

# DOC TP FINAL DOCBOOK

**Auteurs: Sylvie-Auneau/Nicolas-Evano**

sommaire:

1.0-Introduction:

2.0-Définition de la docbook:

3.0-Les outils utilisés pour le TP:

4.0-Le choix de la méthode de génération des documents finaux:

5.0-La méthode de travail utilisée:

6.0-Rappel du fonctionnement XSL-FO

7.0-Le processus de construction des documents:

8.0-Conclusion:

## 1.0-Introduction:

L'objectif est d'utiliser le format [docbook](#) pour générer des documents finaux aux formats PDF et générer des documents [XHTML](#). Les documents seront produits à partir de feuille de style XSLT et seront conformes à la DTD de la [docbook](#).

## 2.0-Définition de la docbook:

La [docbook](#) est à l'origine une définition de type de document DTD développée par l'éditeur O'Reilly .Elle a depuis été portée sous forme de schéma XML ainsi que de règles relax NG pour la version 5. [Docbook](#) est standardisée et maintenue par le DocBook Technical Comittee du consortium OASIS. Le format [docbook](#) peut être utilisé pour tout type de documentation technique.

## 3.0-Les outils utilisés pour le TP:

Pour la mise en oeuvre nous avons utilisé les outils suivants:

-[Ant](#) version 1.7.1 pour exécuter les commandes de transformation de document à partir de feuille xslt.

-Fop0.95 pour générer des documents pdf à partir de de fichier fo.

-Le processeur xslt de la version standart de java1.6.

-Le processeur xslt de [saxon6-5-5](#).

- Le package docbook-xsl-1.76.1 contient les feuilles de styles utilisées pour effectuer la transformation de document [docbook](#) vers [XHTML-1.1](#).
- L'IDE [eclipse galileo 3.5](#).
- Adobe reader 9.3.4 pour visualiser le résultat en PDF.
- Chrome 15.0.874.121 pour visualisé le résultat [XHTML-1.1](#).
- La DTD 4.5 de la [docbook](#)
- Svn 1.6.0 pour versionner le projet sur la plateforme code.google.com.

## 4.0-Le choix de la méthode de génération des documents finaux:

Nous avons choisi d'utiliser le processeur FO plutôt que LaTeX pour la génération des documents finaux.

[XSL-FO](#) est un outil puissant pour générer des documents imprimable d'autre part le format s'adapte bien avec la [docbook](#) et les feuilles de style XSLT existe déjà.

De plus nous avons déjà eu l'opportunité d'utiliser [XSL-FO](#) lors d'un TP précédent aussi il nous a paru intéressant d'explorer certaines des possibilités du format non exploitées durant le TP. Donc nous avons développé notre propre feuille de style pour générer un fichier fo à partir d'un document xml [docbook](#).

## 5.0-La méthode de travail utilisée:

Nous avons étudié la documentation de la [docbook](#) et déterminé quels sont les éléments disponibles et exploitables pour le compte-rendu.

la [docbook](#) définit deux formats de document:

- book pour les grands documents destinés aux livres.
- article pour les documents plus concis; celui utilisé dans notre cas.

Il est possible de générer un document [XHTML](#) avec un document [docbook](#) par l'emploi du package docbook-xsl, conjointement avec le processeur xslt de [saxon](#). Ainsi nous générons des documents au format [XHTML-1.1](#).

Pour les fichiers PDF Nous avons décidé de développer notre propre feuille de style XSLT<sup>1</sup> plutôt que d'utiliser les modèles déjà disponibles dans le package docbook-xsl. Ce choix se justifie surtout car nous voulions mieux comprendre les mécanismes syntaxiques de [XSL-FO](#). Enfin cela nous offre un résultat correspondant mieux à nos attentes.

La structure du projet a été travaillée et présente l'arborescence de répertoire suivante:

- article: qui contient l'ensemble des articles [docbook](#) que nous avons rédigés<sup>2</sup>.

1.Feuille de style xslt à la racine du projet testdoc.xsl.

2.Les articles rédigés sont disponibles dans le répertoire article du projet.

-docbook-xsl: package des modèles de feuille de style xslt définit pour la [docbook](#).

-fo: contient les fichiers fo générés

-html: contient les fichiers [xhtml-1.1](#) générés.

-pdf: contient les fichiers pdf générés.

