**EB Project Log 2017-18**

18-06-21

* **eb09**-s01 converted and validated.

18-06-20

* **eb09**-s02, eb09-s03, and eb09-s04 all converted well to entry files.
* **eb09**-s01 generates an error that I don’t recognize and won’t convert. Seems to happen on every file, too—if I remove the first, it happens on the second, etc. Also tried changing “s01” in the filename, but same thing happened.
  + Turned out to be a problem with the spreadsheet data—there’s a string where it expects numerals.

18-06-19

* **eb09**-s01 and eb-s04: fixed run-over notes and other issues mentioned on inventory file, but left language issues alone. Need to fix stubs for both, then okay to convert.
  + **eb09**-s02—fixed entry stub.

18-06-14

* **eb09**-s01 proofed. Found a fair number of footnotes in middle of page. Haven’t checked notes yet for run overs or other problems to be fixed before conversion to entry files.
* **eb09**-s04 proofed. Need to go back to notes for footnote run-overs before conversion.

18-06-07

* **eb03, eb07, eb09, eb11:**
  + all entry stubs fixed.
  + Extra space around <hi> tags corrected.
  + All entities fixed

18-06-06

* Finished all of **eb03** entry checks. Discovered some issues:
  + In marginal notes, they italicize both author name and book title. Do we follow this, or modernize?
  + Within entries, they include other entries beginning with the same name (variants of the main entry term), using c/sc. Should these be broken out? They are very short, and I have left them.
  + I quit numbering marginal notes in eb03-r02 and r03. Made it much easier to add them.
  + There are a lot of recognition problems throughout eb03.
  + Need to think carefully about how to convert long-s to short-s, without creating a ton of messes where it should be an f instead.

18-06-05

* **eb03-r01**: took a full 3-4 hours to do one long entry that needed 34 notes inserted!! It would be faster to go back into the AFR project and correct notes.
* **RUNOVER FOOTNOTES**: when note text runs over to next page, I’m inserting an empty <pb/>, with no attributes.
* **MARGINAL NOTES**: I’m adding @n for both anchored and unanchored types, but wonder if I should add them for either. It’s difficult to keep the numbers sequential in an article with both marginal and footnotes and is going to involve re-numbering everything. So for now, start new pattern with eb03-r02: *marginal notes have no @n.*
  + Also, I’m not including marginal notes that only reference a plate, since we are not including plates.
  + When the print is unclear and I have to make a guess, I’m using <unclear>. I asked Rachael to insert ## double-hash mark before and after anything that should be marked <unclear>.

18-06-04

NEXT—finish eb03-r01 entry check

* **eb03**: entry terms use a combination of all caps and caps/smallcaps. They generally use all caps for the first word only of a 2-word title, and for the noun only in a two-word title beginning with an adjective, which is in c/sc. Thus on eb03-15-r01-0776, we have:
  + Rowley RAGG
  + RAGMAN’s Roll
  + Cities of REFUGE
  + I need a way to code these originals. Perhaps <rend> or <corr>?
* Python is inconsistent with the c/sc, but usually doesn’t catch it. So there’s no automated solution here except later manual corrections. For now, I’ll correct it as all caps in order to get the basic entry term, but I’m leaving it alone otherwise.
* Stub entries—basic solution for entries split between two sections is to add the entry term to the 2nd half manually, with a marker (I’m using “CONTD”) before running Python. Then manually combine it with the main entry in the previous section and delete from the section section.
  + Note: these runovers have to go through Python to add in pagebreaks and modify footnotes, but footnote numbers will be out of synch with the first half.
* **Marginal notes:** format as <note anchored="true" place="margin" type="authorial" n="1">, keeping in same numerical sequence with footnotes, based on location in text flow, so Python can position them correctly. Operators are formatting with [m] at beginning, so search and replace should allow for updating @place.
  + Some marginal notes have no anchor but are clearly citations for a quotation at that point. We should include these, with @anchored=”false”.

18-06-03

* Finished entry checking on eb07, eb09, and eb11 for “r”. Also fixed last entry of each section (entry stub) in eb07 by adding the entry title to the first page in the TEI input files for the next section, so Python will process it. Worked well, but needs to be done for the others.
* Python is back to adding an extra space around tags; need to have Gary fix it.
* Python is also adding doubled entities, like &amp;amp;.
* Major problem in eb07 was entry heads that ran across two columns. They were never picked up.

18-05-23

* Python conversion for eb09-r (all).
* Discovered in eb09 that AFR sometimes output “&lt;kc.” instead of “&amp;c.” It carries over into the TEI and after Python, it shows up as “<kc.>”, generating an error for an illegal element, and generating a closing tag soon after.
* Search/replace:
  + &amp;amp;c. &amp;c.
  + thc the (none) (none)
  + tlle the

18-05-23

* I replaced long-s throughout eb03, and now see the problem in doing that—it produces words with an “s” where there should be an “f”, because of recognition problems. And it doesn’t fix words with an “f” that should be an “s,” like “prefcribe. So I overwrote the files with the unchanged ones, retaining the long-s.
* Have to review details about Underwood’s normalizer script and test it of these files.
* There’s a lot of recognition problems in these early files, and I need a cleanup strategy.
* eb03 search/replace:
  + &amp;amp;c. &amp;c.
  + thc the
* **&amp;amp;** This doesn’t exist in the TEI-page files, so something in Python is causing it.

18-05-22

* Reran transforms on eb03, 0b07, and eb09 with modified XSLT.
* Created entry files for all of eb03 and eb07.
* Python threw an error when hitting pages that were double-numbered in print edition, duplicating other numbers. EB used a page number with an asterisk: 437\*. But the script won’t accept a non-numeric character. I got it to run by repeating the same page number, and making a note in the inventory file. I then manually changed the xml:id for the pages; xml won’t accept an \*, either, so I added an “a” to the end, as 437a.
* Script is inserting xml declaration and TEI header at every page break. Gary will fix. Meanwhile, I am removing the excess code and everything validates.

