Workflow Automation for Marketing Leads Administration

Created by: Vladimir Martintsov (https://www.linkedin.com/in/vladimir-martintsov/)

Version: 1.0

Date: August 1st, 2019

Definitions used in this document:

1) **Python –** development framework/programming language used to develop scripts
2) **Software Package –** supplementary software pre-made modules used in architecture of the main code/program. Usually require an additional manual installation**.**
3) **Script –** a snippet of code that automates *small repeating tasks (NOT entire processes)*
4) **Directory –** folder, where files are stored
5) **File Extension –** format in which files are store such as .csv or .xls
6) **Text Editor –** software that recognizes different programming languages and code written in them
7) **Command Prompt(cmd) –** OS default interface that runs applications and scripts (on Windows, available as Start -> Command Prompt (typing "Command Prompt" will launch it))
8) **SQL Database –** an organized (by a few column) collection of data that is manipulated via SQL programming language
9) **Admissions Database –** a database that contains information about people who have applied for at least one program at OsgoodePD
10) **Lead –** a potential customer/visitor who expressed interest in certain program offered by OsgoodePD
11) **Weird Lead –** A Lead, whose name is too short, or matches the names of OsgoodePD employees (who were simply testing the system), or email matches to the one of an OsgoodePD employee, or the Type of Lead or the Specialization are not automatically distinguishable by the Script
12) **Source –** marketing platform that records Lead's contact information through means of an informative website advertising a program at OsgoodePD
13) **Converted Lead –** a person who expressed their interest on one of the Sources, and appeared to apply for a program at OsgoodePD afterwards (appeared in Admissions Database)
14) **Raw Source Documents –** Excel documents downloaded from Sources with Lead's information, which are usually organized by default Source's format
15) **Raw Source Document Format –** Structure of the Raw Source Document – like the database – the column names under which the customer's information is organized
16) **Clean Source Document –** an organized document containing the clean data about the Leads extracted from specified Raw Source Documents. Information in this document is ready to be stored in Main Leads Document or Marketing Database.
17) **Main Leads Document –** a legacy Excel document that contains information about Leads from multiple Sources for the past 2 years. It also contains the list of Converted Leads.
18) **Marketing Database –** a MS Access database that stores information about the Leads in a similar table format like the legacy Main Leads Document
19) **Data Entry –** A sequence of characters or numbers in an Excel cell – usually meaningful data
20) **Keyword –** a sequence of characters that appears to be in the Data Entry

21) **Dirty Data –** data entries that contain extra characters, which makes them inconsistent with majority of other data entries, but still have a proper meaning
22) **Specialization –** the program of interest that was chosen by the Lead on the Source Page
23) **Query –** a "question" that is being asked from the Database

Introduction and High Level Goals of this Automation project

Automation of the workflows allows for efficient performance of many working teams. The goal of this workflow automation is to alleviate the manual work that the Marketing team must perform to keep their Leads records up to date. Manual work, due to presence of human errors, can cause the data to be dirty and disorganized. The record update activity is anticipated to be done on bi-weekly or monthly basis, and usually involves two steps:

1) Collecting the Lead's contact information from all Raw Source Documents
2) Cross checking each record against the Admissions Database, and storing the new Leads in Main Leads Document and updating the Converted Leads list

This above workflow is primarily automated by having two respective separate software solutions that the Marketing team may use to manipulate their records more efficiently and in timely manner.

1) Python Script – a script, which systematically goes over Raw Source Documents and automatically produces Clean Source Document ready to be imported into Marketing Database
2) MS Access Marketing database – a SQL database that allows to quickly and automatically extract the relevant information by the means of SQL queries (i.e. Converted Leads list can be created by the means of queries and not manually)

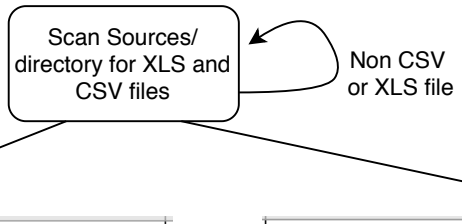
Both Python Script and Marketing Database are highly complex software solutions and require prior understanding of their functionalities. We will follow the sequence of which we introduced them here and start with the Python Script.

Python Script - LLM_Leads_Extracting_Script.py

As mentioned above, the purpose of the Script is to automate the collection of data about Leads from Raw Source Documents, clean the Dirty Data and put the processed information into a Clean Source Document. The Clean Document, once ready, will contain a Sheet with Leads information and another Sheet with Weird Leads.

As the Script extracts the data from the Raw Source Documents, one-by-one. It removes the duplicate entries, sorts out the Weird Leads, cleans the data where possible, prepares it for export, and finally exports it. We will cover the entire process required software installation in Appendix A of this document.

The entire workflow of the Script is best described in the diagrams on the following pages.



