

Présentation professionnelle de Tex(te) avec LyX



par Gerrit Renker
<gerrit.renker(at)gmx.de>

L'auteur:

Diplômé en 2001 en réseaux informatiques. De nombreux jobs avant et, si le libre ne perdure pas, de nombreux jobs dans l'avenir.

Traduit en Français par:
Laurent Richard
<kouran(at)linuxmail.org>



Résumé:

Cet article est destiné aux personnes qui ont beaucoup de textes à écrire et à éditer et qui désirent les faire d'une manière rapide, fiable, bien en forme sans pour cela devoir acquérir des connaissances en traitement de texte ou devoir développer une maîtrise de LateX digne d'un gourou. L'outil principal que cet article va présenter est une interface graphique L^ATEX extrêmement robuste et puissante basée sur une architecture client–serveur.

Ceci, qui n'est guère plus qu'un résumé de l'application, se concentrera sur le processus de rédaction et parlera de mon expérience suite à l'utilisation quotidienne de LyX depuis cinq ans. D'autres points, comme le versioning, les différents « brouillons » préparatoires à la version définitive du texte) ou les graphiques à utiliser seront également abordés et des outils complémentaires à la préparation des documents seront présentés.

Motivation

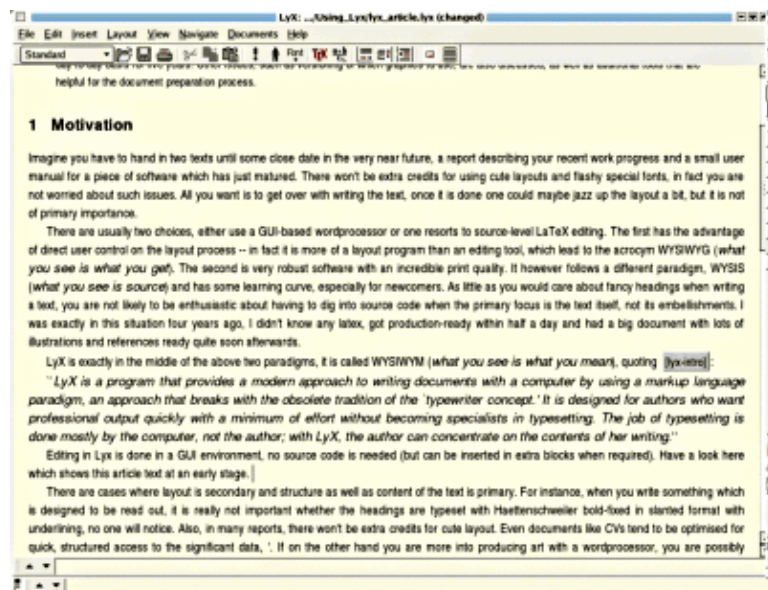
Imaginez que vous devez rédiger deux textes dans un délai extrêmement court. L'un est un rapport décrivant le résultat de vos longues heures de travail et l'autre un petit manuel destiné aux utilisateurs d'un programme qui vient juste d'arriver au stade mature. Vous ne pourrez compter sur aucune rallonge de temps pour réaliser une mise en page attrayante ou des polices spectaculaires. Dans un premier temps, vous ne vous préoccupez pas de ça car tout ce que vous voulez, c'est terminer la rédaction. Ensuite, quand ce gros travail est effectué, vous améliorez un peu la présentation mais ce n'est pas très important.

Il y a généralement deux choix, soit utiliser un traitement de texte basé sur une interface graphique ou s'en remettre à l'édition L^ATeX au niveau des sources même du texte. La première méthode a l'avantage de permettre un contrôle direct sur le processus de mise en page car ici, il s'agit plus d'un programme de mise en page que d'un véritable outil d'édition. Cela est résumé par l'acronyme anglais **WYSIWYG** (*what you see is what you get – ce que vous voyez est ce que vous aurez*). La seconde méthode, quant à elle, est un logiciel solide avec une très grande qualité d'impression. Ce dernier suit cependant une autre logique, celle de l'acronyme **WYSIS** (*what you see is source – ce que vous voyez est la source*) et demande une certaine dose d'apprentissage, en particulier pour les nouveaux venus. Comme vous vous souciez peu d'avoir de jolies polices pour vos titres, vous n'êtes pas enthousiaste à l'idée de plonger dans le code source pour les mettre en place car votre préoccupation première est le texte en lui-même et non son embellissement ou son marquage par balises. J'étais dans cette situation il y a quatre ans. Je ne connaissais rien de L^ATeX, je ne savais pas comment avoir mon texte prêt à être présenté dans la demi-journée sous forme d'un gros document avec beaucoup d'illustrations et de références prêtes presque instantanément. Tout cela a été possible grâce à Lyx.

