

Conseils et astuces pour T_EXShop

v0.4.1–27 mars 2011*

H. Schulz

herbs2@mac.com

Table des matières

1	Introduction	2
1.1	Ce que vous ne trouverez pas ici	2
1.2	Ce qui est ici	2
2	Édition, composition et visualisation — le cycle de travail	2
2.1	Édition du fichier source	2
2.1.1	Palettes L ^A T _E X et Matrice	2
2.1.2	Liste du menu Étiquette	3
2.2	Composition	3
2.3	Affichage de la sortie du fichier pdf	4
2.3.1	Synchronisation entre le pdf et le source	4
2.4	Travail sur un document volumineux	4
2.4.1	Commutation entre fenêtres de source	5
2.5	Citations avec BibDesk	5
2.6	Obtention d’aide pour les extensions	5
3	Contrôle du clavier	5
3.1	Raccourcis de menu et Préférences Système	6
3.2	Plus d’aide à l’édition	6
3.3	Complètement automatique	6
4	Macros	7
4.1	Macros de texte	7
4.2	Macros Applescript	8
5	Complètement de commande	8
5.1	Complètement	8
5.2	Substitutions or abréviations	9
5.3	Hé!, ça ne marche pas!	9
6	Extension du traitement par les moteurs	9
6.1	Moteur pdf _l atexmk	10

*Traduit par René Fritz le 8 avril 2011.

1 Introduction

T_EXShop est une « interface » (*Front End*) pour une distribution T_EX sur Mac OS X. Il permet de créer et d'éditer des fichiers source T_EX, d'interagir avec la distribution T_EX (p. ex., composer le fichier source) et enfin de donner un aperçu du fichier pdf final. Il permet également d'aller et venir entre l'aperçu et le source.

Au fil des ans T_EXShop a ajouté de nombreuses fonctionnalités. Certaines d'entre elles sont claires et sont destinées à aider un novice à ses débuts. D'autres sont un peu plus subtiles dans leur utilisation et leur puissance sous-jacente doit être apprivoisée.

1.1 Ce que vous ne trouverez pas ici

Tout d'abord, cet article ne traite pas de T_EX ou L^AT_EX. Je n'ai pas l'intention de vous apprendre à écrire un source T_EX. Il y a beaucoup de livres et articles bien faits qui vont enseigneront comment devenir un T_EXpert¹ ou, du moins, un T_EXpätzer² comme moi.

Bien qu'il renferme des éléments d'introduction, il n'est cependant pas conçu comme un manuel exhaustif sur T_EXShop pour les nuls. Au fil du temps, il pourrait évoluer vers un tel document, mais je dois commencer quelque part et je n'en suis qu'au début.

1.2 Ce qui est ici

Dans cet article, je souhaite vous présenter quelques-unes des subtilités qui rendront votre vie d'éditeur de source T_EX plus facile. Il s'agit, notamment, d'ajouter des commandes de clavier et d'étendre les capacités d'édition de T_EXShop ; d'alléger votre travail dans la création de documents, etc., en utilisant des macros et le complètement de commande ; et, enfin, de savoir comment on peut étendre les capacités de traitement de T_EXShop grâce aux moteurs.

2 Édition, composition et visualisation — le cycle de travail

Dans ce document, c'est la section la plus proche des débutants.

2.1 Édition du fichier source

La première chose que vous ayez à faire pour créer ce grand ouvrage est de le taper dans le document source qui sera composé et visualisé plus tard. Ceci nécessite d'entrer, à la fois, le balisage L^AT_EX ainsi que vos merveilleux mots dans le document.

