

# TP5 : Construction d'un feuille de style XSLT

Jan 2011

## 1 Experiences XSLT

Dans ce petit exercice, nous allons faire des expériences avec XSLT sous contrôle d'Oxygen.

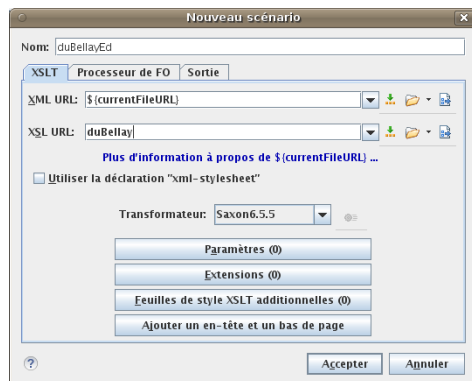
XSLT est un langage très simple, mais pour le maîtriser il faut faire un peu d'effort. Vous trouverez plusieurs sites sur Internet qui peuvent vous aider. Dans ce petit exercice nous espérons vous donner un avant-goût de ces possibilités.

L'application classique de XSLT est de transformer un document TEI XML en HTML pour le visualiser dans un navigateur. On commence donc par là.

## 2 Alimentation du navigateur

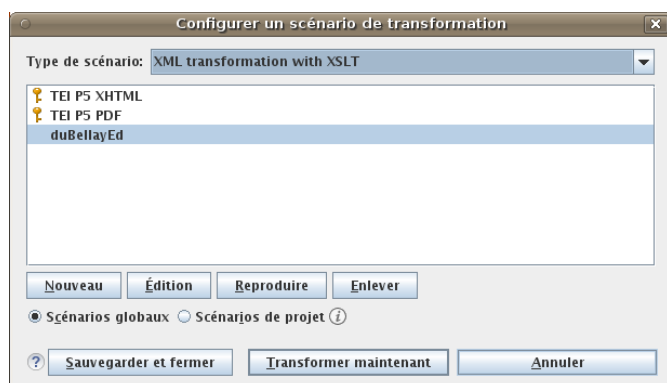
- Ouvrez Oxygen. Sélectionnez Ouvrir sur le menu Fichier et rouvrez le fichier duBellaEd.xml que vous avez enregistré à la fin de l'exercice précédent, c'est-à-dire la version enrichie avec des balises `<choice>`.
- Sur le menu Document sélectionnez Transformation, et ensuite Configurer un scénario de transformation ou tapez `ctrl-maj-C`, ou bien cliquez sur le bouton à droite du triangle rouge sur la barre d'outils.

Cliquez d'abord sur nouveau pour ouvrir la boîte de dialogue Nouveau Scénario.



- Tapez `duBellaEd.xsl` dans le champ XSL URL.
- Cliquez sur l'onglet **Sortie**
- Tapez `output.html` dans le champ Enregistrer sous
- Cochez la case **Ouvrir dans un navigateur**
- Cliquez sur **Accepter**

Vous avez maintenant créé un nouveau scénario de transformation, qui s'appelle duBellaEd.



- Cliquez sur **Transformer maintenant**
- En bas de l'écran, le message **Transformation réussie** apparaît...
- .. et après un bref délai, votre navigateur devrait s'ouvrir pour afficher le fichier `output.html` que vous venez de créer.
- C'est joli, hein ? Peut-être y a-t-il encore un peu de travail à faire...

### 3 Une feuille de style XSLT

Une feuille de style XSLT est un document XML. Il peut donc être édité avec Oxygen.

- Allez de nouveau sur **Fichier - Ouvrir**. Ouvrez le fichier `duBella.xml` dans votre dossier de travail.
- Ce fichier XSLT contient un seul template, qui correspond à l'élément `<TEI>`, l'élément racine de notre document. Son effet sera de produire les balises `<html>` et `</html>`, avec entre les deux le résultat du `<xsl:apply-templates>`
- Ce dernier va essayer d'appliquer tous les templates disponibles... il n'y en a pas, donc il ne va produire que le texte du document.
- ... et en effet, si vous revenez sur votre navigateur et regardez la source du fichier, c'est ce qu'il a fait.

Au travail ! D'abord, nous allons extraire de l'en-tête un titre pour le document HTML, et supprimer le reste du TEI Header.

- Après le `<html>`, tapez `<head>`
- Oxygen ajoute la balise fermante. C'est bien. Continuez donc. Ajoutez `<title>` à l'intérieur de l'élément `<head>`. Pour trouver le titre du document il faut naviguer de la racine du document (l'élément `TEI`) jusqu'à l'élément `<title>`, qui se trouve dans le `<titleStmt>`, dans le `<fileDesc>`, dans le `<teiHeader>`. Cela s'effectue avec le *Xpath* suivant.
- Tapez `<xsl:value-of select="teiHeader/fileDesc/titleStmt/title"/>`.
- Modifiez la balise `<xsl:apply-templates>`, en ajoutant `select="text"` et entourez-la d'un élément HTML `<body>`
- Cliquez sur le bouton **Indentation**. Votre feuille de style maintenant devrait ressembler à ceci :

```

1 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
2   xpath-default-namespace="http://www.tei-c.org/ns/1.0" version="2.0">
3
4 <xsl:template match="TEI">
5   <html>
6     <head>
7       <title>
8         <xsl:value-of select="teiHeader/fileDesc/titleStmt/title"/>
9       </title>
10    </head>
11   <body>
12     <xsl:apply-templates select="text"/>
13   </body>
14 </html>
15 </xsl:template>
16
17 </xsl:stylesheet>
18

```

- Cliquez sur l'icône de la disquette (ou tapez ctrl-s) pour enregistrer les modifications que vous venez de faire.
- Cliquez sur l'onglet duBellaEd.xml pour revenir sur votre document XML.
- Cliquez sur l'icône triangle-rouge pour relancer la transformation (ou tapez CTRL-maj-T)
- Voyons ce que cela donne : cette fois, on ne voit que le `<text>` de notre document. On fait des progrès !

Maintenant, on va ajouter des templates.

- Revenez sur la feuille de style.
- Ajoutez
  - un template pour `<l>`, qui va ajouter une balise `<br/>` après chaque vers
  - un template pour `<head>`, qui va l'entourer de `<h2>`
  - un template pour `<lg>`, qui va l'entourer de `<p>`, et en plus le préfixer du nombre de chaque strophe.
- Voici les templates requis... essayez de comprendre par vous-même le fonctionnement de chacun d'entre eux.

```
..
.7 ▾ <xsl:template match="1">
.8   <xsl:apply-templates/><br/>
.9   </xsl:template>
20
21 ▾ <xsl:template match="head">
22   <h2><xsl:apply-templates/></h2>
23   </xsl:template>
24
25 ▾ <xsl:template match="lg">
26   <xsl:number/>
27   <p><xsl:apply-templates/></p>
28   </xsl:template>
29
```

### 4 L'avantage du XSLT

Jusque là, presque tout ce que l'on a fait est de l'affichage. Nous voyons mieux la puissance du langage XSLT en réfléchissant sur les éléments `<choice>`, éléments dont notre document est dotés.

L'idéal serait d'avoir une feuille de style qui pourrait supprimer ou bien les `<reg>`, ou bien les `<orig>`, en fonction du souhait de l'utilisateur. Pour le moment, on va simplement ajouter un template qui va supprimer les `<orig>`.

Tout ce qu'il faut, c'est ajouter un template comme ceci : `<xsl:template match="orig"/>`. Essayez-le. Est-ce que vous comprenez comment cela fonctionne ? Si oui, vous avez bien compris les principes de XSLT !