ICU4X 2.0: Next Level i18n

Shane Carr, Unicode Technology Workshop 2024

Intro to ICU4X

INTERNATIONALIZATION

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

I18N

Examples of i18n Operations

Today's date in various locales and calendars:

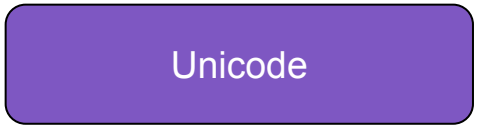
- en: Oct 23, 2024 (Gregorian)
- de-CH: 23. Okt. 2024 (Gregorian)
- he: 21 בתשרי 5785 (Hebrew)
- bn: ২১ রবিউস সানি, ১৪৪৬ যুগ (Islamic)
- zh: 2024年九月21 (Chinese)
- ja: 令和6年10月23日 (Japanese)

Breakpoints in Japanese text (words may be multiple characters wide):

中|ワ|況|写|イノナ|開|億|が|や|へ|者|43|俳|寺|式|7|沢|暮|ル|材|年|る|あん|移|酔|むぎ|え|す|写|情|主|逃|69|引|ぎ|う|ぱ|何|81|昇|フネイマ|本|因|ぱ|不|真|え|断|候|ら|。

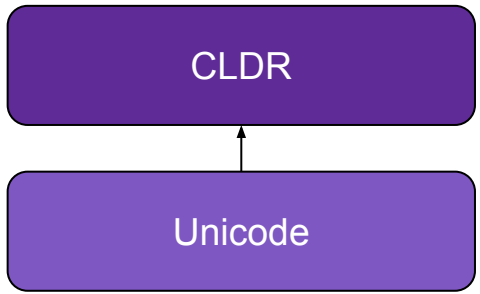
The i18n Stack

Unicode:
Foundational algorithms, specification, character set



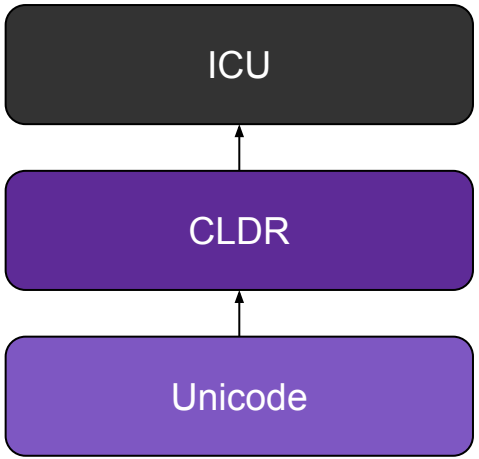
The i18n Stack

CLDR:
Data for hundreds of locales, specs for MessageFormat, person names, keyboards, …



