The Fortune Theatre Records: A Prototype Digital Edition, 
Records of Early English Drama (REED): 
White Paper

Jason Boyd and John Bradley  
with Carolyn Black, Sally-Beth MacLean, and Paul Vetch  
for the Fortune Theatre Records (FTR) Project Team\(^1\)

1. Summary

In 2012, a project team consisting of personnel from Records of Early English Drama (REED), the Department of Digital Humanities-King’s College London (DDH), the Centre for Digital Humanities at Ryerson University (CDH-RU), University of Toronto Libraries (UTL), and independent academic publisher Boydell & Brewer collaborated on “The Fortune Theatre Records: A Prototype Digital Edition for REED” project (hereafter FTR; [http://ereed.cch.kcl.ac.uk/](http://ereed.cch.kcl.ac.uk/)) in order to determine how historical materials of the kind traditionally published by REED in print could be published online. The Andrew W. Mellon Foundation provided twelve months’ funding for an essential prototyping stage where the collaborators’ initial ideas could be more formally explored in preparation for providing a solid theoretical, technical, and operational framework for the task of getting both future and past REED information online. The complex structure of REED’s volumes presents a challenge in the blending of established text markup techniques with techniques for producing highly structured data and DDH brought notable experience in dealing with structured text, structured data, and their innovative presentation through online web applications. The prototype project took a small but coherent subset of new London-area REED research materials already collected but not yet in the established editorial process and used them to develop and explore new protocols, workflows, data formats, and software that will support more ambitious work in the future. Jessica Freeman’s dramatic records relating to London’s Fortune Theatre, one of the most important early performance venues, from the forthcoming REED collection *Middlesex, including*

\(^1\) In addition to the names above, the FTR project team included Paul Caton, Trieneke Gastmeier, John Geck, Patrick Gregory, Tanya Hagen, Stephanie Hovland, Sian Meikle, Daniela Robibero, and José Miguel Monteiro Vieira.  
Westminster, therefore formed the core materials for the project.

After outlining the FTR project’s objectives – primarily, creating a prototype digital edition using a set of records relating to the London-area Fortune Theatre in order to answer a range of questions about how REED’s primary scholarly output might be reconceived in digital form (part 2) – this paper turns to a discussion of the technology (software and standards) that made up the editing platform used to create the digital edition (part 3). It then describes the work that was undertaken to enable the conversion of records text marked up in REED’s own “at-codes” schema to the XML-based schema developed by the Text Encoding Initiative (TEI), which includes necessary changes to the implementation of some of the at-codes (part 4). A description of the at-codes to TEI conversion process follows, followed by an outline of the additional TEI markup that was done on the records, primarily focusing on those features of the original documents that are currently being described using textual notes. It is suggested that there may be a number of advantages to using TEI instead of textual notes for many (while not all) original document features currently covered by textual notes (part 5). The creation of an “ontology” enabling searching of the Fortune Theatre records (built upon an analysis of REED print collection indexes), using the Entity Authority Tool Set (EATS) technology, is explained, and the process of implementing the EATS markup, including some of the challenges faced and questions raised, is described (part 6). The changes to standard REED editorial practices and workflow are then described, with a range of related issues foregrounded for the future (part 7). The two major components of the eREED platform (Kiln and Mezzanine) are described, and some future refinements regarding the navigation and display of the records are noted (part 8). The issue of data interoperability between eREED and REED’s two other digital resources – the Patrons and Performances and Early Modern London Theatres (EMLoT) databases – is discussed, both in terms of what was done to link together these resources for the prototype and how this could be further developed and refined using Linked Data specifications (part 9). A section on the potential for user-customized on-demand printing follows (part 10), and the white paper concludes with an enumeration of the issues and areas for future exploration identified during the development of the prototype (part 11). The white paper has three appendices: Appendix A describes the Subversion version
control system folder structure developed for eREED materials; Appendix B lists and describes the FTR project files available to the public on the REED website (http://www.reed.utoronto.ca/FortuneDirectory.zip); and Appendix C comprises the “Guidelines for REED at-codes Application and TEI P5 Markup.”

2. Project Objectives

The primary objective of the FTR project was to explore how REED’s primary scholarly output, currently produced as a series of print collections, could be both edited and published online and integrated with REED’s current digital resources, *Patrons and Performances* and *Early Modern London Theatres* (EMLoT). In undertaking this exploration, the project aimed to address and go some way towards answering a number of key questions: What form would a digital REED collection take and what web standards for information representation and exchange would be used? How might information in the *Patrons* database and EMLoT interoperate with information in eREED? What are the requirements of an online editing environment for the REED collections? How will this digital transformation impact on REED’s current editorial roles, tasks, and workflows? What kind of interface would enable optimal user searching, navigation, and reading of a REED digital edition? How might a user manipulate these results for the purpose of creating publishable/printable outputs?

It was decided that these questions were best explored through the development of a Prototype Digital Edition that would provide possible answers in a concrete form, which could then be analyzed and lead to a process of further refinement. Accordingly, the FTR project took a small but coherent subset of new London-area REED research materials already collected but not yet in the established editorial process – the records relating to London’s Fortune Theatre, one of the most important early performance venues from the forthcoming collection *Middlesex, including Westminster* – and used the process of preparing them as a digital edition to explore and develop protocols, workflows, data formats, and software that, in the future, can support more ambitious work.

---

3. Editorial Work: Technology

The editorial platform for the FTR Prototype was based upon standard technology already used by DDH in other projects. It recognized the existence of two kinds of editorial work that needed to be undertaken:

1. The REED records and document/source information needed to be edited in Word, then converted into TEI, and then edited in TEI.
2. The records needed to be classified in various ways.

For the work involved in group one, the project was provided with access to DDH’s Subversion (http://www.syncrosvnclient.com) server, and was supported in the use of the oXygen XML editor (http://www.oxygenxml.com). It was also provided with a few Perl scripts and XSLT stylesheets that helped with the TEI work. For the work involved in the second category, an EATS server (https://github.com/ajenhl/eats) was established and configured to support the identification of the kinds of entities that were expected to be used to categorize the records, and the oXygen EATS plugin was configured so links between the EATS entities and the REED records could be created from within the record oXygen editing environment.

Software in the REED record workflow

Some significant effort was made to explore how these tools could best fit with REED’s workflow needs. It was recognized that it would be too much to expect most of REED’s collection editors to make use of the TEI format for their record transcription work. Indeed, past experience had suggested that the use of Word supplemented with REED’s own at-code markup system had worked well at the early (pre-typesetting) stage of record preparation and could be managed by almost all of REED’s editors. Thus, it was decided to explore how Microsoft MS Word could continue to be the format in which material was submitted by the editor, and for which the initial paleographic editorial work would be carried out. At some point, the records in Word would be converted to TEI-XML, ready for use
in the larger Fortune Theatre editing environment. Once this switch occurred, the TEI-XML version of the records became the main copy against which any further editorial changes would be applied.

Thus, the processing of the REED records was split into two technological parts. The first part was carried out in MS Word and took advantage of its Track Changes tool to facilitate the interaction between various people who were working and providing feedback on the records text prior to the conversion to TEI-XML. There is more discussion about this part of the editorial process in section 7 of this paper. When this seemed to be completed, the Word version of the text was converted to TEI with the help of a Perl script and some XSLT stylesheets that had been created by Paul Caton, based on at-codes/TEI equivalents prepared by Jason Boyd. After this point, the TEI version became the main version of the record text – replacing the original Word version. Any further editorial changes to it would be done in the XML editor oXygen. Also, the TEI-XML version of the text became the target for the EATS tagging activity which was carried out after all the paleographic checking and other editorial work was essentially completed.

**The FTR project Subversion repository**

The FTR project Subversion repository is the storage mechanism for all the TEI text. There were two kinds of TEI record files: (a) the record texts themselves and closely associated metadata about each record, and (b) the associated REED document information (corresponding to the material which appears in The Documents section in REED’s print collections).

Within the repository, REED could take advantage of Subversion’s versioning model to capture the gradual development of the record text from its initial Word-oriented file as submitted by the editor through to later Word version, and then through the subsequent TEI versions. This was meant to mirror, in a digital context, REED’s previous practice of a “top copy” of the text which DDH understood to be a versioning concept. By “top copy” one thought of earlier versions of the text being replaced by newer versions as the editorial process was carried out on them. Subversion’s versioning mechanism, which offered the
current version to anyone who asked for it, but which had earlier versions available as well, seemed to operate well for this purpose.

While designing the folder structure for the Subversion repository, DDH tried to take into account the collection orientation of REED’s materials. Thus, the repository design was based on the idea that in future each collection would have its own largely separate document repository. Although each collection would have its own repository, the structure within each one would be the same, so someone who pulled out work stored in a Subversion repository from, say, several years earlier, would be able to find their way around. Thus, the Fortune Theatre repository was meant to be, as far as it was possible to understand future needs for other, larger collections from the Fortune Theatre materials, the one used in all subsequent collection repositories. The proposed structure is described in Appendix A.

Although Subversion was used for the FTR project, there are alternative version control systems available. Some, such as GIT (https://github.com/) or Mercurial (http://mercurial.selenic.com) seem to be more “trendy” at present. In some sense, these different systems employ a sufficiently similar paradigm that it did not matter which one was actually explored for this prototype. However, when a larger REED digital initiative gets underway, REED will need to be involved in a technology for the long term and will have to decide which version control system has the best future ahead of it.

In addition, all these systems (Subversion, GIT, Mercurial) are systems that were really designed to manage version control for computer source code (programming), and are not fully aligned with the needs of a text-driven project such as REED. In particular, computer source code, although textual, is significantly line oriented. Although these systems have been used for text-oriented projects by DDH and by others with good success, they are not a perfect fit for the needs of those projects, nor for REED. Thus, in the future, REED and its technical team need to keep an eye open for systems that provide better support for text-oriented projects.
The Fortune Theatre Confluence repository

In addition to a Subversion repository, DDH also provided a space on its Confluence content management system (http://www.confluencecorp.com/). DDH’s practice is to put documentation about the editorial technologies, software, and practice used in its project work in Confluence. At the end of the project, REED’s confluence space contained twenty-five documents which documented:

- the Subversion document structure
- the TEI encoding scheme developed for this material
- some of the documentation about the at-code conversion process developed by Paul Caton
- the setup for EATS, including the entity types and their relationships, and notes about how EATS was to be used

4. REED at-codes to TEI Customization

As stated in the grant application for this project, it was decided that REED digital collections would be prepared using a widely adopted standard for the production of electronic, machine-readable texts for humanities scholarship: the Text Encoding Initiative (TEI)’s P5: Guidelines for Electronic Text Encoding and Interchange, based on the eXtensible Markup Language (XML), a WWW Consortium (W3C) Recommendation.

XML enables the separation of the structuring/description of information from the formatting/display of that information, which can be done at an additional stage (using XSLT, HTML, and CSS). XML is a set of formal rules for creating tag sets that enable the sharing of information, or data interoperability.

The decision to use TEI-XML markup required making decisions about the continued use of the markup codes that had been specifically designed to facilitate the production of the REED print collections (referred to as “at-codes” because they begin with the at sign [@]). Developed and adopted by REED in the early 1980s, REED collection editors continue to use this markup scheme. Given that
many editors of forthcoming REED collections have learned how to implement the at-codes and are currently preparing collections using the at-codes, it was agreed that, in moving forward, REED should proceed on the basis that it would have to work with the at-codes, at least in the early stages of production, following submission of the records text by the editor(s). This meant that a process had to be developed that would transition text marked up in at-codes to text marked up in TEI-XML.

The ultimate goal in this transition from at-codes to TEI-XML was to replace a system where typography and its layout on a print page was the fundamental bearer of project-specific, context-specific, and often implicit meaning (e.g., the REED convention that the appearance of italics in the text of a transcription signified, at least to those with knowledge of that convention, expanded portions of words abbreviated in the original) with a system where an explicitly descriptive markup standard was the fundamental bearer of meaning. The major difference between the at-codes and TEI-XML is that the former’s objective is presentational (i.e., concerned with how text should look on a printed page) while the latter’s objective is descriptive (i.e., concerned with explicitly describing the structure and semantic features of a text).

A clear example of the difference between the at-codes and TEI-XML can be seen with the use of curly brackets {{...}} as part of the at-codes schema. They are used to indicate that text should be displayed in italic script. However, they are used both for stylistic (i.e., meaning-neutral) purposes, such as footnotes or the (signed) that precedes signatures, and for meaning-bearing purposes, such as expanded portions of words that are abbreviated in the original. Instead of muddling the stylistic and the semantic, TEI-XML tags distinguish a signature (<seg type="signature">…</seg>), an expansion (<ex>…</ex>), and text italicized for stylistic purposes (<hi rend="italic">…</hi>). With XML, a computer can easily distinguish expansions from other content and then format them in italic or bold or underline or not display them at all.

The document “Guidelines for REED at-codes Application and TEI P5 Markup” (Appendix C), developed by TEI Customization Lead Jason Boyd, represents a
The current use of the curly brackets has largely been retained because most uses
are amenable to conversion. Curly brackets used in the records text for expansions and for style-italicization of (blank) (indicating a blank space in the original where writing would have been expected) can be converted to TEI equivalents, and those used for style-italicization in record subheadings and for headnotes for ecclesiastic court cases can be deleted as a part of the conversion process. The decision to retain the style-italicization in the foot/textual notes resulted in some conversion problems, and although revision to the conversion script could solve these problems, they should likely be deleted in conversion as with the record subheadings and ecclesiastical headnotes. This would mean that, if REED house style display conventions for footnotes (using roman and italic) were to be preserved, the individual components of the footnotes would have to be separately tagged using a customization of the <term> and <gloss> tags. These tags might be helpful to implement for reasons beyond display reasons. (In any case, the more significant issue relating to footnotes is whether they will still be done according to current practices, since TEI makes the creation of separate footnotes for most of the issues that footnotes deal with potentially redundant (see part 5).)

The remaining at-codes were those that were incapable of automatic conversion and thus required revision or deprecation. The solution for the deprecated at-codes was to directly markup the relevant features in TEI. The application of two at-codes was revised: @i\ (matter added in the original added in another hand) (see section 1.6) (referred to above) and @m\ (centred text) (see section 1.7). The remaining at-codes are deprecated (see sections 1.8-1.13).

On the whole, because they are largely amenable to automatic conversion, the great majority of the at-codes could be retained. In total, six at-codes pertaining to the markup of Records text have been deprecated (counting the three at-codes used for right, left, and both [right and left] manuscript braces as one).

For the purposes of this project, and perhaps for the future, the paleographic checking (“palcheck”) stage was the optimal time to implement the TEI tags that replace the deprecated at-codes, since the transcriptions and at-codes are checked against reproductions of the original documents, and thus the paleographer could be certain that the TEI markup accurately represented the features of the original.
This means that it is possible for editors never to have to use TEI in their preparation of the record transcriptions. However, this also means additional work on the part of the paleographer, since s/he has to remove the deprecated at-codes and then apply the TEI markup, rather than simply checking to see whether the at-codes have been used in a way that accurately reflects the nature of the original. This work can be time-consuming when the layout of the text is unusual, or even in straightforward columnar or tabular form, especially if manuscript braces are involved. Editors who would be willing to take on this TEI work would help shorten the palcheck stage of the editorial process.

A number of approaches to dealing with text in columnar or tabular format and with manuscript braces were tried. The use of the <list> tag set and the <cb/> (column break) tag for lists over multiple columns was eventually dropped in favour of using the <table> tag set, which the technology (Kiln) could process more easily. In effect, the <table> tag set was used not only for content that was explicitly tabular but as a way of creating a grid on which textual elements could be mapped. This is not ideal since in best practice <table> should be a textual feature that is a table, whereas if it is a list, then <list> should be used. A more consistent and nuanced use of the TEI table feature might be developed. For a list of examples of how tables were used in the Fortune Theatre records, see Appendix C, section 6.

