Overview of TEX engines and formats

Karl Berry, Max Chernoff, Jim Hefferon

Abstract

Among TEX and friends there are lots of moving parts. It can be hard to keep them straight. This brief article gives an overview of current major pieces of TEX software. Previous work goes into more detail [2, 11].

1 What is an engine?

The majority of users today run their documents through the LATEX "format" on the pdfTEX "engine". A format is a collection of TEX code (often called macros), and your document, using that format, is executed by an engine, which is an executable binary program.

The most common engines today are the pdfTEX engine, the LuaTEX engine, the XETEX engine, and (particularly for Japanese users) the upTEX engine. We will say more about them below, in order of appearance.[1]

2 What is a format?

A format is nothing more than a collection of TEX macros, albeit usually quite large. To reduce startup time, these macros are always preloaded into a binary `.fmt` file, though this isn't strictly required.

Formats and engines are generally independent, so you can use most format with most engines, but there are some exceptions.

The vast majority of users use the LATEX format; other formats have quite committed users (including some of the authors of this article), but this is rare in comparison.

After LATEX, other common formats include `plain` TEX, ConTEXt, and OpTEX, which we'll also discuss below.

3 Background

TEX is extraordinarily long-lived software. Donald E. Knuth released the original version of TEX in 1978,[2] but all modern versions are descended from an updated version released in 1982. After more than four decades, it is not surprising that the software has evolved, including the engines that underpin it all.

Since the early 1980s, computers have gained orders of magnitude more memory and speed. And back then there was no World Wide Web, ubiquitous color displays, and many more things that we now take for granted. Newer TEX engines leverage all of those features. The best-practice tradeoffs in the 1980s and 90s among user convenience, memory usage, disk space, and running time, were quite different than those tradeoffs today.

4 Engines (binary executables)

4.1 The TEX82 engine

TEX82 is the basis for all modern TEX engines. Its proper name is just "TEX", but since "TEX" is commonly used to refer to the entire family of programs, here we will refer to the original engine as "TEX82". It is invoked by the command name `tex`, on the command line or in an editor.

Despite its age [4],[3] any document written for TEX82 should produce identical output when compiled using any modern TEX engine (though few documents nowadays restrict themselves to this). Most of the features that we associate with TEX were created for this engine; the modern engines add comparatively few, though significant, features.

4.2 The ε-TEX engine

ε-TEX[4] was the first major TEX extension, first released in 1995 [1]. It adds some low-level programming primitives[5] to TEX82 that all non-`plain` formats depend on. In the past, there were separate ε-TEX variants of most TEX engines,[6] but today all engines except for `tex` include ε-TEX extensions by default. These days, the `etex` command is an incarnation of pdfTEX running in "ε-TEX mode".

4.3 The (e)(u)pTEX engine(s)

pTEX is a Japanese TEX engine. An initial version was released in 1990 [29], although all modern

[1] The footnotes in this article contain trivia about obscure engines/formats that are only of historical interest, included for reference. They can be safely ignored by readers who just want current information.

[2] TEX78 was written in a language called SAIL, invented at Stanford. TEX78 closely resembles modern TEX in both syntax and behavior [16], but documents written for TEX78 are not compatible with TEX82 or any other current engine. See `tug.org/whatis` for a brief history of TEX.

[3] In 1989, Knuth released an updated version of TEX82 known as "TEX 3.0" (occasionally referred to as "TEX90"), which added a few new primitives and the ability to accept eight-bit characters as input [18]. Since then, Knuth has updated TEX a few more times purely to fix bugs (in the main, truly obscure ones; see `tug.org/texmfbug`). When you run the `tex` command today, you get Knuth's latest release. It is backwards compatible with TEX82.

[4] ε-TEX should not be confused with $\varepsilon_{\mathcal{X}}$TEX [28], which was an updated version of $\mathcal{NTS}$ (`ctan.org/pkg/nts`), a Java-based TEX implementation for which ε-TEX was originally written.