LyX se trouve exactement à mi-chemin des deux systèmes décrits plus haut, ici on parle de **WYSIWYM** (*what you see is what you mean – ce que vous voyez est ce que vous voulez dire*) [1(voir les références)] :

"LyX est un programme qui fournit une nouvelle approche de la rédaction de documents avec un ordinateur en permettant une mise en page automatique du texte par balisage. C'est une approche qui rompt avec la tradition vieillotte du 'concept du typographe'. Il est conçu pour les auteurs qui veulent un résultat final de type professionnel rapidement et avec un minimum d'efforts, sans pour cela devenir des typographes. Le travail de mise en page est réalisé principalement par l'ordinateur et non par l'auteur. Avec LyX, l'auteur peut se concentrer sur le contenu de son texte."

Editer avec LyX se fait sous un environnement graphique, aucun code source n'est nécessaire (mais vous pouvez en insérer dans des blocs supplémentaires si nécessaire). Regardez ici ce que donne cet article à un stade peu avancé.



La boîte grise que vous voyez est une référence croisée, vous pouvez cliquer dessus et éditer la référence. La fenêtre d'édition n'est pas à confondre avec le résultat final. Lyx génère un *fichier texte* qui peut ensuite être converti en L^ATeX. Vous pouvez voir le résultat complet en pressant **Ctrl-d**, ce qui enclenche l'apparition d'une fenêtre DVI (**DVI=device-independent-format, un format de fichier intermédiaire utilisé par L^ATeX**), **Ctrl-t** donne la même chose pour le Postscript. Les variantes avec les majuscules **Ctrl-D**, **Ctrl-T** sont utilisées lorsque vous désirez mettre à jour les changements que vous avez réalisé dans la fenêtre de

l'éditeur.

La procédure est donc similaire au processus d'édition L^ATeX classique avec l'avantage d'avoir la structure du texte en face de vous et sans avoir de mélanges avec des balises du code source. La tentation lors de l'édition en pur L^ATeX est souvent, comme avec les autres traitements de texte, de vouloir augmenter un peu la police ici, de changer un peu les paramètres d'environnement là et en conclusion, faire de nombreuses choses qui n'ont aucun intérêt pour le texte en lui-même.

En résumé, lorsque vous avez besoin d'un outil de rédaction fiable et que vous désirez vous concentrer sur le texte en lui-même, Lyx est vivement recommandé. Ses points forts sont de gérer de grands documents (lesquels peuvent être divisés en d'autres de tailles plus petites), son intégration avec les autres logiciels décrits ci-dessous et sa robustesse à tout épreuve. Dès lors que moins de mémoire principale est dépensée lors du cycle du rafraîchissement de la mise en page actuelle, il y a moins de ralentissements lors de l'édition de grands documents. J'ai eu écho que des programmes de traitements de texte se plantaient avec de grands documents (en produisant une poubelle de binaire irrécupérable). Lyx est basé sur du texte et lors que l'on quitte de façon brutale (exemple : panne de courant), il crée un fichier d'urgence. Même avec les paramètres les plus bizarre et des longueurs de texte imposantes, je n'ai pas été témoin d'une diminution des performances. Je l'ai utilisé depuis les cinq dernières années de façon quotidienne, j'ai écrit environ 800 pages de textes, parfois plus et je n'ai jamais perdu plus d'un demi paragraphe. Comme cerise sur le gâteau, le code de L^ATeX peut être affiché dans LyX via le script **reLyX(1)**.

Si vous préférez vraiment faire de l'édition de code L^ATeX, un coup d'oeil à [kile](#) peut être très instructif. C'est un éditeur KDE spécial pour L^ATeX avec des possibilités supplémentaires.