5. File Conversion and Additional TEI Markup

In part because TEI had to be introduced at an early stage of the in-house editorial process (the palcheck stage), it was decided that it would be preferable to work with a single and the final markup schema, rather than with two. The at-codes were always permanently discarded in the typesetting phase of production, and so were seen as a means to an end, facilitating typesetting and the print collection, the “end” text. In contrast, the TEI-XML versions of the records would constitute the ‘end’ text in which the records would be preserved and would be the source of various outputs (such as revised versions). Accordingly, the goal was to convert the at-codes to TEI as soon as possible in the in-house editorial process, with subsequent editorial work being done in TEI. Therefore, once the MS Word files containing the records transcriptions had been through the palcheck stage, and the
deprecated at-codes manually replaced by TEI, the records files were able to be and were converted.

As a part of the conversion process, the “Code List” that is usually prepared for a REED collection was marked up in TEI. The Code List is so called because it comprises a list of the documents from which transcriptions have been made, with each document being given a four-letter code. These codes are used by editors when creating the “unexploded” headings for individual transcriptions, which are marked-up with the at-code @h\...\h. These unexploded headings contain a three-letter code designating the subdivision of “The Records” they are to appear in, the year heading they are to appear under, and the document code, separated by exclamation marks, eg @h\XYZ!1234!ABCD\. This file enables editors to avoid having to enter the heading in full (which can sometimes involve significant reduplication, given that multiple transcriptions can derive from the same document or document collection) and also facilitates revision (in a single place rather than multiple places) and ensures consistency. Since the Code List has no at-code markup that could be converted, it was marked up manually in TEI (again to save reduplication) (see 2 below).

Example of an item in the Code List, marked up in TEI:

```xml
<item>
  <abbr>AAGF</abbr>
  <expan>
    <title type="record_head">Assignment by Agnes Henslowe to Gregory Franklin and John Hammond of Leases from Edward Alleyn to Philip Henslowe</title>
    <seg type="repository_shelfmark">
      <idno type="repository" corresp="#abbr_DCA">DCA</idno>: Muniments, Series 1, Group 53
    </seg>
  </expan>
</item>
```
Formerly, a script written by Abigail Ann Young (in the program language C) “exploded” the coded headings, populating the headings with the expanded document information. Much the same procedure was followed, but using the TEI-XML file and an XSLT script (see number 3 below).

Paul Caton (DDH) designed and created the conversion process for the conversion of at-codes marked up text to TEI marked up text, including the “exploding” of the headings. The process Dr Caton developed is as follows (adapted from the eREED section of the DDH Confluence website: “Technical research development: Conversion of "at" codes”). This is all done using the command line interface (the Terminal application on Macs; Command Prompt on PCs):

In /trunk/fortunetheatre/tools/ (in Subversion) are:

1. at_codes_to_TEI5.perl – this is a simple Perl regular expressions script to transform REED at-coding into TEI P5 markup
2. record_head_code_list.xml – this is the TEI-XML version of the list of heading information that corresponds to the “compressed” codes that are used in the headers of record files [the Code List]
3. add_2nd_stage_markup.xsl – stylesheet that replaces the “compressed” parts of the main record header with the “exploded” equivalents from the code list
4. add_identifiers.xsl – see below

**Suggested workflow**

1. copy /fortunetheatre/tei/records/Finsbury.xml into a temporary dir, eg. temp1/
2. add new <text> to the file if necessary, then cut and paste the appropriate record bits from the Word file
3. run at_codes_to_TEI5.perl on the file in temp1/, with result file going into temp2/ [this stage converts the at-codes to TEI]
4. run add_2nd_stage_markup.xsl on the file in temp2/, with result file going into temp3/ [this stage populates the record headings from the Code List]
5. run add_identifiers.xsl on the file in temp3/, with result file going into temp4/ [this stage automatically generates the xml:id values for the footnotes and marginalia]. Originally the Perl script added xml:id attrs to some elements, but because it just put in a placeholder value instead of giving individual instances unique values, that meant the generated XML was automatically invalid. So now instead it adds n="CHANGE_ME_TO_XMLID," and add_identifiers.xsl adds them in the final processing step
6. copy file in temp4/ back to /fortunetheatre/tei/records/, overwriting existing Finsbury.xml
7. svn commit new version to repository

Examples of commands used

1. using the command line interface (eg, Terminal on Mac), from ~/Desktop, copy the Finsbury.xml into temp1/ and rename it:
cp [PATH TO YOUR WORKING COPY, eg, /Users/[USERNAME]/Documents/eereed/]/trunk/fortunetheatre/tei/records/Finsbury.xml temp1/Finsbury_STAGE.xml
[The path is displayed in the bottom bar of the oXygen interface. /trunk/ should be omitted.]

2. in your Finsbury_STAGE1.xml, add a new <text> inside <group> (if not one already there), so that you have a skeleton like so:
   <text type="record" xml:id="APPLE">
     <body>
       <div type="main_head_wrapper"/>
       <div type="sub_head_wrapper"/>
       <div type="transcription"/>
       <div type="translation"/>
       <div type="endnote"/>
     </body>
   </text>
then cut and paste from the Word file record into the appropriate elements.

3. from inside /fortunetheatre/tools/ directory, run at_codes_to_TEIP5.perl on the file in temp1/, with result file going into temp2/
./at_codes_to_TEIP5.perl < ~/Desktop/temp1/Finsbury_STAGE1.xml > ~/Desktop/temp2/Finsbury_STAGE2.xml

4. still inside /fortunetheatre/tools/ directory, run add_2nd_stage_markup.xsl on the file in temp2/, with result file going into temp3/
/usr/bin/java -jar /Applications/oxygen/plugins/eatslookup-oxygen/saxon9he.jar
-o:~/Users/[YOUR USERNAME]/Desktop/temp3/Finsbury_STAGE3.xml
-s:~/Users/[YOUR USERNAME]/Desktop/temp2/Finsbury_STAGE2.xml
-xsl:add_2nd_stage_markup.xsl

5. still inside /fortunetheatre/tools/ directory, run add_identifiers.xsl on the file in temp3/, with result file going into temp4/
/usr/bin/java -jar /Applications/oxygen/plugins/eatslookup-oxygen/saxon9he.jar
-o:~/Users/[YOUR USERNAME]/Desktop/temp4/Finsbury_STAGE4.xml
-s:~/Users/[YOUR USERNAME]/Desktop/temp3/Finsbury_STAGE3.xml
-xsl:add_identifiers.xsl

6. from ~/Desktop, when satisfied file is OK, copy back into your working copy of the repository:
mv -f temp4/Finsbury_STAGE4.xml [PATH TO YOUR WORKING COPY]/trunk/fortunetheatre/tei/records/Finsbury.xml

7. from inside /fortunetheatre/tei/records/ directory, commit the changed file to main repository:
svn commit -m ’added new records to Finsbury.xml’ Finsbury.xml

Besides those detailed above, there were a few additional steps. When populating
(i.e, manually cutting and pasting) the record section of the basic REED Records TEI
template (see 2 above) from the MS Word document – a tedious and laborious
process that we hope can be automated in future – xml:ids for each record were
manually added, using the process described in section 4 of “Guidelines for REED
at-codes Application and TEI Markup.” It is hoped this process could also be
automated in the future, since the xml:id values are constructed from information
contained in the unexploded record heading and the subheading.
Once the above was finished, the TEI file was then saved in the fortunetheatre/tei/records folder in Subversion, and then processed by the Perl script (see 3 above). The file was then opened and any validation errors fixed. These were mostly caused by errors that had been made in the application of the at-codes and/or TEI. Under the at-code system, a validation script (again written by Abigail Ann Young in C) was used to locate and correct such mistakes. A similar Perl script will need to be created to do the same in the future. Once these errors were corrected, the file was saved and run through step 4 (above). Again, the file was opened to correct at-code coding errors that mainly affected the populating of the record headings from the Code List (improperly applied @h\...\ would result in the heading not being “exploded”). The file was then again saved, and run through steps 5-7 (above). This completed the conversion process.

**Additional TEI markup**

There were two types of TEI markup that were done post-conversion: the first type was markup which was considered to be minimally necessary; the second was markup which explored how TEI markup could be used to describe many of the features of the original manuscript that are currently dealt with by textual notes (footnotes).

The dating of records is an important piece of information, particularly since the records in the print volume are arranged (in part) chronologically. REED uses a system whereby dates falling between 1 January and 24 March are dated with a “slash year” (eg, 1614/15). This is because before 1642 the year was regarded as running 25 March-24 March rather than from 1 January-31 December, so what would be dated today as 2 February 1615 in the pre-1642 period would then be dated 2 February 1614.

Information relating to dates is usually found in two different places in a REED record: the year(s) in the main heading (which in a print REED collection is removed when records are grouped under general year headings) and the day, month, or other more specific sub-year dating information. In order to make dates
machine-readable, it was decided to use the various ‘-iso’ attributes for <date> (ie, when-is0, from-is0, to-is0, notBefore-is0, notAfter-is0), and use modern dating (the justification being that dates falling between 1 January and 24 March could be automatically outputted as a slash year if desired). The full date was manually entered in the <date> tag in the main heading. Date information in the subheading was not marked up unless a record had multiple subheadings with different dates; in these cases, the date was manually marked up in the subheadings, and the date range represented by the combined subheadings was entered in the <date> tag in the main heading.

Subheadings usually contain information about pagination (where in the document the record transcription is from). This was manually marked up in TEI for the start of the records, using <pb type="…" n="…"/> in the same way as page breaks within record transcriptions were marked up during the palcheck stage (see sections 5.2 and 1.12). Single pages were not marked up using the <pb> tag, with the argument that you cannot have a page break with a single page. However, this means that the information in the subheading is unmarked in TEI. Subheadings could possibly be marked up automatically in TEI, rather than manually.

For marginalia and textual notes, it was necessary to connect marginalia to where in the margin they were to appear and textual notes to what in the text they were referring. For marginalia, <anchor corresp="#"/> tags (indicating where the marginalia was to be placed in relation to the body of the record) had already been entered at the palcheck stage and all that was required was the addition of the xml:id value of the relevant marginalia to the corresp attribute in <anchor/>. For textual notes, the relevant portion of the record was enclosed in <ref target="#" type="foot_anchor">…</ref> tags, and the xml:id value of the relevant textual note was added as the target value. See sections 5.3 and 5.4.

When the records were converted, book and journal titles in the endnotes lost their italic formatting. These titles were marked up with <title type="monograph">…</title> or <title type="journal">…</title>. See section 5.7.

The final required TEI markup was to add a series of <link>s to link records to the
relevant document descriptions and *Early Modern London Theatres* (EMLoT) and *Patrons and Performances* records. These tags, with empty target values, should be added to the basic record template.

The TEI P5 Guidelines have many tags and tagging methods to deal with the features of a text that REED editors take note of using textual notes. The non-essential additional TEI markup that was done was largely confined to an exploration of how TEI could be used in place of traditional textual notes and to determine the advantages and challenges of adopting this practice. Examples of this tagging can be seen in section 6 of Appendix C. For the prototype therefore, there is a redundancy in that the textual notes were retained and rendered as footnotes (as in a REED print collection), and the matter these textual notes addressed were also marked up in TEI. Most of this work is not visible on the public interface, because the prototyping stage did not allow for exploration of the ways in which this markup could be processed by XLST. A number of possible outputs could be explored: a fully diplomatic transcription (REED’s transcriptions are semi-diplomatic), a transcription that allowed for a greater interpretative intervention on the part of an editor, and perhaps of most immediate concern, the automatic generation of certain types of textual notes.

In many cases, TEI does effectively and efficiently deal with the matters covered in the footnotes. Using TEI integrates much of the material into the markup, which allows for greater standardization and is potentially much more efficient, in that it eliminates having to create the notes themselves, as well as having to add-in `<ref>` tags and target values, which cannot be easily automated and, therefore, has to be done manually. While one area for future exploration might be how footnotes could be automatically generated from the TEI markup for print outputs of the records, there will always be some textual features that will likely be best dealt with by a conventional textual note, particularly unusual features that are unlikely to recur often and thus are not worth finding automated solutions for.

Some of this tagging also shows how a wish expressed by some REED editors – supplying missing text that REED procedures do not supply – can be met, since a REED version could simply ignore this supplied material. Rec61 (section 6) shows
an example of this.

This experimentation with the footnotes shows that TEI markup has multiple advantages, such as more precision, more accuracy, the ability to standardize and render content dealt with by textual notes machine-readable and manipulable, and greater potential efficiency. The one challenge in this regard is that it would likely require editors to use TEI (or, as in this case, require a TEI editor, who would replace the notes with their TEI equivalents). But it might also have an important advantage in the transcription stage in that it allows editors to fully and formally integrate their observations of a document’s features in the transcription instead of elsewhere (in a textual notes file).

Since much of the other material that could have been marked up in TEI was marked up in EATS (using the <name> tags), there was not a great deal of additional tagging that might have been done, at least that would have had a demonstrable advantage for improving the editorial process or for how REED content is currently used. Some partial work was done on marking up terms and phrases that were different from the main language of a document (using <foreign>), but this might not have much use in the editorial process, although it would enable, say, the study of the use of Latin in English documents and vice-versa. Of more potential value would be the marking up of dates in record transcriptions and endnotes, and the marking up of bibliographical data in endnotes. The markup of the information in editorial subheadings should also be explored (particularly whether it can be automated).

Besides these issues, the question of how TEI can be used for the creation of the Latin and English glossaries, and how this markup could be used to build cumulative glossaries, how the bibliographic markup mentioned above could populate a cumulative bibliography, how the TEI could be used to generate event records for the Patrons and Performances website, and how the introductory matter of a collection could be linked into the records in a more dynamic way are all waiting to be addressed in the future, building on the work done in this project.

6. EATS Implementation
From the earliest stages of planning for the FTR project it seemed evident that one of the major sources for structured data from printed REED collections was each collection’s back-of-book index, which is arguably the primary means of access and navigation of the REED collections. Thus, while planning for REED’s digital future it seemed most useful to assume that material that had in the past gone into separate collection indices would be reconceived as a digital cumulative index across all digital collections. By doing this one would enable cross-collection searching – surely an important feature in any future digital REED resource. The “cross-collection” nature of this material made it fundamentally different from the collection records, which could be managed by the TEI document repository model (described above), and meant that a quite different approach was needed.

Early in the commencement of the project, DDH suggested that the FTR project could explore building an ontology that would in effect take the place of the index of a REED collection. DDH used the word *ontology* in its still emerging Information Science sense: as a formal representation of a knowledge domain by means of a set of entities (akin to index entries on specific things), grouped into classes (akin to broad index categories like “plays and playing”), and connected through explicit declarations of the nature of the relationships between both entities and classes (akin to an index redirect, ie, “see,” “see also,” but much more specific about the nature of the relationship between items. These relationships are shown on the website when a user clicks on the highlighted portions of the records).

Early preliminary analysis by John Bradley of existing REED collection indexes suggested that the entries fell primarily into three types: Persons, Places, and more general Things or Subjects/Topics (eg, “women,” “wrestling,” and “plays”). Jason Boyd, working with undergraduate research assistant Daniela Robibero, compiled combined person, place, and subject indexes for the published London REED collections (*Ecclesiastical London* and *Inns of Court*) as a way of obtaining a clear understanding of the commonalities and differences between REED collection indexes, and as a helpful guide for building an index/ontology for the FTR project. This undertaking helpfully highlighted some inconsistent (or collection specific) indexing practices that needed resolving (most predominantly, the combination or
separation of interrelated entries), as well as enabled an analysis of how person and place names were indexed, what kinds of information they contained beyond first and surnames, and how they were cross-referenced. The place index was particularly helpful as a reference guide to standardized London and area place names that could be consulted when building the ontology.

