

Oxford University Computing Services

TEI for OUCS: An Introduction to XML  
markup



## Contents

<b>1</b>	<b>The Structure of a TEI Text</b>	<b>1</b>
<b>2</b>	<b>Encoding the Body</b>	<b>1</b>
2.1	Text Division Elements	1
2.2	Headings and Closings	2
2.3	Marking Highlighted Phrases	2
2.3.1	Changes of Typeface, etc.	2
2.3.2	Notes	3
2.3.3	Cross References and Links	3
2.3.3.1	Simple Cross References	4
2.3.3.2	Extended Pointers	5
2.3.3.3	Linking Attributes	5
2.3.4	Addresses	5
2.4	Lists	6
2.5	Bibliographic Citations	7
2.6	Tables	8
2.7	Figures and Graphics	9
<b>3</b>	<b>Generated Divisions</b>	<b>9</b>
<b>4</b>	<b>Front and Back Matter</b>	<b>10</b>
4.1	Front Matter	10
4.1.1	Title Page	10
4.2	Back Matter	11
<b>5</b>	<b>The TEI header</b>	<b>11</b>
	<b>List of Elements Described</b>	<b>13</b>
	Global Attributes	13
	Elements in TEI OUCS	13

**Author:** Sebastian Rahtz

**Revision History:** May 2001

© Oxford University Computing Services

Although formal copyright is reserved, members of academic institutions may republish the material in the document subject to due acknowledgement of the source.

This document, derived from the guidelines for *TEI Lite*, provides an introduction to the recommendations of the Text Encoding Initiative (TEI), by describing a subset to, and extension of, the full TEI encoding scheme developed for marking up OUCS web pages and course documentation. It is fully compatible with the TEI as defined by TEI document P3, *Guidelines for Electronic Text Encoding and Interchange*, published in Chicago and Oxford in May 1994. See <http://www.hcu.ox.ac.uk/TEI/> for full details of the TEI and the Guidelines.

## 1 The Structure of a TEI Text

All TEI-conformant texts contain (a) a *TEI header* (marked up as a `<teiHeader>` element) and (b) the text proper (marked up as a `<text>` element).

The TEI header provides information analogous to that provided by the title page of a printed text. It has up to four parts: a bibliographic description of the machine-readable text, a description of the way it has been encoded, a non-bibliographic description of the text (a *text profile*), and a revision history. The header is described in more detail in section 5, *The TEI header*.

A TEI text will be encoded using an overall structure like this:

```
<TEI.2> <teiHeader> [ TEI
Header information ] </teiHeader>
  <text>
    <front> [ front matter ... ] </front>
    <body> [ body of text ... ] </body>
    <back> [ back matter ... ] </back>
  </text>
</TEI.2>
```

In the remainder of this document, we discuss available text structures. The discussion in each case consists of a short list of relevant TEI *elements* with a brief definition of each, followed by definitions for any *attributes* specific to that element. In most cases, short examples are also given.

## 2 Encoding the Body

As indicated above, a simple TEI document at the textual level consists of the following elements:

- `<front>` contains any prefatory matter (headers, title page, prefaces, dedications, etc.) found before the start of a text proper.
- `<body>` contains the whole body of a single unitary text, excluding any front or back matter.
- `<back>` contains any appendixes, etc., following the main part of a text.

Elements specific to front and back matter are described below in section 4, *Front and Back Matter*. In this section we discuss the elements making up the body of a text.

### 2.1 Text Division Elements

The body of a prose text may be just a series of paragraphs, or these paragraphs may be grouped together into chapters, sections, subsections, etc. In the former case, each paragraph is tagged using the `<p>` tag. In the latter case, the `<body>` may be divided into a series of `<div>` elements, which may be further subdivided, as discussed below:

- `<p>` marks paragraphs in prose.
- `<div>` contains a subdivision of the front, body, or back of a text.

When structural divisions smaller than a `<div>` are necessary, inner `<div>` elements may be used, without limit to the depth of nesting.

All these *division* elements take the following three attributes:

- type This indicates the conventional name for this category of text division. Its value might be something like “Preface”.
- id This specifies a unique identifier for the division, which may be used for cross references or other links to it, such as a commentary, as further discussed in section 2.3.3.1, *Simple Cross References*. It is often useful to provide an id attribute for every major structural unit in a text, and to derive the ID values in some systematic way, for example by appending a section number to a short code for the title of the work in question, as in the examples below.
- n The n attribute specifies a mnemonic short name or number for the division, which can be used to identify it in preference to the ID. If a conventional form of reference or abbreviation for the parts of a work already exists (such as the book/chapter/verse pattern of Biblical citations), the n attribute is the place to record it.

