

# Nouvelles de L<sup>A</sup>T<sub>E</sub>X<sub>3</sub>

Première édition, février 2009

## *Bienvenue dans L<sup>A</sup>T<sub>E</sub>X<sub>3</sub>*

La projet L<sup>A</sup>T<sub>E</sub>X<sub>3</sub> connaît un regain de dynamisme. Les fondations expérimentales de la programmation pour L<sup>A</sup>T<sub>E</sub>X<sub>3</sub> sont disponibles, sous le nom de `expl3`, depuis quelques versions de T<sub>E</sub>X Live. Malgré les gros avertissements disant que le code changerait probablement à l'avenir, nous voulions montrer que des progrès étaient en cours, même s'ils étaient lents. Depuis, plusieurs personnes ont regardé le code, fourni des retours, et (c'est le plus important) essayé de l'utiliser réellement. Bien qu'on en soit aux commencements, nous pensons que les idées sous-tendant le code sont solides et que seuls quelques « changements cosmétiques » sont nécessaires avant que `expl3` ne soit prêt pour l'ensemble des auteurs de modules L<sup>A</sup>T<sub>E</sub>X.

## *Ce qui existe déjà*

Le code actuel de L<sup>A</sup>T<sub>E</sub>X<sub>3</sub> consiste en deux branches principales : les modules `expl3` qui définissent l'environnement de programmation de base, et les `xpackages`, qui sont une suite de modules écrits avec l'interface de programmation `expl3` et fournissent des fonctionnalités de plus haut niveau pour ce qui deviendra un jour L<sup>A</sup>T<sub>E</sub>X<sub>3</sub> à proprement parler. `expl3` et `xpackages` sont tous deux conçus pour être utilisés avec L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, de sorte que les modules nouvellement écrits puissent profiter des nouvelles fonctionnalités, tout en étant utilisables en même temps que les nombreux modules L<sup>A</sup>T<sub>E</sub>X 2 du CTAN.

## *Ce qui se passe actuellement*

Pour préparer une petite révision du code de `expl3`, nous sommes en train d'écrire une suite de test exhaustive pour chaque module. Ces tests nous permettent de modifier l'implémentation et de vérifier que le code marche encore comme avant. Ils mettent aussi en valeur des petits problèmes ou omissions dans le code. Pendant que nous écrivons les tests, nos suppositions sur ce qui doit s'appeler comment, les conventions de nommage sous-jacentes pour les fonctions et les types de données sont remises en question, et ces questions notées pour être ruminées davantage.

Au moment où ces lignes sont rédigées, nous avons écrit à peu près la moitié de la suite de tests. Une fois ce travail terminé, ce qui est prévu pour le premier semestre 2009, nous serons prêts à faire des changements sans nous inquiéter de casser quoi que ce soit.

## *Ce qui se passera bientôt*

Bon, que voulons-nous changer ? Il y a dans le code `expl3` des portions datant de la période pré-L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, et d'autres modules conçus plus récemment. Il est visible en lisant les sources d'un bout à l'autre qu'un peu d'unification et de ménage augmenterait la simplicité et la cohérence du code. Dans plusieurs cas, un tel changement ne serait qu'un simple ajustement ou renommage.

Au-delà de ces changements mineurs, nous repensons aussi actuellement la notation exacte qui gouverne les définitions de fonctions. Il y a actuellement un certain nombre de types d'arguments qu'on peut passer aux fonctions (d'un simple lexème non modifié au développement complet d'une liste de lexèmes) et nous ne sommes pas totalement satisfaits de la façon dont ont évolué les choix initiaux, maintenant que le système a un peu grandi. Nous avons eu des bons retours de plusieurs personnes sur des façons potentielles d'améliorer la syntaxe des arguments, et nous espérons attaquer les problèmes que nous voyons maintenant dans la syntaxe actuelle au cours des prochains changements dans les modules `expl3`.

## *Ce qui se passera plus tard*

Une fois finis les changements que nous venons d'évoquer, nous commencerons à geler le cœur de l'interface des modules `expl3`, et nous espérons que plus d'auteurs de modules auront envie d'utiliser ces nouvelles idées pour écrire leur propre code. Alors que les fonctions de base demeureront inchangées, plus de fonctionnalités et de nouveaux modules seront ajoutés lorsque L<sup>A</sup>T<sub>E</sub>X<sub>3</sub> commencera à grandir.

Quelques modules nouveaux et/ou expérimentaux vont être modifiés pour utiliser l'interface de programmation `expl3`, comme `breqn`, `mathtools`, `empheq`, `fontspec` et `unicode-math`. (C'est une des raisons de l'absence d'avancées dans les deux derniers récemment.) Il y aura également une version du module `siunitx` écrite en `expl3`, parallèlement à la version L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. Ces développements apporteront des améliorations à des utilisateurs quotidiens de L<sup>A</sup>T<sub>E</sub>X qui n'ont même pas entendu parler du projet L<sup>A</sup>T<sub>E</sub>X<sub>3</sub>.

Concernant le plus long terme, L<sup>A</sup>T<sub>E</sub>X<sub>3</sub> en tant que système de préparation de documents doit être écrit pratiquement en partant de rien. Il faut concevoir une syntaxe de plus haut niveau, et une multitude de modules sera utilisée comme source d'inspiration pour les modèles de document par défaut. L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> a résisté à l'épreuve du temps (quelque 15 ans et toujours bien vivant) et il est maintenant temps d'écrire un successeur qui survivra pour les prochaines décennies.