

# XSLT

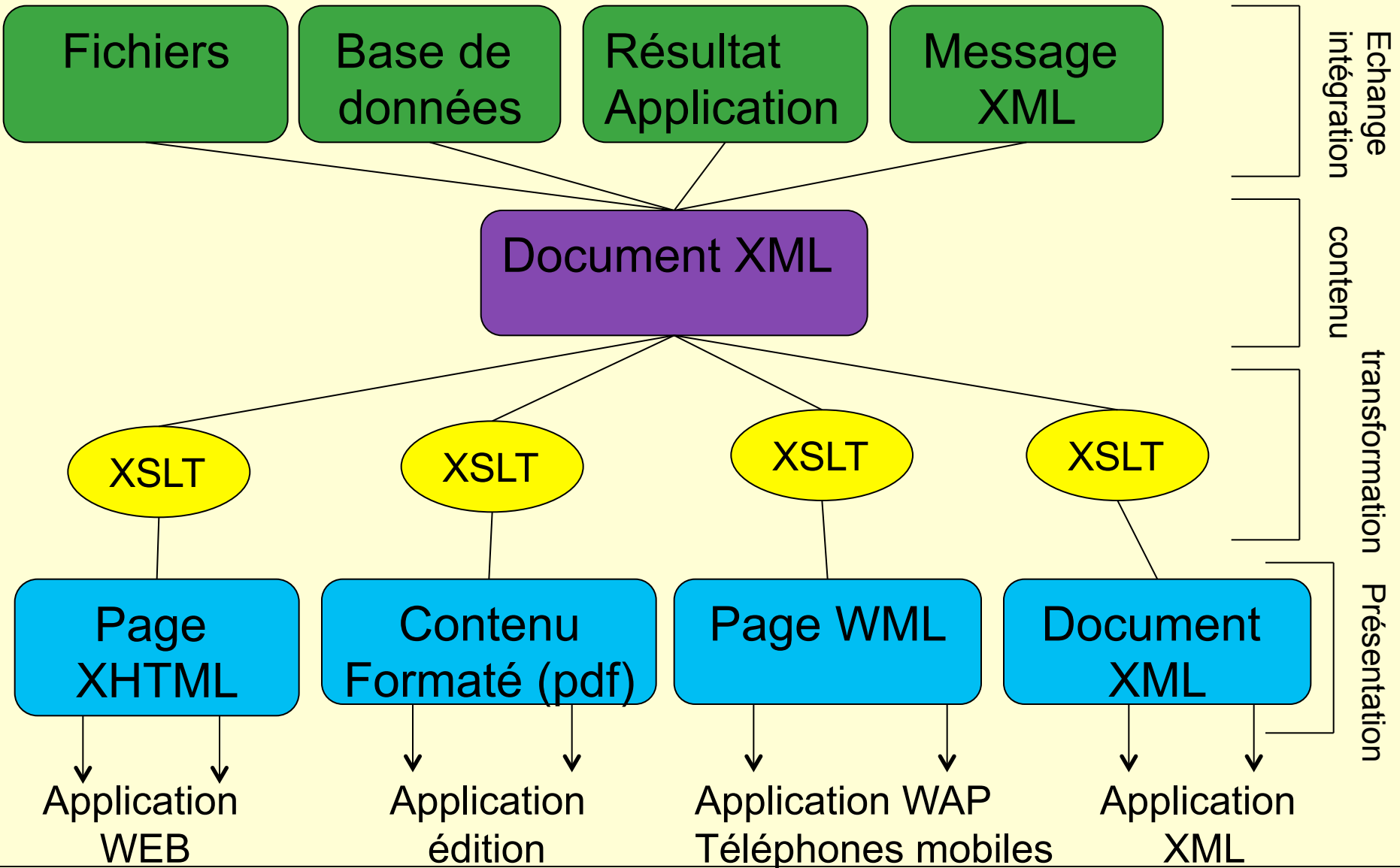
## eXtensible Stylesheet Language Transformation

- Echange et intégration avec XML
- Navigation dans un document XML: XPATH
- Visualisation d'un document XML
  - Programmes XSLT
  - Règles XSLT

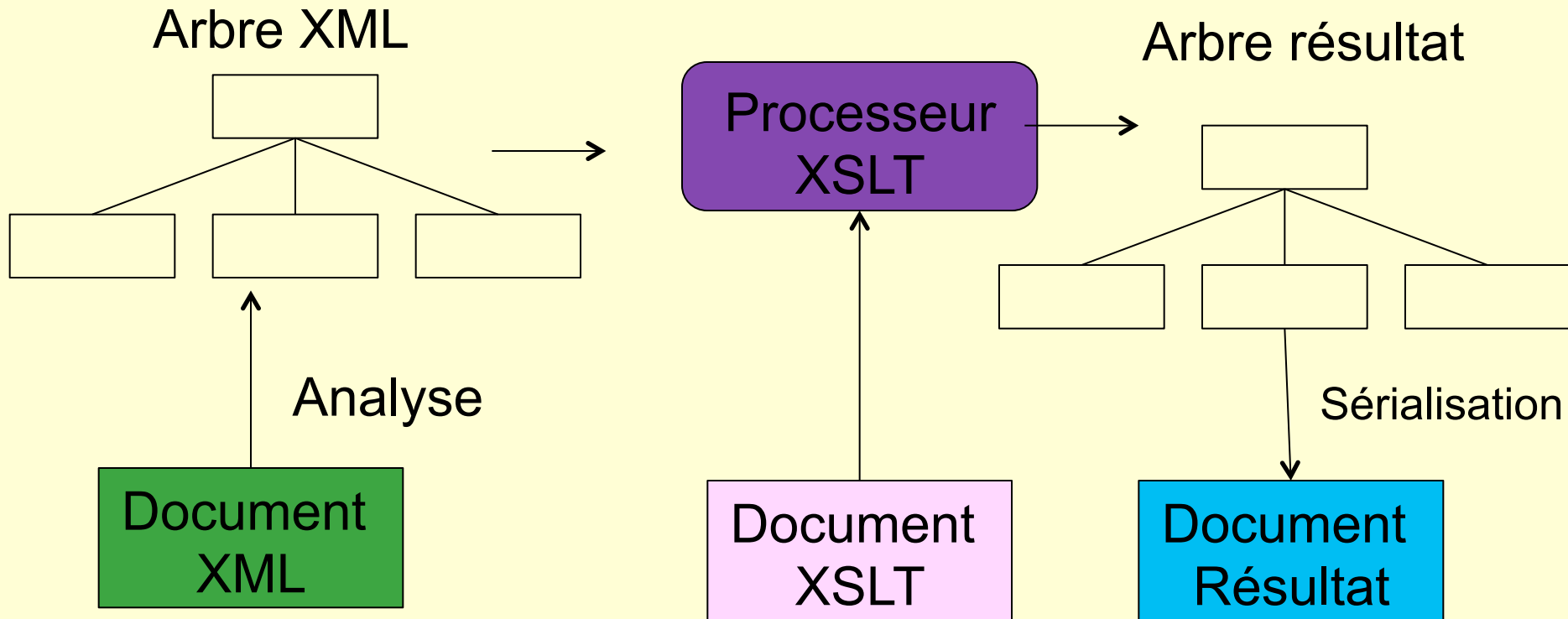
Références:

Comprendre XSLT, Bernd Amann et Philippe Rigaux, eds O'Reilly

# Echange et intégration avec XML



# Visualisation d'un document XML



# Structure d'un programme XSLT pour visualisation

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <!--Sortie html, par défaut la sortie est xml -->
  <xsl:output method="html"/>
  <!-- Règle racine -->
  <xsl:template match="/">
    <html>
      <head> <title> titre du document resultat </title></head>
      <body >
        <!-- Jeu de règles qui déclarent comment traduire des éléments XML
        <xsl:for-each> <xsl:value-of> <xsl:apply-templates>, <xsl:copy-of> etc.... -->
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

# Association à un fichier xml

- Association avec l'instruction de traitement xml-stylesheet

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="message.xsl" ?>

<message >

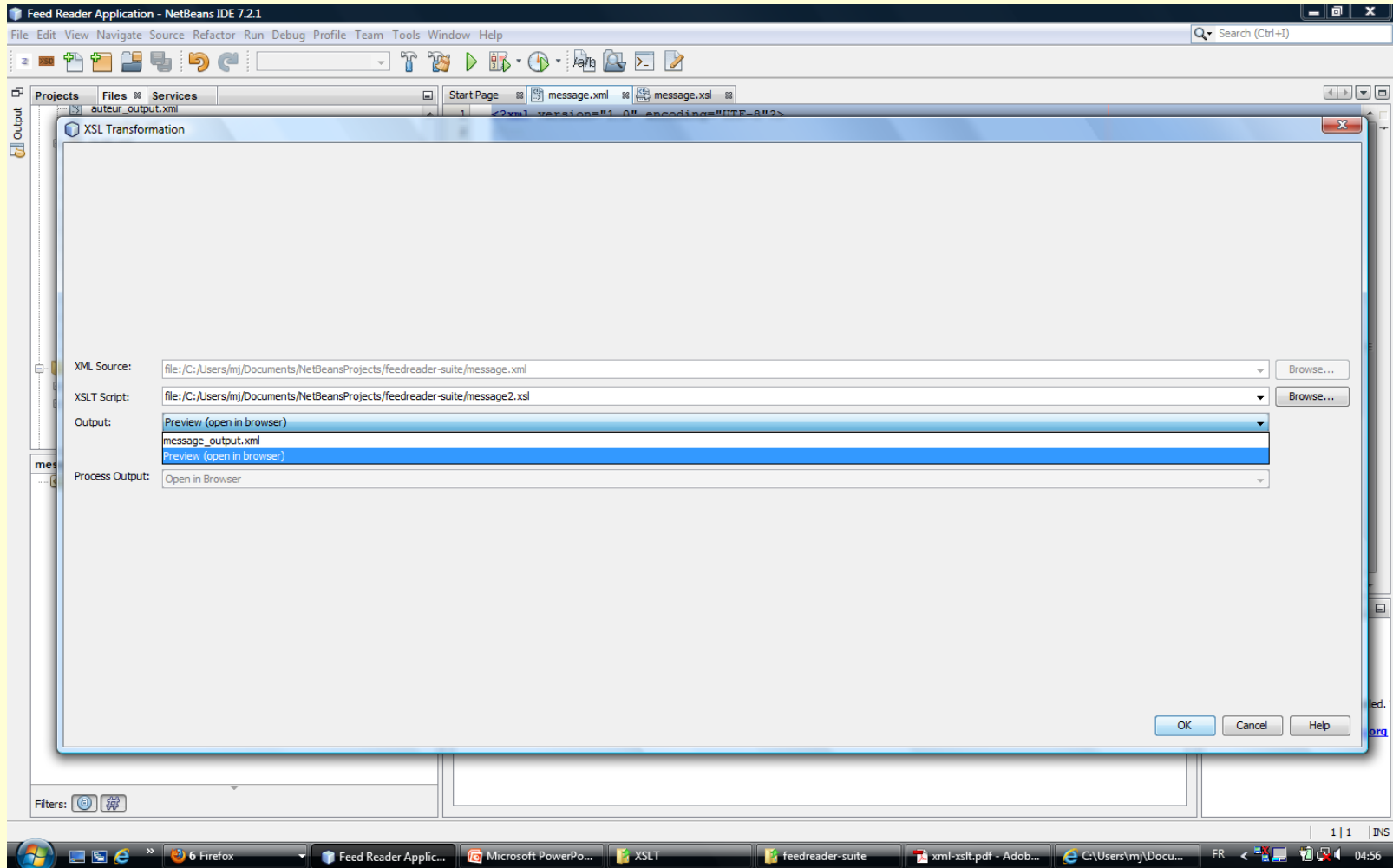
<!-- définition des éléments du message -->

</message>

- Association indirecte

Sous NetBeans, onglet XST transformation

# Sous NetBeans, XST transformation



# Elément xsl:output

Indique au processeur le type de document à produire,  
html, xml ou text

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
  <xsl:output method="html"  
    doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd"  
    doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN"  
    indent="yes"  
    encoding="iso-8859-1" />  
</xsl:stylesheet>
```

Par Défaut transformation XML

# Element xsl:template

Indique une règle de traduction à appliquer au nœud contexte

## Notation

```
<xsl:template match="XPATH_expression">
```

Contenu ou corps de la règle

```
</xsl:template>
```

Définit l'ensemble des nœuds à transformer

```
<xsl:template name="nom_règle">
```

Contenu ou corps de la règle

```
</xsl:template>
```

Règle nommée

Instanciation d'une règle

s'effectue pour un nœud du document source appelé nœud courant

Corps de la règle

chemin Xpath à partir du nœud courant

```
<xsl:template match="/">
```

règle racine

```
<xsl:template match="message/titre">
```

règle appliqué aux nœud titre de message

```
<xsl:template name="ligne">
```

tableau html

règle qui définit la construction d'une ligne de



# Elément xsl:apply-templates

Désigne un ensemble de nœuds avec une expression XPATH et demande l'application d'une règle pour chaque nœud

## Notation

toujours un ensemble de nœuds

**<xsl:apply-templates** [select="XPATH\_expression"] />

## Exemples

**<xsl:apply-templates />**

applique les règles sur les nœuds descendants suivant leur ordre de priorité

**<xsl:apply-templates select="message/titre" />**

applique la règle associée à message/titre

# Règle Racine

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/
  Transform" version="1.0">
  <xsl:output method="xml"/>
  <xsl:template match="/">
    <message>
      <xsl:apply-templates />
    </message>
  </xsl:template>
</xsl:stylesheet>
```

← Règle racine

Corps de la règle

Application des règles

## message\_result.xml

Application des règles par défaut sur les nœuds du document  
Attributs non pris en compte par les règles par défaut

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<message>
```

Festival des Berniques en folie

Association Rock Islais

Place de La Mairie, 85350 Ile d'Yeu

A tous les curieux et amoureux de la mer

Venez nombreux au 13ième festival des Berniques en Folie  
qui se déroule à l'Ile d'Yeu  
du 31 octobre au 3 novembre 2012.