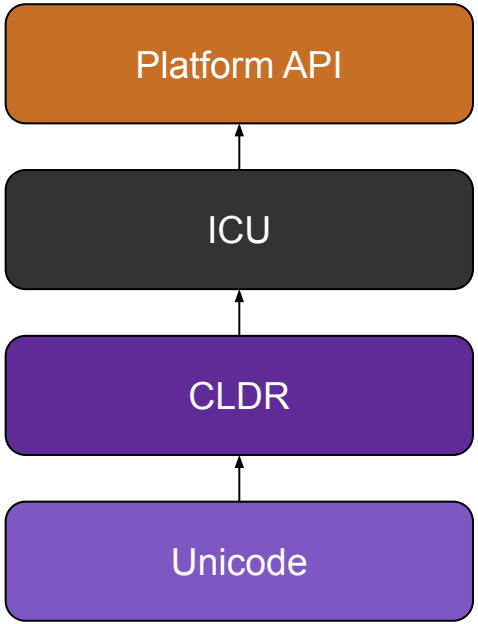
The i18n Stack

ICU:
Code that implements the algorithms in Unicode with the data in CLDR

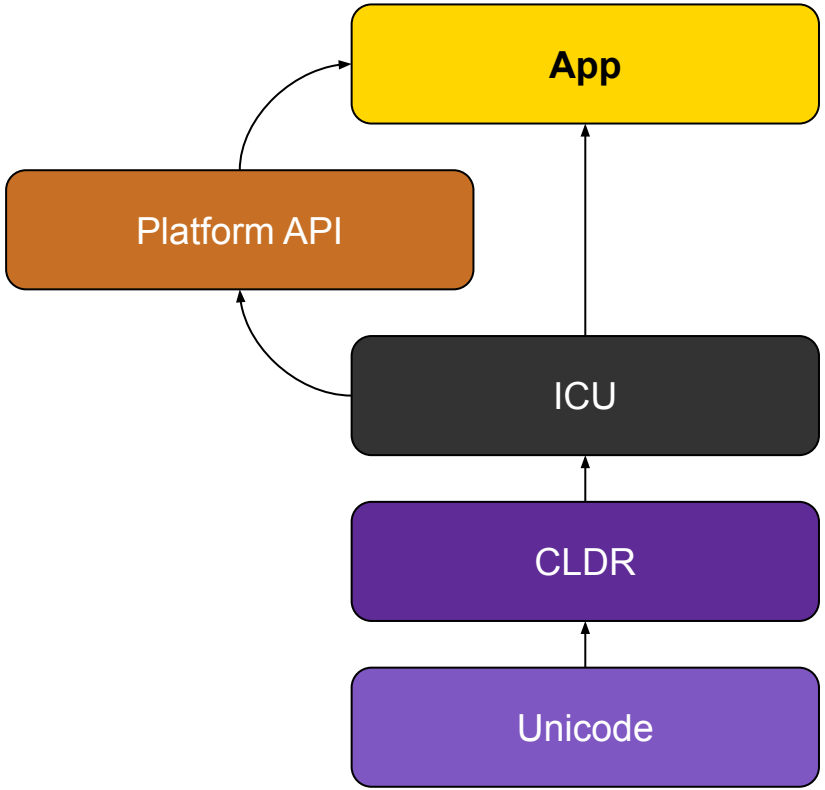


The i18n Stack

Platform API:
Making i18n easy to use from applications
(example: ECMA-402, android.icu)