[5] It also adds support for TEX--**XET**, an early method for typesetting bidirectional text, now obsolete. TEX--**XET** is an improved version of TEX-**XET** (one hyphen!) [15], which are both distinct from XETEX, although all of these add support for bidirectional text.

[6] This is why TEX Live still has binaries named `etex`, `pdfetex`, `eptex`, etc.

doi.org/10.47397/tb/46-3/tb144berry-engines-formats

versions descend from the 1995 version [33].[7] A variant supporting Unicode input called upTEX was released in 2007 [33], and additional variants `eptex` and `euptex` added the ε-TEX extensions. As of 2025, they have all been unified to a single binary (`euptex`).

This engine remains very popular in the Japanese community, but is rarely used by others since English-language documentation is limited.

4.4 The pdfTEX engine

pdfTEX was released in 1996 by Hàn Thế Thành [32]. A user generating a document by running the command `pdflatex filename` (or pushing a button on their editor to do that) is using LATEX with this engine.

All earlier engines only produced DVI files that must be postprocessed[8] to produce PDF (or other) output; pdfTEX was the first engine to produce PDF files directly.

pdfTEX also adds some programming extensions (on top of the ε-TEX extensions); LATEX started requiring these in 2017 [23], dropping support for the original TEX82.

Finally, pdfTEX adds so-called "microtypography" features, which improves the output's appearance by, for instance, imperceptibly stretching and shrinking characters to improve line breaking. As a practical benefit, this also greatly reduces the need to rewrite text to avoid overfull lines.

pdfTEX became popular soon after its release, and is the most popular engine today. It is fast and stable.

4.5 The XeTEX engine

XeTEX was first released in 2005 by Jonathan Kew [14]. XeTEX was developed to enhance multilingual support in TEX, and therefore has full support for Unicode and also lets users use TrueType and OpenType fonts directly, without needing to manually generate `.tfm` files.

XeTEX is a "classic" TEX engine in that it internally produces DVI output,[9] but users are typically unaware of this because it internally calls `xdvipdfmx` to convert the DVI file to PDF automatically.

However, because the PDF file is generated separately after the main TEX run has finished, XeTEX cannot fully support microtypography or the ongoing LATEX tagging work, among other limitations. XeTEX remains fast and stable, but new development has stopped, so other engines may be better choices for new projects.

4.6 The LuaTEX engine

LuaTEX was first released in 2006 by Hans Hagen, Hartmut Henkel, and Taco Hoekwater [12]. It was developed independently of (though concurrently with) XeTEX, and supports the same major features, including native Unicode support and the ability to use TrueType and OpenType fonts. LuaTEX is based on pdfTEX[10] and therefore supports nearly all the same features, including microtypography.

Uniquely, it includes MetaPost as a library linked into the binary, making it possible for graphics generation to be exceptionally fast.

Most importantly, LuaTEX adds integration with Lua, a powerful scripting language (see `lua.org` and [9]). This provides a much more conventional programming environment than TEX's macro language, and allows users to dynamically rewrite most of TEX's internal routines using Lua.

Unlike the other engines, LuaTEX does not aim for perfect backwards compatibility with previous engines, including TEX82.

4.7 The LuaMetaTEX engine

LuaMetaTEX is the newest engine at this writing, and was first released by the ConTEXt developers in 2019 [8]. LuaMetaTEX extends LuaTEX by entirely removing the "backend" and font handling code from the compiled binary, instead requiring the format to provide its own Lua implementations of these. This allows formats to quickly update these parts of the code, without having to wait for a new engine release.

LuaMetaTEX is currently supported only by the ConTEXt format, although using other formats is theoretically possible [19].

5 Formats (TEX macro collections)

5.1 The `plain` TEX format

`plain` TEX is the format designed alongside the original TEX engine. It was originally written for TEX78, was updated for TEX82, and has remained essentially unchanged since then. *The TEXbook* describes both macros provided by `plain` and primitives provided by the engine itself, and most formats provide most of the same macros as `plain`.

