

Exercise 2: A TEI File with a Better `teiHeader` (two parts)

TEI @ Oxford

23 May 2013

1 Exercise 2a: Create a TEI Document

1.1 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
- have used the TEI namespace
- create a minimum TEI header and text body
- check for both validity and well-formedness

1.2 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.

1.3 Start a New XML File

Follow the same steps you did for the first exercise to start a new blank XML file. Although we could start a file with a TEI P5 template, for this particular exercise that would be cheating!

- Load up the oXygen XML Editor if it isn't already loaded by using the Windows Start Menu, or double-clicking the icon on the desktop.
- Once the editor has fully loaded from the 'File' menu select 'New' and under 'New Document' select 'XML Document'. This should open up a blank file with an XML Declaration added.
- An XML Declaration looks like:

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

1.4 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.

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

1 EXERCISE 2A: CREATE A TEI DOCUMENT

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

</TEI>
```

- 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?
- 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.

1.5 Adding a `<teiHeader>`

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

- 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 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>
```

- Notice that your file still has an angry red square rather than a happy green square. This is because there is still stuff needed even though you've added some markup. First, add a title of something like "My 'Strange Meeting' document" 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), `<editor>` (Jon Stallworthy), 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 as with the other embellishments this isn't necessary for this exercise.
- 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."
- Inside `sourceDesc` we should add a `<p>` with some text like: "The primary resource of this file is Strange Meeting from Jon Stallworthy's edition, available on the First World War Poetry Digital Archive." To make this even better, we might surround the

title 'Strange Meeting' with a `<ref>` element with a `@target` attribute with a value of 'http://www.oucs.ox.ac.uk/ww1lit/collections/item/3350' because that is URL from which we got this text.

- Your `<teiHeader>` should now look something like:

```
<teiHeader>
  <fileDesc>
    <titleStmt>
      <title>My 'Strange Meeting' document</title>
    </titleStmt>
    <publicationStmt>
      <p>An exercise for learning TEI.</p>
    </publicationStmt>
    <sourceDesc>
      <p> The primary resource of this file is <ref
        target="http://www.oucs.ox.ac.uk/ww1lit/collections/item/3350">Strange
        Meeting</ref> from Jon Stallworthy's edition, available on the First
World
        War Poetry Digital Archive. </p>
    </sourceDesc>
  </fileDesc>
</teiHeader>
```

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

1.6 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 a `<text>` element. To do this:

- Add a couple of blank lines after the closing `</teiHeader>`.
- Insert a `<text>` element and inside that a `<body>` element. (The `<text>` element requires a `<body>` element because if you don't have a text body, what are you encoding?)
- The `<text>` section of the file should look something like:

```
<text>
  <body>
    <!-- We will put our poem here -->
  </body>
</text>
```

1.7 Adding Our Poem

This is a good start but we need to put something inside the body. Luckily, we have already encoded a poem in the previous exercise, so we can use that!

- 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'. Select the file you saved earlier if you finished the first exercise. If you didn't then in the spoilers directory there is a file called 'ex01.xml' which has the completed first exercise.

2 EXERCISE 2B: IMPROVING A TEIHEADER

- But wait, as soon as you've added this we get a bit of a problem! oXygen will complain that we've got an XML declaration in the middle of our file. Delete this redundant XML declaration!
- 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.

1.8 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 'exercise02a.xml' or another name of your choice.

1.9 Self-Assessment

Check if 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?
- Why do you think these elements and attributes are required in TEI XML?

2 Exercise 2b: Improving a `<teiHeader>`

2.1 Learning Outcomes

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

- read through and analyse encoding in an existing TEI file.
- improve the structure and metadata of a `<teiHeader>`.
- understand the components of a `<fileDesc>` including:
 - `<titleStmt>` for title and intellectual responsibility.
 - `<publicationStmt>` for information about the publication and distribution of the electronic item.
 - `<sourceDesc>` to record metadata about the source document.
- use the `<encodingDesc>` to record the markup used in the file.
- use the `<profileDesc>` to record non-bibliographic aspects of the file.
- record major changes to the file in the `<revisionDesc>`.

2.2 Summary

This part of the exercise gives you a chance to read through a TEI XML file you have not encoded and understand its markup and structure. It walks you through improvements to various aspects of the `<teiHeader>` and how to record additional metadata about the electronic file and its sources.

2.3 Starting Up

In this case we're starting with a sample file that we have created for you. Load up the file called: 'letter-to-LG.xml' in the oXygen XML editor. Check that the file is well-formed and valid. Note the line near the top of the file:

```
<?xml-model href="http://www.tei-c.org/release/xml/tei/custom/schema/relaxng/tei_all.rng"
schematypens="http://relaxng.org/ns/structure/1.0"?>
```

This is what tells the oXygen editor that it should be validating this file with the `tei_all` schema.

2.4 Reading through the file

This file contains a letter from Wilfred Owen to Leslie Gunston. It talks about a forthcoming address to the Field Club, and contains a partial draft of 'The Wrestlers'. It was written in July 1917 from Craiglockhart War Hospital, Edinburgh, Scotland. Images of this letter are available in the materials we've provided.

- Note the very minimal `<teiHeader>`.
- Look at the structure of the document as three divisions and make sure you understand these divisions.
- Note the use of the `<dateline>` element.
- See how the encoder has recorded line-breaks in the prose.
- What other elements has the encoder included? Make sure you understand the meaning of them. If you are unsure of the meaning of them, look them up on the TEI-C website at: <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/REF-ELEMENTS.html>

2.5 Improving the `<titleStmt>`

As you can see the `<teiHeader>` is lacking a lot of information. Let's improve it!

- Inside the `<fileDesc>` the `<titleStmt>` contains only a `<title>`. What else can `<titleStmt>` contain? (hint: typing '<' here will provoke oXygen into providing a dropdown list of possibilities).
- Underneath the `<title>` add an `<author>` element. The content of this should be 'Wilfred Owen'.
- Below this add an `<editor>` element with the content of 'Renée van Baalen'. (She transcribed the letter for our teaching purposes.) How does one type in 'é' in oXygen? *Hint:* the 'Edit' menu contains a 'Insert from Character Map' entry.
- After this add a `<principal>` element to record the person primarily responsible for the project. In this case, use your own name.
- Below this add a `<meeting>` element with the content of 'Digital Humanities at Oxford Summer School 2012'.

2 EXERCISE 2B: IMPROVING A TEIHEADER

- After that add `<respStmt>` with a `<resp>` inside it saying 'Improved encoding' and a `<name>` with your name.
- Your `<titleStmt>` should now look something like:

```
<titleStmt>
  <title>Letter to Leslie Gunston</title>
  <author>Wilfred Owen</author>
  <editor>Renée van Baalen</editor>
  <principal>[your name here]</principal>
  <meeting>Introduction to TEI Course at the British Library</meeting>
  <respStmt>
    <resp>Improved encoding</resp>
    <name>[Your name here]</name>
  </respStmt>
</titleStmt>
```

If you do not understand what any of these elements are for, make sure to look them up on the TEI-C website at the URL given above.

2.6 Improving the `<publicationStmt>`

The `<publicationStmt>` is also fairly limited. It could contain a lot of structured information, but just has a paragraph of prose. Let's replace it!

- Delete the entire paragraph including the starting and ending `<p>` tags.
- Inside `<publicationStmt>` add a `<publisher>` element. In this case, 'TEI @ Oxford' is the publisher.
- Below the `<publisher>` add a `<distributor>` containing 'IT Services, University of Oxford'.
- After this add an `<authority>` element, to detail under who's authority it is published. In this case let's say it is under your authority, so add your name.
- Next, inside a `<pubPlace>` element, which itself contains an `<address>` element with an `<orgName>` (IT Services, University of Oxford), a `<street>` address (13 Banbury Road), a `<settlement>` (Oxford), a `<postCode>` (OX2 6NN), and a `<country>` (United Kingdom).
- After the `<pubPlace>` element but still inside the `<publicationStmt>` add a `<date>` element with content of '14 February 2013, Valentine's Day'. The `<date>` element can have a `@when` attribute to take a standardised YYYY-MM-DD form of the date, add this as well.
- Add an ID number after this using `<idno>`. This should be something like a catalogue number, or a URL at which this document will reside. In this case, make up what you think a sensible ID number would be for your edition of this letter.
- Next add an `<availability>` statement with a `<licence>` containing a description of the licence you would want to distribute this under with a `<p>` containing the prose inside. We recommend you choose a Creative Commons license using <http://creativecommons.org/choose/>. For bonus points you can include a link (using `<ref>` with a `@target` attribute inside your `<p>` to the license you chose and/or using the `@target` attribute on `<licence>` directly.

- Your `<publicationStmt>` should now look something like:

```
<publicationStmt>
  <publisher>TEI @ Oxford</publisher>
  <distributor>IT Services, University of Oxford</distributor>
  <authority>[Your name here]</authority>
  <pubPlace>
    <address>
      <orgName>IT Services, University of Oxford</orgName>
      <street>13 Banbury Road</street>
      <settlement>Oxford</settlement>
      <postCode>OX2 6NN</postCode>
      <country>United Kingdom</country>
    </address>
  </pubPlace>
  <date when="2013-02-14">14 February 2013, Valentine's Day</date>
  <idno>[Insert an ID number here]</idno>
  <availability>
    <licence
      target="http://creativecommons.org/licenses/by/3.0/">
      <p>Licensed with a <ref
        target="http://creativecommons.org/licenses/by/3.0/">Creative
        Commons Attribution</ref> licence.</p>
    </licence>
  </availability>
</publicationStmt>
```

2.7 Improving the `<sourceDesc>`

Our `<sourceDesc>` is also fairly limited.

- Delete the entire paragraph that is currently in the `<sourceDesc>` and replace it with a `<biblStruct>`.
- The `<biblStruct>` should have an `<analytic>` with a `<title>` (Letter to Leslie Gunston), and `<author>` (Wilfred Owen).
- The `<biblStruct>` should also have a `<monogr>` for the collection containing:
 - `<title>` (The Wilfred Owen Collection).
 - A `<ref>` (First World War Poetry Digital Archive) containing a `@target` attribute pointing to `'http://www.oucs.ox.ac.uk/ww1lit/collections/document/5243'`.
 - An `<imprint>` element containing a `<publisher>` (The First World War Poetry Digital Archive), a `<pubPlace>` (Oxford), and a `<bibScope>` (Two pages) with a `@type` attribute of `'pp'`, and a `@n` attribute of `'2'`.
 - Outside the `<monogr>` but inside the `<biblStruct>` add a `<relatedItem>` with a `<bibl>` containing `'The source of this digital resource is a copy from the Harry Ransom Centre.'` You could also wrap `'Harry Ransom Centre'` in a `<distributor>` element. This is an example of a much less structured bibliographic citation inside a structured one.
 - Your `<sourceDesc>` should now look something like:

```
<sourceDesc>
  <biblStruct>
    <analytic>
      <title>Letter to Leslie Gunston</title>
```

```
<author>Wilfred Owen</author>
</analytic>
<monogr>
  <title>The Wilfred Owen Collection</title>
  <ref
    target="http://www.oucs.ox.ac.uk/wllit/collections/document/5243/4769">
    First WorldWar Poetry Digital Archive</ref>
  <imprint>
    <publisher>The First World War Poetry Digital Archive</publisher>
    <pubPlace>Oxford</pubPlace>
    <biblScope type="pp" n="2">Two pages</biblScope>
  </imprint>
</monogr>
<relatedItem>
  <bibl>The source of this digital resource is a copy from the
    <distributor>Harry Ransom Centre</distributor>.</bibl>
</relatedItem>
</biblStruct>
</sourceDesc>
```

2.8 Other components of the <fileDesc>

There are other elements that could appear in your <fileDesc>.

- Immediately after the closing </fileDesc> tag you could add an <editionStmt> with an <edition> containing a descriptive phrase such as 'First Edition' for the current edition of the electronic file.
- Immediately after the closing </editionStmt> you could add an <extent> element with some measure of the size of the text (e.g. '260 words').
- Immediately after the closing </publicationStmt> you could add a <notesStmt> with one or more <note> elements inside it. One could contain something saying 'Transcribed for a TEI Workshop'.

2.9 Adding an <encodingDesc>

An <encodingDesc> element will give us a place to document the encoding practices in the document.

- After the closing </fileDesc> we should add an <encodingDesc> element.
- Inside the <encodingDesc> add a <projectDesc> with a <p> inside it saying something like 'The TEI@Oxford project created teaching materials for use in teaching TEI workshops'.
- Next inside the <encodingDesc> add an <editorialDecl> with a <correction> inside that with a paragraph saying something like 'Apparent errors have been marked as <sic> but correct readings not provided'. Mark up <sic> as an element by using <gi> (generic identifier).
- Also inside the <editorialDecl> add a <hyphenation> with a paragraph saying something like 'Hyphens have been transcribed as they appear'.
- Look at the other options available to you inside <editorialDecl> and <encodingDesc>.
- Your <encodingDesc> should look something like:

```

<encodingDesc>
  <projectDesc>
    <p>The TEI@Oxford project created teaching materials for use in teaching TEI
workshops.</p>
  </projectDesc>
  <editorialDecl>
    <correction>
      <p>Apparent errors have been marked as <gi>sic</gi> but correct readings not
provided.</p>
    </correction>
    <hyphenation>
      <p>Hyphens have been transcribed as they appear.</p>
    </hyphenation>
  </editorialDecl>
</encodingDesc>

```

2.10 Adding a <profileDesc>

A <profileDesc> is a place to store various non-bibliographic information concerning the text.

- After the closing </encodingDesc> add a <profileDesc>.
- Inside this add a <creation> with a <placeName> (Craiglockhart) and a <date> (July 1917) perhaps with a @when attribute ('1917-07').
- In the <profileDesc> next add a <handNotes> with a <handNote> inside it saying something like 'Written in Wilfred Owen's hand'.
- Next, add a <langUsage> inside the <profileDesc> with a <language> inside ('English') with an @ident attribute with a value of 'en' for the English language code.
- Next add a <textClass> with a <classCode> with content of '826' and a @scheme attribute of "http://www.oclc.org/dewey/resources/summaries/default.htm". This is the Dewey classification code for 'English Letters'. (Yes, there are much better systems to use – this is just an example!)
- Your <profileDesc> should now look something like:

```

<profileDesc>
  <creation>
    <placeName>Craiglockhart</placeName>
    <date when="1917-07">July 1917</date>
  </creation>
  <handNotes>
    <handNote>Written in Wilfred Owen's hand</handNote>
  </handNotes>
  <langUsage>
    <language ident="en">English</language>
  </langUsage>
  <textClass>
    <classCode
      scheme="http://www.oclc.org/dewey/resources/summaries/default.htm">826
    </classCode>
  </textClass>
</profileDesc>

```

2.11 Adding a <revisionDesc>

A <revisionDesc> gives you a way to record major stages in revision to a document.

- After the closing </profileDesc> add a <revisionDesc> element.
- Add two <change> elements. On the first one add a @when attribute with today's date. Inside the <change> add a <persName> containing your name, followed by the text 'improved the header'.
- In the second <change> add a @when attribute of '2012-02', with a <persName> of 'Renée van Baalen' saying that she 'transcribed the Letter to Leslie Gunston document'. You may also wish to mark 'Letter to Leslie Gunston' as a <title>.
- It is standard practice for the most recent <change> to be first.
- Your <revisionDesc> should now look something like:

```
<revisionDesc>
  <change when="2013-02-14">
    <persName>[Your name here]</persName> improved the header.</change>
  <change when="2012-07">
    <persName>Renée van Baalen</persName> transcribed the <title>Letter to
      Leslie Gunston</title> document. </change>
</revisionDesc>
```

2.12 Saving Your Work

Let's save our work:

- Is your work well-formed? Do you have a happy green square or an angry red one?
- Have you formatted and indented your work automatically?
- From the 'File' menu select 'Save' or click on the Save icon (looks like an old-style 3.5" disk).
- Or if you prefer use the 'File' then 'Save As' menu item to save the file using the name 'exercise02b.xml' or another name of your choice.

2.13 Self-Assessment

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

- What kinds of metadata can you store in a <titleStmt>?
- What is a <publicationStmt> used for? What can it contain?
- How do you provide details of the source for the file?
- What is the difference between <bibl> and <biblStruct>?
- What is an <encodingDesc> for?
- What order should <change> elements be listed in a <revisionDesc>?

3 Next and 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?
- What other parts of the `<teiHeader>` are there? What are they 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>.
- If you haven't already, look up the main elements in the `<teiHeader>` on the TEI-C website and see what they are allowed to contain.
- You could also have a look at the TEI Guidelines chapter of the Header at <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/HD.html>.