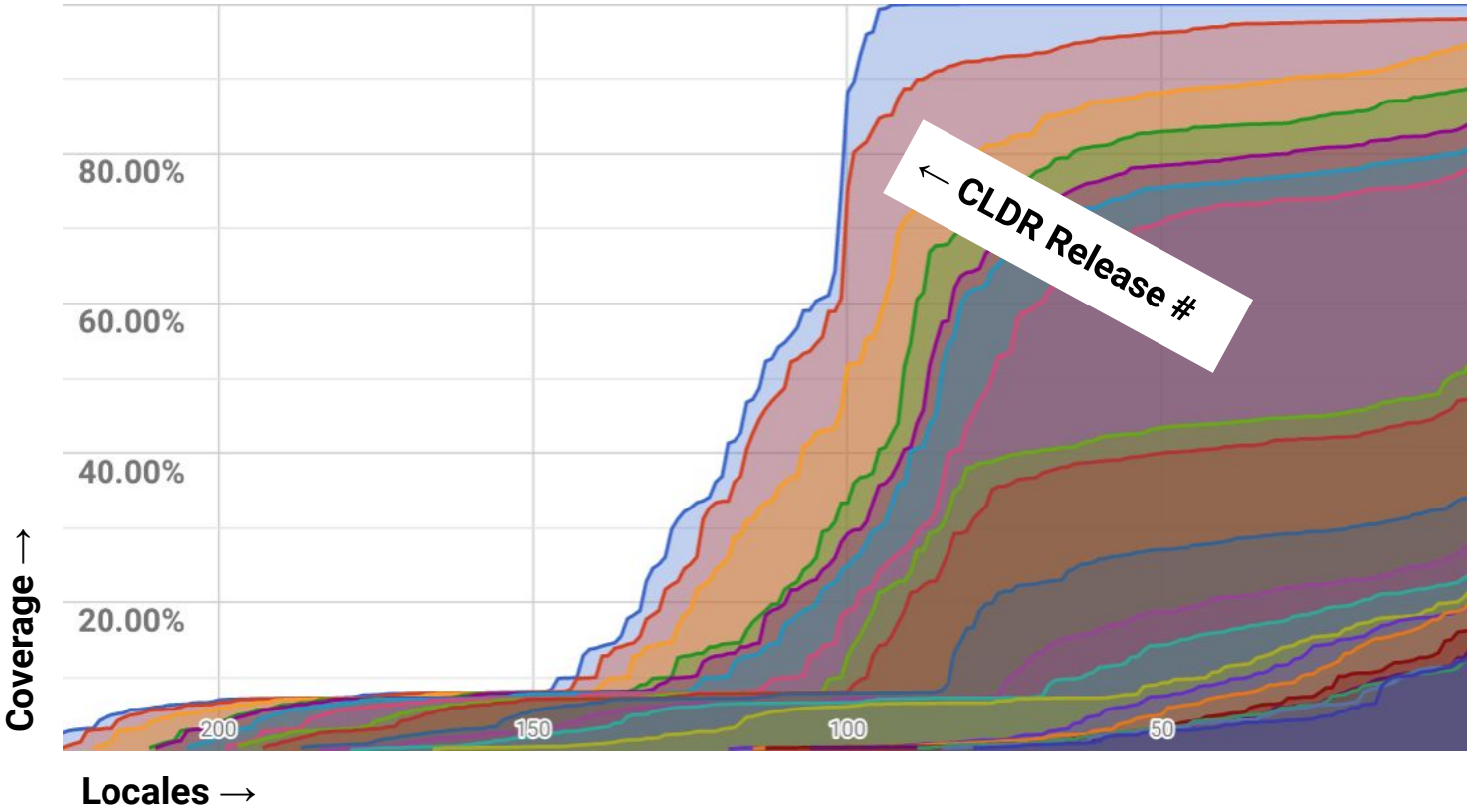


The i18n Stack

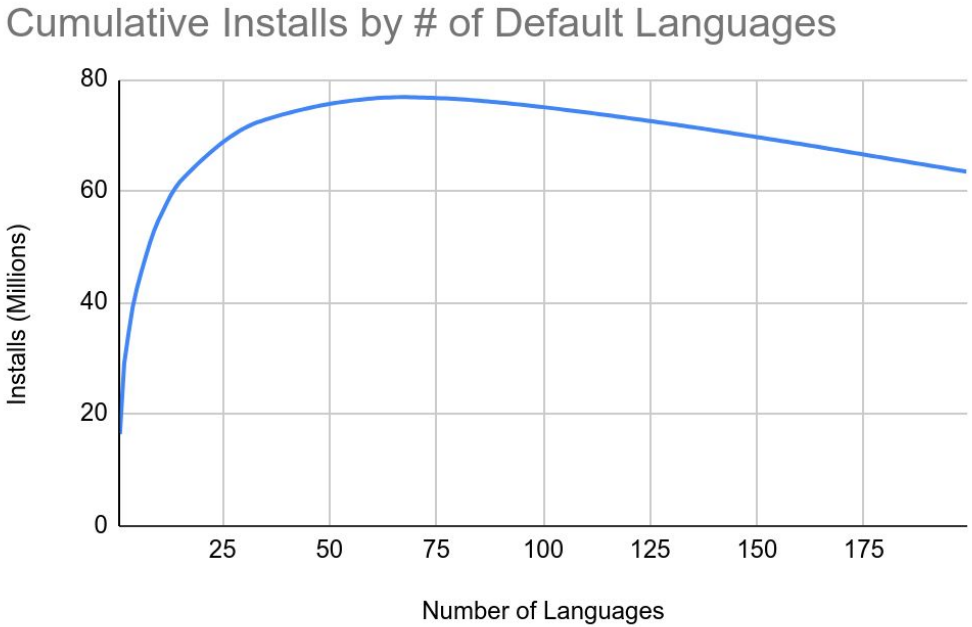


Challenges with Scaling i18n

Quadratic Growth of Data



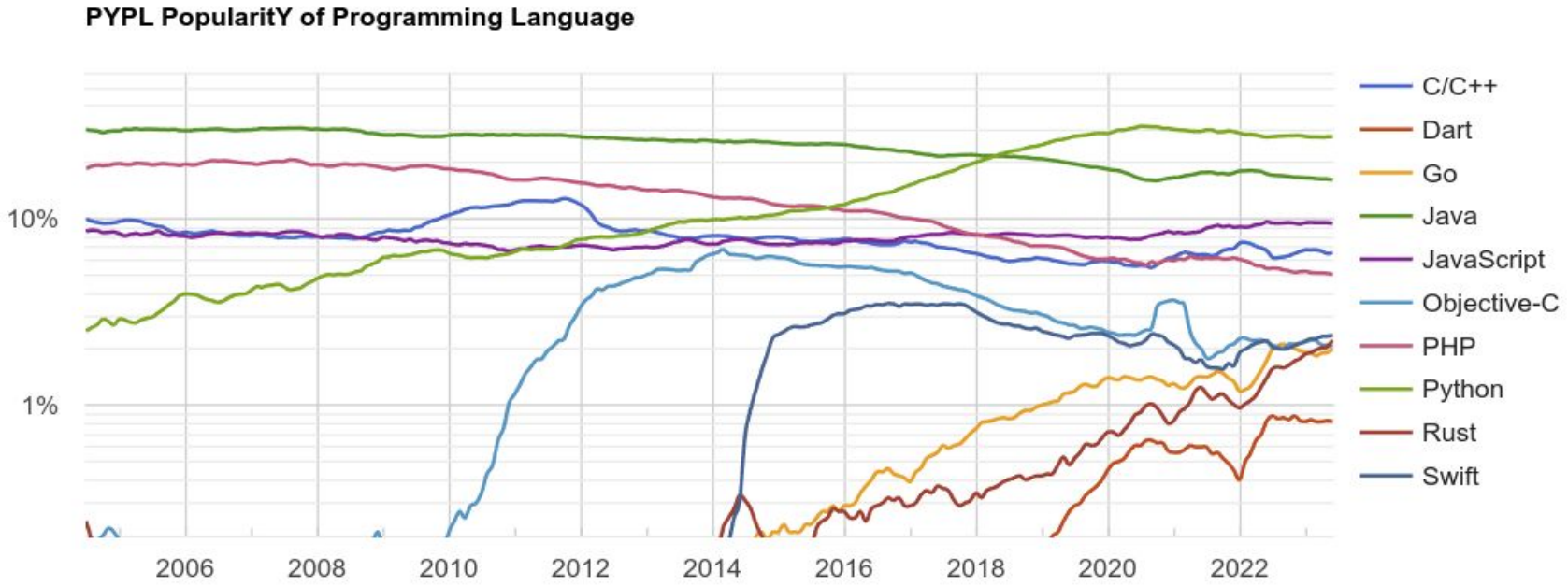
Adding more default languages has diminishing returns



If adding a language increases app size by 100 kB, reducing total downloads by 0.25%

Devices have different requirements

Hot new programming languages every year



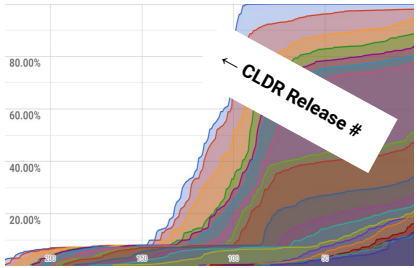
i18n-as-a-service doesn't make sense

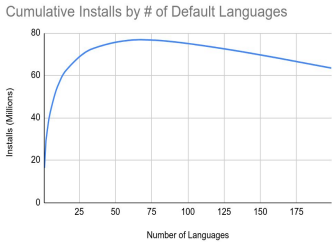
- Low latency requirements
- Data-heavy algorithms
- Privacy implications

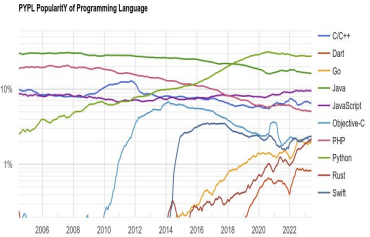
中|ワ|況|写|イノナ|開|億|が|ゃ|へ|者|43|俳|寺|式|7|沢|暮|ル|材|年|る|あん|移|酔|むぎ|え|す|写|情|主|逃|69|引|ぎ|う|ぱ|何|81|昇|フネイマ|本|因|ぱ|不|真|え|断|候|ら|。|吉|群|コクチ|賀|伝|ツエ|別|写|ユリキ|施|見|ら|カ|1|取|目|れ|づ|ぱ|ぞ|然|野|テ|倍|展|ぴ|る|て|け|京|答|週|ネコ|入|整|料|勉|ら|げ|。

Wish List for a Scalable i18n Library

- Pay for what you use and not what you don't use
- Add extra locales on demand