Minimal examples of `plain` documents may appear foreboding to beginners, but in practice, `plain`

[7] pTEX should not be confused with `PTEX` (also known as TEX80), a Pascal implementation of TEX78 from 1980 [17, 34]. There's rarely a reason to discuss `PTEX` nowadays, though, so there is little risk of confusion.

[8] By `dvips`+`ps2pdf`, `dvipdfm`, `dvipdfmx`, etc.

[9] Referred to as "extended DVI" files (with an `.xdv` file extension) [31], not to be confused with the XDVI files produced by Ω (`omega`) and ℵ (`aleph`) [10]. These latter two were earlier engines supporting Unicode.

[10] Also Ω/ℵ, primarily for right-to-left typesetting.

users usually define their own macros, rather than using only the base `plain` macros.

5.2 The LaTeX 2ε format

LaTeX 2ε is the current version of LaTeX,[11] and was first released in 1994 [25]. LaTeX 2ε is actively maintained with semiannual releases, while most new user-level features are provided by third-party packages. This is by far the most popular format, and needs no introduction for most readers.

LaTeX running on LuaTeX (`lualatex`) is typically slower than LaTeX running pdfTeX (`pdflatex`); this may make a difference in some circumstances.

5.3 The LaTeX3 format

LaTeX3 was first announced in 1989 as a new format [27], but was abandoned in 1994 [26]. However, the LaTeX3 programming layer (`expl3`) was revived in 2009 [22], and integrated into the base LaTeX format in 2020 [24]. Effectively, current LaTeX is both LaTeX 2ε and LaTeX3.

5.4 The ConTeXt format (etc.)

The first version of ConTeXt (retroactively named "Mark I") was written in Dutch in 1991 [7, 13] and publicly released in 1995 [5]. An English rewrite named "Mark II" was released in 2004. These versions were compatible with any engine, although they have special support for pdfTeX and XeTeX.

Development on the next version, "Mark IV", began in 2005 and was publicly released in 2006 [6]. Unlike earlier versions, Mark IV runs with only the LuaTeX engine. The latest version is alternatively called "LMTX" or "Mark XL", was publicly released in 2019 [8], and requires LuaMetaTeX. Hans Hagen is the lead developer for all versions of ConTeXt.

ConTeXt is a much more integrated system than the other formats; although many third-party packages exist, a typical document doesn't load any packages at all.

ConTeXt is unique in that the `context` command is not an engine binary which loads a format, but rather a wrapper script that does many other things (e.g., processing the document multiple times if needed). Another key feature of ConTeXt is a universal key–value interface with uniform syntax.

[11] LaTeX version 2.09 was the final release of the original LaTeX format, although people generally use this name to refer to any pre-2ε version of LaTeX. This first version of LaTeX was released in 1983 by Leslie Lamport, and was maintained until 1985 [3, 20, 21]. LaTeX 2.09 used a slightly different syntax in some cases than modern LaTeX, notably `\documentstyle` instead of `\documentclass`, and is now obsolete. However, to this day, modern LaTeX contains a compatibility mode that accurately emulates LaTeX 2.09's behavior.

ConTeXt is primarily a format, but also comprises an engine that only it uses and requires (LuaMetaTeX, described above), a high-level command runner (`context`, just mentioned), and a distribution (the ConTeXt Standalone Distribution, `wiki.contextgarden.net/Introduction`).

ConTeXt has historically driven engine development: some parts of pdfTeX were developed for ConTeXt, and LuaTeX and LuaMetaTeX were created by the ConTeXt developers for use with ConTeXt.

5.5 The OpTeX format

OpTeX is a modern format written by Petr Olšák, first released in 2020 [30]. OpTeX is designed as a "modern plain TeX":[12] it adds a few substantive features (especially regarding fonts, colors, and hyperlinks; `plain`'s historic weaknesses) but otherwise remains similar to and compatible with the `plain` format. OpTeX requires LuaTeX and makes extensive use of its functionalities.

