Recent Status of the TEX Community in Japan

Takuto Asakura

BachoTEX 2025

Today's Topics

Part I: Key Differences in Japanese and English Typesetting

— Providing the background that shapes the Japanese TEX community

Part II: (LA)TEX Systems for Japanese Typesetting

— A snapshot of today's TEX ecosystem in Japan

Part III: Recent Trends in Development and Communication

— Current challenges facing us and how we are addressing them

Part I:
Key Differences in Japanese and English Typesetting

Key Differences in Japanese and English Typesetting

1 Character-Set & Space Model Divergence

- Use variable-width letters plus inter-word spaces
- Rely on *fixed-width glyphs* and virtually *no word spaces*

👉 Line-breaking logic starts from fundamentally different premises

2 Strict Line-Breaking Prohibitions

Large sets of *leading-prohibited* (e.g., closing punctuation) and *trailing-prohibited characters* (e.g., opening brackets)

3 Mixed-Script Composition Is the Norm

- Numerals, English words, formulae, and symbols appear *routinely* inside Japanese prose
- Automatic inter-script spacing (e.g., `\xkanjiskip`), kerning, and baseline alignment are required at every boundary

1 Character-Set & Space Model Divergence

Fixed-width (full-width) vs. variable width

- Builds rhythm with variable-width glyphs + inter-word spaces
- Kanji and kana occupy *a nominal 1-em square grid*

Lack of inter-word spaces — different line-breaking logic

- Candidates occur mainly at space positions and hyphenation points
- Line-break candidates = virtually *every* character boundary

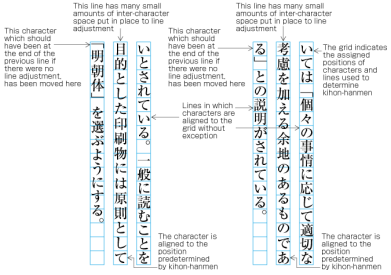
2 Strict Line-Breaking Prohibitions

The rule set of “where you may *not* line break”

- No inter-word spaces = all character boundary is a candidate break
  👉 Prohibitions exclude a large fraction
- Readability hinges on the positions of punctuation marks
  —commas, periods, brackets, repeated dashes, etc.

Core categories: “no break before” vs. “no break after”

- Forbidden at line *start*
  period, comma, closing bracket, etc.
- Forbidden at line *end*
  opening bracket, long dash, etc.



3 Mixed-Script Composition Is the Norm

Latin letters, numerals, math formulae, and code chunks appear *frequently* in Japanese texts (at least in scientific documents)

👉 Designed on the premise that **several scripts share a single line**:

- Automatic insertion of appropriate spacing at script boundaries
- Handling conflicts in line-breaking logic
  i.e., Western hyphenation and Japanese line-breaking prohibition tables are evaluated in parallel
- Baseline alignment & glyph-size harmonization

Without Kanji-Latin Space (`\xkanjiskip` $= 0\,\mathrm{em}$)

この文書はLaTeXとdvipdfmxで作成された。

With Kanji-Latin Space (`\xkanjiskip` $= 0.25\,\mathrm{em}$)

この文書は LaTeX と dvipdfmx で作成された。

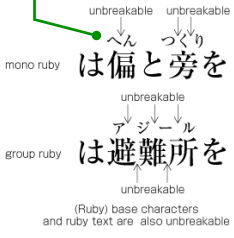
Other Features Unique to Japanese Typesetting

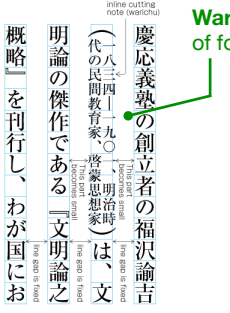
Vertical writing (top-to-bottom, right-to-left)

In recent years, horizontal text has become common in Japanese, but vertical writing is still used in newspaper, literature, etc.

Multi-layer inline layout (variants of emphasis)

Ruby: annotative glosses to show pronunciation

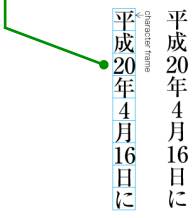




Warichu: Japanese variation of footnote or sidenote

Kenten: emphasis dot

Tate-chu-yoko: 2–4-digit numbers or Latin strings set sideways



Part II: (LA)TEX Systems for Japanese Typesetting

Current (LA)TEX Workflows for Japanese Typesetting

Two practical approaches are in everyday use

- **pTEX variants**: pLATEX or upLATEX + **dvipdfmx**
  - A *Japanese-specific* extension maintained since the 1990s
  - Support for JFM, native line-break prohibitions and vertical-typesetting primitives, etc.
- **LuaTEX-ja**: LuaLATEX + luatexja package
  - Implements the Japanese layer in Lua on top of the LuaTEX engine
  - Meets Japanese requirements while keeping feature parity with standard Western LATEX

pTEX variants and LuaTEX-ja

Engine	Speed	Stability	Sustainability	Users
pTEX	**Fast**	Very high	**Great concern**	General / Passively conservative
LuaTEX	**Slow**	Moderate	**Fine**	Somewhat advanced

pTEX — Core Extensions for Japanese Typesetting

Support for JFM (Japanese Font Metrics)

