# HOWTO-HOWTO


## Mark F. Komarinski



Ce document décrit les outils et procédures, et donne des conseils aux auteurs de HOWTO. (Adaptation française par Nicolas Duboc)


## Table of Contents

# 1. A propos de ce HOWTO

## 1.1. Objectif / Contenu de ce HOWTO

Ce document a été commencé le 26 Août 1999 par Mark F. Komarinski après deux jours de frustration à essayer de faire marcher les outils. Ne se trouverait-il qu'un auteur du LDP (NdT : "Linux Documentation Project", Projet de Documentation Linux) pour trouver de l'aide dans ce document, j'aurais atteint mon objectif.

La version la plus récente de ce document peut être trouvée sur ma page personnelle http://www.cgipc.com/~markk/ au format SGML. D'autres versions peuvent être trouvées dans différents formats sur le site du LDP http://www.linuxdoc.org/.

Il existe de nombreuses manières de contribuer au mouvement Linux sans réellement écrire de code. Une des plus importantes est d'écrire de la documentation, permettant à chaque personne de partager ses connaissances avec des milliers d'autres autour du monde. Ce HOWTO a été créé pour vous familiariser avec les rouages du LDP, et avec les outils dont vous aurez besoin pour écrire votre propre HOWTO.

## 1.2. A propos du LDP

Ce qui suit est un extrait du Manifeste du LDP (http://www.linuxdoc.org/manifesto.html)

Le Projet de Documentation Linux (LDP) travaille sur le développement libre et de haute qualité d'une documentation pour le système d'exploitation GNU/Linux. Le but premier du LDP est de collaborer dans tous les thèmes de documentation pour Linux. Cela inclut la création de "HOWTOs" et de "Guides". Nous espérons établir un système de documentation pour Linux simple d'utilisation et facilitant les recherches. Ce système implique l'intégration des pages de manuel, des docs infos, des HOWTOs, et d'autres documents.

Vous pouvez trouver plus d'informations à propos du Projet de Documentation Linux à l'adresse http://www.linuxdoc.org/

## 1.3. Retour de commentaires

Les commentaires sur ce HOWTO peuvent être envoyés à l'auteur (`<markk@linuxdoc.org>`). (NdT : quant aux commentaires sur la traduction : `<duboc@essi.fr>`)

## 1.4. Copyrights

(c) 1999-2000 Mark F. Komarinski

Ce manuel peut être reproduit en totalité ou en partie, sans frais, sous réserve des restrictions suivantes :

- Cette note de copyright et de permission doit être préservée dans toutes copies partielles ou totales.
- Toutes traductions ou travaux dérivés doivent être approuvés par l'auteur en le prévenant avant leur distribution.
- Si vous distribuez une partie de ce travail, les instructions pour obtenir la version complète devront également être fournies.
- De courts extraits peuvent être reproduits, sans ces notes de permission, dans le cadre d'exposés et de citations si les références sont correctement citées.

Des exceptions à ces règles peuvent être tolérées pour un but éducatif : contactez l'auteur et demandez-lui. Ces restrictions sont là pour nous protéger en tant qu'auteurs, et non pour vous restreindre en tant que lecteurs ou enseignants. Tous les codes sources apparaissant dans ce document sont protégés par la Licence Publique Générale GNU, disponible par FTP anonyme depuis les archives GNU.

### 1.5. Contributions et remerciements

Merci à tous ceux qui ont donné leurs commentaires lorsque j'écrivais ceci. Cela inclut David Lawyer, Deb Richardson, Daniel Barlow, Greg Ferguson, Mark Craig et les autres membres de la liste de diffusion `<ldp-discuss@lists.linuxdoc.org>`. J'ai tiré quelques sections du HOWTO Index (http://www.linuxdoc.org/HOWTO/) (disponible sur tous les sites du LDP) et de la documentation de sgmltools. Les sections sur l'accès réseau à CVS ont été partiellement écrites par Serek (`<ser@serek.arch.pwr.wroc.pl>`). Les sections sur DocBook ont été écrites par Jorge Godoy (`<godoy@conectiva.com.br>`). Un grand merci à tous les deux pour leur aide.

### 1.6. Conventions

Les commandes qui sont listées ont le format suivant. Les commandes sont précédées par le nom du shell utilisé. Suit un $ pour les commandes qui devraient être lancées par un utilisateur normal (pas le super-utilisateur). Si le nom du shell est suivi par un #, cela signifie que la commande doit être lancée par le super-utilisateur.

## 2. Informations générales sur le LDP et SGML

### 2.1. Le LDP

Le Projet de Documentation de Linux (LDP) a été créé pour fournir aux nouveaux utilisateurs un moyen d'obtenir rapidement des informations sur un sujet particulier. Il contient non seulement une liste de livres sur l'administration système, le réseau ou la programmation, mais également un grand nombre de petits travaux sur des sujets plus particuliers, écrits par ceux qui les ont utilisés. Si vous cherchez des informations sur les impressions, vous trouverez le Printing-HOWTO (http://www.linuxdoc.org/HOWTO/Printing-HOWTO.html). Si vous désirez savoir si votre carte Ethernet fonctionne sous Linux, récupérez le Ethernet-HOWTO (http://www.linuxdoc.org/HOWTO/Ethernet-HOWTO.html), etc.

Au départ, la plupart de ces documents étaient au format texte ou HTML. Avec le temps, il fallait trouver un meilleur moyen de gérer ces travaux. Celui-ci permettrait de les lire depuis une page Web, depuis un fichier texte sur un CD-ROM, ou encore depuis votre organiseur de poche. La solution s'est avérée être SGML.

### 2.2. SGML

Le "Standard Generalized Markup Language" (SGML) est un langage qui est basé sur le balisage du texte (de ce point de vue, il est similaire à HTML, mais les similitudes s'arrêtent là). La force de SGML est que contrairement à la philosophie WYSIWYG (What You See Is What You Get, NdT : Ce que vous voyez est ce que vous obtenez), vous ne définissez pas la couleur, la taille de la police, ou d'autres paramètres de formatage. Au lieu de cela, vous définissez des éléments (paragraphes, sections, listes numérotées) et laissez l'interprète SGML et le programme final s'occuper du placement, des couleurs, des polices, et de tout le reste. HTML fait la même chose, puisqu'il est en fait une subdivision de SGML. SGML est en réalité constitué de trois parties. La première est la Structure, qui est appelée la DTD, ou encore Définition du Type du Document. La DTD définit les relations entre chacun des éléments d'un texte. La DTD DocBook, utilisée pour créer le présent document, en est un exemple. La DTD liste les règles que le contenu doit respecter. La seconde partie est DSSSL ou encore "Document Style Semantics and Specification Language" (NdT : Sémantique de Présentation de Documents et Langage de Spécifications). DSSSL indique au programme chargé du rendu final comment convertir le code SGML en quelque chose de lisible par Monsieur-tout-le-monde. Il demande par exemple, de convertir une balise <table> en une fonte grasse 14 points si le format de destination est RTF, ou en une balise <h1> si on veut du HTML. Enfin, la troisième partie est le Contenu, qui est traité par l'interpréteur SGML et qui est finalement visualisé par l'utilisateur. Le présent paragraphe est un contenu, comme pourrait l'être une image, une table, une liste numérotée ou autre chose. Le contenu est entouré de balises pour séparer les différents éléments.

## 2.3. Pourquoi SGML plutôt que HTML ou d'autres formats ?

SGML permet plus qu'une simple mise en forme. Vous pouvez créer automatiquement des index, des tables des matières ou des liens internes ou externes. Les paquetages Jade et OpenJade vous permettent aussi d'exporter (j'appellerai cela générer à partir de maintenant) vos documents SGML vers LaTeX, info, du texte, HTML ou RTF. Vous pouvez alors créer, à partir de ces formats de base, des fichiers d'autres types comme MS Word, PostScript, PDF, etc. Des programmes comme LyX vous permettent d'écrire au format TeX, d'exporter au format SGML et de générer ce que vous voulez depuis SGML. En fin de compte, SGML s'intéresse plus au rôle des éléments dans le document plutôt qu'à leur apparence. Une différence de taille, en tout cas qui vous permettra d'écrire plus vite, puisque vous n'avez plus à vous soucier de la justification des paragraphes, des types et tailles des polices, etc.