Pour démarrer, vous pouvez ouvrir un nouveau document en utilisant Fichier → Nouveau (Cmd-N), puis en remplir le début en choisissant un modèle dans le menu déroulant Modèles situé dans la fenêtre du source ou utiliser — nouveau avec T_EXShop 2.36 — la commande fichier → Nouveau depuis la papeterie... et choisir le source adéquat dans la liste. Notez que la liste des modèles et des articles fournis dans la papeterie est incomplète, si vous en avez dont vous pensez qu'ils seraient très utiles, n'hésitez pas à les soumettre pour inclusion dans T_EXShop. Vous pouvez ajouter vos modèles et sources personnels dans ~/Library/TeXShop/Templates/ et dans ~/Library/TeXShop/Stationery/. **Note : ~/Library/ est le dossier Library dans votre dossier HOME.**

2.1.1 Palettes L^AT_EX et Matrice

Même si je crois que les panneaux avec une interface cliquable font réellement obstacle à l'apprentissage, je vais mentionner que T_EXShop a deux panneaux d'aide à la saisie du code L^AT_EX (la Palette LaTeX...) et à la mise en place de la structure de base d'une matrice ou d'un tableau (la Palette Matrice...). Ils sont activés ou désactivés dans le menu Fenêtre ou avec les raccourcis clavier³ Opt-Cmd - et Opt-Cmd -= respectivement. La FIGURE 1 montre l'aspect de

1. Un T_EXnicien.

2. Un mauvais joueur d'échecs.

3. Les raccourcis sont donnés pour un clavier anglais et peuvent être différents pour d'autres claviers.

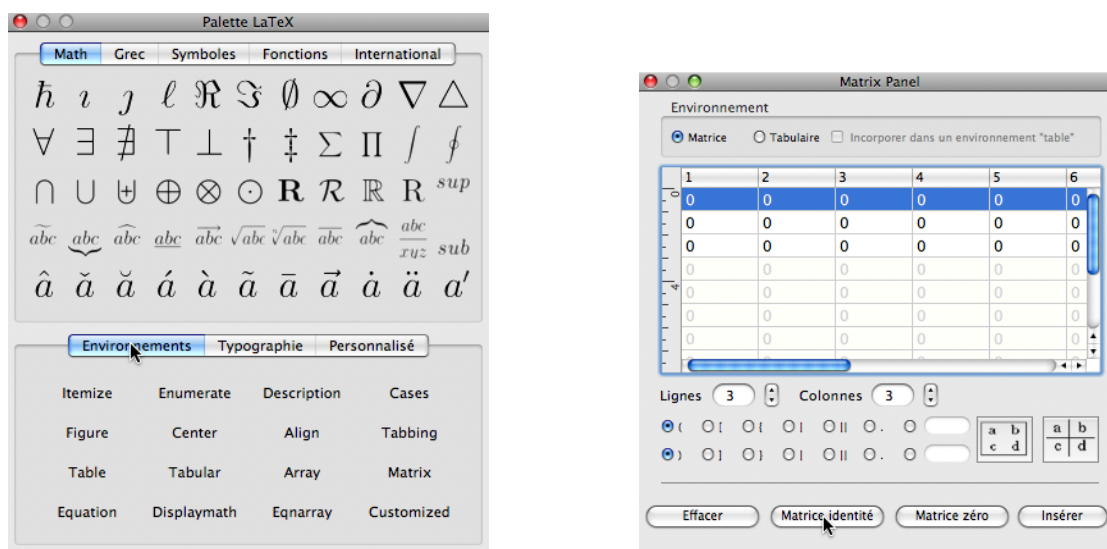


FIGURE 1 – Les panneaux LaTeX et Matrix.

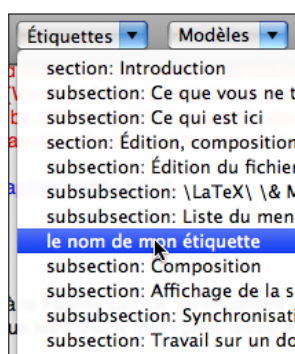


FIGURE 2 – La liste du menu Étiquette.

ces panneaux. Il est possible de faire quelques modifications et ajouts à la palette LaTeX en modifiant le fichier `~/Library/TeXShop/LatexPanel/completion.plist`. **Note : tous les fichiers `plist` doivent être édités en utilisant le codage Unicode UTF-8.**

2.1.2 Liste du menu Étiquette

Le menu déroulant Étiquettes de la barre d'outils du source liste automatiquement les commandes de sectionnement pour vous permettre de passer rapidement à une partie pertinente du source de votre document. Vous pouvez ajouter votre propre étiquette à un endroit particulier de cette liste en plaçant dans le document la ligne

`%;le nom de mon étiquette`

et elle apparaîtra alors à cette position dans la liste déroulante et, ainsi, vous pourrez sauter rapidement à cet endroit. Voir la FIGURE 2. Désolé, les balises ne sont pas incluses de manière récursive pour les fichiers que vous incluez par `\include` ou `\input`.

