

Écrire de la documentation en utilisant DocBook

Un cours accéléré

David Rugge

davidrugge@mindspring.com

Mark Galassi

rosalia@galassi.org

Éric Bischoff

ebischoff@nerim.net

Marc Blanc

marc.blanc@linux-france.org

Écrire de la documentation en utilisant DocBook: Un cours accéléré

par David Ruge, Mark Galassi, Éric Bischoff, et Marc Blanc

Copyright © 1997-2006 David Ruge, Mark Galassi, Éric Bischoff

Ce document est un premier didacticiel sur la façon d'écrire de la documentation en DocBook.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".



Table des matières

1. Introduction.....	1
1.1. À propos de ce document.....	1
1.2. Pourquoi DocBook ?.....	2
1.3. Votre vision du monde	3
1.4. Balisage fondé sur le contenu	4
2. Les premiers pas	5
2.1. Présentation des outils.....	5
2.2. L'ancienne méthode : les DocBook-Tools	5
2.3. La nouvelle méthode : XSLTProc et FOP.....	7
2.4. Mon premier fichier DocBook	8
2.5. Introduction aux feuilles de style	9
3. Notions de base.....	11
3.1. Anatomie d'une balise DocBook	11
3.2. Structure d'un fichier DocBook	11
4. La déclaration de type de document (DTD).....	14
4.1. Utiliser des entités pour du texte partagé	14
4.2. Utiliser les entités pour inclure d'autres fichiers.....	15
4.3. Identification des fichiers avec les ID publics formels.....	15
4.4. Utiliser des sections marquées pour manipuler du contenu conditionnel.....	17
5. Méta-informations	19
6. Listes	21
6.1. La simplelist	21
6.2. La itemizedlist.....	22
6.3. La orderedlist	22
6.4. La variablelist.....	23
6.5. La segmentedlist.....	24
6.6. qandaset.....	24
6.7. Procédures.....	25
7. Les tableaux.....	27
8. Les graphiques	29
9. Les liens.....	32
10. Description de l'interface d'une application	33
10.1. Exemples	33
10.2. Éléments d'interface GUI	34
10.3. Éléments de ligne de commande.....	36
10.4. Description d'une API	38
11. Diverses balises utiles	40
11.1. Balises de désignation.....	40
11.2. Balises de mise en forme	40
11.3. Avertissements, conseils et notes	40
12. Où aller ensuite ?	41
12.1. Ressources sur DocBook	41
A. Licence	42
A.1. Free Documentation Licence	42
B. Mode PSGML de Emacs	50
Glossaire	52



Liste des tableaux

7-1. Kilométrage de la souris..... 28



Chapitre 1. Introduction

1.1. À propos de ce document

Ce document se présente comme une introduction sur la façon d'utiliser toute la puissance et l'efficacité de DocBook. Il essaie de vous introduire en cette matière dans les délais les plus courts, c'est pourquoi il se nomme "Cours éclair de DocBook".

Même si vous n'avez jamais utilisé DocBook ou un autre langage de balisage (comme LinuxDoc) auparavant, vous devriez être capable de devenir compétant simplement en lisant ce guide et en utilisant la version en ligne (<http://www.oasis-open.org/docbook/documentation/reference/html/docbook.html>) ou papier (<http://www.oreilly.com/catalog/docbook/index.html>) du [DocBook - Le Guide Définitif] publiée par O'Reilly & Associates.

Note : Notez que ce didacticiel est destiné à être utilisé avec, et pas à la place de, la Référence DocBook. Il existe un certain nombre de cas où il est plus facile de se reporter à la Référence plutôt que d'essayer de refaire ce qui a déjà été présenté. Utilisez ce guide pour comprendre ce qu'est DocBook et comment utiliser les balises.

Ce didacticiel vous présente suffisamment de DocBook pour écrire une documentation de base. Vous apprendrez :

- Ce qu'est DocBook
- Comment obtenir les outils DocBook et les faire fonctionner
- Le format des balises DocBook
- Comment structurer vos documents proprement
- Comment utiliser les listes et les tables pour organiser vos données
- Comment décrire les éléments GUI et les commandes Unix
- Comment intégrer des graphiques à votre documentation
- Comment insérer des URL et créer des références croisées

En annexe, vous pouvez trouver une courte description du mode psgml de Emacs.

Ce didacticiel provient de la fusion de trois documents :

- le "Introduction to DocBook" de Mark Galassi
- le "KDE crash-course to DocBook" de David Rugge
- des parties du didacticiel de Eric Bischoff sur DocBook



Certaines parties de ce document sont empruntées au "DocBook 3.0 Reference" de Eve Maler de ArborText, Inc et Terry Allen de Fujitsu Software Corporation. Les parties de ce document empruntées sont Copyright © 1992, 1993, 1994, 1995, 1996, 1997 par HaL Computer Systems, Inc., O'Reilly & Associates, Inc., Fujitsu Software Corporation et ArborText, Inc.

1.2. Pourquoi DocBook ?

Le format DocBook a été développé par le consortium OASIS spécifiquement pour la documentation technique. Il fournit un riche ensemble de balises pour décrire le contenu de votre document.

Ici, voyons certains points-clé pour vous aider à comprendre ce qu'est DocBook :

Docbook est un langage de balises

Il est très similaire au HTML. Les balises permettent la structuration de votre document et s'entremêle avec le texte.

Ce point particulier en fait une révolution avec respect de la traduction de la documentation, car la phase DTP (rendant le texte agréable) est faite une fois pour toute indirectement en balisant le texte original. Les traducteurs ont seulement à traduire "entre les balises" et en pressant une simple touche la sortie traduite est générée.

Il est conçu pour la documentation technique

DocBook est parfaitement adapté pour la documentation des pièces détachées de moteurs de voitures. Cependant, il est fortement prédisposé pour la documentation des programmes informatiques.

Il est maintenu par un consortium indépendant

Le consortium OASIS (<http://www.oasis-open.org>) est en charge de la maintenance du standard développé par le Comité Technique DocBook. C'est une garantie d'indépendance vis-à-vis des logiciels propriétaires.

Des acteurs majeurs de l'industrie comme Boeing ou IBM sont membres de OASIS. Voyez la liste des membres (<http://www.oasis-open.org/html/members.htm>) pour plus d'information.

Techniquement, DocBook est une DTD SGML ou XML

Ceci veut dire que l'on peut tirer profit de nombreux outils SGML et XML. Alors que DocBook comme implémentation XML est assez récent, il a une longue histoire en tant qu'implémentation SGML.

DocBook n'est pas un langage de présentation

DocBook n'indique *pas* spécifiquement à quoi le document final ressemblera. Ceci permet à l'auteur de se concentrer sur l'organisation du document. La présentation est gérée par les feuilles de style.



Ceci assure à vos documents une apparence logique.

DocBook est personnalisable

Il est facile de personnaliser la DTD en fonction de vos besoins. Mais ceci doit être fait dans le respect des conventions SGML/XML.

Si DocBook est utilisé concurremment avec les feuilles de style de Norman Walsh, il est également possible de personnaliser un fichier DocBook pour l'impression ou la mise en ligne.

DocBook est intelligible

Le grand nombre de balises défini dans DocBook garantit qu'il peut s'accommoder de situations très variées.

Ceci fait qu'il est un peu difficile à apprendre, mais dans la plupart des cas vous n'aurez à utiliser qu'un nombre limité de balises.

DocBook utilise des balises longues et compréhensibles

Un exemple de ces balises est `<itemizedlist>` ou `<literallayout>`. Ceci rend un texte DocBook plus facile à lire qu'une source HTML. Désavantage, il peut aussi devenir un peu fastidieux de taper ces longues balises, mais certains modes spécialisés (comme le mode `psgml` de Emacs) peuvent vous y aider.

DocBook n'assure pas la compatibilité ascendante entre les révisions majeures

Alors que ceci pourrait sembler être un inconvénient, en fait ce ne l'est pas, car ça assure une présentation propre même si de mauvais choix ont été faits précédemment par le Comité DocBook de OASIS et parce que les documents écrits avec différentes DTD peuvent coexister sur un même ordinateur.

Certains de ces points-clé seront vus plus en détail dans les prochaines sections.

1.3. Votre vision du monde

La plupart des personnes qui font de la composition de texte utilisent des systèmes WYSIWYG dans lesquels ils tapent des balises d'instruction indiquant au processeur la position du texte dans la page (comme TeX et troff).

Ces approches souffrent de quelques sérieux problèmes. Le plus important est la longévité du document : l'information (ce que vous tapez) et qui doit perdurer s'entremêle avec l'information qui sera dépassée (l'information processeur).

Un autre gros problème avec cette approche ancienne est l'absence de *structure*: le balisage n'affecte pas le contenu, mais plutôt la mise en page. Si vous êtes intéressé pour l'indexation de documents écrits en TeX. Il est facile d'indexer toutes les occurrences d'un texte en gras, mais ça n'intéresse pas tout le monde ! À l'inverse, il sera plus facile d'indexer tous les noms de fonction dans une API.



Avec l'ancienne approche de mise en forme vous auriez besoin d'un logiciel d'intelligence artificielle qui comprenne le texte et vous dise "aha ! ceci doit être la définition d'une fonction dans l'API".

1.4. Balisage fondé sur le contenu

Comment baliser vos documents de façon optimale pour que l'information puisse en être extraite et indexée ? L'approche de DocBook est de fournir un ensemble très riche de balises de délimitation qui toutes sont en relation avec la structure et la nature du contenu du document.

