OOo2sDBK

OpenOffice Writer vers la Docbook simplifiée – version 0.3

Éric Bellot

<ebellot@netcourrier.com>

2002-07-05

OOo2sDbk converti les documents OpenOffice-Writer au format Docbook simplifié (vers. 4.1.2.5). Il fonctionne avec Python et avec Saxon.

Table des matières

OOo2sDbk - *OpenOffice-Writer to simplified Docbook* - est un script Python accompagné de feuilles de style XSLT. Il convertit les documents OpenOffice-Writer au format Docbook simplifié.OOo2sDbk fonctionne sous Windows 9x/Me/NT/2000 et GNU/Linux.OOo2sDbk est un logiciel libre sous licence GNU LGPL [http://www.gnu.org/copyleft/lesser.html]

Présentation

OOo2sDBK est un module Python permettant de convertir un document OpenOffice-Writer au format format *XML Docbook simplified* . Il utilise le processeur Saxon et une série de feuilles de styles XSLT.

OOo2sDBK est en cours de développement et doit donc être utilisé avec prudence.

OpenOffice est une suite bureautique libre puissante et conviviale. Elle est facile à prendre en main et disponible pour Windows, GNU/Linux et Solaris (bientôt MacOS X). La version 1.0 du logiciel est disponible en téléchargement chez OpenOffice.org [http://www.openoffice.org/]. OpenOffice est disponible dans de nombreux langages, dont le français.

Le Docbook est un format de document très utilisé par la communauté du libre pour la documentation technique. On peut citer The Linux Documentation Project [http://www.tldp.org/] et le projet de documentation KDE. Il permet d'encoder les documents en SGML et en XML. Le format Docbook est puissant, doté de grandes possibilités de conversion (HTML, TeX/LaTeX, XML, PDF, etc) ; mais sa complexité le rend difficile à prendre en main.

Le groupe Oasis qui développe le Docbook propose un version simplifiée de ce format. Le Docbook simplifié reste parfaitement compatible et peut suffire aux utilisations les plus courantes. Il n'est disponible qu'au format XML.

Principe de fonctionnement

OOo2sDBK converti un document SXW en Docbook en 3 étapes :

1. Décompression des fichiers XML du document OpenOffice-Writer [1] et s'il y a lieu des images incorporées.
2. Concaténation de ces différents fichiers en un gros document XML temporaire,
3. Conversion du document temporaire au format Docbook simplifié par Saxon et la feuille de style `sDocbook.xsl`.

Licence d'OOo2sDbk

Installation

Programmes requis

OOo2sDBK fonctionne sur Windows 9x/Me/NT/2000 et GNU/Linux.

Pour qu'OOo2sDbk fonctionne Python et un processeur XSLT doivent être fonctionnel.

Les processeurs XSLT suivants fonctionnent correctement avec Ooo2sDbk :

Saxon (recommandé)	Il supporte très bien tous les standards (mieux que Xsltproc) mais est assez lent car il fonctionne sous Java)
Xsltproc	Il est très rapide mais semble moins fiable que Saxon, en particulier sous Windows.

[1] Un document OpenOffice est une collection de documents XML compressés au format Zip.

Logiciels	Versions testées
Python 2.1 ou + http://www.python.org [http://www.python.org/]	• Python 2.1 et 2.2 sur Windows 95c et 2000, • Python 2.2 sur Linux-Mandrake 8.2.
Saxon 6.5.2 http://saxon.sourceforge.net [http://saxon.sourceforge.net/] Saxon requiert une plateforme JAVA2 pour fonctionner http://java.sun.com/j2se/1.3/download.html	Saxon 6.5.2 Plateformes JAVA testées : • J2RE 1.3.1_02 et 1.4 de Sun sur Windows 95c et 2000, • J2RE 1.4 et Kaffe sur Linux-Mandrake 8.2.
Xsltproc (fourni avec LibXSL) Requiert : • LibXML2 • LibXSL http://xmlsoft.org/downloads.html	Sous Linux et Windows • LibXML2 2.4.22 • LibXSL 1.0.18 Les versions antérieures ont posé des problèmes de conversion.

Étapes d'installation

1. Téléchargez la dernière version d'OOo2sDBK.
2. Décompressez `ooo2sbk-xxx.zip`. Vous obtenez un répertoire `ooo2sdbk`.
3. Copiez ce répertoire dans un emplacement accessible à Python, c'est à dire référencé dans la variable d'environnement `PYTHONPATH`. Par exemple :
   a. Sous Windows : `C:\program files\python21\`
      ou `C:\program files\python21\Lib\`
   b. Sous Linux : `/usr/lib/python21/`
4. Le script est opérationnel

Utilisation

OOo2sDBK exporte une méthode `convert` qui permet de réaliser la conversion d'un document OpenOffice Writer.