2.2 Composition

Une fois que vous êtes prêt à jeter un œil sur la façon dont votre document apparaîtra vous composerez avec le moteur défini par défaut, `pdflatex`, en allant simplement dans le menu Composition → Composer (Cmd-T).

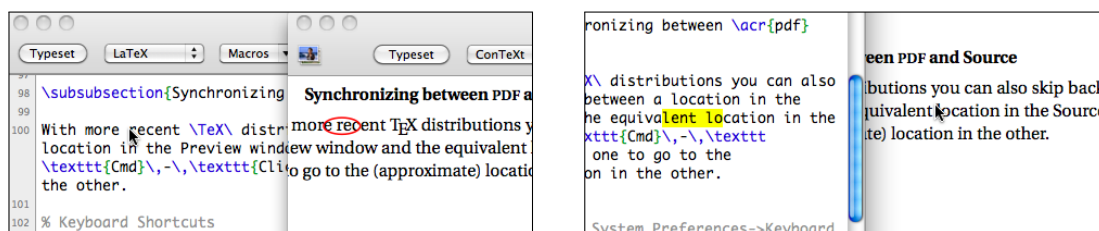


FIGURE 3 – Synchronisation source → aperçu et aperçu → source.

Vous pouvez utiliser un moteur différent de celui défini par défaut. Pour faire ce changement allez dans TeXShop → Préférences... → Composition.

Si vous incluez beaucoup de fichiers graphiques eps dans votre document vous pouvez réaliser la composition en utilisant latex → dvips → ps2pdf car pdf (La)TeX ne permet pas d'inclure directement les fichiers eps⁴. La meilleure façon de le faire est d'entrer la ligne

```
%! TEX TS-program = latex
```

en tête de votre document. Alors, T_EXShop composera avec la méthode latex+distiller, notée ci-dessus, quel que soit le réglage du moteur par défaut. Remplacez latex par pdflatex pour forcer l'utilisation de pdflatex pour composer votre fichier.

2.3 Affichage de la sortie du fichier pdf

En supposant que le document a été composé correctement, le fichier pdf s'ouvre automatiquement dans une fenêtre d'aperçu distincte.

Vous pouvez contrôler la façon dont il est affiché dans le menu Page et modifier les paramètres par défaut dans TeXShop → Préférences → Aperçu du PDF.

2.3.1 Synchronisation entre le pdf et le source

Avec les distributions T_EX les plus récentes, vous pouvez également aller d'un emplacement dans la fenêtre de l'aperçu à l'emplacement correspondant dans la fenêtre du source, et inversement, par la commande clavier Cmd-clic. Voir la FIGURE 3 pour un exemple de synchronisation source → aperçu et aperçu → source.

2.4 Travail sur un document volumineux

Il est souvent utile de diviser un document de grande taille en plusieurs parties subalternes plus faciles à gérer, puis de créer un fichier « principal » qui contient le préambule et des commandes \include pour réunir toutes les parties lors de la composition.