The attributes id and n, indeed, are so widely useful that they are allowed on any element in any TEI DTD: they are *global attributes*. Other global attributes defined in the TEI scheme are discussed in section 2.3.3.3, *Linking Attributes*.

The value of every id attribute must be unique within a document. They may be used to derive the names of HTML pages, so giving sensible mnemonic names is a good idea.

## 2.2 Headings and Closings

Every <div> may have a title or heading at its start, and (less commonly) a closing such as “End of Chapter 1”. The following elements may be used to mark them up:

- <head> contains any heading, for example, the title of a section, or the heading of a list or glossary.
- <trailer> contains a closing title or footer appearing at the end of a division of a text.

## 2.3 Marking Highlighted Phrases

### 2.3.1 Changes of Typeface, etc.

Highlighted words or phrases are those made visibly different from the rest of the text, typically by a change of type font, handwriting style, or ink color, intended to draw the reader’s attention to them.

The global rend attribute can be attached to any element, and used wherever necessary to specify details of the highlighting used for it. Of course, this must be also be explained to rendering software.

- <hi> marks a word or phrase as graphically distinct from the surrounding text, for reasons concerning which no claim is made.

Alternatively, where the cause for the highlighting can be identified with confidence, a number of other, more specific, elements are available. **All but the first two are extensions to the standard TEI markup.**

- <emph> marks words or phrases which are stressed or emphasized for linguistic or rhetorical effect eg *like this*
- <term> contains a single-word, multi-word or symbolic designation which is regarded as a technical term like *this*
- <gi> ! (*This is an extension to the standard TEI markup*) An SGML, XML or HTML element name eg <h1>
- <Key> ! (*This is an extension to the standard TEI markup*) A key to press eg <R>

<Button> ! *(This is an extension to the standard TEI markup)* A button which a user can see eg Logout

<Menu> ! *(This is an extension to the standard TEI markup)* A menu item eg [Save as]

<Input> ! *(This is an extension to the standard TEI markup)* Text for a user to type eg quota

<Prompt> ! *(This is an extension to the standard TEI markup)* A prompt from the computer eg password:

<Output> ! *(This is an extension to the standard TEI markup)* What comes back when you give a command eg job completed

<Link> ! *(This is an extension to the standard TEI markup)* The text of a link which is being described eg IT Information

<Field> ! *(This is an extension to the standard TEI markup)* A labelled input field eg Subject

<Filespec> ! *(This is an extension to the standard TEI markup)* A file or directory specification of any kind eg /usr/local/bin or C:\Windows\My Documents

<Keyword> ! *(This is an extension to the standard TEI markup)* A keyword in some technical code the user is being asked to write eg font-family

<Value> ! *(This is an extension to the standard TEI markup)* A possible value for some option eg Times-Roman 10pt

<Code> ! *(This is an extension to the standard TEI markup)* Some sort of computer language code eg `\textbf{a}$^34$`

If you ever really need it, the <lb> element marks the start of a new (typographic) line.

### 2.3.2 Notes

All notes, whether to be printed as footnotes, endnotes, marginalia, or elsewhere, should be marked using the same element:

<note> contains a note or annotation. Attributes include:

- type describes the type of note.
- resp indicates who is responsible for the annotation: author, editor, translator, etc. The value might be author, editor, etc., or the initials of the individual who added the annotation.
- place indicates where the note appears in the source text. Sample values include inline, interlinear, left, right, foot, and end, for notes which appear as marked paragraphs in the body of the text, between the lines, in the left or right margin, at the foot of the page, or at the end of the chapter or volume, respectively.
- target indicates the point of attachment of a note, or the beginning of the span to which the note is attached.
- targetEnd points to the end of the span to which the note is attached, if the note is not embedded in the text at that point.
- anchored indicates whether the copy text shows the exact place of reference for the note.

Where possible, the body of a note should be inserted in the text at the point at which its identifier or mark first appears. This may not be possible for example with marginalia, which may not be anchored to an exact location. For simplicity, it may be adequate to position marginal notes before the relevant paragraph or other element. Notes may also be placed in a separate division of the text (as end-notes are, in printed books) and linked to the relevant portion of the text using their target attribute.

The n attribute may be used to supply the number or identifier of a note if this is required.

### 2.3.3 Cross References and Links

Explicit cross references or links from one point in a text to another in the same XML document may be encoded using the elements described in section 2.3.3.1, *Simple Cross References*. References or links to elements of some other XML document, or to parts of non-XML documents, may be encoded using the *TEI extended pointers* described in section 2.3.3.2, *Extended Pointers*.