18-05-21

* Further modified eb-html2tei to eliminate <head> codes as well as @bold. Tested both and script is working well. Need to redo transforms on all files for eb03, 07, and 09.
* Did a lot of search/replace in eb11 entry files, to clean them up. There’s a lot of errors. The most consistent were:
  + U li | ll | il
  + E li
  + aU all
  + thc the
  + H li | ll
  + AI M
  + Al M
  + bom born
  + 1/2 ½ (also applies to entries for ¼, ¾ )
  + &amp;amp;c. &amp;c.

18-05-19

* Missed entry terms: quite a few in eb11-r03. Tracked it down to including of <hi rend=”bold”> entity. Python looks for a string of capital letters beginning at a <p> code, so doesn’t recognize those with formatting. Easiest solution is going to be to quit preserving bold—AFR is very inconsistent with it, anyway, and it’s of no use for the entry terms. So I have to modify my xslt script.
* Checked eb11-r04 and r05. r04 has 14 bolded entry terms that will need to be fixed. r05 doesn’t have any.
* Finished entry check on eb11-r03, r04 and r05.

18-05-18

* **eb11-23-r03-408-01.xml “Robes”.** Missing text was in the TEI file, so I corrected it. But it points to a problem with notes that run over from one page onto the next. This one was not indicated in the inventory.
* Frequent truncations in entry terms by script, due to AFR substituting something for a comma, etc. Finished checking & correcting all entry terms for eb11-r02.

18-05-17

* **eb11-23-r03-408-01.xml “Robes”**. Big problem in here. Many footnotes. We lose one on page 414 (n.4 in print text), and the rest are out of sequence and in the wrong position. Text is missing for n.5 on the same page, and it runs over to p.415. The runover text has nowhere to go. Probably have to go back to the html to see if we captured text for n.5 (it’s a long paragraph), then fix the TEI and rerun python. Otherwise, I can fix it all by hand.
* **Python:** Found the errors in **eb11-r04**, which gave the Tkinter callback. The last line of the error report references l. 396 in the main Script module,   
  “fulltxtwithnote = fulltxtwithnote + block + fullstrfnl[index]”   
  and gives an index out of range error. It indicates two @@@ codes in one paragraph. There were three of them all in all in this section, after the initial attempt to fix them, so I think were instances where there were more than one of these on a page. After fixing the TEI, script ran fine.
* **eb11-23-r05-0869-02.xml** “Russia” file validated after adding a <div> code after the <pb> where the error appeared. Same technique worked for another file with same problem.
* ~~Python~~~~missed two entries in eb11-r04, and I created new files for them, but the entry-term lists still need to be proofed. As an undergrad to do that.~~

18-05-16

* Added information to manual on correcting corrupted images.
* Updated all inventory files to remove extra space after p\_vol.

18-05-13

* Created image selections for all “S” sections in all four editions. Also added new “plates” subfolder and removed all plates and blank pages from the selection to be scanned.
* Created inventory files for all of “S”

18-05-11

* Restored inventory files to single file versions.

18-05-10

* ~~TO DO: remove inventory files from collected folder and recreate single file versions.~~
* Continued correcting problems in TEI files that create Python errors, using Luling’s list.
  + Corrected <p> for footnote contains two or more notes, using regex.
    - FIND:
    - ([\.|”|"])[\s|\h]+@@@
    - REPLACE:
    - \1</p><p>@@@
* Tried running script on **eb11-r04**. First 98 files run fine, but then I’m getting an error message I don’t understand. “Exception in Tkinter callback / UnboundLocalError: local variable ‘seondfirstindex’ referenced before assignment.
  + NEXT: Try running Luling’s script in the earlier version, before we changed it.

18-05-09

* Set up Python on office computer with 3 libraries: BeautifulSoup, Pandas, XLRD.
* Modified Python script to add empty TEI header, so files validate.
  + Modification includes changing name of the main module in where it shows up in \_init\_\_5.py, EBrit\_main\_5.py, and Interface\_5.py. Main module was Script\_Modules\_5v02.py. I changed it to Script\_Modules\_5v03.py.
* Output **eb11-r05** entry files and validated.
  + One file, **eb11-23-r05-0869-02.xml** “Russia”, does not validate, but there is nothing wrong with it. It’s almost 50-pp of text, so huge, and I think Oxygen actually can’t keep track of all the codes. \*See note 18-05-17.
* Planned orientation meeting next week for 3 undergraduates, with Bethany.

18-05-01

* Add to manual:
  + ~~How to replace corrupted images—right-click/properties/reload original~~
  + ~~Adjust table rules~~
  + ~~Not correcting typos, old hyphenation, etc.~~
  + ~~Update instructions for inventory files—one per edition letter.~~
  + ~~Use language group eb11 for that edition~~.

18-04-29

* 11th ed For cross references at the end of articles, identify them in TEI by searching **See \w\***. Or **(see \w\***. Term is in caps/small caps.)

18-04-28

* Add Arial Unicode MS to recognition fonts. It has 38,917 characters, compared to 2,790 for Times NR. And we need it for formulae.
  + ~~add to manual to include Arial Unicode MS~~
  + add to have US international keyboard available and how to switch to it.
  + ~~add guidelines for formulae (single-line vs. multi-line), including notes in inventory for multi-line formulae.~~
* Tried marking an area as a picture and outputting. HTML formatted just ignores the picture box completely. Another idea is to insert some keyword which could be replaced in xslt to an image tag.

18-04-27

* Need to know what to do about multi-line formulae, since they are not going to work. One-line math formulas work well. For now, can just indicate pages in the inventory file.
* Need to know if it’s possible to indicate image placement easily or if it’s not necessary.
* ~~Need to add a page to~~ **~~guidelines~~** ~~on running verification on each pag~~e.

