

🌀 LE MYTHIQUE L^AT_EX₃

Traduction de l'article The mythical LaTeX3 publié le 11 novembre 2024 par Joseph Wright sur son blog Some TeX Developments⁶³ [1]. Avec l'aimable autorisation de l'auteur.

Lorsque l'équipe du projet L^AT_EX a pris la suite de Leslie Lamport pour assurer la maintenance de L^AT_EX, L^AT_EX_{2 ϵ} était prévue comme une version intermédiaire pour résoudre les problèmes immédiats de L^AT_EX_{2.09}. Le « ϵ » était censé indiquer un petit changement par rapport à L^AT_EX_{2.09}, les plans à plus long terme étant centrés sur L^AT_EX₃. Comme nous le savons, cela s'est révélé plutôt optimiste, et nous utilisons toujours L^AT_EX_{2 ϵ} aujourd'hui : du moins, c'est ainsi qu'il s'annonce encore.

Les plans des années 1990

Les plans de L^AT_EX₃ élaborés par l'équipe au début des années 1990 envisageaient de réimplémenter L^AT_EX avec un cadre de programmation sous-jacent beaucoup plus propre et une séparation bien définie de l'interface utilisateur, des fonctions de conception et de la programmation. Beaucoup de ces idées ont été prototypées très tôt, avec une version de la couche de programmation créée bien avant la finalisation de L^AT_EX_{2 ϵ} . Mais la performance était le grand problème : oui, vous pouviez tester des choses en tant que développeur, mais c'était essentiellement cela. En même temps, certaines idées n'étaient pas tout à fait prêtes : avant ϵ -T_EX (extensions pdfT_EX, etc.), le moteur ne pouvait pas faire grand-chose, et bien que T_EX de Knuth soit un excellent logiciel, les idées de L^AT_EX₃ le poussaient à la limite.

L^AT_EX_{2 ϵ} a donc été publié en 1994, et le développement s'est concentré sur les extensions pendant de nombreuses années. Cela signifie que « L^AT_EX₃ » est devenu une idée amusante : on en parle, mais il n'y a aucune chance qu'elle aboutisse (de manière visible).

Développement dans les années 2010

Le développement des idées de « L^AT_EX₃ » ne s'est jamais arrêté, bien sûr : c'est juste qu'elles ont cessé de viser principalement un nouveau noyau pour s'appuyer sur L^AT_EX_{2 ϵ} . Il en résulte une extension appelée l3in2e et une autre appelée xpackages sur CTAN, bien que toutes deux aient été marquées comme expérimentales.

Lorsque j'ai rejoint l'équipe du projet L^AT_EX en 2009, c'était en grande partie le cas. Cependant, le paysage avait évolué : les ordinateurs étaient devenus plus puissants, les extensions ϵ -T_EX et pdfT_EX de T_EX étaient très largement disponibles et le développement était plus facile. Cela signifiait que les idées de L^AT_EX₃, en particulier le langage de programmation (expl3), étaient utilisables dans de vrais documents : le package `fontspec` en particulier était écrit en expl3 et était tout à fait utilisable. Je m'en suis inspiré et j'ai écrit la deuxième version de `siunitx` en expl3.

En même temps, une partie de mon premier travail dans l'équipe a été d'amener expl3 au point où le terme *expérimental* n'était plus vraiment

⁶³. <https://www.texdev.net/2024/11/11/the-mythical-latex3>.

vrai : vous pouviez l'utiliser dans une extension et être heureux que le code continue à fonctionner entre les versions. L'autre tâche majeure au début de ma carrière dans l'équipe a été de nous mettre dans une position où les versions, d'abord d'expl3 et ensuite du noyau $\LaTeX 2_{\epsilon}$, étaient faciles à faire à partir de n'importe quelle plate-forme que nous voulions : Linux, macOS et Windows.

Au cours des années 2010, cela a permis à l'équipe de compléter des idées qui avaient été suggérées pour la première fois plus de 20 ans auparavant. Le développement se faisait toujours sous forme d'extensions s'ajoutant au noyau, mais `xparse` (commandes de documents), `xtemplate` (structures de documents flexibles) et `xgalley` (une nouvelle approche de la liste verticale principale) ont toutes été révisées et testables.