#### 2.3.3.1 Simple Cross References

A cross reference from one point within a single document to another can be encoded using either of the following elements:

- `<ref>` a reference to another location in the current document, in terms of one or more identifiable elements, possibly modified by additional text or comment.
- `<ptr>` a pointer to another location in the current document in terms of one or more identifiable elements.

These elements share the following attributes:

- `target` specifies the destination of the pointer as one or more SGML identifiers
- `type` categorizes the pointer in some respect, using any convenient set of categories.
- `targType` specifies the type (or types) of element to which this pointer may point.
- `crDate` specifies when this pointer was made.
- `resp` specifies the creator of the pointer.

The difference between these two elements is that `<ptr>` is an empty element, simply marking a point from which a link is to be made, whereas `<ref>` may contain some text as well — typically the text of the cross-reference itself. The `<ptr>` element would be used for a cross reference which is to be indicated by some non-verbal means such as a symbol or icon, or in an electronic text by a button. It is also useful in document production systems, where the formatter can generate the correct verbal form of the cross reference.

The following two forms, for example, are logically equivalent (assuming we have documented somewhere the exact verbal form of cross references represented by `<ptr>` elements):

```
See especially <ref target="SEC12">section 12 on page
34</ref>.

See especially <ptr
target="SEC12"/>.
```

The value of the `target` attribute must be an XML identifier in the current XML document. This implies that the passage or phrase being pointed at must bear an identifier, and must therefore be tagged as an element of some kind. In the following example, the cross reference is to a `<div>` element:

```
... see especially <ptr
target="SEC12"/>. ... <div id="SEC12"><head>Concerning
Identifiers... ...
```

Because the `id` attribute is global, any element in a document may be pointed to in this way. In the following example, a paragraph has been given an identifier so that it may be pointed at:

```
... this is discussed in <ref
target="pspec">the paragraph on links</ref> ... <p id="pspec">Links
may be made to any kind of element ...
```

The `targType` attribute can be used to specify that the element pointed to must be of a particular type, as in the following example:

```
... this is discussed in
<ref target="dspec" targType='div div'> the section on
links</ref>
```

This reference should fail if the element with identifier `dspec` is not a `<div>`. Note however that this check cannot be carried out by an XML parser alone, since the XML parser can only check that some element `dspec` exists.

The `type` attribute can be used to categorize the link represented by the pointer in any convenient way. The `resp` and `crDate` attributes may also be used to represent the person or agency responsible for making the link, and its date of creation, as in the following example:

```
...
this is discussed in
<ref type="xref" resp="auto" crdate="950521" target="dspec" targType='div div'>
the section on links</ref>
```

These attributes are most likely to be of use in hypertext systems containing very many pointers used for a variety of purposes and created by a variety of means.

Sometimes the target of a cross reference does not correspond with any particular feature of a text, and so may not be tagged as an element of some kind. If the desired target is simply a point in the current document, the easiest way to mark it is by introducing an `<anchor>` element at the appropriate spot.

### 2.3.3.2 Extended Pointers

The elements `<ptr>` and `<ref>` can only be used for cross-references or links whose targets occur within the same XML document as their source. They can also refer only to XML elements. The elements discussed in this section are not restricted in this way.

- `<xptr>` defines a pointer to another location in the current document or an external document.
- `<xref>` defines a pointer to another location in the current document or an external document, possibly modified by additional text or comment.

In addition to the pointer attributes already discussed in section 2.3.3.1, *Simple Cross References* above, these elements share the following additional attributes, which can be used to specify the target of the cross reference or link in place of the `target` attribute, although the full details may not be supported in local renditions:

- `doc` specifies the document within which the required location is to be found, by default the current document.
- `from` specifies the start of the destination of the pointer as an expression in the TEI extended pointer syntax, by default the whole of the document indicated by the `doc` attribute.
- `to` specifies the endpoint of the destination of the pointer as an expression in the TEI extended pointer syntax; may only be specified if the `from` attribute has been.
- `url` ! (*This is an extension to the standard TEI markup*) A Web URL specifying the destination

### 2.3.3.3 Linking Attributes

The following special purpose *linking* attributes are defined for every element in the TEI OUCS DTD:

- `ana` links an element with its interpretation.
- `corresp` links an element with one or more other corresponding elements.
- `next` links an element to the next element in an aggregate.
- `prev` links an element to the previous element in an aggregate.

## 2.3.4 Addresses

The `<address>` element is used to mark a postal address of any kind. It contains one or more `<addrLine>` elements, one for each line of the address.

