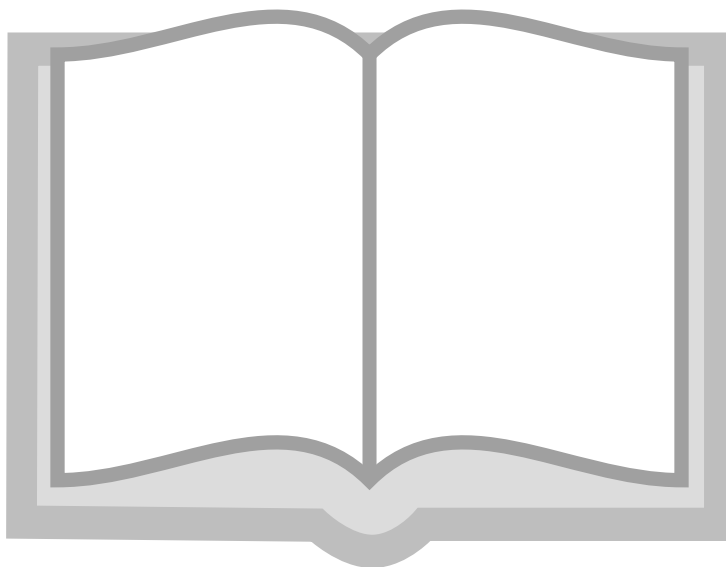

Chaîne de publication 1.0

Guide de l'administrateur et de l'utilisateur

Comment configurer et installer une chaîne de publication



Xavier Berger

Cette œuvre est mise à disposition selon les termes de la [Licence Creative Commons Attribution - Partage dans les Mêmes Conditions 3.0 non transposé](http://creativecommons.org/licenses/by-sa/3.0/)¹. Vous pouvez modifier et redistribuer son contenu tant que vous attribuez son origine à son auteur et partagez votre travail sous la même licence Creative Commons.

AVERTISSEMENT:

Les informations fournies dans ce document viennent sans garantie d'aucune sorte, quelle soit explicite ou implicite. Ce document est distribué en l'état.

Tous les efforts ont été entrepris pour fournir une information la plus juste possible. Les informations peuvent cependant être incomplètes, contenir des erreurs ou devenir obsolètes. L'usage des informations décrites ici est sous votre entière responsabilité et leur usage sur vos systèmes est effectué à vos propres risques.

Résumé

Cet article décrit comment installer et utiliser une chaîne de publication basée sur les logiciels publiés et s'auto-financent. Cet article lui-même a été écrit en utilisant cette chaîne de publication et est utilisé comme exemple de rédaction et de publication.

A propos de l'auteur

¹ <http://creativecommons.org/licenses/by-sa/3.0/>

Xavier Berger travail comme Solution Architect dans une entreprise spécialisée dans les télécoms. Il est expérimenté dans le déploiement de solutions Linux et réseaux. Xavier apprécie aussi la randonnée, le géocaching, le ski et passer du temps avec sa famille. Son site web est: <http://xberger.free.fr>

1. Introduction	2
2. Installation	5
2.1. Installer publican et poedit	5
2.2. Installation de serna-free	5
3. Rédaction	6
3.1. Création du squelette de l'article	6
3.2. Préparation de la structure du document	9
3.3. Ecriture de l'article:	10
4. Publication	14
4.1. Effectuer un rendu du document	14
4.2. Gérer un site web	15
5. Traduction	17
A. Historique des modifications	19

1. Introduction

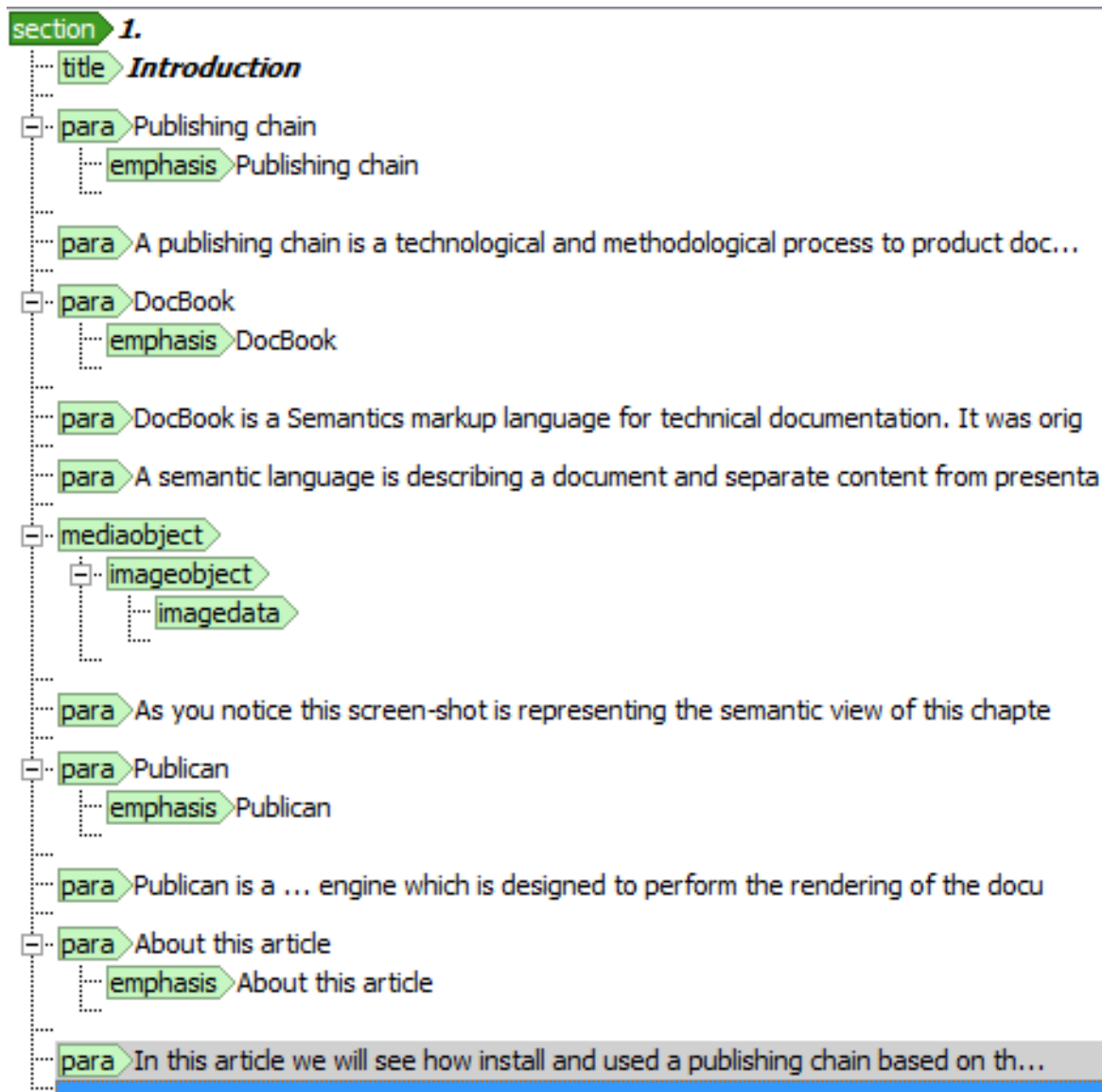
Une chaîne de publication

Une chaîne de publication est une technologie et un processus méthodologique de production de documentation. L'approche d'une chaîne de production est de créer un modèle de document contenant les données et une description de son format. Le document final sera généré dans le format désiré (pdf, html, epub...)