Commençons

Cet article diffère du premier de [mars 1998 \(article 31\)](#) en ce qu'il fournit moins d'introduction généraliste. Ici, je vais plus préciser où vous pouvez trouver de l'information et comment l'obtenir. Au lieu de faire un **guide graphique** (comme initialement prévu) si vous n'avez jamais vu LyX auparavant, je vais plutôt vous faire découvrir le processus d'édition. Cela va vous montrer comment mettre la main à la pâte pour votre premier document d'une très jolie manière : <http://www.lyx.org/LGT/>

Pré-requis

La première chose à faire est de vérifier si L^ATeX est déjà complètement installé sur votre système. Sur les systèmes RedHat/Fedora, il est possible de le faire de manière assez conviviale avec l'installateur graphique de paquets qui possède une simple case à cocher pour installer la totalité de la distribution L^ATeX. Vous pouvez également le faire manuellement via

```
rpm -qa | grep tex
```

ce qui vous permet de lister les paquets suivants (en supposant que vous avez l'implémentation TeTeX de LaTeX):

- **tetex** et **tetex-latex** (paquets de base)
- **tetex-afm** (un convertisseur de polices PostScript pour TEX)
- **tetex-xdvi** (pour pré-visualiser les fichiers **.dvi**)
- **tetex-doc**

Le dernier est très utile si vous désirez vérifier différents paquets. Cette documentation réside dans `/usr/share/texmf/doc/index.html`. Comme pré-requis supplémentaires, vérifier pour **aiksaurus** (un thésaurus utilisé lors de l'édition) et, plus important, le vérificateur orthographique **aspell**. Aspell a également des paquets pour plusieurs langues autres que anglaises. Dans Lyx, le vérificateur orthographique est activé via **F7** ([capture d'écran](#)).

Installation

LyX se présente sous deux variantes, la variante ancienne x-forms (laquelle est à présent un peu complexe à construire) et la variante moderne Qt qui est celle que je vous conseillerais de choisir. Les RPM pour les deux versions sont disponibles. Sous **Debian**, l'installation est aussi simple que

```
apt-get install lyx lyx-common lyx-qt          ## ou lyx-xforms si vous préférez
```

Si vous désirez compiler à partir des sources par vous-même, elles sont disponibles sur www.lyx.org. Vous pourrez trouver le [script \(build-lyx.sh\)](#) de construction assez utile. LyX est un petit chef-d'oeuvre de codage, tout est écrit en C++. Le processus de compilation, si vous préférez cette voie, vous prendra plus longtemps que celle du noyau.

Configuration

LyX a besoin de vérifier tout d'abord votre installation. Avant de commencer à éditer pour la toute première fois, lancer **Editer** → **Reconfigurer**, ce qui va mettre à jour la base de données interne. Lors d'un changement quelconque de votre installation L^ATEX, cette étape doit également être relancée. Les résultats de cette étape peuvent être consultées sous **Aide** → **Configuration L^ATEX** qui nous avertit également où nous procurer d'éventuels paquets manquants. Il est également très utile d'activer la *sauvegarde automatique* sous **Editer** → **Préférences**, cela ne demande pas beaucoup de temps et peut vous en faire gagner énormément en cas de problème. La visionneuse par défaut pour la sortie est `xdvi`. Il est chaudement recommander de mettre à la place `kdvi`, comme montré sur cette [capture d'écran](#). Cela vous donnera une visionneuse bien plus simple à utiliser. Le changement des paramètres n'est appliqué qu'après avoir appuyé sur **Modifier**.

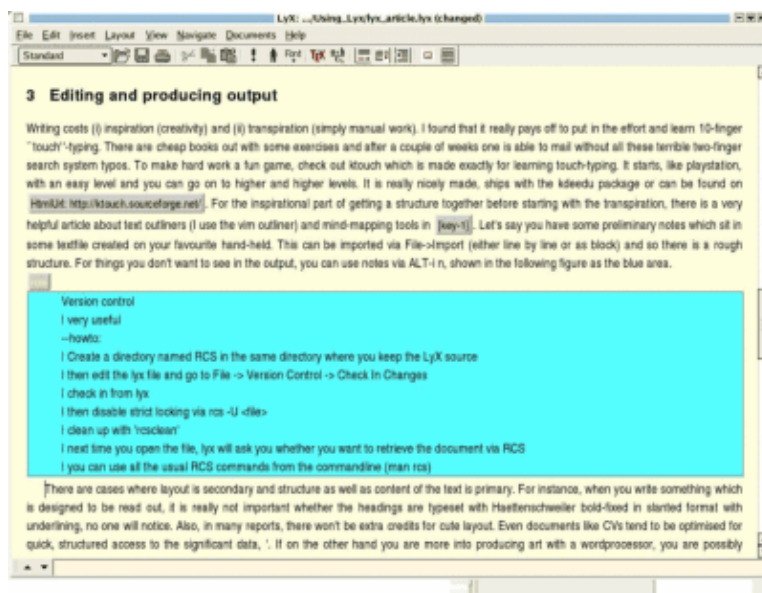
Édition et réalisation d'un document en sortie

