Author Split Program

Austin Vandegriffe

agv8c7@mst.edu

May 16, 2018

1 Basic Use

In the 'Author Split' directory you will find a shortcut application linked to the author split executable located in '.../CODE/dist/author_split'.

1. Double click on the application shortcut.
2. After a couple seconds you should be prompted with a list of XLSX files in the 'Author Split' directory.
3. Enter the number corresponding to the file you wish to split.
4. After a couple seconds a string of authors should appear in the prompt, this is a complete list of every author in the file, if they were part of our institution they will have an email and an institution. There is no need to check this; an XLSX file has appeared in the 'Author Split' directory with a '_Complete', this output file is what you want.
5. Back on the command prompt you will notice a list of authors and numbers again, repeat steps 1-4 until you run out of files or choose to close the program. Note: When no authors remain just hit 'Enter' twice and the program will end.

2 Maintenance

The source code can be found in the 'CODE' directory titled 'author_split.py'. This code relies on the 'Authority Control_Lookups - Faculty.XLS' file for author lookups (explained later), if this file must be updated **make sure its replacement is titled '*Authority Control_Lookups - Faculty*' and has the *XLS* extension**; if this is not done the program will fail because it uses the Python library XLRD which is only compatible with XLS extension; if this must be changed to a new extension one will need to rewrite all places XLRD was used but the logic should be the same (OpenPyXl is the library compatible with XLSX).

If the column of the authors is changed in the files put the new column letters at the end of the text file located in the 'Author Split' folder, the text file name should start with 'zzzz_Line_155_', **this should be the ONLY txt file in this directory, the program will FAIL otherwise**.

3 A Walkthrough

The 'Authority Control_Lookups - Faculty.xls' is loaded in and stored as 'authority_control_lookup'. The program then enters a while loop which is ended by the input '' (just hitting enter when prompted 'Selection Number: '). The path to the 'Author Split' directory is entered into the 'path' variable where Python's OS library changes the working directory to 'path'. All files in the 'path' directory are scanned for '_Complete' then added to a string for a record. It then passes to a FOR loop which parses the directory again for XLSX files which do not have '_Complete' in them nor whose name belongs to the '_Complete' string from the previous loop. The text file's name with author column in it extracts the column and stores it in the 'author_column' variable for later use. Files which were successfully make it through the filter are displayed in the selection prompt; the input selection is taken in and the file is loaded from a list based on the input index. From here 'main()' is ran.

An error log is created to capture any errors which may occur in the program (most errors will occur due to diacritics in the name, just remove the diacritics and rerun the program; others errors are from missing data or incorrect file formatting, i.e. incorrect author row). The author column is then parsed from row 2 to the end max column, each iteration stopping at the cell, parsing the authors in the cell separated by the 'and' delimiter; from here the name is ran through the 'filerName()' function which takes a string as the parameter and returns a Python dictionary with keys: first, last, middle, suffix, email, and institution, all corresponding respectively to the first name, last name, middle name, suffix, email (empty for now), and institution (empty for now). The author is then searched for in the authority control spreadsheet, if they are there the email is filled with the email in the spreadsheet and the institution is filled with 'Missouri University of Science and Technology' (the 2008 name change truncates any historic name even if the professor worked for MSM or UMR); if the is not in the spreadsheet then they are not ours or their name does not correspond to any on file and their email and institution are left blank. The author dictionary is then stored a Python dictionary named 'authorDict' with a key corresponding with the name (if a publisher has multiple names they will have multiple occurrences in this dictionary). After all authors in the initial cell have this filter ran on them the cells starting at BK are filled using the record author function which takes a row integer, column integer, and name string as

parameters and writes, holding the row constant, the first name, middle name, last name, suffix, email, and institution for all authors up to 27 authors.

After this is ran you will be prompted for another spreadsheet and the process repeats until ending criteria is met.