Les années 2020 : $\LaTeX 2_{\epsilon}$ 2020-02-02 et au-delà

En 2020, le travail sur l'idée centrale d'une couche de programmation avait atteint le point où le code était stable et performant, et où l'équipe voulait pouvoir s'y fier pour les développements du noyau. Nous avons donc pris la décision d'intégrer le chargement d'expl3 dans le noyau $\LaTeX 2_{\epsilon}$, avec la description plutôt énigmatique *Improved load times for expl3* (Temps de chargement améliorés pour expl3) dans *\LaTeX News*. Par la suite, `xparse` et `xtemplate` ont également été (en grande partie) ajoutés au noyau, en séparant les parties qui fonctionnaient de celles qui s'avéraient n'être que des expériences.

Avec cette fonctionnalité disponible, de nouvelles idées telles que la gestion des crochets⁶⁴ et la réimplémentation d'anciennes idées telles que les listes sont possibles : certaines d'entre elles ne sont actuellement activées que si vous utilisez `\DocumentMetadata` comme *marqueur*. Pour les documents mis à jour, cela signifie que de nombreuses idées de $\LaTeX 3$ sont utilisées : la couche de programmation, une façon de créer des commandes de document (`\NewDocumentCommand` provenant à l'origine de `xparse`) et des modèles (provenant à l'origine de `xtemplate` et utilisés pour mettre à jour le code des listes, etc.)

Il y a aussi des changements plus subtils dans le noyau, par exemple la mise à jour des commandes pour utiliser UTF-8 partout. De même, on a rendu plus de fonctions insensibles aux moteurs : elles se comportent de la même manière quel que soit le moteur \TeX utilisé⁶⁵. Dans l'ensemble, le $\LaTeX 2_{\epsilon}$ de 2024 est très différent de celui de 1994, mais vous pouvez toujours compter sur la stabilité à long terme qui a toujours été présente.

Conclusions

Il n'y aura jamais de format « $\LaTeX 3$ » autonome, mais les idées qui ont été explorées pour la première fois par l'équipe du projet \LaTeX au début des années 1990 sont désormais à la disposition des utilisateurs dans le noyau \LaTeX . Elles sont utilisées pour fournir un \LaTeX mis à jour, capable de produire des documents accessibles et, plus largement, pour apporter

64. NdT : Ici, « crochet » est la traduction de *hook*. Ceux-ci simplifient grandement la programmation sous \LaTeX (cf. `lthooks-doc.pdf`).

65. NdT : je me permets de gloser ici, DeepL fournissant, plutôt qu'une traduction, un décalque de l'anglais *make more functions engine-robust* avec *rendre plus de fonctions robustes au moteur* dont je doute qu'il soit immédiatement compréhensible.

une personnalisation au noyau. Ainsi, même si vous ne direz, peut-être, jamais que vous utilisez « $\text{\LaTeX}3$ », vous bénéficiez de ces fonctionnalités aujourd'hui.

Joseph Wright

traduit par l'extension `DeepL` pour le navigateur `Firefox`,
révision et correction par Yvon Henel.

Références

- [1] Joseph WRIGHT. *The Mythical LaTeX3*. Consulté le 27 avril 2025. Nov. 2024. URL : <https://www.texdev.net/2024/11/11/the-mythical-latex3>.



🔍 À LA RECHERCHE DES FONTES DE CE NUMÉRO

Que ne découvre-t-on pas en utilisant la commande `fc-list` ? Ou en explorant le répertoire `/usr/local/texlive/2024/texmf-dist/fonts/opentype/`, et au sein de celui-ci, le répertoire `public/` ?

Un dossier m'a attiré l'œil : `qualitytype`. J'ignorais totalement ce dont il s'agissait, et l'instruction `texdoc qualitytype`⁶⁶ ne m'a guère éclairé sur ce point : ces fontes ont été créées par `QualityType`, et grâce à un certain John Colletti, ces fontes sont disponibles sous licence libre (la licence `SIL`, sur laquelle nous écrivons un jour un article⁶⁷) et ont été placées là en 2019 par un nommé Daniel Benjamin Miller en vue d'un futur package [1]. Le dossier contient la version `OpenType`, datant de 2009, des 151 fontes originales de 1992. Et me voilà bien avancé.

J'ai entrepris de tester quelques-unes de ces fontes, tout simplement en insérant la commande `\setmainfont{<fonte>}`, et j'ai été heureusement surpris par leur variété. Quelle heureuse surprise que de voir apparaître la fonte `QTabbie` !

66. Ce dossier fait partie de la distribution `TpX Live` : il est donc logique d'essayer d'en apprendre plus à son propos grâce à cette commande.

67. Voir la note 32, au bas de la page 24 de la *Lettre* 54.