```
</message>
```

## Element xsl:value-of

Evaluer une expression sur le document source puis convertir le résultat en chaîne de caractères

### Notation

**<xsl:value-of select="XPATH\_expression" />**

### Exemples

**<xsl:value-of select="message/titre" />**

Festival des Berniques en folie

**<xsl:value-of select="message/auteur/nom" />**

Association Rock Islais

**<xsl:value-of select="message/salutation" />**

A tous les curieux et amoureux de la mer

### Fonction utile

**concat(string<sub>1</sub>, ..., string<sub>n</sub>)** concatène les chaînes de caractères

## Message4.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:output method="html"/>
  <xsl:template match="/">
    <html>
      <head> <xsl:apply-templates select="message/titre" /> </head>
      <body bgcolor="blue">
        </body>
      </html>
    </xsl:template>
    <xsl:template match="message/titre">
      <title> <xsl:value-of select=""/></title>
    </xsl:template>
  </xsl:stylesheet>
```

Noeud titre du message

Contenu du nœud courant

# Element xsl:copy-of

Copie un nœud , ainsi que tous ses descendants

## Notation

**<xsl:copy-of select="XPATH\_expression" />**

## Exemple

**<xsl:copy-of select=" message/adresse" />**

**<adresse> Place de La Mairie, 85350 Ile d'Yeu </adresse>**

**<xsl:copy-of select="message/contenu" />**

**<contenu> Venez nombreux au <i> 13i&egrave;me festival des  
Berniques en Folie </i> <BR> qui se d&eacute;roule &agrave;  
l'Ile d'Yeu <BR> du 31 octobre au 3 novembre 2012. </contenu>**

# Message.xsl

(...)

```
<xsl:template match="/">
```

```
  <html>
```

```
    <head>
```

```
      <title> <xsl:value-of select="message/titre/text()"/></title>
```

```
    </head>
```

```
    <body bgcolor="blue">
```

```
      <P> <B> <xsl:value-of select="message/titre"/></B> </P>
```

```
      <P><xsl:value-of select="message/auteur/nom"/></P>
```

```
      <P><xsl:copy-of select="message/adresse"/></P>
```

```
      <ol><xsl:value-of select="message/salutation" /> </ol>
```

```
      <ol><xsl:copy-of select="message/contenu" /></ol>
```

```
    </body>
```

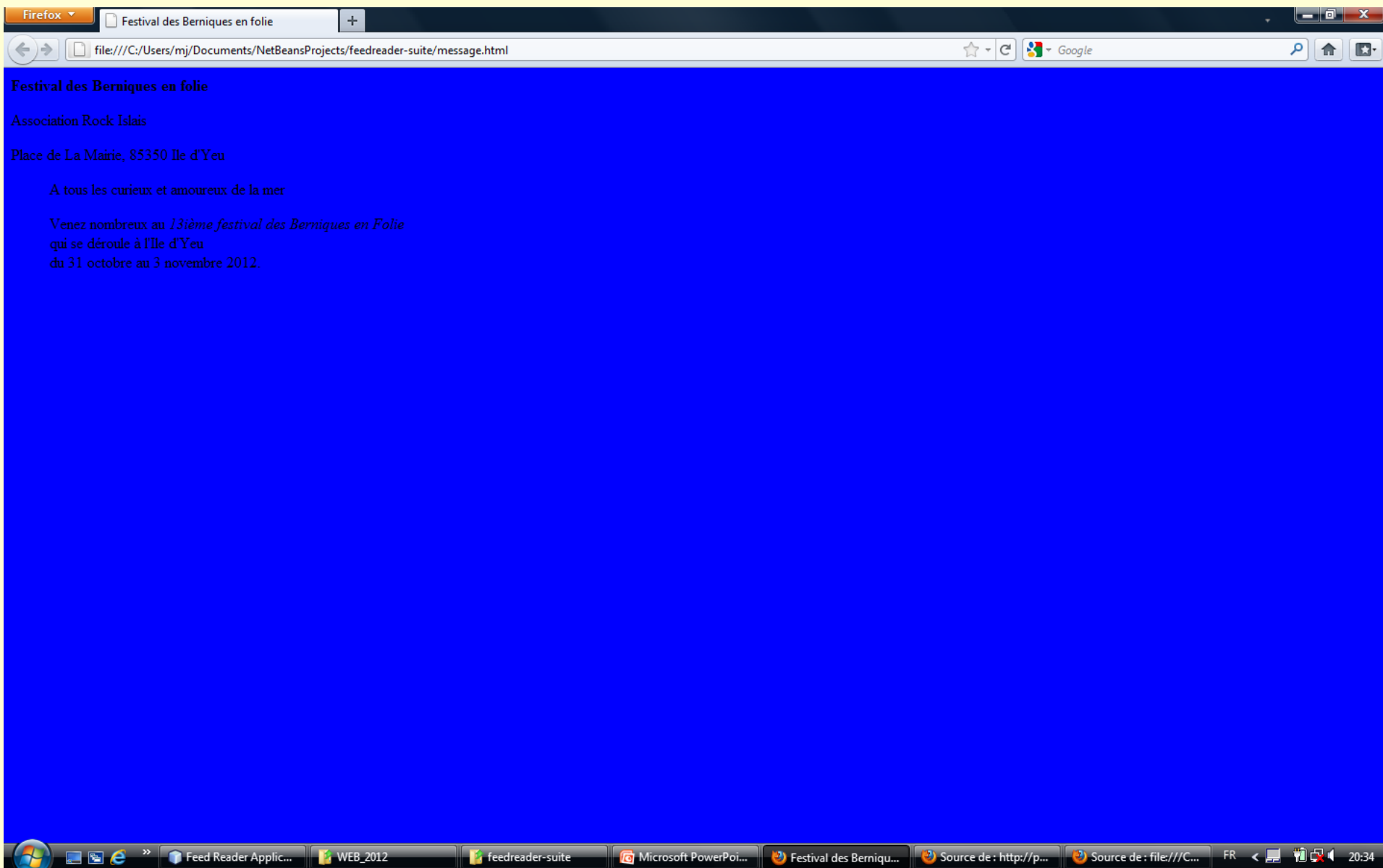
```
  </html>
```

```
</xsl:template>
```

```
<html>
<head> <META http-equiv="Content-Type" content="text/html;
      charset=UTF-8">
<title> Festival des Berniques en folie</title> </head>
<body bgcolor="blue">
<P> <B> Festival des Berniques en folie</B> </P>
<P>Association Rock Islais</P>
<P> <adresse> Place de La Mairie, 85350 Ile d'Yeu </adresse> </P>
      <ol>A tous les curieux et amoureux de la mer</ol>
<ol> <contenu>Venez nombreux au <i> 13i&egrave;me festival des
      Berniques en Folie </i> <BR> qui se d&eacute;roule &agrave; l'Ile
      d'Yeu <BR> du 31 octobre au 3 novembre 2012.</contenu> </ol>
</body> </html>
```