Rédiger demande de l'inspiration (de la créativité) et de la transpiration (du travail manuel). J'ai trouvé qu'il est vraiment utile de faire l'effort d'apprendre la dactylo à 10 doigts. Il existe des livres très bon marché avec pas mal d'exercices et après quelques semaines, on est capable de taper sans perdre toujours un temps fou à rechercher où se trouve la lettre qu'on désire utiliser. Afin de faire de cet exercice fastidieux un amusement, jetez un coup d'oeil à [ktouch](#) qui est conçu pour apprendre la dactylographie. Cela commence comme un jeu avec un niveau facile et vous pouvez atteindre des niveaux de plus en plus difficiles. C'est vraiment bien fait, il vient avec le paquet **kdedu** ou il peut être trouvé sur <http://ktouch.sourceforge.net/>.

Pour le *côté inspiration* – préparer une structure solide avant de débiter le travail, il existe un article très utile à propos des *text outliners* et des *outils mind-mapping* que vous pourrez trouver au [2(voir références)].

Édition de base

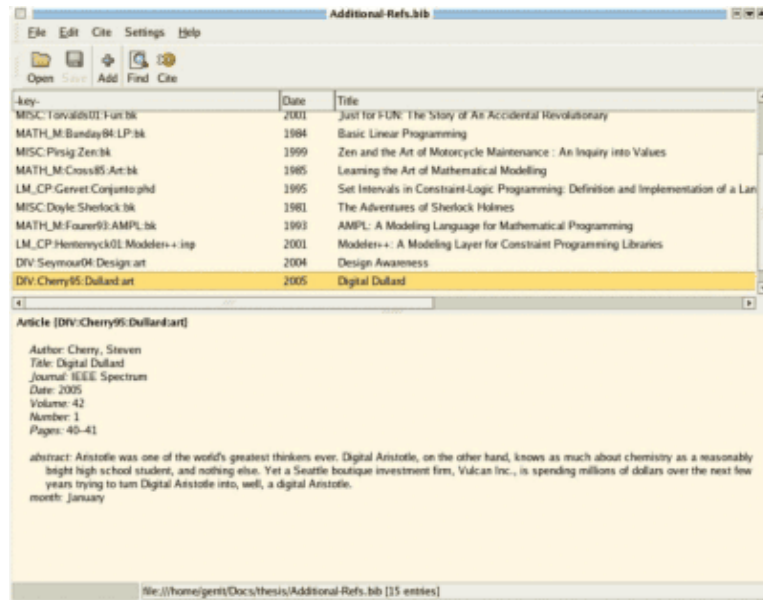
Il n'y a pas grand chose à dire à ce propos étant donné que l'interface contient suffisamment d'explications. Si vous êtes bloqué, jetez un coup d'oeil à la [visite graphique](#) mentionnée plus haut. Disons que vous avez quelques notes préliminaires sous un format texte créées avec votre assistant personnel portable. Elles peuvent être importées via **Fichier**→**Importer** (soit en lignes ou en paragraphes), ce qui vous donne une structure brute pour démarrer. Celle-ci peut maintenant être convertie en segments de texte. La *liste* disponible sur la barre d'outils présente les options possibles pour un style de fichier défini. Généralement, on peut affecter le style *standard* via **ALT-p s** et les *titres* via **ALT-p n** où **n** est un chiffre entre 1 et 6 (plus le chiffre est petit, plus le titre sera grand). Pour les éléments que vous ne voulez pas voir apparaître dans la version finale, vous pouvez utiliser les *notes* via **ALT-i n**, comme montré dans l'image suivante dans la zone bleue.



Une fois le texte terminé, vous pouvez utiliser **Visualiser** et vérifier ou l'exporter (**Latex, PDF, Ascii, HTML** (voir les notes en bas de page), ...) via **Fichier**→**Exporter**.