## 2.4. Pour les nouveaux auteurs

Si vous êtes un nouvel auteur au sein du LDP, et que vous voulez prendre en main un HOWTO (ou un Mini-HOWTO) non maintenu ou en écrire un vous-même, contactez le coordinateur du LDP à l'adresse `<ldp-submit@lists.linuxdoc.org>`. Cela lui permet de savoir qui travaille sur quel document.

Une fois cela effectué, vous pouvez écrire votre documentation dans le format que vous voulez et soumettre un brouillon à `<ldp-submit@lists.linuxdoc.org>` pour qu'il soit relu par un volontaire du LDP. Quelques jours plus tard vous recevrez les corrections et commentaires du relecteur. Après avoir appliqué les modifications suggérées, vous pouvez envoyer cette version à la liste ldp-submit une nouvelle fois pour la soumission finale au LDP.

A partir de là, un autre volontaire du LDP traduira votre document en DocBook et vous renverra cette version finale. Maintenant, toutes les soumissions au LDP doivent être au format DocBook. Si vous avez des questions sur les balises, vous pouvez toujours demander aux volontaires qui vous ont aidé, ou directement à la liste DocBook du LDP.

## 2.5. Les listes de discussion

Il y a quelques listes de discussion auxquelles vous pouvez vous abonner pour prendre part au fonctionnement du LDP. La première est `<ldp-discuss@lists.linuxdoc.org>`, qui est le principal lieu de discussion du LDP. Pour s'abonner, il suffit d'envoyer un message avec "subscribe" comme sujet à l'adresse `<ldp-discuss-request@lists.linuxdoc.org>`. Pour se désabonner, même adresse avec "unsubscribe" comme sujet du message.

Une autre liste est la liste `<ldp-docbook@lists.linuxdoc.org>`, qui est prévue pour les questions sur DocBook. Si vous rencontrez des problèmes avec une balise particulière, vous pouvez y envoyer une question. Vous pouvez vous abonner à la liste DocBook en envoyant un message "subscribe" à `<ldp-docbook-request@lists.linuxdoc.org>`.

# 3. Les outils

Dans cette section, nous allons survoler les outils dont vous aurez besoin ou que vous voudrez utiliser pour créer votre propre documentation LDP. Nous allons les décrire ici, et mieux les explorer plus tard, en même temps qu'expliquer leur procédure d'installation. Si vous utilisez d'autres outils pour écrire des manuels du LDP, faites-le moi savoir, j'ajouterai un descriptif ici.

## 3.1. DSSSL

La version de Norman Walsh est nécessaire, la version du LDP est optionnelle.

### 3.1.1. DSSSL de Norman Walsh

http://nwalsh.com/docbook/dsssl/db152.zip

NdT : ce site de Norman Walsh sur DocBook (http://nwalsh.com/docbook/) est une mine d'informations. On y trouve maintenant une version de DSSSL plus récente que celle proposée par l'auteur.

La Sémantique et le Langage de Spécification de Style (DSSSL) indique à Jade comment générer un document imprimable ou visualisable en ligne à partir d'une version SGML. DSSSL est ce qui convertit une balise de titre en une balise <H1> en HTML, ou en une police Times Roman gras 14 points pour RTF par exemple. La documentation sur DSSSL se situe à l'adresse http://nwalsh.com/docbook/dsssl/db152d.zip. Notez que modifier le DSSSL ne modifie en rien DocBook lui-même. Cela ne change que la manière dont sera rendu le texte. Le LDP lui-même utilise un DSSSL modifié qui ajoute une table des matières.

### 3.1.2. LDP DSSSL

http://metalab.unc.edu/gferg/ldp/ldp.dsl

Le DSSSL du LDP requière la version de Norman Walsh (voir plus haut) mais est en fait une version du DSSSL légèrement modifiée pour fournir quelques fonctionnalités supplémentaires telles qu'une table des matières.

## 3.2. La DTD DocBook (version 3.1)

Nécessaire - http://www.oasis-open.org/docbook/sgml/3.1/docbk31.zip

La DTD DocBook définit les balises et les structures du document SGML DocBook. Modifier la DTD, par exemple ajouter des balises, donne une DTD qui n'est plus DocBook.

## 3.3. Jade

Jade et OpenJade sont deux programmes qui réalisent le travail de conversion et de validation du code en utilisant la DTD et DSSSL. Un de ces deux programmes est nécessaire et doit être installé après la DTD et DSSSL.

### 3.3.1. Jade

ftp://ftp.jclark.com/pub/jade/jade-1.2.1.tar.gz

Jade est le programme qui se charge du traitement du document SGML. Jade utilise DSSSL et la DTD DocBook pour effectuer la vérification du texte SGML et sa conversion vers le format choisi.

### 3.3.2. OpenJade

http://openjade.sourceforge.net/

Une évolution de Jade écrite par la communauté DSSSL. Certaines applications exigent Jade, mais elles sont/seront mises à jour pour accepter les deux logiciels.

## 3.4. Interfaces ("Front-ends") pour Jade

Ces outils sont optionnels et peuvent être installés après Jade, DSSSL et la DTD.

### 3.4.1. sgmltools-lite

http://sgmltools-lite.sourceforge.net/

C'est le successeur du projet sgmltools, qui a officiellement été arrêté il y a à peu près un an. Depuis, Cees de Groot a créé un projet légèrement différent qui consiste en une interface utilisateur qui enveloppe Jade. Il cache la plupart des lourdeurs de syntaxe de Jade. L'auteur du présent HOWTO a réussi à installer l'ancien paquetage sgmltools suivi de sgmltools-lite et a pu formater ce document assez facilement. Il y a une page de manuel expliquant la syntaxe de sgmltools.

### 3.4.2. Les outils DocBook de Cygnus

Peut-être spécifique à RedHat - http://www.redhat.com/

RedHat distribue trois paquetages, depuis la version 6.2, qui offrent un support pour DocBook et fournissent plusieurs outils. Ces derniers s'installent facilement, vous permettant ainsi de vous concentrer sur votre document plutôt que vous battre avec les procédures d'installation. Il faut au préalable avoir installé TeTeX, Jade et JadeTeX. Ces trois paquetages sont disponibles sur le CD.