18-04-26

* Discovered problem with Python script, deleting <div> tags that my xslt was adding for <head> tags. Also led to loss of page text prior to the <head> tag. Luling made changes and reran on all of **eb11-r05**. Corrected output is in digital-editions folder and needs to be validated.
* Contacted ABBYY about output placement of footnotes being unpredictable. Their solution (identifying the note box as footnote in the context menu) works and I’ve gone ahead and changed the procedure in reference manual.

18-04-10

* re-output eb11-r05-0007 html and reran TEI for all **eb11-r05**.
* TO DO on manual:
  + ~~add para on image resolution to AFR section: if >600 dpi, fix all pages before drawing boxes.~~
  + training the user files: can zero in on something by drawing a small text box around it and leaving the rest of the page blank

18-04-07

* Began redoing all of **eb11-r01.**

18-04-06

* Need to re-output eb11-r05-0007. Entry ROTTENBURG is at end of page, but missing from html. (DONE 18-04-10)
* Need to create new xslt template for TEI entry files, after Python is done. Remove <p> elements before and after <pb>.
  + Currently we have:   
     “ridge </p> <pb xml:id="eb11-23-r05-0002"/> <p> deep down”  
    We want to remove those paragraph elements and let it occur within the <p>, as:  
     “ridge <pb xml:id="eb11-23-r05-0002"/> deep down”

18-04-05

* I re-ran **eb07-r03** as html.
* Reviewed python Entry-Stats file for Luling’s test output of **eb11-r05**. Only problems are OCR errors, and we need one change to the script, to allow hyphen and apostrophe as legal characters in an entry term. He’s working on it.
  + Also need to identify reason it failed to identify one entry, ROTTENBURG.

18-04-01

* All “R” sections are proofed. **eb11-r01** needs to be completely redone. **eb07-r03** needs to be re-output. Everything else is good to go. Notes made in inventory and production sheets.

18-03-21

* Notes on Python output
  + Need to proof the Entry\_Stats file, to see how well it did with splitting entries.
  + It begins with the first entry on the first page. The beginning text matter, from a previous page is ignored. Similarly, the end of the last entry is going to be missing.
  + One word entry titles are properly captured within <label>.
  + Current output substitutes “0001” for the actual page number; I’d guess Luling did not enter in the data on the start screen?
  + <pb> is properly formatted and increments the starting page # by one.
  + Currently, it stands on its own, as:   
     “ridge </p> <pb xml:id="eb11-23-r05-0002"/> <p> deep down”  
    We want to remove those paragraph elements and let it occur within the <p>, as:  
     “ridge <pb xml:id="eb11-23-r05-0002"/> deep down”
  + It’s adding spaces around both <hi rend=”italic> and </hi>, so there is always white space before and after each tag, regardless of spacing in the input.
  + Look at 0002 of this html output: it maintains line breaks with text that wraps around an illustration, when we don’t want to preserve them. Also 0004.
  + The <p> inserted at column breaks—think about ways to remove it.
  + space around quotes is not being removed: “ bush ” .
  + It’s correct to use the angled rather than straight quotes, but better to use character entities: &ldquo; &rdquo; &lsquo; &rsquo;. This could be built done in an xslt template. JF also recommends using @rend with <quote> or < q> and the prefix and postfix keywords, rather than including the quote marks in the text. Thus:  
     <quote rend="pre(&ldquo;)post(&rdquo;)">How delightful,</quote> she thought to herself.
    - NB: if we do this, all four entities need to be declared.

18-03-17

* Created inventories for:
  + **eb03-r01, r02, r03**
  + **eb07-r01, r02, r03**
  + **eb09-r03**
* Mapped all “S” sections in “production” spreadsheet.

18-03-16

* **eb11-01** was done with separate odd/even using b&w instead of color, and accuracy results are below our more recent output. **It should be completely redone**. I corrected a few things, made an inventory file, and output the existing odds as html.
* **eb11-r02** and **eb11-r03:** both checked, inventory created, footnotes corrected, transformed into TEI and validated.

18-03-13

* Ran test on alternate originals for **eb07**. They result in better OCR than the current collection. I’ve asked Bethany to take charge of downloading them (when she’s free in a few weeks).
  + Learned that AFR can’t read the pdf until it’s converted to image format. It finds garbage in the pdf. I tried the different settings but nothing worked.
* Added missing page to **eb11-r04.**
* Discovered that **author initials sometimes appear in footnotes**, as author of the note**.** See eb11-r05-0020.
* Fixed pages in **eb11-r05**, where there were notes in both cols that were treated as one box instead of two.

18-03-08

* created inventory for **eb09-r02.** Haven’t seen same problems as in **eb09-r01**, though accuracy can be improved with same procedure. I think best thing is to move on, though, instead of redoing it.
* problems with XSLT template for dealing with @’s and italics. Not working right, so needs to be corrected, and then rerun transforms for eb11 and eb09.

18-03-07

* TO DO:
  + Finish XSLT and transform all HTML files
  + Download and import Project Gutenberg files
* **eb09-r01** had some untagged tables that I corrected and re-recognized the pages.
  + Also tested the “verify” option in AFR, which works excellently and would give 100% accuracy on each page, but it’s time consuming.
  + Changed large diagrams (full and half-page) into images, to see if it will incorporate them in the output in a way that would be useful.
  + AFR is not correcting the page images automatically, and that’s reducing accuracy slightly. Need to revisit procedure to see how it can be automated.
  + Trained user file and added words to dictionary. Accuracy is much improved, and the colored background that’s showing up in the output window is gone. I think OCR settings were not right when this was done. I finished up to print p. 260 (71), but the rest needs to be redone. Accuracy is so good, the corrections are easy. But my procedure included preprocessing the image.
* Inspected **eb11-05**. Procedure is to make the well-formed, then search for all variations on @style values, to make sure I’m capturing all recorded font properties. Then to look for any other elements or formatting issues. Finally, check transforms for notes with italics—stylesheet differentiates 2x from 3x @’s, but I haven’t been able to verify that it works.
* **eb11-r01** was output 2/28/18, with the old even/odd split, but odds were output as .rtf, so should be re-output.
* **eb11-r02** hasn’t been corrected yet.
* **eb11-r03** looks like individual pages (w/footnotes?) were re-output; the rest from 11/1/17 was okay.
* **eb11-r04** is missing print page 628 in the html output. Ask Bethany to check and re-output. Inventory good. DONE
* **eb11-r05** checked by me, transformed to TEI—good to go.
* **eb09-r01** output 2/28/18. Needs updated inventory. Ready for inspection.
* **eb09-r02** output 2/28/18. Needs updated inventory. Ready for inspection.
* **eb09-r03** mixed output from 12/12/17 and 1/13/18. Needs inventory. Check with BF.
* **eb09-r04** checked by me, transformed to TEI—good to go. Inventory good.