Étapes

1. Avec un éditeur de texte, rédigez le script qui appellera la méthode `convert` du module `ooo2sbk` (voir *3.2 Syntaxe* pour plus de détails). Par exemple, pour convertir le document `myDoc.sxw` en un document Docbook appelé `myDocbook.xml` avec le processeur Saxon, on rédigera le script suivant :

```
import ooo2sdbk
ooo2sdbk.convert("myDoc.sxw", docbook="myDocbook.xml", command="saxon")
```

1. Enregistrez le script avec l'extension `*.py` (par exemple : `conversion.py`) dans le même répertoire que le document OpenOffice
2. Ouvrez un shell sous Linux ou une session DOS sous Windows :
   a. Placez-vous dans le dossier ou se trouve votre script avec la commande `CD`
   b. Saisissez la commande : `python conversion.py`
3. Après quelques instants, la conversion est faite (la durée dépend de la taille du document OpenOffice et de la rapidité du processeur).

Syntaxe

```
convert(openoffice-filename, command=commandName, \
[docbook=docbook-filename], [imagesrew=0|1] [deltemp=0|1])
```

OpenOffice-filename (requit) Chemin d'accès (relatif ou absolu) au fichier OpenOffice-Writer à convertir

Command=commandName (requit) `commandName` est le nom de la commande utilisée pour la conversion.

La liste des commandes disponibles se trouve le fichier de configuration `config.xml`. Le fichier contient 2 commandes préconfigurées : `xsltproc` et `saxon` (Voir section *3.4 Fichier de configuration*).

docbook=Docbook-filename (facultatif) Chemin d'accès (relatif ou absolu) au fichier Docbook produit.

Si ce paramètre est omit, le fichier docbook aura le même nom que le fichier OpenOffice avec l'extension `*.xml` et sera enregistré dans le même répertoire. Par exemple : `myDoc.sxw` => `myDoc.xml`

Imagesrew=0|1 (facultatif, 1 par défaut) *Images Rewriting*. Ne concerne que les images incorporées au document OpenOffice.

Si la valeur est 1, les images incorporées au document OpenOffice écrasent les images ayant le même nom dans le répertoire de destination.

Si la valeur est 0, les images déjà présentes dans le répertoire de destination ne sont pas réécrites.

Deltemp=0|1 (facultatif, 1 par défaut) *Delete temporary file*. Si `deltemp` est *`0`*, le fichier temporaire `global.xml` est préservé. Par défaut, il est détruit à la fin de la conversion (`deltemp=1`). Le fichier `global.xml` est la concaténation de tous les fichiers XML du document

OpenOffice. Il sert de base à la conversion. Surtout utile pour le développement. Par défaut, `deltemp=1`.

Exemples de script

Exemple 1 (Windows)

Les fichiers sont ici indiqués avec de chemins absolus, on utilise le processeur Saxon.

```
import ooo2sbk
ooo2sbk.convert("C:\monDocumentOOo.sxw",docbook="C:\monDocbook.xml",\
command="saxon")
```

Exemple 2 (Windows ou Linux)

Les chemins d'accès aux fichiers sont relatifs, cela suppose, ici, que le script se trouve dans le même répertoire que les fichiers. Le fichier Docbook produit sera : `monDocumentOOo.xml`

```
import ooo2sbk
ooo2sbk.convert("monDocumentOOo.sxw", command="saxon")
```

Exemple 3 (Linux)

Chemins d'accès absolus et utilisation de **xsltproc** (avec **libxslt** et **libxml2**).

```
import ooo2sbk
ooo2sbk.convert("/home/Documents/doc.sxw",\
docbook="/home/Documents/monDocbook.xml", command="xsltproc")
```

Fichier de configuration

Le fichier de configuration `config.xml` se trouve à la racine du répertoire `ooo2sdbk`. Il permet notamment de configurer les modèles de commandes des processeurs XSLT disponibles

```
<config>
     <xslt-command
         name="xsltproc"
         command="xsltproc -o %o %s %i"/>
     <xslt-command
         name="saxon"
         command="java com.icl.saxon.StyleSheet -o %o %i %s"/>
     <xslt-stylesheet
         stylesheetPath="docbook.xsl"/>
     <images
         imageNameRoot="img"
         imagesRelativeDirectory="images"/>
</config>
```

xslt-command

Permet de définir un modèle de commande pour le processeur XSLT

name Nom du modèle de commande. Il sera appelé dans le script Python.

commandCommande de conversion du processeur XSL. Trois variables sont disponibles :

%o *output*, le nom du fichier XML produit

%i *input*, le nom du fichier XML entrant

%s *stylesheet*, le nom de la feuille de style XSL

Voici quelques exemple de commandes utilisables (à adaptés à votre système):

```
java com.icl.saxon.StyleSheet %i %s > %o
java -jar /usr/java/classes/saxon/saxon.jar -o %o %i %s (Linux)
java -cp C:\saxon\saxon.jar com.icl.saxon.StyleSheet %i %s > %o (Windows)
```

xslt-stylesheet

Permet de définir le chemin d'accès vers la feuille de style `docbook.xsl`

stylesheetPath Chemin *relatif* vers la feuille de style XSL à partir du fichier `ooo2sdbk.py`.

images

Permet de configurer l'extraction des images incorporées au document OpenOffice.