`address` contains a postal or other address, for example of a publisher, an organization, or an individual.

`addrLine` contains one line of a postal or other address.

Here is a simple example:

```
<address>
<addrLine>Computer Center (M/C 135)</addrLine>
<addrLine>1940 W. Taylor, Room 124</addrLine>
<addrLine>Chicago, IL 60612-7352</addrLine>
<addrLine>U.S.A.</addrLine>
</address>
```

## 2.4 Lists

The element `<list>` is used to mark any kind of *list*. A list is a sequence of text items, which may be ordered, unordered, or a glossary list. Each item may be preceded by an item label (in a glossary list, this label is the term being defined):

`<list>` contains any sequence of items organized as a list. Attributes include:

`type` describes the form of the list. Suggested values include: `ordered`, `bulleted` (for lists with numbered or lettered items, and lists with bullet-marked items, respectively), `gloss` (for lists consisting of a set of technical terms, each marked with a `<label>` element and accompanied by a gloss or definition marked as an `<item>`), and `simple` (for lists with items not marked with number or bullets).

`<item>` contains one component of a list.

`<label>` contains the label associated with an item in a list; in glossaries, marks the term being defined.

Individual list items are tagged with `<item>`. The first `<item>` may optionally be preceded by a `<head>`, which gives a heading for the list. The numbering of a list may be omitted (if reconstructible), indicated using the `n` attribute on each item, or (rarely) tagged as content using the `<label>` element. The following are all thus equivalent:

```
<list>
<head>A short list</head>
<item>First item in list.</item>
<item>Second item in list.</item>
<item>Third item in list.</item>
</list>

<list>
<head>A short list</head>
<item n=1>First item in list.</item>
<item n=2>Second item in list.</item>
<item n=3>Third item in list.</item>
</list>

<list>
<head>A short list</head>
<label>1</label><item>First item in list.</item>
<label>2</label><item>Second item in list.</item>
<label>3</label><item>Third item in list.</item>
</list>
```

The styles should not be mixed in the same list.

A simple two-column table may be treated as a *glossary list*, tagged `<list type=gloss>`. Here, each item comprises a *term* and a *gloss*, marked with `<label>` and `<item>` respectively. These correspond to the elements `<term>` and `<gloss>`, which can occur anywhere in prose text.

```

<list type="gloss">
<head>Vocabulary</head>
<label lang="enm">nu</label>      <item>now</item>
<label lang="enm">lhude</label>   <item>loudly</item>
<label lang="enm">bloweth</label> <item>blooms</item>
<label lang="enm">med</label>     <item>meadow</item>
<label lang="enm">wude</label>    <item>wood</item>
<label lang="enm">awe</label>     <item>ewe</item>
<label lang="enm">lhouth</label>  <item>lows</item>
<label lang="enm">sterteth</label> <item>bounds, frisks</item>
<label lang="enm">verteth</label> <item lang="lat">pedit</item>
<label lang="enm">murie</label>   <item>merrily</item>
<label lang="enm">swik</label>    <item>cease</item>
<label lang="enm">naver</label>  <item>never</item>
</list>

```

Where the internal structure of a list item is more complex, it may be preferable to regard the list as a *table*, for which special-purpose tagging is defined in an additional TEI tag set.

Lists of whatever kind can, of course, nest within list items to any depth required. Here, for example, a glossary list contains two items, each of which is itself a simple list:

```

<list type="gloss"><label>EVIL</label>
<item><list type="simple">
  <item>I am cast upon a horrible desolate island, void
    of all hope of recovery.</item>
  <item>I am singled out and separated as it were from
    all the world to be miserable.</item>
  <item>I am divided from mankind &mdash; a solitaire; one
    banished from human society.</item>
</list> <!-- end of first nested list --></item>
<label>GOOD</label>
<item><list type="simple">
  <item>But I am alive; and not drowned, as all my
    ship's company were.</item>
  <item>But I am singled out, too, from all the ship's
    crew, to be spared from death...</item>
  <item>But I am not starved, and perishing on a barren place,
    affording no sustenances...</item>
</list><!-- end of second nested list --></item>
</list><!-- end of glossary list -->

```

Lists of bibliographic items should be tagged using the `<listBibl>` element, described in the next section.

## 2.5 Bibliographic Citations

It is often useful to distinguish bibliographic citations where they occur within texts being transcribed for research, if only so that they will be properly formatted when the text is printed out. The element `<bibl>` is provided for this purpose:

`<bibl>` contains a loosely-structured bibliographic citation of which the sub-components may or may not be explicitly tagged.