18-03-06

* Inspected all html and TEI files in **eb09-04** carefully. AFR will combine font attributes—I found a combination small-caps and italic—so I’ve begun adding combinations to style sheet. Decided to use a nesting format, rather than create custom values for combinations in @rend. Also inventoried all @style values in the html. These were limited to italic, bold, small caps. Created a controlled vocabulary for @rend and added to online manual. Updated XSLT captured everything correctly in eb09-04.
* Inspected AFR CSS files. Each page has its own set of font definitions, and the class names have no consistency among pages. Almost all classes are for 7, 8, or 9 point serif type (in eb09), with occasional larger sans serif fonts. I don’t see any way we can make use of this data without converting all class names to a consistent pattern, but there’s no semantic significance to these size changes. They don’t indicate the entry titles, for example.

18-02-28

* Created new format for inventory files and filled in image information for **eb11-05** and **eb09-04** (the two most recent outputs by Bethany). Need to do more inventories.

18-02-06

* Reviewed Python script with Luling, section by section. Removed some html-specific sections. So parts are so complex we couldn’t figure them out. I’ll output 20 pages of TEI for him to try it on, with his new note function. Goal for xslt is
  + create regex to identify superscripted number following @ code and remove it.
  + create regex to locate @ codes nested within italics and remove them.

18-02-04

* Refined XSLT. Tables work well, italics, small caps, bold, underline is preserved, but css font style classes are eliminated. Heads (h1, h2, h3 …) are preserved as <head>. Tables retain formatting well.
* Next: spend more time considering problem of footnote symbols contained with italics, etc., and whether or not to retain or remove it. Footnote text is a separate question, since AFR sometimes formats the note and text the same, which I’d want to preserve if, for example, the note starts with a book title.

18-02-01

* eb09-04 output does not include tables and has to be redone.
* eb09-r03 is missing the hidden project files and needs to be redone.
* Inventoried all AFR files and discovered 5 that need to have tables included and re-output. eb11-r04 and r05; eb07-r01 and r03
* eb03-r03 boxes include the leading word for the next page and needs to be redone.
* Updated manual with more information on treatment of image captions, blank pages, and tabular matter.
* Created instructions for Bethany on correcting existing AFR project files.

18-01-31

* Jim finished the renaming for AFR file folders.
* The problems below were also corrected.
* Bethany reviewing project manual. I’ve given her file permissions.

18-01-24

* eb11-r05: Footnote text is still missing the note number and first character in the html output, throughout.
* eb11-r04: Footnote text looks good, but did find same problem in #0088. In #0012 there’s an instance of 4-@’s. Otherwise it looks okay.
* eb11-r03: Footnote text looks good. Problems:
  + note text but no reference: #0062, 0063, 0073, 0074, 0077.
  + different number of references and notes: 0085, 0125, 0126
  + triple @ instead of double @: 0093, 0133
  + Ask Jim to proof the notes—use Oxygen to find them in the batch.
* eb11-r02: all good.
* No file renaming yet on eb11.

18-01-19

* Worked with Luling on Python script. New convention established for note naming and page breaks. It’s all in a new section of the project manual. Talked about how to handle footnotes that are more than a single paragraph. Need to let Jim know to lookout for those. No solution at the moment.
* Moved new folder structure onto the server and backup up everything to RedHD.

18-01-13

* html-to-tei is not ready. It is getting rid of all italics. Tested it on “Rook” in eb09-20-r03-0163. Compare with image file …0876.jp2. It’s also getting rid of @@@ codes! And the <div> tag, with all the data in it.
* About 10 pages of eb09-r03 have not text in the html. Find out from Jim what that’s all about.
* We need inventory spreadsheets for these new html files.
* Currently, I’ve left off the volume # from filenames, for some reason, and it makes it difficult to locate the original image, so I need to change that in the new procedures for the production folder.
* add section on filenames at the beginning, in the Naming Conventions file.

17-12-17

* Tested Stanford NER on TEI files and it works fine, adding tags directly to the XML.

17-12-04

* Jim had a problem with @ coding—AFR wasn’t updated after the computer was upgraded to Win10, and every time he entered an @ code, it overwrote the text. So he’s redoing them all for 11 r04 and r05.

17-11-22

* reorganized the entire project structure to allow for the later stage of creating digital editions. Moved all relevant files on the home machine into “eb-project”.
  + Missing the corrected html output from JK for eb11-r03. Is it on the DSC server? [located on CLA backup]. There’s also quite a mess of file disorder in there.
* Preserve older “eb\_project/eb-pilot” folders for backup, until I’m sure I can get rid of them. I’ve turned off syncing for the older folders on Box, and turned it on for the new structure only.