DocBook

DocBook est un langage balisé sémantique pour la documentation technique. Il a été originellement conçu pour la redaction de documents techniques en relation avec le matériel informatique ou logiciel mais il peut aussi être utilisé pour rédiger tout autre sorte de documentation.

Un langage sémantique décrit un document et sépare le contenu de la présentation. Les fichiers **Doc-Book** stoquent les informations au format XML. La copie d'écran ci-dessous montre comment un langage sémantique balisé peut être représenté.

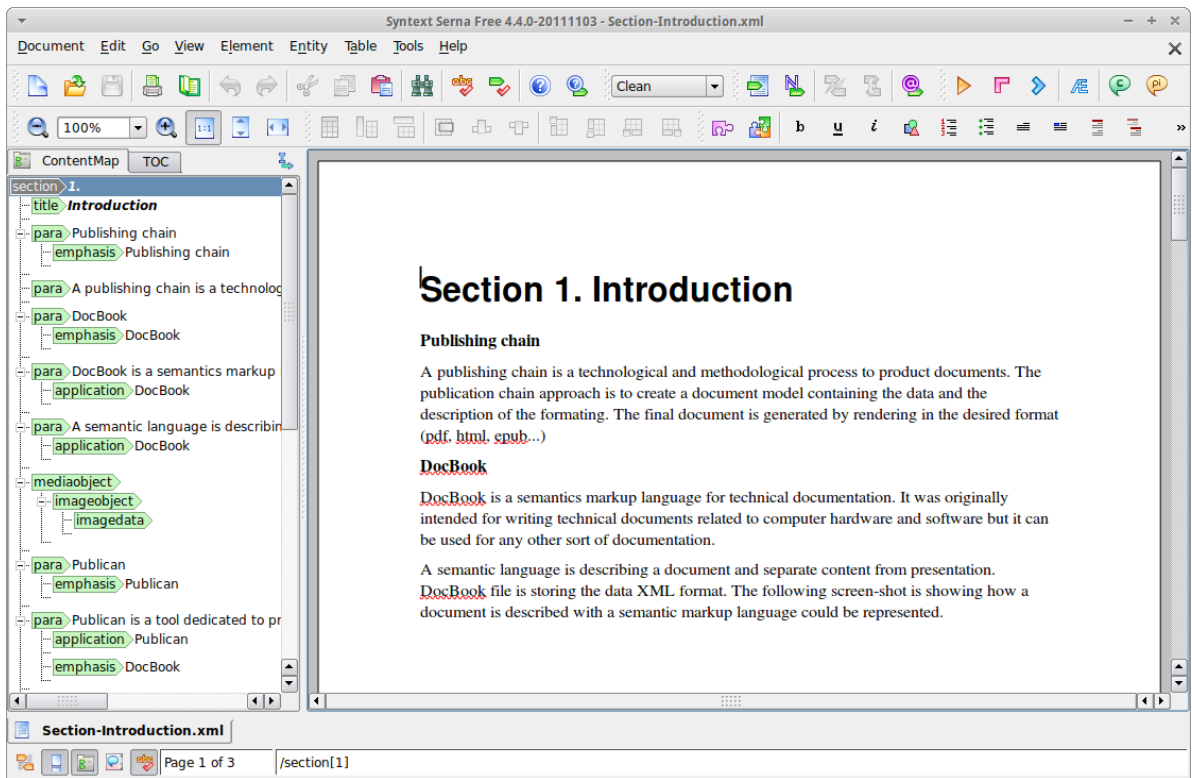


Publican

Publican est un outil dédié au traitement de fichier XML **DocBook**. C'est un moteur de rendu qui peut générer des documents dans différents formats en appliquant une transformation aux données XML. Le résultat ainsi obtenu sera un document facile à lire dans un format pdf, html ou ePub.

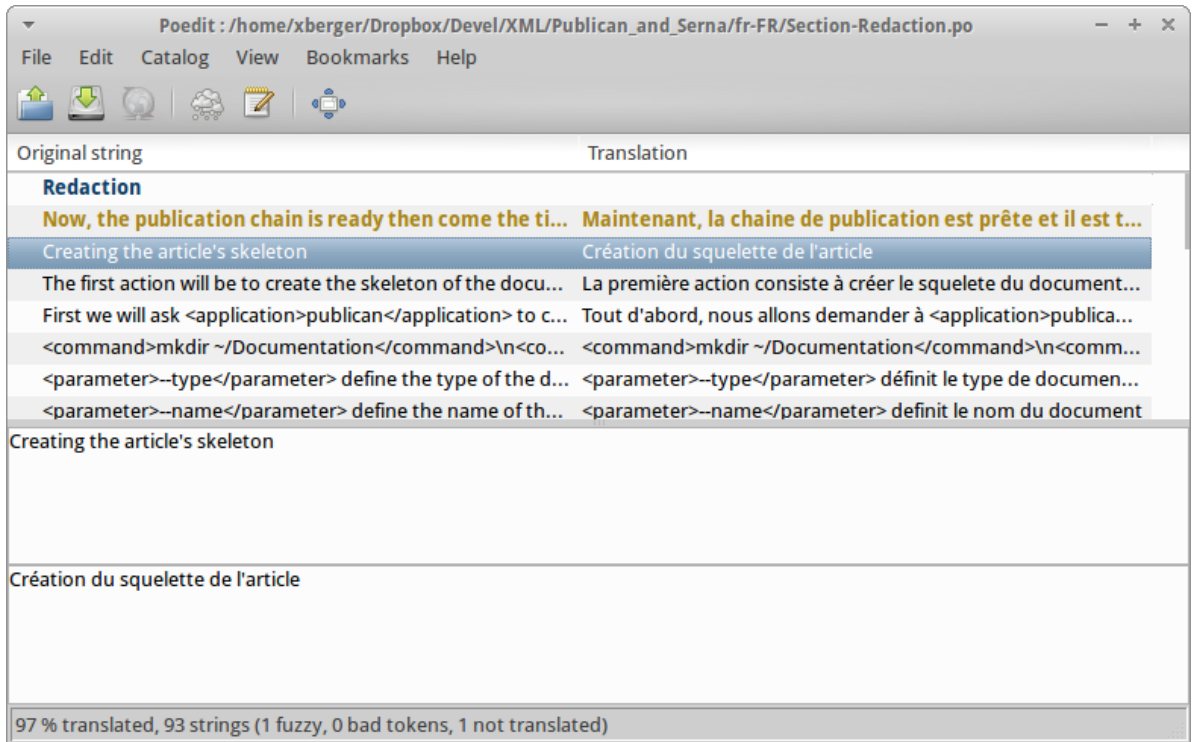
Serna

serna-free est un logiciel libre conçu pour la manipulation de fichier XML de différents formats. **Doc-Book** est l'un de ces formats. **serna-free** présente une interface WYSIWYM: What You See Is What You Mean - Ce que vous voyez est ce que vous signifier. Elle permet une édition facile de documents XML.



