

# Practical Exercise 2: Making a Valid TEI Document

James Cummings

19 September 2013

## Exercise 2: Making a Valid TEI Document

### 1 Learning Outcomes

When you successfully complete the first part of this exercise you should be able to:

- discern the elements and attributes needed for a minimum valid TEI XML file
- associate a TEI XML file with a schema
- use the TEI namespace
- create a minimal TEI header and text body
- check for both validity and well-formedness

### 2 Summary

This part of this exercise will walk you through creating a TEI XML file and inserting the work you did previously into it. You'll learn about the required aspects of the `<teiHeader>` and the basic structure of a TEI file. We will also add some phrase-level markup if we have time.

### 3 Starting A New XML File

Let's start a new XML file by following each of these steps:

1. Load up the oXygen XML Editor if it isn't already loaded by using the Start Menu (in Windows), or double-clicking the icon on the desktop.
2. Once the editor has fully loaded from the **File** menu select **New** and expand the **New Document** section so you can select **XML Document**. Doing this should open up a blank document with an XML Declaration added.

### 4 Inserting a `<TEI>` Element

All TEI files start either with a `<TEI>` element or a `<teiCorpus>` element. In most cases you'll want a `<TEI>` element. These elements have a special pseudo-attribute called 'xmlns' that indicates the namespace a set of elements are from. This is inherited by any elements inside it (unless overridden). This is how we can be sure we're talking about, say, a `<title>` element from the TEI rather than any other schema.

1. Add a `<TEI>` element and then add it to the TEI namespace (<http://www.tei-c.org/ns/1.0>). Add a few blank lines between the starting and closing tag. Your file should look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">

</TEI>
```

2. Notice what happens in oXygen and how it helps you input this. Also notice that your file may now have an angry red square rather than a happy green one! Is your file well-formed? (yes, it is!) Why is this red then?
3. If it is red it is because your version of oXygen is prepackaged with all sorts of TEI goodness, and in this case it recognises that files starting with `<TEI>` in the TEI namespace are to be associated automatically with a TEI schema that it has stored. It is complaining that you do not have a `<teiHeader>` in your file because all valid TEI files must have this.

### 5 Adding a `<teiHeader>`

Inside the `<TEI>` element we need to add a `<teiHeader>` element.

1. Put the cursor between the starting and closing `<TEI>` element and type in a `<teiHeader>` element. Notice that oXygen provides the closing `</teiHeader>` element. If the correct option is set in oXygen, it understands the TEI schema and knows that certain content is required inside a `<teiHeader>`. It can automatically provide that markup. If not, you'll have to type it in! The resulting file at this point should look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
<teiHeader>
<fileDesc>
<titleStmt>
<title></title>
</titleStmt>
<publicationStmt/>
<sourceDesc/>
</fileDesc>
</teiHeader></TEI>
```

2. Notice that your file still has an angry red square rather than a happy green square. This is because there are some required elements missing, even though you've added some markup.
3. First, add a title of something like "Letter to Leslie Gunston" by adding this text between the starting and closing `<title>` tags. There are other elements which are allowed here in `<titleStmt>` such as `<author>` (Wilfred Owen) that you could add but aren't really required for this exercise. You could use the more general `<respStmt>` (with a `<name>` element with your name and a `<resp>` element with something like 'TEI P5 Encoding' in it) to record your own work if you wish, but this also isn't necessary for this exercise.
4. Then add a paragraph `<p>` inside the `<publicationStmt>` with some text to record what this file is for, perhaps something like "An exercise for learning TEI."
5. Inside `sourceDesc` we should add a `<p>` with some text like: "This file has been made from a digital photograph of a Letter to Leslie Gunston" To make this even better, we might surround the title 'Letter to Leslie Gunston' with a `<ref>` element with a `@target` attribute with a value of 'http://www.oucs.ox.ac.uk/ww1lit/collections/document/5243/4769' because that is URL from which we got this text.
6. Your file should now look something like:

```

<TEI xmlns="http://www.tei-c.org/ns/1.0">
<teiHeader>
<fileDesc>
<titleStmt>
<title>Letter to Leslie Gunston</title>
</titleStmt>
<publicationStmt>
<p>An exercise for learning TEI.</p>
</publicationStmt>
<sourceDesc>
<p>This file has been made from a digital photograph of a <ref
target="http://www.oucs.ox.ac.uk/wllit/collections/document/5243/4769">Letter
to Leslie Gunston</ref>
</p>
</sourceDesc>
</fileDesc>
</teiHeader></TEI>