Where the components of a bibliographic reference are to be distinguished, the following elements may be used as appropriate. It is generally useful to mark at least those parts (such as the titles of articles, books, and journals) which will need special formatting. The other elements are provided for cases where particular interest attaches to such details.

`<author>` in a bibliographic reference, contains the name of the author(s), personal or corporate, of a work; the primary *statement of responsibility* for any bibliographic item.

`<biblScope>` defines the scope of a bibliographic reference, for example as a list of page numbers, or a named subdivision of a larger work.

- `<date>` contains a date in any format.
- `<editor>` secondary *statement of responsibility* for a bibliographic item, for example the name of an individual, institution or organization, (or of several such) acting as editor, compiler, translator, etc. Attributes include:
  - `role` specifies the nature of the intellectual responsibility. Sample values include `translator`, `compiler`, `illustrator`, etc.; the default value is `editor`.
- `<imprint>` groups information relating to the publication or distribution of a bibliographic item.
- `<publisher>` provides the name of the organization responsible for the publication or distribution of a bibliographic item.
- `<pubPlace>` contains the name of the place where a bibliographic item was published.
- `<series>` contains information about the series in which a book or other bibliographic item has appeared.
- `<title>` contains the title of a work, whether article, book, journal, or series, including any alternative titles or subtitles. Attributes include
  - `type` categorizes the title in some way, for example as a main, subordinate, etc.
  - `level` indicates the bibliographic *level* or class of title. Legal values are described in section 2.3.1, *Changes of Typeface, etc.*

For example, the following editorial note might be transcribed as shown:

He was a member of Parliament for Warwickshire in 1445, and died March 14, 1470 (according to Kittredge, *Harvard Studies* 5. 88ff).

```
He was a member of Parliament for Warwickshire in 1445, and died
March 14, 1470 (according to <bibl><author>Kittredge</author>,
<title>Harvard Studies</title> <biblScope>5. 88ff</biblScope></bibl>).
```

For lists of bibliographic citations, the `<listBibl>` element should be used; it may contain a series of `<bibl>` elements.

## 2.6 Tables

Tables represent a sizable challenge for any text processing system, but simple tables, at least, appear in so many texts that even in the simplified TEI tag set presented here, markup for tables is necessary. The following elements are provided for this purpose:

- `<table>` contains text displayed in tabular form, in rows and columns. Attributes include:
  - `rows` indicates the number of rows in the table.
  - `cols` indicates the number of columns in each row of the table.
- `<row>` contains one row of a table. Attributes include:
  - `role` indicates the kind of information held in the cells of this row. Suggested values include `label` for labels or descriptive information, and `data` for actual data values.
- `<cell>` contains one cell of a table. Attributes include:
  - `role` indicates the kind of information held in the cell. Suggested values include `label` for labels or descriptive information, and `data` for actual data values.
  - `cols` indicates the number of columns occupied by this cell.
  - `rows` indicates the number of rows occupied by this cell.

! (*This is an extension to the standard TEI markup*) The `<table>` element can also take the `align`, `summary`, `width`, `border`, `frame`, `rules`, `cellspacing` and `cellpadding` attributes defined in HTML, and the conversion to HTML will pass them straight through.

As an example, Defoe uses mortality tables like the following in the *Journal of the Plague Year* to show the rise and ebb of the epidemic:

```
<p>It was indeed coming on amain, for the burials that
same week were in the next adjoining parishes thus:&mdash;
<table rows="5" cols="4">
<row role='data'>
<cell role='label'>St. Leonard's, Shoreditch</cell>
  <cell>64</cell> <cell>84</cell> <cell>119</cell></row>
<cell role='label'>St. Botolph's, Bishopsgate</row>
  <cell>65</cell> <cell>105</cell> <cell>116</cell></row>
<cell role='label'>St. Giles's, Cripplegate</row>
  <cell>213</cell> <cell>421</cell> <cell>554</cell></row>
</table>
<p>This shutting up of houses was at first counted a very cruel
and unchristian method, and the poor people so confined made
bitter lamentations. ... </p>
```

## 2.7 Figures and Graphics

Not all the components of a document are necessarily textual. The most straightforward text will often contain diagrams or illustrations, to say nothing of documents in which image and text are inextricably intertwined, or electronic resources in which the two are complementary.

`<figure>` marks the spot at which a graphic is to be inserted in a document.

Attributes include:

`entity` the name of a pre-defined system entity containing a digitized version of the graphic to be inserted.