poedit

poedit est un logiciel dédié à la traduction. Il offre au traducteur une interface graphique et affiche l'état de traduction du document. Nous allons voir dans la dernière partie de ce document comment l'utiliser pour produire une documentation multilingue.



A propos de cet article

Cet article a été écrit en utilisant **serna-free**, **publican** et **poedit**. Nous allons voir comment installer et utiliser une chaine de publication basée sur ces trois logiciels.

2. Installation

Dans ce chapitre nous allons installer les logiciels composant la chaîne de publication.

2.1. Installer publican et poedit

publican est disponible dans le dépôt officiel d'**Ubuntu**. L'installation de **publican** s'effectue avec la commande suivante:

```
sudo apt-get install publican libservlet2.4-java
```

poedit est lui aussi disponible dans le dépôt officiel d'Ubuntu et s'installe avec la commande suivante:

```
sudo apt-get install poedit
```

2.2. Installation de serna-free

serna-free n'est pas disponible dans le dépôt officiel d'Ubuntu ni sous forme en paquet installable deb. Nous allons utiliser le paquet rpm et le logiciel **alien** pour effectuer une installation propre.

```
alien
```

Télécharger le rpm dans le dépôt officiel:

```
wget http://downloads.sourceforge.net/project/sernafree/mirror/serna-free-4.4-4.4.0-20111103.i686.rpm -O serna-free.rpm
```

Convertissez le rpm en deb avec **alien** (Cela peut prendre un peu de temps)

```
sudo alien -dck serna-free.rpm
```

Installez le paquet créé en utilisant la commande:

```
sudo dpkg -i serna-free*.deb
```

Exécutez le script de post installation afin de finaliser l'installation:

```
sudo sh -c 'export SERNA_EXE=serna.bin; export SERNA_TAG=serna-free-4.4; export INSTALL_PREFIX=/opt; /opt/serna-free-4.4/bin/serna-postin.sh'
```

Créez un lien symbolique dans **/usr/local/bin/** afin de l'avoir dans le PATH

```
cd /usr/local/bin
sudo ln -s /opt/bin/serna
```

Enfin, ajoutez un raccourci dans le menu du bureau en créant le fichier **~/ .local/share/applications/serna.desktop** avec le contenu suivant:

```
[Desktop Entry]
Name=Serna
Comment=XML WISWIYM Editor
Exec=serna
Icon=/opt/serna-free-4.4/icons/SernaIcon32.png
Terminal=false
Type=Application
```

```
Categories=Application;Office;  
StartupNotify=true
```

L'application **serna-free** est maintenant disponible dans **Ubuntu** → **Bureautique** → **Serna**

3. Rédaction

Maintenant, la chaîne de publication est prête et il est temps d'écrire un article ou un livre (book).

3.1. Création du squelette de l'article

La première action consiste à créer le squelette du document. Nous allons prendre la rédaction de ce document comme exemple.

Tout d'abord, nous allons demander à **publican** de créer le squelette de l'article avec les commandes suivantes:

```
mkdir ~/Documentation  
cd ~/Documentation  
publican create --type=article --name "Publican and Serna" --product "Publication Chain"
```

--type définit le type de document. Le type peut être *article* ou *book*.

--name définit le nom du document.

--product définit le nom du produit pour lequel le document est écrit.

La commande crée un répertoire `~/Documentation/Publican_and_Serna/` directement construit à partir de la valeur du paramètre `--name`. Ce répertoire contient le fichier de configuration de **publican**: `~/Documentation/Publican_and_Serna/publican.cfg` et cinq autres fichiers dans le sous-répertoire `~/Documentation/Publican_and_Serna/en-US/`. *en-US* est la langue par défaut de **publican**.

Jetons un œil à ces fichiers:

`~/Documentation/Publican_and_Serna/en-US/Publican_and_Serna.xml`

```
<?xml version='1.0' encoding='utf-8' ?>  
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN" "http://www.oasis-open.org/  
docbook/xml/4.5/docbookx.dtd" [  
<!ENTITY % BOOK_ENTITIES SYSTEM "Template.ent">  
%BOOK_ENTITIES;  
>  
<article>  
  <xi:include href="Article_Info.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />  
  <para>  
    This is a test paragraph  
  </para>  
  <xi:include href="Revision_History.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />  
  <index />  
</article>
```

Le nom de ce fichier vient directement du paramètre `--name` de la ligne de commande. C'est le document racine. Il inclut les autres documents générés automatiquement comme le fichier contenant les informations de l'article au début du document et l'historique des révisions à la fin, juste avant l'index.

Dans ce squelette, il est proposé d'écrire directement l'article dans ce fichier. Nous verrons dans le chapitre suivant comment concevoir un document plus facile à éditer en incluant des sections écrites dans des fichiers dédiés.

~/Documentation/Publican_and_Serna/en-US/Article_Info.xml

```

<?xml version='1.0' encoding='utf-8' ?>
<!DOCTYPE articleinfo PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN" "http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd" [
<!ENTITY % BOOK_ENTITIES SYSTEM "Publican_and_Serna.ent">
%BOOK_ENTITIES;
]>
<articleinfo id="arti-Publication_Chain-Publican_and_Serna">
<title>Publican and Serna</title>
<subtitle>short description</subtitle>
<productname>Publication Chain</productname>
<productnumber>0.1</productnumber>
<edition>0</edition>
<pubsnumber>0</pubsnumber>
<abstract>
<para>
A short overview and summary of the book's subject and purpose, traditionally no more than one paragraph long. Note: the abstract will appear in the front matter of your book and will also be placed in the description field of the book's RPM spec file.
</para>
</abstract>
<corpauthor>
<inlinemediaobject>
<imageobject>
<imagedata fileref="Common_Content/images/title_logo.svg" format="SVG" />
</imageobject>
</inlinemediaobject>
</corpauthor>
<xi:include href="Common_Content/Legal_Notice.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="Author_Group.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
</articleinfo>

```

Ce fichier contient les informations de l'article: le titre du document (title), le produit associé (product), la version du produit et du document ainsi qu'un résumé de l'article. Il inclut aussi d'autres documents:

- La notice légale fait partie des modèles de **publican**.
- La liste des auteurs générée par la commande précédente.

