Exercise 2: A Basic `<msDesc>`

March 2009

1 Before you start

This exercise is designed to get you to familiarise yourself with both the oXygen editing environment and the TEI manuscript description module by taking a very basic manuscript description and giving it some real structure.

2 Starting oXygen

Start up oxygen. When it starts up it may ask for a license key. If it does, navigate to your USB key and load up the file 'license.txt' and cut-and-paste its contents into the registration box. This is a trial license lasting for about 60 days and gives you full access to the program. Optionally you may wish to shut down some of the helper views that oXygen provides on the left and right of the main editing screen.

3 Loading a document

Once oXygen has loaded and you've dismissed any helpful tips it wants to give you, load a file (File -> Open) by browsing to the usb key's folder and opening the file 'ex2.xml'.

4 Our basic `<msDesc>`

As you will notice this contains a fully complete, valid, and well-formed manuscript description. Technically it validates against the ENRICH schema, but consists only of the required elements and attributes. The msDesc element has not only the TEI namespace delcaration but the required xml:id and xml:lang elements. The msDesc contains inside it only two children, the `<msIdentifier>` element (with the required `<repository>`) and a `<p>` element containing the prose of the manuscript description from your first exercise.

5 Making a real `<msIdentifier>`

1. Begin by highlighting and deleting the `<repository>` element and its contents. To do this move the cursor to before the start of the `<repository>` tag, hold down the shift key, move to the end of the closing `</repository>` tag, and then press delete. When you do so notice that closing `</msIdentifier>` tag gets underlined in red, that a red line appears across from it on the right-hand bar, and there is a red box in the upper right-hand corner. These are all signs to you that there is an error and where it is. Hovering your mouse over the red box in the upper right-hand side of the editor will provide a tooltip pop-up indicating the nature of the error. Alternatively, there should be an error message in the status bar at the bottom. In this case oXygen is complaining because the ENRICH schema requires there to be either `<repository>` or a `<msName>`

2. Return to the paragraph and delete the phrase 'Stored in'.

3. Just after that highlight from the word 'Lithuania' through until the end of the word 'Department.' including the period. 'Cut' this text by pressing (usually) 'control-x', and then after moving back up to the `<msIdentifier>` paste it by pressing 'control-v'. oXygen should still complain about you putting text here because it expects some structure

4. Highlight the word 'Lithuania', and then select the menu item 'Document -> XML Refactoring -> Surround with Tags' (note that the shortcut key for this is 'control-e'). This should pop up a window which allows you to choose any element valid at this point to surround this bit of text with. Type 'country' and notice how it completes the word as you type it. Pressing enter will inside the element around the text 'Lithuania'.

5. Delete the comma after Lithuania, press enter to move 'Vilnius' to a new line, highlight 'Vilnius', press 'control-e' and type 'settlement'. Press enter to insert this element.

6. Delete the comma and press enter to move the rest of the text down to a new line.

7. Use this same technique to add a `<institution>` element around the text: 'Lithuanian National M. Mazvydas Library'; and a `<repository>` element around 'Rare Book and Manuscript Department'. Delete any unnecessary commas or periods.

8. Your document should now be well-formed and valid. (There should be no red lines or error messages, and that box in the upper right-hand corner should be green. If that isn't the case, make sure you have deleted any stray punctuation in between the elements.

9. Move to just before the closing `</msIdentifier>` tag and press enter to insert a blank line inside it. Type the open angle bracket '<' and notice what happens. oXygen should provide you with a drop down list of elements which it is valid to insert at this point. Choose `<idno>` and notice that it inserts the element and leaves you inside the opening tag in case you wanted to add some attributes. Move down and cut and paste 'F101-19' in between the starting `<idno>` tag and its closing `</idno>` tag. Delete the stray colon that you did not cut and paste.

10. Your `<msIdentifier>` should now look something like:

```
<msIdentifier>
 <country>Lithuania</country>
 <settlement>Vilnius</settlement>
 <institution>Lithuanian National M. Mazvydas Library</institution>
 <repository>Rare Book and Manuscript Department</repository>
 <idno>F101-19</idno>
</msIdentifier>
```

and your document should still be well-formed and valid. If it isn't, correct it before continuing. When it is you can choose the menu item 'Document -> XML Document -> Format and Indent' to tidy up the indenting of the elements.

6   Creating an `<msContents>` element

1. Immediately after the closing `</msIdentifier>` tag, type an angle bracket: '<' to get oXygen to present a drop-down list of elements valid at this point. Select `<msContents>`. Notice that the `<p>` below is now underlined in red, because the ENRICH schema doesn't allow a vague pararaph and structural divisions to co-exist in this manner.

2. Move in between the opening and closing tags of the `<msContents>` element and again type '<' to see a list of possible elements. In this case choose `<summary>`.

3. Go to the paragraph down below and copy and paste everything from 'An original copy of a' all the way to 'Bug (Bugk) river.' into the `<summary>` element you have just created. This is a prose description of the intellectual content of the manuscript.

4. Press enter between the closing `</summary>` tag and the closing `</msContents>` tag and then add a new `<textLang>` element by typing '<'. Notice how oXygen (usually) automatically adds the required mainLang attribute and places your cursor and just the right place to type in the ISO language code 'lat' for Latin. Type it in.

5. Move to just after the closing double-quote of the mainLang attribute. Press space. oXygen should provide a drop down menu of all other attributes valid on this element. Scroll down and choose 'otherLangs', and since the descriptive paragraph below tells us it contains Polish as well as Latin, add the code 'pol' as the otherLangs attribute value.

6. Cut and paste the text 'Latin. There are postscripts in Polish.' into the middle of the `<textLangs>` element.

7. After the `<textLangs>` element but before the closing `</msContents>` tag, add in an `<msItem>` element by typing it in. Inside the `<msItem>` element create an `<incipit>` element in a similar manner. Cut and paste the text 'In nomine Domini amen. Ne orror obliuionis...' in between the `<incipit>` tags, including the final ellipsis.

8. While the document still isn't valid, because of that `<p>` down below, there should be no red lines inside your `<msContents>` which should look something like:

```
<msContents>
 <summary>An original copy of a charter where Alexander (Alexander), King
  of Poland and Grand Duke of Lithuania, transumes and confirms the
  privilege of January 26, 1380, by which Vytautas (Vitholdus), Grand
  Duke of Lithuania, gives permission to establish a monastery of the
  Augustinian Order in Brest (Bresth) and founds plots of land and
  Kostomoloty (Koszthomlothy) village near the Bug (Bugk)
  river.</summary>
 <textLang mainLang="lat" otherLangs="pol">Latin.  There are postscripts
  in Polish.</textLang>
 <msItem>
  <incipit>In nomine Domini amen.  Ne orror obliuionis...</incipit>
 </msItem>
</msContents>
```

3

7   Structuring a `<physDesc>`

Using the techniques you've learned above of wrapping an element around some text (control-e) or just starting to type the element in ('<'), separate the physical description information into a `<physDesc>` element and its possible children, something like below:

```
<physDesc>
 <objectDesc form="leaf">
  <supportDesc material="perg">
   <support>Parchment.  Single sheet format:  485 x 385 + 115 mm.</support>
  </supportDesc>
  <layoutDesc>
   <layout columns="1">Text:  375 x 225 mm.</layout>
  </layoutDesc>
 </objectDesc>
 <decoDesc>
  <decoNote>The ornamented initial "I" takes up 30 lines, 15 cm.</decoNote>
 </decoDesc>
 <sealDesc>
  <seal>
   <p>The lesser red wax attached seal of GDL (dia.  4 cm) in a black wax
bowl (dia.  8.5 cm) is affixed to the document with a string of rosy thread
(parchment fold-up is 11.5 cm, two slits for the string are spaced 8 cm
apart).</p>
  </seal>
 </sealDesc>
</physDesc>
```

Pay particular attention to the required attributes on some of the elements. These are not required in TEI generally, but here we are using the ENRICH schema, which holds us to a higher standard. Thus, if these elements are present, we must provide a form attribute on `<objectDesc>`, a material attribute on `<supportDesc>` and a columns attribute on `<layout>`.

8   The importance of `<history>`

Most of the text of the paragraph below should now be gone. Use the remaining text to create a `<history>` element containing the crucially important origin date and place (`<origDate>` and `<origPlace>`). Your history element should look something like:

```
<history>
 <origin>
  <origDate when="1502-06-24">[June 24], 1502.</origDate>
  <origPlace xml:lang="lat">Kamenec (Kamieniecz)</origPlace>
  <p>Authenticity - original.</p>
 </origin>
</history>
```

Here we have also added a paragraph stating that this is an original of the charter (rather than a contemporary copy).

   Make sure to save a copy of the completed file, as we will want to use it in our next exercise.

9   Other things to try ...

1. Looking at the image of the manuscript, what things have we not included in this basic manuscript description?

2. What TEI elements might exist to record this information?

3. What TEI phrase level elements might you use to mark up the information further?