17-11-18

* Luling has solved the moveNotes.py function and is integrating it into the old script
* James appears to be making no progress. He started on eb11-r04 several weeks ago and it still isn’t done. At this point, we are so far behind in production I have to recalculate for grant proposals.
  + Talk to him about folder names; he’s using “2 – afr” instead of “2-afr” and it’s going to affect syncback’s handling of those hidden files. eb11-r04-afr is missing those hidden files. [okay on server 11/27/17]
* Reorganized folder scheme. Most of the Production folders remain the same, slightly stream-lined. A new Editions folder will hold the xml master files and xslt scripts, and that replaced the older “entries” folder. It’s a way simplify the digital edition structure.
  + TO DO: rework the eb-documentation dita files to reflect the new structure.

17-10-30

* Jim is redoing the @@@ coding for eb11-r03.

17-10-28

* TO DO: talk with Jim about footnotes. He’s coding the ref and text both with @@@. Also needs to create an inventory file.
* Refined html-to-tei-pl1.xslt more. Integrated <table> templates. I also discovered problems with the <sup> and <italic> templates, and explained them in comments in the file itself. It is all working in 90% of the cases, but I’m hoping I can refine it to correct these problems.
* Learned that matches() looks for any instance of the match term anywhere inside the context, and it’s not quantity sensitive, so regex expression like \d{1,2} doesn’t do anything, beyond finding whether or not there’s a single digit match anywhere in there and then returning a TRUE. May need to switch to replace() or tokenize() or xsl:analyze-string.

17-10-27

* In html-to-tei-pl1.xslt, I was able to strip out all italic and font rendering for @@ codes, strip out note numbers, but leave all <sup>, <italic>, and <underline> in place as TEI formatting.
  + The trick was using <xsl:choose> <xsl:when> with regex inside matches() functions, and <xsl:otherwise> for everything else.

17-10-25

* Final TEI code for notes should be “Body text runs along <note>insert note text</note>. Body text continues”
* Python will locate the note text and move it into place, using the @ codes. But I can strip out all of the <hi rend=”sup”> tags. Leave the note numbers since I think Python uses them to sequence the notes on the page. It does not need to output them though in the entry files. I may be able to get rid of them, but that depends on what the script looks like.
* Spent some time working on xsl to identify the <sup>codes in the html that are notes and get rid of them while leaving the <sups> elsewhere.

17-10-18

* Improvements to basic html:tei stylesheet. Fixed template for converting italics. Also have one for sup tags. Am stripping out all other font tags at the moment.
  + Question is whether I need the <sup> tags at all. As long as operators are identifying notes with @ codes, there should be a way to use that to specify notes and their formatting. But I’m not sure how to identify them as xpath nodes yet.

17-10-14

* Succeeded in creating a primitive template to convert html to tei. Have text pulled over and singled out <p>. Began working on <span> to convert italics to tei and was able to isolate only the ones specifying italics.

17-10-11

* All images files are in place to finish “R” in all four editions.
* TO DO: Decide how to handle author attribution, using TEI:byline element.
* Reorganize folders, following Laura Mandell’s example, on the assumption we will not have a single AFR, HTML, TEI, and TXT folder for each section, rather than multiple.

17-09-27

* Folder is now setup for group work with OCR projects. Added new instructions to the project manual.
* **EB03—if I take a list of common English words in plain text, I can replace “s” with long-s and import into the eb03 dictionary file to improve recognition.** See <http://help.abbyy.com/en-us/finereader/14/addwordtodictionary> .

17-09-18

* NEXT—Continue working on AFR output problems with eb09-02.
* XSLT—for changing string codes, look into the “translate” function.
* There’s a lot of problems in eb09-02, and I’m starting to think it was output with unusual settings. May need to redo all of it. But the biggest problem is that AFR is not passing the note numbers to html. Instead, it generates empty <sup> tags.
  + I tried everything. AFR includes the notes in Word output. It recognizes it on the screen. I tried to change its font styling (no change). And I re-recognized the page and added back the @ codes (no change).
  + Finally got it to work by re-recognizing, saving it without any modification as docx, then as html. It saved the note numbers. Then I added the @ codes, and it still saved the note numbers.
  + In my testing, I only saw this pattern in the footnote text section, not the body section, where note numbers were preserved.
* Also had strange reordering of lines in the footnote text on 170918-eb09-r02-0001.htm. This was also losing the note numbers, and I only focused on that, but I need to check the line order problem. I notice that it looks fine in afr on flexible (2-col) layout, but you can see the text out of order when you switch to formatted (1-col) layout.
* Problem code in eb09-r02, first page:
* @@</span><span  
   class="font2" style="font-variant:small-caps;"><sup>4</sup></span>
  + In some cases, we’re getting special font formatting just for the @ codes:
    - <span class=”font1” style=”font-style:italic;”@@</span>
  + Can strip it out with this regex:
    - Find: <span class="font\d" style="font-style:italic;">(@{2,3})</span>
    - Repl: \1
* Changed file @ for all AFR project folders to remove @hidden, on both my HDD and the dsc-server. SyncBack is doing something I don’t understand—deleting a file on my HDD after copying it to the dsc-server, while it’s set to synchronize. I changed some settings specifically for the afr-folders and created a separate routine for it. Run it manually, until I’m sure of what it’s doing. And keep an eye on the file @. I set it to copy them.
* EB09
  + checked Hebrew in eb09-r01. Rescanned one page using Hebrew and checked the html output. It is no longer using @dir=”rtl” in AFR14, so the output is actually fine. Instead, it uses CSS and identifies it as <span class="font1">, which is just a font+size specification. html2tei2 ignores the css anyway, and translates it as <hi rend="font1">.
  + existing eb09-r01 is actually AFR12 output, not AFR14. I have re-output it as 170918-eb09-r01.
  + checked eb09-r02. Html is AFR14 and there is not @dir or “rtl” or ‘ltr”, so I think this output is okay.
* CONCLUSION: okay to use Hebrew language setting now. Redo the DITA document to correct this.