Références

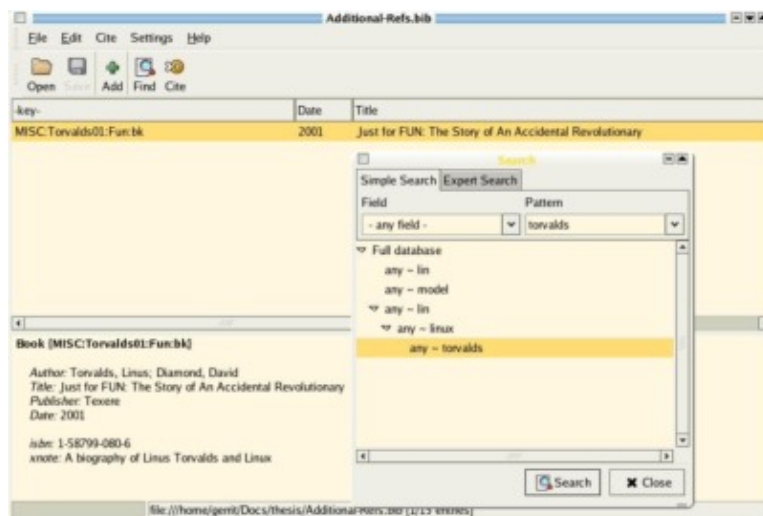
L^ATEX fun commence avec un choix extraordinaire de possibilités de sections, sous-sections, pages, images, tables de *références croisées* ainsi que des données bibliographiques externes. Afin de *référencer* quelque chose dans le texte, une *étiquette* doit tout d'abord être créée via **ALT-i q**. Tout texte peut être utilisé pour faire une étiquette, à condition d'éviter d'y mettre des espaces. Après avoir fait cela, ce label peut être pré (ou post) référencé n'importe où dans le texte par simple insertion via **ALT-i r** (pour les références), ce qui vous va vous donner le choix entre différents modes de référencement. Pour la citation d'*URL* à l'intérieur du texte, utilisez **ALT-i u**.



Concernant les *références bibliographiques*, vous avez deux possibilités :

1. Créer une *section bibliographique* manuellement à la fin du document en sélectionnant '**Bibliographie**' depuis le sélecteur de style.
2. Utiliser une *base de données BibTeX* externe ce qui est référencé en faisant un **Insérer** → **Listes&Tdm** → **Références BibTeX** (autant que vous le désirez, cela sera mis à la fin du texte).

Pour la première option, vous pouvez vérifier la première version de cet article (NdT : en anglais) [ici](#) ([lyx_article.lyx](#)) comme exemple. Si vous n'avez jamais vu une **base de données BibTeX** auparavant, voici un [fichier exemple](#) ([exemple_bibtex_database.bib](#)) . Vous pouvez remarquer que c'est un *signet-texte* standardisé de données bibliographiques. Il n'est pas du tout nécessaire de se compliquer la vie avec cela étant donné qu'il existe un superbe *outil de base de données* appelé **publiographer**. Il permet la gestion complète de toutes sortes de base de données incluant les recherches de texte incrémentielles multi-attributs et la conversion vers d'autres formats comme le HTML par exemple. Voici une capture d'écran d'une recherche incrémentielle.



Lorsque vous avez réalisé une des deux alternatives, vous pouvez utiliser les *références bibliographiques* n'importe où dans le texte via **ALT-i c** (pour citer), comme [ici](#). Mais il y a encore une bien meilleure manière

qui va vous faire découvrir toute la flexibilité de LyX. [Pybliographer](#), comme d'autres outils peut s'interfacier avec le procédé du serveur Lyx. Cela se passe via les *tunnels* `$HOME/.lyx/lyxpipe.in` (et `lyxpipe.out`). Le protocole du serveur est décrit en détail dans la documentation de LyX. Dans le cadre de cet article, je m'en tiendrai à vous dire que vous pouvez simplement cliquer sur '**Citation**' dans pybliographer pour insérer la référence que vous venez juste de redécouvrir comme partie d'une recherche de base de données incrémentielle. Pour les autres applications, ce protocole de communication ouvre des portes comme la génération automatique de textes à partir de scripts et ... bien d'autres choses que je vous laisse imaginer.