## 3.5. Editeurs

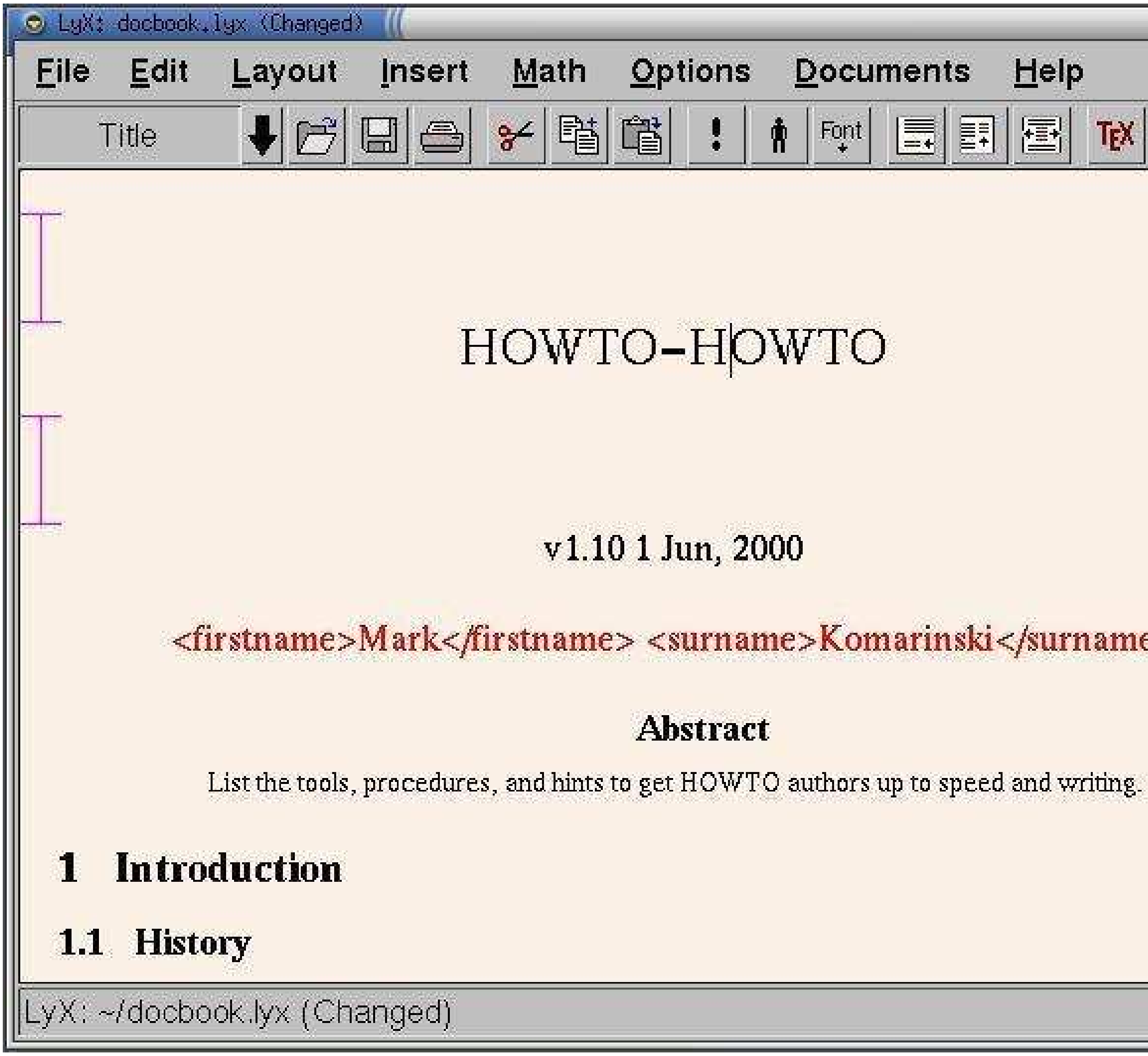
Les outils suivants peuvent être utilisés pour créer, éditer et valider votre HOWTO.

### 3.5.1. LyX

http://www.lyx.org/

LyX vous permet d'écrire du SGML avec la facilité d'un traitement de texte standard. Ce n'est pas un programme WYSIWYG (NdT : Ce que vous voyez est ce que vous obtenez), mais plutôt WYSIWYM (NdT :Ce que vous voyez est ce que vous voulez dire), puisque ce que vous voyez à l'écran n'est pas forcément ce que vous obtiendrez une fois que le processeur SGML aura fait son travail. L'affichage que fournit LyX est proche, mais n'est pas exactement ce que Jade vous donnera en sortie. Néanmoins, cela est suffisant pour se rendre compte de la structure du document. Les sections et sous-sections sont numérotées et mises en caractères gras, et des polices différentes sont utilisées pour représenter les balises comme <code> ou <URL>. La plupart des balises vous sont cachées pendant que vous éditez votre texte, puisque LyX génère des documents en TeX, avant de les convertir en SGML.

**Figure 1. Capture d'écran de LyX**




### 3.5.2. Emacs (PSGML)

Optionnel - http://www.lysator.liu.se/~lenst/about_psgml/

Emacs dispose d'un mode d'édition SGML appelé psgml qui est un mode majeur utilisé pour les documents SGML et XML. Il permet une coloration syntaxique et un affichage agréable qui font ressortir les balises SGML. Il fournit un moyen d'insérer les balises sans avoir à les taper à la main, et est capable de valider la syntaxe de votre document lors de sa rédaction.

Pour les utilisateurs d'Emacs, c'est un excellent outil, et beaucoup pensent qu'il permet une plus grande polyvalence que tout autre éditeur de code SGML. Il fonctionne aussi bien avec DocBook et LinuxDoc que d'autres DTD.

### 3.5.3. VIM

http://www.vim.org

On ne peut pas parler d'Emacs sans parler de vi. L'éditeur VIM (Vi IMproved) dispose de toutes les fonctions du vi classique, mais aussi d'un mode SGML qui utilise la coloration syntaxique pour faire ressortir les balises.

### 3.5.4. WordPerfect 9 (Corel Office 2000)

http://www.corel.com/

WordPerfect 9 pour Windows supporte SGML et DocBook 3.0. WordPerfect 9 pour Linux n'a pas ces fonctions SGML.

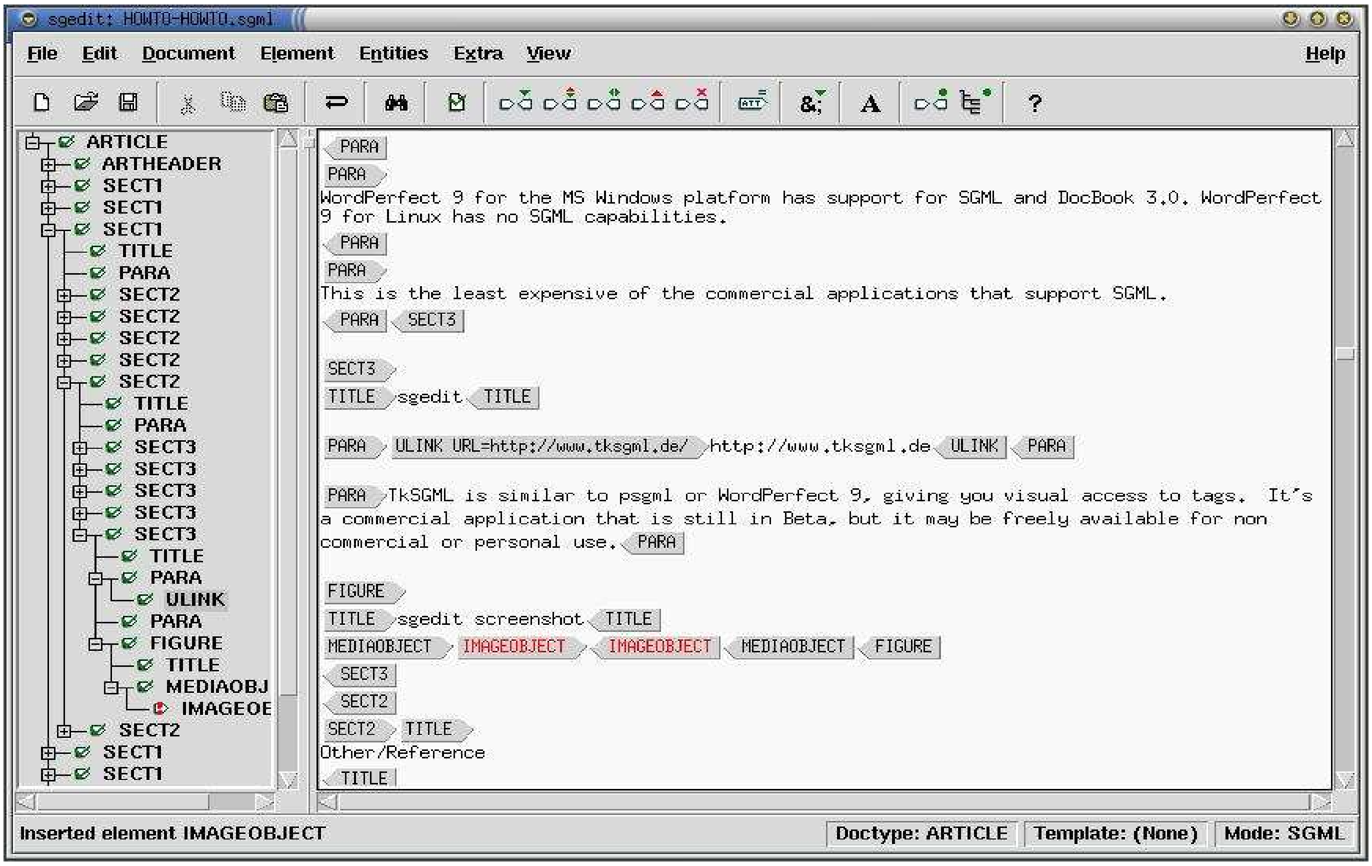
C'est la moins chère des applications commerciales qui supportent SGML.