6 TeX distributions

TeX distributions are how users install TeX on their computers. Distributions handle collecting package updates from package authors and distributing them in a uniform way to users. TeX Live and MiKTeX are the two most popular distributions; both support all the major platforms. MacTeX is TeX Live for macOS, with a native installer and other add-ons.

Full releases of TeX Live happen annually, but packages are updated continuously. Operating systems such as Ubuntu have their own TeX; these are updated on the OS's own schedule.

7 Editors

TeX source files, typically given the extension `.tex`, are plain text files, so they can be edited with any text editor. However, many editors have special support for TeX; common examples include TeXShop, TeXStudio, Emacs, and Vim. In general, what editor you use is irrelevant to your documents' output.

Overleaf (`overleaf.com`) combines a TeX distribution (based on TeX Live) with a web-based editor. Otherwise, it is like any other TeX installation.

8 Summary

For those using pdfLaTeX today, switching to LuaLaTeX may be worth considering, for the ongoing accessibility work, the ability to use OpenType and TrueType fonts, and better Unicode support.

[12] Older formats extending `plain` include Eplain, Lollipop, AMS-TeX, and LaAMS-TeX, of which only the first is still maintained. They all have entries on CTAN, and aside from LaAMS-TeX, are all still included in TeX Live.

A great variety of software continues to be developed as part of the TEX world, but older documents can nearly always be processed by current versions, as reasonable compatibility remains a paramount consideration.

References

[1] P. Breitenlohner. *etex.ch*, Mar. 1998. mirrors.ctan.org/obsolete/systems/e-tex/v2/web/etex.ch

[2] M. Chernoff. Comparing TEX engines and formats. Presentation at TUG'22. Slides: tug.org/tug2022/assets/served/Max_Chernoff-TUG2022-chernoff-engines-slides.pdf. Video: youtube.com/watch?v=MNdAoza8VHU

[3] D. Fuchs. News from all over. *TUGboat* 4(1):14–15, Apr. 1983. tug.org/TUGboat/tb04-1/tb07site.pdf

[4] D. Fuchs. News from the TEX project. *TUGboat* 5(1):22–23, May 1984. tug.org/TUGboat/tb05-1/tb09site.pdf

[5] H. Hagen. The state of ConTEXt. *TUGboat* 25(1):48–51, 2004. tug.org/TUGboat/tb25-1/hagen-context.pdf

[6] H. Hagen. LuaTEX: Howling to the moon. *TUGboat* 26(2):152–157, 2005. tug.org/TUGboat/tb26-2/hagen.pdf

[7] H. Hagen. MKII–MKIV. *TUGboat* 27(2):219–227, 2006. tug.org/TUGboat/tb27-2/tb87hagen-context.pdf

[8] H. Hagen. ConTEXt LMTX. *TUGboat* 40(1):34–37, May 2019. tug.org/TUGboat/tb40-1/tb124hagen-lmtx.pdf

[9] H. Hagen. Lua in TEX. *Journal of Computer Languages*, 85:101354, 2025. doi.org/10.1016/j.cola.2025.101354

[10] Y. Haralambous, J. Plaice. First applications of Ω: Adobe Poetica, Arabic, Greek, Khmer. *TUGboat* 15(3):344–352, Sept. 1994. tug.org/TUGboat/tb15-3/tb44haralambous-omega.pdf

[11] J. Hefferon, K. Berry. The TEX family in 2009. *Notices of the AMS*, 56(3):348–354, Mar. 2009. tug.org/notices/

[12] T. Hoekwater. LuaTEX. *TUGboat* 28(3):312–313, Sept. 2007. tug.org/TUGboat/tb28-3/tb90hoekwater-luatex.pdf

[13] T. Hoekwater. ConTEXt: A short history in 2^4 parts, Mar. 2007. meeting.contextgarden.net/2007/share/taco/history.pdf

[14] J. Kew. XHTEX, the Multilingual Lion: TEX meets Unicode and smart font technologies. *TUGboat* 26(2):115–124, Nov. 2005. tug.org/TUGboat/tb26-2/kew.pdf

[15] D. Knuth, P. MacKay. Mixing right-to-left texts with left-to-right texts. *TUGboat* 8(1):14–25, Apr. 1987. tug.org/TUGboat/tb08-1/tb17knutmix.pdf

[16] D.E. Knuth. *TEX and METAFONT*. Digital Press, Bedford, Ma, Aug. 1979.