# Visualisation



## Fonctions définies sur un ensemble de nœuds

<b>position()</b>	position du nœud courant
<b>name()</b>	nom de la balise du nœud courant
<b>last()</b>	position du dernier nœud dans le contexte
<b>count()</b>	taille d' un ensemble de nœuds
<b>.</b>	Contenu du nœud courant

## Elément xsl:for-each

Instancie son contenu pour chacun des nœuds désignés par l'expression de l'attribut select

⇒ Le nœud est pris à chaque fois comme nœud courant

### Notation

```
<xsl:for-each select="XPATH_expression" >  
  (xsl:sort*,corps de la règle )  
</xsl:for-each>
```

# Visualisation de livres.xml

(...) <xsl:template match="/">

<html>

<head> <title> Illustration For each </title> </head>

<body bgcolor="grey"> <xsl:apply-templates select="livres"/>

</body>

</html>

</xsl:template>

Règle  
racine

<xsl:template match="livres">

Règle sur livres

<table> <dl>

<xsl:for-each select="livre">

Itération sur livre

<!-- traitement du noeud livre courant -->

</xsl:for-each>

</dl> </table>

</xsl:template>

# Traitement du livre courant

position() retourne la position du livre courant

last() le nombre de nœuds descendants de livres

```
<xsl:template match="livres">
```

```
  <table>  <dl>
```

```
    <xsl:for-each select="livre">
```

```
      <tr>
```

```
        <td>  <i>
```

```
          <xsl:value-of select="concat(position(),' /',last(),' ',@id, ' :')" />
```

```
        </i>  </td>
```

```
      <td>  <xsl:copy-of select="./titre" />  </td>
```

```
    </tr>
```

```
  </xsl:for-each>
```

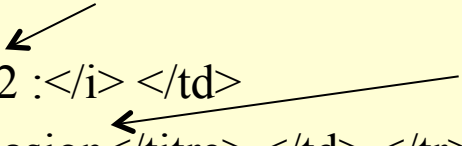
```
</dl>
```

```
</table>
```

```
</xsl:template>
```

# Source html

```
<html> <head>
<META http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title> Illustration For each </title>
</head> <body bgcolor="yellow">
<table>
    <xsl:value-of select="concat(position(), ' ', last(), ' ', @id, ' :')" />
    <dl> <tr>
<td> <i>1 /5 AJ02 :</i> </td>
<td> <titre>Persuasion</titre> </td> </tr>
<tr> <td><i>2 /5 DL01 :</i></td>
<td> <titre>Gone, Baby Gone</titre> </td> </tr>
<tr> <td><i>3 /5 DL02 :</i></td>
<td> <titre>Pri&egrave;res pour la pluie, (Prayers for Rain)</titre> </td> </tr>
<tr> <td><i>4 /5 DL03 :</i></td>
<td> <titre>T&eacute;n&egrave;bres, prenez-moi la main, (Darkness, Take My Hand)</
    titre> </td> </tr>
<tr> <td><i>5 /5 DL04 :</i></td>
<td> <titre>Un dernier verre avant la guerre (A Drink Before the War)</titre> </td> </tr>
    </dl> </table> </body> </html>
```

The diagram consists of two arrows. One arrow points from the text `<xsl:value-of select="concat(position(), ' ', last(), ' ', @id, ' :')" />` to the `<i>` tag in the first row of the table. The second arrow points from the text `<xsl:copy-of select="./titre" />` to the `<titre>` tag in the first row of the table.

# Visualisation

The screenshot shows a web browser window with the address bar displaying `file:///C:/Users/mj/Documents/NetBeansProjects/feedreader-suite/livres.html`. The browser has several tabs open, including 'Illustration For each' and 'Livres'. The main content area is yellow and displays a list of books with annotations:

- 1 /5 AJ02 : Persuasion
- 2 /5 DL01 : Gone, Baby Gone
- 3 /5 DL02 : Prières pour la pluie, (Prayers for Rain)
- 4 /5 DL03 : Ténèbres, prenez-moi la main, (Darkness, Take My Hand)
- 5 /5 DL04 : Un dernier verre avant la guerre (A Drink Before the War)

An arrow points from the first part of the list to a text box that says: "1,2,3,4 et 5 désignent la position du livre courant".

Below this, another text box explains the notation: "x/5 où 5 désigne le nombre de nœuds *livre*".

The Windows taskbar at the bottom shows several open applications: Microsoft PowerPoint, Feed Reader Application, and two instances of 'Illustration For each ...'. The system clock indicates the date is 10/22/14 and the time is 18:26.

# For-each avec appel de règle nommée

(...) <xsl:template match="/">

<html>

<head> <title> Livres </title> </head>

<body bgcolor="grey"> <xsl:apply-templates select="livres"/> </body>

</html>

</xsl:template>

<xsl:template match="livres">

<table>

<xsl:for-each select="livre">

<tr> <xsl:call-template name="ligne" /></tr>

</xsl:for-each>

</table>

</xsl:template>

<xsl:template name="ligne">

<td> <i> <xsl:value-of select="concat(@id, ' :')" /> </i> </td>

<td> <xsl:copy-of select="./titre" /></td>

</xsl:template>

Règle  
racine

Règle  
livres

Règle  
livre



# Source html

```
<html> <head>
<META http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title> Livres </title> </head>
<body bgcolor="grey">
<table> <tr> <td><i>AJ02 :</i></td>
<td> <titre>Persuasion</titre> </td> </tr>
<tr> <td><i>DL01 :</i></td>
<td> <titre>Gone, Baby Gone</titre> </td> </tr>
<tr> <td><i>DL02 :</i></td>
<td> <titre>Pri&egrave;res pour la pluie, (Prayers for Rain)</titre> </td> </tr>
<tr> <td><i>DL03 :</i></td>
<td> <titre>T&eacute;n&egrave;bres, prenez-moi la main, (Darkness, Take My
    Hand)</titre> </td> </tr>
<tr> <td><i>DL04 :</i></td><td> <titre>Un dernier verre avant la guerre (A Drink
    Before the War)</titre> </td> </tr>
</table> </body> </html>
```

# Visualisation

The screenshot shows a web browser window with the address bar displaying `portbelosta:8082/servlet/org.netbeans.modules.xml.transform.TransformServlet/`. The browser has several tabs open, including 'Illustration For each' and 'Livres'. The main content area displays a list of books on the left and an XSL-FO template on the right.