### 3.5.5. sgedit

http://www.tksgml.de/

Le programme sgedit vous permet d'éditer visuellement vos fichiers SGML. Vous n'avez pas besoin de connaître Emacs ou vi pour l'utiliser. De plus, il fonctionne sous diverses plate-formes, dont Windows et Linux. C'est une application commerciale, mais le prix n'a pas encore été fixé. Il y aura des licences gratuites pour une utilisation privée ou dans un cadre éducatif.

En plus d'une édition visuelle, sgedit peut vérifier la syntaxe de votre document lors du chargement ainsi qu'à la demande avec la commande Document⟶Validate.




## 3.6. Autres outils / Références

Les éléments de cette section sont des livres de référence ou d'autres utilitaires qui sont difficilement classables (pour l'instant).

### 3.6.1. DocBook: The Definitive Guide

http://www.docbook.org/

Ce livre est publié par O'Reilly depuis Octobre 1999. C'est une très bonne référence sur DocBook. Je n'ai pas trouvé qu'il soit vraiment pratique, de plus il est très orienté vers XML, mais les balises DocBook 3.1 sont toutes listées dans un format commode. Vous pourrez le trouver chez votre libraire favori. Une version en ligne est également disponible (aux formats HTML et SGML) à l'adresse précédente.

### 3.6.2. Aspell

Optionnel - http://aspell.sourceforge.net/

C'est un correcteur d'orthographe qui sait reconnaître les balises SGML et qui ne vérifie que le contenu du document. Les correcteurs standards comme ispell essaieraient de corriger les balises, provoquant une erreur à chacune de celles-ci.

# 4. Démarrer avec DocBook

Cette section couvre les nouvelles méthodes pour écrire de la documentation pour le LDP, en utilisant la DTD DocBook 3.1. Nous verrons comment récupérer, installer et utiliser les outils, ainsi qu'une introduction aux balises DocBook. Etant donné qu'il y a près de 300 balises DocBook, nous ne les verrons pas toutes ici. Les lecteurs vraiment intéressés pourront se tourner vers http://www.docbook.org (http://www.docbook.org/) pour plus d'informations.

## 4.1. Télécharger et installer les outils

### 4.1.1. Méthode manuelle pour Jade/OpenJade

Ceci est la méthode rapide mais pas très propre ("quick and dirty") qui devrait marcher pour toutes les distributions.

1. Créer un répertoire de base où installer tous les outils, par exemple `/usr/local/sgml/`. Dans la suite nous l'appellerons $_toolroot.
2. Installer Jade, la DTD DocBook et DSSSL dans le répertoire $_toolroot (en y créant les sous-répertoires `$_toolroot/jade-1.2.1`, `$_toolroot/dtd`, `$_toolroot/dsssl`).
3. Vous aurez besoin de définir la variable d'environnement SGML_CATALOG_FILES pour indiquer les catalogues dont vous disposez dans `$_toolroot`. Vous pouvez le faire avec la commande **bash$ export SGML_CATALOG_FILES = $_toolroot/dtd/docbook.cat:$_toolroot/dsssl/docbook/catalog:$_toolroot/jade-1.2.1/dsssl/catalog**
4. Maintenant vous pouvez utiliser Jade. Pour créer des fichiers HTML séparés :

   **$_toolroot/jade-1.2.1/jade/jade -t sgml -i html -d $_toolroot/dsssl/docbook/html/docbook.dsl howto.sgml**

5. Pour créer un unique document HTML, ajoutez *-V nochunks* à la liste des paramètres passés à Jade.

### 4.1.2. sgmltools

Contrairement à LinuxDoc, vous aurez besoin d'une version 2.x de sgmltools pour utiliser DocBook. Puisque la plupart des distributions ne contiennent qu'une version 1.x, vous devrez supprimer le paquetage sgmltools 1.x et installer une version 2.x ou une version CVS. Pour récupérer les sources de la dernière version CVS vous pouvez utiliser les commandes suivantes :

```
bash$ CVSROOT=:pserver:cvs@cvs.sgmltools.org:/home/cvs
bash$ export CVSROOT
bash$ cvs login
bash$ cvs -z6 get sgmltools
```

Le mot de passe CVS est 'cvs'. Une fois le téléchargement terminé, vous pouvez utiliser

```
bash$ ./compile
bash$ make
bash# make install
```

pour installer sgmltools. Pour les systèmes RedHat (utilisant RPM) vous pouvez utiliser la commande rpmfind pour récupérer la dernière version. Le programme rpmfind est disponible à l'adresse http://www.rpmfind.net/

(http://www.rpmfind.net). Vérifiez que vous récupérez bien sgmltools et non sgml-tools, car ce dernier paquetage est en fait une version 1.0.9 qui ne marche que pour les documents utilisant LinuxDoc. Pour les systèmes Debian, version 2.2 "Potato" ou plus, apt-get récupérera le bon paquetage pour vous :

```
bash# apt-get install sgmltools-2
```

Comme pour les systèmes RedHat, le paquetage sgml-tools est obsolète. Assurez-vous de récupérer sgmltools-2.

### 4.1.3. Les outils DocBook de Cygnus

Ces outils sont fournis avec la Red Hat 6.2. Vérifiez que les paquetages suivants sont installés :

- sgml-common
- docbook
- stylesheets

Vous trouverez la dernière version sur le site de Red Hat :
http://www.redhat.com/support/errata/RHBA-2000022-01.html.

Téléchargez/récupérez les fichiers RPM sur votre machine et installez-les de la manière habituelle (en tant que super-utilisateur : **rpm -Uvh nomdufichier**). Une fois les RPMs installés, vous pouvez utiliser les commandes suivantes pour convertir les fichiers DocBook :

```
bash$ db2html nomdufichier
```

pour convertir du DocBook en HTML. Un sous-répertoire de même nom que le fichier (sans l'extension .sgml) est créé et les fichiers HTML y sont placés.

```
bash$ db2pdf nomdufichier
```

pour convertir un fichier DocBook en un fichier PDF.

## 4.2. Ecrire du SGML à la main

Tout est expliqué dans le document de Jorge Godoy intitulé Using DocBook (NdT : malheureusement uniquement en anglais). Les personnes intéressées peuvent le lire en ligne à l'adresse
http://metalab.unc.edu/godoy/using-docbook/using-docbook.html