~/Documentation/Publican_and_Serna/en-US/Author_Group.xml

```

<?xml version='1.0' encoding='utf-8' ?>
<!DOCTYPE authorgroup PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN" "http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd" [
<!ENTITY % BOOK_ENTITIES SYSTEM "Publican_and_Serna.ent">
%BOOK_ENTITIES;
]>
<authorgroup>
<author>
<firstname>Dude</firstname>
<surname>McPants</surname>
<affiliation>
<orgname>Somewhere</orgname>
<orgdiv>Someone</orgdiv>
</affiliation>
<email>Dude.McPants@example.com</email>
</author>
</authorgroup>

```

Ce fichier rassemble la liste des auteurs du document. Le contenu de ce fichier est inclus dans les informations de l'article.

~/Documentation/Publican_and_Serna/en-US/Revision_History.xml

```
<?xml version='1.0' encoding='utf-8' ?>
<!DOCTYPE appendix PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN" "http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd" [
<!ENTITY % BOOK_ENTITIES SYSTEM "Publican_and_Serna.ent">
%BOOK_ENTITIES;
]>
<appendix id="appe-Publican_and_Serna-Revision_History">
<title>Revision History</title>
<simpara>
<revhistory>
<revision>
<revnumber>0-0</revnumber>
<date>Fri Sep 21 2012</date>
<author>
<firstname>Dude</firstname>
<surname>McPants</surname>
<email>Dude.McPants@example.com</email>
</author>
<revdescription>
<simplelist>
<member>Initial creation of book by publican</member>
</simplelist>
</revdescription>
</revision>
</revhistory>
</simpara>
</appendix>
```

Ce fichier rassemble la liste des révisions de l'article. Il sera visible dans les annexes à la fin du document (Il est en effet inclus à la fin du fichier racine).

~/Documentation/Publican_and_Serna/en-US/Publican_and_Serna.ent

```
<!ENTITY PRODUCT "Publication Chain">
<!ENTITY BOOKID "Publican and Serna">
<!ENTITY YEAR "2012">
<!ENTITY HOLDER "| You need to change the HOLDER entity in the en-US/Publican_and_Serna.ent file |">
```

Ce fichier contient les entités. Ce fichier est inséré dans tous les fichiers générés. Les variables définies dans ce fichier peuvent être utilisées comme paramètre dans les documents XML.

~/Documentation/Publican_and_Serna/publican.cfg

```
xml_lang: "en-US"
type: Article
brand: common
```

Ce fichier est utilisé par **publican** pendant la génération du document. Il définit le type du document (article) et la marque (brand) qui va définir le style du rendu final. Avec les paquets fournis par **Ubuntu**, seul la marque '**common**' est disponible.

Effectuons maintenant le rendu de l'article avec la commande:

```
cd ~/Documentation/Publican_and_Serna
publican build --langs en-US --formats html-single
```

et voyons à quoi notre document ressemble dans **firefox**


```
firefox tmp/en-US/html-single/index.html
```

Nous allons voir en détail comment utiliser cette commande et générer des documents agréables à lire.

3.2. Préparation de la structure du document

Nous devons maintenant modifier les fichiers générés automatiquement et remplacer les valeurs par défaut par de vrais informations. Ceci devrait être effectué directement en éditant le fichier dans un éditeur de texte classique.

Modifiez le fichier **publican.cfg** en y ajoutant les lignes suivantes:

```
docname: Publican_and_Serna
mainfile: Publican_and_Serna
```

Ceci permettra de changer le titre du document sans changer le nom du fichier XML. Si ces valeurs ne sont pas définies, **publican** déterminera le nom du fichier racine en fonction du titre inscrit dans les informations de l'article (**Article_Info.xml** ou **Book_Info.xml**).

Modifiez les fichiers **Article_Info.xml**, **Author_Group.xml**, **Revision_History.xml** et **Publican_and_Serna.ent**. Modifiez leur contenu avec les informations relatives au projet et aux auteurs.

Maintenant, pour simplifier la rédaction de notre article, nous allons créer des sections que nous incluons dans le document racine : **Publican_and_Serna.xml**.

Chaque section est stockée dans un fichier et contient le texte et la description du format de chaque chapitre.

Pour la rédaction de cet article, nous créerons les sections suivantes:

- **Introduction** stockée dans le fichier **Section-Introduction.xml**
- **Installation** stockée dans le fichier **Section-Installation.xml**
- **Redaction** stockée dans le fichier **Section-Redaction.xml**
- **Publication** stockée dans le fichier **Section-Publication.xml**

Avec le contenu suivant:

```
<?xml version='1.0' encoding='utf-8' ?>
<!DOCTYPE section PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN" "http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd" [
<!ENTITY % BOOK_ENTITIES SYSTEM "Publican_and_Serna.ent">
%BOOK_ENTITIES;
]>
<section>
<title/>
<para/>
</section>
```

Ce code XML décrit une section. Cette section contient un titre vide et un paragraphe vide.

Nous mettons alors le document racine **Publican_and_Serna.xml** à jour pour y inclure les sections dans leur ordre d'apparition.

La racine du document contient maintenant ce contenu:

```
<?xml version='1.0' encoding='utf-8' ?>
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN" "http://www.oasis-open.org/
docbook/xml/4.5/docbookx.dtd" [
<!ENTITY % BOOK_ENTITIES SYSTEM "Publican_and_Serna.ent">
%BOOK_ENTITIES;
]>
<article>
<xi:include href="Article_Info.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="Section-Introduction.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="Section-Installation.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="Section-Redaction.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="Section-Publication.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="Section-Revision_History.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
<!--index /-->
</article>
```