Past experience within DDH suggested that the software it uses to manage entities, called the Entity Authority Tool Set (EATS) (http://researcharchive.vuw.ac.nz/handle/10063/220), could be applied to the FTR Prototype and would provide an efficient way to explore how this kind of material could be modelled. EATS was originally created by Jamie Norrish of the New Zealand Electronic Text Centre (NZETC) and is used there as the tool to manage authority information (mostly people and places) that arose from a broad range of historical sources. Within DDH it has been used with good effect with several other projects (see the abstract of the paper given about one of these projects at DH2012 in Hamburg at http://www.dh2012.uni-hamburg.de/conference/programme/abstracts/complex-entity-management-through-eats-the-case-of-the-gascon-rolls-project/), and EATS also integrated well with DDH’s emerging publishing framework Kiln, which was going to be used for the FTR Prototype.

EATS is an entity manager, and operates according to a base model that is in part compatible with current thinking within, say, the ICOM (International Council of Museums) International Committee for Documentation Conceptual Reference Model (CIDOC-CRM, http://www.cidoc-crm.org/) and Resource Description Framework Schema (RDFS, http://www.w3.org/TR/rdf-schema/) about entity identifications and their relationships. Entities in EATS represent particular individuals – people, places, for example – but can be typed (or classed, to use RDFS terminology) in a way compatible with RDFS’ class model. Relationships between entities are typed too, and connect entity instances together. They are defined as relating certain types of entities together – similar in concept to RDFS’ domain and range concepts, and CIDOC-CRM’s properties.

Although EATS has many features that align it well with Semantic Web thinking, it
does not provide full support for ontologies such as those described in the Semantic Web world through the use of RDFS and Web Ontology Language (OWL): in particular, EATS has no sense of subclasses or subproperties. However, EATS data can be exported as full RDF data in forms that would be compatible with RDFS/OWL ontologies, and would allow for the data and its ontology to be brought together in an ontology-aware RDF triple store.

Paul Caton (DDH), based on the index analysis undertaken by Jason Boyd, produced an entity framework that could be used to model REED’s “knowledge domain.” He established six entity supertypes under which all (or most) entities could be classed: person, place, collective, thing, occasion. Then he established a basic or sample set of entity types that derived from the supertypes: man, woman (person); place_theatre, place_religion, place_court (place); guild (collective); instrument, prop, foodstuff, drink, animal (thing); occasion_religion, occasion_tradition, occasion_event (occasion). Other entities created that did not have a supertype were rank, office, and activity. As well, some basic entity relations were created to link person entity instances according to family relationships (spouse to spouse, parents to children, or to indicate that two people were unspecified “relatives”), patronage (eg, Elizabeth I, patron_of Queen’s Men; Queen’s Men patronized_by Elizabeth I), and occupation, office, and rank (eg, holds_office). A catch-all entity relation for relations not covered above, associated_with, enabled the linking of entity instances without having to develop a system that would inform the creation of a complex, coherent, and controlled set of entity relations, which was regarded as too ambitious for the scope of the FTR project.

On the basis of this framework, Centre for Digital Humanities (Ryerson)-based, Mellon-funded research assistant Trieneke Gastmeier, working in collaboration with Jason Boyd, proceeded to “mark up” the Fortune Theatre records, using the EATS plugin for the oXygen XML Editor. This component of the process was relatively simple: one highlights the relevant section of the text (eg, a name or term), and then selects “EATS Lookup” from the relevant oXygen menu. A pop-up box appears that enables a search to determine whether an entity instance has already been created for the highlighted item. This feature was somewhat tricky in terms of REED content: because of variation in REED transcriptions (particularly
spelling), there was the risk that multiple entity instances might be created because it was not evident that, say, a person entity instance had already been created for an individual whose surname had been spelled a different way in another document. Multiple variant searches and partial term searches to some extent helped with this problem, but this is not an ideal solution and is also labour intensive. Compounding this problem was the interface provided for navigating the EATS ontology built for the FTR project: one could search for entity instances using a basic search box, but one could not view a structured list of the entity instances that had been created. Therefore one was created manually on Confluence by Ms Gastmeier in tandem with the EATS markup. This helpfully provided a record and resource that could be consulted to avoid duplication or inconsistent EATS implementation, but was time-consuming and inefficient.

If an entity instance has already been created for the highlighted records text, then one simply selects the instance and the TEI tags are automatically created to enclose the highlighted text. If an entity instance does not yet exist, one must first be created using the EATS interface (which has a “create new entity” link). This brings one to a template or web form with various text boxes and pull-down menus. Text boxes are for the names of the entity instances, pull-down menus for entity types and entity relations. In order to deal with the variation in personal names REED commonly encounters, the entity instance name section enables the inclusion of multiple variants, and a pull-down menu that allows for the classification of the variants as the main, preferred form (index), as a “birth” surname (eg, maiden names), as an “alias” name (eg, also known as, multiple married names), or as a variant name (eg, variant spellings of a surname). This attention to variation reflects the REED indexing practice of including such variation in parenthesis after the preferred and index form of a name.

After a new entity instance had been created, then it could be searched for using the EATS Lookup on oXygen, and applied to the highlighted records text. Because the ontology was being created from scratch in tandem with its implementation, this proved to be a very time-consuming practice, although it would become less so as the ontology was increasingly populated with pre-existing entity instances. The EATS-related TEI in the records takes a common form: <name key="[URL to the
entity instance record] type="[entity type]">...</name>. Because of the length and frequency of occurrences of this EATS-related markup, it has a tendency to atomize the transcription, making it difficult to parse both the transcription and the non-EATS TEI markup, which may impact on editing efficiency.

Although in retrospect it may have been more appropriate to the scope of the FTR project to have restricted the EATS implementation to the search facets that have been made currently publicly available on the Prototype Digital Edition website, it was felt that, as an exploratory project, it was important to experiment with EATS' ability to deal with the full range of content covered in a conventional REED index. In trying to do so, two things become evident. The first was that this experimentation would have been more easily and usefully undertaken at a stage when all editorial and research work was complete: because the Fortune Theatre records were submitted in a draft form, a challenge repeatedly faced was trying to implement EATS markup for records where the collection editor had not yet created an expected endnote explaining aspects of the record that were not explicit in the record text, such as elliptical or cryptic references to people, places, or events. The second was that EATS implementation is most effectively done by someone who can easily parse, understand, and even further research REED transcriptions and detect their salient aspects in terms of REED indexing and thus for EATS implementation purposes. This level of skill has been expected of indexers of the REED collections, and it is no different when using EATS. It is a testament to Ms Gastmeier's skill, intelligence, and acumen that she was able to do such a fine job implementing EATS markup with no previous REED-related knowledge. It proved essential for her to consult with the editorial team during the later stages of her work, as has been the case with previous REED indexers.

The ambitious scope of the EATS implementation enabled a sense of how the basic framework established by Dr Caton might be refined and extended. For example, it became clear that entity types for place need further specification: place_theatre was changed to the more inclusive place_playing (which could then be applied to places like inns where playing occurred). Place_built (for buildings that were not classifiable as place_playing, place_court, place_religion) was created, as was place_region (for places larger than a town or city such as a county),
place_settlement (for towns and cities), and place_sub_settlement (for places within towns or cities like streets). A new entity relation was also created to associate various place types with each other (is_part_of / has_part). Council, court, court_legal, troupe were created as instances of “collective” to deal with entities like the privy council, the royal court, legal courts, and patronized troupes.

While these were fairly straightforward additions, replicating other aspects of a REED index proved more challenging. In particular, the broader topics or subjects that connected together a range of content in a REED index (eg, “entertainers and entertainment”) proved to be difficult to implement in EATS, in part because this level of indexing was often tailored to meet the nature of the collection and therefore there was no set standard that would have informed the EATS work. These larger categories also proved to be, in some cases, challenging to mark up: for example, what part of a record was to be enclosed in the EATS TEI tags when the entity instance applied to the entire record rather than a specific part? In many cases, it was not possible to tag an entire record and still have a valid TEI document. Another challenge was that many common REED index entries had no pre-existing entity type, and so had to be created on the fly (eg, the entity type “issue_social” was created for the entity instance “sickness and plague”).

What this work ultimately revealed is that building an ontology for a digital REED corpus (rather than for a single collection) will require extensive planning, information science expertise, standards and technologies, and significant resources in the form of time, labour, and period expertise to implement.

7. Editorial and Workflow Changes and Challenges

One of the main conceptual changes that occurred when moving to a digital edition from print was thinking about the individual record, rather than the “collection,” as the primary unit of stand-alone REED content. This re-conceptualization was informed by an understanding of REED’s scholarly output in digital form as a single corpus of records, and thus by an understanding that users would very likely want to curate their own collections with scopes different from the city/county scope of the typical REED collection, or even the single theatre scope of the FTR project.
To suit the new conceptualization, a record transcription now had to be explicitly associated (ie, in a way readable by a computer) with those parts of the editorial apparatus pertinent to it: textual notes, endnotes, document description. Formerly, the structure of a REED print collection dictated the structure of the electronic files used to create it. The endnotes, for instance, were a separate section within each collection. They were created, edited, and submitted to the typesetter as one file. For the FTR project, this structure could have been imitated by linking record transcriptions to a section on the website containing all the endnotes, but instead the material was structured on the principle that all parts of the apparatus that were specific and unique to the transcription (textual notes, endnotes) should be included as a part of the record. Therefore it was decided that, like textual and collation notes, endnotes should be combined with the records to which they related, in the same files.

In future, whether it will be more efficient to use large files containing multiple transcriptions and ancillary material, or one file for each transcription and associated ancillary material, is still under discussion. Greater granularity means uploading may happen earlier, but it also means having to convert each file separately, as well as open myriad separate files for larger research activities, for example, tracking the appearance of one individual or place throughout a series of transcriptions.

One important factor mitigated against putting all the Fortune Theatre record files into a single file: the Subversion repository. In the Subversion system, each person has a local working copy of the file on his/her computer. Changes are made to the files in the working copy and then the working copy is synced, and changes committed to, the master copy. To prevent conflicts, it is important that a file only be worked on by one person at a time, and any changes committed to the master copy, before another person works on the file. Because the records were in different files, the conversion process, and the TEI editing, could start much sooner (while the paleographer’s check was still happening) and the TEI editing could be done simultaneously with the EATS markup. This made for a much more efficient workflow, even if it posed some problems (eg, the TEI editing would detect
conversion problems that would affect EATS markup).

The editorial personnel from the REED office hoped that working with files within the Subversion environment would allow them to reduce paper use and streamline the editorial process. To achieve this, they decided to work only with a digital, rather than paper, “top copy.” To produce a print collection, they had worked with a paper top copy of each file: a print-out of the original submitted file with every subsequent editorial change marked up by hand. The typesetter would consult the top copy for any changes needed to the electronic file. The proofreader would then consult the top copy, to ensure all changes had been made correctly by the typesetter. The paleographer’s often extensive corrections to the transcriptions (mostly Latin and early modern English) were a challenge for even the most competent typesetter to interpret and input, which necessitated a time-consuming proofread of the corrected file before typesetting, along with a proofread after typesetting. The top copy provided a comprehensive and authoritative record of all in-house change and was used for proofreading.

For the Fortune Theatre material, however, the editorial personnel marked up changes and added comments in the digital files only, using the Track Changes feature of Word to record the changes. Such changes took the place of handwritten corrections on a paper top copy. As before, the paleographer performed his check of transcriptions first, digitally marking up changes instead of marking them up by hand, followed by the copy editor and general editor working on the same electronic file. Changes were not “accepted” (producing a clean, unmarked file) until immediately before the Word file was converted to TEI-XML. The version of the file before changes were accepted took the place of the paper top copy, and all corrections were now embedded in the electronic file. This removed one stage of correction by the typesetter and, therefore, one stage where error could enter the files. With the paleographer and copy editor now entering changes directly, and no typesetting to follow, the personnel wondered whether only one proofread was necessary, rather than two. But when would this one happen? And what should be proofed against what? The TEI-XML conversion and subsequent EATS tagging helped decide the matter. Once the checked and edited Word files were converted to TEI-XML, these files were again opened and EATS tags added, along with the
potential to introduce mistakes into the files. Because of these later-stage manipulations, the personnel determined one final proof must happen, where the final TEI-XML files were proofed against the Word top copy. This meant that any last-minute editorial changes that happened after files were converted would also need to be marked up on the Word files, to ensure an accurate top copy for proofing against.

Although Subversion provided a common shared repository for the files of the project, which was a significant improvement over the paper top copy and multiple copies of files residing on individual computers (which inevitably generated variant files that had to be resolved), the FTR project did not explore as fully as it might have the use of an online project management platform (in this case, Confluence) for managing project workflow and communication. While DDH personnel were familiar with the use of Confluence for project management and used it to provide documentation for the project, REED and Ryerson’s CDH were not and did not use it to manage their own work beyond the shared work with DDH. The value of exploring this in the future is suggested, to take one evident example, by the rather hectic email correspondence that ensued about remaining issues and tasks as deadlines for the project loomed. Compounding the hecticness was simply the number of people involved, the amount of email generated (and mixed with other email), other time demands, and the fact that the team was split up over three institutions, one of which was five hours ahead of the other two. At one point, a Google Doc was compiled from previous emailed Word documents to provide a shared online source for outstanding tasks because it was becoming difficult to grasp how many things were left to be done. A project management platform might helpfully and more effectively facilitate, record, organize, and make accessible project workflow and communication.

Another important aspect of the editorial process that could not be explored by the FTR project, but will have to be addressed in the future, is how collection editors will participate in this new editorial process. This applies in particular to the work on the collection after the conversion to TEI. If, as was the case with the Fortune Theatre records, REED proceeds on the basis that it will not require its editors to learn to use TEI, an editorial interface will likely have to be created that outputs a
TEI file in a format that is conventionally readable and that fully represents the TEI markup in order that the editor can understand and approve any interpretative readings represented by TEI markup.


The FTR Prototype public server is a Django-based platform (https://www.djangoproject.com/) made up of two integrated components:
(a) Kiln (https://github.com/kcl-ddh/kiln), as the server for the Fortune Theatre data, and
(b) Mezzanine (http://mezzanine.jupo.org/), a CMS (Content Management System) platform from which the supporting documentation about the FTR Prototype was created and served.

Kiln is an open source multi-platform framework developed by DDH for building and deploying complex websites whose source content is primarily in XML. It brings together various independent software components, with Apache Cocoon at their centre, into an integrated whole that provides the infrastructure and base functionality for such sites (see http://kiln.readthedocs.org/en/latest/). It, and its predecessor xMod, has supported more than fifty projects at DDH and continues to be developed as a general platform for the publishing of XML (specifically TEI) collections.

Kiln was configured by Miguel Vieira at DDH to work with the TEI record format for the records and document materials, and he combined it with the Fortune Theatre’s EATS repository (which was extracted from EATS into one of its standard XML formats). Vieira tailored Kiln to:

(a) handle the display details of the records. The format you see on the site was the result of significant consultation between DDH and REED about how various aspects of the presentation of the records would work best in a digital context.
(b) handle the searching of the records by a user. In the prototype this is primarily the data that came from EATS about the records. Kiln presents this navigation/browse/search process in the form of a faceted search paradigm (see

DDH was, of course, constrained by the limited funding available for this part of the work through this grant (about a month’s worth of work – and this had to cover the time to do both the Kiln and EATS configuration) to do only a modest setup. Further refinements will be needed to bring the server’s presentation of the data more in line with the enriched understanding of the possibilities of eREED digital that had already come out of the work with the FTR Prototype and that will grow further in the future.

Two refinements to the website that seem obvious are the creation of a page with all the document descriptions linked to the relevant transcriptions, thereby enabling navigation of the records by source document, and the generation of the “Symbols” page from a TEI-XML file (currently this page was manually created like the “About” page). This would enable the creation of links between records with document repository or print source acronyms/abbreviations and the expansion of those acronyms/abbreviations.