Contrôle de version

C'est particulièrement utile lorsque le texte est long et change régulièrement. Lyx est prêt pour le RCS. Il est donc simple de le mettre en place.

1. créez un répertoire nommé **RCS** dans le même répertoire que celui des sources de LyX
2. éditez le fichier LyX et allez sur **Fichier** → **Contrôle de Version** → **Initialiser le Contrôle (dialogue de texte)**
3. la prochaine fois que vous ouvrirez le fichier, vous obtiendrez une boîte de dialogue celle-ci :



Vous pourrez alors avoir toutes les commandes habituelles RCS depuis la ligne de commande (man **rscs** ou **rscsintro**). Je n'utilise généralement que **rscs -l** pour verrouiller, **co -l** quand je souhaite mettre à jour la version verrouillées et **rscsclean** pour supprimer tous les fichiers non modifiés. Si vous êtes l'auteur unique, désactivez le verrouillage strict avec **rscs -U <file>**, ce qui simplifiera considérablement les processus.

Question Math

Si vous avez besoin d'éditer pas mal de formules, alors LyX/L^AT_EX propose une des meilleures option qui existe. Jetez un coup d'oeil au *module mathématique*, en y accédant via **ALT-m m** (eh oui, deux fois) et vous pourrez choisir un symbole depuis **Insérer**→**Math**→**Palette Mathématique**. Quand vous tapez des commandes math LaTeX, LyX les reconnaît. Essayez par exemple **ALT-m m** et ensuite la séquence **\mathcal P**, puis *espace* et finalement **\frac (n-1) TAB 2**.

Graphiques

Une variété de formats de graphiques peut être utilisée. Le Postscript encapsulé (**.eps**) est souvent la meilleure option pour produire des **.ps**, mais les **.png**, les **.jpg** et les **.gif** fonctionnent également. Vous pouvez mettre des images directement via **ALT-i g** (graphismes), mais c'est mieux pour la mise en page d'utiliser les couramment appelés *flottants*, qui sont de simples « styles décoratifs » et qui permettent différentes options de positionnement ainsi que l'ajout d'une légende descriptive. Pour créer un flottant, utilisez **ALT-i o** et ensuite '**Figure**'. Lorsque vous faites un clic droit sur le cadre extérieur, vous pouvez forcer le positionnement (jusqu'à une certaine limite, c'est du LaTeX, pas un programme de mise en page).

Paramètres du document et formats de mise en page

Je n'ai pas beaucoup parlé du *format de mise en page* du document qui est défini via **Format**→**Document**. Il

y a normalement une sélection de base de formats comme ceux pour un article, un rapport, une lettre, un livre, etc ... tout ce que Lyx a trouvé lors de son démarrage de test. En choisissant l'un d'entre eux, la mise en page de l'ensemble du document change. Cela change directement la liste des styles disponibles sur la gauche du menu. En interne, ces choix sont des styles de mise en page de Lyx enveloppant les classes de documents LaTeX. Plus bas dans cet article, je vous donne un petit exemple sur la manière de modifier une telle mise en page (ce qui n'est pas bien difficile). Mais pour la plupart des utilisations, l'un des formats donnés devrait plus que convenir vu qu'ils peuvent se voir adjoindre des caractéristiques et des ajouts sélectionnables supplémentaires.

Une option intéressante est celle de l'utilisation de *modèles* spécialement pour la rédaction de lettres. Les modèles sont de simples fichiers textes qui peuvent être modifiés, par ex. pour votre propre format de lettre. Essayez **Fichier → Nouveau avec Modèle**. Cela ouvre l'affichage du modèle dans `/usr/share/lyx/templates`. Un modèle très intéressant est par exemple la **g-brief** (de l'allemand "Geschäftsbrief" = lettre commerciale) qui fonctionne pour les enveloppes qui ont une petite fenêtre transparente sur la gauche. Le texte bleu clair dans le modèle sera invisible dans le document final en sortie et n'est là que pour aider afin de remplir le texte.