Books listed:

- AJ02 : Persuasion
- DL01 : Gone, Baby Gone
- DL02 : Prières pour la pluie, (Prayers for Rain)
- DL03 : Ténèbres, prenez-moi la main, (Darkness, Take My Hand)
- DL04 : Un dernier verre avant la guerre (A Drink Before the War)

XSL-FO Template:

```
<xsl:template name="ligne">
  <td> <i> <xsl:value-of select="concat(@id, ' :')" /> </i> </td>
  <td> <xsl:copy-of select="./titre" /></td>
</xsl:template>
```

The browser's taskbar at the bottom shows the Windows logo, several application icons, and the system clock indicating 18:15 on 10/22/14.

# Règles sur livres

1) Pour chaque livre introduire la ligne:

Position du livre dans le nœud courant livres / nombre de nœuds

2) Pour chaque livre, la règle *traiteLivre* va être appliquée

```
<xsl:template match="livres">
```

```
<dl>
```

```
<xsl:for-each select="livre">
```

```
<p>
```

```
<xsl:value-of select="concat(position(), '/', last(), ' Livre' )"/>
```

```
</p>
```

```
<xsl:call-template name="traiteLivre"/>
```

```
</xsl:for-each>
```

```
</dl>
```

```
</xsl:template>
```

# Règle sur traitement d'un livre

```
<xsl:template name="traiteLivre">
```

```
  <dt>
```

Traitement des attributs

```
  <xsl:for-each select="./@*">
```

```
    <xsl:value-of select="concat(position(), '/', last(), ' ', name(), ': ', ', ', '; ')" />
```

```
  </xsl:for-each>
```

```
  </dt>
```

Traitement des nœuds  
descendants de livre

```
  <xsl:for-each select="./*">
```

```
    <dd> <b>
```

```
    <xsl:value-of select="concat(position(), '/', last(), ' ', name(), ' : ')" />
```

```
    <xsl:copy-of select="." />
```

```
    </b> </dd>
```

```
  </xsl:for-each>
```

```
</xsl:template>
```

```
<body bgcolor="yellow">
```

```
<dl>
```

```
<p>1/5 Livre</p>
```

Concaténation id avec « : » et  
valeur de cet attribut pour le livre courant

```
<dt>1/2 id: AJ02; 2/2 auteurs: AustenJane; </dt>
```

```
<dd>
```

Copie du nœud titre  
du livre courant

```
<b>1/4 titre : <titre>Persuasion</titre> </b> </dd>
```

```
<dd>
```

```
<b>2/4 traduction : <traduction>Andr&eacute; Belamich</traduction> </b> </dd>
```

```
<dd>
```

```
<b>3/4 editeur : <editeur>Christian Bourgois &Eacute;diteur</editeur> </b> </dd>
```

```
<dd>
```

```
<b>4/4 parution : <parution>1980</parution> </b> </dd>
```

```
<p>2/5 Livre</p>
```

```
....
```

# Visualisation

Firefox ▾

Default Priority × Livres × Illustration For each × +

portbelosta:8082/servlet/org.netbeans.modules.xml.transform.TransformServlet/ ☆ ▾ ↻ Google 🔍

1/5 Livre

1/2 id: AJ02; 2/2 auteurs: AustenJane;  
1/4 titre : Persuasion  
2/4 traduction : André Belamich  
3/4 editeur : Christian Bourgois Éditeur  
4/4 parution : 1980

2/5 Livre

1/2 id: DL01; 2/2 auteurs: DenisLehane;  
1/4 titre : Gone, Baby Gone  
2/4 traduction : Isabelle Maillet  
3/4 editeur : Rivages/Thriller  
4/4 parution : 2003

3/5 Livre

1/2 id: DL02; 2/2 auteurs: DenisLehane;  
1/3 titre : Prières pour la pluie, (Prayers for Rain)  
2/3 editeur : Rivages/Thriller  
3/3 parution : 2004

4/5 Livre

1/2 id: DL03; 2/2 auteurs: DenisLehane;  
1/3 titre : Ténèbres, prenez-moi la main, (Darkness, Take My Hand)  
2/3 editeur : Rivages/Thriller  
3/3 parution : 2000

5/5 Livre

1/2 id: DL04; 2/2 auteurs: DenisLehane;  
1/4 titre : Un dernier verre avant la guerre (A Drink Before the War)  
2/4 traduction : Mona de Pracontal  
3/4 editeur : Rivages/Thriller  
4/4 parution : 1999

Windows taskbar: Feed Reader Applica..., Illustration For each ..., Microsoft PowerPoi..., FR, 14:35

Affiche un message sur la sortie du processeur XSLT.

- Syntaxe

**<xsl:message**

**terminate**=("yes" | "no") >

**corps de la règle**

**</xsl:message>**

- Remarque

L'attribut *terminate* avec la valeur yes permet de forcer l'interruption du programme

# Priorité des règles

- Priorité 0,5  
Règle avec expression comprenant des prédicats
- Priorité 0  
Règle avec expression XPATH constituées d'une seule étapes (avec un nom d'élément ou un attribut)
- Priorité -0,25  
Règle avec expression XPATH constituée d'une étape + un filtre
- Priorité -0,5  
Règle avec expression XPATH constitués de filtres sans espace de noms et autres qu'un nom d'élément ou attribut



# Application des règles par défaut

```
<xsl:template match="livres">
```

```
  <xsl:message> Regle livres </xsl:message>
```

```
  <dl> <xsl:apply-templates /> </dl>
```

```
</xsl:template>
```

```
<xsl:template match="livre" >
```

```
  <xsl:message> Regle noeud livre </xsl:message>
```

```
  <xsl:apply-templates />
```

```
</xsl:template>
```

```
<xsl:template match="livre/@*" >
```

```
  <xsl:message> Regle sur tout attribut </xsl:message>
```

```
  <dt> <xsl:value-of select="concat(name(), ': ', ', ', '; ')" /> </dt>
```

```
</xsl:template>
```

```
<xsl:template match="*" >
```

```
  <xsl:message> Regle sur tout noeud descendant </xsl:message>
```

```
  <dd> <b> <xsl:value-of select="concat(name(), ' :: ', '. ')" /> </b></dd>
```

```
</xsl:template>
```