It is clear that from the current presentation of the records online, there are refinements to the markup and output of the records that will have to be made. To take perhaps the most obvious example, marginalia are not being displayed in the correct places next to the main text. A careful post-launch audit is being planned that will examine the TEI-XML file against the website display to determine what TEI tags are not being processed properly (or at all), and where the TEI markup needs to be changed and edited. Cross-browser and screen-size display issues may also have to be addressed in this regard.

9. **Patrons and EMLot Interoperability**

For a number of years, REED’s research work that contributes to the print volumes
has also resulted in the creation of new and related data in REED’s *Patrons and Performances* and *Early Modern London Theatres* (EMLoT) databases. As a result, an important ongoing goal is to find a way to digitally connect related work between eREED (currently the FTR Prototype), *Patrons*, and EMLoT. Furthermore, planning how data between these three repositories could be linked fits with both the larger web world’s current interest in Linked Data between separate data repositories ([http://linkeddata.org/](http://linkeddata.org/)) and REED’s wish to develop its digital infrastructure in ways that allow it to share its data with other projects through Linked Data and other Open Data initiatives.

For data from one project, say EMLoT, to connect to data in another, say the FTR Prototype, there has to be a semantic connection – thus, part of the work here was to explore what data actually linked semantically between REED’s three digital repositories. What kinds of data were being shared between the FTR Prototype and EMLoT or between the FTR Prototype and *Patrons*? The connection between the prototype and EMLoT was strong since both are centred on London theatre activity, and data related to the Fortune Theatre materials had, in fact, been put into EMLoT by REED for exactly this purpose. The obvious points of contact were People, Troupes, Venues/Theatres, and Primary Sources. While the same structural similarities exist between the Fortune Theatre and the *Patrons* data, they have less data in common as yet, because the *Patrons* materials have been for the most part provincial in focus. In the longer term, however, the Fortune Theatre team is aware that *Patrons* will expand its domain into what is now the greater London area when collections for the historic counties of Middlesex and Surrey are digitally published. At that point, connections to people/patrons/troupes and to venues such as the Curtain and Theatre as well as the Fortune (Middlesex) and the Globe and Rose (Surrey) will become more prominent. Furthermore, the plan is to have eREED expand its domain outside of London as well, at which point data with provincial focus will begin to connect to *Patrons*.

In addition to the People, Troupes, Venues, and Sources, other possible semantically linked data would be REED’s records with what are called “events” in both *Patrons* and EMLoT. There had been, unfortunately, no time to include this significantly in the FTR Prototype work although the potential could be explored
through sample patrons and performance troupes associated with the Fortune that also appear in the provincial data available and mapped on *Patrons*, and the bibliographic data on EMLoT directly relevant for records included in the FTR Prototype.

Given the independent, but semantically connected, nature of data in eREED/Fortune Theatre and EMLoT (and potentially, in the future, *Patrons*) it would seem sensible to take a Linked Data approach to the connections between them. One of the key ideas in the Linked Data initiative (that structured data for each project be available as RDF [Resource Description Framework] triples, [http://www.w3.org/RDF/](http://www.w3.org/RDF/) is at least conceivable for all three of them. Although none of the projects have RDF-oriented data access points (as, say, SPARQL [a query language for RDF, [http://www.w3.org/TR/rdf-sparql-query/](http://www.w3.org/TR/rdf-sparql-query/)] end points) both EMLoT and *Patrons* have highly structured data behind them (in the form of relational datasets) that are highly compatible with an RDF representation. Nonetheless, although full linked data characteristics are not available without some further development work, EMLoT does already have published URL entry points (permaLinks) to the data for several of its entity types, including to particular People, Troupes, Venues, and Sources. These are constructed as RESTful URLs ([http://en.wikipedia.org/wiki/Representational_state_transfer](http://en.wikipedia.org/wiki/Representational_state_transfer)) and if entered directly into a browser take one directly to the EMLoT display for this particular Person, Troupe, Venue, or Source. Thus, for these historical entities these same URLs can act in ways similar to RDF/Linked Data URIs within EMLoT.

On the Fortune Theatre side of any link: Persons, Troupes, and Venues all are represented as EATS entities, and EATS’ meta-structure mechanisms allow one to associate any number of URLs to any EATS entity. Thus, we have exploited this EATS mechanism to allow particular persons or troupes entered into EATS for the prototype to be linked to their corresponding EMLoT entity. The exact semantics behind this link in EATS is not very specific, so the nature of the link here (which, reading the two linked entities in semantic web terms, should be interpreted as, essentially, an *owl:sameAs* link) could not be that tightly specified. However, the linking mechanism, as informal as it was, did allow the REED team to establish the links and explore what it was like to have these links available between Fortune
Theatre and EMLoT records.

The database structure behind the Patrons site, with its presumed identification of individual instance of an entity by means of a database record key, means that it would certainly be possible to expand its web application so that RESTful permalinks, based on database keys, could be established and made to work for the Patrons web application. If this was done, then these permalink URLs that took one directly to Patrons data about a particular patron, or place, say, could be referenced in the same way as they currently can be done for EMLoT.

The mechanisms in place in the prototype support linking from the Fortune Theatre materials to EMLoT, but it would be desirable to support linking in the other direction too – from EMLoT to Fortune Theatre Records. Indeed, a landing point within the Fortune Theatre materials is in fact already available since EATS entities are already uniquely identified by means of a URI mechanism. In fact, it is the URI that provides the way to point to an EATS entity from within the text of a TEI marked up record. Unfortunately, however, there was no opportunity in this project to explore the expansion of the EMLoT structure, so links from EMLoT to the Fortune Theatre dataset could be similarly recorded and made available. This would have required an extension to the EMLoT data and software and there was no funding available to do this.

Although in the end the amount of exploring of linked data between the Fortune Theatre prototype, EMLoT, and Patrons was, of necessity, limited, it did allow the REED team to focus their thinking on the issues involved in making this linked work and allowed them to explore how significant it might be by providing actual examples of links.

**10. User-customized On-demand Printing**

The eREED prototype platform has afforded an opportunity not only for us to reflect on the prospect of provisioning print-on-demand functionality, but also on the wider potential for bringing the Create Once, Publish Everywhere – or COPE – model to REED. In this context, print-on-demand can be seen as part of a
constellation of potential functionality and output channels (potentially including printed books produced very much in the image and spirit of the traditional REED county volume) delivered from a unified, web-based editing and publication environment. The COPE model will clearly need to lie at the heart of a reinvented REED and represents a logical conclusion to the technical and methodological trajectory traced by the project since its earliest publications.

**The potential of print-on-demand**

The most attractive possibility of print-on-demand for REED is the ability to completely break down the traditional “by county” partition in generating offline, for-reference collation of REED’s archival sources. Although the prototype represents only a tiny dataset, potential use cases for print-on-demand are still very evident as, for example, when navigating through source documents pertaining to the Place:Settlement entity, “Finsbury.” Even now though, a look at the Finsbury location entity generates 171 source records, and it is easy to imagine how the entity-based approach that eREED uses could form the basis of a number of powerful searches that would yield large quantities of source materials, which a user may well want to then study en bloc and offline. Whilst it is possible to consult each document in sequence on the eREED prototype website, some simple additional functionality would allow a user to produce a print-ready version of the entire set of materials (in PDF format, say), with custom ordering; or to elect to save this particular set of results into a “basket” of primary source materials which could be added to and then aggregated together into a larger, compound print output. Numerous studies have suggested that hard copy print remains an important part of the research process (even where it duplicates content freely available online), and in this context it is also worth considering that actual finished, bound, printed copy is a viable output too – harnessing the networks established by companies such as Lightning Source or Amazon’s CreateSpace, but combining this with the credibility that the imprimatur of a reputable scholarly publisher (such as Boydell & Brewer, REED’s print publisher) inherently brings. (See for example A. Ciula and T. Lopez, “Reflecting on a dual publication: Henry III Fine Rolls print and web,” *Literary and Linguistic Computing* 24.2 (June 2009): 129-41. Various studies of research practices in the context of EEBO and ECCO
have shown that they are used as finding aids but rarely cited in favour of a traditional printed edition.)

Increasingly in the digital world the notion of a “printed” output refers to an output analogous to the printed page, but not necessarily physical at the point of use (nor necessarily envisaged as a vehicle for physical delivery as is really the case with PDF); thus, EPUB would therefore be an equally attractive format for a finished output, or indeed Word, Rich Text, or XML as “working data” for further use, manipulation, or processing by the end user. Indeed the Kiln framework employed by the eREED prototype already has EPUB, PDF, and XML export capabilities.

**Print-on-demand in broader context**

The current push towards Open Access problematizes most modes of publication in the broadly conceived Arts and Humanities, particularly from a sustainability point of view. Reconciling open access with the need for revenue, and ensuring the OA imperative does not strangle the potential for income, is a difficult challenge. Although the implications of this are yet to be fully understood, it is clear that financial sustainability will be one of the major ongoing considerations for the eREED project, as with any large scale, digitally enabled research enterprise, where costs do not stop at the point of publication, but are ongoing. Moreover, “sustainability” here cannot only mean keeping the power and lights on, and ensuring eREED remains available; it must imply both academic sustainability, enabling research and editorial work to continue; and functional sustainability, allowing the eREED platform to develop and to keep pace with technology and prevailing delivery mechanisms. As an active research project, eREED will need to be able to rely on external support from funding bodies for at least some time into the future, and Open Access is, for most, a fundamental condition of grant awards, thus the dual concern of building up a model for financial sustainability whilst remaining true to the principles of Open Access is a live and complex issue. In order to understand where print-on-demand might fit into this picture, it is worth first looking at the currently emerging charging models in the sector.

There are three potentially interesting analogues elsewhere in the world of
digitally enabled scholarship in the humanities. The first is the direction being taken by large academic presses in the sale and marketing of their own digital editions and outputs. A product such as Oxford Scholarly Editions Online (http://www.oxfordscholarlyeditions.com), for example, typifies the prevailing approach: access is available for institutions and individuals on a short to medium subscription basis, or institutions have the option to pay an (unpublished) one off fee for unlimited access. This model largely replicates that employed successfully by those large national newspapers which undertook to digitize their archives in recent years; access is available to (say) the UK Guardian, or US New York Times, digital archives for periods of between premium-priced short term access (anywhere from a single day to a week) for individuals all the way to institutional access in periods of 12 months. From the point of view of institutional libraries however, both these models are potentially concerning; they represent “rental” of content rather than outright purchase, and investments in products of this sort have to be carefully tracked and monitored against actual institutional uptake in order to justify continued subscription. In both cases noted above, the annual subscription does not necessarily guarantee an increase in the amount of content available; in the case of newspaper archives of course the materials are finite, and for OSEO, content is broken down into individual “modules” each commanding their own fee (thus if new content is added in the future the expectation would be that the overall cost of subscribing to the entire “corpus” would rise, or that an individual institution would carefully tailor its subscription only to include modules that were in specific local demand). Crucially in these cases, the publishers are under no obligation to actively “improve” or enhance their offerings for subscribers over time – only to ensure continued access to them. Thus the risk to the publisher is low, whilst the subscriber must continuously scrutinize the value proposition in continuing to invest in access to the product.

A second model and an alternative to that of straightforward subscription and content “rental” is that of the Cambridge University Press “Cambridge Edition of the Works of Ben Jonson.” Published in two forms – a traditional seven volume print edition, and a greatly expanded digital edition (funded by the Andrew W. Mellon foundation), due to go online later this year – the decision has been made to offer the digital edition to individuals for the same, one off fee (at a price mirroring that
of the print edition). Whilst short or fixed term subscription options will probably be available as well, those users and institutions which elect to purchase the product outright will benefit from any updates the product receives (in terms of form, content, or function) into the future. In addition, some parts of the expanded digital edition will be free at point of use, providing a benefit for those who have purchased the print edition (in the form of a set of over eighty substantive textual essays which were too expansive to consider for inclusion in the print volumes) and a sense of the functionality of the full digital edition (via a performance archive which hints at the searching and cross linking capabilities of the full digital edition). This is a hybrid, and somewhat experimental model, and it remains to be seen whether it will be a success; continued development of the Digital Edition will ultimately rely on continued sales.

It will be noted that none of the models considered briefly here deals in any real way with the issue of Open Access since in none of these cases is full access to content on offer without subscription. For a final analogue, then, we turn to a different research project, also based at the University of Toronto: the Lexicon of Early Modern English (LEME), a three-way partnership between the University of Toronto Press, University of Toronto Libraries, and the scholarly research team. The LEME resource runs on a Freemium model, whereby all the data is freely available, but access to some of the more useful and granular tools for marshalling and interrogating it is “gated,” requiring subscription (specifically three of the five available indices are accessible only to subscribers). LEME is, to date at least, a success story, in that this model is actively sustaining the resource (defraying costs not only for maintenance and hosting, but also for continued content development), and the partnership between the three contributing partners is effective; UTP provide the subscription services and hosting (the digital equivalent of sales and marketing); and the UT Libraries provided the hosting, and continuing product design and development effort (the digital equivalent of product design, in some sense copy editing, and printing). Although the number of paid subscriptions is relatively low compared with the number of users of the site who are content to use the freely available tools, institutions whose scholars will benefit from the advanced features clearly perceive the benefits of subscribing and in so doing they directly help to develop and sustain the product.
In the Freemium context the attractiveness of print-on-demand functionality (at least at this moment in time) can be seen as one of the potential building blocks of a sustainability model for eREED which might be achievable without compromising on the project’s ongoing commitment to Open Access.

The prospect both of a diverse range of print-on-demand offerings, and a COPE model which may well see the web-deliverable eREED corpus rapidly become the principle output of the project, raises further questions about what the role of an academic publisher should, and needs to, be in the future. As already noted, the imprimatur of a publisher is an important badge of credibility for scholarly publications and this will no doubt continue to be the case (equally from a practical point of view, if hardcopy output – whether on demand or not – is to continue, print publishers are clearly best placed to handle it). However, as the example of LEME illustrates, the way in which publishers have acted as enablers of or vehicles to publication is clearly changing. As the current publisher of the REED print volumes, Boydell & Brewer is clear that a crucial aspect of their role going forward should continue to be to act as an interface between the marketplace and end users (whether they are consuming content in book or, indeed, any other form). Equally, the decision on exactly where the line is to be drawn between the OA and enriched / premium versions of the eREED platform will be one of the next challenges for the project to face (together with the development of a strategy for the ongoing development of the project which ensures that subscription remains a sufficiently attractive option in order to ensure sustainability); this again is another area where publishers have the necessary experience to be able to advise based on their market knowledge.

11. Challenges for Future Exploration/Resolution

While certainly not an exhaustive accounting of future areas of development, the broad areas listed below are those most significant for future research.

Conversion/TEI markup
As detailed in Appendix C, there are a number of issues and questions that remain to be addressed and answered regarding REED’s TEI Customization and the conversion scripts. As well, there are additional issues such as incorporating REED’s TEI Customization into the <teiHeader> or creating an ODD (One Document Does It All) for this customization, against which REED TEI documents can be validated; developing further methods to automate the conversion process (e.g., converting MS Word files into TEI, generating record id numbers, converting editorial subheadings); and exploring how material covered by textual notes should be dealt with. Further refinements to the TEI Customization will undoubtedly be needed, and it will also need to be expanded to deal with other sections of a REED collection (Introduction, Select Bibliography, Glossaries).

A validation script also needs to be created to check that at-codes are properly applied before the conversion to TEI.

**Editing environment**

For the future, it seems preferable to develop an integrated online editing environment that would enable start to finish collection creation, where editors could edit, store, and even make accessible drafts of their collections in the process of creation to the stage where a collection is ready to be added to the publicly accessible corpus of REED records. This environment should also be capable of enabling the re-editing of collections already in print.