Pour vous donner des exemples de balises qui peuvent vous aider à générer des index automatiques : `<attribution>` et `<command>`. Si vous avez une grosse documentation (par exemple, tout les logiciels et matériels Sun sont documentés avec DocBook) vous pouvez faire une recherche facilement dans les documents qui traitent d'une commande appelée **mount** ou une citation attribuée à Ken Thompson. Avec une recherche structurée vous pourriez seulement trouver les occurrences de **mount** quand c'est un nom de commande et de Thompson quand il est l'auteur de la citation.

Maintenant imaginez ce qui se passerait si le Web tout entier utilisait un langage de balisage basé sur le contenu au lieu du HTML : un moteur de recherche vous donnerait l'information dont vous avez besoin sans toutes les références supplémentaires nécessaires pour utiliser ces termes. Une recherche sur `mount` sur le Web ne trouvera pas dans la plupart des cas les références à la commande UNIX **mount**.

Un riche langage de balises comme DocBook est une excellente idée à divers points de vue, mais il peut aussi être difficile à utiliser. DocBook possède des centaines de balises (à l'opposé de quelques unes pour le HTML), ainsi l'apprentissage est plus dur. C'est vrai et le seul moyen pour y remédier est d'écrire une documentation sur l'utilisation de DocBook !

D'un autre côté, si vous êtes familier avec DocBook ça ne vous prendra pas trop de temps de taper les balises. Gardez à l'esprit que la plupart du temps une personne n'écrit pas mais se concentre sur la mise en forme de son document. Si vous utilisez DocBook vous passerez plus de temps à l'écriture et moins à la mise en forme.



Chapitre 2. Les premiers pas

2.1. Présentation des outils

Cette section décrit comment travailler avec DocBook sur les systèmes Unix-like comme Linux. Si malheureusement vous travaillez sur d'autres systèmes, vous devrez assembler et configurer les outils nécessaires ou acheter une solution commerciale.

La technologie concernant DocBook évolue. Dans le passé :

- DocBook a démarré comme une "application SGML" comme le HTML
- Les documents DocBook étaient convertis à l'aide des feuilles de style DSSSL.
- La mise en page PDF était réalisée par le moteur de composition TeX.

Maintenant, au contraire :

- DocBook suit la syntaxe XML, comme le fait le HTML.
- Il est converti dans d'autres formats par les feuilles de style XSLT.
- La mise en page PDF étant réalisée par le moteur XSL-FO.

Les outils utilisés pour générer des fichiers DocBook doivent suivre cette évolution. Nous présenterons successivement les outils DocBook (technologie SGML/DSSSL) et XSLTProc/FOP (technologie XML/XSL).

Note : Il existe une autre variante des SGML/DSSSL Tools appelée "SGML-Tools (Lite)". Pour plus d'information voir <http://www.sgmttools.org>.

2.2. L'ancienne méthode : les DocBook-Tools

Les DocBook-tools consistent en plusieurs paquetages qui fonctionnent ensemble pour convertir des fichiers SGML DocBook dans d'autres formats, et accomplir d'autres opérations. Ces formats comprennent :

- HTML
- TeX et DVI
- PostScript
- RTF
- PDF
- Man pages
- TexInfo



Note : Nous avons dit que les DocBook-Tools étaient destinés au SGML DocBook. C'est vrai, mais ils peuvent aussi s'utiliser pour des fichiers XML DocBook.

La page du projet DocBook-Tools est sur <http://sources.redhat.com/docbook-tools> et les paquets eux-mêmes peuvent être obtenus depuis <ftp://sources.redhat.com/pub/docbook-tools/new-trials/> ou un des miroirs. Certaines distributions commerciales ont adopté les DocBook-Tools et fournissent leur propres CD.

Les DocBook-Tools comprennent les paquets suivants :

sgml-common

déclarations SGML de base et outils

jade

moteur de feuilles de style SGML et DSSSL

jadetex

ensemble de macros TeX utilisées par les fichiers générés par Jade

docbook-dtdXX-sgml

DTD SGML DocBook (un paquetage par version de la DTD)

docbook-style-dsssl

Feuilles de style DSSSL de Norman Walsh pour DocBook

perl-SGMLspm

Interface entre Perl et SGML

docbook-utils

Scripts Shell et utilitaires perl

psgml

Mode majeur pour Emacs pour éditer des fichiers SGML

Nous décrirons seulement la version RPM de ces paquets. Pour les autres formats vous devrez adapter.

1. Télécharger les paquets
2. Les installer

comme dans l'exemple suivant :

```
$ ncftp ftp://sources.redhat.com
ncftp> cd pub/docbook-tools/new-trials/RPMS
ncftp> mget i386/*.rpm
```



```
ncftp> mget noarch/*.rpm
ncftp> quit
$ su
Password: ultra-sucure
# rpm -ih sgml-common*.rpm
# rpm -ih jade*.rpm
# rpm -ih jadetex*.rpm
# rpm -ih docbook-dtd44-sgml*.rpm
# rpm -ih docbook-style-dsssl*.rpm
# rpm -ih perl-SGMLSpM*.rpm
# rpm -ih docbook-utils*.rpm
# exit
```

Note : L'ordre d'installation des paquetages est important.

Note : Quand vous faites une mise à jour, au lieu d'installer pour la première fois, l'étape `rpm -ih` sera remplacée par `rpm -uh`.

Vous êtes maintenant prêt pour éditer des documents SGML/DocBook et ensuite les convertir en d'autres formats. Pour ceci vous utiliserez une des commandes suivantes :

- **docbook2html** => HTML
- **docbook2ps** => PostScript
- **docbook2pdf** => PDF
- **docbook2rtf** => Rich Text Format
- etc.

2.3. La nouvelle méthode : XSLTProc et FOP

Les outils basés sur XML sont plus faciles à utiliser que ceux fondés sur SGML. Les utilitaires de base peuvent être utilisés directement.

Les formats de sortie sont :

- HTML
- PDF
- MIF
- PCL
- PS
- SVG



RTF est planifié par les versions expérimentales de FOP.

Sur les systèmes basés sur RPM, vous aurez besoin des paquetages suivants :

docbook-dtdXX-xml

DTD DocBook XML (un paquetage par version de la DTD)

docbook-style-xsl

Feuilles de style XSL de Norman Walsh pour DocBook

libxml2

Bibliothèque XML de Daniel Veillard nécessaire pour libxslt

libxslt

Moteur de transformation XSLT de Daniel Veillard

java-1_4_2-sun

... ou un autre environnement Java, nécessaire pour fop

fop

Processeur XSL-FO

La plupart de ces paquetages sont habituellement pré installés dans votre distribution Linux. Pour certaines, utilisez la commande **rpm -ih nom du paquetage**.

Vous êtes maintenant prêts à éditer vos documents XML/DocBook et les convertir dans d'autres formats. Pour faire ceci, vous exécuterez les commandes suivantes :

- **xsltproc**: DocBook vers HTML ou XSL-FO
- **fop**: XSL-FO vers PDF, PS, etc.

2.4. Mon premier fichier DocBook

Premièrement, vous avez besoin d'un fichier DocBook pour convertir. Prenez un simple éditeur de texte et tapez (ou copiez-collez) les lignes suivantes :

Exemple 2-1. Fichier DocBook minimal

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.4//EN"
"/usr/share/xml/docbook/schema/dtd/4.4/docbookx.dtd">
<book>

<bookinfo>
```



```
<title>Hello, world</title>
</bookinfo>

<chapter>
<title>Hello, world</title>

<para>C'est mon premier fichier DocBook.</para>

</chapter>
</book>
```

Ensuite, pour l'enregistrer, faites, *myfile.docbook*. Si vous utilisez les SGML Tools, le chemin entre parenthèses sur la troisième ligne n'est pas nécessaire.

Pour convertir ce fichier du DocBook au format HTML, utilisez la commande :

```
$ docbook2html myfile.docbook
```

ou

```
$ xsltproc /usr/share/xml/docbook/stylesheet/nwalsh/current/html/docbook.xsl myfile.docbook -o myfil
```

selon votre boîte à outils (DocBook-tools ou XSLTProc).

Jade ou xsltproc vont s'exécuter et si votre document ne contient pas d'erreurs vous obtiendrez un ensemble de fichiers HTML. Utilisez un navigateur Web pour ouvrir le fichier principal, *book1.htm*.

Si vous obtenez des erreurs, lisez le journal d'erreur et corrigez les en commençant par le début. Souvent, il s'agit d'une balise non refermée qui cause des erreurs en cascade.

2.5. Introduction aux feuilles de style

Note : Cette section est un peu périmée. Elle devrait décrire les feuilles de style XSLT.

Les feuilles de style peuvent optimiser la conversion de façon à ce que les fichiers résultants aient des noms plus intelligibles. Changez les balises `<book>` et `<chapter>` de l'exemple ci-dessus comme suit :

Exemple 2-2. Fichier DocBook minimal, avec certains attributs

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.4//EN">

<book lang="en">
<!-- Please remark the "lang" attribute here -->
```



```
<bookinfo>
<title>Hello, world</title>
</bookinfo>

<chapter id="introduction">
<title>Hello, world</title>

<para>C'est mon premier fichier DocBook.</para>

</chapter>
</book>
```

Le texte entre `<!--` et `-->` est un commentaire; utilisez les pour attirer l'attention du lecteur du code source DocBook. Ils ne seront jamais interprétés.