- Run on all types of devices
- Work in both today's and tomorrow's programming languages
- Run everything on-device







中|ワ|況|写|イノナ|開|億|が|や|へ|者|43|俳|寺|式|7|沢|暮|ル|材|年|る|あん|移|酔|むぎ|え|す|写|情|主|逃|69|引|ぎ|う|ば|何|81|昇|フネイマ|本|因|ば|不|真|え|断|候|ら|。|吉|群|コクチ|賀|伝|ツエ|別|写|ユリキ|施|見|ら|カ|1|取|目|れ|づ|ば|ぞ|然|野|テ|倍|展|ぴ|る|て|け|京|答|週|ネコ|入|整|料|勉|ら|げ|。

Enter ICU4X

ICU4X is an internationalization library that is:

1. Portable by design
    - Runs on all types of devices with supported wrappers for multiple programming languages
2. Lightweight
    - Modular design, great for compile-time dead-code elimination and data slicing
3. Secure
    - Written in Rust, a memory-safe language

ICU4X-TC

ICU4X, CLDR, and ICU

- ICU4C, ICU4J, ICU4X: Same data (CLDR), same spec (UTS 35), different code, similar behavior
- See the conformance dashboard for details on feature coverage and behavior differences

ICU Data Driven Test Summary

Read more about the Unicode Conformance project.

