

# TEI XSLT Stylesheets for Transformation

James Cummings

January 2010



## Using TEI XSL

This is a family of XSL stylesheets which are designed to render **simple** TEI documents. For the purpose of the TEI Consortium, they

- Implement the processing of ODD files behind Roma to make schemas and documentation
- ... and thus generate the TEI Guidelines in HTML
- ... and transform the TEI Guidelines to LaTeX for typesetting
- Render TEI Lite documents to
  1. HTML
  2. XHTML
  3. XSL FO (formatting objects, for page makeup)
  4. LaTeX (for typesetting)

The stylesheets have internal documentation, using P&P Software's XSLTdoc system

## Limitations

These stylesheets only do what were designed to do!

- They do not provide a rendering of all TEI elements
- They do not implement all possible values of every *@rend* attribute
- The different output formats are not always in sync, or give the same result

but they do deal with quite a few common problems.

## Related stylesheets

We also maintain in XSLT:

- A simple Docbook to TEI conversion
- Conversions to and from OpenOffice XML
- Conversions to and from Word 2007 XML
- Conversion from TEI P4 to TEI P5

## Output assumptions

The stylesheets attempt to work in the same way with each of the three supported output formats, but note:

- The HTML output is designed to work with an associated CSS stylesheet, which takes care of much of the detailed spacing and font work; however, the HTML is in charge of features such as the numbering of sections.
- The LaTeX output is designed for people who understand how to use existing LaTeX packages and classes; it therefore tries to produce reasonably readable TeX markup, with high-level commands whose effects will be determined by LaTeX (including numbering and spacing).
- The XSL FO output produces a very detailed specification of the output layout, with all the details of fonts, numbering, vertical and horizontal spacing specified *in situ*. The FO processor is only responsible for line and page breaking, and hyphenation.


## Many parameters

There are dozens and dozens of parameters which affect the stylesheet output; you can set values for these by

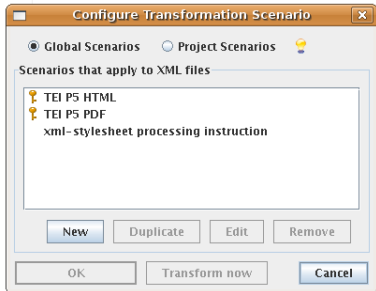
- specifying parameter names and values directly in oXygen
- setting them on a command line
- constructing a small local stylesheet which imports the public one, and adds overrides

## Invoking an XSLT transform from oXygen



When you have loaded an XML file, look for the  symbol in the menu and press it.

The first time, it will ask you which transformation scenario to use:



# Simple result

A TEI Project

## Punch, or the London Charivari, Vol. 147, July 1, 1914

---

### Table of contents


1. [PROGRESS.](#)
2. [THE ENCHANTED CASTLE.](#)
3. [Correspondence.](#)
6. [EGYPT IN VENICE."La Légende de Joseph2](#)
8. [ENIGMA.](#)
10. [A SCANDALMONGRIAN ROMANCE.\(By Francis Scribble.\)](#)
12. [CHARIVARIA.](#)
13. [THE COLLECTORS.](#)
14. [KINDNESS TO SUBJECTS.](#)
16. [THE WALKERS.](#)
17. [King Peter of Servia.](#)
19. [ESSENCE OF PARLIAMENT.\(Extracted from the Diary of Toby, M.P.\)](#)
22. [BUS-TALKING.](#)

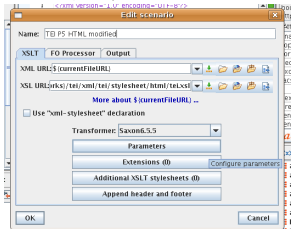
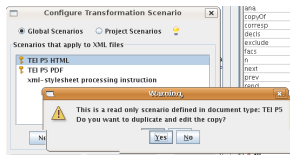
Find: feed Previous Next Highlight all Match case