Pour que T_EXShop « sache » quel fichier il doit composer lorsque vous travaillez sur un fichier secondaire entrez la ligne

```
% !TEX root = path/to/rootfile.tex
```

en tête de votre fichier secondaire; path/to/rootfile.tex est le chemin relatif ou absolu vers le fichier principal de ce document. Cela fait, T_EXShop composera le fichier principal si vous appuyez sur Composition → Composer (Cmd-T) même si vous modifiez un fichier subalterne dont le source est correctement synchronisé avec le pdf. P. ex., si le fichier principal est appelé mongrandlivre.tex et que les fichiers de chapitre, chapter1.tex, etc., sont dans un sous-dossier chapitres d'un niveau inférieur au fichier principal, placez la ligne

```
% !TEX root = ../mongrandlivre.tex
```

4. Le programme pdflatex à partir de MacT_EX-2010 peut faire, au vol, cette conversion des fichiers eps.

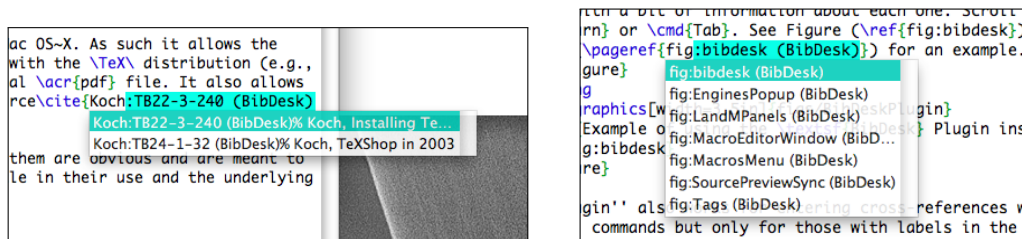


FIGURE 4 – Exemples d’utilisation du plugin de BibDesk pour des citations et des références croisées dans T_EXShop.

en tête de chacun des fichiers de chapitre. Le `../` signifie qu’il faut rechercher le fichier principal dans le dossier de niveau supérieur.

2.4.1 Commutation entre fenêtres de source

Si vous avez plusieurs fichiers source ouverts, vous pouvez basculer uniquement entre ces fichiers en utilisant les commandes du menu fenêtre → Fenêtre source suivante/précédente (Cmd-F2/Maj-Cmd-F2).

2.5 Citations avec BibDesk

T_EXShop est doté d’un « plugin » qui interagit avec l’application bibliographique BibDesk pour vous permettre de compléter les références citées dans la commande `\cite`. Pour autoriser l’utilisation de ce « plugin » assurez-vous que TeXShop → Préférences... → Document → Éditeur → Complétions de BibDesk soit coché.

Pour l’utiliser vous devez d’abord ouvrir les fichiers bibliographiques nécessaires (bib) dans BibDesk. Entrez plusieurs caractères de l’étiquette de la référence dans la commande `\cite` et appuyez sur F5 pour obtenir la liste des références correspondantes à partir du ou des fichier(s) bib avec un peu d’informations sur chacun d’eux. Faites défiler jusqu’à celle que vous voulez et appuyez sur Retour ou Tab. Voir la FIGURE 4 pour un exemple.

Le « plugin » fonctionne également pour la saisie des références croisées dans les commandes `\ref` ou `\pageref`, mais seulement pour celles qui ont des étiquettes dans le fichier que vous éditez.

2.6 Obtention d’aide pour les extensions

Souvent, il peut être utile d’afficher l’aide d’une extension donnée. T_EXShop possède une interface avec `texdoc` qui fera apparaître cette documentation. Cliquez sur Aide → Afficher l’aide pour le package... (Opt-Cmd-I) et entrez le nom de l’extension.

Vous pouvez, tout aussi aisément, regarder la documentation d’une extension avec la commande Aide → Ouvrir le fichier style... et entrer le nom complet du fichier *y compris la bonne extension* (p. ex., `.sty` pour les extensions ou `.cls` pour les classes de document).

3 Contrôle du clavier

Un des meilleurs moyens d’accélérer l’entrée de votre texte dans un fichier source est de garder les mains sur le clavier autant que possible — c’est une des raisons pour laquelle je n’aime pas l’interface « cliquable » des panneaux L^AT_EX et Matrix. Il y a beaucoup de raccourcis associés au système de menu de T_EXShop mais cette section traite de leurs modifications, de l’ajout d’autres raccourcis, ainsi que d’autres personnalisations du clavier.

