

## Exercise 3: Creating a Better TEI Header

James Cummings – @jamescummings

2016-07-05

### 1 Learning Outcomes

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

- 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.
- record major changes to the file in the `<revisionDesc>`.
- create and use classifications of a user-created `<taxonomy>`

### 2 Summary

This exercise gives you a chance to improve various aspects of the `<teiHeader>` and shows how to record additional metadata about the electronic file, its sources, and revisions.

### 3 Starting Up

Load up a the file you finished in the previous exercise. If you didn't have time to finish it all, or want to start afresh, then you can get a spoiler for exercise 2 at [view-source:http://tei.it.ox.ac.uk/Talks/2016-07-dhoxss/spoilers/ex02-spoiler.xml?style=raw](http://tei.it.ox.ac.uk/Talks/2016-07-dhoxss/spoilers/ex02-spoiler.xml?style=raw) or <http://tei.it.ox.ac.uk/Talks/2016-07-dhoxss/spoilers/ex02-spoiler.xml?style=raw> and save that. Load up your file and save it as `exercise03.xml` or similar.

### 4 The current `<teiHeader>`

The current `<teiHeader>` isn't very complete. Currently it looks something like:

```
<teiHeader>
  <fileDesc>
    <titleStmt>
      <title>My Shakespeare Collection</title>
    </titleStmt>
    <publicationStmt>
      <p>Created as part of a TEI Workshop</p>
    </publicationStmt>
    <sourceDesc>
      <p>Shakespeare's Sonnets and First Folio.</p>
    </sourceDesc>
  </fileDesc>
</teiHeader>
```

While this is technically valid it is minimally informative given the potential richness of the `<teiHeader>` to store all kinds of metadata!

### 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 other elements 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 'William Shakespeare'. (Although <titleStmt> is about the digital file, Shakespeare is still the author!)
- Below this add an <editor> element with your name as the content. If your name has a character with an accent in it, how would you type it? You could just use keyboard shortcuts that you already know about (if it is your name!) but you can also use the 'Insert from Character Map' entry from the 'Edit' menu.
- After that add <respStmt> with a <resp> inside it saying 'TEI Encoding' and a <name> with your name.
- Your <titleStmt> should now look something like:

```
<titleStmt>
  <title>My Shakespeare Collection</title>
  <author>William Shakespeare</author>
  <editor>[Your Name Here]</editor>
  <respStmt>
    <resp>TEI 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 Guidelines.

### 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. When you do your file should be invalid (because <publicationStmt> requires something in it).
- Inside <publicationStmt> add a <publisher> element. The publisher is the organisation responsible for the publication, in this case, 'TEI Workshop' is the publisher.
- Next, inside a <address> element create a child <orgName> (e.g. IT Services, University of Oxford), next to this a <street> address (e.g. 13 Banbury Road), a <settlement> (Oxford), a <postCode> (e.g. OX2 6NN), and a <country> (e.g. United Kingdom). (Or use the address of your own institution!)
- After the <address> element add a <date> element with content of '5 July 2016'. The <date> element can have a @when attribute to take a standardised YYYY-MM-DD form of the date, add this as well.
- Below the <date> add a <distributor> containing 'IT Services, University of Oxford'. This is the organisation responsible for the distribution of the digital text.

- After this add an `<authority>` element, to detail under whose authority it is published. In this case let's say it is under your authority, so add your name.
- Next add an `<availability>` statement with a `<licence>` containing a description of the licence you would want to distribute this under with a `@target` attribute pointing to the description of the license in question. For example, you might chose the license at the following target `'https://creativecommons.org/licenses/by/3.0/'`.
- 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 file. Otherwise you could use something like `'TEI-Shakespeare-Collection-001'`.
- Your `<publicationStmt>` should now look something like:

```

<publicationStmt>
  <publisher>TEI Workshop</publisher>
  <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>
  <date when="2016-07-05">5 July 2016</date>
  <distributor>IT Services, University of Oxford</distributor>
  <authority>[Your Name Here]</authority>
  <availability>
    <licence target="https://creativecommons.org/licenses/by/3.0/">Creative
    Commons
      Attribution 3.0 Unported</licence>
  </availability>
  <idno>TEI-Shakespeare-Collection-001</idno>
</publicationStmt>

```

## 7 Improving the `<sourceDesc>`

Our `<sourceDesc>` is also fairly limited.

- Delete the entire paragraph `<p>` element that is currently in the `<sourceDesc>`. Add a `<bibl>` element with an `@xml:id` attribute of `'Sonnets'`.
- Inside this `<bibl>` add an `<author>` element of `'William Shakespeare'` and after this, inside a pair of round parentheses, add a `<date>` element with the content `'1609'`. Put a full-stop (or period for Americans) after the closing parenthesis. Follow this with a `<title>` element of `'Shake-speares Sonnets: Never Before Imprinted.'`, after this a `<pubPlace>` of `'London'` followed by a colon, and a `<publisher>` of `'Thomas Thorpe'` followed by another full-stop.
- This `<bibl>` should now look something like:

```

<bibl xml:id="Sonnets">
  <author>William Shakespeare</author> (<date>1609</date>). <title>Shake-
  speares Sonnets:
    Never Before Imprinted.</title>
  <pubPlace>London</pubPlace>: <publisher>Thomas Thorpe</publisher>.
</bibl>

```

Notice how the loose and free structure of `<bibl>` enables you to mix and match punctuation and elements as you desire. In formatting a bibliography would you prefer this or the more structured format of `<biblStruct>`?

- That has added a bibliographic record for the source of our sonnet, what about the First Folio copy of *Henry IV Part One*? Add a second `<bibl>` with an `@xml:id` attribute of 'FirstFolio'.
- The First Folio bibliographic citation has some other metadata we want to store with it. The record as a whole contains the following aspects. Try to create a `<bibl>` for it and check your result with ours below.
  - Author: Shakespeare, William, 1564-1616.
  - Title (statement): Mr. William Shakespeares comedies, histories, & tragedies.: Published according to the true originall copies.
  - Title (variant): Mr. VWilliam Shakespeares comedies, histories, & tragedies
  - Title: (distinctive): First Folio
  - Publication Place: London, England:
  - Publishers: William Jaggard, Edward Blount, John Smethwicke
  - Date (canonical): 1623
  - Date (entry): 8 November 1623 (entered)
  - ID number (shelfmark): Bodleian Library, Arch. G c.7
  - ID number (estcCitationNo): S111228

Use this information to create a `<bibl>` element for the Bodleian's First Folio.

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

```
<sourceDesc>
  <bibl xml:id="Sonnets">
    <author>William Shakespeare</author> (<date>1609</date>). <title>Shake-
speares Sonnets:
  Never Before Imprinted.</title>
  <pubPlace>London</pubPlace>: <publisher>Thomas Thorpe</publisher>. </bibl>
  <bibl xml:id="FirstFolio">
    <author>Shakespeare, William, 1564-1616.</author>
    <title type="statement"> Mr. William Shakespeares comedies, histories, &
tragedies.:
  Published according to the true originall copies.</title>
    <title type="variant">Mr. VWilliam Shakespeares comedies, histories, &
tragedies</title>
    <title type="distinctive">First Folio</title>
    <pubPlace>
      <settlement>London</settlement>, <country>England</country>
    </pubPlace>: <publisher>
      <persName>William Jaggard</persName>, <persName>Edward Blount</persName>,
<persName>John
      Smethwicke</persName>
    </publisher>
    <date type="canonical" when="1623">1623</date>
    <date type="entry" when="1623-11-08">8 November 1623 (entered)</date>
    <idno type="shelfmark">Bodleian Library, Arch. G c.7</idno>
    <idno type="estcCitationNo">S111228</idno>
  </bibl>
</sourceDesc>
```

---

## 8 Adding an `<encodingDesc>`

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

- After the closing `</fileDesc>` you should add an `<encodingDesc>` element.
- Inside an `<encodingDesc>` there are elements for describing the project, the declaration of editorial principles and methods, and several other things. We're going to use it to encode only a project description and a nested heirarchy of categories.
- Inside `<encodingDesc>` add a `<projectDesc>` with a `<p>` inside that. Usually your project description would have multiple paragraphs, in this case we're just going to have the one saying something like 'This collection was created as part of a TEI Training Workshop.'
- After the `<projectDesc>`, as a sibling to it, we're going to add a `<classDecl>` with a `<taxonomy>` inside that, and a `<category>` inside that. The `<category>` should have an `@xml:id` attribute of 'genres', and a child `<catDesc>` element of something like 'A list of genres'.
- Your `<encodingDesc>` should look something like:

```
<encodingDesc>
  <projectDesc>
    <p>This collection was created as part of a TEI Training Workshop.</p>
  </projectDesc>
  <classDecl>
    <taxonomy>
      <category xml:id="genres">
        <catDesc>A list of genres</catDesc>
      </category>
    </taxonomy>
  </classDecl>
</encodingDesc>
```

- Now let's add some more heirarchy to this – the great thing about taxonomies is they can be extremely flexible and arbitrarily deeply-nested. Also, by pointing to one category you automatically are expressing a relationship to all the rest of them above it. Add `<category>` elements with `@xml:id` attributes and `<catDesc>` elements for the following genres:
  - prose: Prose Texts
  - poetry: Poetic Texts
  - drama: Dramatic Texts
- This `<category>` should now look something like:

```
<category xml:id="genres">
  <catDesc>A list of genres</catDesc>
  <category xml:id="prose">
    <catDesc>Prose Texts</catDesc>
  </category>
  <category xml:id="poetry">
    <catDesc>Poetic Texts</catDesc>
  </category>
```

```
<category xml:id="drama">
  <catDesc>Dramatic Texts</catDesc>
</category>
</category>
```

- Inside the 'poetry' category, add a sub `<category>` for Sonnets (*@xml:id* sonnets), and inside this another for Shakespearean Sonnets (*@xml:id* shakesSon)
- Now your `encodingDesc` should look something like:

```
<encodingDesc>
  <projectDesc>
    <p>This collection was created as part of a TEI Training Workshop.</p>
  </projectDesc>
  <classDecl>
    <taxonomy>
      <category xml:id="genres">
        <catDesc>A list of genres</catDesc>
        <category xml:id="prose">
          <catDesc>Prose Texts</catDesc>
        </category>
        <category xml:id="poetry">
          <catDesc>Poetic Texts</catDesc>
          <category xml:id="sonnets">
            <catDesc>Sonnets</catDesc>
            <category xml:id="shakesSon">
              <catDesc>Shakespearean Sonnets</catDesc>
            </category>
          </category>
        </category>
      </category>
      <category xml:id="drama">
        <catDesc>Dramatic Texts</catDesc>
      </category>
    </taxonomy>
  </classDecl>
</encodingDesc>
```

## 9 Adding a `<revisionDesc>`

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

- After the closing `</encodingDesc>` add a `<revisionDesc>` element.
- Add two `<change>` elements. On the first one add a *@when* attribute with today's date (YYYY-MM-DD). Inside the `<change>` add a `<name>` containing your name, following this element by the text 'updated the file to have a better header'.
- In the second `<change>` add a *@when* attribute of yesterday's date, with a `<name>` containing your name, with some text saying 'created the initial file'.
- It is usual practice for the most recent `<change>` to be first.
- Your `<revisionDesc>` should now look something like:

---

```

<revisionDesc>
  <change when="2016-07-05">
    <name>[YOUR NAME]</name> updated the file to have a better header </change>
  <change when="2016-07-04">
    <name>[YOUR NAME]</name> created the intial file </change>
</revisionDesc>

```

## 10 Using those <categories>

Remember up above when you added those categories? Now is the time to use them.

- Go down to the <div> which contains your sonnet. We want to add a bit of analysis that this is a Shakespearean Sonnet. We also want to note the source of this bit of the collection. To do the former we add an @ana attribute with one or more white-spaced separated URIs. To add the latter we do the same but on a @decls attribute (which says that a particular part of the header applies to this be of the text). We will discuss URI-base pointing later, but this could look like '#anIDInTheDocument', 'filename.xml#AnID', 'http://www.example.com/filename.xml#AnID' or using a private URI syntax (which we'd document in the header) 'myProj:AnID'. In our case we have an existing @xml:id of 'shakesSon' which we can use to indicate this is a Shakespearean Sonnet, and an @xml:id of 'Sonnets' on the bibliographic citation. Add an @ana attribute with value of '#shakesSon' and a @decls attribute with value of '#Sonnets'. It should look like:

```

<div type="verse" ana="#shakesSon"
  decls="#Sonnets">
<!-- rest of div here -->
</div>

```

- Further down on your <div> that contains a bit of a play, we want to do the same kind of thing but reference the 'FirstFolio' id of the bibliographic source and the category of 'drama'. It should look like:

```

<div type="drama" ana="#drama"
  decls="#FirstFolio">
<!-- rest of div here -->
</div>

```

What do these associations we've made do? Not much by themselves, it has documented the associations of the bits of the text with various other bits of metadata. However, these could then be used by a step in processing this to, for example, find all the parts of a collection that were in one category or the other (or both).

## 11 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 'exercise02.xml' or another name of your choice.

### 12 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>`?

### 13 Next and Further Reading

- You may wish to browse through the TEI Guidelines online at <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/index.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 parts of the `<teiHeader>` are there? What are they for?
- 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>.