# Sortie du processeur XSLT

Regle livres

Regle noeud livre

Regle sur tout noeud descendant

Regle sur tout noeud descendant

Regle sur tout noeud descendant

Regle sur tout noeud descendant

Regle noeud livre

Regle sur tout noeud descendant

Regle sur tout noeud descendant

Regle sur tout noeud descendant

Regle sur tout noeud descendant

Regle noeud livre

Regle sur tout noeud descendant

Regle sur tout noeud descendant

Regle sur tout noeud descendant

Regle noeud livre

Regle sur tout noeud descendant

Regle sur tout noeud descendant

Regle sur tout noeud descendant

Regle noeud livre

Regle sur tout noeud descendant

Regle sur tout noeud descendant

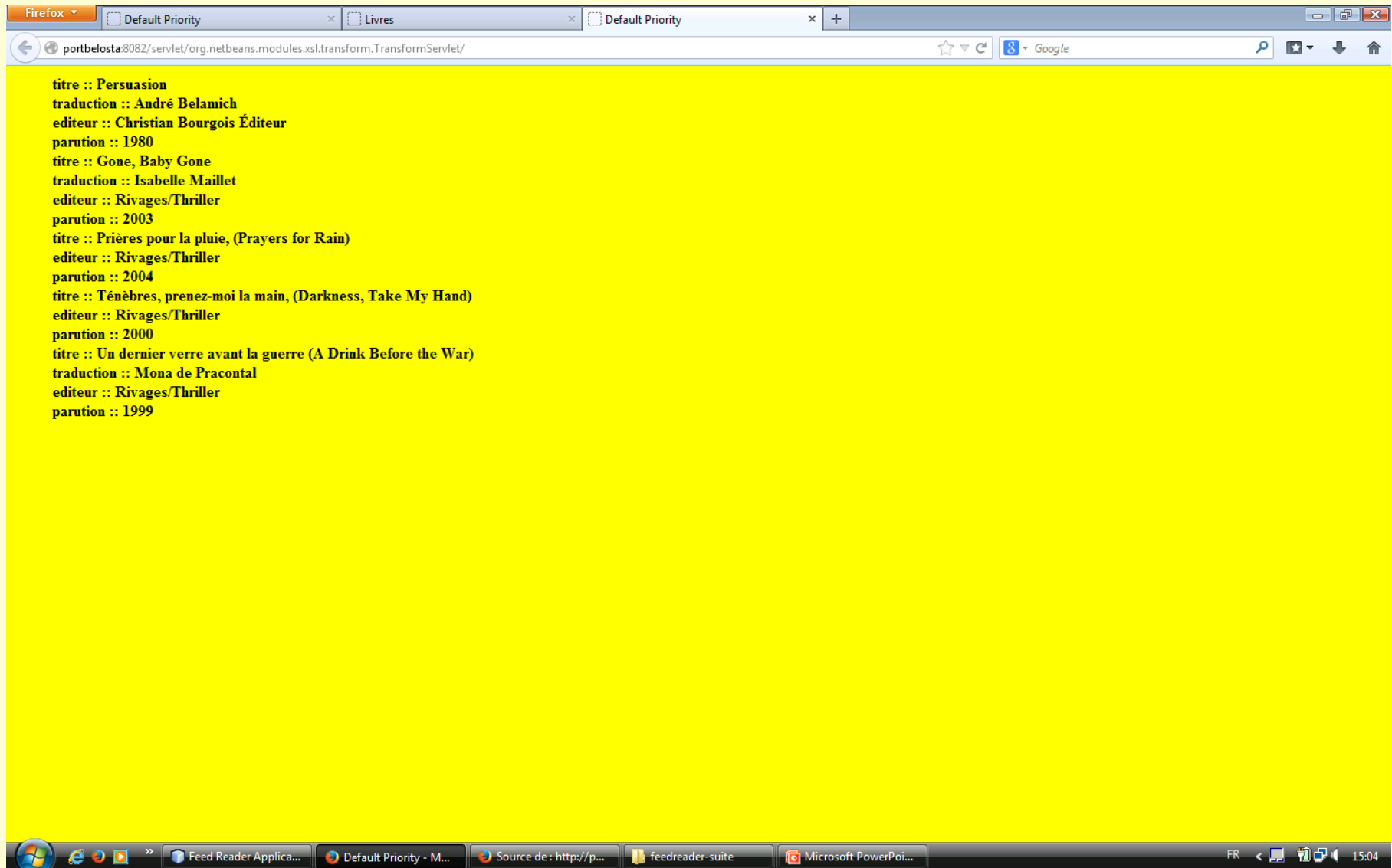
Regle sur tout noeud descendant

Regle sur tout noeud descendant

Non application de la règle  
De traitement des attributs

# Visualisation

## Non traitement des attributs



# Appel explicite sur les nœuds éléments de livre

```
<xsl:template match="livres">
```

```
  <xsl:message> Regle livres </xsl:message>
```

```
  <dl> <xsl:apply-templates select="livre" /> </dl>
```

```
</xsl:template>
```

```
<xsl:template match="livre" >
```

```
  <xsl:message> Regle noeud livre </xsl:message>
```

```
  <xsl:apply-templates select="@*|*" />
```

```
</xsl:template>
```

```
<xsl:template match="@*" >
```

```
  <xsl:message> Regle sur tout attribut </xsl:message>
```

```
  <dt><xsl:value-of select="concat(name(), ': ', ' ; ')" /> </dt>
```

```
</xsl:template>
```

```
<xsl:template match="*" >
```

```
  <xsl:message> Regle sur tout noeud descendant </xsl:message>
```

```
  <dd> <b> <xsl:value-of select="concat(name(), ' :: ', text())" /> </b> </dd>
```

```
</xsl:template>
```

Regle noeud livre  
Regle sur tout attribut  
Regle sur tout attribut  
Regle sur tout noeud descendant  
Regle sur tout noeud descendant  
Regle sur tout noeud descendant  
Regle sur tout noeud descendant  
Regle noeud livre  
Regle sur tout attribut  
Regle sur tout attribut  
Regle sur tout noeud descendant  
Regle sur tout noeud descendant  
Regle sur tout noeud descendant  
Regle noeud livre  
Regle sur tout attribut  
Regle sur tout attribut  
Regle sur tout noeud descendant  
Regle sur tout noeud descendant  
Regle sur tout noeud descendant  
Regle noeud livre  
Regle sur tout attribut  
Regle sur tout attribut  
Regle sur tout noeud descendant  
Regle sur tout noeud descendant  
Regle sur tout noeud descendant  
Regle sur tout noeud descendant  
Transformation finished.

# Visualisation

The screenshot shows a Firefox browser window with a yellow background. The address bar displays the URL `portbelosta:8082/servlet/org.netbeans.modules.xml.transform.TransformServlet/`. The page content is XML data representing a list of books. The browser's tab bar shows several tabs, most labeled 'Default Priority', with one labeled 'Livres'. The Windows taskbar at the bottom includes icons for the Start menu, Internet Explorer, a media player, and several application windows including 'Feed Reader Applica...', 'Default Priority - M...', 'Source de : http://p...', 'feedreader-suite', and 'Microsoft PowerPoi...'. The system clock in the bottom right corner shows the date '10/22/14' and time '15:23'.

```
id: AJ02;  
auteurs: AustenJane;  
  titre :: Persuasion  
  traduction :: André Belamich  
  editeur :: Christian Bourgois Éditeur  
  parution :: 1980  
id: DL01;  
auteurs: DenisLehane;  
  titre :: Gone, Baby Gone  
  traduction :: Isabelle Maillet  
  editeur :: Rivages/Thriller  
  parution :: 2003  
id: DL02;  
auteurs: DenisLehane;  
  titre :: Prières pour la pluie, (Prayers for Rain)  
  editeur :: Rivages/Thriller  
  parution :: 2004  
id: DL03;  
auteurs: DenisLehane;  
  titre :: Ténèbres, prenez-moi la main, (Darkness, Take My Hand)  
  editeur :: Rivages/Thriller  
  parution :: 2000  
id: DL04;  
auteurs: DenisLehane;  
  titre :: Un dernier verre avant la guerre (A Drink Before the War)  
  traduction :: Mona de Pracontal  
  editeur :: Rivages/Thriller  
  parution :: 1999
```