17-09-13

* PYTHON SCRIPT: Move Notes function is still tied to html instead of xml coding. Luling is updating it. That *may* be why it’s adding the spurious code to the pagebreaks.
  + Luling modified the script to remove all font tags except those for notes. I asked him to retain the italics.
* XSLT: update script to normalize all @@ codes to be nested inside the <hi rend=”sup”> tags, and not outside it. And use XSLT to remove all font tags except the ones I need. Then remove that section of the python routine. [Better to keep that level of coding within xslt, I think.]
* EB11
  + The new html output still won’t run for eb11-r02 in Python.
  + Still need to locate coding problem in eb11-r01 that’s preventing it from running.
* EB09
  + Needs to be redone without Hebrew. Check both r01 and r02.

17-09-11

* ON EB11: Reran afr output for eb11-r02 as formatted rather than flexible, to get rid of the table coding for all pages. Also did the html2tei2 conversion and moved everything onto the servers.
* It runs fine, output good, but it’s inserting code for several of the footnotes:
  + <a name="footnote1"></a><a href="#bookmark0">1</a>
* oXygen catches it because it reused “footnote1” and TEI converts that to an xml:id, so you have more than one of them. It only happens in one of 250 files, but happens twice there, in 170911-eb11-r02-0098.htm.
* I manually deleted it for now, so Luling could run the script.
* We’re encountering some problems in eb11-r02 with large segments of text getting formatted as <sup>, so that needs correction.
* ON EB09: It was output from AFR with Hebrew and has to be redone. Not sure if both r01 and r02 need it.
* One several pages, the script is adding bad code to the <pb>:
  + </p> <p> <hi rend="sup">  
    <pb n="eb09-1-R-0014"/>  
    <p>
  + All that should be there is the </p> <pb n="eb09-1-R-0014"/>  
    <p>
  + It happens 84 times in maybe 30 files. First instance is eb09-1-R-0013\_02.xml.

17-09-08

* eb11-r02 html was output in columns, instead of with the flexible format. If the updated code works for all the other editions now, the only correction needed is to redo the output--but make sure we still have the afr project file intact. [update: it didn’t run]
* 170620-eb11-r01-0119. Pythron script throws an error. I looked and there’s a problem with the note encoding. @@ always needs to be nested within <sup>, not preceding it. Then find a way in xslt to correct it. But look also at 118 and 120, which also have @@ inconsistencies.

17-09-06

* Luling ran a test on some of the AFR14 files with the python script. If he strips out all of the extra <hi> tags, the script runs. His thought is that the script looks for the tags related to notes and inserts the <hi> tags, and finding them already there may be throwing it for a loop.
  + NEXT—modify html2tei2.xsl to ignore all font tags except the notes and rerun on the afr14 files. Then retry the python script.

17-08-22

* Syncback deleted tons of files off the server for some reason, for all four editions—mostly the archive and output files. Spent much of the day trying to repair the damage and restore the files.
  + Traced it to 8-19-17 6:00 pm run of “sync eb desktop dsc-server.” Same profile ran perfectly the day before, so what changed?
  + It’s a smart sync profile
* Gary’s script is unable to parse html files for eb07, but okay for eb09 and eb11. The difference I noticed is that 7 uses CSS while 9 and 11 do not. I think this may have been a change introduced in AFR14, and I see no way to control it from within ABBYY. A solution may be to modify html2tei2 to stripout the <span> tags. The only thing we need to retain is the <sup> for notes, I think. Take a look at it all next.

17-08-17

* AFR has instructions for group work with OCR projects. Set it up on the server so all student workers can access the right project files, dictionaries, etc.
* <sup> tag problem: trying to create a routine is xslt requires more skill than I have currently—perhaps it can be done as a string? In the meantime, ADD the find/replace routine to the current instructions for getting files to be well-formed.
* Problems with AFR project files: The root director for OCR projects has to include 4 files (three of them hidden): \_FRBatch.pac, batch.options.xml, desktop.ini, and packet.ico. (The ini and ico files just change the folder icon.) Some of the folders are missing all of these files except batch.options.xml. Owlbox does not sync FRBatch.pac or packet.ico. I believe this is where they are getting eliminated, somehow, so I’ve stopped synching the 2-afr folders to box. Syncback by default ignores desktop.ini files, so I’ve changed that option to include it now.
* Box sync ignores hidden files. <https://community.box.com/t5/How-to-Guides-for-Sync/File-Types-Ignored-or-Blocked-by-Box-Sync/ta-p/117>  
   That would explain the problem. But in order to use box, we would have to routinely change those files to visible when the project folder is created. I don’t think that’s going to work and it’s too dangerous.

17-08-15

* Discovered that both the eb\_projects and eb-projects directories are on the server, and Nahn has been saving some files into the older one. He’s reconciling the two, and we’ll delete eb\_projects.
* Gary discovered that the script gets tripped up when the @@ code for notes precedes the <sup> tag, instead of being contained within it. That’s something that happens at the operator stage, but it’s easier to fix it in the html2tei2.xsl script, rather than slow down the operator, so I’ll add a line to it and that will take care of it going forward. In the meantime, Gary and Nahn will take care of it in the existing files.
* Reviewed grant proposal responses with Marcus.

17-08-11

* Gary has fixed the python script to convert tei files to entries. He’s proofreading some of the output for eb11-r01 and discovering some problems in the operator coding of footnotes, which he’ll correct in the original AFR project folders and re-output.
* Need to think about moving forward, whether to focus on simultaneously doing all 4 editions, or to focus on one at a time.
* Nahn has finished r02 for all 4 editions and converted to TEI, so we now have a complete set of 500 TEI-page files for all 4 editions.