An optimal web application would provide a document management system (like Subversion), a project management system (like Confluence), and an editing interface, perhaps even one with a user-friendly TEI tool bar that could apply TEI markup to highlighted text while representing that markup to the editor according to the standard REED practice (e.g., the expanded portion of a word in a record could be highlighted, an “expansion” button clicked, and the highlighted section marked up with <ex>…</ex>, although the editor would see this section represented in italics.

**Editorial process**
A proposed editorial workflow for the creation of a digital edition is provided in Appendix C, Section 8. The next logical step is creating a full digital edition, with all the relevant components, to fully explore the entire editorial sequence and develop from that experience an editorial process that factors in every component of the creation of a REED collection. One issue the new editorial process will have to address is the extent of and strategy for proofing XML files, which was not resolved in the FTR project.

**eREED reading/viewing interface**

In terms of the eREED interface and improving its usability, there are a number of areas in which it is evident further work needs to be done. As a prototype, the primary aim was to develop a concrete example of what a collection of REED records would look like on an online interface. As can be seen from the prototype, there are basic issues relating to the reading experience (font sizes, hyperlink highlighting, the use of fluid/variable width instead of fixed width, which results in records displaying in very long lines on large monitors, the lack of a clear connection between footnotes and associated text, marginal notes not displaying in the correct place in the margins, overlapping popup boxes) that need to be fixed.

Beyond these relatively minor issues, the larger issue of the the reading interface for a REED Corpus requires considerable and careful planning. Will users want to be able to read multiple records on the same screen? Will they want to see relevant content from other sections (eg, document descriptions, glossaries, index) alongside a record and how (full view, partial view, link)? Could there be multiple “views” of a record (XML view, semi-diplomatic, full diplomatic, REED version, editor version)?

**Searching/navigation**

The search and navigation functionality for a REED Record Corpus will have to be expansive enough to address a range of research queries. A robust cumulative index/ontology will be a major component of this functionality (as a faceted
search), but other components of the TEI markup (ie, type values) might be another method of navigating/organizing content. Besides providing the means to access records according to the source document, a user should be able to organize the records in multiple ways beyond the default chronological order (for instance according to the REED print collection order or by document type). Fuzzy searching could be explored as a way of doing general text searches in order to deal with the unstandardized spelling in REED records.

**Ontology/indexing**

There will need to be extensive research undertaken to develop a standards-based process for EATS/indexing work in consultation with information science experts especially in regards to a classification for index subjects/themes. Strategies will also need to be developed to implement and manage a cross-collection index that factors in global linked data approaches.

**Data interoperability/linked data**

A major goal for future REED Online resources is a richer integration between EMLoT, Patrons, and eREED collections. This should include provision to make REED data not only available via a web application front end, but more directly for other kinds of data manipulation, eg, through the use of linked data strategies to allow REED data to participate in, say, mashups.

**Online user collaboration**

A REED Online interface should incorporate what are often called Web 2.0 functionalities in order to facilitate scholarly dialogue around and to build on REED scholarship, in other words, to build a virtual scholarly community that can demonstrate the value of REED’s endeavour.

In the same way as an online space is envisioned for REED editors to develop and display their in-progress collections, future development should consider providing an online space for individual and collaborative curation, reuse, addition,
and dissemination of REED and REED-related content on the part of individuals and teams that want to adopt the standards and technologies REED is developing to create new content or new uses of existing content and link it with REED content.

12. Appendices

Appendix A

A Subversion Folder Structure for eREED
Prepared by: John Bradley
Department of Digital Humanities, King’s College London

Subversion’s design encourages one to think of documents/files in nested folders as the model for storing materials. This organization approach has proven to match pretty well the needs and orientation of the eREED work.

Near the beginning of the eREED project DDH proposed a model for discussion of how these folders and files could be organized. The idea was to have a master folder for each collection, and a similar structure within it. Indeed, whenever possible the structure would identical, so that someone who pulled out work stored in Subversion from, say, several years earlier, would be able to find their way around.

Here is the structure of the Fortune Theatre repository, first presented as an hierarchical list, each item being a folder in the proposed common structure, followed by a brief description of the files that go into each folder:

Folder structure

- collection (see note 1 below)
  - submitted (see note 2)
  - tei (note 3)
    - documents (note 4)
    - records (note 5)
    - glossary (note 6)
- supporting (note 7)
  o tools (note 8)
  o generated (note 9)
  o documentation (note 10)

About the folders

1. The *collection* folder acts as the name for the particular repository and holds all the documents relevant to a particular collection. For the Fortune Theatre collection it has been called eREED, but in future it would be most sensible if it was named using a name that reflects the collection it represented. Thus, the Essex collection would have a top-level folder name of *Essex*. In Subversion, the collection folder would be the root folder for a new Subversion repository. Each collection, then, would have its own repository in Subversion.

2. The *submitted* folder holds the Word documents that were received from the editor. As it turned out, it made sense within the eREED workflow that some significant amount of editorial work was done on the Word documents themselves before they were turned into TEI-XML. For this purpose, Subversion’s versioning mechanisms (with Word document tracking) was used to deal with edited versions. Thus, if, say, the editor submitted a document called "dulwich.doc," and some editing was subsequently done on it, the revised version of the document would still be called "dulwich.doc," but Subversion’s versioning mechanisms could then be used to recover the original, unedited, version. An alternative mechanism to handle versioning is to use some sort of subfolder mechanism to store edited versions, but this was not tried out in the eREED prototype.

3. The *tei* folder holds all the TEI documents generated for this collection. Note the subfolders within it, discussed below.

4. The *documents* subfolder holds the TEI-XML containing the documents materials.

5. The *records* subfolder holds the TEI-XML files for the records. If there were a large number of TEI-XML files (not the case with the FTR collection), then it would be possible to add subfolders within the records folder to help to organize them.

6. The *glossary* subfolder could be used to hold the files needed to manage the creation of glossaries. We did not work on the glossary materials in our prototype, so further specifications for the folder have not been worked out as of yet.
7. The *supporting* folder was meant to contain the TEI materials for prose written by the editor and meant to accompany the published collection: things like the Historical Background, the Editorial Procedures, the prose-oriented appendices, etc. Here too, no such material was created for the FTR Prototype.

8. The *tools* folder was meant to hold things like Perl/Python scripts or XSLT stylesheets that were used to process the text from the tei folder. By having a tools folder in the folders managed by Subversion one is able to ensure that a particular version of tools that had been used to process materials from a particular collection would be available into the future.

9. The *generated* folder was meant to hold files that the tools had created when they had been applied. If, for example, there was an XSLT stylesheet that processed the TEI records files to prepare an HTML file that could go back to the editor for checking, this HTML file would be generated into this folder.

10. The *documentation* folder was meant to hold files that documented the processes and decisions made for this particular collection. An example of a file to put here might be a file directory file that outlined how materials for this collection had been organized. Perhaps important email texts that documented exchanges about this collection could be stored here too.
Appendix B
A List and Description of the Files in the FTR Project ZIP Folder

All open-source materials that have been created as a result of the work done on the FTR Prototype digital edition were held in the project’s Subversion repository, and have been packaged up in a ZIP file available from http://www.reed.utoronto.ca/FortuneDirectory.zip. The ZIP file contains the following top level folders:

- **development**: a set of files primarily created by Paul Caton (DDH) to support his work on the TEI encoding.

- **django**: the Fortune Theatre Records Prototype edition is made available as a django driven (https://www.djangoproject.com/) web application. In this folder are the files that configure components of the entire application, in particular for the eats and tmapi (http://www.tmapi.org/) components.

- **fortunetheatre**: contains the record documents (mainly Word and TEI-XML files) created for the FTR record collection.

  Subfolders:
  - documentation: Empty.
  - generated: sample Word file 8, after paleographer’s check
  - submitted: sample Word file 8

  tei:

  - glossary: Empty. Glossaries not created as a part of the FTR project records: The Fortune Theatre records in 9 xml files (files 7 and 8 combined). File naming convention: Midd_Finsbury##.xml

    Supporting:

    - **Symbols.xml**: The Symbols section for the Fortune Theatre records
    - **The Documents Fortune.xml**: The Documents section of the Fortune Theatre records

  tools:

  - add_2nd_stage_markup.xsl
  - add_identifiers.xsl
  - at_codes_to_TEI5.plr
The above three files are the scripts used in the at-codes to TEI conversion. See section 5 above.

**collection_context_template.xml**: The proposed TEI template for The Documents section of a REED collection. This was not implemented.

**convert_from_specific_name_elements.xsl**

**convert_to_specific_name_elements.xsl**

The above two files are a part of the EATS implementation process (?)

**record_head_code_list.xml**: The Code List for the Fortune Theatre record headings

**record_template.xml**: The proposed template for the Records section of a REED collection.

**record_template2.xml**: A cleaned-up version (comments removed) for use in the conversion process.

**documents_template.xml**

**REED_teiHeader_template.xml**

- **kiln**: contains the files necessary to define and operate the kiln web application, including “build files” necessary to construct the kiln application.
Appendix C

Guidelines for REED at-codes Application and TEI P5 Markup
(The Fortune Theatre Records: A Prototype Digital Edition)
Prepared by: Jason Boyd
Centre for Digital Humanities, Ryerson University

CONTENTS

Preconversion

1. Changes to at-codes and at-code application
   Typographical at-codes
   1.1 @e...@e (boldface)
   1.2 @j...@j (bold italic)
   1.3 @k...@k (small caps) (including A and AC symbols)
   1.4 @q...@q (small caps italic)
   Curly brackets
   1.5 {...} (italics)
       1.5.1 Expansions
       1.5.2 {{signed}}
       1.5.3 {{blank}}
       1.5.4 Foliation/editorial record subheading (presentational style)
       1.5.5 Headnotes for ecclesiastical court cases (presentational style)
       1.5.6 Footnotes (presentational style)
   Manuscript features at-codes
   1.6 @i\ (matter in the original added in another hand)
   1.7 @m\...@m \ (centred text)
   1.8 @p (pica space)
   1.9 <t#> (text in columnar format)
   1.10 @bb\, @lb\...@lb \, @rb\...@rb \ (manuscript braces)
   1.11 @p[ (flush right sums in accounts)
   1.12 | (change of folio, membrane, page, or sheet in continuous text)
1.13 {(signed)} (signatures)
1.14 Symbols for Antiquarian Compilations (A) and Collections (AC)

2. TEI mark-up during paleographic checking
   2.1 Marginalia
   2.2 Text in columnar/tabular format; unusual layouts
   2.3 Flush right sums in accounts
   2.4 Manuscript braces
   2.5 Change of folio, membrane, page, or sheet in continuous text
   2.6 Collations
   2.7 Signatures
   2.8 Text blocks in the Record body

3. Marking up the Document “Code List” in TEI

Conversion

4. At-codes and TEI equivalents (for Perl script conversion)
   A. A note on the generation of record xml:ids
   4.1 Special characters and symbols codes & Unicode equivalents
   4.2 Structural codes
   4.3 Headings & Notes
   4.4 Summary of required revisions to the Perl script

Post-Conversion

5. Required TEI markup post-conversion of at-codes to TEI
   5.1 Marginalia
   5.2 Foliation/editorial subheading
   5.3 Collation notes
   5.4 Textual notes/footnotes
   5.5 Editorial/foliation subheading in records text
   5.6 Collation notes
   5.7 Titles of monographs, journals, etc. (in endnotes)
5.8 Links to Document Descriptions, EMLoT, and *Patrons* website

6. Additional TEI markup in the Records

7. Non-Records TEI Mark up
   - 7.1 "The Documents" (document descriptions)
   - 7.2 "Symbols" page
   - 7.3 <teiHeader>

8. Proposed Workflow

9. Alphabetical List of TEI Tags used in The Fortune Theatre Records

1. Changes to at-codes and at-code application

*(Note: sections 1.1 to 1.4 deal with at-codes used primarily in the Code List/Record Headings. See section 3)*

Typographical at-codes

1.1 @e...@e \ (boldface)
   **Deprecated.** In The Records, boldface is used for the year headings that chronologically organize the records. This at-code is added at an in-house processing stage.

On the eREED website, each record is displayed on a separate webpage, so each record, unlike in a REED print collection, has a year heading, which currently is displaying after the MIDDLESEX: FINSBURY heading (on the same line), in a large font.

1.2 @j...@j \ (bold italic)
   **Deprecated.** In The Records, bold italic is used for the document titles in the record heading. This at-code is added at an in-house processing stage.
XSLT can be used to render content within the <title type="record_head">…</title> tags in bold italic. Not currently implemented.

1.3 @k\...@k \ (small caps) (including A and Ac symbols)

Deprecated. In The Records, small caps are used for document repository acronyms in the record heading and for the A and AC symbols. This at-code is added at an in-house processing stage.

XSLT can be used to render content within <idno type="repository">…</idno> tags in small caps. Not currently implemented.

Symbols for Antiquarian Compilations (A) and Collections (AC)

Deprecated. Previous practise was to place these symbols, marked up in the at codes for small caps before the record header (marked up in @h\...\). In the print collections these symbols appear in the margin next to the record title. Given the layout of the current web interface, replicating the print collection practise was not possible (there is no margin), so these symbols are placed after the record title in parentheses. This follows the print collection practice of putting these symbols, in parenthesis and italics, after the titles of works in The Documents section.

These symbols will now be included as a part of the Code List, rather than the Records text, the rationale being that the designation of a document as A or AC should be done at the level of the document rather than the individual transcription, since it pertains (like the rest of the information in the Code List) to all occurrences of the document in the records. See section 3.2.

Not yet implemented with XSLT.

1.4 @q\...@q \ (small caps italic)

Deprecated. In The Records, small caps italic are used in the record heading for a document source that is not a repository but a titled/published work indicated by an acronym (STC, ESTC). This at-code is added at an in-house processing stage.
XSLT can be used to render content within, eg

```xml
<idno type="publication" corresp="#abbr_STC">STC</idno>
<idno type="publication" corresp="#abbr_ESTC">ESTC</idno>
```

tags in small caps italic. Not currently implemented.

**Curly brackets {...}**

1.5 {...} {italic}
In The Records, curly brackets are used in the the record body for expansions (which are presented in italic). Use of italic (italic hand, italic type) in the original record is indicated by textual notes. Elsewhere, curly brackets indicate that the enclosed text should be presented in italic in the typeset version.

1.5.1 Expansions (record body)
Retained. Will be converted to TEI. See section 4.2.10.

1.5.2 {{signed}} precedes signatures (record body)
Depreciated. See section 1.13.

1.5.3 {{blank}} blank space in original where writing would be expected (record body)
Retained. Will be converted to TEI. See section 4.2.14.

1.5.4 Foliation/editorial record subheading (presentational style for date [usually day/month or month or a liturgical date] information and/or a description of the record excerpt)
Depreciated. These will be stripped out in conversion.

Since this material is always presented in parentheses, XSLT can be used to render content within `<head type="sub">...</head>` and `<ab type="subhead">` tags that are within parentheses (including the parentheses themselves) in italic. Not currently implemented.
1.5.5 Headnotes for ecclesiastical court cases (presentational style)

**Deprecated.** These will be stripped out in conversion.

XSLT can be used to render content within `<note type="ecclesiastical">...</note>` tags in italic. Not currently implemented (no ecclesiastical court cases in Fortune Theatre record set).

1.5.6 Footnotes (presentational style)

**Retained.** Footnotes are rendered in italics, except for the word or phrase being glossed and any provided word or phrase that is being suggested as a substitute (eg, a correction), which are presented in roman font, eg, finding: 

*for* finding him

This is currently causing problems at the conversion stage, since footnotes can include curly brackets that indicate expansions as well as text to be rendered into italic for presentation purposes. The Perl script is sometimes converting these curly brackets into `<hi rend="italic">...</hi>` and `<ex>...</ex>` in an incorrect way (eg, expansion brackets are sometimes converted as `<hi rend="italic">...</hi>` and text to be italicized as `<ex>...</ex>`). Currently, these are being manually corrected post-conversion. Revisions to the Perl script might prevent this problem in future. A possible solution is to mark up the word/phrase glossed and the gloss itself separately (using `<term>` and `<gloss>` tags), which could enable the gloss to be italicized (although this would require text in the gloss that is be in roman to be marked up as well, using `<term>`). This is needed only if one wants to output the records in a format that replicates the print collections.