## Création explicite d' une balise html *img*

### Notation

```

```

### Exemples

**Nom de l'image est le contenu de l'élément *image***

```
<image> nom image</image>
```

```
<p>  </p>
```

**Nom de l'image est dans l'attribut *source***

```
<image2 source="nomImage"> texte image</image2>
```

```
<p> </p>
```

```
<p> <xsl:value-of select="."/> </p>
```

## Exemple: page.xml

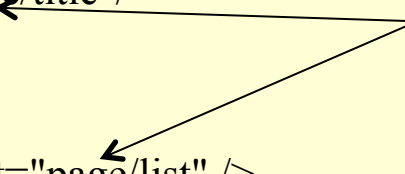
```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="images.xsl" ?>
<page>
<title>Photos 2011</title>
<list>
<image> ///C:/Users/mj/Pictures/Photos/rock2011.jpg </image>
<image2 source="///C:/Users/mj/Pictures/Photos/yeu1.jpg">
Vieux chateau fort de l'Ile d'Yeu démoli sous Louis XIV
</image2>
<image3 alt="Gaetan Roussel La Cigale 2011">
///C:/Users/mj/Pictures/Photos/rock2011.jpg
</image3>
</list>
<comment>Commentaires écrit par MJB</comment>
</page>
```



# Images.xsl –règles racine et règle titre

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="html"/>
<xsl:template match="/">
  <html>
  <head>
    <title><xsl:value-of select="page/title"/>
  </title>
  </head>
  <body><xsl:apply-templates select="page/list" />
  </body>
  </html>
</xsl:template>
<xsl:template match="title">
<h1 align="center">
  <xsl:value-of select="."/>
</h1>
</xsl:template>
```

**Attention**  
Par rapport à la racine  
Le chemin doit référencer  
l'élément page



# Images.xsl –règles pour traiter les images

```
<xsl:template select="list">  
  <xsl:apply-templates select="image"/>  
  <xsl:apply-templates select="image2"/>  
  <xsl:apply-templates select="image3"/>  
</xsl:template>
```

## Element image

Le nom du fichier est le contenu de l'élément image

### Règle

```
<xsl:template match="image">  
  <p>  </p>  
</xsl:template>
```

### Sortie html

```
<p>  
  
</p>
```

## Element image2

le nom du fichier est dans l'attribut source

### Règle

```
<xsl:template match="image2">  
<p> </p>  
<p> <xsl:value-of select="."/> </p>  
</xsl:template>
```

### Sortie html

```
<p>  
  
</p>
```

## Element image3

le nom du fichier est le contenu de l'élément  
le texte alternatif est utilisé dans un paragraphe

### Règle

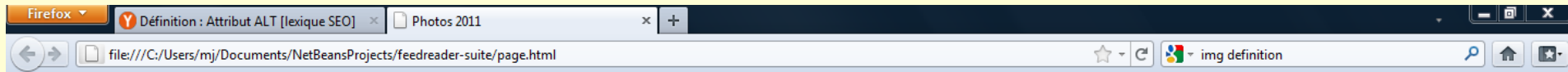
```
<xsl:template match="image3">  
  <p>  </p>  
  <p><xsl:value-of select="string(@alt)"/> </p>  
</xsl:template>
```

### Sortie html

```
<p>  
  
</p>  
<p>Gaetan Roussel La Cigale 2011</p>
```

```
<html>
<head> <META http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Photos 2011</title>
</head>
<body>
<p> </p>
<p> </p>
<p> Vieux chateau fort de l'Ile d'Yeu d&eacute;moli sous Louis XIV</p>
<p>  </p>
<p>Gaetan Roussel La Cigale 2011</p>
</body>
</html>
```

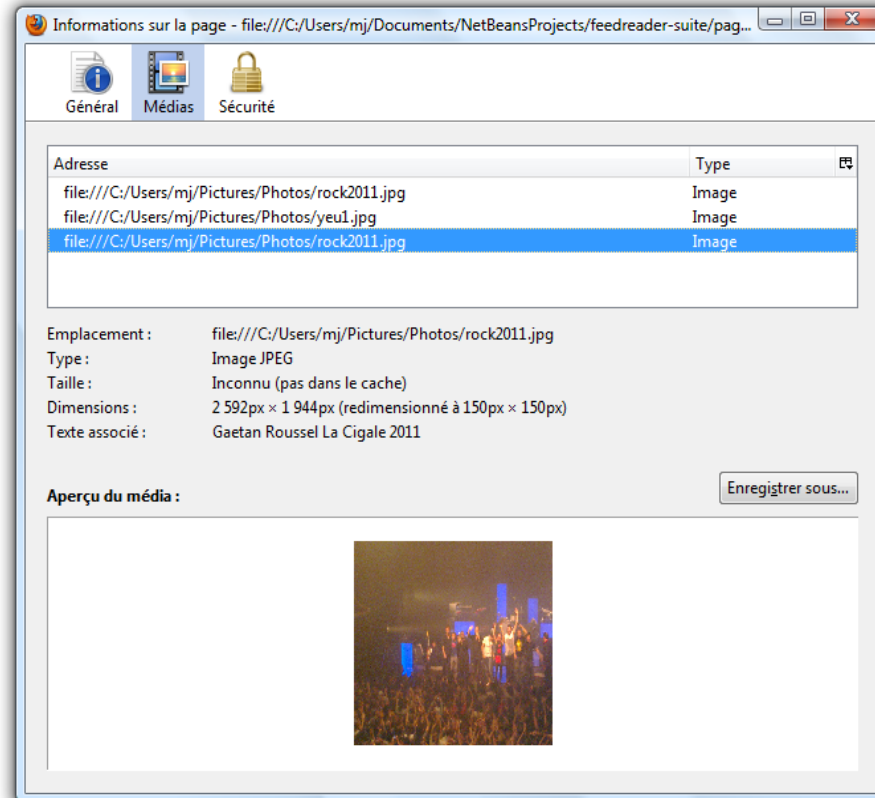
# Visualisation de page.xml



Vieux chateau fort de l'île d'Yeu démolì sous Louis XIV



Gaetan Roussel La Cigale 2011



## traiteImage.xsl : Traitement de l'attribut source de l'élément Image

Peut être soit une CDATA soit une entité

- Règle qui appelle le traitement d'une image avec en paramètre le premier attribut

```
<xsl:template match="image" priority="1">
  <xsl:message> traitement de l'élément image</xsl:message>
  <xsl:call-template name="imageSource">
    <xsl:with-param name="att1" select="./@*[position()=1]"/>
  </xsl:call-template>
</xsl:template>
```



# Traitement du premier attribut de l'élément image

**unparsed-entity-uri (string):** Fonction qui retourne URI d'une entité non interprétable

