Exercise: Searching with XPath in oXygen

TEI@Oxford *2008-07*

1 Introduction

Now that you are experts in XPath, it is time to practice using it! One of the useful features in oXygen is that you are able to use XPath to search for nodes in your current document. This can be quite helpful if you want to find instances of a certain element in a very large document.

For now, open the file sample-1.xml which you saved at the end of the last exercise[1] : you should now have a version of this in which names of people and places have been marked up. Make sure that it is valid. (Check for the happy green square in the upper right-hand corner!)

2 Working with XPath

You can explore this Punch file (or other XML files of your choice) by typing XPath expressions into the box at the top left:
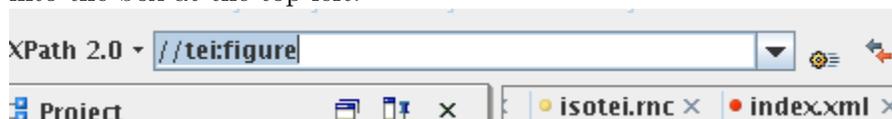
.
If you use XPath 2.0 (the default), you can put in expressions such as `//figure` without a namespace prefix. If you use XPath 1.0, you have to first set up the namespace by pressing the small icon to the right of the XPath box:
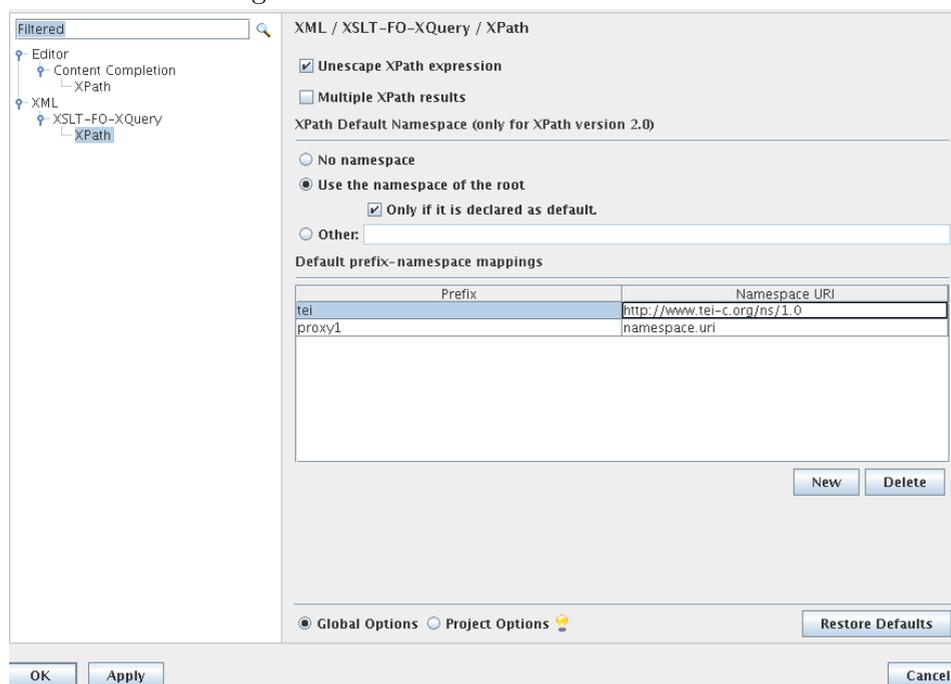

and linking the prefix `tei` with the namespace `http://www.tei-c.org/ns/1.0`. It is probably easier for this exercise to stay with XPath 2.0 and use unprefixed element names.

Type `//figure` into the box, press return once or twice, and select one of the results listed in the window at the bottom. You should see something like this:

---

[1]If you've mislaid or corrupted your copy of this file, make a copy of the file sample.xml in your teidata folder instead.
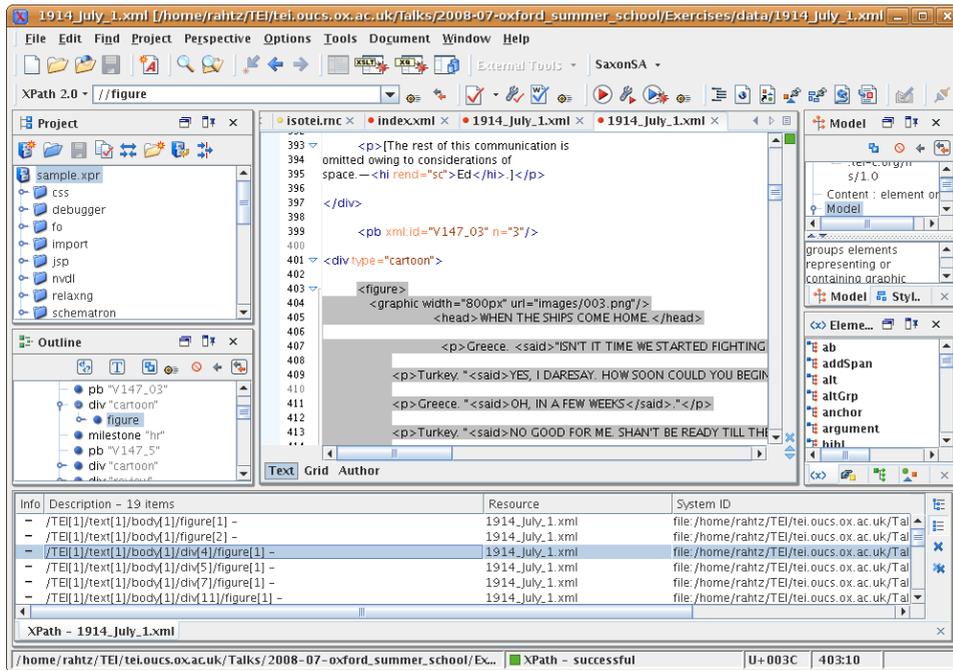
Notice that when you click on one of the results you are taken to that result with the editor window moving to that location and highlighting the element you have matched.

Notice as well that you can dismiss the results window by clicking on the little X in the bottom right-hand corner.

Try to answer the following questions:

1. How many `head` elements are there that are inside `figure` elements?

2. If you didn't remove the **name** elements in the previous exericse, how many do **not** have a rend attribute?

3. How many foreign phrases are there (marked as `<foreign>`) and what is the content of their @xml:lang attribute?

4. How many `<div>` elements are there in total? There are two ways to do this: if you search for `<div>` elements oXygen will tell you how many items are returned. Another method if you wanted to just get the number would be to use the `count()` function. Try each of these!

5. How many `<div>` elements contain more than 5 paragraphs?

6. And if using the file you worked on in the last exercise then search for how many `<persName>` elements you added. How `<person>` elements did you add? Can you find any persNames that do not (hint: not()) have a @ref attribute? (Add one as a test if you have none!) How many persName/@ref attributes `starts-with()` a value of '#'? How many start with a value of 'http'?

7. You can also find the fifth instance of something by filtering with a `[5]`. Who is the fifth `<persName>`?

8. You can also look back up the tree. For example try to find any paragraphs in the document whose `ancestor::div` has a @type attribute whose value is 'cartoon'?

9. In some instances your can use XPaths which find the same nodes but do so by a different route. For example, you can test things part way through an XPath or wait until the end

and check with `parent::` or `ancestor::` that they have the right values. Try a few more searches of your own invention!

3   Finding and Replacing with XPath

To assist you in finding and replacing bits of text, oXygen provides a fully featured 'Find/Replace' accessible from the 'Find' menu, the icon that looks like a magnifying glass, or its shortcut key (usually control-f). Open up this Find/Replace window now and try searching for a bit of text. Notice the various options available to you, including the ability to include/exclude the text of tags, case sensitivity, wrapping around, and regular expressions.

XPath searching is also enabled using the separate entry box on this window. If you put something like `//persName[@ref]` in this box, then it will limit your search to inside `<persName>` elements that have a @ref attribute.

Put `//hi[@rend='it']` in the XPath box in this window, find the word 'Business' and replace it with 'Pretending to work'. Notice that as you replace this word you only are doing so inside `<hi>` elements with a @rend attribute of 'it'.

You may wish to either Undo that last change, or repeat it with the values reversed.

Experiment with finding and replacing other values such as attributes (hint: make sure `Search in tags` is checked).