`file` ! (*This is an extension to the standard TEI markup*) The simple name of a graphics file

`width` ! (*This is an extension to the standard TEI markup*) The width to which the graphic should be scaled

`height` ! (*This is an extension to the standard TEI markup*) The height to which the graphic should be scaled

`scale` ! (*This is an extension to the standard TEI markup*) The extent which the graphic should be scaled (eg 0.5)

`<figDesc>` contains a textual description of the appearance or content of a graphic, for use when documenting an image without displaying it.

Any textual information accompanying the graphic, such as a heading and/or caption, may be included within the `<figure>` element itself, in a `<head>` and one or more `<p>` elements, as may also any text appearing within the graphic itself. It is strongly recommended that a prose description of the image be supplied, as the content of a `<figDesc>` element, for the use of applications which are not able to render the graphic, and to render the document accessible to vision-impaired readers. (Such text is not normally considered part of the document proper.)

Usually, a graphic will have at the least an identifying title, which should be encoded using the `<head>` element. It is also often convenient to include a brief description of the image using `<figDesc>`. The filename of the image at the appropriate point in the document is supplied using the `file` attribute of the `<figure>` element:

```
<figure file="fezziPic.png">
<head>Mr Fezziwig's Ball</head>
<figDesc>A Cruikshank engraving showing Mr Fezziwig leading
  a group of revellers.</figDesc>
</figure>
```

## 3 Generated Divisions

Most modern document production systems have the ability to generate automatically whole sections such as a table of contents or an index. The TEI OUCS scheme provides an element to mark the location at which such a generated section should be placed.

`<divGen>` indicates the location at which a textual division generated automatically by a text-processing application is to appear. Attributes include:

`type` specifies what type of generated text division (e.g. index, table of contents, etc.) is to appear. Sample values include: `index` (an index is to be generated and inserted at this point), `toc` (a table of contents) `figlist` (a list of figures) `tablist` (a list of tables).

The `<divGen>` element can be placed anywhere that a division element would be legal, as in the following example:

```
<front>
<titlePage> ... </titlePage>
<divGen type="toc"/>
<div type='Preface'><head>Preface</head> ... </div>
</front>
<body> ... </body>
<back>
<div><head>Appendix</head> ... </div>
<divGen type="index" n='Index' />
</back>
```

This example also demonstrates the use of the `type` attribute to distinguish the different kinds of division to be generated: in the first case a table of contents (a `toc`) and in the second an index.

When an existing index or table of contents is to be encoded (rather than one being generated) for some reason, the `<list>` element discussed in section 2.4, *Lists* should be used.

## 4 Front and Back Matter

### 4.1 Front Matter

For many purposes, particularly in older texts, the preliminary material such as title pages, prefatory epistles, etc., may provide very useful additional linguistic or social information. P3 provides a set of recommendations for distinguishing the textual elements most commonly encountered in front matter, which are summarized here.

#### 4.1.1 Title Page

The start of a title page should be marked with the element `<titlePage>`. All text contained on the page should be transcribed and tagged with the appropriate element from the following list:

- `<titlePage>` contains the title page of a text, appearing within the front or back matter.
- `<docTitle>` contains the title of a document, including all its constituents, as given on a title page. Must be divided into `<titlePart>` elements.
- `<titlePart>` contains a subsection or division of the title of a work, as indicated on a title page; also used for free-floating fragments of the title page not part of the document title, authorship attribution, etc. Attributes include:
  - `type` specifies the role of this subdivision of the title. Suggested values include: `main` (main title), `sub` (subtitle), `desc` (a descriptive paraphrase of the work included in the title), and `alt` (alternative title).

- `<docAuthor>` contains the name of the author of the document, as given on the title page (often but not always contained in a `<byline>`).
- `<docDate>` contains the date of the document, as given (usually) on the title page.
- `<docEdition>` contains an edition statement as presented on a title page of a document.
- `<docImprint>` contains the imprint statement (place and date of publication, publisher name), as given (usually) at the foot of a title page.

## 4.2 Back Matter

Because of variations in publishing practice, back matter can contain virtually any of the elements listed above for front matter, and the same elements should be used where this is so. Additionally, back matter may contain the following types of matter within the `<back>` element. Like the structural divisions of the body, these should be marked as `<div>` elements, and distinguished by the following suggested values of the `type` attribute:

- `appendix` an appendix.
- `glossary` a list of words and definitions, typically in the form of a `<list type=gloss>`.
- `bibliography` a series of bibliographic references, typically in the form of a special bibliographic-list element `<listBibl>`, whose items are individual `<bibl>` elements.
- `index` a set of index entries, possibly represented as a structured list or glossary list, with optional leading `<head>` and perhaps some paragraphs of introductory or closing text.