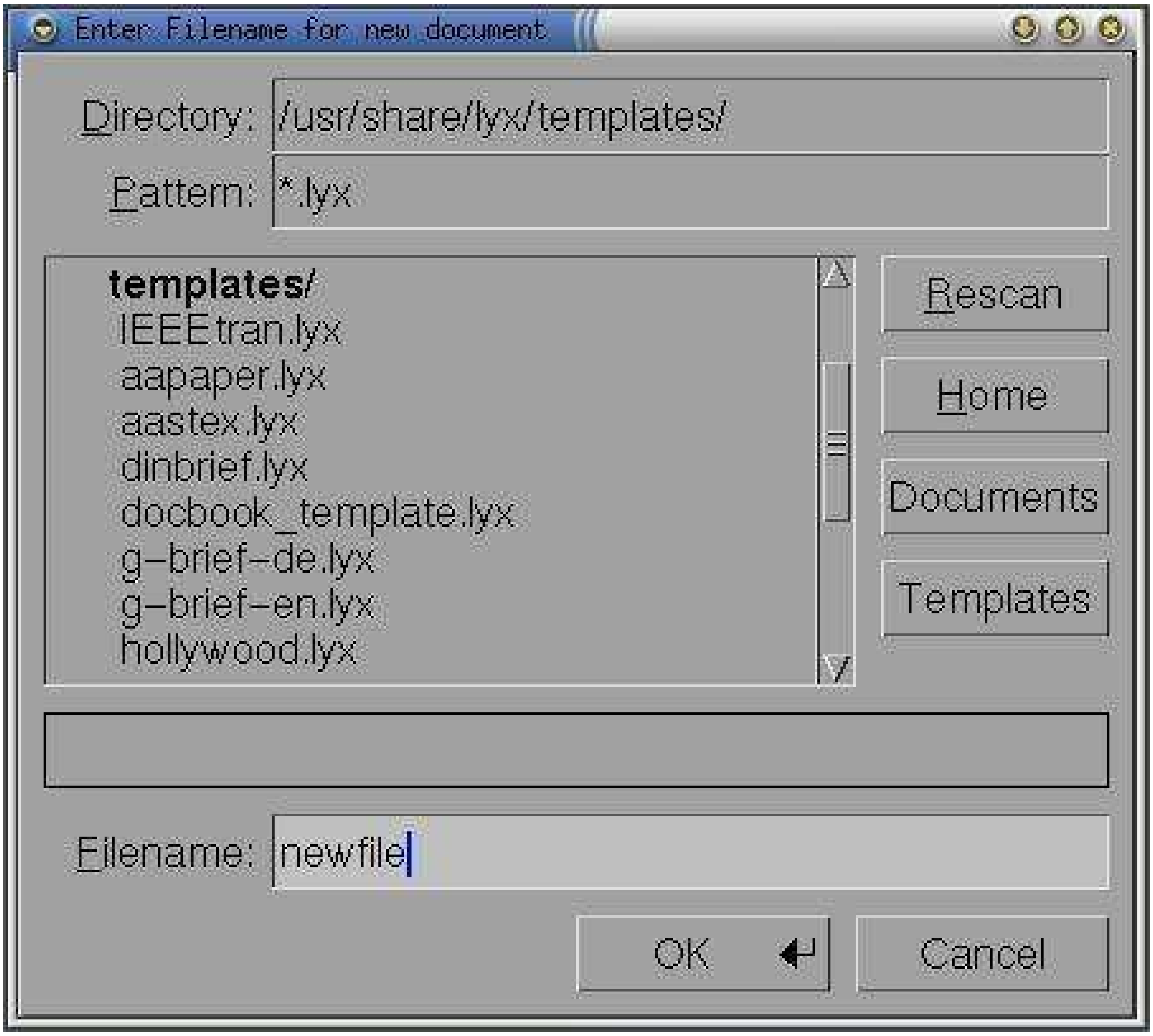
> **Si vous écrivez du SGML à la main:** SGML dispose de plus de 300 balises, et les utilise bien plus massivement que HTML. Il est recommandé de prendre un HOWTO existant comme modèle pour voir comment les autres auteurs utilisent ces balises. Il est également conseillé d'utiliser un éditeur de texte convivial comme Emacs-PSGML ou WordPerfect pour Windows, qui listent une grande partie des balises utilisables.

## 4.3. Ecrire du SGML en utilisant LyX

### 4.3.1. Nouveaux documents

Vous pouvez facilement commencer un nouvel HOWTO en utilisant LyX. La commande de menu Fichier⟶Nouveau depuis modèle... vous permet de choisir un modèle de document. Cliquez sur *Modèles* sur la droite de la boîte de dialogue puis sélectionnez *docbook_template.lyx* dans la liste des fichiers. Cliquez sur OK et vous aurez alors un document vierge. Remplissez les champs proposés tels que le titre, le résumé, le nom de l'auteur, puis commencez la rédaction.

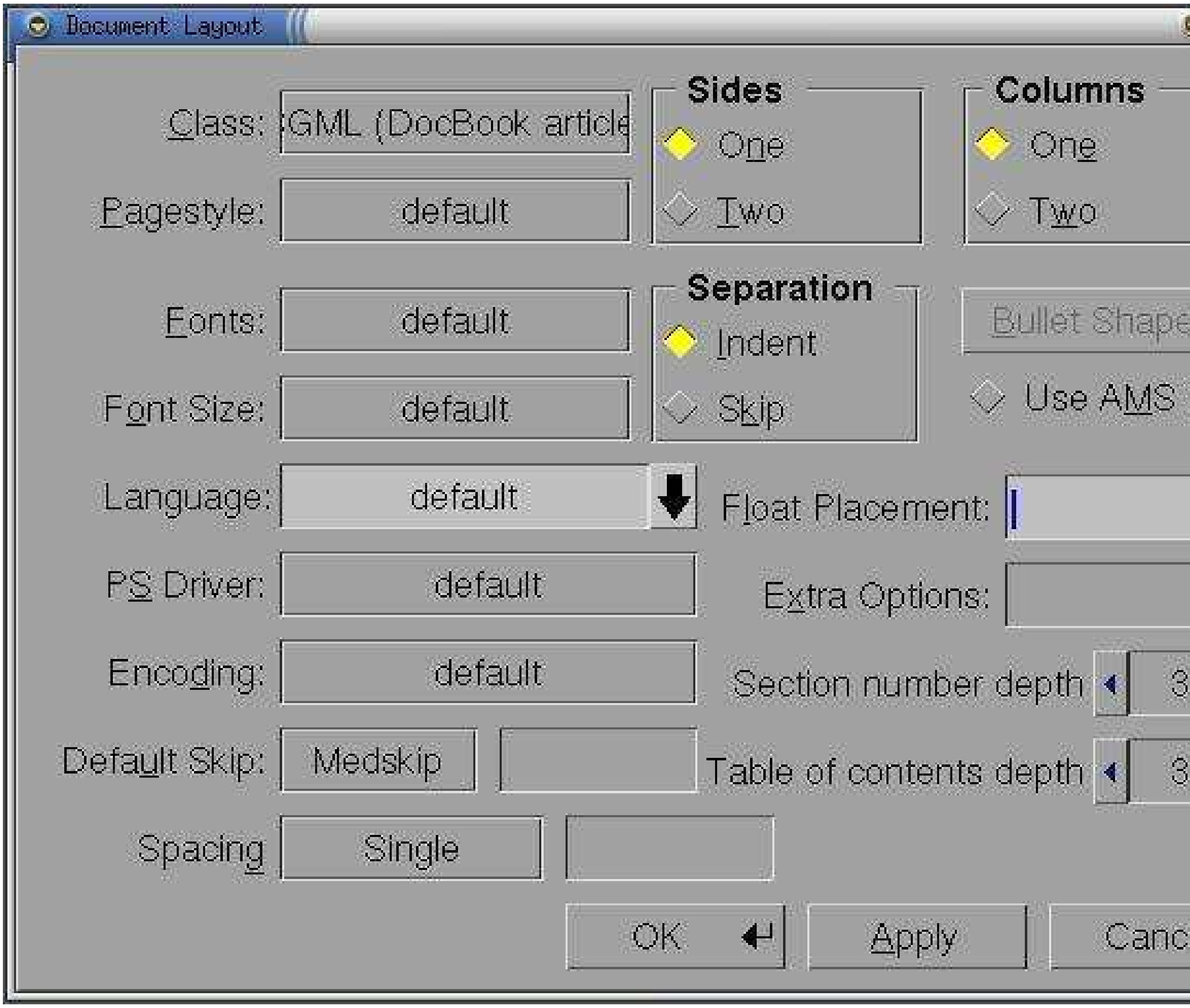
**Figure 2. Sélection du modèle DocBook dans LyX**



#### 4.3.2. Documents existants

Si vous disposez d'un document LyX, TeX ou texte, vous pouvez l'importer dans LyX grâce à la commande Fichier⟶importer. Une fois le fichier importé, utilisez la commande Mise en Page⟶Document... Choisissez l'entrée *SGML (DocBook Article)* dans le menu Type. Il vous sera demandé si vous voulez convertir tout le texte ; dites Oui. Vous devrez réappliquer les balises, mais cela consiste juste à sélectionner du texte et à choisir un style. La plupart des fonctions LyX disposent d'un raccourci clavier.

**Figure 3. Ecran de mise en page du document**



#### 4.3.3. Exporter vers SGML

Pour commencer, sauvegardez votre document au format LyX. Cela vous permettra d'éditer les futures versions plus facilement. Utilisez ensuite la commande Fichier⟶Exporter⟶DocBook... pour exporter votre fichier au format DocBook.

### 4.4. Ecrire du SGML en utilisant PSGML

#### 4.4.1. Introduction

Si vous disposez d'une distribution récente, PSGML est certainement déjà installé avec Emacs. Pour vérifier, lancez Emacs et cherchez la documentation sur PSGML (**C-h i m psgml**).