Maintenant, utilisons la feuille de style fournie avec le `docbook-utils` :

```
$ cp /usr/share/sgml/docbook/utils*/docbook-utils.dsl .
$ docbook2html -d docbook-utils.dsl#html myfile.docbook
```

Les fichiers iront vers un répertoire HTML et seront nommés `index.html` et `introduction.html`, à la place de noms comme `book1.html`. Le fichier principal sera toujours nommé `index.html` et les chapitres comme `<chapter id="introduction">` iront vers les fichiers nommés après l'attribut `id`. Ces modifications ont été faites par les feuilles de style.

Note : Utilisez `#print` à la place de `#html` pour spécifier la partie de la feuille de style à utiliser si vous vous servez de certaines commandes comme `docbook2pdf` au lieu de `docbook2html`.

En fait les feuilles de style sont des outils très puissants. Elles permettent d'éliminer les problèmes du style "Je veux que ça ressemble à ça". Si vous vous posez des questions pendant l'écriture d'un fichier DocBook, ça indique que quelque chose est erroné dans votre approche.

Si vous jetez un oeil dans le fichier nommé `docbook-utils.dsl`, vous verrez qu'il est écrit dans un langage crypté nommé DSSSL, qui ressemble à du LISP. Ceci malheureusement indique qu'une bonne connaissance de la programmation est nécessaire pour accorder les feuilles de style.

Plus d'information sur la façon de personnaliser les feuilles de style peut être trouvée sur <http://www.nwalsh.com/docbook/dsssl/doc/>. Et également sur DSSSL sur <http://www.jclark.com/dsssl/> (<http://www.jclark.com/dsssl/>).



Chapitre 3. Notions de base

3.1. Anatomie d'une balise DocBook

Une balise DocBook consiste en un élément et des attributs. Par exemple `<chapter id="introduction">` contient l'élément `chapter` et l'attribut `id`. L'élément modifie le texte balisé et les attributs modifient l'élément. Par exemple, l'élément `chapter` indique que tout texte compris entre les balises d'ouverture et de fermeture sera traité comme un chapitre, tandis que l'attribut `id` désigne le chapitre et ainsi il peut être lié ou utilisé comme un nom de fichier quand DocBook le convertit dans un autre format.

La plupart des balises DocBook possèdent un ensemble commun d'attributs. Les attributs communs que vous utiliserez le plus souvent sont `lang`, qui spécifie le langage de données à l'intérieur de la balise et `id`, qui désigne la balise.

Important : Toutes (ou à peu près) les balises DocBook doivent avoir une balise de début et une de fin. Si vous lisez la Référence DocBook vous noterez que certaines balises DocBook n'ont pas de repères de début et de fin. Omettre les repères de fin où ils ne sont pas nécessaires n'est pas une bonne idée car ceci ne fonctionne qu'avec le SGML DocBook et pas avec le XML DocBook.

De même, assurez vous de respecter la casse de la balise car XML est pointilleux à ce sujet.

Pour une information détaillée au sujet des balises et une liste de toutes les balises disponibles, voir DocBook Reference

(<http://www.oasis-open.org/docbook/documentation/reference/html/docbook.html>) et DocBook Quick Reference (<http://www.docbook.org/tdg/en/html/quickref.html>). Ces guides concernent la version 3.1 de DocBook mais peuvent être utilisés avec quelques petites modifications.

3.2. Structure d'un fichier DocBook

Les balises concernées dans cette section sont énumérées ci-dessous.

book - Livre
article - Article
refentry - Équivalent d'une page de manuel
chapter - Chapitre d'un livre ou d'un article
sect1 ... sect5 - Sections et sous-sections d'un chapitre
title - Texte d'en-tête ou titre d'un élément orienté-bloc
para - Paragraphe

Exemple 3-1. Chapitres et sections

```
<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN">
```



```
<book id="hello-world" lang="en">

<bookinfo>
<title>Hello, world</title>
</bookinfo>

<chapter id="introduction">
<title>Introduction</title>

<para>Ceci est l'introduction. Elle a deux sections</para>

<sect1 id="about-this-book">
<title>À propos de ce livre</title>

<para>C'est mon premier fichier DocBook</para>

</sect1>

<sect1 id="work-in-progress">
<title>Attention</title>

<para>Ceci est encore en construction</para>

</sect1>

</chapter>
</book>
```

L'exemple ci-dessus présente un squelette des balises de structure. La première ligne est la déclaration de DTD qui indique quelle DTD utiliser pour compiler ce document (nommément DocBook 4.2). Cette information sera décrite plus en détail dans la section Document Type Declaration.

Ensuite vient le modèle de contenu, qui est `<book>` ici. Vous pouvez utiliser `<article>`, qui est plus léger que `<book>` ou `<refentry>` qui est l'équivalent des pages de manuel UNIX.

Notez l'usage de l'attribut `lang` dans la balise `<book>`. L'attribut langage devrait toujours être utilisé pour rendre plus aisé de déterminer dans quelle langue le document a été écrit.

Après la balise `<book>` vient la meta information pour le document qui est encapsulée dans le repère `<bookinfo>`. Cette information sera décrite en détail dans la section Meta Information.

Ensuite viennent les chapitres de votre livre, qui contiennent une ou plusieurs balises de section (`<sect1>` - `<sect5>`). Les attributs (non numériques) ID pour `<chapter>` et `<sect>` des balises sont nécessaires pour deux raisons :

- Désigner toutes les sections de votre document vous permet de plus facilement faire des références croisées de ce document avec des hyperliens.
- jade utilise les ID des chapitres pour nommer les fichiers de sortie, ainsi si vous ne précisez pas les ID pour tous vos chapitres, les noms de fichiers seront différents chaque fois que les documents seront mis à jour, ce qui gaspille de l'espace dans le CVS.



Les chapitres et les sections doivent contenir au moins un `<title>` et une balise `<para>` vide. Les endroits où certains éléments doivent, ou ne doivent pas, ou peuvent apparaître est défini par la DTD DocBook et présenté en détail par le Guide de Référence.

Le contenu dans DocBook est inséré dans la balise `<para>` qui est très similaire à la balise `<p>` du HTML et du LinuxDoc sauf qu'elle doit toujours avoir un repère de fermeture `</para>`. Chaque fois qu'il y a un saut de ligne (comme dans une liste) le texte devra être inséré entre les balises `<para>`.

Résumons ce que nous avons vu. Un livre sera structuré de la façon suivante :

```
book
  meta information
  chapter
    sect1
    sect2
  sect1
  chapter
    sect1
  appendix
    sect1
  appendix
    sect1
  ...
  glossary
```

Un article sera structuré de la façon suivante :

```
article
  meta information
  sect1
  sect1
    sect2
  sect1
  ...
```



Chapitre 4. La déclaration de type de document (DTD)

Dans "la vie réelle", la DTD (les toutes premières lignes de votre fichier DocBook) est légèrement plus compliquée que la simple ligne présentée dans le squelette de la section précédente.

Nous examinerons dans les sections suivantes pourquoi il y a un sens à ajouter des déclarations ici.

4.1. Utiliser des entités pour du texte partagé

Exemple 4-1. Entités utilisées pour partager du texte

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.4//EN" [
  <!ENTITY cereals "<productname>Céréales Quelgoss</productname>">
  <!ENTITY orange-juice "<productname>Orangeade Piedboeuf</productname>">
]>

<book id="marketing-study" lang="en">

<bookinfo>
<title>Étude de marché sur les &cereals;</title>
</bookinfo>

<chapter id="introduction">
<title>Introduction</title>

<para>
Cette étude décrit l'impact attendu du nouveau produit
&cereals; et examine l'opportunité d'une campagne
publicitaire combinée avec l'&orange-juice;.
</para>

</chapter>
</book>
```

Vous trouvez ici les "définitions d'entités" qui font de &cereals; un synonyme du nom de produit "Céréales Quelgoss", balisé comme <productname>Céréales Quelgoss</productname>.

Cet exemple ressemblera à quelque chose comme ça quand il sera converti :

Cette étude décrit l'impact attendu du nouveau produit Céréales Quelgoss et examine l'opportunité d'une campagne publicitaire combinée avec l'Orangeade Piedboeuf.



Procéder de cette façon présente plusieurs avantages :

- Ça évite de taper plusieurs fois la même chose
- Ça permet de centraliser les modifications en un seul endroit
- Si le nom de l'entité est choisi avec précaution, il rend le texte source plus lisible

4.2. Utiliser les entités pour inclure d'autres fichiers

La DTD est aussi le bon endroit pour inclure d'autres fichiers. La documentation tend habituellement à devenir très lourde, ainsi c'est une bonne idée de la diviser en plusieurs fichiers qui sont inclus dans un seul fichier principal comme ceci :

Exemple 4-2. Entités utilisées pour inclure d'autres fichiers

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.4//EN" [
  <!ENTITY introduction SYSTEM "introduction.docbook">
  <!ENTITY marketing SYSTEM "marketing-plan.docbook">
  <!ENTITY advertising SYSTEM "advertising-campain.docbook">
]>

<book id="marketing-study" lang="en">

<bookinfo>
<title>Marketing study about &cereals;</title>
</bookinfo>

&introduction;
&marketing;
&advertising;
&conclusion;

</book>
```

Bien sûr, vous devez créer les fichiers nommés `introduction.docbook`, `marketing-plan.docbook`, etc. qui contiendront vos chapitres, pour que cet exemple fonctionne.