```
<xsl:template name="imageSource">
  <xsl:param name="att1"/>
  <xsl:message> traitement de l'attribut de l'image</xsl:message>
  <xsl:choose >
    <xsl:when test="unparsed-entity-uri($att1)">
      <xsl:message> traitement d'entite (ENTITY) </xsl:message>
      <xsl:call-template name="imageFile">
        <xsl:with-param name="file" select="unparsed-entity-uri($att1)"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:message> traitement du nom du fichier (CDATA)</xsl:message>
      <xsl:call-template name="imageFile">
        <xsl:with-param name="file" select="$att1"/>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

## Traitement du fichier avec largeur et hauteur prédéfinie

```
<xsl:template name="imageFile">
  <xsl:param name="file"/>
  <xsl:message> nom fichier :
    <xsl:value-of select="$file"/>
  </xsl:message>
  <div>
    
  </div>
</xsl:template>
```

# xsl:include

Permet d'inclure un traitement xsl défini dans un fichier externe

- AlbumImage.xsl

```
<xsl:include href="traiteImage.xsl"/>
```

```
<xsl:template match="/">
```

```
  <html>
```

```
    <head>
```

```
      <title>
```

```
        <xsl:value-of select='/node()/titre/text()'/>
```

```
      </title>
```

```
    </head>
```

```
    <body>
```

```
      <table width="400">
```

```
        <td align="justify">
```

```
          <xsl:message> traitement de l'élément album</xsl:message>
```

```
          <xsl:apply-templates select="*" />
```

```
        </td>
```

```
      </table>
```

```
    </body>
```

```
  </html>
```

```
</xsl:template>
```

# albumImage.xml - Sortie du processeur XSL

traitement de l'élément album

traitement de l'élément image

traitement de l'attribut de l'image

traitement d'entite (ENTITY)

nom fichier :

file:///C:/Users/mj/Pictures/Photos/Nepal\_619.jpg

traitement de l'élément image

traitement de l'attribut de l'image

traitement d'entite (ENTITY)

nom fichier :

file:///C:/Users/mj/Pictures/Photos/KTM\_396.jpg

traitement de l'élément image

traitement de l'attribut de l'image

traitement d'entite (ENTITY)

nom fichier :

file:///C:/Users/mj/Pictures/Photos/Nepal\_607.jpg

traitement de l'élément image

traitement de l'attribut de l'image

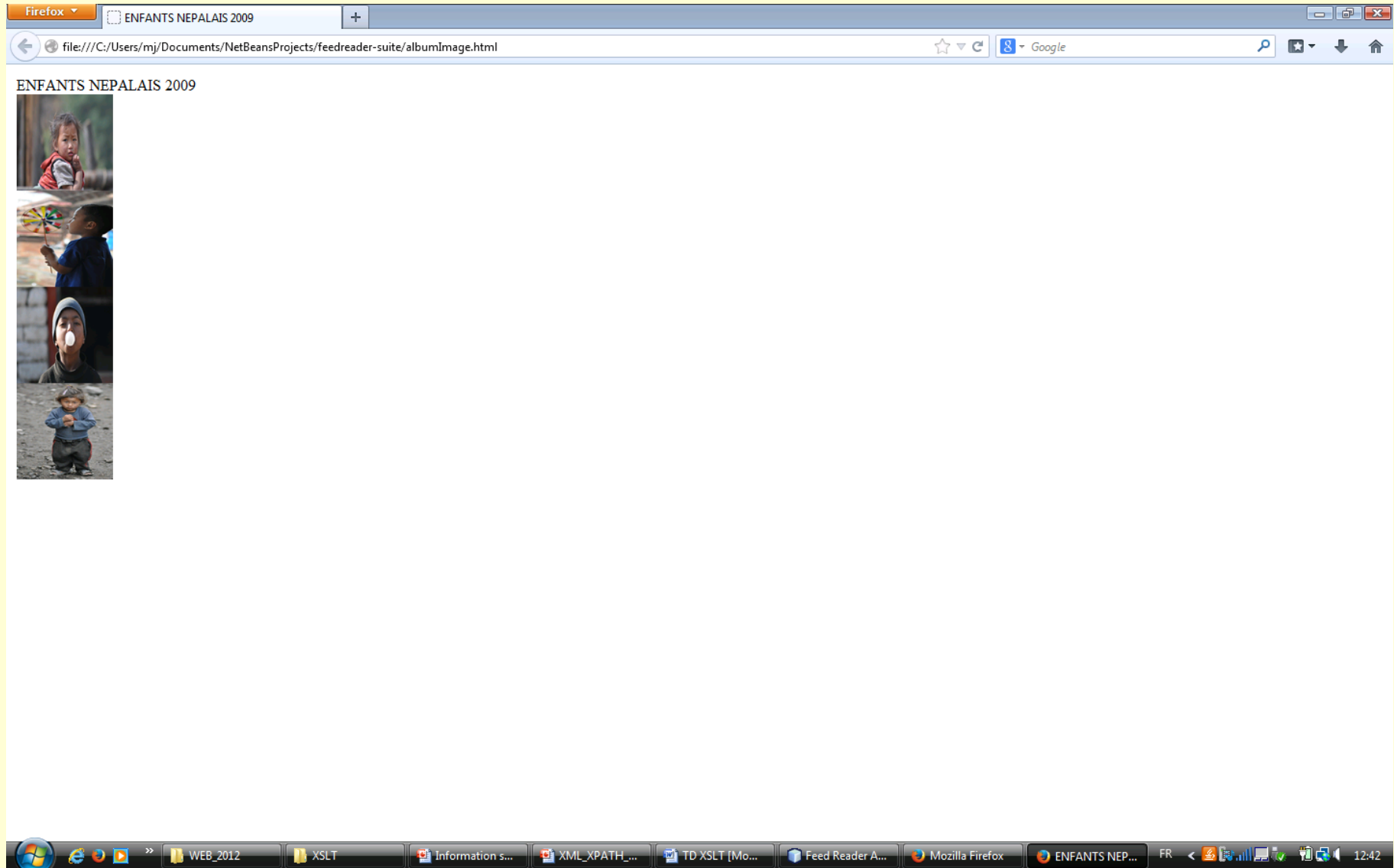
traitement d'entite (ENTITY)

nom fichier :

file:///C:/Users/mj/Pictures/Photos/Nepal\_622.jpg

Transformation finished.

# Visualisation



### Eléments conduisant à une insertion dans le document résultat

- xsl:attribute**      Insère un attribut dans un élément du document résultat
- xsl:element**      Insère un nœud *element* dans le document résultat
- xsl:text**          Insère un nœud *text* dans le document résultat
- xsl:number**        Numérote les nœuds du document résultat
- xsl:comment**      Insère un nœud *commentaire* dans le document résultat

### Eléments de contrôle

- xsl:fallback**      Règle déclenchée si le processeur ne reconnaît pas une instruction
- xsl:if**            Effectue un branchement conditionnel
- xsl:variable**      Définit une variable
- xsl:apply-imports**      Applique une règle importée tout en la complétant par un nouveau corps