A partir de maintenant nous supposerons que PSGML est installé correctement au sein d'une version récente de GNU Emacs. Si tout cela va trop vite pour vous, consultez le chapitre librement disponible du livre de Bob Ducharme : http://www.snee.com/bob/sgmlfree/.

#### 4.4.2. Mise à jour du fichier .emacs pour PSGML

Si vous voulez que GNU Emacs utilise automatiquement le mode PSGML quand vous ouvrez un fichier *.sgml*, vérifier que PSGML saura où trouver la DTD DocBook. Si PSGML était déjà installé sur votre système, vous n'aurez certainement rien à faire. Sinon, vous devrez définir une variable d'environnement qui indiquera où trouver le catalogue SGML (la liste des DTD).

Par exemple :

```
bash$ export SGML_CATALOG_FILES=/usr/lib/sgml/catalog
```

Ensuite ajouter les lignes suivantes à votre fichier .emacs

```
;; ****************************************************************
;; set up psgml mode...
;; use psgml-mode instead of emacs native sgml-mode ;;

(autoload 'sgml-mode "psgml" "Major mode to edit SGML files." t )
(setq auto-mode-alist (append (list '("\\.sgm$" . sgml-mode)
'("\\.sgml$" . sgml-mode) ) auto-mode-alist))

;; set some psgml variables

(setq sgml-auto-activate-dtd t) (setq sgml-omittag-transparent t)
(setq sgml-balanced-tag-edit t) (setq
sgml-auto-insert-required-elements t) (setq
sgml-live-element-indicator t) (setq sgml-indent-step nil)

;; create faces to assign to markup categories

(make-face 'sgml-comment-face) (make-face 'sgml-start-tag-face)
(make-face 'sgml-end-tag-face) (make-face 'sgml-entity-face)
(make-face 'sgml-doctype-face) ; DOCTYPE data
(make-face 'sgml-ignored-face) ; data ignored by PSGML
```

```
(make-face 'sgml-ms-start-face) ; marked sections start
(make-face 'sgml-ms-end-face) ; end of marked section
(make-face 'sgml-pi-face) ; processing instructions
(make-face 'sgml-sgml-face) ; the SGML declaration
(make-face 'sgml-shortref-face) ; short references

;; view a list of available colors with the emacs-lisp command: ;; ;;
list-colors-display ;; ;; please assign your own groovy colors,
;; because these are pretty bad
(set-face-foreground 'sgml-comment-face "coral"
;(set-face-background 'sgml-comment-face "cornflowerblue")
(set-face-foreground 'sgml-start-tag-face "slateblue")
;(set-face-background 'sgml-start-tag-face "cornflowerblue")
(set-face-foreground 'sgml-end-tag-face "slateblue")
;(set-face-background 'sgml-end-tag-face "cornflowerblue")
(set-face-foreground 'sgml-entity-face "lavender")
;(set-face-background 'sgml-entity-face "cornflowerblue")
(set-face-foreground 'sgml-doctype-face "lavender")
;(set-face-background 'sgml-doctype-face "cornflowerblue")
(set-face-foreground 'sgml-ignored-face "cornflowerblue")
;(set-face-background 'sgml-ignored-face "cornflowerblue")
(set-face-foreground 'sgml-ms-start-face "coral")
;(set-face-background 'sgml-ms-start-face "cornflowerblue")
(set-face-foreground 'sgml-ms-end-face "coral")
;(set-face-background 'sgml-ms-end-face "cornflowerblue")
(set-face-foreground 'sgml-pi-face "coral")
;(set-face-background 'sgml-pi-face "cornflowerblue")
(set-face-foreground 'sgml-sgml-face "coral")
;(set-face-background 'sgml-sgml-face "cornflowerblue")
(set-face-foreground 'sgml-shortref-face "coral")
;(set-face-background 'sgml-shortref-face "cornflowerblue")

;; assign faces to markup categories

(setq sgml-markup-faces ' ( (comment . sgml-comment-face) (start-tag
  . sgml-start-tag-face) (end-tag . sgml-end-tag-face) (entity
  . sgml-entity-face) (doctype . sgml-doctype-face) (ignored
  . sgml-ignored-face) (ms-start . sgml-ms-start-face) (ms-end
  . sgml-ms-end-face) (pi . sgml-pi-face) (sgml . sgml-sgml-face)
  (shortref . sgml-shortref-face) ))

;; tell PSGML to pay attention to face settings ;;
(setq sgml-set-face t)
;; ...done setting up psgml-mode.  ;;
*******************************************************************
```

Puis relancer Emacs.

#### 4.4.3. SGML "Smoke Test"

Essayez le test suivant. Créez un nouveau document, */tmp/test.sgml* par exemple, et insérez le texte suivant :

```
<!DOCTYPE test [ <!ELEMENT test - - (#PCDATA)> ]>
```

Tapez **C-c C-p**. Si Emacs parvient à accéder à votre DTD, vous verrez le message *Parsing prolog...done* dans le minibuffer. Essayez ensuite *C-c C-e ENTREE* pour insérer un élément *<test>*. Si tout s'est bien passé, vous devriez voir ceci :

```
<!DOCTYPE test [ <!ELEMENT test - - (#PCDATA)> ]>
<test></test>
```

#### 4.4.4. Ecrire un nouveau HOWTO avec DocBook

Démarrez un nouveau fichier pour le HOWTO avec le texte suivant :

```
<!DOCTYPE ARTICLE PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
```

Tapez **C-cC-p** et retenez votre souffle. Si tout se passe comme prévu, Emacs va tourner pendant quelques secondes puis afficher le message *Parsing prolog...done* dans le minibuffer.

A partir de là, utilisez **C-cC-eRETURN** pour insérer une balise *<article>* et commencer à taper votre texte.

#### 4.4.5. Référence non-exhaustive sur PSGML pour Emacs

Voir le document d'introduction de Nik Clayton, dans la documentation FreeBSD :
http://www.freebsd.org/tutorials/docproj-primer/psgml-mode.html

## 5. Conventions de style

Cette section fournit des informations sur les conventions adoptées par le LDP dans le but de donner le même style à tous ses documents. Gardez ces conseils en mémoire lors de la rédaction de votre document.

### 5.1. Format des dates

La balise <pubdate>, dans l'en-tête, doit encadrer un texte de la forme suivante :

```
v1.0, 21 Avril 2000
```

### 5.2. Utilisation des images

Si vous devez envoyer des images avec votre document, envoyez pour chaque image une version au format EPS et une autre au format GIF ou JPEG. Bien qu'il donne de meilleurs résultats que le format JPEG, faites attention au problème des brevets sur le format GIF.

### 5.3. Versions de DocBook

Seule la version 3.1 de DocBook est supportée par le LDP pour le moment. Le LDP envisage l'utilisation future de la version 4.0. La plupart des documents en version 3.1 peuvent être facilement convertis en 4.0 en évitant d'utiliser les balises obsolètes.

L'en-tête DocBook de votre document devrait ressembler à ceci :

```
<!doctype article public "-//OASIS//DTD
DocBook V3.1//EN">
```

### 5.4. Balises obsolètes

Le LDP encourage à ne pas utiliser les balises indiquées comme obsolètes dans le livre *DocBook: The Definitive Guide*. La manière d'utiliser les nouvelles balises est indiquée dans la section Trucs et Astuces.

### 5.5. Réduction des balises

La réduction des balises consiste à utiliser </> à la place de la forme complète de fin de balise (comme </para>). Puisque cette pratique rend le document plus confus pour les futurs auteurs et les membres du LDP, et qu'elle n'est pas permise en DocBook XML, il est demandé de l'éviter.

### 5.6. Conventions

Les conventions sur les différents types de texte sont les suivantes :

Si vous voulez montrer l'utilisation d'une commande, mettez-la sous la forme d'une vraie ligne de commande sous shell. L'invite de l'interpréteur de commande doit contenir le nom du shell (bash, tcsh, zsh, etc.) suivi de $ pour les commandes d'un utilisateur normal ou # pour les commandes du super-utilisateur (root).