4.3. Identification des fichiers avec les ID publics formels

Note : C'est une technologie avancée. Vous pouvez sauter cette étape si vous voulez vous contenter de l'information de base.



Chapitre 4. La déclaration de type de document (DTD)

Vous pouvez toujours identifier les fichiers à inclure par leur nom et leur chemin (identifiant système). Mais il y a plusieurs raisons pour lesquelles vous pouvez vouloir utiliser les ID publics, ou FPI. La principale raison est de rendre plus facile le déplacement de fichiers sans avoir à changer vos documents. Une autre raison est de rendre plus aisé l'échange de vos fichiers avec d'autres personnes où les répertoires de leurs systèmes peuvent être différents. Les FPI vous permettent également de revendiquer la propriété d'une information importante, comme votre DTD.

Avec les FPI, vous identifiez un fichier par un nom résumé dans votre document et ensuite indiquez l'endroit de ce fichier dans un catalogue, parfois appelé fichier mappé ou manager d'entité. Le catalogue est un autre fichier, séparé, de votre document.

Si un fichier utilisé dans votre document est déplacé, vous changez simplement sa localisation dans le catalogue plutôt que changer son emplacement dans votre document ou un autre document qui l'utilise. Si vous échangez des fichiers avec quelqu'un d'autre, ou simplement déplacez les fichiers vers un autre ordinateur avec diverses catégories, vous n'aurez qu'à modifier l'information de l'emplacement une seule fois dans le catalogue.

Les FPI doivent avoir une structure spécifique. Deux barres obliques sont utilisés pour marquer la séparation entre chaque partie de la structure, comme :

```
"Registration//Owner//Keyword Description//Language"
```

Registration

Le premier caractère indique si le FPI est formellement enregistré (+) ou non (-) avec un service d'enregistrement approuvé par l'ISO. Si vous définissez votre propre FPI et ne l'enregistrez pas, utiliser le trait d'union.

Owner

Le propriétaire du fichier est la seconde partie du FPI. Ce peut être une entreprise, une organisation ou une personne.

Keyword

Il existe plusieurs mots-clé indiquant le type d'information dans le fichier. Certains des mots-clé les plus communs sont DTD, ELEMENT, et TEXT. DTD utilisé seulement pour les fichiers DTD, ELEMENT est habituellement utilisé pour des fragments de DTD qui contiennent seulement des déclarations d'entité ou d'élément. TEXT sert pour le contenu SGML (texte et balises).

Description

Toute description que vous voulez fournir au sujet du fichier. Ceci peut inclure des numéros de version ou un court texte significatif pour vous et unique pour le système SGML.

Language

C'est un code ISO à deux caractères qui identifie le langage natif du fichier. EN est utilisé pour l'Anglais. Pour une DTD comme DocBook, c'est la langue du balisage de texte, pas la langue du document lui-même !



Pour utiliser ces Formal Public Identifiers (FPI) dans votre document, remplacer le mot-clé SYSTEM par PUBLIC dans la DTD au début de votre fichier. Ensuite, pour savoir où trouver le fichier, vous devrez établir une correspondance entre le FPI et le nom réel dans un "catalogue". Regardez les fichiers nommés "catalog" dans under /usr/share/sgml pour savoir comment ceci fonctionne.

4.4. Utiliser des sections marquées pour manipuler du contenu conditionnel

Note : C'est une technologie avancée. Vous pouvez sauter cette étape si vous voulez vous en tenir à l'information de base.

Faites attention que les sections marquées ne sont pas prises en charge par XML. Passez cette section si vous utilisez XML.

Parfois vous avez besoin d'avoir différentes versions pour divers propos. Il existe plusieurs moyens pour faire cela en utilisant SGML, et un de ceux-ci est appelé sections marquées. Un simple exemple de contenu conditionnel peut être la description de clés utilisées dans un programme où elles apparaissent dans des tableaux dans le manuel imprimé mais sont en bleu à l'intérieur de parenthèses sur le site Web ou sur le CD. Plutôt que d'avoir deux versions séparées pour chaque édition du manuel, vous pouvez utiliser les sections marquées pour conserver les différences dans le même document.

Il existe différents types de sections marquées, mais les types qui permettent de faire du contrôle de contenu conditionnel sont les sections ignore/include. Ces marqueurs agissent comme les commutateurs on/off pour permettre au contenu d'être inclu ou ignoré dans les différentes situations.

Le marqueur pour une section marquée include ressemble à ça :

```
<![INCLUDE [ keys are boxed, such as <key>F1</key> ]]>
```

tandis qu'une section marquée ignore sera :

```
<![IGNORE [ keys are blue inside brackets, such as <key>F1</key> ]]>
```

INCLUDE et IGNORE sont des mots-clé qui indiquent au système SGML quoi inclure ou ignorer. Comme l'exemple le montre, les sections marquées peuvent contenir du texte et des balises tant que les balises dans les marqueurs sont équilibrées (si une balise démarre dans un marqueur, elle finit dans le même marqueur).

Dans cet exemple, nous laissons les marqueurs pour la version imprimable à INCLUDE et les marqueurs pour la version électronique à IGNORE quand nous imprimons une copie principale. Quand nous créons un livre électronique ou HTML pour un site Web, nous changeons les marqueurs pour la version imprimable en IGNORE et les marqueurs pour la version électronique à INCLUDE. Ceci fonctionne bien, à moins que vous ayez différentes sections vous aurez alors besoin de include et ignore ensemble - c'est gênant de changer chaque fois manuellement et vous pouvez aisément faire une erreur.



Chapitre 4. La déclaration de type de document (DTD)

Ainsi au lieu de ça, vous pouvez définir des paramètres d'entités sous n'importe quel nom que vous voulez et ensuite changer les entités pour modifier les commutateurs include/ignore. Pour faire cela, vous ajoutez des paramètres de déclarations d'entités dans le sous-ensemble interne au début de votre document maître SGML. Par exemple :

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V4.4//EN" [  
...  
<!ENTITY % hardcopy "INCLUDE">  
<!ENTITY % softcopy "IGNORE">  
... ]>
```

Vous utilisez alors les noms pour chaque section marquée, comme suit :

```
<![%hardcopy; [ keys are boxed, such as <key>F1</key> ]]>  
<![%softcopy; [ keys are blue inside brackets, such as <key>F1</key> ]]>
```

Pour imprimer une copie principale, vous laissez les déclarations d'entités comme indiqué plus haut. Le système SGML interprète chaque %hardcopy; il trouve INCLUDE et inclut les sections marquées. Le %softcopy; est interprété comme IGNORE et les sections sont ignorées. Quand vous êtes prêt à éditer la version électronique, vous n'avez qu'à changer les déclarations d'entités au début de votre fichier, comme ceci :

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V4.4//EN" [  
...  
<!ENTITY % hardcopy "IGNORE">  
<!ENTITY % softcopy "INCLUDE">  
... ]>
```

Avec cette modification unique, les versions électroniques sont incluses et les versions imprimées ignorées.

Les sections marquées peuvent être simples, mais elles ne sont pas toujours le meilleur choix pour manipuler du texte conditionnel. Elles sont le meilleur choix pour gagner du temps et dans des situations très claires. Certains problèmes qui peuvent survenir sont :

- Les sections marquées peuvent être emboîtées (une section marquée dans une autre section marquée), mais ceci peut embrouiller votre système SGML et peut ne pas produire les effets escomptés. Par exemple, les systèmes SGML ne peuvent pas maintenir correctement une section marquée incluse dans une section marquée ignorée.
- Si vous utilisez beaucoup de sections nommées différentes, il est facile de perdre la trace du contenu et peut faire que votre document SGML soit invalide si certaines structures requises sont placées à IGNORE.
- Enfin, ce n'est pas géré en XML. Passer au XML peut être difficile si vous utilisez beaucoup les sections marquées.



Chapitre 5. Méta-informations

Les balises concernées par cette section sont listées ci-dessous.

bookinfo - Metainformation pour un livre
title - Texte d'un en-tête ou le titre d'un élément orienté-bloc
authorgroup - Information sur l'auteur
author - Auteur du document
authorinitials - Initiales ou autre identifiant pour l'auteur d'une révision ou d'un commentaire
firstname - Prénom
othername - Nom qui n'est pas un prénom, surnom ou nom de famille
surname - Nom de famille
keywordset - Ensemble de termes décrivant le contenu du document
keyword - Terme décrivant le contenu d'un document
releaseInfo - Information sur une version particulière du document
revhistory - Révisions d'un document
revision - Entrée dans revhistory, décrivant certaines révisions faites au texte
revnumber - Numéro d'une révision
revremark - Description d'une révision
abstract - Sommaire du document
date - Date de publication ou de révision du document

Exemple 5-1. Metainformation de DocBook

```
<bookinfo>
<title>The Twiddle Handbook</title>
<authorgroup>
<author>
<firstname>George</firstname>
<othername>N.</othername>
<surname>Ugnacious</surname>
</author>
</authorgroup>

<date>03/04/1999</date>
<releaseinfo>1.01.00</releaseinfo>

<abstract>
<para>
<application>twiddle</application> is an application specially
designed to do nothing you would ever want.
</para>
</abstract>

<keywordset>
<keyword>twiddle</keyword>
<keyword>sample application</keyword>
</keywordset>

</bookinfo>
```

La balise <bookinfo> contient toute la metainformation de votre document.