17-07-25

* fixed html output for eb03-r01-170720 and converted to tei successfully.
* output txt for eb03-r01-170720 without tables, so I have complete set for all 4 editions.
* **OCR-Normalizer test**—used file compare, and it’s making too many errors. Changed “forts” to “sorts” when it should have left it alone, and introduced some other new problems. I was comparing eb03-all.txt with eb03-all.clean.txt. The file is too big and needs to be segmented, but my first reaction is to stop using OCR-Normalizer and look for something better
  + Compared # of unique words before and after OCR-Normalizer. It makes a big difference—cleaned file has 85% as many unique words. (w/out table data).
  + Spend an hour looking online for other OCR correction software and can’t find anything suitable. I think the best option is going to be modifying Underwood’s routine through a process of trial and error.
  + I set up a folder with everything Gary needs to refine the OCRnormalizer while I’m gone.

17-07-20

* Modified *tei-to-txt2* to eliminate tables from output by adding an empty template:
  + <xsl:template match="table"/>
* Discovered that tei-to-text.xsl replaces “ſ” with “s”.
* tei-to-txt is treating <hi rend=“ital”> and bold as box elements and creating new lines. It treats <hi rend=“sup”> and sub as inline elements, meaning the footnote number is getting appended to the word, creating a new type. Not sure how to correct this. Spent hours on the box/inline problem to no avail. Perhaps with sub/sup, just eliminate it.
* Redid output for eb07, 09, 11 without tables. eb03 had recognitions problems, so I spent some time training and modifying the alphabet before rerunning.
* NEXT: fix html output for eb03-r01-170720—I didn’t have time to save it.

17-07-18

* Began running analysis on 4-editions r01 corpus. TTR done (listed in notebook p. 60).
  + Discovered that there are 847 occurrences of “os” in eb03; probably a long-s mistake for “of.” And there’s 1154 instances of “â”. I’d guess many are in error.
  + Manually corrected “os” to “of” after checking—they are consistent.
* Created eb-r01-word-freqs.xlsx to record data from R analysis. Highlighted gender terms in eb03 and eb11, just to see what the proportions are.
* In R, calculated TTR, output word frequency tables as .csv files, and plotted word top-10 word freqs (this last was pointless, other than to see that they do indeed follow Zapf’s law).
* NEXT: figure out how to eliminate tables and redo all calculations.

17-07-13

* Created corpus in R folder of all txt files for all 4 volumes (using “normalized” text files for eb03). Used command line to concatenate individual files into one for testing purposes (copy \*.txt eb03-all.txt)
  + except for eb03.clean, need to specify in R encoding = "UTF-8".
  + reprocessed content for all 4 files
* output txt from straight pages tei as corpus to begin trialing some textual analysis. Did r01 for all 4 editions, using tei-to-txt.
* Nahn finish eb07-r02. I’ve imported and transformed to TEI.
* updated project org chart to include txt folder and added file-naming guidelines. I’ve slightly modified the entry guidelines. Previously, we had *eb11-23-r-0773-01*. Instead, use: *eb11-r01-23-0773-01*. That’s *ebnn-rnn-vv-pppp-ee*, where *ebnn* is the edition, *rnn* the batch, *vv* the volume, *pppp* the page, and *ee* the relative entry position.
* tei-to-txt retains all table data, removed from table format. Do I want it for running text analysis? or not? My first thought is to exclude it---the numbers are not part of analysis, and the words are all out of context as cell labels (“trespassing. not trespassing. other persons. killed. injured. highway crossings. total of servants.”)

17-07-11

* NEXT: Create corpus of txt files for 1st 4 editions, r01, and get started with basic R routines. eb03 is done.
* Modified html2tei2 to strip out td @valign and td @width
* Output tei for:
  + eb03-r01
  + eb11-r02
  + eb09-r02
* The conversion process is working very well now, with 1-2 pages of html that my regex don’t catch. Further refinement possible, but this is quick and easy already.
* I have TEI for the first batch of all four editions and can begin R tests that don’t involve individual entries. But I’d get better results if I could clean the eb03 output first.
* Cloned Underwood’s DataMunging site to desktop. It contains his OCR normalizer. But the normalizer is a python script—it has a routine for disambiguating words that could be an error or correct by checking words on either side of them. Works on txt files. Runs fine on my DSC machine on the command line, and has enough information to try it.
  + Tried it on a group of 10 files. Converted to txt with tei-to-txt.xsl. The py script fixes many errors, but leaves others. A modest but significant improvement. Good enough to get started with text-mining the pilot group.

17-07-06

* eb03 170615 output has many errors. I tried improving recognition of the long-s, but Nahn is already limiting fonts to Caslon and his user-file is good (he doesn’t quite understand the idea of a ligament yet, so I need to talk with him about that).
  + The next step is to devise an R routine that can parse either the html or tei files and replace the most common errors mechanically. I think Gary devised a list of substitution last summer. Start there on Tuesday. **ſ**
* HEBREW LANGUAGE PAGES: New policy.
  + Do not use Hebrew in recognizing pages. Instead, identify them in the inventory. Later, we can go back to rerun those pages, but for now it causes too many problems.

17-07-05

* I’ve traced the problem with @dir=”rtl” to the Hebrew language setting. It reverses the order of characters from the original, so simply stripping @dir leaves us with erroneous output. When I remove Hebrew, the problem disappears. Unfortunately, AFR then does a horrible job with Hebrew characters. Could change the language settings for individual pages, but then they have to be output separately, rather than as part of the 250-page batch, so that slows things down. The rtl problem only seems to come up on pages with tables, though.
  + For now, the only solution I see is to keep the Hebrew and manually re-recognize and output without Hebrew those pages where it causes a problem. That means opening all 250 pages as an Oxygen project and validating, to locate the problems. About 20 pages came up in eb09.
* Re-recognized and re-output eb09 and transformed successfully into TEI. Used the above procedure. There were exactly 11/250 pages with dir-rtl problems. It took quite a long time, mostly from training the OCR (which Andrea never did) and re-entering @@ codes.
* Perhaps the best solution is to have the operator make a note on any pages that contain Hebrew and go back and output those separately, leaving it off for everything else.