Générer le document final

LyX vous cache la compilation en interne de L^ATEX. Il crée un répertoire temporaire dans `/tmp`, ce que vous pouvez voir comme ceci :

```
ls -l /tmp/lyx_tmpdir*/lyx_tmpbuf[0-9]/*
```

Il est utile de le savoir lorsqu'on exporte en HTML. Dans tout les autres formats (**PS, PDF, Ascii, L^ATEX, DVI**), LyX crée simplement le fichier désiré dans votre répertoire courant. Avec le HTML, il copie la page HTML mais pas les images ni les fichiers de style qui y sont liés et générés par `latex2html` (ceux-ci peuvent simplement être copiés depuis le répertoire de compilation temporaire). Pour finir, produire le document final ne nécessite même pas d'interface graphique, **lyx(1)** possède nombre d'options en ligne commande comme, par exemple, *générer un fichier PDF* à partir d'un fichier source donné au travers de la commande suivante :

```
lyx -e pdf my_lyx_source_file.lyx
```

ce qui produit un nouveau fichier. J'utilise cette commande pour un grand nombre de dépôts de documentations où les fichiers PS et PDF sont générés automatiquement par un script depuis les fichiers d'entrée de Lyx.

Trucs et astuces

Erreurs, le préambule et les liens hypertextes



Si la compilation LaTeX échoue, une boîte d'erreur comme celle sur la gauche s'affiche. Généralement, *la localisation d'erreur dépendante du contexte* est assez bonne et vous pouvez ignorer l'erreur qui est notifiée dans votre document par une boîte d'erreur rouge via **ALT-g**. Si cela ne marche pas, essayer de voir la sortie stderr output (dans le cas où vous avez démarré LyX depuis un console) ou essayez **Visualiser -> Journal de Bord LaTeX**. Parfois, vous pourriez rencontrer de petits problèmes car un certain *style*

de document n'est pas inclus (par ex. '*paquet xyz non trouvé*'). Cela peut facilement être résolu en l'ajoutant dans le préambule Latex qui a le même usage pour les documents Latex que pour la section <principale> qui sert pour le HTML, par ex. la déclaration des fichiers de style, les inclusions, etc. Pour atteindre le préambule, sélectionnez **Format -> Document -> Préambule**. Par exemple, vous pouvez ajouter

```
\usepackage{url}           %% pour les URLs (également dans Bibtex)
\usepackage{ae,aecompl}   %% génère de meilleurs fichiers de sortie PDF
(polices de meilleures qualité)
\usepackage{hyperref}     %% le meilleur pour la fin, génère des PDF avec des
liens hypertextes
```

Le dernier, hyperref, vaut vraiment la peine de s'y attarder. Vous connaissez ces documents PDF où il existe des liens cliquables, colorés dans leur contenu et une section marque-pages où vous pouvez choisir la section vers laquelle vous pouvez vous rendre. De tels documents sont créés en utilisant le paquet hyperref. Afin d'être sûr de l'ajouter, mettez-le à la fin de la liste vu qu'il supprime les paramètres des autres paquets.

Chemins utiles

Mis à part de faire des *sauvegardes d'urgence* et des fichiers de sauvegarde automatique, Lyx garde une sauvegarde de chaque fichier dans **\$HOME/.lyx/backups**. Si vous supprimez accidentellement un fichier LyX, vous pouvez trouver la dernière version là. C'est également l'endroit où les fichiers de style définis par l'utilisateur sont conservés (**\$HOME/.lyx/layouts**). Si vous créez un ou plusieurs fichiers de style pour vous-même, mettez-les de préférence dans **\$HOME/.lyx/** plutôt que dans **/usr/share/lyx/layouts** étant donné que le contenu du répertoire est effacé à chaque nouvelle installation ou mise à jour des RPM. Je pense que c'est une bonne habitude de faire régulièrement la sauvegarde dans votre répertoire personnel.

Ajouter des styles LATEX

Si vous avez besoin d'ajouter de nouveaux fichiers style pour Lyx (et LaTeX) afin de les utiliser, voici la mise en place en deux étapes. **Premièrement**, le fichier de style doit être placé dans un répertoire où LaTeX peut le retrouver. Généralement, vous pouvez utiliser n'importe quel sous-répertoire de **/usr/share/texmf/tex/latex/**. Afin d'éviter qu'une mise à jour d'un RPM le retire, il est bon de créer un sous-répertoire (par ex. **mes_fichiers_style**) et de le sauvegarder régulièrement. Une fois que le nouveau fichier style réside dans ce répertoire, la *configuration de LaTeX* doit être mise à jour, il suffit juste de lancer :

```
texhash
```

(en tant que root). Si vous êtes curieux, vous pouvez également essayer 'texconfig'. Les fichiers de style requis peuvent presque tous être trouvés sur le [CTAN](#). La **seconde étape** requière de lancer

Editer→**Reconfigurer** sous LyX et de le relancer .