C'est l'information qui décrit votre document. Elle permet de :

- retrouver quelque chose de spécifique parmi beaucoup d'autres sur votre disque dur local
- imprime certaines informations sur le page de garde, sous le contrôle des feuilles de style
- permet de vous retrouver pour les moteurs de recherche si vous publiez en ligne au format HTML
- etc.



Chapitre 6. Listes

Les listes DocBook sont très similaires à leur contreparties en HTML sauf que DocBook contient des types de liste supplémentaires spécialisés.

Les balises concernées par ce chapitre sont listées ci-dessous.

simplelist - Liste de mots et de courtes phrases
member - Membre d'une liste
itemizedlist - Liste dans laquelle chaque élément est signalé avec une puce, un trait ou autre ornement typographique
listitem - Enveloppe pour les éléments d'articles dans une itemizedlist ou orderdlist
orderedlist - Liste dans laquelle chaque élément est signalé avec une étiquette qui augment à chaque fois
variablelist - Liste dans laquelle chaque élément est composé d'ensembles de un ou plusieurs termes associés avec les
varlistentry - Enveloppe pour un terme et sa listitem associée dans une variablelist
term - Terme attaché à une listitem dans une varlistentry dans une variablelist
segmentedlist - Liste d'ensembles d'information
segtitle - Titre qui appartient à un seg dans chaque seglistitem
seglistitem - Liste d'article dans une segmentedlist
seg - Composant d'une segmentedlist
qandaset - Ensemble de question-réponse
qandentry - Ensemble de question-réponse dans une qandaset
question - Une question dans une qandaset
answer - Une réponse à une question posée dans une qandaset
procedure - Liste d'opérations à exécuter
step - Partie d'une procédure
substeps - Enveloppe pour des étapes dans des étapes

6.1. La simplelist

La plus simple de toutes les listes à utiliser est la simplelist. Elle est destinée aux listes de phrases courtes (comme une liste de courses) et ne nécessite que deux balises comme vous pouvez le voir dans l'exemple ci-dessous. Une simplelist ne peut contenir d'autres listes.

Exemple 6-1. Une simplelist

```
<simplelist>  
<member>Apples</member>  
<member>Oranges</member>  
<member>Bananas</member>  
<member>Grapefruit</member>  
<member>Black Beans</member>  
</simplelist>
```

Lorsqu'elle est convertie, une simplelist ressemble à ça :

Apples



Oranges
Bananas
Grapefruit
Black Beans

6.2. La itemizedlist

Une itemizedlist est similaire à la simplelist sauf que chaque élément contient un paragraphe au lieu d'une simple phrase, vous permettant de placer d'avantage de contenu dans votre liste. Itemizedlist peut contenir d'autres listes.

Exemple 6-2. Une itemizedlist

```
<itemizedlist>  
<listitem><para>Apples - my favorite fruit.</para></listitem>  
<listitem><para>Oranges - yummy, but sticky.</para></listitem>  
<listitem><para>Bananas - they ripen too quickly!</para></listitem>  
<listitem><para>Grapefruit - great when eaten in halves.</para></listitem>  
<listitem><para>Black Beans - go well with rice.</para></listitem>  
</itemizedlist>
```

L'exemple ressemble à ça lorsqu'il est converti :

- Apples - my favorite fruit.
- Oranges - yummy, but sticky.
- Bananas - they ripen too quickly!
- Grapefruit - great when eaten in halves.
- Black Beans - go well with rice.

6.3. La orderedlist

La orderedlist est comme la itemizedlist sauf que chaque listitem est numéroté ou alphabétisé. L'attribut de numérotation spécifie quelle sorte de numérotage sera utilisé et peut être une des valeurs suivantes : arabique, alphanumérique majuscule ou minuscule, romain majuscule ou minuscule. Il existe plusieurs autres attributs qui contrôlent l'apparence d'une orderedlist, voir la Référence DocBook pour les détails. Orderedlist peut contenir d'autres listes.



Exemple 6-3. Une orderedlist

```
<orderedlist numeration="arabic">  
<listitem><para>Wake up.</para></listitem>  
<listitem><para>Eat Breakfast.</para></listitem>  
<listitem><para>Take a shower.</para></listitem>  
<listitem><para>Contemplate my navel.</para></listitem>  
<listitem><para>Go to Sleep.</para></listitem>  
</orderedlist>
```

L'exemple, lorsqu'il est converti, ressemble à ça :

1. Wake up.
2. Eat Breakfast.
3. Take a shower.
4. Contemplate my navel.
5. Go to Sleep.

6.4. La variablelist

La variablelist est similaire à une définition de liste HTML. Elle est utilisée quand vous avez une liste de termes et de définitions. Elle consiste en plusieurs balises : <varlistentry>, utilisées pour grouper des termes ensemble, <term>, contient le terme et <listitem>, contient la définition du terme.

Exemple 6-4. Une variablelist

```
<variablelist>  
<varlistentry>  
<term>Black Beans</term>  
<listitem><para>My favorite black bean recipe is black bean  
soup, but they also go well with rice.</para></listitem>  
</varlistentry>  
<varlistentry>  
<term>Apples</term>  
<term>Bananas</term>  
<listitem><para>You can eat them straight, but they also go  
well in salads and in desserts.</para></listitem>  
</varlistentry>  
</variablelist>
```

Une fois converti, l'exemple ressemble à ça :



Black Beans

My favorite black bean recipe is black bean soup, but they also go well with rice.

Apples

Bananas

You can eat them straight, but they also go well in salads and in desserts.

6.5. La segmentedlist

les segmentedlists sont utilisées pour lister l'information dans des champs distincts comme le contenu d'un carnet d'adresses. Le nom de chaque champ est placé dans une balise <segtitle>. Ensuite, utilisez la balise <seglistitem> pour démarrer et finir chaque ensemble de données. Les données sont placées dans la balise <seg>.

Exemple 6-5. Une segmentedlist

```
<segmentedlist>
<segtitle>Name</segtitle>
<segtitle>Occupation</segtitle>
<segtitle>Favorite Food</segtitle>
<seglistitem>
<seg>Tux</seg>
<seg>Linux mascot</seg>
<seg>Herring</seg>
</seglistitem>
<seglistitem>
<seg>Konqui</seg>
<seg>The KDE Dragon</seg>
<seg>Gnomes</seg>
</seglistitem>
</segmentedlist>
```

Une fois converti, ça donne :

Name: Tux

Occupation: Linux mascot

Favorite Food: Herring

Name: Konqui

Occupation: The KDE Dragon

Favorite Food: Gnomes



6.6. qandaset

Le qandaset est spécialisée dans les listes de questions et réponses, comme vous pourriez le voir dans une FAQ. Chaque ensemble de questions et réponses est contenu dans une balise `<qandaentry>`. Les balises `<question>` et `<answer>` contiennent les questions et réponses respectivement.

Exemple 6-6. Une qandaset

```
<qandaset>
<qandaentry>
<question>
<para>What are little boys made of?</para>
</question>
<answer>
<para>Snips and snails and puppy dog tails.</para>
</answer>
</qandaentry>
<qandaentry>
<question>
<para>What are little girls made of?</para>
</question>
<answer>
<para>Sugar and spice and everything nice.</para>
</answer>
</qandaentry>
</qandaset>
```

Une fois converti, ça donne :

1. What are little boys made of?

Snips and snails and puppy dog tails.

2. What are little girls made of?

Sugar and spice and everything nice.

6.7. Procédures

les listes procedure sont destinées à lister étape par étape des procédures comme vous pourriez le voir dans une recette de cuisine ou un HowTo Linux.



Exemple 6-7. Une liste procedure

```
<procedure>
<title>Waking Up</title>
<para>This is what you must do to awaken.</para>
<step performance="required">
<para>
Bring yourself to a hypnopompic state, either from an ongoing dream or by use of
your internal clock. You may feel unable to move, but you will no longer be
dreaming. </para>
<para>Now you are ready for real-world readjustment.</para>
<substeps>
<step performance="optional">
<para>Roll over.</para>
</step>
<step performance="required">
<para>Squint out of one eye.</para>
</step>
</substeps>
</step>
<step performance="required">
<para>Yawn and rise from your bed.
</para>
</step>
</procedure>
```

L'exemple, une fois converti, ressemble à ça :

Waking Up

This is what you must do to awaken.

1. Bring yourself to a hypnopompic state, either from an ongoing dream or by use of your internal clock. You may feel unable to move, but you will no longer be dreaming.

Now you are ready for real-world readjustment.

- a. Roll over.
 - b. Squint out of one eye.
2. Yawn and rise from your bed.



Chapitre 7. Les tableaux

Les balises concernées sont listées ci-dessous.

table - Tableau dans un document
informaltable - Tableau sans titre
thead - En-tête d'une ligne d'un tableau
tfoot - Pied d'une ligne d'un tableau
tgroup - Enveloppe de la partie d'un tableau qui contient une colonne avec ses informations de mise en forme
tbody - Enveloppe des lignes d'un tableau ou informaltable
row - Ligne dans un tbody, thead ou tfoot
entry - Cellule dans un tableau
entrytbl - Sous-tableau apparaissant comme une cellule de tableau

Les tableaux sont utilisés pour organiser les données dans un format de colonne avec des titres optionnels, des en-têtes et des pieds. Les tableaux DocBook sont de deux sortes : la table, qui nécessite un titre et la informaltable, qui ne doit pas avoir de titre. Toutes les autres caractéristiques de ces deux types de tableaux sont identiques.