# Configuring the scenario in oXygen

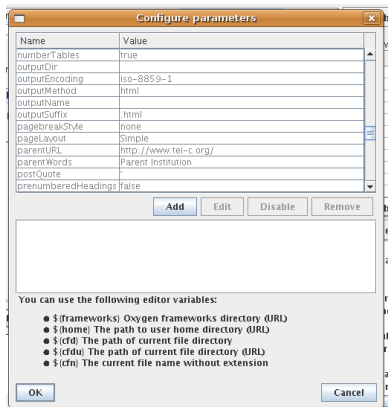


Look for the  symbol. This produces asking if you want to change the setup. Choose yes, and you see



# Changing parameters in oXygen

Now you can supply values for parameters:



# Change pageLayout

The screenshot shows a 'Configure parameters' dialog box with a table of parameters. The 'pageLayout' parameter is selected. An 'Edit' dialog box is open over it, showing the 'Name' as 'pageLayout' and the 'Value' as 'CSS'. Below the table are buttons for 'Add', 'Edit', 'Disable', and 'Remove'. At the bottom, there is a section for 'generated pages' with a text description and a scrollable area.

Name	Value
numberTables	true
outputDir	
outputEncoding	iso-8859-1
outputMethod	html
outputName	
outputSuffix	.html
pagebreakStyle	none
pageLayout	Simple
parentURL	http://www.tei-c.org/
parentWords	Parent Institution
postQuote	'
prenumberedHeadings	false

generated pages:  
The choice is between  
SimpleA linear presentation is created  
CSSThe page is created as a series of  
nested  
<div>s which can be arranged using CSS into a multicolumn layout  
TableThe  
page is created as an HTML table  
Default value: Simple

## 2 column display

[Skip links](#)

[Home](#)

[Parent Institution](#)

1. PROGRESS.
2. THE ENCHANTED CASTLE.
3. Correspondence.
6. EGYPT IN VENICE."La Légende de Joseph2
8. ENIGMA.
10. A SCANDALMONGRIAN ROMANCE.(By Francis Scribble.)
12. CHARIVARIA.
13. THE COLLECTORS.
14. KINDNESS TO SUBJECTS.
16. THE WALKERS.



## Areas of customization

- Standard page features
- Layout
- Headings
- Numbering
- Output
- Table of contents generation
- Internationalization
- CSS
- Tables
- Figures and graphics
- Inline Style

Remember that in HTML a lot will be done with CSS and Java



## Understanding the customization

There are six levels of interaction with the stylesheet family:

1. setting parameters
2. overriding templates provided for this purposed (listed in customization guide)
3. writing templates which implement the empty 'hooks' (listed in the customization guide)
4. adding new templates for elements not covered by the family
5. providing complete replacements for low-level templates

**Always** make changes by overriding — never hack the originals!

# Changing things around a bit

## Punch, or the London Charivari, Vol. 147, July 1, 1914

[Skip links](#)

[Punch central](#)

[The Land of Charivari](#)

PROGRESS.

THE ENCHANTED  
CASTLE.

Correspondence.

EGYPT IN VENICE."La  
Légende de Joseph2

ENIGMA.

A SCANDALMONGRIAN  
ROMANCE.(By  
Francis Scribble.)

CHARIVARIA.

THE COLLECTORS.

KINDNESS TO  
SUBJECTS.

THE WALKERS.

King Peter of Servia.

ESSENCE OF



## On the command line

You might turn test.xml into test.html by typing

```
xsltproc -o test.html  
  /usr/share/xml/tei/stylesheet/xhtml2/tei.xsl  
  test.xml
```

or

```
saxon -o test.html  
  test.xml  
  /usr/share/xml/tei/stylesheet/xhtml2/tei.xsl
```

and then change the result by passing a parameter to specify which CSS file to use:

```
saxon  
  -o test.html  
  test.xml  
  cssFile=http://localhost/mytei.css  
  numberheadings=false  
  topNavigationPanel=true  
  /usr/share/xml/tei/stylesheet/xhtml2/tei.xsl
```