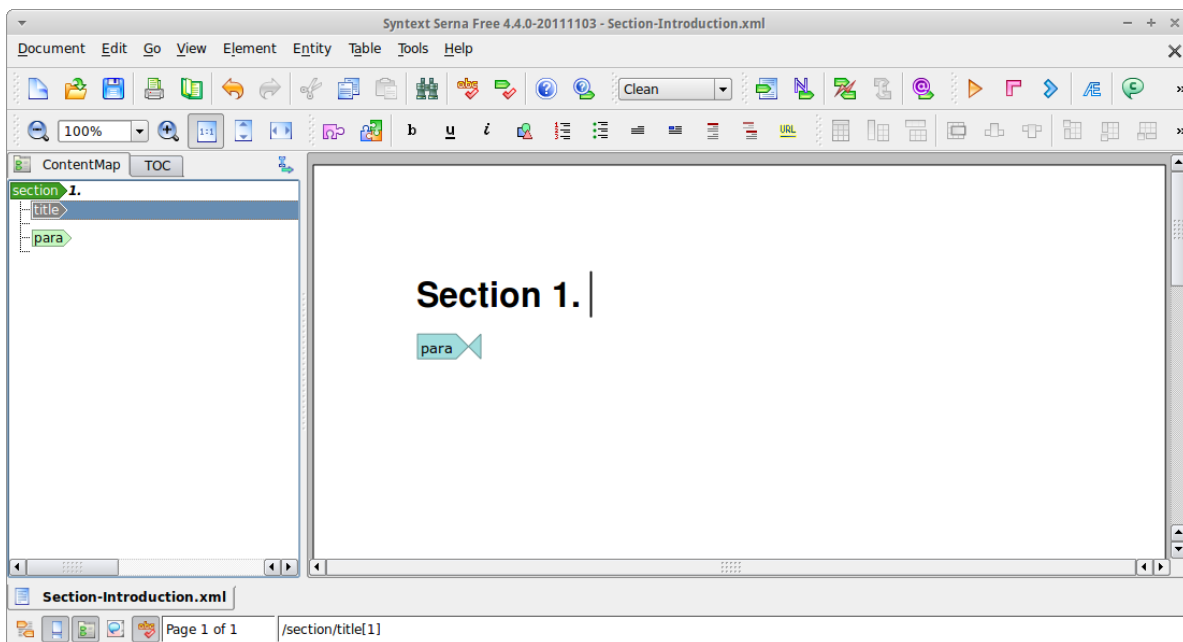
Nous avons commenté l'**index** qui n'est pas nécessaire pour ce projet.

3.3. Ecriture de l'article:

3.3.1. Utilisation de serna-free

La structure de notre document est prête. Nous pouvons commencer à écrire notre article. Pour rendre cette rédaction plus facile nous allons utiliser **serna-free**. Cet outil permet l'édition de document en **WYSIWYM**. Il effectue aussi la validation à la volée de la conformité du document avec les règles **DocBook**. Si vous essayez d'enfreindre une de ces règles, le logiciel vous retournera une erreur et annulera l'action fautive.

Démarrez **serna-free** et ouvrez le document **Section-Introduction.xml** en utilisant le menu **Document** → **Open...**



serna-free est divisé en deux zones: sur la gauche *Content Map / TOC* et sur la droite la zone d'édition.

La zone *Content Map / TOC* affiche une vue de la structure du document XML. Cette zone est interactive et permet la réorganisation du document par simple glisser/déposer.

La zone de droite affiche une vue formatée du document. Cette représentation ne reflète pas la vue exacte du document final mais un aperçu de ce à quoi le document ressemblera.

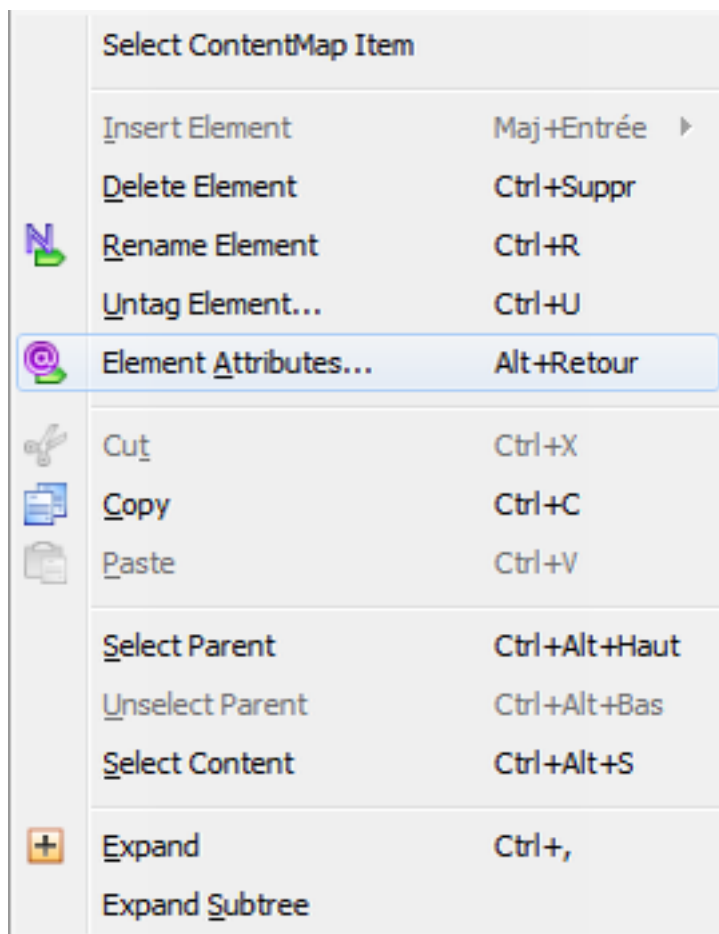
L'usage des deux zones permet l'édition du document. La première chose à faire est d'ajouter le titre de la section ainsi que le texte du paragraphe.

Dans un article **DocBook** tout comme dans un fichier XML, le formatage n'est pas effectué durant la rédaction de l'article. Nous effectuerons une description du format en utilisant des balises XML. Les balises XML sont le sujet du prochain paragraphe.

3.3.2. Les balises ou éléments XML

Ce chapitre rassemble quelques éléments de base de **DocBook** et explique comment les mettre en oeuvre dans **serna-free**. Nous verrons aussi comment ces éléments sont encodés en **DocBook XML**.

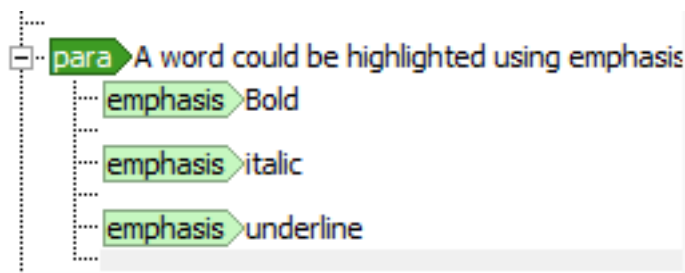
Dans **serna-free**, le menu contextuel peut être activé dans les deux zones d'édition. Il offre un accès à toutes les possibilités d'édition du document. Quand le texte est sélectionné, il est possible de le 'taguer' et de l'insérer entre deux balises XML.