Un tableau est une mise en forme d'informations et de blocs de données. Il y a quelques attributs qui peuvent être ajustés pour améliorer l'apparence de vos données. Ce didacticiel ne présente que les attributs de formatage de base. Pour plus de détails, voir la Référence DocBook.

Les tableaux débutent avec les balises <table> ou <informaltable>. Ensuite définissez un titre en utilisant la balise <title> si vous vous servez d'un tableau simple. Enfin, vous placez la balise <tgroup> qui contient les en-têtes, pieds et lignes. Vous pouvez avoir plus d'un tgroup si vous voulez changer les options de mise en forme pour une section du tableau. La balise <tgroup> possède plusieurs paramètres de mise en forme optionnels, mais l'attribut COLS, qui spécifie le nombre de colonnes, est nécessaire. Les <thead>, <tfoot> et <tbody> contiennent les données de votre tableau. Les données dans thead apparaissent au début du tableau, tbody apparaît au milieu et tfoot en fin de tableau.

Les données d'un tableau sont contenues dans les lignes et les éléments, désignées par les balises <row> et <entry> respectivement. Utilisez la balise <row> pour débiter une ligne, la balise <entry> pour chaque donnée et une balise de fermeture <row> pour terminer chaque ligne. Si vous voulez insérer un tableau dans un autre tableau, vous devez utiliser la balise <entrytbl>.

Exemple 7-1. Un tableau

```
<table>
<title>Kilométrage de la souris</title>
<tgroup cols="3">
<thead>
<row>
<entry>Mois</entry>
<entry>Semaine</entry>
<entry>Mètres parcourus</entry>
</row>
```



```
</thead>
<tfoot>
<row>
<entry>Total</entry>
<entry></entry>
<entry>1753</entry>
</row>
</tfoot>
<tbody>
<row>
<entry>Août</entry>
<entry>1</entry>
<entry>987</entry>
</row>
<row>
<entry>Août</entry>
<entry>2</entry>
<entry>657</entry>
</row>
<row>
<entry>Août</entry>
<entry>3</entry>
<entry>109</entry>
</row>
</tbody>
</tgroup>
</table>
```

Une fois converti ça donne cela :

Tableau 7-1. Kilométrage de la souris

Mois	Semaine	Mètres parcourus
Août	1	987
Août	2	657
Août	3	109
Total		1753



Chapitre 8. Les graphiques

Ci-dessous, le liste des balises concernées :

screeninfo - Information sur la capture d'écran

screenshot - Représentation sur la façon dont l'utilisateur voit une capture d'écran

mediaobject - Une image, un son, un texte qui peut être encodé en différents formats en même temps, non interprété da

inlinemediaobject - Une image, un son, un texte, qui peut être encodé en différents formats en même temps, sera interp

imagedata - Un des formats d'encodage d'image, comme EPS pour l'impression et JPG pour la mise en ligne

La documentation pour les applications graphiques demandent des captures d'écran, des icônes, des boutons et d'autres éléments graphiques. DocBook possède des balises pour les captures d'écran, les graphiques et les graphiques dans le corps du texte. Les exemples ci-dessous contiennent la même image en tant que capture d'écran et graphique dans le corps du texte.

Exemple 8-1. Un mediaobject dans le corps du texte

```
<para>
Voici un tas de rectangles
<inlinemediaobject>
<imageobject> <imagedata fileref="rectangles.eps" format="EPS" /> </imageobject>
<imageobject> <imagedata fileref="rectangles.png" format="PNG" /> </imageobject>
<textobject> <phrase>Un tas de rectangles</phrase> </textobject>
</inlinemediaobject>
</para>
```

La balise `<inlinemediaobject>` est affichée à côté du texte. L'attribut `fileref` de la balise `<imagedata>` contient le nom du fichier graphique, le format contient le type de fichier graphique et l'attribut optionnel `align` modifie l'alignement du graphique. Converti, l'exemple donne ceci :

Voici un tas de rectangles



Exemple 8-2. Une capture d'écran

```
<screenshot>
<screeninfo>Colored Rectangles</screeninfo>
<mediaobject>
<imageobject> <imagedata fileref="rectangles.eps" format="EPS" /> </imageobject>
<imageobject> <imagedata fileref="rectangles.png" format="PNG" /> </imageobject>
<textobject> <phrase>A bunch of rectangles</phrase> </textobject>
<caption>
<para>Here are a bunch of rectangles</para>
</caption>
</mediaobject>
</screenshot>
```

La balise `<screenshot>` désigne une capture d'écran avec la balise `<screeninfo>` qui fournit une description textuelle. Le coeur de la capture d'écran est la balise `<mediaobject>` qui pointe vers le fichier graphique comme indiqué dans le premier exemple. Une capture d'écran, une fois convertie donc ceci :



Here are a bunch of rectangles



Chapitre 9. Les liens

Liste des balises concernées par cette section :

- anchor - Spot dans le texte
- email - Adresse email dans une adresse
- link - Lien hypertexte
- ulink - Lien adressant sa cible par une URL (Uniform Resource Locator).

Les liens en DocBook sont très similaires au HTML. La balise `<link>` est utilisée pour lier certains éléments avec un ID dans un document et peut aussi être utilisée pour lier d'autres fichiers locaux. La balise `<ulink>` sert lorsque vous avez besoin de lier vers une URL. La balise `<email>` est une forme spécialisée de balise `ulink` pour les adresses email. `<anchor>` sert à baliser un point dans le texte que vous voulez référencer plus tard avec un lien.

Exemple 9-1. Plusieurs sortes de liens

```
<para id="mylink">
This is a paragraph that will be linked to with a link tag. Oh, and by
the way, check out <ulink url="http://www.kde.org">my favorite web
site</ulink> while you are here. <anchor id="anotherlink"/> Kilroy was
here.</para>
<para>
The above paragraph can be located <link linkend="mylink">here</link>.
My email address is <email>konqui@kde.org</email></para>
```

Notez que les liens utilisent l'attribut `linked` et `ulink` utilise l'attribut `URL` pour leurs cibles. L'exemple ci-dessus, une fois converti, donne :

This is a paragraph that will be linked to with a link tag. Oh, and by the way, check out my favorite web site (<http://www.kde.org>) while you are here. Kilroy was here.

The above paragraph can be located here. My email address is konqui@kde.org



Chapitre 10. Description de l'interface d'une application

10.1. Exemples

Liste des balises concernées par cette section :

literallayout - Enveloppe pour l'ensemble des lignes hors du texte principal qui ne sont pas balisées comme écrans, exemple - Exemple d'un programme d'ordinateur ou une information relative
informalexample - Exemple sans titre
programlisting - Listing de toutes les parties d'un programme
screen - Texte que l'utilisateur voit sur son écran

Il existe plusieurs situations dans lesquelles vous devez intégrer des exemples de code source, commandes ou actions en interface graphique dans votre documentation. DocBook possède plusieurs balises pour ces besoins. Chaque fois que vous voulez citer des exemples dans votre document, placez simplement une balise `<example>` ou `<informalexample>` aux environs de votre texte ou graphique d'exemple.

Exemple 10-1. Un exemple

```
<example>
<title>A BASIC Example</title>
<programlisting>
10 PRINT "HELLO WORLD"
20 GOTO 10
</programlisting>
</example>
```

Dans le premier exemple, nous avons présenté un programme simple en BASIC. Le code contenu dans la balise `<programlisting>` est présenté avec l'espacement et les coupures de lignes intacts ce qui est très utile pour des exemples de code et situations similaires dans lesquelles vous devez préserver la mise en forme exacte. Les balises `LiteralLayout` et `Screen` fonctionnent de la même façon, mais sont utilisées pour indiquer différents types de contenu. `Screen` contient des sorties qui apparaîtront à l'écran, tandis que `LiteralLayout` sert pour le texte qui doit être rendu avec les coupures et les tabulations.

L'exemple ressemblera à ça :

Exemple 10-2. Un programlisting

```
10 PRINT "HELLO WORLD"
20 GOTO 10
```



Un problème peut se produire avec les balises `LiteralLayout`, `ProgramListing` et `Screen` : tout le texte est rendu littéralement, mais les balises DocBook sont interprétées comme balises et non comme texte. Que devez vous faire quand vous voulez montrer du texte sans avoir les balises interprétées ? La réponse est d'utiliser `<![CDATA[]]>`, qui désigne le texte contenu dans les parenthèses comme données et ne sera pas interprété par l'analyseur SGML. Tout texte entre parenthèses sera rendu tel quel après la conversion, ainsi l'exemple ci-dessus reproduira correctement les balises.

