Meeting with Sinai:

* Looked through the versions and files Dr. Blackwell had made
  1. GitHub clone of Sinai’s work
     + converted the .tsv into .cex for DUCAT compatability
     + discussion of urn’s
       - urn:cts:\*namespace\*:\*textgroup.work.version.exemplar\*:citation hierarchy:
     + no canonical grouping of Hebrew text (i.e. lines, paragraphs, versus, etc.)
* Discussion of new development of files with new and additional breakdowns
  1. i.e. more specific/meticulous .xml styles to match English and Hebrew to allow for a more in-depth and easier alignment in DUCAT
     + Final project will be derived from the \*FINAL\* .xml file to keep all version to come from the same root
* Things to come:
  1. add map for the alignment piece, with GeoNar or Placenames?
  2. Keep page breaks, headers, and segs? for the Hebrew edition
     + more easily sited and blocked for sectioning
  3. Keep additional formatting with the recogito <placename>
  4. resolution for the recogito data for easy access to coordinates
  5. Advancements in DUCAT:
     + shortcut for the alignment key

OR

* + - hovering of the top tool bar as you scroll for ease of access to the alignment key
  1. Separation of the Placenames from the prefix to allow for individual tokenization alignment
* Side Note:
  1. Hebrew “:” = English “.”
  2. Hebrew “.” = English “,”
  3. Hebrew “,” = English not identified
* Goals for next meeting: February 18th at 9am EST
  1. Setup GitHub repository
     + post notes
  2. Hebrew preliminary view of .cex with the Asher and Adler texts – Blackwell
  3. Manipulation of original Hebrew and English for break points/canonical annotation for ease of alignment – Sinai
  4. Manipulate DUCAT maybe? – Blackwell
  5. Setup Zoom meeting link for Feb. 18th – Mackense – Done

#Aligning hebrew reloaded (2nd meeting)

**CB on scripts to create .cex files: config.sc**

The "Scala" directory > cexy.sc is a script that will read in XML and generate one .cex file for each document (and it has been adapted to generate more to allow the creation of various versions for different tasks).

(By "text" we mean text units that should be considered separate, not just many XML files: for example, an editorial introduction and the actual text in one document, footnotes separated from the actual text, or various versions of the same text).

**config.sc contains a "Text Config" section where the script can be configured to process any number of XML files (CAVEAT: it only handles 1-2 levels deep citation schemes for the moment, in our particular case (1st top-level division) and (second-level divisions)).**

**CexyConfig defines the configuration of each text. It contains:**

* filePath pointing to the XML file. In each CexyConfig, it is possible that the filepath will be the same, if you are trying to differentiate between two sections of the file that pertain to two texts (e.g. the editorial introduction to Benjamin of Tudela and Tudela's text itself)
* topNodes describes the path to where the actual text is in the TEI (i.e. in the element in our case)
* otherNodes indicates the other element that contains text, i.e. in our case. It can also be an empty string in 1-level deep xml.
* stripFromIDs strips selected XML ids from the citation scheme: for example, while in XML element ids are necessary with a certain syntax, in our architecture the id name can be removed, leaving just the standard order of text divisions (e.g. GeoNar1 in XML becomes simply 1 in .cex)
* omitNodes strips all elements that intrude in the actual text, such as footnotes. They are not going to appear in the .cex file.
* urnString: metadata that will go in the .cex file (can be arbitrarily defined)
* textGroup, work, version, citation, lang: all the necessary parts of the metadata that "explain" the different sections of the URN with human-readable things, such as the author of the text, title, edition.

**The second CexyConfig**

Pertains to the transformation of ONLY the editorial introduction in our file, i.e. treats it as a separate text. The criteria are largely the same, obviously with different elements to strip/omit and a different citation. (for this particular transformation we converted all TEI elements in the file into elements with IDs corresponding to "head", "body", etc., so that the transformation could be based off of the same elements)

**Forbidden characters**

* "#" replaced with another UniCode character that looks exactly the same, so that it does not mess up with the CTS URN
* linebreaks \n
* multiple spaces " +""
* space and comma " ,"
* space and period " \."

**Command line instructions**

sbt console (runs the script) :load cexy.sc

It will create a file named "text.cex" that contains:

* Information about the cts library
* ctscatalog with information about the texts
* various ctsdata, one per text, organized as a two-column expression with urn and corresponding text passage, with exclusively the words that belong to the selected text (no footnotes, no editorial intro, etc.).

**Using the .cex files in the CITE environment**

For exploration of the text itself, or for making simple queries (e.g. n-grams, string search). Each query also produces the corresponding citation to the passage, which can be called in the environment to be read in context.

**Other notes**

It is also possible to configure the script so that it includes desired XML tags and elements, such as the placenames and their IDs, so that they can be viewed in a CITE reading environment.

The reason why semantic annotations (for example , or elements) included in the TEI are NOT included in .cex transformation as citation elements, is because they do NOT indicate sections of the text, but indexable material that can be processed to query the text for those things and produce a separate file, an "index".