ImageNameRoot Radical du nom attribué aux images extraites du document OpenOffice.

Le radical "img" produira des image ayant pour nom : `img001.png`, `img002.jpg`, etc.

Le radical "pix" produira des images ayant pour nom : `pix001.png`, `pix003.jpg`, etc.

imagesRelativeDirectory Nom du répertoire qui accueille les images extraites du document OpenOffice. Ce répertoire est généré automatiquement dans le même répertoire que le document Docbook.

Problèmes de conversion

1. Avant la version 2.4.22, Xsltproc ne gère pas correctement la conversion du *Texte préformatté*. Les retours de ligne, les espaces multiples et les tabulations sont perdus.

2. Sous Windows, Xsltproc ne gère pas correctement les chemins dans les balises `import` et `include` des feuilles XSLT (échec total de la conversion).

Éléments supportés par le convertisseur

Attention. Cette section n'est plus à jour.

Résumé des limites essentielles de la conversion

- Les *images ancrées à la page* sont perdues.
- Dans les tableaux, les *fusions verticales de cellules* ne sont pas supportées.
- Le document doit avoir une *hiérarchie cohérente des titres* :
  - le premier titre doit être de niveau 1.
  - Les niveaux doivent être correctement imbriqués (pas de niveau 1 suivi d'un niveau 3, sans niveau 2 intermédiaire).
- Les sous-listes produisent sont précédées d'un item de liste vide.

Titres et sections

- Les sections et titres des sections du Docbook sont construits à partir des niveaux de chapitres du document SXW (menu *Outils* > *Numérotation des chapitres*).
- S'il comporte des chapitres, le 1er chapitre *doit* être un chapitre de niveau 1.
- Le document peut ne pas comporter de chapitre, dans ce cas, aucune section n'est générée.

Métadonnées

- Les styles *Titre principal* et *Sous-titre* sont convertis en `<title>` et `<subtitle>` dans `<infoarticle>`.

Les métadonnées d'OOo sont intégrées à la section `<infoarticle>`.

- *Auteur* => `<AuthorGroup>`Docbook subtile des identités. Il distingue le titre, le prénom, le nom et le surnom d'une personne (respectivement `<honorific>`, `<firstname>`, `<surname>` et `<othername>`). Il distingue les différents auteurs d'un document (`<author>`), il permet d'indiquer d'autres informations (email, adresse, etc.).

  Autant d'éléments qu'OpenOffice, vraiment indigent en la matière, ignore superbement. Il ne propose qu'un champ *Auteur* sans plus.

  Le convertisseur est capable de rendre correctement le nom et le prénom d'un auteur unique s'il est rédigé sous la forme `[Nom, prénom]`; la virgule servant de séparateur.

- *Sujet* => `<subject>`
- *Mots-clefs* => `<keyword>`Les mots-clefs doivent être séparés par des virgules.

- *Description* => `<abstract>`
- *Date de dernière modification* => `<pubdate>`

Paragraphes

- *Corps de texte* et *Standard* => `<para>`
- *Texte préformatté* => `<programlisting>`Les retours de ligne, espaces et tabulations sont préservés (les tabulations sont converties en 4 espaces)
- *Citation* => `<quotation>`
- Les autres paragraphes sont convertis en `<para>`.

Caractères

- *Accentuation* => `<emphasis>`
- *Accentuation forte* => `<emphasis role="strong">`

  Par défaut, la Docbook ne définie pas d'accentuation forte (en gras le plus souvent). Les *Docbook XSL stylesheet* doivent être modifiées pour supporter cet ajout, sinon il est tout simplement rendu comme une accentuation standard. Voir TLDP pour plus d'information.
- *Exemple*, *Texte non proportionnel* et *Texte source* => `<literal>`
- *Saisie de l'utilisateur* => `<userentry>`
- *Citation* => `<citetitle>` (titre d'une oeuvre citée)
- *Variable* => `<varname>`

  Mais `<varname>` n'est pas intégré à la Docbook simplifiée, seulement dans la version intégrale.
- Les autres styles de caractères ne sont pas rendus (sauf les appels de notes qui sont produits automatiquement lors de la création d'une note de bas de page).