Exemple 10-3. Utilisation de CDATA

```
<example>
<title>Du balisage</title>
<screen>
<![CDATA[
<para>Ceci est un exemple DocBook.</Para>
]]>

</screen>
</example>
```

Une fois converti :

Exemple 10-4. Du balisage

```
<para>Ceci est un exemple DocBook.</para>
```

10.2. Éléments d'interface GUI

Accel - Touche utilisée avec une touche meta pour activer une Interface Utilisateur Graphique (GUI)
KeyCap - Texte imprimé par une touche physique de clavier d'ordinateur, pas nécessairement la même chose que KeyC
KeyCode - Désignation numérique d'une touche sur un clavier d'ordinateur
KeyCombo - Combinaison d'actions de saisie
KeySym - Nom de symbole de touche, pas nécessairement la même chose que KeyCap
MenuChoice - Sélection de menu ou de séries
MouseButton - Nom conventionnel d'un bouton de souris
Interface - Élément de GUI
InterfaceDefinition - Nom complet ou court d'une spécification formelle d'un GUI
GUIButton - Texte sur un bouton dans un GUI
GUIIcon - Graphique et/ou texte apparaissant comme icône dans un GUI
GUILabel - Texte dans un GUI
GUIMenu - Nom de menu dans un GUI
GUIMenuItem - Nom d'un menu de terminal dans un GUI
GUISubmenu - Nom d'un sous-menu dans un GUI
Action - Fonction invoquée en réponse à une action utilisateur

On peut parfois dire qu'il y a de trop nombreuses balises dans DocBook pour décrire les éléments de



GUI. La plupart des balises ci-dessus peuvent être utilisées dans une variété de contextes, mais quelques unes, comme `<keycap>`, doivent être utilisées dans d'autres balises. L'exemple ci-dessous ne présentera pas toutes les balises décrites au-dessus. Cette liste est faite pour votre commodité car la Référence DocBook ne groupe pas les balises par leur fonction.

Toutes les balises GUI peuvent être utilisées dans un contexte de paragraphe normal. Si vous voulez parler d'une icône de corbeille ou d'un bouton de corbeille vide, vous utiliserez simplement les balises `<guiicon>` et `<guibutton>` comme ceci : `<guiicon>Trash</guiicon>` icon, `<guibutton>Empty Trash</guibutton>` button. Notez que toutes les balises GUI peuvent aussi contenir des graphiques dans le corps du texte.

Ci-dessous, un exemple un peu plus compliqué d'une utilisation de balise GUI.

Exemple 10-5. guimenu et shortcut

```
<variablelist>
<varlistentry>
<term><menuchoice>
<shortcut>
<keycombo><keycap>Ctrl</keycap><keycap>n</keycap></keycombo>
</shortcut>
<guimenu>File</guimenu>
<guimenuitem>New</guimenuitem>
</menuchoice></term>
<listitem><para><action>Creates a new document</action></para></listitem>
</varlistentry>
<varlistentry>
<term><menuchoice>
<shortcut>
<keycombo><keycap>Ctrl</keycap><keycap>s</keycap></keycombo>
</shortcut>
<guimenu>File</guimenu>
<guimenuitem>Save</guimenuitem>
</menuchoice></term>
<listitem><para><action>Saves the document</action></para></listitem>
</varlistentry>
<varlistentry>
<term><menuchoice>
<shortcut>
<keycombo><keycap>Ctrl</keycap><keycap>q</keycap></keycombo>
</shortcut>
<guimenu>File</guimenu>
<guimenuitem>Quit</guimenuitem>
</menuchoice></term>
<listitem><para><action>Quits</action> application>Kapp</application></para></listitem>
</varlistentry>
</variablelist>
```



J'espère que vos yeux ne clignent pas à la vue de toutes ces balises ! La partie la plus compliquée de cet exemple est la balise `<Shortcut>` qui désigne les raccourcis clavier des éléments de menus. Les raccourcis contiennent soit une `KeyCombo` ou une simple `KeyCap` contenant la touche ou le groupe de touches à presser pour invoquer le menu depuis le clavier. Il est important d'utiliser les balises `KeyCombo` et `KeyCap` dans les balises `Shortcut` car il est incorrect d'utiliser les caractères de données (le texte `Ctrl-q` par exemple) dans un raccourci. .

D'autres balises de l'exemple qui doivent être mentionnées sont `menuchoice`, `action` et `application`. `Menuchoice` désigne un choix de menu et contient en raccourci (si besoin est) le nom du menu dans le GUI et le nom du menuitem dans `guimenuitem`. `Action` désigne une phrase qui décrit ce que fait `menuitem` (ou un autre élément d'interface). `Application` est une balise utilisée pour désigner les noms d'applications.

L'exemple, une fois converti :

File→New (**Ctrl-n**)

Creates a new document

File→Save (**Ctrl-s**)

Saves the document

File→Quit (**Ctrl-q**)

Quits Kapp

10.3. Éléments de ligne de commande

Les balises suivantes sont utilisées pour désigner les éléments d'une commande:

`type` - Classification d'une valeur

`literal` - Chaîne littérale, utilisée dans le corps du texte, qui est une partie de donnée dans un ordinateur

`userinput` - Donnée saisie par l'utilisateur

`symbol` - Nom remplacé par une valeur avant l'exécution

`replaceable` - Contenu pouvant être remplacé dans un sommaire ou une ligne de commande

`filename` - Nom d'un fichier, avec possibilité de préciser le chemin

`prompt` - Caractère indiquant le début d'un champ saisi

`paramdef` - Information sur le type de donnée et le nom du paramètre qui s'applique à cette information

`parameter` - Partie d'une instruction à un ordinateur

`option` - Option pour une commande de programme d'ordinateur

`envar` - Variable d'environnement

`command` - Programme exécutable ou la saisie effectuée par l'utilisateur pour exécuter une commande

`cmdsynopsis` - Sommaire d'une commande

`arg` - Argument dans `cmdsynopsis`

`computeroutput` - Donnée présentée à l'utilisateur par l'ordinateur

`funcsynopsis` - Syntaxe de sommaire pour une définition de fonction

`funcsynopsisinfo` - Information générale sur l'utilisation de la fonction

`funcprototype` - Prototype de la fonction

`funcdef` - Nom et valeur de retour de la fonction

`function` - Nom de la fonction

`paramdef` - Nom et type du paramètre d'une fonction

`parameter` - Nom du paramètre d'une fonction



Il existe deux situations dans lesquelles vous pouvez vouloir décrire une commande : montrer un exemple d'un type de ligne de commande et une description détaillée de tous les arguments et options d'une commande comme vous pouvez le voir dans les pages de manuel.

DocBook gère ces deux contextes avec les balises `<command>` et `<cmdsynopsis>`.

Exemple 10-6. Une commande et sa sortie

```
<screen>
<prompt>bash$</prompt> <command>tourneboule -c 1 <replaceable>monfichier</replaceable>
</command>
tourneboulage de monfichier.....accompli !
</screen>
```

Qui apparaîtra comme :

```
bash$ tourneboule -c 1 monfichier
tourneboulage de monfichier.....accompli !
```

La balise `command` peut également être utilisée à l'intérieur d'un paragraphe pour baliser le nom d'une commande. Par exemple :

```
The <command>twiddle</command> command is used to twiddle
files. Twiddled files will be marked with the .twid extension, so if I <command>twiddle<
<replaceable>myfile</replaceable> then it will become
<replaceable>myfile.twid</replaceable>. Errors are written to the
file <filename>twiddle.err</filename>.
```

"La commande **twiddle** est utilisée pour manier les fichiers. Les fichiers manipulés seront marqués avec l'extension `.twid`, ainsi si je fais **twiddle** `myfile` il deviendra `myfile.twid`. Les erreurs sont écrites dans le fichier `twiddle.err`".

La balise `<prompt>` est utilisée pour désigner l'invite de commande d'une ligne de commande. `Replaceable` désigne le texte qui sera remplacé par l'utilisateur. Dans l'exemple, `myfile` est juste un nom arbitraire pour un fichier car nous ne connaissons pas le nom du fichier, nous voulons simplement montrer comment fonctionne la commande. Si un nom de fichier dans une commande est connu, nous utilisons la balise `<filename>` à la place.

Baliser une `cmdsynopsis` est un peu plus difficile. Ici un exemple tiré de la Référence DocBook :

Exemple 10-7. Une `cmdsynopsis`

```
<cmdsynopsis>
  <!-- This is a synopsis for the command foo.
       The options -a and -x are optional and exclusive
       The option -c takes a cheese and is optional and repeatable
```



Chapitre 10. Description de l'interface d'une application

The options -t and -k are referred to in another fragment
The options -i, -j, and -k are required and exclusive
The option -f takes a filename and is required
The -t and -k options specify the kind of milk and mold in an optional and repeatable group

```
-->
<command>foo</command>
<group>
  <arg>-a</arg>
  <arg>-x</arg>
</group>
<group>
<arg rep="repeat">-c <replaceable>cheese</replaceable></arg>
<synopfragmentref linkend="cheesetype">cheesetype</synopfragmentref>
</group>
<group choice="req">
  <arg>-i</arg>
  <arg>-j</arg>
  <arg>-k</arg>
</group>
<arg choice="req">-f <replaceable>filename</replaceable></arg>
<synopfragment id="cheesetype">
  <group rep="repeat">
    <arg>-t <replaceable>milk</replaceable></arg>
    <arg>-k <replaceable>mold</replaceable></arg>
  </group>
</synopfragment>
</cmdsynopsis>
```

Qui donnera ceci :

```
foo [-a | -x] [-c cheese ❶cheesetype] {-i | -j | -k} {-f filename}
```

```
❶ [-t milk | -k mold]...
```

Une cmdsynopsis contient une commande, des groupes d'arguments relatifs, des arguments indépendants et synopfragments. Le <arg> désigne les arguments de la commande. arg a deux attributs : choice et rep. choice est utilisé pour indiquer soit que la balise est optionnelle (par défaut), requise (req), soit qu'elle sera affichée sans aucune décoration (plain). La balise <group> sert à grouper les args relatifs. synopfragment est la plus compliquée des balises cmdsysopsis. Elle est utilisée pour fournir une description plus détaillée des options d'un argument. Une synopfragment consiste en deux parties : le synopfragment, qui contient les Args additionnels et la synopfragmentref qui pointe vers la description détaillée.