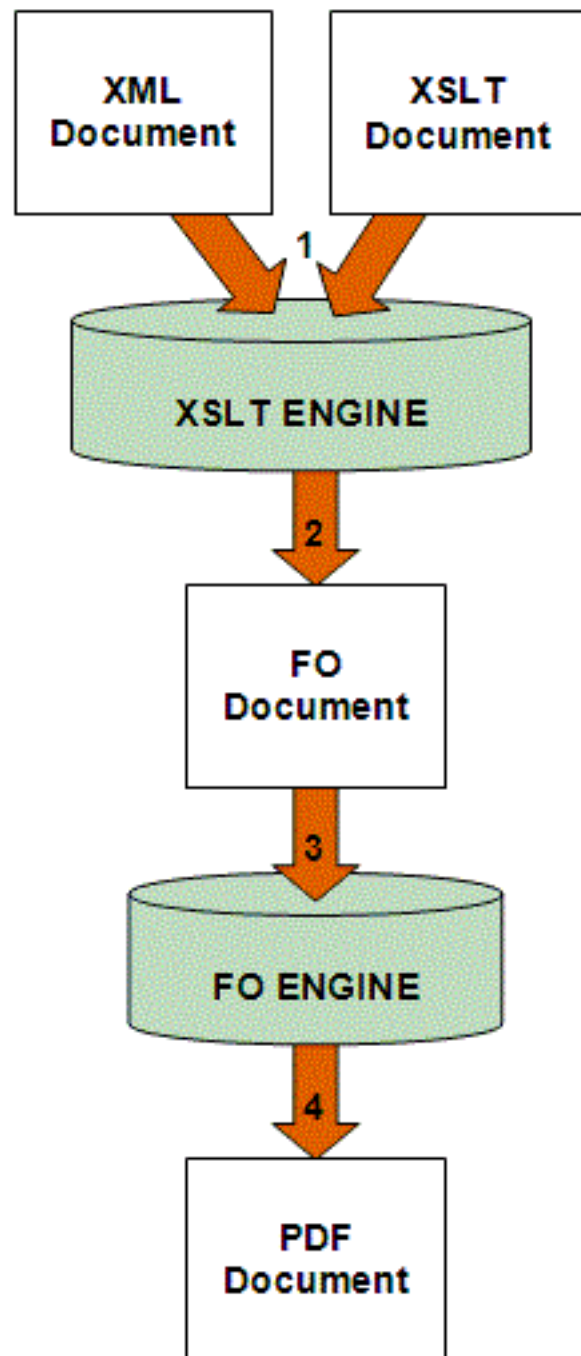
-image contient les images importées dans les documents

-lib: héberge les librairies nécessaires pour utiliser [saxon](#) ainsi que le processeur fop.

-pdf: utilisé pour stocker les fichiers finaux PDF.

## 6.0-Rappel du fonctionnement XSL-FO

Le [XSL-FO](#) permet de créer des documents finaux à partir de fichier xml et de feuille de style XSLT.



processus de génération PDF utilisé avec XSL-FO.

1 Un ou plusieurs documents xml et XSLT sont fournis à un processeur xslt

2 en utilisant les règles définies dans les document XSLT le processeur génère un fichier fo à partir du ou des fichiers xml.

3 le document fo est fourni au processeur fo.

4 le processeur fo applique les règles de formatage définies dans le document fo pour générer un fichier final au format PDF.

## 7.0-Le processus de construction des documents:

Le processus de construction des documents est défini dans le fichier [ant](#) build.xml situé à la racine du projet [eclipse](#).

Deux cibles<sup>3</sup> sont à retenir:

-clean qui vide les répertoires html, fo, pdf.

-newPDF qui génère les fichiers PDF dans le répertoire PDF, les fichiers fo utilisés sont stockés dans le répertoire fo, enfin cette cible génère également les fichiers résultat html dans le répertoire html en appelant la cible build-html.

## 8.0-Conclusion:

La [docbook](#) est un format de document riche et complet il s'intègre particulièrement bien avec le processeur fo pour générer des fichiers imprimables et autorise simplement des transformations vers divers formats comme le [xhtml-1.1](#) il est même possible de générer des slides pour une présentation. D'autre par la [docbook](#) est un format xml il offre donc un bien meilleur niveau d'interopérabilité que LaTeX.

<sup>3</sup>.ant doit être installé pour pouvoir les utiliser.

## Bibliographie

Cette bibliographie donne des références vers des livres.

Auteur	Titre
Dave-Pawson	XSL-FO
Stephen-Edlich	Ant Précis et concis
Michael-H. Kay	XSLT: Programmer's Reference, Second Edition
Karim-Djaafar	Développement JEE 5 avec Eclipse Europa
Norman-Walsh	DocBook 5: The Definitive Guide
Raphaël-Goetter	XHTML mémento