3.1 Raccourcis de menu et Préférences Système

Il peut arriver que vous souhaitiez créer un raccourci vers un élément de menu qui n'en a pas ou en ajouter un pour une commande dont vous n'aimez pas le raccourci. À partir de Mac OS X 10.4 (Tiger) il existe une méthode pour ajouter des raccourcis vers des éléments de menu spécifiques à la fois au niveau global et dans des programmes spécifiques. Cette fonction est devenue beaucoup plus fiable dans OS X 10.5 et surtout dans OS X 10.6.

Un exemple en utilisant Mac OS X 10.6 (Snow Leopard) : \TeX Shop 2.36 a ajouté une commande fichier → Nouveau à partir de papeterie. . . , sans un raccourci, qui peut être très utile lorsque vous installez la « papeterie » à votre guise. Pour ajouter Opt-Command-N comme raccourci pour cet élément de menu : ouvrez l'application Préférences Système. . . puis Clavier et souris → Raccourcis clavier et sélectionnez Raccourcis clavier de l'application ; appuyez sur le bouton + pour ajouter un raccourci ; sélectionnez \TeX Shop comme application ; entrez le titre exact du menu [Nouveau depuis la papeterie. . . — notez que vous *devez* entrer le caractère véritable des points de suspension, « . . . », (Opt- ; avec un clavier anglais, Opt-. avec un clavier français)] ; dès lors, vous pouvez utiliser Opt-Command-N comme raccourci.

3.2 Plus d'aide à l'édition

\TeX Shop est construit en utilisant les interfaces des programmeurs d'Apple (appelées *frameworks*) et hérite donc de toutes les propriétés et de la fonctionnalité de ces interfaces. Il y a beaucoup de choses disponibles dans le *Text framework* qui ne sont pas liées au clavier par défaut ; p. ex., de nombreuses commandes clavier « de type emacs », mais Apple a permis d'ajouter ces commandes à toutes les applications qui utilisent le *Text framework* ; p. ex., TextEdit et Mail ainsi que \TeX Shop.

Ceci se fait en créant un fichier spécial, `DefaultKeyBinding.dict`, et en le plaçant dans un endroit particulier, `~/Library/KeyBindings/` (vous aurez à créer le dossier `KeyBindings` à cet endroit s'il n'existe pas déjà).

Vous obtiendrez de plus amples renseignements à ce sujet, ainsi qu'un exemple (utile), en téléchargeant le fichier `KeyBindings.zip` depuis <http://public.me.com/herbs2>.

3.3 Complètement automatique

Outre l'ajout de raccourcis pour des éléments de menu, vous pouvez, dans \TeX Shop, réellement lier les frappes de touche afin de les développer dans des groupes de caractères. C'est ce qu'on appelle l'achèvement automatique (*Auto Completion*) qui est activé en cochant l'option suivante \TeX Shop → Préférences. . . → Document → Éditeur → Compléter automatiquement. Ne pas confondre avec le complètement de commande dont il est question dans la section 5 page 8.

P. ex., avec un clavier américain, le raccourci Opt-, entre généralement ≤ dans votre document, mais si le complètement automatique est activé, il entrera \leq. De même, si vous sélectionnez du texte, en appuyant sur " il sera entouré par ' ' et ' '.

Vous pouvez ajouter vos propres raccourcis clavier ou supprimer ceux dont vous n'avez pas besoin en éditant `~/Library/TeXShop/Keyboard/autocomplete.plist`. Pour ajouter un raccourci à la touche _ de sorte qu'en la pressant lors d'une sélection elle produise la sélection entouré par _ { et } entrer les lignes

```
<key>_</key>
<string>_{#SEL##INS#}</string>
```

à la liste `autocomplete.plist`.