Mettre en évidence des mots

Un mot peut être mis en évidence en le mettant en : **Gras (bold)**, *italique (italic)* ou en le soulignant (underline)

Pour faire ceci, sélectionnez le texte à mettre en évidence dans la zone d'édition puis cliquez sur les boutons de la barre d'outils. Le document sera modifié comme représenté dans la copie d'écran ci-dessous:



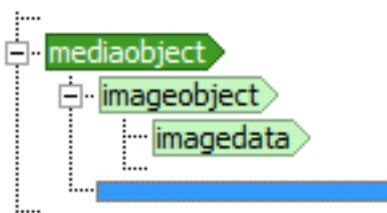
Cette image représente le code XML ainsi généré:

```
<para>A word could be highlighted using emphasis: <emphasis role="bold">Bold</emphasis>, <emphasis role="italic">italic</emphasis>, <emphasis role="underline">underline</emphasis></para>
```

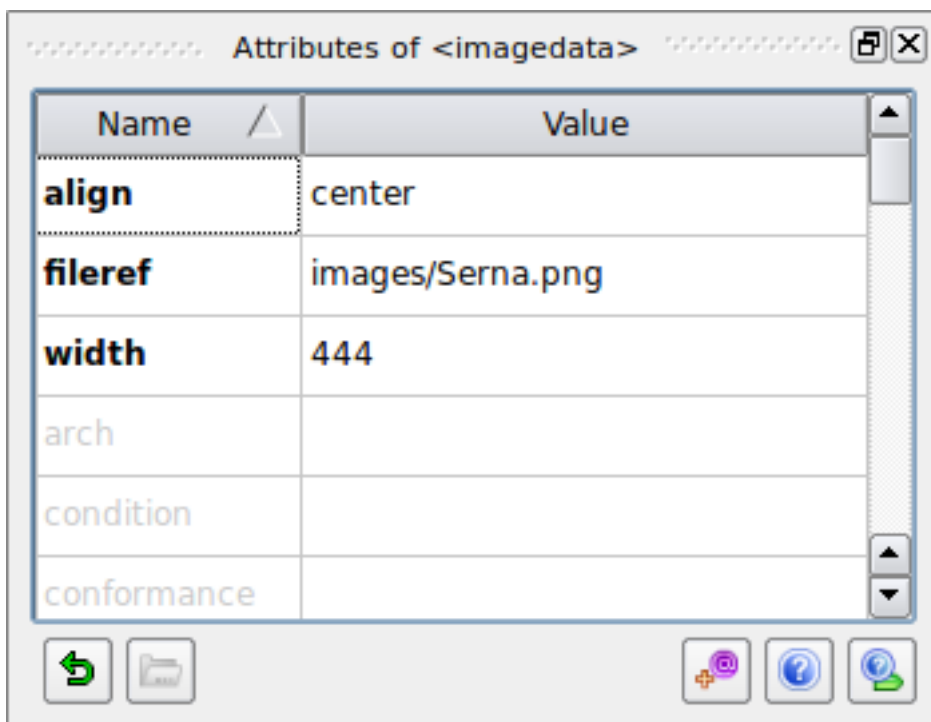
Un mot peut aussi représenter un **fichier (filename)**, une **application**, une **commande (command)**, un **code**, une phrase, un **nom de classe (classname)**... Il y a de nombreuses possibilités offertes par les références **DocBook**. Pour mettre en évidence un mot ou une phrase, vous devez le sélectionner puis utiliser le menu contextuel **Wrap Info Element**

Images

Les images sont des objets spécifiques et doivent être insérées dans des balises **<mediaobject>** ou **<inlinemediaobject>** composées par un **<imageobject>** lui-même contenant une **<imagedata>**.



L'élément **<imagedata>** contient des attributs accessibles depuis le menu contextuel **Element Attributes...**



L'image à afficher est définie par l'attribut `filerref`. Vous pouvez utiliser la petite icône en forme de dossier jaune pour sélectionner l'image à insérer.

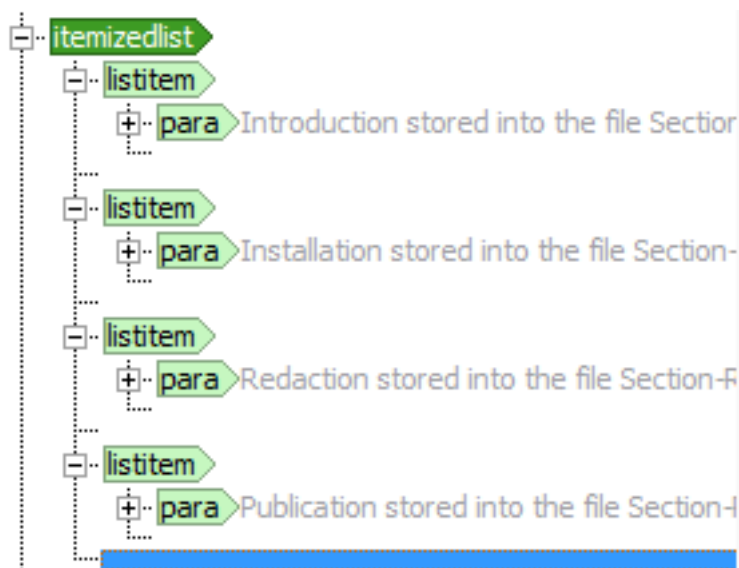
Le code XML ainsi généré ressemblera à ceci:

```
<mediaobject>
  <imageobject>
    <imagedata align="center" filerref="images/MediaObject.png"/>
  </imageobject>
</mediaobject>
```

Listes

Deux types de listes existent: `<itemizedlist>` et `<orderedlist>`. Pour la première balise, chaque élément de la liste sera précédé par un point. Dans la liste ordonnée, chaque élément est précédé par un nombre.

Les éléments `<itemizedlist>` et `<orderedlist>` représentent les points d'entrées des listes. Ces éléments sont composés de `<listitem>` qui contiennent l'information à afficher, un `<para>` dans l'exemple ci dessous:



Le code XML ainsi généré ressemblera à ceci:

```
<itemizedlist>
  <listitem>
    <para>Introduction stored into the file <filename>Section-Introduction.xml</filename></para>
  </listitem>
  <listitem>
    <para>Installation stored into the file <filename>Section-Installation.xml</filename></para>
  </listitem>
  <listitem>
    <para>Redaction stored into the file <filename>Section-Redaction.xml</filename></para>
  </listitem>
  <listitem>
    <para>Publication stored into the file <filename>Section-Publication.xml</filename></para>
  </listitem>
</itemizedlist>
```

Dans ce document nous n'utilisons que des `<itemizedlist>`. La création de sous-liste peut se faire en insérant un `<itemizedlist>` ou `<orderedlist>` dans un `<listitem>`.

Listing de programme

L'élément `<programlisting>` permet d'insérer des morceaux de code dans le document. **publican** offre la possibilité d'effectuer une coloration syntaxique du texte pour rendre le code plus lisible. Pour permettre une coloration syntaxique correcte, il faut définir le langage utilisé. Ce langage sera défini dans l'attribut **language**. La liste des langages supportés se trouve dans la page suivante:

- <http://search.cpan.org/~szabgab/Syntax-Highlight-Engine-Kate-0.06/lib/Syntax/Highlight/Engine/Kate.pm#PLUGINS>

La première colonne représente les mots clés à ajouter dans l'attribut.