Tests and platforms

Report generated: 2024-10-10 23:59:22

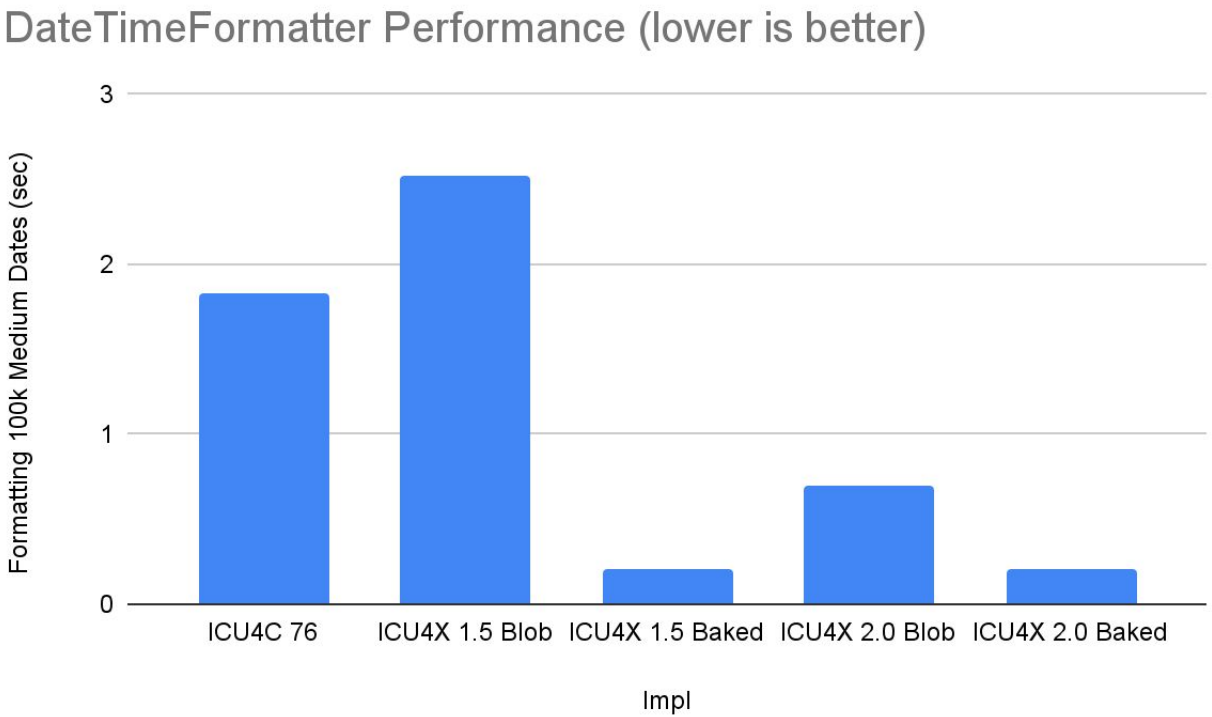
Executors verified: cpp_71.1, cpp_72.1, cpp_73.1, cpp_74.2, cpp_75.1, dart_web_v20.1.0, icu4j_73.2, icu4j_74.2, icu4j_75.1, node_v14.18.3, node_v14.21.3, node_v18.7.0, node_v18.14.2, node_v20.1.0, node_v21.6.0, node_v22.9.0, rust_1.3.2, rust_1.4.0
Tests verified: collation_short, number_fmt, plural_rules, datetime_fmt, list_fmt, lang_names, rdt_fmt, likely_subtags, message_fmt2