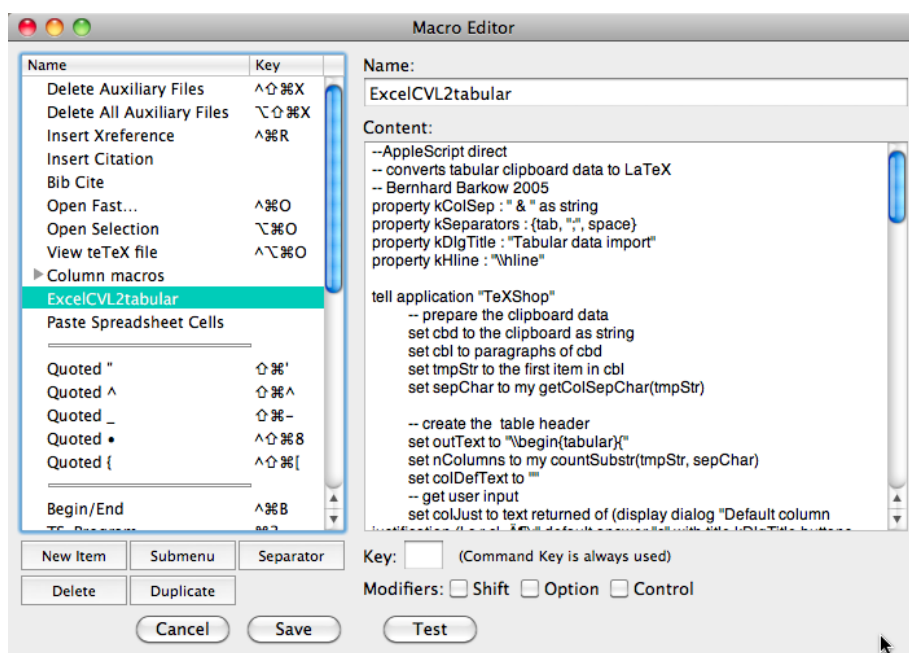


FIGURE 5 – Fenêtre de l'éditeur de macros.

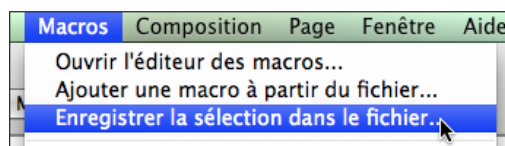


FIGURE 6 – Éléments de menu supplémentaires lorsque l'éditeur de macros est ouvert.

4 Macros

Les macros peuvent être de simples substitutions de texte ou des programmes Applescript qui peuvent faire toutes sortes de traitement sur un fichier. Vous pouvez aussi assigner un raccourci clavier à une macro pour une exécution immédiate. Celles qui font partie de \TeX Shop se trouvent sous le menu Macros.

Vous pouvez supprimer ou ajouter des macros supplémentaires au menu à l'aide du Macro Editor (utilisez la commande Macros → Ouvrir l'éditeur de macros. . .). La fenêtre du Macro Editor et des éléments supplémentaires dans le menu Macros sont respectivement présentés dans les FIGURES 5 et 6.

Outre l'écriture de vos propres macros, vous pouvez ajouter des macros fournies par des tiers dans le menu Macros de deux façons : soit copier et coller la version texte de la macro dans un Nouvel item du Macro Editor, soit obtenir la macro en tant que fichier `plist` et l'utiliser avec la commande Ajouter une macro à partir du fichier. . . trouvée dans le menu Macros lorsque le Macro Editor est ouvert (encore une fois, voir la FIGURE 6).

Pour plus d'informations sur les macros rechercher dans le menu Aide → Aide de \TeX Shop. . .

4.1 Macros de texte

Les macros de texte sont de simples substitutions de texte. Vous pouvez aussi dire à \TeX Shop d'insérer n'importe quel texte sélectionné en utilisant `##SEL`, de placer le curseur à l'aide `##INS` et même de mettre la macro elle-même sur plusieurs lignes. Ensuite, vous pouvez

affecter la macro de texte à un raccourci clavier.