Pour aller plus loin

Des vidéos commentées en anglais expliquant l'utilisation de **serna** sont disponible ici:

- <http://vimeo.com/groups/111908>

Des recommandations pour écrire un article **DocBook** sont disponible en anglais ici:

- https://hudson.jboss.org/hudson/job/PressGang_Documentation_Guide/lastSuccessfulBuild/artifact/target/docbook/publish/en-US/html/sg-Structure_Guidelines.html

4. Publication

4.1. Effectuer un rendu du document

Notre document est maintenant écrit au format **DocBook** XML mais ce format n'est pas facile à lire. Nous avons donc besoin de le publier dans un autre format. **publican** propose plusieurs rendu de sortie:

- *html*: Document html divisé en de multiples pages avec des boutons de navigation en haut et en bas de chaque page
- *html-single*: document html en une seule page
- *html-desktop*: Format identique à *html* mais avec une table des matières sur la gauche
- *pdf*: Le document est rendu en un fichier pdf
- *epub*: le document est rendu en ebook

La commande suivante va générer un document pdf

```
cd ~/Document/Publican_and_Serna publican build --lang en-US --format pdf
```

Le paramètre `--lang` est obligatoire et peut être un code de langue ou le mot clé **all**.

Le paramètre `--format` définit le format de sortie et doit être l'un des formats décrit précédemment. Il est possible de générer de multiples formats de sortie en séparant les formats par des virgules.

Quand le document est prêt pour une publication, le paramètre `--publish` doit être ajouté à la commande. Le document sera alors stocké dans le répertoire `./publish/Langues/NomDuPro-`

duit/VersionDuProduit/Fromat/NomDuDocument/ comme par exemple : `./publish/en-US/Publication_Chain/1.0/pdf/Publican_and_Serna/`. Ce sont ces fichiers qui seront automatiquement poussés vers le site web si vous créez un site de publication tel que décrit dans le chapitre suivant.

4.2. Gérer un site web

Dans ce chapitre, nous allons voir comment créer un site web qui pourra rassembler les documents et donner un accès aux versions successives de vos documents avec un minimum d'efforts.

4.2.1. Créer la structure de la page

Commençons par créer le site web en utilisant la commande **publican**:

```
mkdir ~/Documentation/WebSite
cd ~/Documentation/WebSite
publican create_site --site_config website.cfg --db_file website.db --toc_path html
```

Il crée de nombreux fichiers dans le répertoire courant:

- **website.cfg** rassemble les paramètres de configuration du site web
- **website.db** rassemble les informations concernant les documents installés sur le site web
- **html/** rassemble l'ensemble des pages du site web ainsi que les articles et livres (book)

Il est possible de surcharger le style du site web en ajoutant le formatage dans le fichier **html/site_overrides.css**. Nous allons juste créer ce fichier sans modification pour éviter d'avoir des erreurs dans le code html généré:

```
touch html/site_overrides.css
```

Editez le fichier **website.cfg** et rajoutez y les lignes suivantes:

```
def_lang: "en-US"
search: '<div/>'
manual_toc_update: 1
```

Dans cet exemple nous avons désactivé la barre de recherche et nous avons défini une mise à jour de la table des matières manuelle. Cette technique est utile si vous désirez personnaliser le contenu de la table des matières. Pour qu'un nouveau livre ou nouvel article apparaisse dans la table des matières, il faudra exécuter la commande:

```
publican update_site --site_config ~/Documentation/WebSite/website.cfg
```

Garde en mémoire que la commande écrasera les fichiers **toc.html** et que vous perdrez les modifications que vous auriez effectuées précédemment.

4.2.2. Créer la page d'accueil

La structure de notre site est maintenant prête. Nous devons maintenant créer la page d'accueil qui sera affichée par défaut lorsque qu'un utilisateur atteindra le site. Cette page est créée comme un article contenant une seule page; la page d'accueil.

Exécutez la commande suivante pour créer l'article page d'accueil:

```
cd ~/Documentation/WebSite/  
publican create --type Article --name Home_Page  
cd Home_Page
```

Ajouter dans **publican.cfg** le type d'article que nous éditons:

```
web_type: home
```

Modifiez le fichier **Home_Page.xml** qui représente la page d'accueil:

```
<?xml version='1.0' encoding='UTF-8'?>  
<!-- This document was created with Syntex Serna Free. --><!DOCTYPE article PUBLIC "-//  
OASIS//DTD DocBook XML V4.5//EN" "http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd" [  
<!ENTITY % BOOK_ENTITIES SYSTEM "Home_Page.ent">  
%BOOK_ENTITIES;  
>  
>  
<article>  
  <title>What's up Doc!</title>  
  <mediaobject>  
    <imageobject>  
      <imagedata fileref="images/title_logo.svg" align="center"/>  
    </imageobject>  
  </mediaobject>  
  <para>The web site is not a blog, it is not a standard web site but it is a web site  
  gathering various documentation I wrote around Linux. I write these documentation to  
  remember how to do the thing with Linux and while I doing technical researches.</para>  
</article>
```

Dans cet exemple, nous ajoutons simplement une image et un petit mot d'accueil.

La page est maintenant prête à être publiée sur le site web.

Pour faire ceci, exécutez la commande suivante:

```
publican build --publish --formats html-single --embedtoc --langs all  
publican install_book --site_config ~/Documentation/website.cfg --lang all
```

Le paramètre `--embedtoc` dit à **publican** d'ajouter la table des matières du site sur la gauche de la page.

Pour ajouter un logo dans l'angle haut gauche de la page, vous devez créer l'image: `~/Documentation/WebSite/Home_Page/en-US/images/web_logo.png` et publier les modifications de la page d'accueil avec la commande précédente.

4.2.3. Publier un article

Publier un article ou un livre (book) sur le site web s'effectue avec les commandes suivantes:

```
cd ~/Documentation/Publican_and_Serna/  
publican build --formats=html,html-single,txt,pdf,epub --langs=en-US --embedtoc --publish  
publican install_book --site_config ~/Documentation/WebSite/website.cfg --lang en-US
```

Le retrait d'un article peut s'effectuer avec la commande suivante:

```
cd ~/Documentation/Publican_and_Serna/  
publican remove_book --site_config ~/Documentation/WebSite/website.cfg --lang en-US
```

Et voici à quoi ressemble le site web ainsi créé



Sur cette copie d'écran, le fichier **toc.html** a été modifié afin de rendre accessible les formats *txt* et *DocBook*

5. Traduction

Notre document et notre site web sont maintenant disponible en Anglais. Il serait intéressant d'y avoir accès dans un autre langage. Nous allons voir dans ce chapitre comment facilement traduire un document créé avec notre chaîne de publication.

Quand le document est gelé, nous pouvons le préparer en vue de sa traduction. Cette préparation va consister à extraire toutes les chaînes de texte du document dans des fichiers *.pot*.