Summary of all tests

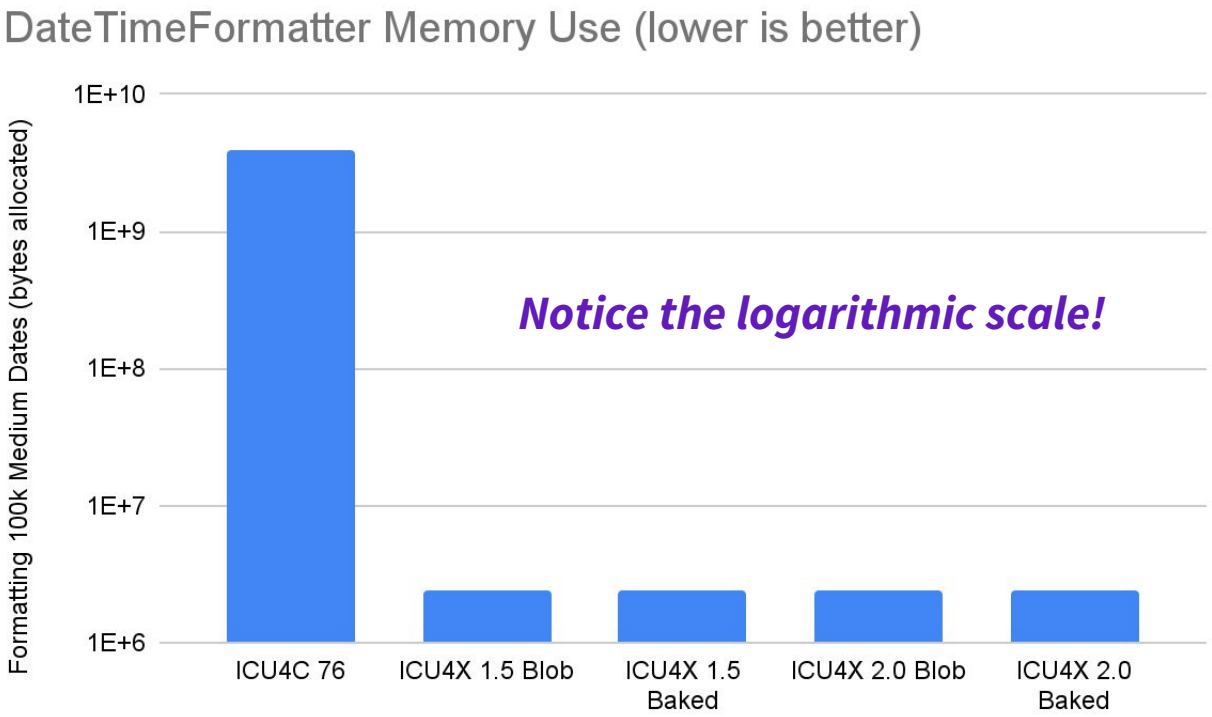
Test Type	Dart Web	Details	ICU4C	Details	ICU4J	Details	ICU4X
collation_short	v20.1.0 73.1	icu73	71.1 71.1 72.1 72.1 73.1 73.1 74.2 74.2 75.1 75.1	icu71 icu72 icu73 icu74 icu75	73.2 73 74.2 74 75.1 75	icu73 icu74 icu75	1.3.2 icu/tr/2023-05-02/73.x 1.4.0 release-74.1
datetime_fmt			72.1 72.1 73.1 73.1 74.2 74.2 75.1 75.1	icu72 icu73 icu74 icu75	73.2 73 74.2 74 75.1 75	icu73 icu74 icu75	1.3.2 icu/tr/2023-05-02/73.x 1.4.0 release-74.1
lang_names	v20.1.0 73.1	icu73	73.1 73.1 74.2 74.2 75.1 75.1	icu73 icu74 icu75	73.2 73 74.2 74 75.1 75	icu73 icu74 icu75	1.3.2 icu/tr/2023-05-02/73.x 1.4.0 release-74.1
likely_subtags			74.2 74.2	icu74 icu75	74.2 74	icu74 icu75	1.4.0 release-74.1

Benchmarks

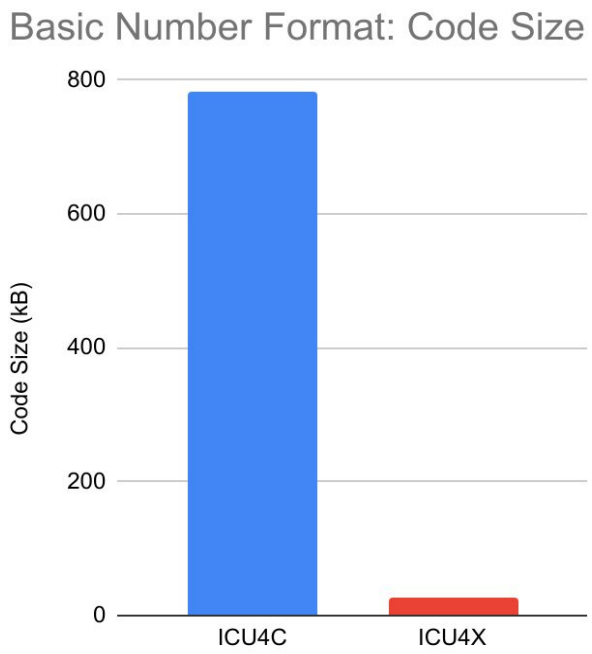
ICU4C 76 vs ICU4X 1.5 vs ICU4X 2.0: Performance