Ainsi une commande devrait ressembler à ceci :

```
bash$ command "en tant qu'utilisateur normal"
bash# command "en tant que root"
tcsh# setenv DISPLAY :0.0
```

## 6. Trucs et astuces pour DocBook

Cette section contient d'autres informations dont vous pourrez avoir besoin pour écrire vos documents.

### 6.1. Inclure des images

Contrairement à LinuxDoc, DocBook permet d'inclure des images dans votre HOWTO. Voici un exemple :

```
<figure> <title>LyX screen shot</title>
  <mediaobject>
    <imageobject>
```

```
      <imagedata fileref="lyx_screenshot.eps" format="eps">
    </imageobject>
    <imageobject>
      <imagedata fileref="lyx_screenshot.jpg" format="jpg">
    </imageobject>
    <textobject>
      <phrase> Screen shot of the LyX document
                     processing program
      </phrase>
    </textobject>
  </mediaobject>
</figure>
```

Il est préférable d'utiliser cette technique plutôt que la balise <graphic> pour deux raisons. Premièrement, la balise <graphic> ne sera plus disponible avec DocBook 5.0. Il faudra utiliser <mediaobject>. Donc, autant utiliser dès maintenant la bonne méthode. Deuxièmement, <mediaobject> permet de définir plusieurs types d'objets suivant le format dans lequel sera converti le document. Dans cet exemple, le premier <imageobject> est un fichier PostScript encapsulé (EPS) à utiliser pour les sorties basées sur TeX, comme DVI, PS et PDF. Le second <imageobject> est une image JPEG principalement utilisée pour HTML. La balise <textobject> ne sera utilisée que si le format ne supporte pas les images (TXT). On peut comparer ce dernier cas à la balise <alt> de HTML.

## 6.2. Nom des fichiers HTML séparés

Par défaut, quand les fichiers HTML sont générés, le processeur SGML donne des noms arbitraires aux différents fichiers. Cela peut être gênant pour les personnes voulant indexer une des pages pour pouvoir facilement en voir les changements, ou même pour vous, pour savoir ce que contient chaque fichier. Quelque soit votre raison, voici comment indiquer les noms à utiliser :

Dans votre première balise <article> (qui devrait être la seule), ajoutez un paramètre id et appelez-le index. Cela devrait vous donner ceci :

```
<article id="index">
```

Ne modifiez pas la première balise <sect1> puisqu'elle correspond généralement à une introduction et que vous voulez certainement qu'elle apparaisse sur la première page. Pour les autres balises <sect>, ajoutez un paramètre id avec comme valeur le nom voulu. Ces noms ne doivent contenir que des caractères alphanumériques et être suffisamment courts pour être compréhensibles.

```
<sect1 id="tips">
```

## 6.3. Utiliser ldp.dsl

Le LDP utilise son propre fichier DSSSL, qui ajoute quelques fonctions dont un fond de page blanc et la table des matières que vous voyez au début des HOWTO. Vous trouverez la version la plus récente de ce fichier à l'adresse http://metalab.unc.edu/gferg/ldp/ldp.dsl.

Une fois que vous disposez du fichier, vous devrez en éditer les premières lignes pour modifier le chemin vers les fichiers DSSSL DocBook. Mon exemple utilise les outils Cygnus.

Placez le fichier `ldp.dsl` dans `/usr/lib/sgml/stylesheets` et ouvrez-le avec l'éditeur de texte de votre choix. Vous devriez voir quelque chose comme ceci :

```
<!DOCTYPE style-sheet PUBLIC "-//James Clark//DTD DSSSL Style
Sheet//EN" [ <!ENTITY % html "IGNORE"> <![%html;[ <!ENTITY %
print "IGNORE"> <!ENTITY docbook.dsl SYSTEM "docbook.dsl❶" CDATA dsssl> ]]> <!ENTITY % print "INCLUD
<![%print;[ <!ENTITY docbook.dsl SYSTEM "docbook.dsl❷" CDATA dsssl> ]]> ]>
```

❶ Changez le premier "docbook.dsl" en `/usr/lib/sgml/stylesheets/nwalsh-modular/html/docbook.dsl`

❷ Changez le second "docbook.dsl" en `/usr/lib/sgml/stylesheets/nwalsh-modular/print/docbook.dsl`

Si vous utilisez un autre DSSSL, changez les chemins pour les faire correspondre avec les répertoires html et print de votre DSSSL.

Une fois cela effectué, vous pouvez générer des fichiers HTML :

```
bash$ mkdir HOWTO-HOWTO ; cd HOWTO-HOWTO
bash$ jade -t sgml -ihtml -d \
  /usr/lib/sgml/stylesheets/ldp.dsl\#html \
  ../HOWTO-HOWTO.sgml
```

La première commande crée un nouveau répertoire où placer les fichiers. La seconde (l'appel à Jade) génère un fichier HTML pour chaque section de votre document. Si vous préférez générer des fichiers RTF, vous pouvez utiliser cette commande :

```
bash$ jade -t rtf -d /usr/lib/sgml/stylesheets/ldp.dsl \
../HOWTO-HOWTO.sgml
```

# 7. CVS

Le LDP est en train de mettre un place un accès CVS pour les auteurs. Il y a en effet de bonnes raisons d'utiliser CVS :

1. CVS gère une sauvegarde des documents. Si vous passez un document à un autre auteur, ce dernier peut récupérer le document depuis CVS et continuer à travailler dessus. Si vous avez besoin de revenir à une ancienne version, vous pouvez également la récupérer.
2. CVS est très appréciable si plusieurs auteurs travaillent sur le même document. Vous pouvez lui demander de vous indiquer quelles modifications ont été faites pendant que vous travailliez sur le document, et directement intégrer ces changements.
3. CVS garde un historique des modifications du document. Cet historique peut être placé automatiquement dans le document si vous utilisez certaines balises qui seront exécutées avant l'interpréteur SGML.
4. CVS peut permettre, grâce à un programme, la mise à jour automatique du site Web du LDP, et ce dès qu'un document est terminé et soumis à CVS. Cela n'est pas encore en place, mais est envisagé. Pour l'instant, les

mises à jour par CVS signalent au coordinateur des HOWTO qu’une mise à jour du site Web est nécessaire. Si vous utilisez CVS, vous n’avez donc plus besoin d’envoyer votre document SGML par courrier électronique.

Si CVS est quelque chose de nouveau pour vous, voici quelques pages Web qui pourront vous aider :

- http://www.sourcegear.com/CVS/Docs/blandy
- https://wroclaw.art.pl/~ser/docs/cvs.html

## 7.1. Obtenir un compte CVS

D’abord, il vous faudra obtenir un compte dans le "repository" CVS du LDP (NdT : lieu de stockage et de dépôt des documents pour CVS). Il s’agit souvent du répertoire racine utilisé par CVS, où chaque projet (HOWTO, Mini-HOWTO, ...) dispose d’un sous-répertoire.

Vous devrez créer un mot de passe crypté et un identifiant d’utilisateur pour votre compte. Crypter votre mot de passe vous permet de l’envoyer au groupe CVS sans qu’on ait besoin de le voir en clair. Vous pouvez le faire par les commandes suivantes, depuis un shell bash (ou sh) :

```
bash$ echo your_password | perl -e "print crypt(<>,\ join
”,(’.’, ’/’, 0..9, ’A’..’Z’, ’a’..’z’)[rand 64, rand 64]),\"\n\""
```

Envoyez la sortie de cette commande avec l’identifiant d’utilisateur à `<cvsadmin@cvslist.linuxdoc.org>`. Votre propre répertoire CVSROOT sera créé et vous recevrez un e-mail avec une réponse. Quand vous aurez obtenu la réponse, connectez-vous sur votre CVSROOT et vérifiez que tout est configuré correctement :

```
bash$ export CVSROOT=:pserver:your_userid@cvs.linuxdoc.org:/cvsroot
bash$ cvs -d $CVSROOT login
```

(Remplacez *your_userid* par ce qui vous a été indiqué dans la réponse.)