J'aime utiliser Cmd-B et Cmd-I pour insérer `\textbf{...}` et `\emph{...}` dans le document où ... est n'importe quel texte sélectionné. Les macros pour faire cela sont déjà dans le menu Macros → Text Styles ; donc, il ne nous reste plus qu'à leur attribuer des raccourcis clavier. Pour attribuer Cmd-I à la macro `emphasize` : ouvrir le Macro Editor où la forme du menu Macros apparaît dans le volet de gauche, cliquez sur la macro `emphasize` qui se trouve sous Text Styles ; cliquez sur la case d'insertion de la clé (Touche) où il suffit d'insérer un « i » minuscule (la touche cmd est toujours prise en charge et des touches de modification supplémentaires peuvent être cochées).

4.2 Macros Applescript

Dans le menu Macros, il n'est pas possible de distinguer les macros Applescript des macros de texte, pourtant elles peuvent faire des traitements compliqués et ajouter ou modifier le fichier source dans TeXShop. P. ex., la macro Program, définie par défaut, crée la ligne suivante

```
%! TEX TS-program = xxxx
```

en tête d'un fichier avec le moteur de votre choix à la place de xxxx. Vous pouvez voir le code Applescript de cette macro en cliquant sur son nom dans l'éditeur de macros (Macro Editor).

5 Complètement de commande

Le balisage de L^AT_EX est plutôt prolix ce qui est agréable, car il décrit ce qu'il est censé faire, mais un peu pénible à écrire. Le complètement de commande vous permet d'insérer des environnements complets et des commandes en quelques frappes suivies d'une ou plusieurs pressions sur la touche de déclenchement (qui est, par défaut, Esc mais peut être remplacée par Tab en allant dans TeXShop → Préférences... → Document → Commande de complétion déclenchée par).

Les commandes avec arguments ont généralement un repère (•) placé à leur niveau. Pour aller à l'argument suivant, utilisez la commande Source → Complètement → Repères → Repère suivant (Ctl-Cmd-F [ou Opt-Esc/Tab]). Ceci sélectionne en même temps le repère pour qu'il soit automatiquement remplacé par l'information saisie. Voir la documentation complète, avec les listes de commandes et abréviations, fournie avec TeXShop dans le dossier ~/Library/TeXShop/CommandCompletion/, pour beaucoup plus d'informations.

5.1 Complètement

Vous pouvez compléter de nombreuses commandes en commençant à les saisir puis en appuyant sur la touche de déclenchement. Les commandes qui diffèrent par leur nombre d'arguments optionnels sont générées par des pressions successives sur la touche de déclenchement. Ainsi, taper `\sec` puis la touche de déclenchement, sur une nouvelle ligne, produit

```
\section{█}
```

tandis qu'une seconde pression sur la touche de déclenchement donne

```
\section*{█}
```

la variante étoilée (*) et qu'une dernière pression de cette touche donne la commande

```
\section[█]{•}
```

avec l'argument optionnel.

5.2 Substitutions or abréviations

Outre le complètement des commandes entrées partiellement, il y a aussi de nombreuses abréviations. Celles-ci sont de courts mnémoniques pour accéder à des substitutions complètes.

Toutes les abréviations pour les environnements commencent par un « b ». Pour obtenir l'environnement `itemize` complet, entrez `\bitem` sur une nouvelle ligne et appuyez sur la touche de déclenchement ce qui produira

```
\begin{itemize}
\item
•
\end{itemize}•
```

terminé par un repère pour que vous puissiez facilement accéder à la fin de l'environnement. Des items supplémentaires peuvent être créés en tapant `\it` puis la touche de déclenchement afin d'obtenir

```
\item
•
```

prêt pour l'entrée d'un texte.

En outre, les commandes de sous-sectionnement possèdent leurs abréviations. Elles peuvent être obtenues en tapant `\sssec` puis en pressant la touche de déclenchement ce qui donnera

```
\subsection{•}
```

des pressions successives sur la touche de déclenchement donneront, la variante étoilée (*) et, enfin, la variante avec l'argument optionnel.

En dernier exemple, l'entrée de `\tt` suivi d'une pression sur la touche de déclenchement donnera la commande `\texttt{•}`; une seconde pression de cette la touche donnera la déclaration `\ttfamily` : des résultats similaires sont obtenus avec les autres commandes de changement de style de police.