## Using the a wrapper stylesheet

The simplest example of making a wrapper for the HTML stylesheets is:

```
<xsl:stylesheet version="2.0">  
  <xsl:include  
    href="http://www.tei-  
c.org/release/xml/tei/stylesheet/latex2/tei.xsl"/>  
</xsl:stylesheet>
```

## Using the a wrapper stylesheet (2)

Now you can build on it:

```
<xsl:stylesheet version="2.0">
  <xsl:include
    href="http://www.tei-
c.org/release/xml/tei/stylesheet/latex2/tei.xsl"/>
  <xsl:param name="logoFile">../.. /logo.png</xsl:param>
  <xsl:param name="logoWidth">60</xsl:param>
  <xsl:param name="logoHeight">60</xsl:param>
  <xsl:param name="cssFile">myTEI.css</xsl:param>
  <xsl:param name="pageLayout">CSS</xsl:param>
  <xsl:param name="outputMethod">xml</xsl:param>
  <xsl:param name="parentWords">The Punch Project</xsl:param>
  <xsl:param name="institution">The University of Punch</xsl:param>
</xsl:stylesheet>
```

## Using the a wrapper stylesheet (3)

And start to add your own templates:

```
<xsl:stylesheet version="2.0">
  <xsl:include
    href="http://www.tei-
c. org/release/xml/tei/stylesheet/latex2/tei. xsl"/>
  <xsl: param name="logoFile">../.. /logo. png</xsl: param>
  <xsl: param name="logoWidth">60</xsl: param>
  <xsl: param name="logoHeight">60</xsl: param>
  <xsl: param name="cssFile">myTEI. css</xsl: param>
  <xsl: param name="pageLayout">CSS</xsl: param>
  <xsl: param name="outputMethod">xml</xsl: param>
  <xsl: param name="parentWords">The Punch Project</xsl: param>
  <xsl: param name="parentURL">http://tei. oucs. ox. ac. uk/Punch/</xsl: param>
  <xsl: param name="institution">The University of Punch</xsl: param>
  <xsl: template match="tei:hi[@end='upside-down']">
    <span class="upside-down">
      <xsl: apply- templates/>
    </span>
  </xsl: template>
</xsl: stylesheet>
```

## Organisation of stylesheet files

The stylesheets for XSLT 2.0 are divided into four directories:

`common2` templates which are independent of output type

`fo2` templates for making XSL FO output

`xhtml2` templates for making HTML output

`latex2` templates for making LaTeX output

Within each directory there is a separate file for the templates which implement each of the TEI modules (eg `textstructure.xsl`, `linking.xsl`, or `drama.xsl`); these are included by a master file `tei.xsl`. This also includes a parameterization layer in the file `tei-param.xsl`, and the parameterization file from the `common` directory. The `tei.xsl` does any necessary declaration of constants and XSL keys.

## Using the XSL FO stylesheets

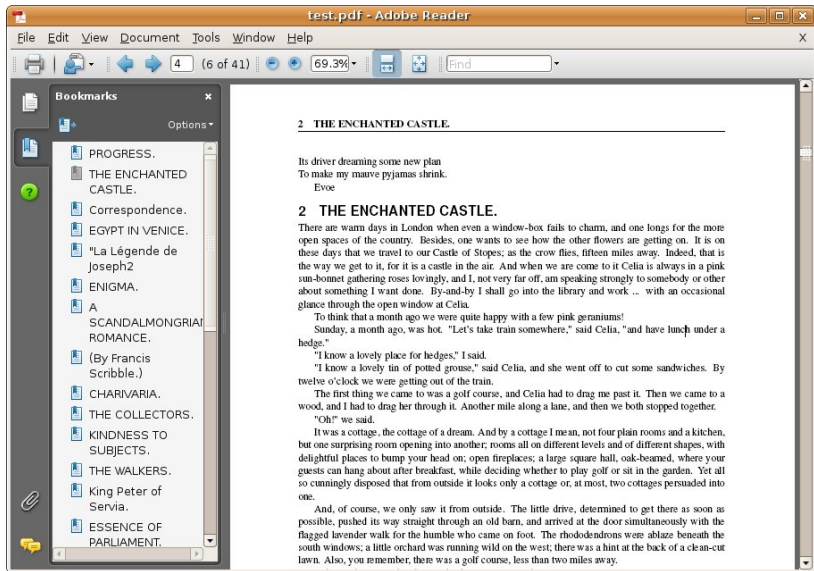
Each of the available XSL FO engines has some extensions to the Recommendation, and some have limitations; the stylesheets therefore have conditional sections to cater for this. The parameter `foEngine` can be set to one of the following values