## 5 The TEI header

The *TEI Header* is a complex and powerful construct, which can be used to store a great deal of information about a text. You can read a detailed explanation in the TEI Guidelines, or a shorter essay in the TEI Lite description (see <http://www.hcu.ox.ac.uk/TEI/Lite/>). For immediate purposes, the following is a useful template to follow:

```
<teiHeader>
  <fileDesc>
    <titleStmt>
      <title>Markup for OUCS documents</title>
      <author>Sebastian Rahtz</author>
    </titleStmt>
    <editionStmt>
      <edition>
        <date>October 2000</date>
      </edition>
    </editionStmt>
    <publicationStmt>
      <authority>OUCS</authority>
      <address>
        <email>advisory@oucs.ox.ac.uk</email>
      </address>
    </publicationStmt>
    <sourceDesc>
      <p>This is the master version of an original document.</p>
    </sourceDesc>
  </fileDesc>

  <revisionDesc>
    <change>
      <date>Date: 2001/05/08 $</date>
      <respStmt>
```

## TEI for OUCS: An Introduction to XML markup

```
        <name>$Author: rahtz $</name>
    </respStmt>
    <item>$Revision: #2 $</item>
</change>
</revisionDesc>
</teiHeader>
```

Note that the \$ keywords are expanded by the change management system and replaced with current information.

## List of Elements Described

### Global Attributes

All elements in the TEI Lite document type definition have the following global attributes:

- `ana` links an element with its interpretation.
- `corresp` links an element with one or more other corresponding elements.
- `id` Unique identifier for the element; must begin with a letter, can contain letters, digits, hyphens, and periods.
- `lang` language of the text in this element; if not specified, language is assumed to be the same as in the surrounding context.
- `n` Name or number of this element; may be any string of characters. Often used for recording traditional reference systems.
- `next` links an element to the next element in an aggregate.
- `prev` links an element to the previous element in an aggregate.
- `rend` physical realization of the element in the copy text: *italic*, `roman`, `display block`, etc. Value may be any string of characters.

### Elements in TEI OUCS

The following list shows all the elements defined for the TEI OUCS DTD, with a brief description of each:

- `<Button>` A button which a user can see
- `<Code>` Some sort of computer language code eg `\textbf{a}`<sup>34</sup>
- `<Field>` A labelled input field
- `<Filespec>` A file or directory specification of any kind
- `<Input>` Text for a user to type
- `<Key>` A key to press
- `<Keyword>` A keyword in some technical code the user is being asked to write eg `font-family`
- `<Link>` The text of a link which is being described
- `<Menu>` A menu item
- `<Output>` What comes back when you give a command eg `job completed`
- `<Program>` A verbatim listing of a computer program or other file
- `<Prompt>` A prompt from the computer
- `<Screen>` A verbatim listing of the output from a computer program
- `<Value>` A possible value for some option
- `<abbr>` contains an abbreviation of any sort; expansion may be given in the `expan` attribute.
- `<address>` contains a postal or other address, for example of a publisher, an organization, or an individual.
- `<addrLine>` contains one line of a postal or other address.
- `<anchor>` specifies a location or point within a document so that it may be pointed to.
- `<author>` in a bibliographic reference, contains the name of the author(s), personal or corporate, of a work; the primary *statement of responsibility* for any bibliographic item.
- `<authority>` supplies the name of a person or other agency responsible for making an electronic file available, other than a publisher or distributor.
- `<availability>` supplies information about the availability of a text, for example any restrictions on its use or distribution, its copyright status, etc.
- `<back>` contains any appendixes, etc., following the main part of a text.