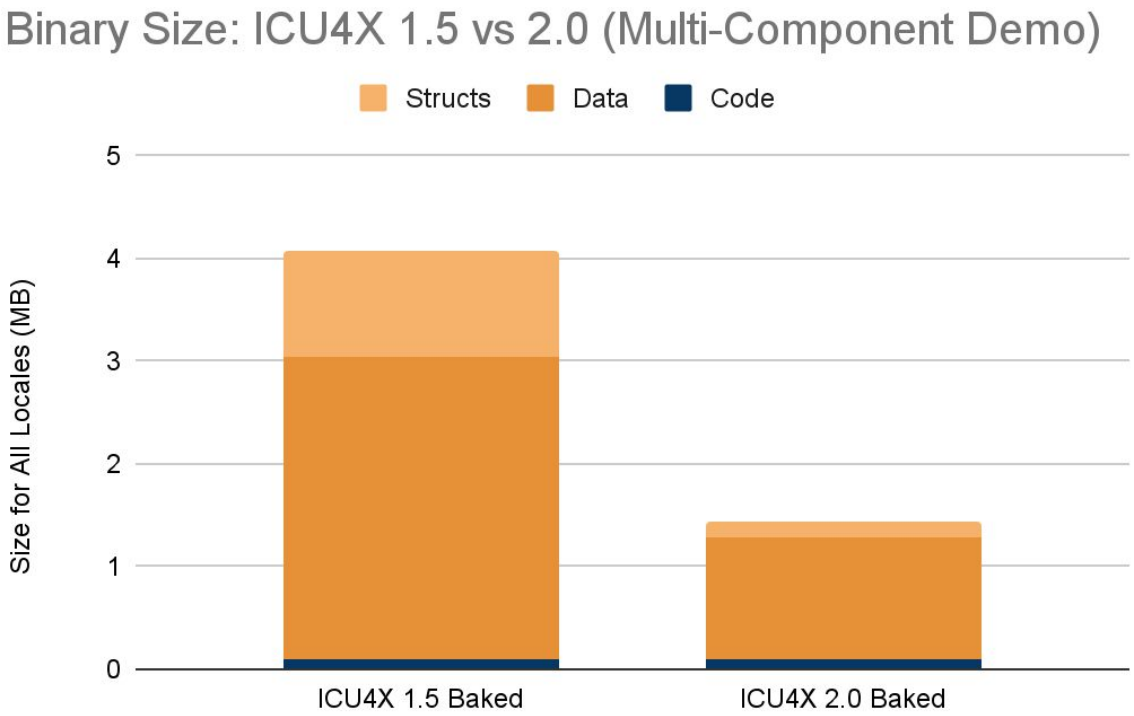
ICU4C 76 vs ICU4X 1.5 vs ICU4X 2.0: Memory Use



ICU4C vs ICU4X: Binary size (code, no data)



ICU4X 1.5 vs ICU4X 2.0: Binary size



What other metrics would you like to see?

New in ICU4X 2.0: Semantic Date and Time

Why Semantic Skeletons?

Correctness

Smaller and Faster

Field Sets: Restrict to only valid combinations!

Date

Day
Weekday
Day + Weekday
Month + Day
Month + Day + Weekday
Year + Month + Day
Year + Month + Day + Weekday

Calendar Period

Month
Year
Year + Month

Composite

Date + Time
Date + Time Zone
Date + Time + Time Zone
Time + Time Zone

Time

Hour
Hour + Minute
Hour + Minute + Second

Time Zone

This is a comprehensive list of CLDR 46 field sets!