[17] D.E. Knuth. TEX and METAFONT: Errata and changes. *TUGboat* 4(2):Supplement, Sept. 1983. tug.org/TUGboat/tb04-2/texmferr-09sep1983.pdf

[18] D.E. Knuth. The new versions of TEX and METAFONT. *TUGboat* 10(3):325–328, Nov. 1989. tug.org/TUGboat/tb10-3/tb25knut.pdf

[19] M. Krüger. LuaMetaLATEX, July 2022. github.com/zauguin/luametalatex

[20] L. Lamport. latex.tex, Mar. 1983. ftp.math.utah.edu/pub/tex/historic/macros/latex-saildart/latex-0.90/latex.tex

[21] L. Lamport. Announcement of LATEX version 2.09. *TUGboat* 6(3):150–151, Nov. 1985. tug.org/TUGboat/tb06-3/tb13lamport.pdf

[22] LATEX Project Team. LATEX3 news, issues 1–2. *TUGboat* 30(1):105–106, 2009. tug.org/TUGboat/tb30-1/tb94l3news.pdf

[23] LATEX Project Team. LATEX news, issue 26, January 2017. *TUGboat* 38(1):54–55, 2017. tug.org/TUGboat/tb38-1/tb118ltnews26.pdf

[24] LATEX Project Team. LATEX news, issue 31, February 2020. *TUGboat* 41(1):34–38, 2020. tug.org/TUGboat/tb41-1/tb127ltnews31.pdf

[25] LATEX Project Team. *LATEX 2ε for authors*, May 2025. ctan.org/pkg/usrguide

[26] F. Mittelbach. LATEX anniversaries — a look in two directions. *TUGboat* 44(1):77–86, 2023. doi.org/10.47397/tb/44-1/tb136mitt-history

[27] F. Mittelbach, C. Rowley. LATEX 2.09 ↪ LATEX3. *TUGboat* 13(1):96–101, Apr. 1992. tug.org/TUGboat/tb13-1/tb34mittl3.pdf

[28] G. Neugebauer. *εχTEX User's Guide*, July 2005. extex.org/documentation/man/extex-users.pdf

[29] H. Okumura. pTEX and Japanese typesetting. *The Asian Journal of TEX*, 2(1):43–51, Apr. 2008. web.archive.org/web/20240819102104/https://ajt.ktug.org/2008/0201okumura.pdf

[30] P. Olšák. OpTEX — A new generation of Plain TEX. *TUGboat* 41(3):348–354, Nov. 2020. tug.org/TUGboat/tb41-3/tb129olsak-optex.pdf

[31] W. Robertson, K. Hosny, K. Berry. *The XHTEX reference guide*, Feb. 2025. ctan.org/pkg/xetex

[32] P. Sojka, Hàn Thế Thành, J. Zlatuška. The joy of TEX2PDF — Acrobatics with an alternative to DVI format. *TUGboat* 17(3):244–251, Sept. 1996. tug.org/TUGboat/tb17-3/tb52sojk.pdf

[33] T. Tanaka. upTEX — Unicode version of pTEX with CJK extensions. *TUGboat* 34(3):285–288, Dec. 2013. tug.org/TUGboat/tb34-3/tb108tanaka.pdf

[34] I. Zabala. Some feedback from PTEX installations. *TUGboat* 2(2):16–19, July 1981. tug.org/TUGboat/tb02-2/tb03zab.pdf

◇ Karl Berry, Max Chernoff, Jim Hefferon
https://tug.org/levels