**Manuscript features at-codes**

1.6 @i\ (matter in the original added in another hand)

**Revised.** A new closing code, @i \, must be added to all additions by another hand.

*To be implemented during the paleographic checking stage.*
1.7 @m\...@m \ (centred text)

Revised. These codes should enclose independently-centred units of text. For example, if a unit of text has five lines, each of which is to be independently centred, then each line should be enclosed with the @m\...@m\. If the five lines are to be centred as a unit, a single set of codes should enclose these five lines. If the first two lines are to be independently centred from the remaining three lines, a set of codes should enclose the first two lines and another set of codes the remaining three lines.

*To be implemented during the paleographic checking stage.*

Examples:
Rec02: a centred heading broken over two lines with a <lb/>; each line is separately centred.

Note: The eREED platform automatically centres every line separately.

1.8 @p (pica space)

Deprecated. This has been used in The Records for laying out columnar accounts and indenting paragraphs (usually in continuous prose). Columns and indenting will now be handled using TEI. See sections 2.2 and 2.8.

1.9 <t#> (text in columnar format)

Deprecated. This will now be handled using TEI.

*To be implemented during the paleographic checking stage. See section 2.2.*

1.10 @bb\, @lb\,...@lb\, @rb\,...@rb \ (manuscript braces)

Deprecated. This will now be handled using TEI.

*To be implemented during the paleographic checking stage. See section 2.4.*

1.11 @p[ (flush right sums in accounts)

Deprecated. This will now be handled using TEI.

*To be implemented during the paleographic checking stage. See section 2.3.*

1.12 \ (pipe) (change of folio, membrane, page, or sheet in continuous text)
Deprecated. These changes will be marked up using TEI.  
To be implemented during the paleographic checking stage. See section 2.5.

1.13 {(signed)} (signatures)  
Deprecated. Signatures will be marked up using TEI.  
To be implemented during the paleographic checking stage. See section 2.7.

2. TEI mark-up during paleographic checking

For revised at-code application to be implemented at the paleographic checking stage, see section 1.6-7.

2.1 Marginalia

Current practise: The word or start of phrase to which the marginal note refers is indicated by a pencilled-in “M” enclosed in a circle place immediately after the word/phrase – this is the line on which the marginale is to start in the typeset version. The marginalia are placed before the record text in @l\...@l \ (left marginalia) or @r\...@r \ (right marginalia) codes.

New practise: The circled M to be replaced with the tag `<anchor corresp="#" />`. The corresp value is added later. See 4.2.5.

Given the oddity of the way this displays on the eREED interface, perhaps this tag should be added BEFORE the relevant word/start of phrase. Generally, the alignment of marginalia and record body requires finessing.

2.2 Text in columnar/tabular format; unusual layouts

Current practice: Currently, columnar/tabular text and text in unusual layouts is being marked up using the tab `<t>` code, with a number indicating by how many tabs the text is to be indented (eg, `<t1>`, `<t2>`, etc.).
New practise: All such text should be marked up using TEI P5’s <table> tag set. This tag set is implemented on a record-by-record basis. In general, however, text that is not in a straightforward grid format (which is easily marked up using <table> tags), can usually be dealt with using the cols and rows attributes within <cell> tags (ie, for textual units that violate a grid format by running over multiple columns or rows. The general approach is to determine what the most common textual unit of the relevant text is (ie, the units that would be enclosed in <cell> tags), and then used the attributes mentioned above for text units that range over cells.

The implementation of the TEI <table> tag set and how this content is represented on the website requires a great deal of finessing. A cursory survey of how the content marked up using the <table> tags in Midd_Finsbury01.xml revealed a range of display issues that will have to be addressed. A thorough audit of how the TEI is being processed and outputted on the website will have to be done and there will be revisions More specific guidelines may emerge out of the process.

2.3 Flush right sums in accounts

Current practise: The sum in preceded by the at-code @p[.

New practice: Using <table> tags, the sums can be put in a <cell> with rend="right".

A further refinement to be added would be a "right_bottom" rend value, so that the sum is flush with the last line of the description of that sum in the left column. The website currently displays these sums flush with the first line.

2.4 Manuscript braces

Current practise: at-codes are used to enclose text that is braced.

New practise: Using TEI <table> tags, braces are given in their own column, and, within the <cell> tag, role="braced_right" or "braced_left" are used, with rows="#" specifying the range of the brace. See rec11, and rec12 for examples of manuscript brace markup.
2.5 Change of folio, membrane, page, or sheet in continuous text

Current practice: A pipe (|) is used to indicate the change.

New practise: In place of the pipe, the change should be explicitly marked up using <pb type="..." n="..." />

Example:

<pb type="folio" n="41v" />

Note: All changes of folio have to be explicitly marked up as above, including the folio, etc., on which a record excerpt starts (which is contained in the record subheading). The <pb/> tag should be inserted immediately after the opening <div type="transcription"> tag. This should even be done, but has not currently been done, for single sheets and membranes. If the information in the subheading could be automatically marked up in TEI, this additional work would not have to be done (see section 4.3.4).

Note: editorial subheadings that appear within records text rather than with the main record heading, will have to be edited post-conversion because <head> is invalid within <div>. See section 5.5. This might be addressed by revising the Perl script.

2.6 Collations

Since no collations were encountered in the Fortune Theatres Record set, what follows is a proposed but not yet implemented revised practice.

Collation notes (which, like textual notes, appear at the bottom of the page in print collections), begin with a list of documents (designated by their repository and shelfmark) with which the record is being collated and the specific pages, etc, from those documents consulted for the collation. Each collation source is labelled with a
different capital letter. This is followed by a word/phrase-by-word/phrase

collation consisting of:

a) the line number of the word/phrase being collated in the record above;
b) the word/phrase being collated (followed by a closing square bracket)
c) the collated word(s)/phrase(s), each followed by the capital letter label of the
relevant document source.

Current practice: Numbered collation notes are submitted by editors in a separate
file. Recently introduced in-house practice incorporates the collation notes into the
records text after the word or phrase to which they refer, enclosed in @c...@c \codes, during the paleographic checking stage.

Normally, a project would automate collation after having marked up up the texts
being collated in TEI. The suggested revisions below are based on the assumption
that REED is not going to mark up multiple witnesses to a record transcription and
then automate the collation, but instead continue to use collation notes.

Suggested revision: Instead of using the at-codes, use TEI.

<ref target="#c1" type="collation_anchor">Prohibemus</ref>
<note xml:id="c1" type="collation">
  <list>
    <item>Precipimus etiam<ptr target="#B"/></item>
    <item>precipimus<ptr target="#C"/><ptr target="#E"/></item>
  </list>
</note> vt cimieria sint bene…

The <ptr> tags point to the relevant document description in The Documents
section. The only information that is not contained in the document description that
is contained in the collation note is the specific page or page range, etc, consulted
for the collation. This information could be placed in the document description
(using <locus>).

Another possible solution is to enclosed the collated words/phrases in <seg> tags
with ids (xml:id or n) and with either the sameAs or exclude attribute to associate the <segs> and enclose them all in <choice type="collation">…</choice> tags.

2.7 Signatures

Signatures should be enclosed in <seg type="signed">…</seg> or, for personal marks, <seg="signed_mark">…</seg> tags. See also section 2.7.

2.8 Text Blocks in Record Body

The body of a record (including marginalia) and endnotes should be enclosed in one or more sets of <ab>…</ab> (anonymous block) tags, depending on how many distinctive blocks of text constitute it. The <ab> tags should always include the type attribute (see section 9 for a list of values).

Although recommended by DDH, it remains unclear what the value is of using <ab> in such an extensive way. Also it is unclear whether to use both <ab> and <p> (paragraph) tags, or make paragraphs (both indented and exdented) a value of an <ab> type attribute. These issues require further discussion.

3. Marking up the Code List in TEI

The Code List will be marked up in TEI. XSLT will be used to merge the code list with the records text for output (formerly done with a script written in C by Abigail Ann Young).

The Code List will consist of two lists: one for the codes that indicate the section under which the records appears (eg, a parish or borough), another for the list of documents.

3.1 Records Division

Example:
<list n="records_div">
  <item n="parish">
    <abbr>FIN</abbr>
    <expan>FINSBURY</expan>
  </item>
</list>

3.2 List of Documents (Code List)

<list n="documents">
  ...
</list>

  a)  Enclose each document in <item>…</item>
  b)  Enclose the 4-letter code in <abbr>…</abbr>
  c)  Enclose the rest of the document information in <expan>…</expan>
  d)  Enclose the document title in <title type="record_head">…</title>
  e)  Enclose the rest of the information in one or more <idno type="">…</idno> tags.
      Printed work: <idno type="publication">…</idno>
      Repository and shelfmark:
          <idno type="repository">…</idno>
          <idno type="shelfmark">…</idno>
      STC number (also for Wing):
          <idno type="publication">STC</idno>
          <idno type="STC_number">…</idno>
  f)  If you use more than one set of <idno> tags, enclose these in <seg type="">…</seg>. For type values, see section 9.
  g)  For content in <idno> tags that are abbreviations, add a corresp="#" element to the opening <indo> tag (after the type element). For corresp values, see section 9.
  h)  If the document in question is a Antiquarian Compilation (A) or Antiquarian Collection
      (AC), add <note="antiquarian_source" corresp="#abbr_A[or
Example:

```xml
<list n="documents">
  <item>
    <abbr>ABDC</abbr>
    <expan>
      <title type="record_head">Extracts from the Audit Books of Dulwich College</title>
      <seg type="repository_shelfmark">
        <idno type="repository" corresp="#abbr_BL">BL</idno>:</idno>
        <idno type="shelfmark">Additional MS. 29479 A</idno>
      </seg>
    </expan>
  </item>
  ...
</list>
```

4. **At-codes and TEI equivalents (for Perl script conversion)**

A. A note on the generation of record xml:ids

The only manual entry being done during the conversion process is adding xml:ids to the <text> tag. Process for creating the xml:id:

1. Start by numbering the record: "rec##_. Records are being numbered incrementally starting from Midd_Finsbury01.xml;
2. Use the unexploded heading, replacing "!" with ",", ie, FIN!1608!LWNE becomes FIN_1608_LWNE_. For slash years, use "s" for the forward slash, eg, 1609s10;
3. Add the subheading information (use "ss" for "single sheet", "smb" for "single membrane").
Example:

xml:id="rec09_FIN_1608_LWNE_f1"

Whether this process can be automated should be an area for future exploration.

4.1 Special characters and symbols codes & Unicode equivalents

This section includes the REED at-codes for special characters and symbols that are included in the Records text and that can be simply converted to their Unicode equivalents (HTML Entity hex encoding used). Source for hex encodings: http://www.fileformat.info/info/unicode/char/a.htm

4.1.1 ae in ligature
   @ ae                   &#x00E6;
   @ AE                   &#x00C6;

The first at-code is not being converted by the Perl script.

4.1.2 capitulum
   @C                    &#x204C; (as close as we can get for now)

4.1.3 eth
   @d                    &#x00F0;

4.1.4 eng
   @n                    &#x014A;
   @N                    &#x014B;

4.1.5 oe in ligature
   @oe                   &#x0152;
   @OE                   &#x0153;
4.1.6 paragraphos
@P &amp;#x204B;

4.1.7 thorn
@th &amp;#x00FE;
@TH &amp;#x00DE;

4.1.8 Middle Welsh w
@v &amp;#x1EFD;

4.1.9 wynn
@y &amp;#x01BF;

4.1.10 yogh
@z &amp;#x021D;
@Z &amp;#x021C;

4.1.11 sterling symbol
@$ &amp;#x00A3;

4.1.12 acute accent with vowels
@’a, e, i, o, u &amp;#x00E1; 00E9 00ED 00F3 00FA
@’A, E, I, O, U &amp;#x00C1; 00C9 00CD 00D3 00DA

4.1.13 grave accent with vowels
@,a, e, i, o, u &amp;#x00E0; 00E8 00EC 00F2 00F9
@,A, E, I, O, U &amp;#x00C0; 00C8 00CC 00D2 00D9

4.1.14 circumflex accent with vowels
@^a, e, i, o, u &amp;#x00E2; 00EA 00EE 00F4 00FB
@^A, E, I, O, U &amp;#x00C2; 00CA 00CE 00D4 00D8

4.1.15 c-cedilla
@?c &amp;#x007;
4.1.16 punctus elevatus
@; &punctelev; (not yet approved by Unicode: see <http://www.mufi.info/specs/MUFI-Alphabetic-3-0.pdf> p 146 and p 11 for details)

4.1.17 middle dot
@* &x00B7;

4.1.18 exclamation mark
@! &x0021;

4.1.19 caret
^ &x28C; (Unicode name: “Latin Small Letter Turned V”)

The Unicode value initially provided was the value for open square bracket (&x005B;) The Perl script will have to revised to correct this error.

Since the caret mark appears on the standard keyboard (shift+6), is the use of the Unicode value necessary?

4.1.20 em-dash
&x2013;

4.1.21 ellipsis
... &x2026;
Not currently implemented in the Perl script: the ellipsis only applies to footnotes (<note type="foot">...</note>), where ellipses sometime appear; the Perl script currently
converts this to <gap/>. It may be fine to use <gap/> and convert it to Unicode than to have to replace <gap/> with the Unicode value for ellipsis.

4.1.22 ampersand
&

If other diacritics (e.g., umlaut (@:), tilde (@"), or macron (@-)) come up we can deal with them on a case-by-case basis. There are also a few codes not included here: we devised them for a document in Wales using an eccentric orthography. It does not seem likely that we would need them for a future collection and when we come to prepare Wales for web publication we can deal with them on a case-by-case basis.

4.2 Structural codes

This section includes at-codes and other markup that concerns the structure, layout and original “markup” in the Records text.

4.2.1 text written above the line  [TEI P5 3.4.3, 11.3.4]
@a\ <add rend="above">  
@a \ </add>

These tags can be converted to Unicode equivalents for upper half

brackets (&#x2E22; &amp;#x2E23;) by XSLT

Attribute “rend” should be changed to “place” in the Perl script.

4.2.2 text written below the line  [TEI P5 3.4.3, 11.3.4]
@b\ <add rend="below">  
@b \ </add>

These tags can be converted to Unicode equivalents for lower half brackets (&#x2E24; &amp;#x2E25;) by XSLT
Attribute “rend” should be changed to “place” in the Perl script.

4.2.3 exdented text paragraph (first line flush left, subsequent lines indented)
[TEI P5 3.1]
@g\ <p rend="exdented">
@g \ </p>
Formerly the at-code for Greek text. This was redefined to code
exdented paragraphs in printed performance texts in
Inns of Court.

May be changed to <ab type="p_exdented">…</ab>.

4.2.4 matter in the original added in another hand [TEI 11.4.1]
@i\ <add><handshift/>
@i \ </add>
These tags can be converted to Unicode equivalents for interpolation “bubbles” (&#x00B0; &deg; – actually the “degree” symbol) by XSLT. Note that the closing at-code is a new addition to the at-codes.