Notes de bas de page et notes de fin

Elles sont supportées avec les limites suivantes :

- La Docbook ne distingue pas les notes de bas de page et les notes de fin de document. Les deux types de notes sont donc convertis en `<footnote>` sans distinction.
- Les appels de notes par caractères spéciaux sont supportés par OOo2sDBK mais ne sont pas rendues lors des conversions avec les *Docbook XSL stylesheets.* Ces appels de notes sont inclus dans la numérotation automatique.

Tableaux

Les tableaux sont supportés avec les limites suivantes :

- *Les fusions verticales de cellules sont perdues y compris le texte qu'elles contiennent*. De plus le tableau est totalement déstructuré.
- Les fusions horizontales sont correctement rendues.
- Les tailles de cellules sont supportées. Toutefois, le Docbook produit ne sera correctement converti qu'en utilisant les extensions fournies par Norman Walsh avec ses *Docbook XSL stylesheets*.

Listes

OOo2sDBK supporte les listes à puces et numérotées:

- Les listes à plusieurs niveaux sont possibles.
- Les parties d'items de listes séparés par des retours de ligne seront convertis en autant de paragraphes.

Les limitations sont les suivantes :

- Les styles de puces ou de numérotation personnalisés ne sont pas conservés. Ils sont remplacés par les styles standards dans le Docbook.
- Les puces graphiques ne sont pas supportées.
- Une sous-liste numérotée produit un item vide (bogue du XML d'OpenOffice semble-t-il).

Hyperliens et références croisés

La plupart des fonctionnalités hypertextes sont supportées.

Les URLs

- Les hyperliens d'OOo sont convertis en `<ulink>`.
- L'attribut *Frame* des hyperliens d'OOo est converti en attribut *Type* dans le Docbook. Il permet de passer les valeurs de *Target* (`_parent`, `_blank`, etc.).

Les images liées

- Seules les images ancrées *comme caractères* supportent les hyperliens.
- Les liens sur images flottantes sont perdus (ancrage *au paragraphe* ou *au caractère*).

Les repères de texte et les renvois internes

- Les repères de textes et les marques de renvois sont convertis en `<anchor>`.
- Les renvois sont convertis en `<link>`.

Les renvois bibliographiques

- Les renvois bibliographiques automatiques sont convertis en `<xref>`.

Images

- *Les images ancrées à la page sont perdues.*

- Les images ancrées *comme caractère* sont converties en `<InlineMediaObject>`.
- Les images ancrées *au paragraphe* ou *au caractère* sont converties `<MediaObject>`.

- L'attribut *texte facultatif* d'une image est converti en `<TextObject>` (texte de remplacement en cas d'affichage textuel).
- Les dimensions d'affichage des images sont supportées.

Images liées et images incorporées

OOo2sDbk supporte les images liées et incorporées.

- Les images incorporées sont extraites du fichier SXW et rassemblées dans un répertoire `images` créé dans le même répertoire que le document Docbook :
  - Les images sont nommées `img001.xxx`, `img002.xxx`, etc. La numérotation suit l'ordre d'apparition des images dans le document.
  - Les images ne subissent aucune conversion.
  - Il est possible de modifier le nom du répertoire `images` et le radical "img" des images dans le fichier de configuration `config.xml`.
- Les images liées ne sont pas touchées :
  - les liens sont passés au document Docbook sans modification.

Positionnement vertical des images flottantes

OpenOffice permet un placement subtil des images flottantes. Il est impossible à rendre dans le Docbook.:

- L'alignement horizontal des images flottantes est perdu. A ma connaissance, le Docbook ne supporte pas cette fonction.
- Les images flottantes alignées *en haut*, *au milieu* ou *d'en haut* seront placées au-dessus du paragraphe d'ancrage dans le Docbook.
- Les images flottantes positionnées *en bas* seront placées à la suite du paragraphe d'ancrage dans le Docbook.

Images légendées et cadres

La commande *Légende* appliquée à une image créée un cadre qui incorpore l'image et le texte de la légende.

- La légende est converti en `<caption>` dans `<MediaObject>`.
- OOo2sDBK n'est pas capable de géré les cadres ancrés *comme caractère*. Au cas où cela se produirait les

cadres seraient placés au dessus du paragraphe de référence.

Cette page a été réalisée sur OpenOffice 1.0 et convertie au format simple Docbook avec OOo2sDBK 0.3.

Le document Docbook produit a été converti en HTML avec les feuilles XSLT pour Docbook (v. 1.50) [http://docbook.sourceforge.net/projects/xsl/index.html] de Norman Walsh.

Éric Bellot – http://www.chez.com/ebellot/ooo2sdbk