17-06-27

* Found newer arl files for eb09. Moved them to External HD. Transformed to TEI. It has many @dir=”rtl” attributes within tables. I’ve removed Hebrew from languages, so don’t know why AFR is adding them. LOOK INTO NEXT.
* Transformed first batch (r1) out 250 pages of eb07 to TEI. Different from eb11 output in that nesting errors are with <span> rather than <font> elements, and sup/sub tags. Created new regex to correct.
* Updated transformation.md for github.
* Create spreadsheet (production.xlsx) to track what has been output to html, what converted to tei, then segmented to entries.
* Discovered some problems in collection of image files for eb03 and eb09—they start in the wrong place—and left a note for Nahn to correct.

17-06-22

* Successfully converted 250 pages of eb11 html to TEI using Oxygen and html2tei2. Discovered a need to run a regex find-and-replace on files to correct an error in tag nesting to make the files well-formed.
* Wrote up complete instructions for the html2tei transformation process and posted them on Github pages.
* Gary will test py script on the new tei pages.
* Showed Nahn the process we use for getting from AFR output to TEI-xml. I’ll have him check Gary’s py output against the originals to make sure all entries are properly segmented.

17-06-20

* html2tei—discovered the means of batch-converting html files in Oxygen. They have to be within a project, and then can be selected as a group and transformed. I’ve modified html2tei2.xsl to retain original filenames and output to the correct folder.
* eb11 missing text—I discovered two xml pages are missing text from the page (0785\_04, 0793\_02). The text is in the htm output, so this is a product of the py script. I’ve added a comment to both files where the text is missing. But clearly we’ll have to review all output pages after Gary finished revising the py script.
* eb11—the last 2 pages in the first 250-page group are actually unnumbered plates. I removed them and inserted the next two text pages in the group.
* eb11—Andrea did not combine even/odd pages for his latest output. I converted the most recent instance (161201) to AFR14, corrected an error in the last two pages of the group, and output to html. I then renamed and combined the output files into a single folder (170620), which we can use for running tei2html.
  + I found one instance where he combined odd/evens (161104), but it does not include text areas for the notes. I’ve converted it to AFR14 and added it to the production folder, as a backup.
* It is possible to add additional images to an OCR Project—AFR appends them to the end.
  + Click File > Open Image....
  + In the dialog box that opens, select one or more images and click Open. The images will be appended to the end of the open OCR project, and their copies will be saved in the OCR project folder.
  + You can also right-click images in Windows Explorer and select Convert with ABBYY FineReader 14 > Open in OCR Editor on the shortcut menu. The selected files will be added to a new OCR project.
* Copying from SilverHDD—Hidden files needed for AFR to recognize OCR Projects as such gets left behind. Instead, go back to SilverHDD and use AFR14 to open and convert the projects, saving to the production folder.
* eb07—Nahn discovered a sequence of original page images in eb07 where the same two page numbers are repeated with an asterisk. The continuity is correct, so we’ll have to incorporate it.

17-06-15

* Updated folder structure to accommodate multiple stages of file conversions. Also changed file and folder naming conventions.
* Nahn has finished basic AFR output for eb03 and eb09. I’ve moved them into the production folder.

17-06-13

* I modified html2tei.xsl to convert table attributes properly. The revised version is “html2tei2.xsl”.
* Transformed one of Andrea’s html files (with @@ coding). Ran it through html2tei.xsl. Result retains the @@ coding in the proper position. So we can safely change the production process. Instead of html>py>tei, we’ll do html>tei>py.
* Gary is working on modifying Andrea’s python script so it will work on tei files instead of html files. Basically, he needs to strip out all of the coding that converted html to xml-tei.
* Nahn is working on eb03. He was training the long-s incorrectly to map to the short-s, so we’re retraining it and he’ll see if accuracy of output improves.

17-05-24

* Footnotes test: opened Andrea’s afr file for eb11. It still has his @@@ and @@ additions in the output view. Converted to AFR14 and it retains his additions.
* Fixed problems in the eb07 image files—mispagination and one missing file found and added into the eb07-pilot-img folder.

17-05-23

* Tested AFR html-flexible and html-formatted output. The two have more in common than not. They both output with a .css file, but the only pertinent information in there is the font. In the text, it adds numerous *span* elements for font specifications. Some of these are incorrect—it identified some all-caps headers as small caps, and the entire paragraph was in that format; it missed instances of actual small caps. It also inserted &nbsp; entities at the end of lines, instead of spaces. Basic differences:
  + **flexible** puts entire page in a table for layout. It also include *text-align* and *text-indent* attributes. This includes coding that marks the column break: <td valign="top" width="50%">.
  + **formatted** skips the page layout table. It does not retain block formatting attributes, like text-align and text-indent. Unfortunately, it also eliminates the column break code.
  + **html2tei**script does not import the .css information. It converts &nbsp; to normal spaces. It converts table elements properly, but it does not convert the table attributes to the corresponding tei attributes, nor does it delete them. This causes an error in tei-xml file for the following illegal attributes:
    - html:cellpadding
    - valign
    - width
    - rowspan
    - colspan
  + Everything else in the test page converted fine. Apparently, the xsl script works well, but it needs additional coding for tables.
* CONCLUSIONS
  + The advantage for the flexible format is in identifying the column break. Everything else will need to be stripped out. Removing the page layout table would be easy, because it has a regularity to it at the opening and end of the document. (Andrea worked in flexible layout, but output everything in formatted.)
    - *Question: can I remove it using xsl?* If so, I could add it to the html2tei script, so there would not be additional steps, and I would retain the column break. The files would also retain slightly more information, like indents for paragraph breaks, that could be helpful.
* TO DO
  + Find the .htm files output by AFR for eb11 and eb09. If they don’t exist, rerun the output from the AFR files. Then see how difficult it will be to convert them into TEI-xml using html2tei before running the py script that will join the files with identifying page numbers, and segment them into entries.

17-05-19

* Reorganized all project folders, to tame the mess and make it possible to see what we have.
* Created new document explaining project organization structure.