Réaliser votre propre mise en page

C'est parti pour ce tutoriel de soixante secondes consacré à définir un document selon vos goûts. Vu que nous sommes sous Linux, nous pouvons nous exercer aux vertus classiques de la réutilisation de logiciels afin de faire quelque chose de nouveau. Supposons que vous être un touriste professionnel et que vous désirez rédiger un catalogue à propos de vos expériences vacancières intéressantes. Vous aimeriez avoir quelque chose qui commence une nouvelle section avec des mots en gras disant : « Ma passionnante expérience de vacances numéro ... ». LyX s'attend à une description de la mise en page du texte qui lui dise quels paquets Latex utiliser (nous allons ici simplement utiliser le paquet `article`) et ce que vous désirez supporter comme environnement dans votre document. La première chose importante, les deux premières lignes qui doivent être quelque chose comme ceci :

```
## Do not delete the line below; configure depends on this
# \DeclareLaTeXClass[article]{article - à propos de mes vacances }
```

L'argument entre crochets donne le(s) paquet(s) LaTeX à utiliser. Vous pouvez en indiquer plusieurs en utilisant des virgules. Le deuxième argument dit à Lyx comment nommer ce style. Après avoir sauvegardé le fichier dans `$HOME/.lyx/layouts` et lancé **Editer**→**Reconfigurer**, vous pourrez lire la chaîne « `article - à propos de mes vacances` » (ou la façon dont vous avez choisi de le nommer) sous **Format** → **Document** → **Format** → **Classe de document**. Mais nous n'avons pas encore terminé. D'abord, nous allons simplement importer une mise en page existante appelée `article.layout` (aucun chemin n'est nécessaire. LyX va chercher son répertoire de mise en page ou `$HOME/.lyx/layouts`)

```
Input article.layout
```

Ensuite, nous allons construire une commande LaTeX convenable qui doit aller dans le préambule (entête du document). Afin de mettre quelque chose dans le préambule, nous allons utiliser l'environnement **Préambule**.

```
Preamble
  \newtheorem{holiday}{Mon expérience de vacances la plus géniale Nr. }
EndPreamble
```

Nous terminons en disant à Lyx comment utiliser cette commande

```
Style Mes_Vacances
  Margin First_Dynamic
  LatexType Environment
  LatexName holiday                               ## la commande Latex
  AlignPossible Block, Left
  LabelType Static
  LabelString "Experience de vacance #. "         ## Ce que vous verrez dans le
document
End
```

Après avoir édité ce fichier (a_modified_article.layout), stockez-le dans `$HOME/.lyx/layouts`, reconfigurez-le, relancez et contemplez – vous pouvez dorénavant trouver un environnement appelé « **Mes_Vacances** » dans le sélecteur de style. Ceci est un exemple de document qui utilise le nouveau fichier de style (a_modified_article.layout). Plus d'informations peuvent être trouvées dans [3(voir les références)].

Conclusions

Il est certain que si vous ne devez rédiger que votre liste des courses ou un mémo de deux paragraphes, utiliser Lyx est excessif. Mais si vous cherchez à rédiger des textes de plus de 500 mots avec une certaine structure et des références croisées, vous pouvez vous attendre à découvrir de grandes perspectives avec Lyx.

- [Wiki de LyX](#)
- [Liste de diffusion de LyX](#)
- [Trucs & Astuces](#)

Bibliographie

- [1] "*Introduction à LyX*". Par l'équipe de LyX. Disponible sous **Aide** → **Introduction**.
- [2] "*Outliners and Mind Mapping*". Par Jimmy O'Regan. Linux Gazette n° 108, Novembre 2004.
- [3] "*Personnaliser LyX: Fonctions pour l'utilisateur avancé*". Equipe de LyX. Sous **Aide** → **Personnalisation**.

Site Web maintenu par l'équipe d'édition LinuxFocus © Gerrit Renker "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org
--

Translation information: en --> -- : Gerrit Renker <gerrit.renker(at)gmx.de> en --> fr: Laurent Richard <kouran(at)linuxmail.org>