Options

Option Name	Values	Fields
Length	Long, Medium, Short	Year, Month, Day, Weekday, Hour, Zone
Alignment	Inline, Column	Year, Month, Day, Hour
Year Style	Auto, Full, With Era	Year
Hour Cycle	Auto, H11, H12, …	Hour
Fractional Second Digits	Auto, 0, 1, …, 9	Second
Time Zone Style	Specific, Generic, …	Zone

Live Demo! (Of ICU4X Docs)

Note: This is 2.0 Alpha, and it could still change. Please send your feedback!

https://unicode-org.github.io/icu4x/rustdoc/icu/datetime/fieldset/index.html

Technical Note: How This Achieves Smaller Data

Your formatter is *specific to your field set*! In many cases, hidden by Rust type inference.

- `DateTimeFormatter<YMD>` **~=** `YearMonthDayFormatter`
- `DateTimeFormatter<HMZ>` **~=** `HourMinuteSpecificTimeZoneFormatter`

The smaller types enable compile-time dead code elimination.

Also! The smaller data payloads are more easily deduplicated.

More About ICU4X 2.0

A deep dive to give appreciation to everything that makes us achieve our goals

Top 8 Contributors in ICU4X 1.4, 1.5, and 2.0

Robert Bastian
396 Commits

Shane Carr
319 Commits

Younies Mahmoud
117 Commits

μ

Manish Goregaokar
108 Commits

José Julián Espina
17 Commits

Kartavya
17 Commits

Makoto Kato
14 Commits

Henri Sivonen
11 Commits

And so many more!

- 10 Zibi Braniecki
- 8 Bruce Mitchener
- 8 Kevin Ness
- 8 Tyler K
- 5 Robin Leroy
- 4 Ashutosh Jha
- 4 Kevin
- 4 Steven R. Loomis
- 3 Blayne Marjama
- 3 Colin Rofls
- 3 Kartavya Vashishtha
- 3 Sean Burke
- 2 Adam Chalmers
- 2 Ben Beasley
- 2 CanadaHonk
- 2 Elango Cheran
- 2 Linus Groh

New Features: A few highlights beyond datetime

- CLDR 46
- New experimental components: Duration Formatter, Unit Formatter, Currency Formatter, Person Names, Unit Converter
- Graduated casemap component to stable
- All new Diplomat WASM demo
- Add LocaleData parameter for word/sentence segmenter
- Support Unicode 15.1 for line segmenter
- Add PluralElements for algorithmic plural selection
- IXDTF Parsing: Date, Time, DateTime, ZonedDateTime try_from_str
- Time zone end-to-end display names without TZDB

Infrastructure Upgrades: A few highlights

- Revamped data request architecture with locale and attributes
- All string input functions robustly handle ill-formed UTF-8
- Improvements to ICU4X datagen APIs and options (e.g. locale families)
- DryDataProvider for dry-run loading of data (e.g. supported locale queries)
- Add script fallback priority; improve language/region fallback priority
- Introduce borrowed variants of normalizer and collator structs
- Underlying zerovec data format optimized to pack variable-length items better
- Improvements in non-Gregorian calendrical calculations
- Use ZeroTrie in more places, reducing size and improving efficiency
- Fine-grained error enums

FFI Improvements in ICU4X 2.0

- Dart now supported, on top of C, C++, JS, and TypeScript
- More usage of idiomatic features in languages that provide them, like getters, setters, constructors, and stringifiers
- APIs follow more idiomatic per-language naming conventions
- Namespacing in C++

Status of ICU4X 2.0

- Done: All major feature work. Today is "alpha" quality.
- To-do before "beta": full API re-review; migration guide
- To-do before "final": FFI, finish time zone API, small function renames
- Stretch: type-safe locales, more fallback/datagen, nice-to-haves

ICU4X-TC ships when things are ready, not necessarily on a regular cadence like ICU-TC. (Hint: if you want something shipped sooner, become a contributor!)

Truly exceptional engineering for 4+ years has brought ICU4X from a concept to reality.

Plans for 2025

1. Compatibility with ICU4C/ICU4J for easier partial migration
2. Reduce duplication of work between ICU-TC and ICU4X-TC
3. ECMA-402 100% coverage, tested
4. Super-well polished for 3P adoption

What do *you* want to see in this list?

Conclusion and Call to Action

Considering ICU4X? Let's chat! Help us improve the offering!

We love mentoring! Lots of projects both big and small!

Learn more on our GitHub: github.com/unicode-org/icu4x

Docs: icu4x.unicode.org

Version 2.0 Shipping Soon!

Subscribe to our low traffic mailing list! QR Code:

My contact info:

shane@unicode.org
linkedin.com/in/shanecarr
Twitter: @_sffc
Mastodon, QR Code: