

Manual for TEI at Wellcome Collection

2020

INTRODUCTION	3
WHAT IS TEI?	3
WHAT DOES THIS MANUAL CONTAIN?	3
A NOTE ON DEVIATIONS FROM THE MSDESC SCHEMA.	3
A NOTE ON TEI EDITORS	4
SINGLE OR COMPOSITE MANUSCRIPT?	4
SINGLE-UNIT MANUSCRIPTS	4
COMPOSITE MANUSCRIPTS	5
SINGLE-PART MANUSCRIPTS	6
THE STRUCTURE OF THE TEI FILE	6
THE PROCESSING INSTRUCTIONS	7
<TITLESTMT>	8
<RESPSTMT>	9
<PUBLICATIONSTMT>	10
A NOTE ON <SOURCEDESC> AND <MSDESC>	10
<MSIDENTIFER>	11
RECORDING SIERRA NUMBER	12
<MSCONTENTS>	12
<SUMMARY>	12
LANGUAGE	13
<MSITEM>	13
<PHYSDESC>	15
<OBJECTDESC>	15
<HANDDESC>	15
<HISTORY>	15
<ADDITIONAL>	15
<REVISIONDESC>	15
MULTI-PART MANUSCRIPTS	16
APPLYING TRANSFORMATION SCENARIOS	17
GLOSSARY	21
COLOPHON	21
INCIPIT	21
EXPLICIT	21
RECTO	21
VERSO	21

Introduction

What is TEI?

This manual provides guidelines for TEI manuscript cataloguing at [Wellcome Collection](#). Wellcome holds over 20 000 manuscripts from diverse historical and cultural contexts and across a range of languages and scribal practices. The flexible, extensible manuscript description interface provided by the [Text Encoding Interface \(TEI\)](#) presents a possibility for manuscript cataloguing practice that is responsive and flexible in dealing with a range of manuscript materials, without sacrificing coherence and consistency.

The starting point for TEI for manuscript description at Wellcome Collection is the schema and guidelines developed at Oxford and Cambridge, commonly referred to as the [msDesc schema](#). Wellcome aims to contribute to the further development of the msDesc schema. By embracing cross-organisational work towards common guidelines we wish to facilitate joint cataloguing efforts and avoid duplication of effort.

A detailed overview of the msDesc schema is available in [HTML](#) format.

What does this manual contain?

This manual is intended as a gentle, step-by-step introduction to TEI for manuscript description at Wellcome Collection. Unlike other introductions to TEI, which work on a more conceptual level, this manual takes its starting point in Wellcome Collection's [TEI templates](#):

- [Single-part manuscript template](#).
- [Multi-part manuscript template](#).

The manual is divided into two major sections, one for each template. Each section provides a section-by-section walkthrough of each template and is meant to guide manuscript entry in TEI across projects, materials, and sources.

The manual does not cover every conceivable element available through the msDesc schema, but rather seeks to enable the TEI encoder to use the Wellcome templates to make informed choices appropriate to the manuscript material they are working on.

A note on deviations from the msDesc schema.

Wellcome Collection contributes TEI descriptions of our Arabic manuscripts to the Fihrist database. These files are fully compliant with the msDesc schema. However, manuscript cataloguing projects conducted internally in Wellcome Collection require particular solutions, leading to some deviations:

- There is no central authority file in use, and so the Schematron processing instruction is not applied in the template.

- Manuscript descriptions at Wellcome Collection must be coordinated with other collections information practices at Wellcome, such as recording of Sierra numbers, etc. Specific solutions to these situations are reflected in the templates and this manual.
- On a more general level, Wellcome Collection's use of TEI is primarily intended to address lacunae in our cataloguing practices and to make a wider selection of manuscript collections accessible and discoverable. Consequently, we are currently (2020) focusing on recording as much material as possible in TEI, prioritising completion of whole collections over details on the item-level.

A note on TEI editors

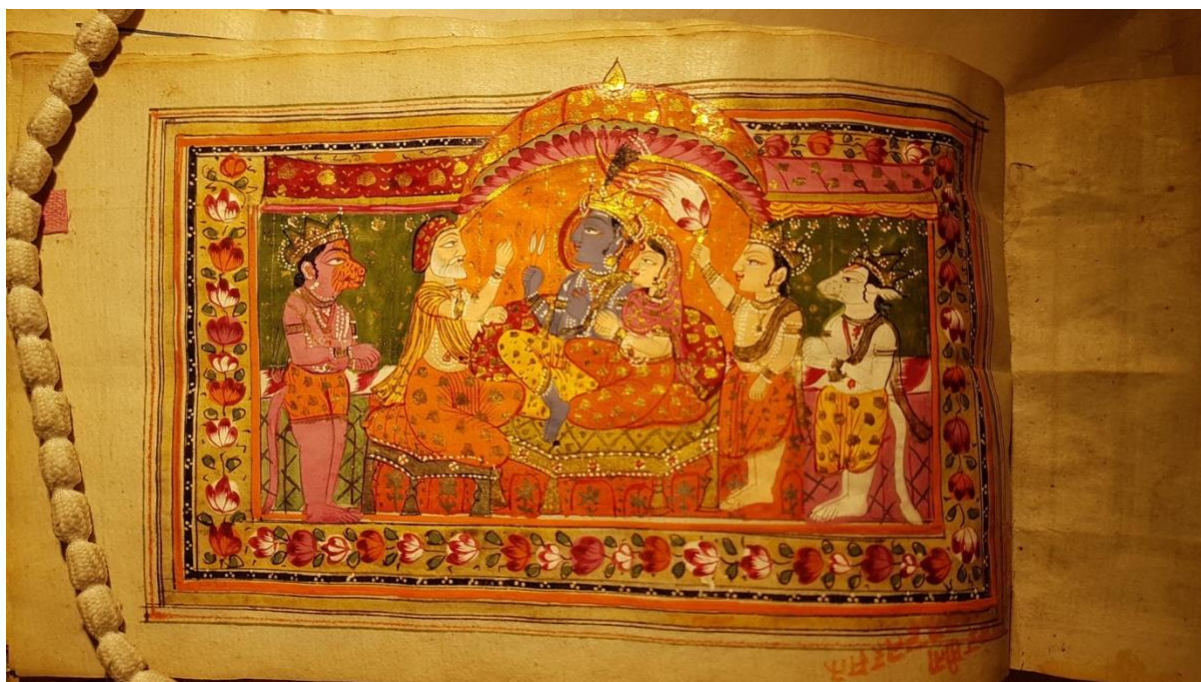
The standard TEI editor at Wellcome Collection is the [Oxygen XML Editor](#), and the screenshots in this manual are mostly from Oxygen. However, this manual is not a manual for using Oxygen, and we are happy to work with TEI files developed in other editors.

Single or composite manuscript?

The starting point for manuscript description is to determine whether the manuscript is a single codicological unit or several.

Single-unit manuscripts

A manuscript containing several different texts that were all produced as part of the same manuscript is a single-unit manuscript. An example would be [MS Indic Alpha 978](#), which is a bound volume containing a number of Sanskrit mantras and hymns, each of them prefaced with a single-page illustration and a title page with Nastaliq script. While the manuscript clearly contains a number of distinct titles or items, these items were all produced as part of a single, uniform unit.



A detail from MS Indic Alpha 978.

In these cases, we use the [single-part manuscript template](#).

Composite manuscripts

A composite manuscript is a manuscript in which previously distinct units have been gathered into one volume. For example, [Wellcome MS Malay 7](#) contains a number of different papers, each with their distinct origin, collected inside a single folder. In these situations, we use the [multi-part manuscript template](#).

[Wellcome MS Malay 7 is fully digitised](#).

For more on multi-part manuscripts, see [1.6.7. Multi-part, composite and similar units](#) in the msDesc guidelines.

Single-part manuscripts

The single-part manuscript template contains a single <TEI> element, which in turn contains three elements: <teiHeader>, <facsimile>, and <body>. Unless we want to enter links to digitised versions of the manuscript or provide a full transcript of its contents, we do not enter any information in the two latter elements.

Over the following sections, we will only deal with the various components of the <teiHeader> element, which is where we enter all of the metadata about the manuscript and the file itself. The headings of the following sections all refer to elements within the <teiHeader>.

The structure of the TEI file

Before we can delve into the particular groups of elements that together make up the TEI file, it's useful to take a bird's eye view of the file as a whole.

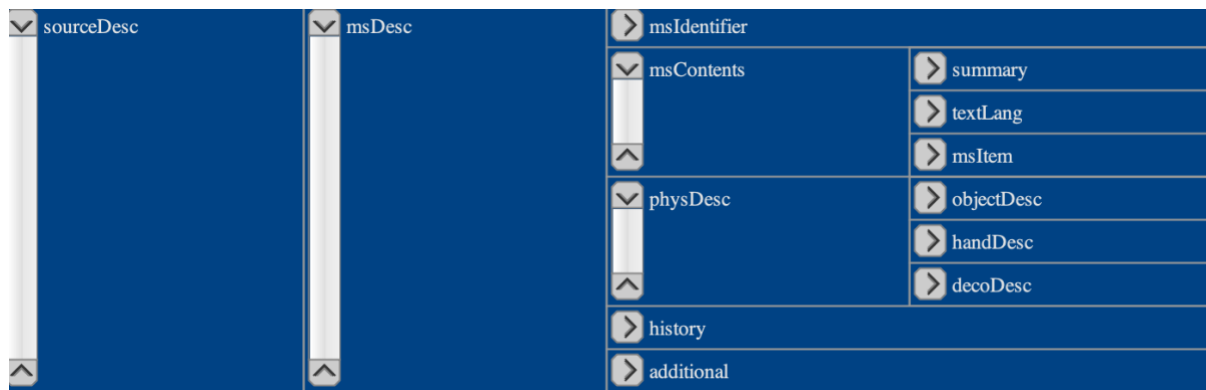
We can take [Wellcome Hebrew A1](#) as our example and use the 'grid' function in Oxygen XML to study its structure.

TEI	@xmlns	http://www.tei-c.org/ns/1.0
	@xml:id	Hebrew_A_1
	teiHeader	fileDesc
		revisionDesc
	facsimile	
	text	

Reading the grid from left to right, we see that the <TEI> element contains three elements, <teiHeader>, <facsimile>, and <text>. Of these three, <teiHeader> contains two further elements, <fileDesc> and <revisionDesc>. (The two top rows of the right column contain information specific to the TEI namespace and the values of the file itself.)

teiHeader	fileDesc	titleStmt
		publicationStmt
		sourceDesc
	revisionDesc	

Looking closer, we see that <fileDesc> in turn contains three elements, <titleStmt>, <publicationStmt>, and <sourceDesc>. Of these three, <sourceDesc> is by far the most expansive.



It is clear from the above grid view that the majority of our work is done within the `<sourceDesc>` element.

In the following, we will focus on the contents of individual elements within this structure. However, it is always helpful to remember where each element sits within the whole. If anything ceases to make sense, try to take one step back and look at where the element you're working on sits in the overall structure.

The processing instructions

The first two lines of the file contain what is known as the operating instructions. These inform your editor that you are working on an XML file and to apply the instructions that are specific to the `msDesc` schema.

As noted above (A note on deviations from the `msDesc` schema.), the template does not use `msDesc` schema's Schematron processing instruction to further restrict responses. When contributing to union catalogues, however, we use templates that also contain this instruction. We are also free to create our own Schematron instructions to steer particular responses, should any of our projects require us to do so.

For most projects, however, we will only need the two processing instructions given in the template file.

<titleStmt>

```
<titleStmt>
  <title></title>
  <title type="collection">Wellcome Collection</title>
  <respStmt xml:id="">
    <resp when="">Cataloguer</resp>
    <persName><!-- Insert name --></persName>
  </respStmt>
  <respStmt xml:id="">
    <resp when="">TEI Encoder</resp>
    <persName><!-- Insert name --></persName>
  </respStmt>
  <!-- Add another <respStmt> for every resp -->
</titleStmt>
```

We use the <titleStmt> to record the name of the file itself, and the names of the people who have worked on it.

We first record the title of the file (not the manuscript itself!) in the <title> element. We also need to give this title as the attribute of the xml:id value of the <TEI> element itself:

```
<TEI xmlns="http://www.tei-c.org/ns/1.0" xml:id="">
```

You must use underscore instead of space in an xml:id, like so:

```
<TEI xmlns="http://www.tei-c.org/ns/1.0" xml:id="Hebrew_A_1">
```

Also keep in mind that xml:id contents must always be unique within the same file. We'll come back to this throughout the following, but it is especially important when we work with multiple <msItem> elements.

<respStmt>

The <respStmt> element allows us to record the people who have worked on the information in the file or the file itself, and helps us maintain a helpful log of changes in the <revisionDesc>.

For instance, we often begin our work by entering cataloguing information about the manuscript from a legacy catalogue. In these cases, we want to record both the people involved with the cataloguing and the work of the encoders. Here is an example from [Wellcome MS Malay 5](#):

```
<respStmt xml:id="LXT">
  <resp when="2019">Cataloguer</resp>
  <persName>Li-Xuan Teo</persName>
</respStmt>
<respStmt xml:id="RV">
  <resp when="1982">Cataloguer</resp>
  <persName>M. C. Ricklefs and P. Voorhoeve</persName>
</respStmt>
<respStmt xml:id="EHM">
  <resp when="1981">Cataloguer</resp>
  <persName>R.F. Ellen, M.B. Hooker and A.C. Milner</persName>
</respStmt>
<respStmt xml:id="BLE">
  <resp when="2020">TEI Encoder</resp>
  <persName>Branwen Lloyd-Evans</persName>
</respStmt>
```

In this example, the TEI encoder was working from three different catalogue entries for the same manuscript, done by different people at different times.

You can use as many <respStmt> elements as you need. Remember to give a unique xml:id identifier to each <respStmt>, so that you can refer back to the person throughout the rest of the file. Initials are plenty.

You can also record the date more precisely than just recording the year, but it's not necessary. If several people are working on the same file within the same year, it works just fine to record the dates either of you did something in the <revisionDesc>.

<publicationStmt>

The <publicationStmt> element records information about the institution responsible for publishing the TEI file itself. As such, we can just use the information that is already entered into the template.

```
<publicationStmt>
  <publisher>Wellcome Collection</publisher>
  <idno>UkLW</idno>
  <address>
    <orgName type="institution">Wellcome Collection</orgName>
    <street>215 Euston Rd</street>
    <settlement>London</settlement>
    <postCode>NW1 2BE</postCode>
  </address>
  <distributor>
    <email>library@wellcome.ac.uk</email>
  </distributor>
  <idno type="msID"></idno>
  <idno type="catalogue"></idno>
</publicationStmt>
```

The only elements that need additional information here are the final two <idno> elements. The one with the attribute 'msID' should contain the shelfmark or identifying title of the manuscript itself, while the 'catalogue' one should record the name of the catalogue from which the information is taken. If no catalogue is used, the element can be dropped. If multiple catalogues have been used, additional <idno> elements with the "catalogue" attribute can be added. Below you can see an example of these elements in [Wellcome Malay 9](#):

```
<idno type="msID">Wellcome Malay 9</idno>
<idno type="catalogue">The Hervey Malay Collection in the Wellcome Institute</idno>
<idno type="catalogue">Catalogue of Malay manuscripts in the Wellcome Institute for the History of Medicine</idno>
```

A note on <sourceDesc> and <msDesc>

As you can see in the section on TEI file structure above, a major part of the file consists of the element <msDesc> which is in turn nested in the element <sourceDesc>. The following sections are mostly all about the elements within <msDesc>.

<msIdentifier>

The <msIdentifier> is, like the <publicationStmt> another element with mostly prefilled data. The first half of it looks like this:

```
<msIdentifier>
  <country>United Kingdom</country>
  <settlement>London</settlement>
  <institution>Wellcome Collection</institution>
  <repository>Wellcome Library</repository>
```

The final section, however, is important and must be approached with care. Its purpose is to record all of the identifiers attached to the manuscript. In the template, it looks like this:

```
<idno type="shelfmark"></idno>
<!-- Use <altIdentifier> if former or alternative <idno> is available -->
<altIdentifier type="">
  <idno></idno>
</altIdentifier>
```

As you can see, the default identifier, or <idno>, is the “shelfmark”. This is the shelfmark number or identifier currently given to the manuscript.

In the <altIdentifier>, you can give any other identifiers available for the manuscript. These might include accession numbers, deprecated identifiers from earlier cataloguing work, and anything else. You can give any type value here. For instance, [MS Indic Beta 2183](#) gives both its current shelfmark and an alternative identifier from Professor Raghavan’s earlier cataloguing work on Wellcome’s South Asian manuscript collection:

```
<idno type="shelfmark">MS Indic Beta 2183</idno>
<altIdentifier type="Raghavan">
  <idno>B.19.aa</idno>
</altIdentifier>
```

As you can see, “Raghavan” has been given as the type value for the <altIdentifier> element. You can add any kind of added identifier in this way.

Recording Sierra number

Most important, however, is recording the Sierra number where this is available. It is done in exactly the same way as in the examples above. Here is how it's given in [Hebrew A21](#):

```
<idno type="shelfmark">WMS Hebrew A21</idno>
  <altIdentifier type="Sierra">
    <idno>b3227466x</idno>
  </altIdentifier>
```

In many instances, we only have a dummy entry in Sierra while the core of the manuscript metadata is recorded in TEI. Recording the Sierra number in the <msIdentifier> is crucial to link the two records.

<msContents>

<msContents> contains, as you would expect, information about the contents of the manuscript. <physDesc>, on the other hand, details all of the aspects of the manuscript as a physical object. As such, <msContents> is concerned with both intellectually distinct contents within the manuscript and whatever information might hold true for these as a whole.

The thinking here is that a manuscript that contains, for instance, a series of three different recipes, can be said to contain three intellectually distinct items. Each recipe is such an item. In another example, manuscripts often contain both a text and a commentary on that text. The text and the commentary then constitute two different items.

However, both examples might have some common information. For instance, the text and its commentary might both be in Arabic, and the combined recipes can be described as a recipe collection.

<msContents> is generally divided into <msItem> elements, one for each of the distinct items that make up the manuscript. The first few elements of <msContents>, however, aim to capture only the information that holds true for all of these <msItems>.

<summary>

The <summary> element is a space to record overview-level information that holds true across each <msItem> element and the manuscript as a whole. For instance, [Wellcome Malay 5](#) contains a variety of catalogues of different items and has been given the following <summary> element:

```
<summary>Catalogues of Malayan trees, roots, herbs, fish, snakes and frogs</summary>
```

Note that we give the <summary> element also for manuscripts with a single <msItem> element. This may in some cases just be a repetition of the same information. Note also that we tend not to use the <head> element described in point 1.6.2 of the [msDesc guidelines](#).

Language

The <textLang> element following <summary> is also only used if it applies for every part of the manuscript. If the manuscript contains several <msItem> elements all in the same language, we record this once in <textLang>. See, for instance, [Hebrew A19](#):

```
<summary>Collection of 206 prescriptions with table of contents, magical charms, folk medicine and nursery rhymes.</summary>  
<textLang mainLang="yi" source="IANA">Yiddish</textLang>
```

The <textLang> element typically has the format <textLang mainLang="XXXX" source="IANA">, as in the example above. The language codes are currently from the [IANA language registry](#), but this is open for change.

<msItem>

Before we go on, it is worthwhile to remember the distinction between <msItem> and <msPart>. We only use <msItem> to describe different items within a manuscript that is clearly created as a unitary object. If the items come from different sources entirely and are then bound or gathered together to make up a new manuscript, we use <msPart>. The distinction can be fine at times and is partly up to the encoder's discretion.

The <msItem> element itself takes the following format:

```
<msItem n="1" xml:id="XXX"> [CONTENT] </msItem>
```

Note the following points:

- <msItem> elements are always numbered using n="1".
- Each <msItem> element requires a unique xml:id. We usually construct these by adding underscore and the value of the *n* attribute to the xml:id of the file itself. For instance, a manuscript may have the xml:id 'Manuscript_1', given in the opening <TEI> element; its first <msItem> element will then appear as <msItem n="1" xml:id="Manuscript_1_1".

Our templates typically contain the elements covered in the following.

Locus

The <locus> element is used to indicate in which parts of the manuscript the <msItem> element is found. Here's an example from [Hebrew 17A](#):

```
<msItem n="1" xml:id="Hebrew_A_17_1">
  <locus from="1r" to="17v">ff. 1<hi rend="superscript">r</hi>-17<hi rend="superscript">v</hi></locus>
```

As is clear from the example, the <msItem> element covers folios 1r to 17v (see Glossary for more on *r* and *v*). The added <hi> elements of the content gives superscript rendering.

Title and Author

The <title> and <author> elements give, as you would expect, the title and author of the <msItem> in question. If one or both are unavailable, the elements can be deleted.

You can run a [VIAF](#) search for the author and title, and if available there, give the VIAF-code in the following format from [Indic Alpha 1221](#):

```
<msItem xml:id="Indic_Alpha_1221_1">
  <title key="work_182307811">Raghuvaṃśā</title>
  <author key="person_100903230">Kālidāsa</author>
```

textLang

The <textLang> element within an <msItem> element is used when the language(s) used differ(s) from other <msItem> elements within the same manuscript. If there are several languages in use, we can register this by using several <textLang> elements and giving them the attributes 'mainLang' or 'otherLangs'. See, for instance, how the use of both Latin and Italian is registered in [Hebrew A24](#):

```
<textLang otherLangs="la" source="IANA">Latin</textLang>
<textLang otherLangs="it" source="IANA"> Italian</textLang>
```

Again, the language codes are drawn from [IANA](#).

incipit, explicit, colophon

The slightly forbiddingly sounding elements <incipit>, <explicit>, and <colophon> are all used to record brief transcripts from the manuscript that are intended to make the manuscript item easier to identify. This is especially helpful for less-familiar texts, where the title and author metadata might be entered differently across manuscript catalogues. Having the opening ('incipit') and closing ('explicit') lines of the text available makes a world of difference. Finally, the *colophon* is the metadata entered by the scribe or owner, if available – this is of course also tremendously helpful in identifying things such as date, location, and any other information.

As for the data entry itself, just follow the layout as given in the template.

Filiation

If there are references in the catalogue entry or elsewhere to other manuscripts, you can enter this information in the <filiation> element.

Note

The <note> element can be used to record anything that doesn't fit neatly within the elements covered above. For instance, if the <msItem> does not have any information as to an author or a title, a <note> element can be used to record basic description about the item's content.

<physDesc>

The <physDesc> element is, along with <msItem>, a main element for the recording of manuscript metadata. As its name suggests, the information here is all about the physical description of the manuscript as an item, and the different sub-elements of <physDesc> reflect this. We will here cover these too in turn.

<objectDesc>

The <objectDesc> itself contains several important elements: <supportDesc> and <layoutDesc>.

<handDesc>

<decoDesc>

<additions>

<history>

<additional>

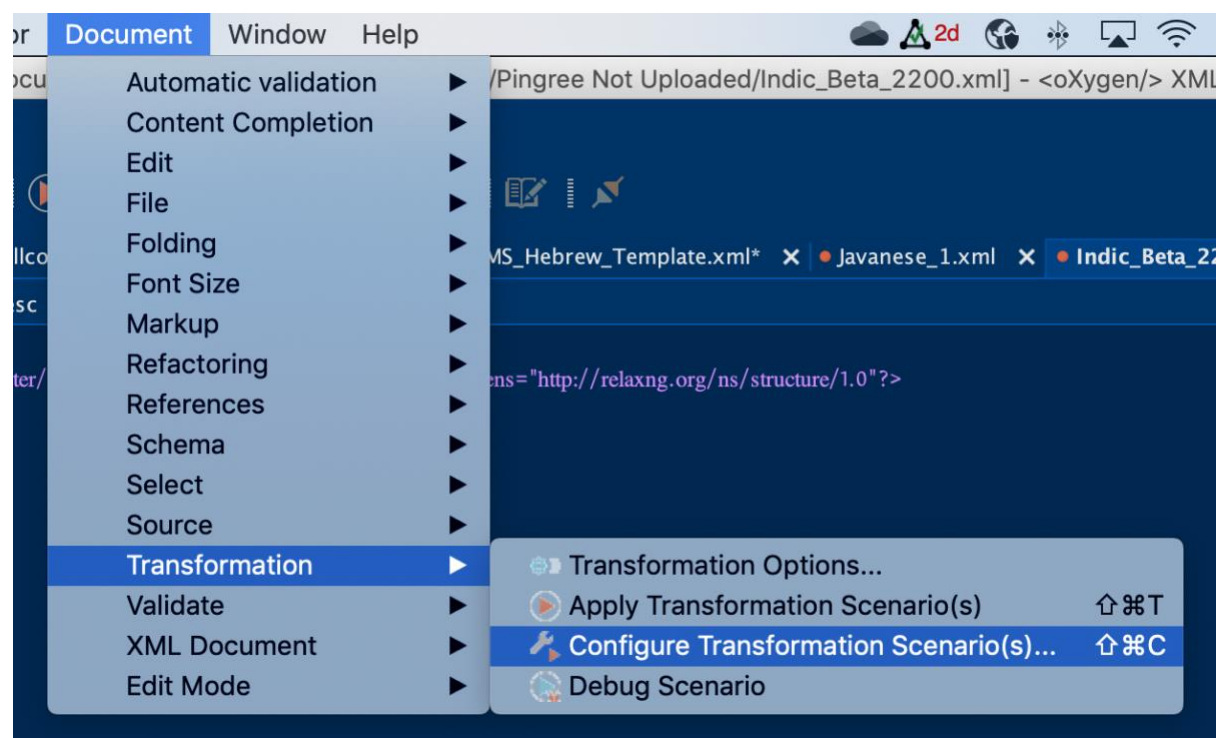
<revisionDesc>

Multi-part Manuscripts

Applying Transformation Scenarios

This is a simplified version of the [guide for applying transformation scenarios](#) written by Andrew Morrison. Use it to get a sense of how your TEI file might look in HTML.

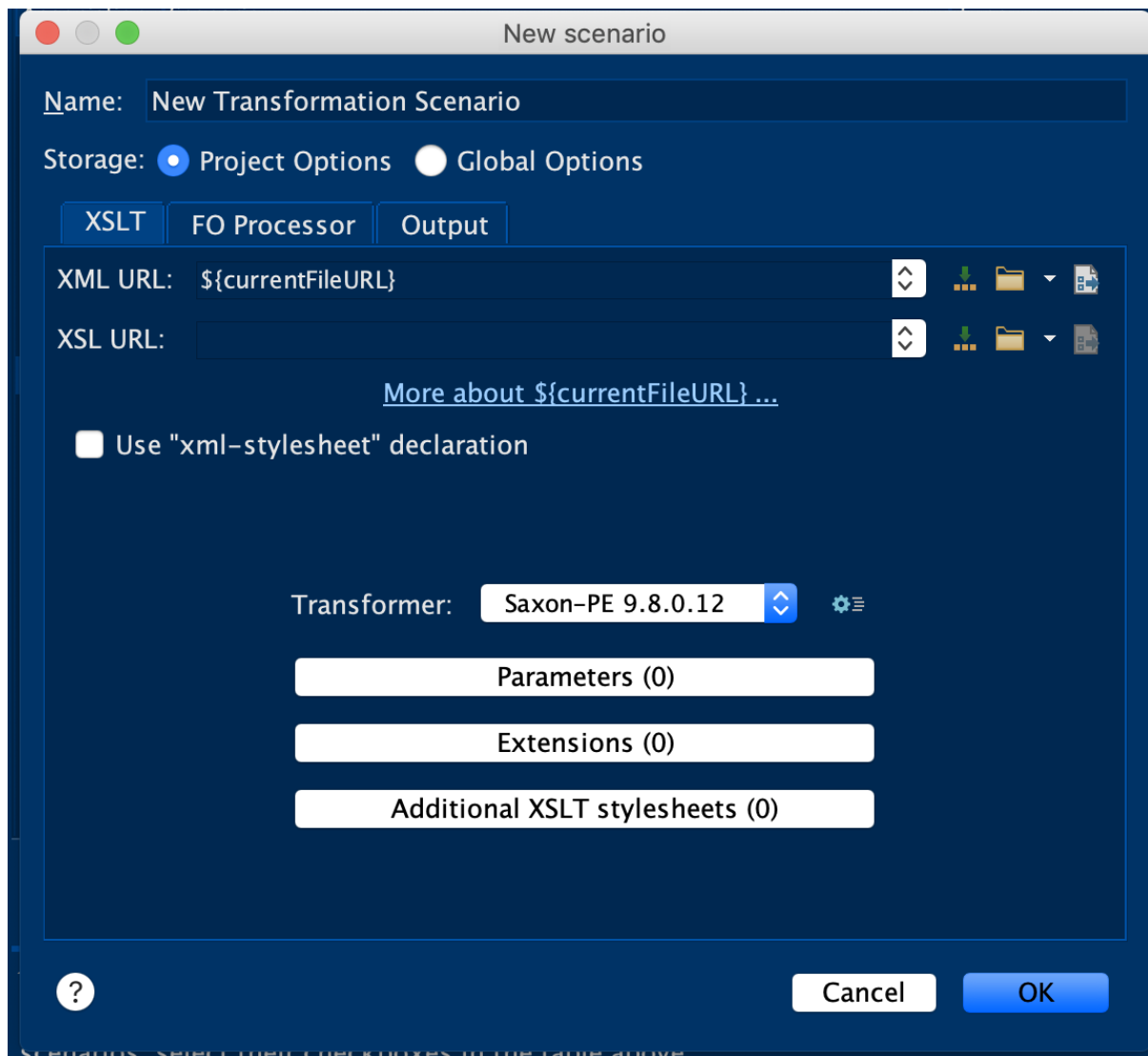
1: In Oxygen XML, go to Document > Transformation > Configure Transformation Scenario(s).



2: In the window that opens, click the 'New' button and choose 'XML transformation with XSLT'.



3. The following window opens. You can give your transformation scenario a name or leave it as the name of your file (this is the default).

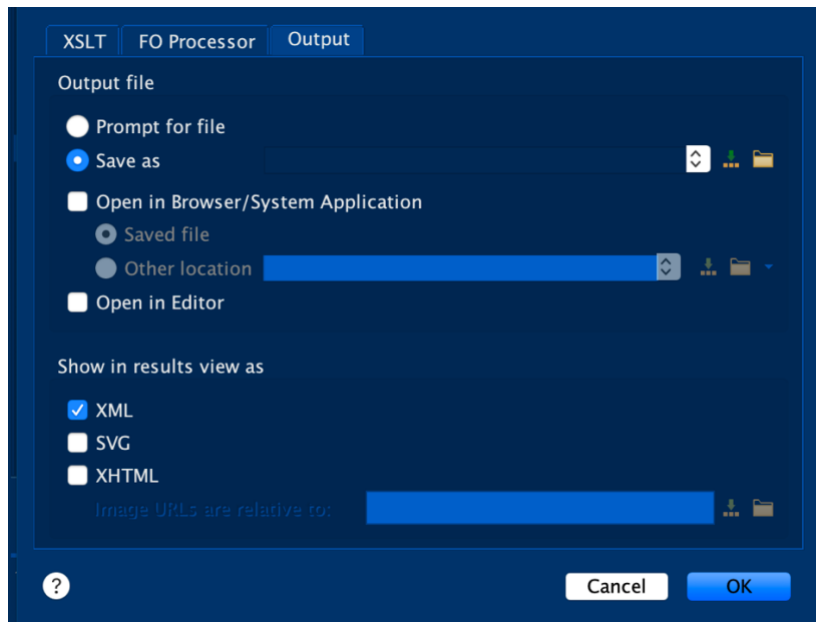


4. In the XSL URL field, copy-paste the following URL:

<https://raw.githubusercontent.com/msDesc/consolidated-tei-schema/master/preview.xsl>

You can also find the URL in step 8 of the msDesc guide.

5. Click the Output tab.



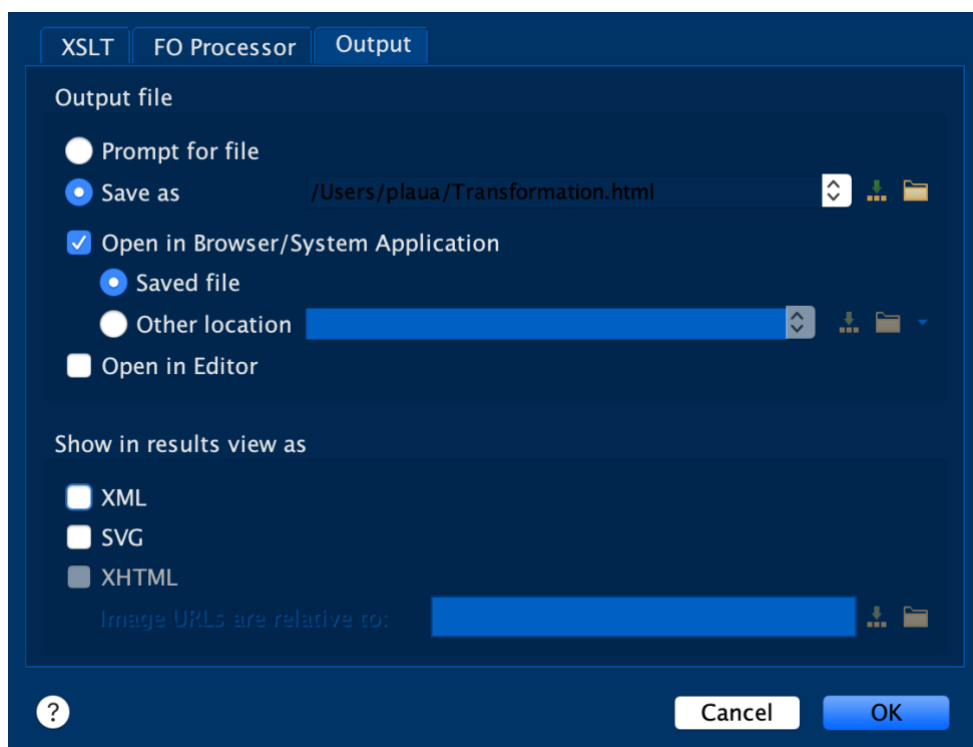
6. Click the folder icon next to the 'Save as' field and specify where you want the output HTML file to be saved. The desktop is always a good choice.

7. Take special care that you give a file name for the output that ends with '.html'!

This is crucial – it won't work without a .html ending.

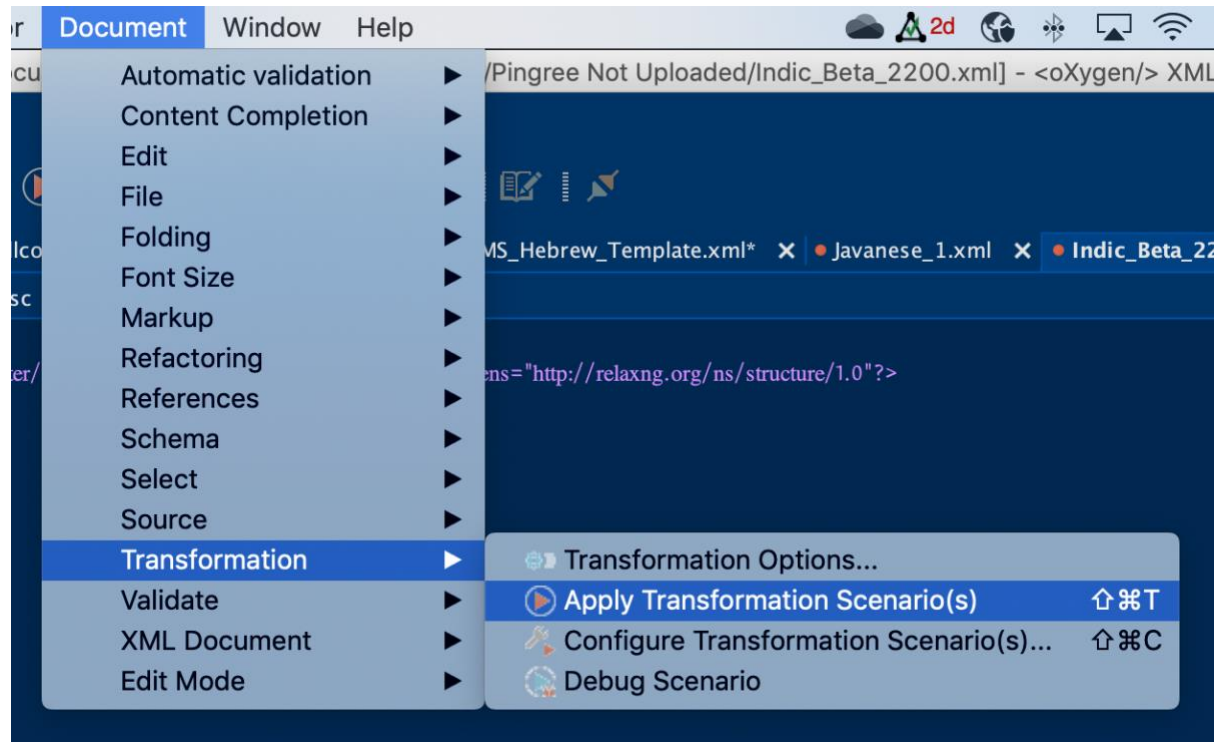
8. Tick the 'Open in Browser/System Application' box.

9. Untick all the 'Show in results view as' boxes. It should look something like this now:



10. Click 'OK', then Save and Close.

11. To run the transformation, go to Document > Transformation > Apply Transformation Scenario(s).



12. Sit back and watch!

NOTE: You can do this once and then use it for every file. Just go back to Configure Transformation Scenario(s) and tick the box for the scenario you've made. You can also go looking for other XSLT scenarios to apply or experiment with writing your own.

Glossary

Colophon

Metadata entered into the manuscript itself by a scribe or owner or similar.

Incipit

The opening lines of a manuscript or an item within a manuscript.

Explicit

The closing lines of a manuscript or an item within a manuscript.

Recto

The right-hand side of a manuscript folio.

Verso

The left-hand side of a manuscript folio.