example, it should contain a [security](#) subdirectory.

`--ucd-tag`

Specifies the release tag (e.g., **17.0.0**) for automated fetching from unicode.org/Public/.

`--unihan-root`

Defines the location of the [Unihan.zip](#) file for Han character processing.

PR #7882 Proposal & Feedback

`--unihan-root` Deprecation

This is a proposal by **Robert Bastian**. We are seeking feedback on the best way to reference UCD data files.

Unihan data reference pattern would become:

```
{ucd-root}/ucd/Unihan.zip/Unihan_IRGSources.txt
```

Active Work Streams

Potential changes in upcoming ICU4X releases

☰ Unicode Line Segmenter

PR #7898: Updating [UAX #14](#) implementation to the latest Unicode 17.0 standards.

This PR contains an implementation of work by Robin Leroy, implemented by Robert Bastian. We are exploring this approach.

- Standardizing line break logic for novel Unicode 17 behaviors.
- Ensuring consistency between C++ and Rust implementations.

⬇ Direct UCD Data Loading

PR #7904: Early-stage work by Robert Bastian to move away from intermediate [icuexportdata](#) to direct UCD ingestion.

- Reduces dependency on ICU4C's internal data pipelines.
- Enables ICU4X to use earlier-stage UCD data.

TC39 Update

TC39-TG2 Proposal Activity

Progress summary from Jan–Apr 2026 meetings

STAGE 4 Ready for Ship

Intl Locale Info: Shipped in all major engines

Temporal: Shipped in Chrome, Firefox, and other engines

Intl Era and MonthCode: Shipped in Chrome and Firefox

STAGE 2.7 Refinement

Intl Keep Trailing Zeros: Advanced to Stage 2.7; bugfix for internal formatting behavior.

Active Development & Exploration

STAGE 2

Intl Unit Protocol: Got Stage 2 to establish a protocol for value-unit pairs.

STAGE 1

STAGE 0

String toTitleCase: Re-triaged in Apr; focus on titlecaseSegment and community use cases.

Amount: Status at Stage 1 with focus on "number with precision" model.

Intl Energy Units: Updated Stage 1 with focus on utility-based use cases.

String toTitleCase Debate

Analysis of April 2026 TC39-TG2 meeting discussions

STAGE 0

Building Blocks vs. End-to-End Solutions

Proposal Focus: titlecaseSegment

The debate centered on whether JS should provide a high-level "one-step" solution or low-level primitives.

Key Points of Contention:

- **Naming Confusion:** Delegates felt the name "TitleCase" is misleading as it doesn't solve for complex rules like stopwords. Should it be called "SentenceCase"? Or "TitleCaseSegment"?
- **CSS Overlap:** Mozilla delegates argued that text transformation often belongs in CSS (declarative) rather than JS (imperative).
- **Dutch and Spanish Rules:** Debate touched on linguistic complexities like Dutch "IJ" and Spanish leading punctuation.