- Per-character-class settings for width, side bearings, kerns, etc.
- Auto-inserted glue: `\kanjiskip` (between Japanese chars), `\xkanjiskip`

Line-break prohibitions built into the core

- Line-head / line-end prohibition is integrated inside the engine core
- During break search, `\prebreakpenalty` and `\postbreakpenalty` are assigned to eliminate illegal candidates

Native support for vertical writing

Direction switches with `\yoko` (horizontal) and `\tate` (vertical); the engine rotates the glyph coordinate system accordingly

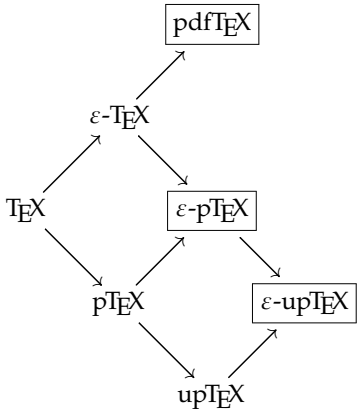
💡 **LuaTEX-ja handles the same issues via flexible Lua callbacks**

The pTEX Legacy (1) pTEX vs. upTEX

pTEX 8-bit JIS encoding; kanji are stored as two bytes, making user-defined characters hard to extend

upTEX Internal Unicode encoding; so that users can use characters outside of the 8-bit JIS encoding

It covers all BMP (Basic Multilingual Plane) CJK characters plus IVS (Ideographic Variation Sequence) support and more



💡 From TL 2024 onward, an unified binary
→ `ptex` is now upTEX's pTEX-compatible mode

The pTEX Legacy (2) Encodings and \kcatcode

Input vs. internal encoding

Input chosen by auto-detection or by `-kanji` option ((s)jis, euc, utf8)

Internal pTEX: fixed 8-bit JIS code
upTEX: Unicode

👉 Thus pTEX converts UTF-8 **before** tokenization

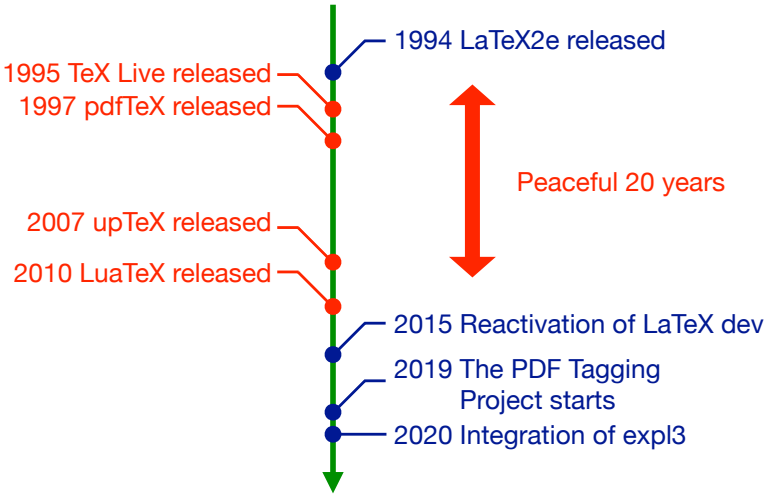
\kcatcode—Japanese-specific category code

- Every multi-byte Japanese character receives a \kcatcode
  E.g., 15 = Non-CJK, 16 = Kanji, 17 = Kana, 18 = Others
- Used *both* for lexical scanning (e.g., judging control sequences) and for line-break/prohibition class look-ups
- **The specification differ noticeably between pTEX and upTEX**

❗ See “Guide to pTEX for developers unfamiliar with Japanese” (ptex-guide-en.pdf)

Part III:
Recent Trends in Development and Communication

Brief History of LATEX 2ε and Some TEX Engines



pLATEX Predicament — A Mountain of Technical Debt

A giant patch-work

Decades of incremental extensions mean every upstream LATEX change triggers a new cascade of local fixes

Rapid LATEX-kernel evolution (since 2020)

- Migration from NFSS 2 to the *new* NFSS (2020)
- Introduction of the new hook system and the plug/socket mechanism (2020–)

👉 Japan-specific extensions now need frequent “catch-up patch” cycle

Severely limited manpower

All Japanese engines and formats are effectively maintained by only **one or two** active developers each

Root Cause: pTEX's Incompatible Token Model

(u)pTEX = "8-bit engine + special Japanese tokens"