```

7. Notice that even though this is a complete `<teiHeader>` with all the required aspects, our file as a whole isn't valid yet.

## 6 Add a `<text>`

All TEI files, in addition to a `<teiHeader>` with `<fileDesc>` containing a `<titleStmt>`, `<publicationStmt>`, and `<sourceDesc>`, need to follow the header with at least one of: `<sourceDoc>`, `<facsimile>`, or `<text>`. In our case we're going to add the `<text>` element by putting in the file we made earlier. To do this:

1. Add a couple of blank lines after the closing `</teiHeader>` just for some space.
2. With the cursor in between the opening and closing `<body>` tags go to the **Document** menu on the menu bar, and select **File**, and **Insert File**. **Note that this is from the 'Document' menu!** Select the file you saved earlier if you finished the first exercise. If you didn't then there is a file called 'exercise1-finished.xml' in the 'spoilers' directory which has the fully completed first exercise.
3. As soon as you've added this oXygen will still think our file is invalid because we have got that XML declaration before the `<text>` element. Delete this redundant XML declaration!
4. Your document should now be valid and have a happy green square in the upper right-hand corner! If it isn't, try to solve the problem by looking at the error message that is provided.

## 7 Associating a Japanese `tei_all` Schema

One of the great things about the TEI is that you are able to get the descriptions of elements in a variety of languages. This is powered not by something like Google, but volunteers, so sometimes takes awhile to get things updated. (If you want to volunteer, email [tei@it.ox.ac.uk](mailto:tei@it.ox.ac.uk)). We've been using the default TEI framework which comes with oXygen and works on documents that start with the `<TEI>` element in the TEI Namespace. Let's associate our document with a Japanese version of `tei_all` instead!

1. With your document loaded you can use the **Document** menu and under that the **Schema** sub-menu and **Associate Schema**. (Or you can use the icon, that looks like a push pin with a red square and blue triangle.)

2. Click on the file folder and navigate to the materials folder and select the file 'tei\_all-ja.rnc'.
3. This should put something at the top of the file which looks like:

```
<?xml-model href="../../../tei_all-ja.rnc" type="application/relax-ng-compact-syntax"?>
```

## 8 Improving Our Markup a bit

### 8.1 A Better <opener>

The <opener> element at the start of letter could be a bit better.

1. Using the techniques you already know, wrap a <dateline> element around the text starting 'Craiglockhart' and ending after 'Wednesday'
2. Add a @rend attribute to the opening tag of the <dateline> with a value of 'right' in order to indicate that it is aligned to the right.
3. Inside the <dateline> add a <date> element which covers all of 'July 1917. Wednesday' and add a @when attribute to this with a value of '1917-07' (which means July 1917, unfortunately we don't know which Wednesday!).
4. Wrap a <salute> around 'Dear L.'. This is a semantic element for containing opening or closing salutations.
5. Your <opener> should now look like:

```
<opener>  
  <dateline rend="right"> Craiglockhart. <lb/>  
    <date when="1917-07">July 1917.<lb/> Wednesday</date>  
  </dateline>  
  <salute>Dear L.</salute>  
</opener>
```

### 8.2 Adding <add> and <del>

1. The <add> element notes when something is an addition. This can be a bit of text added in immediately by the author when they realise the sentence doesn't make sense, or a later scribe. The <del> element is similar but marks a deletion.
2. There are a number of deletions in the first paragraph marked with hyphens immediately before and after the deleted word. These are:
  - my card -of- posted
  - remember: -my- you old
  - subject -is- has
  - Stimuli -shown- + Devices

Mark each one of them using <del> instead of a hyphen. So the last one above will become:

```
Stimuli <del>shown</del> + Devices
```

3. There are only two additions in the letter, both in the first paragraph. These are when he writes 'you [have] had my card' and 'when I am [one of the] founders'. The transcriber has indicated that the 'one of the' was added above by putting it in square brackets. Replace both of these instances with an `<add>` element
4. We can make our `<add>` elements even better by giving them a `@place` attribute with a value of 'above'. When you do this, note how oXygen prompts you with a list of suggested values.