10.4. Description d'une API

DocBook possède des moyens plutôt détaillés pour baliser les descriptions de comportement d'une fonction. La balise concernée est <funcsynopsis>. Voici un exemple :



Exemple 10-8. Description d'une fonction dans la bibliothèque C API

```
<functsynopsis>  
<functsynopsisinfo>#include <stdlib.h></functsynopsisinfo>  
<funcprototype>  
<funcdef>double <function>atof</function></funcdef>  
<paramdef>const char *<parameter>nptr</parameter></paramdef>  
</funcprototype>  
</functsynopsis>
```

Qui ressemble à ça :

```
#include <stdlib.h>  
double atof(const char *nptr);
```



Chapitre 11. Diverses balises utiles

11.1. Balises de désignation

Application - Nom d'un logiciel

Markup - Balisage dans le texte, ce que nous voulons représenter littéralement

La balise application désigne le nom de l'application (programme-logiciel) La balise <markup> sert à désigner le texte balisé, comme les balises HTML ou TeX.

11.2. Balises de mise en forme

emphasis - texte mis en évidence

subscript - Sous-script

superscript - Superscript

Ces balises s'expliquent d'elles-mêmes. *This is an emphasized sentence with ^{superscripted} and _{subscripted} text.*

11.3. Avertissements, conseils et notes

Les balises suivantes sont utilisées pour séparer les paragraphes du reste du texte.

caution - Avertissement séparé du texte

important - Avertissement séparé du texte

note - Message à l'utilisateur, séparé du texte

tip - Suggestion à l'utilisateur, séparé du texte

warning - Avertissement séparé du texte

Exemple :

```
<warning><para>Danger, Will Robinson!</para></warning>
```

Ce qui ressemblera à ça :





Chapitre 12. Où aller ensuite ?

Il existe plusieurs parties utiles de DocBook qui n'ont pas été traitées dans ce didacticiel. Elles concernent les pages Callouts, Indexes, Glossaries et Reference. Si vous avez besoin de ces fonctionnalités, vous pouvez toujours consulter la Référence DocBook et/ou acheter un livre sur DocBook.

12.1. Ressources sur DocBook

Ci-dessous une liste de ressources qui vous aiderons à démarrer dans ce domaine.

- DocBook technical committee (<http://www.oasis-open.org/docbook/>) page chez OASIS
- DocBook - the Definitive Guide (<http://www.oreilly.com/catalog/docbook/chapter/book/docbook.html>) livre en ligne
- DocBook 3.1 Quick Reference (<http://www.oreilly.com/catalog/docbook/chapter/book/quickref.html>)
- Norman Walsh DocBook Site (<http://nwalsh.com/docbook/>)
- Jorge Godoy DocBook HOWTO (<http://docbook.godoy.homeip.net:81/sgml/docbook/howto/>)
- Mark Galassi DocBook Intro (<http://nis-www.lanl.gov/~rosalia/mydocs/docbook-intro.html>)
- FreeBSD Documentation Primer (http://www.freebsd.org/doc/en_US.ISO8859-1/books/fdp-primer/index.html)
- DocBook Install mini-HOWTO (<http://www.tldp.org/HOWTO/mini/DocBook-Install/index.html>)
- utile si vous voulez compiler vous-même vos outils
- Bob DuCharme PSGML tips (<http://www.snee.com/bob/sgmlfree/emcspsgm.html>) pour Emacs



Annexe A. Licence

This document is released under Free Documentation licence; the terms of this licence are detailed below.

A.1. Free Documentation Licence

GNU Free Documentation License
Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".



A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.



2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the



Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document,



- unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitile any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.



In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this



License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.



Annexe A. Licence

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.



Annexe B. Mode PSGML de Emacs

Il existe peu d'éditeurs SGML. Un des plus connu est Emacs avec le mode PSGML. Il possède la complétion *sur les balises*.

Ici, quelques points sur la façon d'utiliser ce mode. En premier installer le paquetage "psgml", ensuite charger un fichier DocBook dans Emacs.

Note : Emacs avec le mode PSGML ne gère pas les noms de fichiers se terminant par .docbook. Votre nom de fichier doit finir par .sgml ou .xml comme *myfile.sgml*.

Note : (NdT : Je ne sais pas pour Emacs, mais les extensions .docbook fonctionnent très bien avec XEmacs et le mode PSGML, en tous cas dans la version 21.4.19 sous Fedora 4).

Si vous tapez "C-c C-e" il vous demandera un élément et permettra la complétion seulement si l'élément est valide.

À titre d'exemple, j'ai tapé :

```
C-c C-e variab<SPACE BAR><RETURN>
```

et il a inséré ce texte dans le tampon :

```
<variablelist>
  <varlistentry>
    <term></term>
    <listitem>
      <!-- one of (epigraph authorblurb abstract highlights comment bridgehead anchor sidebar procedure ms-
gset table figure example equation informatable informalexample informalequation graphicco graphic block-
quote address simpara para formalpara funcsynopsis cmdsynopsis synopsis screenshot screenco screen pro-
gramlistingco programlisting literallayout warning tip note important caution variablelist simplelist seg-
mentedlist orderedlist itemizedlist glosslist calloutlist) -->
    </listitem>
  </varlistentry>
</variablelist>
```

Un autre exemple :



`C-c C-e i<SPACE BAR>`

et il m'indique les complétions suivantes :

Click mouse-2 on a completion to select it.

In this buffer, type RET to select the completion near point.

Possible completions are:

important	indexterm
informalequation	informalexample
informaltable	itemizedlist



(American Standard Code for Information Interchange) Schéma standard d'encodage de caractères, utilisé extensivement dans la transmission de données.

ANSI

(American National Standards Institute) Groupe U.S membre de l'organisation qui maintient l'ISO, International Organization for Standardization.

attribut

Un attribut fournit d'avantage d'informations sur un élément comme niveau de classification, identifiants de référence unique, ou mise en forme d'informations.

CCITT Group 4

(International Consultative Committee on Telegraphy and Telephony) Standard graphique CALS qui permet d'alléger les graphiques très lourds. Format CCITT/4 compressé.

CITIS

(Contractor Integrated Technical Information Service) Partie de CALS phase II, CITIS est un brouillon de spécification fonctionnelle pour les services.

CGM

(Computer Graphics Metafile) CGM est un des formats standard CALS pour représenter les illustrations techniques en 2-D. CGM est un format graphique orienté objet.

DSSSL

(Document Style Semantics and Specification Language) Langage de feuilles de style pour les documents SGML dérivé du langage Scheme et normalisé sous le numéro ISO/IEC 10179:1996.



DTD

(Document Type Definition) La DTD est la définition formelle des éléments, structures et règles pour le balisage de documents SGML ou XML. Vous pouvez placer une DTD au début d'un document ou à l'extérieur dans un fichier séparé.

EDI

(Electronic Data Interchange) Ensemble de standards d'échange entre ordinateurs pour des documents de travail comme factures, ordres d'achat, etc.

élément

Un élément est une partie de donnée dans un document qui contient soit du texte soit d'autres sous-éléments comme un paragraphe, un chapitre, etc.

déclaration d'élément

Une instruction dans la DTD définissant un élément et déclarant l'ordre dans lequel il peut apparaître et quel autre élément il peut contenir.

entité

Une entité est une partie de donnée auto-contenue qui peut être référencée comme une unité. Vous pouvez vous référer à une entité par un nom symbolique dans la DTD ou le document. Une entité peut être une chaîne de caractères, un caractère symbole (indisponible sur un clavier standard), un fichier texte séparé ou un fichier graphique séparé.

déclaration d'entité

Une instruction dans la DTD ou le document qui assigne un nom SGML ou XML à une entité que vous pouvez référencer.

FOSI

(Formatting Output Specification Instance) Un FOSI est utilisé pour mettre en forme des documents SGML pour l'impression et d'autres sorties. C'est un fichier séparé qui contient les informations de mise en forme pour chaque élément du document.



HTML

(HyperText Markup Language) Format de fichiers publiés sur le World Wide Web. HTML est une application de SGML. Pour publier en HTML en utilisant le SGML, vous avez simplement besoin d'une DTD HTML.

IGES

(Initial Graphics Exchange Specification) IGES est un des standards graphiques CALS.

Internet

Réseau de communication mondial, à l'origine développé aux U.S.A.

ISO

(International Organization for Standardization) L'ISO est une organisation soutenue par l'industrie qui établit des standards mondiaux pour beaucoup de choses, depuis le format d'échange de données jusqu'aux spécifications de vitesse de films.

balisage

Le balisage est ce qu'on ajoute au contenu d'un document pour décrire le texte.

analyseur

Un analyseur (*parser*) est un programme spécialisé qui reconnaît le balisage SGML ou XML dans un document.

balise

Dans le monde SGML et XML, une balise est un marqueur placé dans un document qui indique le propos ou la fonction de l'élément. Chaque élément possède une balise de début et une de fin.

World Wide Web

Souvent appelé WWW ou le Web, se réfère habituellement aux informations disponibles sur Internet et auxquelles on peut facilement accéder avec un logiciel appelé « navigateur. » Les organisations publient leurs informations sur le Web dans un format connu comme HTML.



XSLT

(eXtended Style Sheet transformations) Langage de feuilles de style pour les documents XML et qui est lui-même un dialecte de XML.