Ceci est réalisé avec la commande:

```
cd ~/Documentation/Publican_and_Serna
publican update_pot
```

Ensuite vous devez choisir le langage que vous souhaitez ajouter et exécuter la commande:

```
cd ~/Documentation/Publican_and_Serna
publican update_po --lang=fr-FR
```

Dans cet exemple nous décidons de créer la version Française du document. Cette commande crée le fichier **.po** dans le sous répertoire *fr-FR*. Les fichiers **.po** sont ceux que nous allons utiliser pour la traduction.

La commande **publican** avec les paramètres *update_pot* puis *update_po* doit être exécutée quand le document original a été modifié. Ces commandes mettrons à jour les chaînes contenues dans les fichiers de traduction.

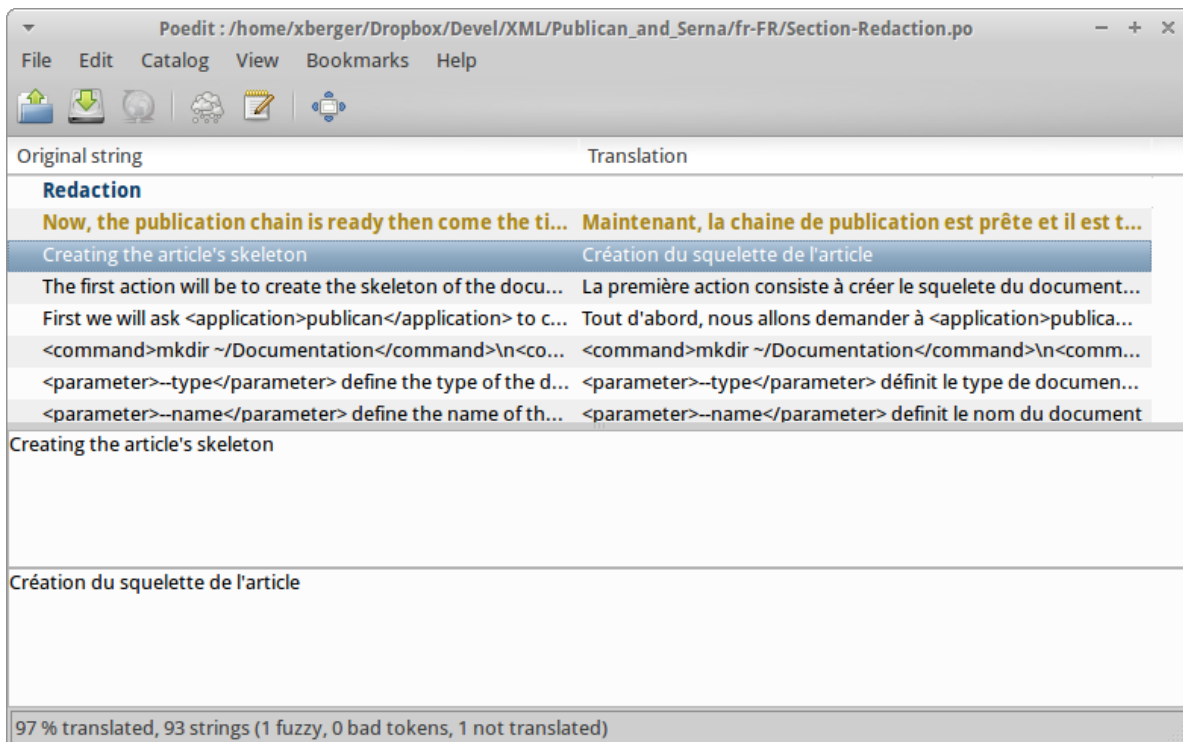
Un fichier **.po** contient toutes les chaînes et leurs traductions. Si vous ouvrez un fichier **.po** vous verrez le genre de lignes suivantes:

```
#. Tag: para#, no-c-formatmsgid
ra#, no-c-
formatmsgid"<emphasis role=\"bold\">Publication chain</emphasis>"
msgstr "<emphasis role=\"bold\">Une chaîne de publication</emphasis>"
```

Les lignes commentées décrivent le tag et l'état de la traduction. *msgid* est le texte original et *msgstr* la traduction.

Il est possible d'éditer directement ce genre de fichier mais il existe des applications qui rendent le travail plus facile.

poedit est ce genre d'application. Démarrez l'application avec le menu **Ubuntu** → **Developpement** → **Poedit**. Après avoir défini les informations dans la fenêtre des préférences, vous pouvez ouvrir un fichier **.po** avec le menu **Fichier** → **Ouvrir...** Dans cet exemple, nous avons ouvert le fichier `~/Documentation/Publican_and_Serna/fr-FR/Section-Redaction.po` et traduit quasiment l'ensemble du document.



Les chaînes pas encore traduites apparaissent en bleu. Les chaînes en jaune sont dites floues (fuzzy). Cela signifie que la source a été modifiée depuis la dernière traduction et que le traducteur devrait vérifier la concordance de la traduction. Ceci arrive quand les fichiers **.pot** et **.po** sont mis à jour suite à la modification du document original.

Les deux zones inférieures affichent le texte original et permettent au traducteur de traduire les phrases.

Pour créer une nouvelle langue, il est aussi nécessaire de traduire la page d'accueil du site web en utilisant la procédure que nous venons de décrire.

La publication de la nouvelle traduction s'effectue comme une publication standard mais avec un autre code de langue. Pour faire ceci, exécutez la commande suivante:

```
cd ~/Documentation/Publican_and_Serna/  
publican build --formats=html,html-single,txt,pdf,epub --langs=fr-FR --embedtoc --publish  
publican install_book --site_config ~/Documentation/WebSite/website.cfg --lang en-FR
```

et, pour le site web:

```
cd ~/Documentation/WebSite/Home_Page/  
publican build --formats=html-single --langs=fr-FR --embedtoc --publish  
publican install_book --site_config ~/Documentation/WebSite/website.cfg --lang en-FR
```

Maintenant nous devons mettre à jour la table des matières avec la commande suivante et sans oublier de faire une sauvegarde du fichier **toc.html** qui sera écrasé.

```
publican update_site --site_config ~/Documentation/WebSite/website.cfg
```

Ceci conclut notre article sur l'installation et l'usage d'une chaîne de publication. Pour aller plus loin, vous pouvez consulter la documentation officielle (en Anglais) à:

- <http://jfean.fedorapeople.org/en-US/Publican/>

A. Historique des modifications

Version 1.0 Fri Sep 21 2012

Xavier Berger berger.xavier@gmail.com

Ed1

Version initiale du document



Note: Possibilité d'amélioration de document

- Personnalisation du **Branding**
- **svn** pour gérer l'évolution des documents dans le temps
- **sigil** pour éditer les epub est peut-être un complément à étudier
- Ajouter des **scripts** pour faciliter l'usage de publican