### 8.3 Recording Some Underlining

1. Recording underlining does not have an element of its own, instead it is viewed as one of the many forms of highlighting. Thus, we use the `@rend` attribute when it coincides with another element, like `<emph>` when we think the purpose of the underlining was emphasis, or the general-purpose `<hi>` element for highlighting .
2. In the first paragraph where it says '(herewith) and I'm take German', this is likely Owen emphasising his exasperation. Mark this up as:

```
(herewith) <emph rend="ul">and</emph> I'm take German
```

3. Similarly in the next paragraph mark the underlined 'Antaeus v. Heracles' as

```
bit of Blank: on <hi rend="ul">Antaeus v. Heracles</hi>
```

This means we are not claiming to know why Owen underlined this bit. (He may have meant it as a title, but we are not sure.)

4. In the final paragraph of the last prose division, where it says 'be equally happy. You are lucky!', mark the 'are' up as an `<emph>` with a `@rend` attribute of 'ul'.

### 8.4 Improving Our `<closer>`

1. The `<closer>` at the end of the letter also has some underlining. In this case we should use a `<salute>` element to wrap around the 'Ever yours', but also add a `@rend` attribute with a value of 'ul'.
2. Similarly, the bit of text the transcriber has recorded as 'WEO' is in fact a single character formed of these three letters. It is Owen's way of using a monogram to sign his letters. There is an element `<signed>` which we can wrap around this and give a `@rend` attribute of 'monogram'.
3. Your `<closer>` should now look something like:

```
<closer>  
  <salute rend="ul">Ever yours</salute>  
  <signed rend="monogram">WEO</signed>  
</closer>
```

### 8.5 Marking Apparent `<sic>Erros</sic>`

There are a number of errors in the letter where Owen has made mistakes such as only partly writing words or writing the wrong word.

1. In the first paragraph where it says 'you old Black Molt?', I think Owen meant to write 'your'. We could mark this up with `<choice>` to give the error and the correction, but instead let's just wrap a `<sic>` to show that this is an error.
2. In the same paragraph where it says 'I'm take German Lessons' Owen probably meant 'taking'. Mark the 'take' with a `<sic>` as well.

## 9 Saving Your Work

Let's save our work:

- Have you formatted and indented your work automatically?
- Is your work well-formed? Do you have a happy green square or an angry red one?
- From the 'File' menu select 'Save' or click on the Save icon (looks like an old-style 3.5" disk)
- Save the file using the name 'exercise02.xml' or another name of your choice.

## 10 Self-Assessment

Check that you understand some of the core principles of this part of the exercise by answering the following questions:

- Which elements and attributes do you need for a minimum valid TEI XML document?
- What three parts of the `<teiHeader>` are required in all TEI conformant documents?
- Where are these elements and attributes allowed?
- What is the function of each element and attribute you've used? (If you are unsure, look them up!)
- Why do you think these elements and attributes are required in TEI XML?

## 11 Further Reading

These exercises should have given you some experience editing XML and making a valid TEI file.

- If you are finished early you may wish to browse through the TEI Guidelines online at <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/index-toc.html>.
- In particular you might want to look at the Elements appendix of reference pages for individual elements. Consider looking up all the elements you've used in this file to see how they are defined.
- What other elements are allowed inside the `<text>` element? What would you use them for?
- You may wish to read the chapters on Default Text Structure <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/DS.html> or Elements Available to All TEI Documents <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/CO.html>.