4.2.5 marginalia [TEI P5 3.8.1]
left
@l\ <note type="marginal" place="margin_left" xml:id=""/>
@l \ </note>
right
@r\ <note type="marginal" place="margin_right" xml:id=""/>
@r \ </note>
The enclosed text can have an @ added before it (by XSLT), using the Unicode for the “registered” symbol (&#x00AE; &reg;). The xml:id values will be added post-conversion. See section 5.1.
4.2.6 centred text  [TEI P5 3.3.2.2]
@m\ <hi rend="center">@m 
</hi>

4.2.7 indented paragraph (was @p, pica space)
@p\ <p type="indented">
@p \ <!-- May be changed to <ab type="p_indented">…</ab>. -->

4.2.8 superscription  [TEI P5 3.3.2.2]
@s\ <hi rend="superscript">@s \ </hi>

4.2.9 hard carriage return (new line)  [TEI P5 3.10.3]
! <lb/>

4.2.10 expansions  [TEI P5 11.3.1]
{...} <ex>...</ex>
Text between these tags can be converted into italics by XSLT.
Note that these codes are actually the markup for italic (see 1.5) and they are used for that purpose in the records heading and subheading and footnotes. The conversion script should convert {...} in the record text proper to <ex>...</ex>.

4.2.11 cancellation in the original  [TEI P5 11.3.4]
[...] <del>...</del>
This can be converted to square brackets by XSLT.

4.2.12 lost or illegible letters in the original  [TEI P5 11.5]
<.> <damage><gap unit="chars" extent="1"/></damage>
The last may be accompanied by a textual note describing the extent of the damage when the damage is larger than 3 characters. These can be converted to diamond brackets and dots by XSLT. See also Section 6. Attribute “extent” should replace “quantity” in current Perl script.

4.2.13 ellipsis of original matter  [TEI P5 3.4.3]

Example...

This can be converted to the Unicode equivalent for ellipsis (&#x2026;) by XSLT. Attribute and value reason="omitted" needs to be added to the Perl script.

4.2.14 blank in original where writing would be expected  [TEI P5 11.6.1]

Example:

This can be converted to (blank) by XSLT. Currently, the XSLT does not italicize (blank).

4.3 Headings & Notes

4.3.1 collation note
@c\#\n@c\...@c \n<note xml:id="" type="collation">...</note>

See 2.6.

4.3.2 textual note/footnote
@f\#\n@f\...@f 
<note xml:id="" type="foot">...</note>

Numbered textual notes are submitted by editors in a separate file. Recently introduced in-house practice
incorporates the footnotes into the records text after
the word or phrase to which they refer, enclosed in
@f\...@f \ codes, during the paleographic checking
stage. See also section 6.

4.3.3 document heading
@h\...\n
<head><placeName>…</head>

Example:

@h\FIN!1571!ABDC\!

The Perl script converts @h\ to <head type="main"> <placeName> . The first ! (hard
return) is converted to </placeName><date> . The second ! is converted to
</date> . The final \! is converted to </head> </lb> :

Post-conversion:
<head type="main">
<placeName>FINSBURY</placeName>
<date>1571</date>
</head><lb/>

4.3.4 foliation/editorial subheading
@w\...\n
<head type="sub">…</head>

For post-conversion markup of subheadings that
appear within records, see section 5.5.

This proposed conversion (akin to the one proposed
and implemented above) has not been implemented:

Example:

@w\ f. 40* {{(12 January) (Fortune: Warrant from the
Lord High Admiral, the Earl of Nottingham for its building)}

The conversion process could parse the content contained within the parentheses for the presence of at least one of the twelve months, and if the condition was true the parentheses could be converted to <date>…</date>. Otherwise, the parentheses would be converted to <note type="record_desc">…</note>.

Example converted:

<head type="sub">
  <pb type="" n=""/>f. 40*
  <date>(12 January)</date>
  <note type="record_desc">(Fortune: Warrant from the Lord High Admiral, the Earl of Nottingham for its building)</note>
</head><lb/>

Note: The manual addition of the asterisk to indicate an associated endnote is now redundant.

4.3.5 names of court personnel for ecclesiastical court cases
@x\...@x \ <note type="eccles_court">...</note>

4.4 Summary of required revisions to the Perl script

4.4.1 @ ae to convert to Unicode equivalent
4.4.2 caret to convert to correct Unicode value (or not be included in conversion at all)
4.4.3 within <note type="foot">, convert series of three period to Unicode for ellipses (or leave as is)
4.4.4 @a \ to convert to <add place="above">
4.4.5 @b \ to convert to <add place="below">
4.4.6 for damaged text (see section 4.2.12), <gap/> attribute “quantity” changed to “extent”
4.4.7 <gap/> to <gap reason="omitted"/>
4.4.8 <gap unit="" quantity=""/>: “quantity” changed to “extent”
4.4.9 <note type="marginale">: “marginale” to “marginal” [although “marginale” is the spelling in REED collections: see Symbols page]
4.4.10 value “marginLeft” to “margin_left” (usually marginal notes)
4.4.11 square brackets in <head type="sub"> and <ab type="subhead"> should not be converted to <del>…</del>
4.4.12 Additional conversion of subheading (see section 4.3.4)?

5. Required TEI markup post-conversion of at-codes to TEI

5.1 Main record heading: <date>

Always use new style calendar dates.

For single dates, use:
<date when-iso="yyyy-mm-dd">…</date>

For date spans, use (for events etc, that occurred over span of time):
<date from-iso="yyyy-mm-dd" to-iso="yyyy-mm-dd">…</date>
Use this also when you have a record with multiple subheadings with dates that represent a date range; enter this date range in the main heading <date> and add <date> tags to the subheadings.

For events, etc that occurred at some point with a date range

<date notBefore-iso="yyyy-mm-dd" notAfter-iso="yyyy-mm-dd">…</date>

For circa dates, type="circa" is added to the opening <date>.
5.2 Initial page breaks

A <pb> tag is added immediately after the <div type="transcription"> tag. Exception: single sheets (no tag needed).

Single sheets should be marked up using <pb>; but, if the record subheadings can be automatically marked up, then it would not be necessary to do this manually.

In the <pb/> tag, “type” and “n” (number values) attributes have to be added, along with the relevant values.

Example:

f 40

<pb type="folio" n="40"/>

5.3 Marginalia

The xml:ids of marginalia (added during the conversion process), eg “rec10_mn50” have to be added to the relevant <anchor corresp="#"/> tags (indicating where the marginalia will appear in relation to the body of the record. See sections 2.1 and 3.2.5

5.4 Textual notes/footnotes

Place <ref> tags around the word or phrase to which the footnote is referring and add the xml:id value in the note to the target attribute.

Example:

<ref target="#rec15_fn01" type="foot_anchor">Prohibemus</ref><note xml:id="rec15_fn01" type="foot">...<note>
For the markup of the content of textual notes, see section 6.

5.5 Editorial/foliation subheading within records text

Because <head> is not valid within records text (more precisely, within records text enclosed in <ab> tags), the <head type="sub">... </head> has to be replaced with <ab type="subhead">... </ab>. The <pb/> relevant to the subheading is added after this <ab> tags.

The Perl script could be modified to convert all @w\...\ within <div type="transcription">... </div> tags to <ab type="subhead">... </ab> tags.

5.6 Collation notes

Not encountered in the Fortune Theatre Records.

5.7 Titles of monographs, journals, etc. (in endnotes)

<title type="monograph">... </title>

For relevant values, see section 9.

5.8 Links to Document Descriptions, EMLoT, and Patrons website

These are added between the closing </body> and </text> tags of an individual record.

Example:

<link type="document_desc" target="#Midd_WHLA"/>
<link type="emlot" target="#"/>
<link type="patrons" target="http://link.library.utoronto.ca/reed/patron.cfm?PeopleListID=853"/>

6. Additional TEI markup in the Records
Additional markup was largely focussed on TEI-izing aspects of the records addressed in the textual notes. Some exploratory work was done with TEI mark up relating to shifts in language (see example from rec99, below), and with the calendar attribute for <date>. Further exploration of mark up for dating content in records and in endnotes, and the integration of the "Saint’s Days and Festivals" apparatus should be explored. Marking up bibliographic references in endnotes might also be explored.

Future exploration of how textual notes are to be done in relation to TEI and digital collections is needed. The automated generation of some textual notes using this markup should be explored.

The examples below are not exhaustive; they have been chosen either because they are representative or because they are atypical.

Specific, simple features:
- Rec05: document damaged, portions of text physically missing
- Rec39: catch-words (in printed texts)
- Rec07: dating notes
- Rec07: dittography
- Rec77: text meant to be cancelled but left uncancelled
- Rec02: personal marks (in lieu of signature)
- Rec168: use of REED transcription special characters (eg, [ ]) in original

Overlapping features (palimpsest): <subst>
- Rec71: text written over erased text
- Rec17: word written over illegible word
- Rec03: text written over illegible text
- Rec71: text written over legible text
- Rec71: erased text with line fillers

Editorial interpretation (regularization, correction): <choice>
- Rec01: regularizing punctuation
Rec09: scribal error/omission
Rec59: misspelled word (omitted letter)
Rec61: missing letters supplied
Rec69: a probable correction
Rec84: possible ditto-graphy or wrong/misspelt word
Rec134: missing mark of abbreviation; unexpanded word
Rec197: regularization of historical proper names

Editorial supplementation: <supplied>
Rec202: tacit content supplied for clarification
Rec203: content supplied from another place

Atypical blanks: <space/>
Rec74: an unfilled space in the text
Rec128: filled in blank

Unusual text layout: Rec194

File 1

Rec01: regularizing punctuation

<ref target="#rec01_fn01" type="foot_anchor">
  <choice>
    <orig>.</orig>
    <reg>(</reg>
  </choice>
</ref>

<note xml:id="rec01_fn01" type="foot">: <hi rend="italic">for opening parenthesis</hi></note>

While this could have been tagged with <sic> and <corr> instead of <orig> and <reg>, given the unsystematic and inconsistent use of punctuation in the period,
this seemed to more accurately represent the nature of the editorial intervention.

Rec02: personal marks (in lieu of signature)

```xml
<name key="http://reed.utoronto.ca/django/eats/entity/927/"
type="man">will<ex>ia</ex>m
<ref target="#rec02_fn04" type="foot_anchor">
  <seg type="signed_mark">WG</seg>
</ref>
<note type="foot" xml:id="rec02_fn04">WG: <hi rend="italic">Garruld’s personal
  sign</hi></note> Garruld</name>
```

In some cases a visual description of personal mark is provided, which is best dealt with by a textual note.

Rec03: text written over illegible text

```xml
<ref target="#rec03_fn14" type="foot_anchor">
  <subst>
    <del type="overwritten">[c/]</del>
    <add place="over">y</add>
  </subst>
</ref>
<note xml:id="rec03_fn14" type="foot">ys: y <hi rend="italic">written over
  another letter</hi></note>
```

If the text written over was legible or provided, it would be enclosed in <del> tags, rather than an empty <del> tag as above. See also File 1: Rec17

Rec05: document damaged, portions of text physically missing

```xml
<ref target="#rec05_fn16" type="foot_anchor">
  <damage agent="tearing">gap unit="mm" extent="35"/></damage>
</ref>
```
<note xml:id="rec05_fn16" type="foot">
    <damage agent="tearing"><gap unit="mm" extent="35" /></damage>: 
    <hi rend="italic">35mm lost due to tear</hi>
</note>

Rec07: dittography

<name key="http://reed.utoronto.ca/django/eats/entity/4254/" type="man">Thomas Sansburie</name><ref target="rec07_fn22" type="foot_anchor"><surplus reason="dittography">Sansburie</surplus></ref>
<note xml:id="rec07_fn22" type="foot">Sansburie Sansburie: <hi rend="italic">dittography</hi></note>

Rec07: dating notes

<ref target="#rec07_fn25" type="foot_anchor">
    <date when-iso="1610-06-24">midsummer day next ensuing</date>
</ref>
<note xml:id="rec07_fn25" type="foot">midsummer...ensuing: <hi rend="italic">24 June 1610</hi></note>

Rec09: scribal error/omission

<ref target="#rec09_fn36" type="foot_anchor">
    <choice>
        <sic>finding</sic>
        <corr cert="high" resp="#JF">finding him</corr>
    </choice>
</ref>
<note xml:id="rec09_fn36" type="foot">finding: <hi rend="italic">for</hi> finding him</note>

This could also have been dealt with by tagging "him" using <supplied>; however, the above tagging seemed appropriate given that the editorial intervention is
correcting a clear error. <supplied> is typically used "because the original cannot be read because of physical damage or loss to the original" (TEI P5 Guidelines). For a use of <supplied>, see

Rec17: word written over illegible word

```xml
<ref target="#rec17_fn57" type="foot_anchor">
  <subst>
    <del type="overwritten"><w/></del>
    <add place="over">had</add>
  </subst>
</ref>
<note xml:id="rec17_fn57" type="foot">had: <hi rend="italic">written over another word</hi></note>
```

The <w/> is included to indicate that it has been determined the overwritten text is an entire word.

File2

Rec39: catch-words (in printed texts)

```xml
<fw type="catch" place="foot">The</fw>
  <pb type="signature" n="B3v"/>
  <ref target="#rec21_fn22" type="foot_anchor">The</ref>
  <note xml:id="rec21_fn22" type="foot">The: <hi rend="italic">also printed as a catchword at the foot of B3</hi></note>
```

Rec59: misspelled word (omitted letter)

```xml
<ref target="#rec42_fn25" type="foot_anchor">
  <choice>
    <sic>beweene</sic>
    <corr>betweene</corr>
  </choice>
</ref>
```
File 3

Rec61: missing letters supplied

Some REED editors have wanted to supply missing letters which they have judged to be obvious. This is an example of how they could do this. The user could potentially see the record displayed without or with the supplied letters, in the latter case, with the information of who supplied the letters and their degree of certainty.

Rec69: a probable correction
</sic>
<corr cert="medium">your worshipps</corr>
</choice>
<note xml:id="rec10_fn21" type="foot">y<ex>ou</ex>r: <hi rend="italic">probably for</hi> your worshipps</note>

The set cert value of "medium" is used as the equivalent of "probably".

Rec71: text written over erased text

<subj>
   <del type="erasure"/>
   <add place="over">
      <ref target="#rec12_fn26" type="foot_anchor">Twentyfoure</ref>
   </add>
</subj>
<note xml:id="rec12_fn26" type="foot">Twentyfoure: <hi rend="italic">written over erasure</hi></note>

Rec71: text written over legible text

<ref target="#rec12_fn27">wi
   <subj>
      <del type="overwritten">t</del>
      <add place="over">c</add>
   </subj>h
</ref>
<note xml:id="rec12_fn27" type="foot">wich: c <hi rend="italic">written over</hi>t</note>

Rec71: erased text with line fillers

<ref target="#rec12_fn35" type="foot_anchor">
   <del type="erasure" unit="mm" extent="14">
<gap unit="chars" extent="3" rend="line_fillers"/>
</del>
</ref>
<note xml:id="rec12_fn35" type="foot"><gap unit="chars" extent="3"/>
<hi rend="italic">erased space of 14mm with fillers</hi></note>

Rec74: an unfilled space in the text

<ref target="#rec15_fn111" type="foot_anchor">
    <damage type="natural_hole">
        <space rend="covered" unit="mm" extent="15"/>
    </damage>
</ref>
<note xml:id="rec15_fn111" type="foot">were: <hi rend="italic">followed by a space of 15mm by the insertion of a small piece of parchment to cover a natural hole in the membrane</hi></note>

Rec77: text meant to be cancelled but left unc cancelled

<ref target="#rec18_fn146" type="foot_anchor">
    play<surplus reason="unc cancelled">h</surplus><del><gap agent="erasure" unit="chars" extent="2"/></del>
</ref>
<note type="foot" xml:id="rec18_fn146">play<surplus reason="unc cancelled">h</surplus><del><gap agent="erasure" unit="chars" extent="2"/></del>: <hi rend="italic">for</hi> play</note>

Rec84: Possible dit tography or wrong/misspelt word

<ref target="#rec25_fn210">as</ref>
<choice>
<orig>
    <surplus reason="dittography" cert="unknown" resp="#JF">as</surplus>
</orig>
<corr cert="unknown" resp="#JF">is</corr>
</choice>
</ref>
<note xml:id="rec25_fn210" type="foot">as as: <hi rend="italic">dittography; or second</hi> as <hi rend="italic">for</hi> is</note>

<sic> could be used in place of <orig>, but <orig> is neutral as regards to whether what is enclosed is an error or not.