On vous demandera de saisir votre mot de passe, à la suite de quoi vous aurez accès au repository CVS en lecture-écriture. Après avoir utilisé **cvs login** une fois et obtenu l’accès au système, votre mot de passe sera stocké dans `.cvsroot` et vous n’aurez plus besoin d’utiliser **cvs login**. Positionnez CVSROOT correctement et c’est parti. Vous pouvez récupérer le repository linuxdoc en entier avec cette commande :

```
bash$ cvs get LDP
```

Ou vous pouvez obtenir le fichier source SGML de votre propre document par :

```
bash$ cvs get howto/YOUR-HOWTO.sgml
bash$ cvs get minihowto/YOURDOC.sgml
```

## 7.2. Autres informations sur CVS

### 7.2.1. Accès CVS anonyme

Un accès CVS anonyme est disponible pour ceux qui n’ont pas besoin de compte (tels ceux désirant juste récupérer les documents du LDP). Ce repository est en lecture seule :

```
bash$ cvs -d \
```

```
:pserver:cvs@anoncvs.linuxdoc.org:/cvsroot login
```

Utilisez "cvs" comme mot de passe. Vous pouvez alors accéder aux modules linuxdoc comme décrit ci-dessus. Notez que les changements apparaissent sur le CVS anonyme une demi-heure environ après le site principal.

#### 7.2.2. Accès CVS par le Web

Vous pouvez accéder au repository CVS par le Web à l'adresse http://cvsweb.linuxdoc.org/index.cgi/linuxdoc.

#### 7.2.3. Accès graphique à CVS

Il existe des interfaces graphiques pour CVS, et vous en trouverez une liste sur le site http://freshmeat.net/appindex. Lancez-y une recherche sur: "CVS".

### 7.3. CVS et la mise à jour des fichiers

CVS reconnaît une balise spéciale, *$Id*, que vous pouvez utiliser pour insérer automatiquement la date et le numéro de version dans votre document. Lors de la mise à jour, CVS remplace cette balise par quelque chose comme *$Id: HOWTO-HOWTO.sgml,v 1.5 2000/06/12 21:45:51 gferg Exp $*. En plaçant cette balise dans votre document, elle sera modifiée à chaque changement du document, permettant une incrémentation automatique du numéro de version.

Quand vous voulez copier votre fichier modifié sur le serveur CVS, utilisez la commande **cvs ci -m "commentaires" VOTRE-HOWTO.sgml**. Le paramètre -m "commentaires" n'est pas obligatoire, mais si vous ne le mettez pas, votre éditeur de texte sera lancé (en général vi, ou du moins l'éditeur indiqué par la variable d'environnement EDITOR) et vous pourrez taper un commentaire à propos des changements.

Vous pouvez suivre toutes les discussions à propos de CVS sur la liste ldp-discuss. Pour l'instant, les soumissions LDP doivent toujours être envoyées à `<ldp-submit@lists.linuxdoc.org>`.

## 8. Distribuer votre documentation

### 8.1. Avant la distribution

Avant de distribuer votre prose à des millions de lecteurs potentiels, il y a quelques petites choses à faire.

D'abord, vérifiez le bon français de votre texte. La plupart des outils que vous utiliserez pour écrire en SGML disposent de modules de vérification d'orthographe. Si ce n'est pas le cas, il y a toujours ASPELL.

Deuxièmement, faites relire votre document par quelqu'un d'autre que vous pour obtenir des commentaires et d'éventuelles corrections. La documentation publiée par le LDP a besoin d'être la plus correcte possible, car des millions d'utilisateurs de Linux peuvent avoir besoin de la lire. Si vous faites partie d'une liste de discussion parlant du sujet traité, demandez aux autres intervenants de vous aider.

Troisièmement, créez un site Web où vous pourrez distribuer vos réalisations. Ce n'est pas indispensable, mais très utile aux autres pour retrouver l'origine du document.

#### 8.1.1. Valider votre code SGML

En utilisant Jade ou nsgmls, vous pouvez valider votre code SGML par rapport à la DTD pour être sûr qu'il ne comporte pas d'erreurs.

```
bash$ nsgmls -s HOWTO-HOWTO.sgml
```

S'il n'y a pas de problème, aucun message ne sera affiché.

### 8.2. Problèmes de copyright et de licence

Pour qu'un document soit accepté par le LDP, il doit utiliser une licence qui soit en accord avec la section "LICENSE REQUIREMENTS" du Manifeste du LDP disponible à l'adresse http://www.linuxdoc.org/manifesto.html. En tant qu'auteur, vous pourrez conserver le copyright et ajouter d'autres restrictions (par exemple, que vous deviez approuver toutes traductions ou travaux dérivés). Un exemple de licence est disponible à l'adresse http://www.linuxdoc.org/COPYRIGHT.html. Si vous choisissez d'utiliser ce copyright, copiez-le dans votre code source dans une section intitulée "Copyright et licence" ou quelque chose de similaire. Incluez également une clause de propriété (puisque vous serez le propriétaire du document). Si vous êtes le nouveau responsable d'un HOWTO qui existait déjà, vous devez inclure les clauses de copyright des anciens auteurs et indiquer les périodes où ils ont maintenu le document.

Vous remarquerez que la licence du HOWTO-HOWTO requiert une notification à l'auteur de tous travaux dérivés ou traductions. En outre, tous les codes sources inclus sont protégés par la GPL. Si votre document inclut des sources, et désirez que d'autres puissent les utiliser, faites-en de même.

### 8.3. Soumission au LDP

Une fois que votre document a été soigneusement relu, vous pouvez le soumettre au LDP. Envoyez un e-mail à `<ldp-submit@lists.linuxdoc.org>` avec votre fichier source (que vous pouvez gzipper si vous le désirez) en attachement .

Indiquez le nom de votre HOWTO dans le sujet du mail, et utilisez le corps du message pour rendre compte des principaux changements effectués. Cela permet aux coordinateurs de faire leur travail plus rapidement, et de ne pas avoir à trop attendre avant que votre HOWTO ne soit disponible sur le site Web du LDP. Si vous n'avez pas de réponse dans les sept jours suivants, renvoyez un mail pour savoir si votre HOWTO est toujours bien en cours de traitement.

Si votre HOWTO contient des extras, comme des images ou un catalogue particulier, placez-les tous ainsi que votre fichier source SGML dans une archive tar.gz et envoyez cette dernière en attachement avec votre mail à la liste ldp-submit.

### 8.4. Maintenance des HOWTO

Maintenant que vous êtes un auteur de HOWTO, vous devriez maintenir votre document et le mettre à jour au fil des nouvelles versions de logiciels. Vous devriez également répondre aux commentaires et questions raisonnables de vos lecteurs. Vous n'avez pas à tous les aider, surtout si les réponses sont déjà dans votre HOWTO. Néanmoins, un des objectifs du LDP est d'aider au maximum les utilisateurs. C'est également un bon moyen d'accroître la popularité de Linux.

# 9. FAQ à propos du LDP

## 9.1. Je désire aider le LDP. Comment puis-je le faire ?

La façon la plus simple, c'est de trouver un sujet et d'en faire un document. Regardez également la liste des HOWTO non maintenus et voyez s'il n'y en a pas un que vous pourriez continuer.

## 9.2. Je voudrais publier un ensemble de documents du LDP dans un livre. Comment le contenu du LDP est-il protégé ?

Référez-vous à la page http://www.linuxdoc.org/COPYRIGHT.html. Notez que cela n'est qu'un exemple destiné aux auteurs. Néanmoins, la licence utilisée ne peut pas être plus restrictive que celle donnée à cette URL.

## 9.3. J'ai trouvé une erreur dans un document du LDP. Puis-je la corriger ?

Contactez l'auteur du document, ou le coordinateur du LDP sur la liste `<ldp-submit@lists.linuxdoc.org>` et informez-le du problème en lui donnant une éventuelle correction.

## 9.4. Mais je ne connais pas SGML / Je n'arrive pas à faire marcher les outils / Je n'aime pas SGML.

Pas de problème. Vous avez la possibilité d'écrire un brouillon de votre HOWTO dans le format que vous voulez puis de le soumettre au LDP. Un volontaire se chargera de le relire et de le convertir en DocBook pour vous. Il vous sera alors plus facile de le maintenir à jour. Si vous avez des questions vous pouvez toujours faire appel aux volontaires de la liste DocBook du LDP à l'adresse `<ldp-docbook@lists.linuxdoc.org>`.