- <bibl> contains a loosely-structured bibliographic citation of which the sub-components may or may not be explicitly tagged.
- <biblFull> contains a fully-structured bibliographic citation, in which all components of the TEI file description are present.
- <biblScope> defines the scope of a bibliographic reference, for example as a list of page numbers, or a named subdivision of a larger work.
- <body> contains the whole body of a single unitary text, excluding any front or back matter.
- <cell> contains one cell of a table.
- <date> contains a date in any format, with normalized value in the *value* attribute.
- <dateline> contains a brief description of the place, date, time, etc., of production of a letter, newspaper story, or other work, prefixed or suffixed to it as a kind of heading or trailer.
- <div> contains a subdivision of the front, body, or back of a text.
- <divGen> indicates the location at which a textual division generated automatically by a text-processing application is to appear; the *type* attribute specifies whether it is an index, table of contents, or something else.
- <docAuthor> contains the name of the author of the document, as given on the title page (often but not always contained in a <byline>).
- <docDate> contains the date of the document, as given (usually) on the title page.
- <docEdition> contains an edition statement as presented on a title page of a document.
- <docImprint> contains the imprint statement (place and date of publication, publisher name), as given (usually) at the foot of a title page.
- <docTitle> contains the title of a document, including all its constituents, as given on a title page. Must be divided into <titlePart> elements.
- <edition> describes the particularities of one edition of a text.
- <editionStmt> groups information relating to one edition of a text.
- <editor> secondary *statement of responsibility* for a bibliographic item, for example the name of an individual, institution or organization, (or of several such) acting as editor, compiler, translator, etc.
- <editorialDecl> provides details of editorial principles and practices applied during the encoding of a text.
- <eg> contains a single short example of some technical topic being discussed, e.g. a code fragment or a sample of SGML encoding.
- <emph> marks words or phrases which are stressed or emphasized for linguistic or rhetorical effect.
- <encodingDesc> documents the relationship between an electronic text and the source or sources from which it was derived.
- <figure> marks the spot at which a graphic is to be inserted in a document. Attributes may be used to indicate a file name, or an SGML entity containing the image itself; paragraphs within the <figure> element may be used to provide captions.
- <fileDesc> contains a full bibliographic description of an electronic file.
- <front> contains any prefatory matter (headers, title page, prefaces, dedications, etc.) found before the start of a text proper.
- <gi> contains a special type of identifier: an SGML generic identifier, or element name.
- <gloss> marks a word or phrase which provides a gloss or definition for some other word or phrase.
- <head> contains any heading, for example, the title of a section, or the heading of a list or glossary.

- <hi> marks a word or phrase as graphically distinct from the surrounding text, for reasons concerning which no claim is made.
- <idno> supplies any standard or non-standard number used to identify a bibliographic item; the `type` attribute identifies the scheme or standard.
- <imprint> groups information relating to the publication or distribution of a bibliographic item.
- <item> contains one component of a list.
- <keywords> contains a list of keywords or phrases identifying the topic or nature of a text; if the keywords come from a controlled vocabulary, it can be identified by the `scheme` attribute.
- <label> contains the label associated with an item in a list; in glossaries, marks the term being defined.
- <li> force a new line.
- <list> contains any sequence of items organized as a list, whether of numbered, bulleted, or other type.
- <listBibl> contains a list of bibliographic citations of any kind.
- <note> contains a note or annotation, with attributes to indicate the type, location, and source of the note.
- <p> marks paragraphs in prose.
- <ptr> a pointer to another location in the current document in terms of one or more identifiable elements.
- <publicationStmt> groups information concerning the publication or distribution of an electronic or other text.
- <publisher> provides the name of the organization responsible for the publication or distribution of a bibliographic item.
- <pubPlace> contains the name of the place where a bibliographic item was published.
- <q> contains a quotation or apparent quotation.
- <ref> a reference to another location in the current document, in terms of one or more identifiable elements, possibly modified by additional text or comment.
- <refsDecl> specifies how canonical references are constructed for this text.
- <respStmt> supplies a statement of responsibility for someone responsible for the intellectual content of a text, edition, recording, or series, where the specialized elements for authors, editors, etc., do not suffice or do not apply.
- <revisionDesc> summarizes the revision history for a file.
- <row> contains one row of a table.
- <series> contains information about the series in which a book or other bibliographic item has appeared.
- <seriesStmt> groups information about the *series*, if any, to which a publication belongs.
- <sourceDesc> supplies a bibliographic description of the copy text(s) from which an electronic text was derived or generated.
- <table> contains text displayed in tabular form, in rows and columns.
- <tagsDecl> provides detailed information about the tagging applied to an SGML document.
- <tagUsage> supplies information about the usage of a specific element within the outermost <text> of a TEI conformant document.
- <term> contains a single-word, multi-word or symbolic designation which is regarded as a technical term.
- <textClass> groups information which describes the nature or topic of a text in terms of a standard classification scheme, thesaurus, etc.
- <title> contains the title of a work, whether article, book, journal, or series, including any alternative titles or subtitles.

- `<titlePage>` contains the title page of a text, appearing within the front or back matter.
- `<titlePart>` contains a subsection or division of the title of a work, as indicated on a title page; also used for free-floating fragments of the title page not part of the document title, authorship attribution, etc.
- `<titleStmt>` groups information about the title of a work and those responsible for its intellectual content.
- `<trailer>` contains a closing title or footer appearing at the end of a division of a text.
- `<xptr>` defines a pointer to another location in the current document or an external document.
- `<xref>` defines a pointer to another location in the current document or an external document, possibly modified by additional text or comment.