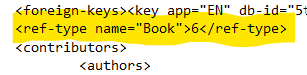
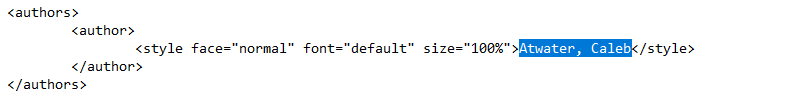
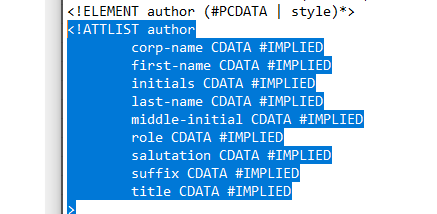
**Semi-automatic workflow to get by for now**

1. Export Forms data manually
2. Run an R script in RStudio to clean and transform Forms data into EndNote-acceptable XML
   1. Inputs:
      * 1. XLSX of Forms data
        2. XLSX field -> XML tag map
   2. Data processing:
      * 1. Map XLSX columns to XML columns (merge, concatenate, mutate, etc.)
        2. Compare XLSX to previous XLSX. Need to test how EndNote deals with duplicate records but we may have to remove duplicates before transforming to XML.
        3. Transform data frame to XML text file.
        4. Write XML file to directory.
   3. Outputs:
      * 1. XML of Forms data
3. Manually import data into EndNote
   1. Use EndNote’s native File -> Import function

**Final implementation**

1. Automate Forms data export and cleaning (this is also a good idea from a data backup perspective)
   1. Power Automate
      1. This would produce an XLSX like the one you’re already manually pulling out but on a schedule and with a consistent target directory and file naming convention.
      2. A Power Automate routine that extracts Forms data and then runs an R script to clean and transform data to XML.
      3. <https://techcommunity.microsoft.com/t5/microsoft-forms/api-to-access-ms-forms/m-p/1463830>
   2. R script
      1. A scheduled Windows task that runs an R script to extract Forms data to data frame via the Forms API, then clean and transform data, and save the data as XML.
      2. This would produce a data frame that would undergo cleaning and then be written be written to a specified directory as XML.
      3. <https://techcommunity.microsoft.com/t5/microsoft-forms/api-to-access-ms-forms/m-p/1463830/page/2>
2. Manually import data into EndNote
   1. Use EndNote’s native File -> Import function
   2. I see no way to automate this step. The Endnote API is only available via an EndNote Plugin. Plugins ‘work’ by calling their Init() function on startup of an EndNote Desktop session. That means EndNote’s API functionality (e.g., ImportRecords()) is unavailable outside of an EndNote session.

**To do:**

* Do you have control over field name vs field alias? Name is what outputs into the xlsx. Alias is what what the user sees when viewing the form. If yes:
  + Rename Forms fields in lower snakecase
    - No capitalization and no spaces. E.g.,
      * ‘Series Title’ becomes ‘series\_title’
    - No special characters. E.g.,
      * ‘Author(s)’ becomes ‘authors’
    - Use a descriptive name instead of numbers. E.g.,
      * ‘Volume2’ becomes ‘volume\_journal’
      * ‘Volume3’ becomes ‘volume\_book’
      * ‘Year2’ becomes ‘year\_journal’
      * ‘Year3’ becomes ‘year\_book’
* Pretend you’re a user of this tool and enter in both Forms and Endnote one real source of each reference type you’re using. E.g., enter the same book in both Forms and Endnote. Then export the XLSX Forms output and the XML Endnote output and send those to me.
  + At least one example needs to include an editor because the Endnote XML documentation doesn’t include an editor field…
  + These exports will give me a lot of useful information:
    - What does Endnote do with blank or missing fields? XLSX leaves a blank but still includes the column. I need to know if Endnote is expecting me to pass empty XML tags or just omit empty tags.
    - Endnote seems to use both a type name (a character value like “Book”) and a numeric ‘ref-type’ id (a key field like 6). The Form output only provides the type name (“Book”). I’ll need to build a lookup table of those key-value pairs to recreate ‘ref-type’ in the script.
* Revise author field into multiple fields?
  + To match EndNote, you may need to make your author(s) field more complex. Maybe with a first and last name box and a + button to add another author, like Zotero.
  + I want to see XML examples of how Endnote parses author into the fields the documentation claims it stores.
    - The Atwater, Caleb example looks like one character string.
    - Documentation says endnote parses several author name attributes and I’d need example syntax to match their style:

**Map of XLSX -> XML fields**

* + We need to map each column from the spreadsheet Forms output to its correct XML tag (i.e., a field in EndNote).
  + The R script will use this map to merge repeated XLSX fields into the correct XML tag.
    - E.g., Title, Title2, Title3… will merge into:
    - <titles><title>An article’s title</title></titles>

|  |  |  |
| --- | --- | --- |
| XLSX | XML | Correct? |
| Reference Type | <ref-type name="">  </ref-type> |  |
| Title; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; Book Title; | <titles><title><style>  </style><title></titles> |  |
| Author(s); 2…9; Author(s)/Reporter(s) | <contributors><authors><author><style> </style></author></authors></contributors> |  |
| Series Title; Series title2; Series Title3 | ? |  |
| Year; 2…10 | <dates><year><style>  </style></year></dates> |  |
| Attach files (i.e., pdf, photograph); 2…10 | <urls><pdf-urls><url>  </url>><pdf-urls><urls> |  |
| Where did the research take place/what is the location of the reference?; 2…10 | ? |  |
| Reference description; 2…10 | ? |  |
| Cover type; 2; 3; 4; 5 | ? |  |
| Stable URL or DOI; 2…10 | <urls><related-urls><url><style>  </style></url></related-urls><urls> |  |
| Journal; Newspaper | <periodical><full-title>  </full-title></periodical> |  |
| Volume; 2…5; Series Volume; 2; Number of Volumes; 2 | <volume>  </volume> |  |
| Issue; 2 | <issue>  </issue> |  |
| Page Range; 2…4 | <pages>  <end> </end>  <start> </start>  </pages> |  |
| Series Editor; Series editor2; Series Editor3 (AD; BC; DU) | ? |  |
| Place Published; 2…3; Place published 4 | <pub-location><style>  </pub-location></style> |  |
| Institution (AG) | ? |  |
| Document Number (AJ); 2 | ? |  |
| Number of Pages; 2 | <pages>  </pages> |  |
| Publisher; 2…4 | <publisher>  </publisher> |  |
| Edition; 2 | <edition>  </edition> |  |
| Session (BN) | ? |  |
| Paper Number (BO) | ? |  |
| Editor; 2 | ? |  |
| Conference name (BQ) | ? |  |
| Conference location (BR) | ? |  |
| Academic Department (CE) | ? |  |
| University (CF) | ? |  |
| Degree (CG) | ? |  |
| Advisor (CI) | ? |  |
| Chapter Number (DD) | ? |  |

Next steps

1. Immediate test. Can we import endnote-generated XML back into EndNote?
2. Extract xml schema from example endnote outputs?
   1. We need to extract tags and programmatically compare our output to a ‘known-correct’ EndNote XML schema.
   2. <https://www.r-bloggers.com/2017/01/using-xml-schema-and-xslt-in-r/>