File 4

Rec99: words and phrases in a language different from the main language of the text

<ref target="#rec06_fn02" type="foot_anchor">
   <foreign xml:lang="la">Qui quondam Lixa, Lanista.</foreign>
</ref>Qui&lt;gap&gt;Lanista: <hi rend="italic">'He who was once a sutler, is (now) a trainer'</hi></note>

File 5

Rec128: filled in blank

<ref target="#rec13_fn33" type="foot_anchor">
   <add place="over">Com</ex>itatu</ex> Midd&lt;ex&gt;lesexie</ex>
   <space unit="mm" extent="35" rend="line_fillers"/>
</add>
</ref>
<note xml:id="rec13_fn33" type="foot">Com&lt;ex&gt;itatu</ex> Midd&lt;ex&gt;lesexie</ex>: <hi rend="italic">inserted into space left blank; 35mm filled with line-fillers</hi></note>

This could be enclosed in <seg type="filled_blank">…</seg> tags if the markup
above is not sufficient to indicate that the text and line fillers were inserted into a
former blank space.

Rec134: missing mark of abbreviation; unexpanded word

<ref target="#rec19_fn45">
  <choice>
    <sic rend="missing_abbr_mark">siliter</sic>
    <corr>similiter</corr></choice>
</ref>
<note xml:id="rec19_fn45" type="foot">siliter: <hi rend="italic">for</hi>
similiter <hi rend="italic">;</hi> mark of abbreviation missing</note>

File 6

Rec168: use of REED transcription special characters in original

<ref target="#rec20_fn27" type="foot_anchor">
  <orig>[</orig>Fortune<orig>]</orig>
</ref>
<note xml:id="rec20_fn27" type="foot">[Fortune]: <hi rend="italic">square
brackets in original transcript</hi></note>

File 9

Rec194: Unusual text layout

<ab type="initial_letter">
  <w n="This" part="l">T</w><anchor corresp="#rec19_mn126"/>
  <ref target="#rec19_fn127" type="foot_anchor">
    <note xml:id="rec19_mn126" place="margin_right"><hi rend="underline">1608</hi></note></ref>
  <note xml:id="rec19_fn127" type="foot">1608: <hi rend="italic">underlined
and written to the right of the upright of the large and highly elaborated</hi> T <hi>
This

Indenture made

Countie of

South

Hampshire

Countie of

South

Hampshire

Another

Another: ie, another

warrant

warrant
Rec203: content supplied from another place

<ref target="#rec02_fn06" type="foot_anchor">
  <damage>
  <gap unit="chars" extent="3" agent="tear"/>
  <supplied reason="illegible" source="#interrogatory">one Thomas Grymes</supplied>
  </damage>
</ref>

<note xml:id="rec02_fn06" type="foot">
  <damage>
  <gap unit="chars" extent="3" agent="tear"/>
  </damage>: <hi rend="italic">letters lost along tear at crease; Interrogatory reads</hi> one <name key="http://reed.utoronto.ca/django/eats/entity/23667/" type="man">Thomas Grymes</name>
</note>

7. Non-Records TEI mark up

7.1 "The Documents" (document descriptions)

A. A note about naming conventions for xml:ids for document descriptions

Document description xml:ids were created from the four-letter code for the document (from the “Code List”) prefaced by an abbreviation of the title of the REED collection to which they belong and an underscore. For The Fortune Theatre Records, the abbreviation was “Midd” (for Middlesex). This abbreviation is necessary to ensure that collection that might use the same code can be differentiated. These xml:ids are the target for the <link type="document_desc"> in the records.
Example:

```xml
<div type="the_documents">
  <head type="document_auspices">Households</head>
  <head type="main">BERTIE OF GRIMSTHORPE</head>
  <div type="document_desc" xml:id="Linc_KBHA">
    <head type="sub">Richard and Katherine Bertie's Household Account</head>
    <note>Richard Bertie (1517-82) was married to Katherine (1518/19-80), dowager duchess of Suffolk and Baroness Willoughby de Eresby. The account runs from December 1560 to September 1562.</note>
    <msDesc>
      <msIdentifier>
        <settlement>Lincoln</settlement>
        <repository>Lincolnshire Archives</repository>
        <idno>1 ANC 7/A/2</idno>
      </msIdentifier>
      <msContents>
        <textLang>English</textLang>
      </msContents>
      <physDesc>
        <p>paper; 386 leaves; 305mm x 210mm (variable); modern foliation 1-193; booklets stitched together, original parchment cover.</p>
      </physDesc>
    </msDesc>
    <history>
      <origin>1560-2</origin>
    </history>
  </div>
  <link target="#record1 #record2"/>
</div>
```

The xml:ids of the records transcribed from a document are entered as values of the target attribute in `<link/>`, prefaced by “#.” Multiple xml:ids can be entered as target values.

7.2 "Symbols" page

Example:
<text type="abbreviations_symbols">
  <body>
    <head>Symbols</head>
    <div type="repository_abbr">
      <list>
        <item xml:id="abbr_BL">
          <abbr>BL</abbr>
          <expan>British Library</expan>
        </item>
        <div type="symbols">
          <list>
            <item xml:id="abbr_A">
              <abbr>A</abbr>
              <expan>Antiquarian Compilation</expan>
            </item>
            <item xml:id="symbol_del">
              <term>[ ]</term>
              <gloss>cancellation in the original</gloss>
            </item>
          </list>
        </div>
      </list>
    </div>
  </body>
</text>

The xml:ids for repository abbreviations and for STC, Wing, etc. are used in the <idno type="repository[or publication]" corresp="#">…</idno> in records headings.

This has not been implemented with XSLT. The page on the eREED website is not currently generated from the Symbols.xml file in Subversion.

7.3 <teiHeader>

The <fileDesc> was the <teiHeader> section primarily used. It contains the following subsections:

1. <titleStmt>: The title of the work, the funder, the people involved in creating the
work and the nature of their responsibility.
2. <editionStmt>: The edition or version of the work.
3. <publicationStmt>: The person or organization authorizing the publication of the work (REED), and the term under which it is made available (a Creative Commons licence).
4. <seriesStmt>: The series (REED).
5. <sourceDesc>: the source from which the work is derived. The following statement was used: "Born digital. Information on original sources for transcriptions can be accessed via the transcriptions."

A rudimentary <encodingDesc> and <revisionDesc> were also included as placeholders.

It will be important in the future to develop the <encodingDesc>. The Editorial Principles section in the REED collections might be incorporated here, as might the information contained in these guidelines. If REED intends to disseminate the XML files of the collections, then it will be important to have the <tagDecl> completed so that users can have information about how the TEI tags are being used for the REED collections. As well, instructions for how the <revisionDesc> is to be used has to be developed. Alternately or additionally, a ODD (One Document Does it All) file of the TEI Customization could be created.

8. Proposed workflow
(post-inclusion review)

1. Creation of Subversion file structure
2. Managing editor/paleographer groups Records by individual borough/parish, religious house, household or, for Records with broader scope, by county, diocese, province. The files are named [need naming system] and placed into Subversion.
   - We will need to develop type attributes to classify the records as above
   - We will need an additional set of type attributes to classify the organizations that produced the document if we want to
reproduce the order in which Records are put within the above categories (ie, civic, guild, church, miscellaneous)

- File format? .docx?
- Should submitting editors be encouraged to submit in a way that facilitates this practise?

3. Paleographic checking, including incorporation of foot and collation notes and addition of TEI markup replacing deprecated at-codes.

- Should corrections be directly entered into file, on the understanding that the corrected file can be easily compared with earlier versions (ie with changes visibly highlighted?). Should we dispense with corrections on paper?

4. Validation of at-codes (Perl script needs to be created)

5. Validation of pre-conversion TEI markup

A Perl script should be created to facilitate the first conversion of the at-codes marked up .doc files. The record transcription template is:

```xml
<text type="record" xml:id="ORANGE">
  <body>
    <div type="main_head_wrapper"></div>
    <div type="sub_head_wrapper"></div>
    <div type="transcription"></div>
    <div type="translation"/>
    <div type="endnote"></div>
  </body>
</text>
```

A Perl script could generate the xml:id for <text> automatically by parsing the content between the backslashes of @h and replacing the ! with underscores (_) and then parsing everything in between the backslashes of @w before the * (which will probably be deprecated) or an opening curly bracket (ie, {), inserting a underscore for any blank space. @h and @w content would populate the appropriate divs. Everything between the closing of @w and the <ab type="translation"> or <ab type="endnote">
would be included in the appropriate <div>s. <link>s should also be included in the template for the doc desc, EMLoT and Patrons links.

6. Finalization and TEI markup of Code List
7. Cutting & pasting of records into TEI record template

In order to eliminate error messages caused by having duplicate record xml:ids (the template used the names of fruit as placeholders, xml:ids were added during the cut and paste phase. The xml:ids were created by replication the information in the “unexploded” record header/subheader (enclosed in @h\...\) with the following modifications:

a) A “rec##” / “rec###” with the #’s being the number of the record in the Finsbury division, order sequentially from the first Finsbury file, followed by an underscore (_)
b) Exclamation marks between information (!) replaced by underscores (_)c) Slashes (/) in slash years replaced by “s”
d) The folio, etc information in the subheader as an unbroken unit with dashes in number ranges changed to underscores, preceded by an underscore; “single sheet” was abbreviated to “ss”

Example: rec02_FIN_1599s1600_PIFI_ff1v_2

8. At-codes to TEI conversion (Perl script)
9. Adding the headers from the Code List (XSLT)
10. Adding xml:ids to the footnotes (XSLT)
11. Required post-conversion markup
12. Validation
13. Additional TEI markup
14. Validation
15. Uploading records to staging server to check output

9. Alphabetical list of TEI tags used in The Fortune Theatre Records (including attributes (a) and values (v) used)
A. Note about attribute value naming conventions

In XML terminology, *values* are given to *attributes*, which modify *elements*. In this TEl markup – `<hi rend="center">…</hi>` – the `<hi>` is the element (text highlighted as distinct in some undefined way from the surrounding text), *rend* is the attribute (how the text is distinct in the original) and *center* is the value (the text is centered in the original). Except in some specific cases, values, unlike attributes and elements, are not predetermined. The standardize practice adopted for values is all lower-case, with values consisting of multiple words separated by a underscore, eg, "margin_left".

1. Records text (includes tags, etc, from the Code List that end up in the Records)

#: takes a numerical value
id: takes an id value

<ab> (anonymous block)

a: xml:lang
v: "en" [English]; "la" [Latin]

a: type
v: "body"; "body_p"; "body_pIndented"; "closer"; "endnote"; "head"; "initial_letter"; "list"; "marginalia"; "opener"; "play_text"; "prologue"; "subhead"; "table"; "title_page"

a: rend
v: "right"; "left"

<add>

a: place; rend [change to place]
v: "above"; "over"

a: scope
v: "partial"

<bibl> (file4, rec106)

<body>

<cell> (empty cell: <cell/>)

a: cols; rows
v: [#]

a: rend
v: "center"; "right"; ["bottom" might be added]

a: role
v: "braced_left"; "braced_right"

<choice>

<cit> (file4, rec106)

<corr>

a: resp
v: [id]

a: cert
v: "high"; "medium"

<damage>
a: agent
v: "bookbinding", "creasing", "creasing_tearing", "cropping", "erasure", "fading", "natural_hole", "smudging", "wearing"

a: degree
v: "low"

<date>

a: type
v: "circa"

a: when-iso; from-iso; to-iso; notBefore-iso; notAfter-iso
v: "yyyy-mm-dd"

a: calendar [not fully implemented: only file 1]
v: "old_style", "regnal"

<del>

a: type
v: "erasure", "overwritten"

<delSpan>

a: spanTo
v: [id]

<div>

a: type
v: "endnote", "transcription", "translation"

<ex>
<foreign> [not fully implemented]

a: xml:lang
v: "en" [English]; "gr" [Greek]; "la"[Latin]

<fw>

a: type
v: "catch"

a: place
v: "bottom" [change to "foot"]; "foot"

<gap/> = omission (editorial or because of damage/illegibility)

a: reason
v: "illegible", "omitted"

a: agent
v: "erasure"

a: unit
v: "chars", "line_remainder" [use a: scope instead?]; "lines"; "mm"

a: extent
v: [#]

a: rend
v: "line_fillers"*

*This should probably be used in <space/>, as line fillers designate (unused) space, rather than an omission.
<group>

<handShift/>

<head>

a: type
v: "main"; "sub"

<hi>

a: rend
v: "braced"; "center"; "display_script"; "initial_flourish"; "italic"; "small_caps";
"superscript"; "underline"; "vertical"

</idno>

a: type
v: "author_surname" "publication"; "repository"; "shelfmark"; "short_title";
"STC_number"; "Wing_number"

a: corresp
v: [id]

</item>

</lb/>

a: rend
v: "lineated"

</link>

a: type
v: "document_desc"; "emlot"; "patrons"

a: target
v: [id]

<list>

a: rend
v: "right"

<name> [EATS]

<note>

a: xml:id
v: [#]

a: type;
v: "foot"; "marginal"

a: place
v: "margin_left"

<orig>

<p> (rec04 [file 1], etc; rec161 [file 6])

a: rend
v: "indented"

<pb/>

a: type
v: "folio"; "membrane"; "page"; "signature"; "sheet"
<placeName> (record heading only)

<ref>

a: type
v: "foot_anchor"

a: target
v: [id]

<reg>

<row>

a: cols
v: [#]

<seg>

a: type
v: "publication"; "repository_shelfmark"; "signed"; "signed_mark"; "STC_ref"; "Wing_ref"

a: rend
v: "right"

<sic>

a: rend
v: "missing_abbr_mark"
<space/> = space left in the original text

a: rend
v: "covered"

a: unit
v: "mm"

a: extent
v: [#]

<subst>

<supplied> (file 3, rec61; file 10, rec202)

a: reason
v: "illegible"; "non_visible"; "tacit"

<surplus>

a: reason
v: "dittography"; "uncancelled"

<table>

a: rows; cols
v: [#]

<text>

a: type
v: "REED_record_group"; "record"
2. Code List: particular tags and/or attributes and values
(tags in bold italic also used in Records text)

#: takes a numerical value
id: takes an id value

<abbr>

<expan>

<item>

a: n
v: "parish"
<list>

a: n
v: "documents"; "records_div"

<note>

a: type
v: "antiquarian_source"

a: corresp
v: [id]

<text>

a: type
v: "code_list"

a: subtype
v: "REED_Midd"

3. The Documents: particular tags and/or attributes and values
(tags in bold italic also used in Records text)

#: takes a numerical value
id: takes an id value

<altIdentifier>

<bibl>

<div>
For the record:

APotype
digitation

which the page:
type

"div1"; "div2"; "div3"; "document_desc" "the_documents"

xml:id

<head>

<history>

<lb />

<link> [not used; for linking doc descriptions to transcriptions]

<msContents>

<msDesc>

<msIdentifier>

<origDate>

<origin>
<p></p>

<physDesc>

<repository>

<settlement>

<textLang>

<title>

a: type
v: "title_page"

4. TEI Header <teiHeader>

<address>

<addrLine>

<authority>

<availability>

<change>

a: n
v: [#]

a: when
v: [date]

a: who
v: [id]

<edition>

<editionStmt>

<encodingDesc> [not fully implemented]

<fileDesc>

<funder>

<licence>

a: target
v: [Creative Commons URL]

<orgName>

<persName>

a: xml:id
v: [id]

<projectDesc>

<publicationStmt>

<resp>

<respStmt>

<revisionDesc>
<seriesStmt>

<sourceDesc>

<titleStmt>

a: type
v: "main"