5.3 Hé!, ça ne marche pas!

Si ces exemples ne fonctionnent pas, vous devrez probablement permettre à `TEXShop` de mettre à jour le dossier `~/Library/TeXShop/CommandCompletion/`; supprimez purement et simplement ce dossier de `~/Library/TeXShop/` et redémarrez `TEXShop`.

6 Extension du traitement par les moteurs

`TEXShop` propose plusieurs « moteurs » par défaut (aussi appelée « scripts », ce qui n'a pas changé depuis l'origine) dans son menu `Composition`. Ils comprennent, notamment, `Plain TeX` ou `LaTeX` (soit en utilisant `pdftex` ou `TeX+DVI`), `BibTeX`, `MakeIndex`, `MetaPost` ou `ConTeXt`. Mais, dans ce que vous souhaitez faire, beaucoup de choses ne relèveront pas de cette offre limitée; si bien que `TEXShop` vous permet également d'installer de nouveaux moteurs qui sont stockés dans `~/Library/TeXShop/Engines/`. Ces moteurs supplémentaires n'apparaissent pas dans le menu `Composition`, mais seulement dans la liste déroulante de la barre d'outils du source (voir la FIGURE 7).

Vous pouvez utiliser ces moteurs en les sélectionnant dans cette liste déroulante, puis en appuyant sur le bouton `Composition` ou mieux, si vous utilisez différents moteurs pour différents documents, en mettant une ligne comme

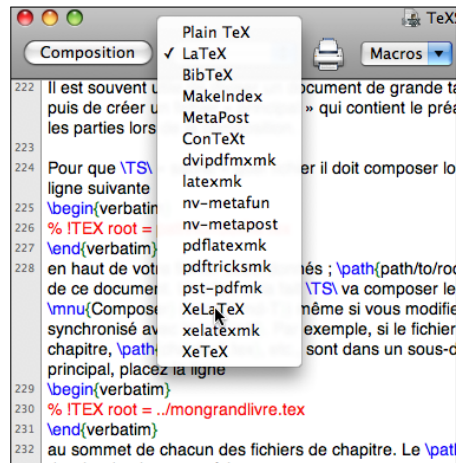


FIGURE 7 – Menu déroulant des moteurs dans la barre d’outils du source.

```
% !TEX TS-program = xelatex
```

en tête de votre fichier source ; dans cet exemple le moteur `xelatex` sera utilisé pour ce fichier indépendamment des autres choix.

D’origine, seuls quelques moteurs sont activés dans `TeXShop` (c.-à-d., directement dans le dossier `~/Library/TeXShop/Engines/`), mais il y a aussi d’autres moteurs dans le dossier `~/Library/TeXShop/Engines/Inactive/`. À titre d’exemple, nous allons activer et utiliser le moteur `pdflatexmk` trouvé dans `~/Library/TeXShop/Engines/Inactive/Latexmk/`.

6.1 Moteur `pdflatexmk`

Si votre document comportait des références croisées, des bibliographies ou des index, il faudrait plusieurs compilations avec `pdflatex`, mais aussi d’autres encore avec `bibtex` ou `makeindex`, pour créer les bibliographies, les index et résoudre toutes les références croisées. Le moteur `pdflatexmk` automatise ce processus.

Pour activer ce moteur il suffit de déplacer le fichier `pdflatexmk.engine` du répertoire `~/Library/TeXShop/Engines/Inactive/Latexmk/` pour le mettre, deux dossiers en amont, dans `~/Library/TeXShop/Engines/`. Au redémarrage de `TeXShop` vous pourrez vérifier que `pdflatexmk` figure maintenant dans le menu déroulant.

Il ne reste plus qu’à entrer la ligne

```
% !TEX TS-program = pdflatexmk
```

en tête de votre fichier source. Dès cet instant, il suffit de composer le fichier (Composition → Composer or Cmd-T) pour que `TeXShop` utilise ce moteur et que le processus complet de mise en page du document s’effectue jusqu’à sa forme finale.