- Western characters live in the 0–255 range, while Japanese characters are stored as code points ≥ 256
- Japanese tokens have no `\catcode`; hence *active characters* and similar mechanisms cannot be assigned to them

Examples of practical breakage

- expl3 string modules (l3regex, l3str-convert, etc.) misbehave or fail
- `\detokenize` that mixes 8-bit chars and Japanese tokens yields illegal token lists

A radical fix would require rewriting both the specification and the C core implementation of pTEX—realistically infeasible

It Is Not Just an Engine- or Kernel-Level Issue

Relatively manageable items

- Generic Japanese document classes (e.g., jarticle, jsarticle, jlreq)
- Japan-specific LATEX packages

Hard-to-maintain items

- Patch collections that adapt *foreign* packages cf. plautopatch
- Journal / society templates and other publisher-specific macros

👉 Long tail of legacies makes a clean break from pTEX **extremely hard**

- doc (latex)
  → pldocverb (platex-tools)
- tracefnt (latex)
  → ptrace/uptrace (platex/uplatex)
- fltrace (latex)
  → pfltrace (platex)
- array (latex-tools)
  → plarray (platex-tools)
- array (latex-tools) + plext (platex)
  → plextarray (platex-tools)
- delarray (latex-tools) + plext (platex)
  → plextdelarray (platex-tools)
- colortbl + plext (platex)
  → plextcolortbl (platex-tools)
- arydshln
  → plarydshln
- arydshln + plext (platex)
  → plextarydshln
- siunitx
  → plsiunitx
- collcell
  → plcollcell
- everysel (ms)
  → pxeverysel (platex-tools)
- everyshi (ms)
  → pxeveryshi (platex-tools)
- atbegshi (oberdiek)
  → pxatbegshi (platex-tools)
- ftnright (latex-tools)
  → pxftnright (platex-tools)
- multicol (latex-tools)
  → pxmulticol (platex-tools)
- xspace (latex-tools)
  → pxxspace (platex-tools)
- textpos
  → pxtextpos (gentombow)
- eso-pic
  → pxesopic (gentombow)
- pdfpages
  → pxpdfpages (gentombow)
- stfloats (sttools)
  → pxstfloats (pxsttools)
- hyperref
  → pxjahyper
- pgfrcs (pgf)
  → pxpgfrcs
- pgfcore (pgf)
  → pxpgfmark

Community Status & Possible Paths Forward

Communication channels

- Mostly text-based: Japanese TEX Users Slack and GitHub Issues
- TEXConf: Annual Japanese TEX Users Conference

npTEX concept — a XƎTEX-based, legacy-free successor

- Specification still undefined, prototyping status for a few years
- *LATEX News 40* (Nov 2024) announced **an end to full XƎTEX support**

Where I fit in

- Not an engine hacker; my focus is *documentation and tooling*
- My efforts to stay engaged with the global TEX community include Texdoc development and translating LearnLaTeX.org
- Talks like this aim to *bridge the information gap with the global TEX community*

Summary

Japanese vs. Western typesetting: what really differs

Zero inter-word spaces, script mixing, and strict break-prohibition rules demand their own line-break logic, JFM glue, and vertical-writing support

Two practical Japanese workflows, two philosophies

pTEX variants = fast, engine-level C extensions; LuaTEX-ja = flexible Lua callbacks on a modern UTF-8 core — each with clear strengths and growing maintenance costs

The road ahead: limited hands, rising kernel changes

One-to-two active developers per engine, legacy token issues, and the still-vague npTeX idea mean collaboration and fresh contributors are urgently needed

References

- Takuto Asakura, The BXghost Package, Version 0.5.1 (2023).
- Japanese TEX Development Community (texjporg). pTEX Manual, Version p4.1.1 (June 2024).
- Japanese TEX Development Community (texjporg). Guide to pTEX for developers unfamiliar with Japanese, Version p4.1.1 (June, 2024).
- Hironori Kitagawa. Japanese Typesetting with LuaTEX. EuroTEX 2012. https://github.com/h-kitagawa/presentations/blob/HEAD/eurotex12.pdf.
- Hironori Kitagawa. About "XETEX-based npTEX" (in Japanese). https://qiita.com/h-kitagawa/items/ebbf3684b3b1be8f0747 (2023) (accessed on 2025-05-01).
- The LATEX Team. LATEX News Issue 40 (November 2024).
- The LuaTEX-ja Project Team. The LuaTEX-ja Package, Version 20250401.0 (April 2025).
- W3C. Requirements for Japanese Text Layout (JLReq). https://www.w3.org/TR/jlreq/ (2020) (accessed on 2025-05-01).
- Hironobu Yamashita, pLATEX may be seriously in danger (in Japanese). https://acetaminophen.hatenablog.com/entry/2021/06/18/022108 (2021) (accessed on 2025-05-01).
- Hironobu Yamashita, Package plautopatch, Version 0.9q (2021).