- *antenna* (Antenna House processor)
- *fop* (Apache FOP)
- *passivetex* (TeX-based PassiveTeX)
- *xep* (RenderX XEP)

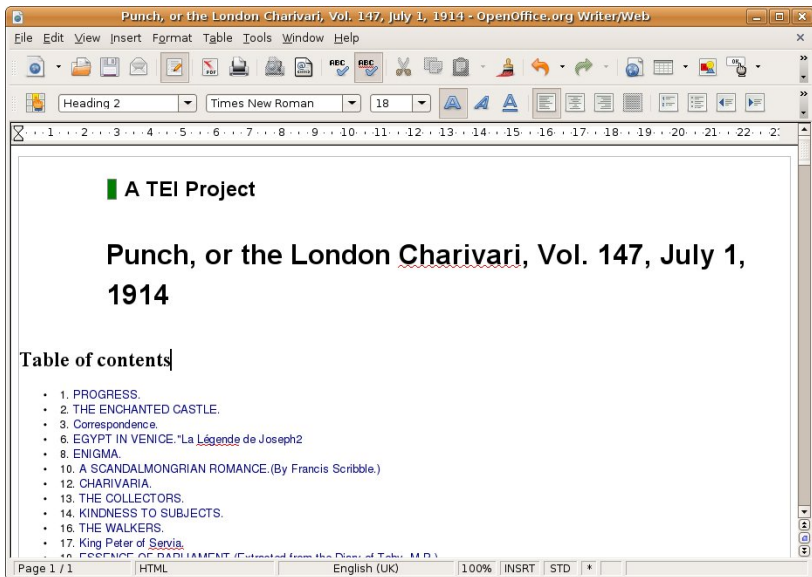
## Recommendations

- Use XSLT to make web pages. Easy.
- If you have LaTeX expertise, convert XML to LaTeX and prepare typeset pages that way
- If you have a workflow which can accept XSL FO, use it
- If you need to get Word files:
  - create HTML and load it into Word or OpenOffice
  - convert the TEI XML into OpenOffice using the TEIOO filters

# PDF generated from LaTeX



# Loading HTML into Open Office



The screenshot shows the OpenOffice Writer interface with a document titled "Punch, or the London Charivari, Vol. 147, July 1, 1914 - OpenOffice.org Writer/Web". The document content is as follows:

**A TEI Project**

**Punch, or the London Charivari, Vol. 147, July 1, 1914**

**Table of contents**

- 1. PROGRESS.
- 2. THE ENCHANTED CASTLE.
- 3. Correspondence.
- 6. EGYPT IN VENICE."La [Légende](#) de Joseph2
- 8. ENIGMA.
- 10. A SCANDALMONGRIAN ROMANCE.(By Francis Scribble.)
- 12. CHARIVARIA.
- 13. THE COLLECTORS.
- 14. KINDNESS TO SUBJECTS.
- 16. THE WALKERS.
- 17. King Peter of [Servia](#).
- 18. ESSENCE OF PARLIAMENT (Extracted from the Diary of Taku, M.P.)

The status bar at the bottom indicates "Page 1 / 1", "HTML", "English (UK)", "100%", "INSRT", "STD", and "\*" icons.