Functionality	Script Line Number/ Comments	Handling
Read the File Columns	55-67	Exception, if column not found or date is not
Get the Name, Email, Type of Lead, Specialization 1	Lower all the letters, strip the trailing/ leading white spaces	
Clean the City	Lowers all the letters, only first is capital	Keep only Alphanumeric characters, periods and spaces
Clean the Province	Lowers all the letters, only first is capital If only 2 letters - keep both capital (i.e ON)	Keep only Alphanumeric characters, periods and spaces
Extract Weird Leads	30-47 + Emails w/ spaces, commas and wrong number of '@'	Extract the Source File name the Weird Lead came from
Remove the Duplicates	Criteria: First+last Name + Specialization OR Criteria #2: Email + Specialization	
Add the cleaned records into respective pools of Leads	Weird Leads are separated from Normal Leads, respective leads counts are updated	

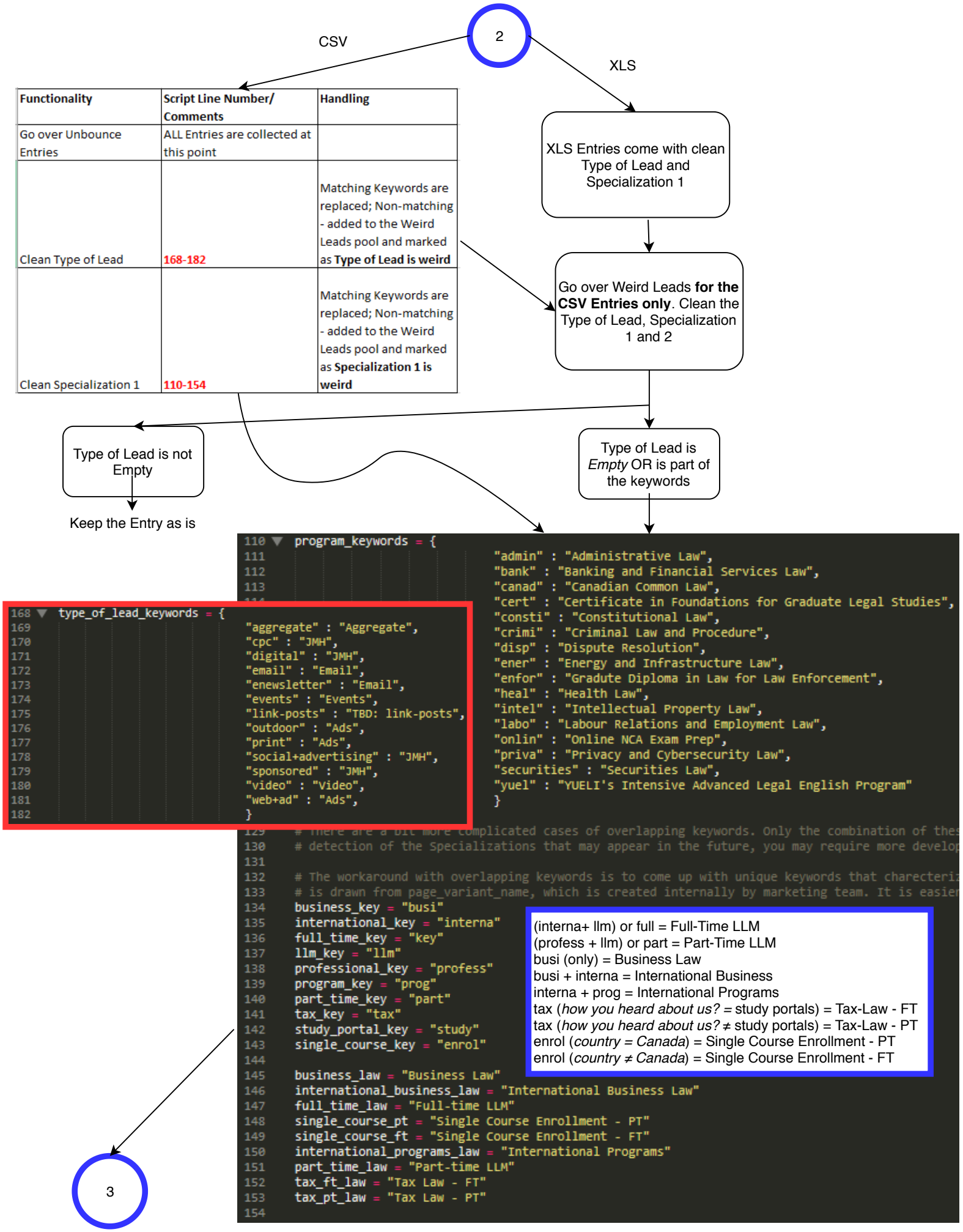
Functionality	Script Line Number/ Comments	Handling
Read the File Columns	80-94	Exception, if column not found or date is not parsed
Get the Name, Email	Lower all the letters, strip the trailing/ leading white spaces	
Clean the City	Lowers all the letters, only first is capital	Keep only Alphanumeric characters, periods and spaces
Clean the Province	Lowers all the letters, only first is capital If only 2 letters - keep both capital (i.e ON)	Keep only Alphanumeric characters, periods and spaces
Extract Weird Leads	30-47 + Emails w/ spaces, commas and wrong number of '@'	Extract the Source File name the Weird Lead came from
Remove the Duplicates	Criteria: First+last Name + Specialization OR Criteria #2: Email + Specialization	
Add the cleaned records into respective pools of Leads	Weird Leads are separated from Normal Leads, respective leads counts are updated	

```
54
55      unbounce_date_submitted_column = "date_submitted"
56      unbounce_time_submitted_column = "time_submitted"
57      unbounce_first_name_column = "first_name"
58      unbounce_last_name_column = "last_name"
59      unbounce_email_column = "email"
60      unbounce_how_you_found_about_us = "utm_source"
61      unbounce_type_of_lead_column = "utm_medium"
62      unbounce_specialization_column = "page_variant_name"
63      unbounce_jd_llb_column = "do_you_have_a_jdllb"
64      unbounce_phone = "Phone Number"
65      unbounce_city = "city"
66      unbounce_province = "stateprovince"
67      unbounce_country = "country"
```

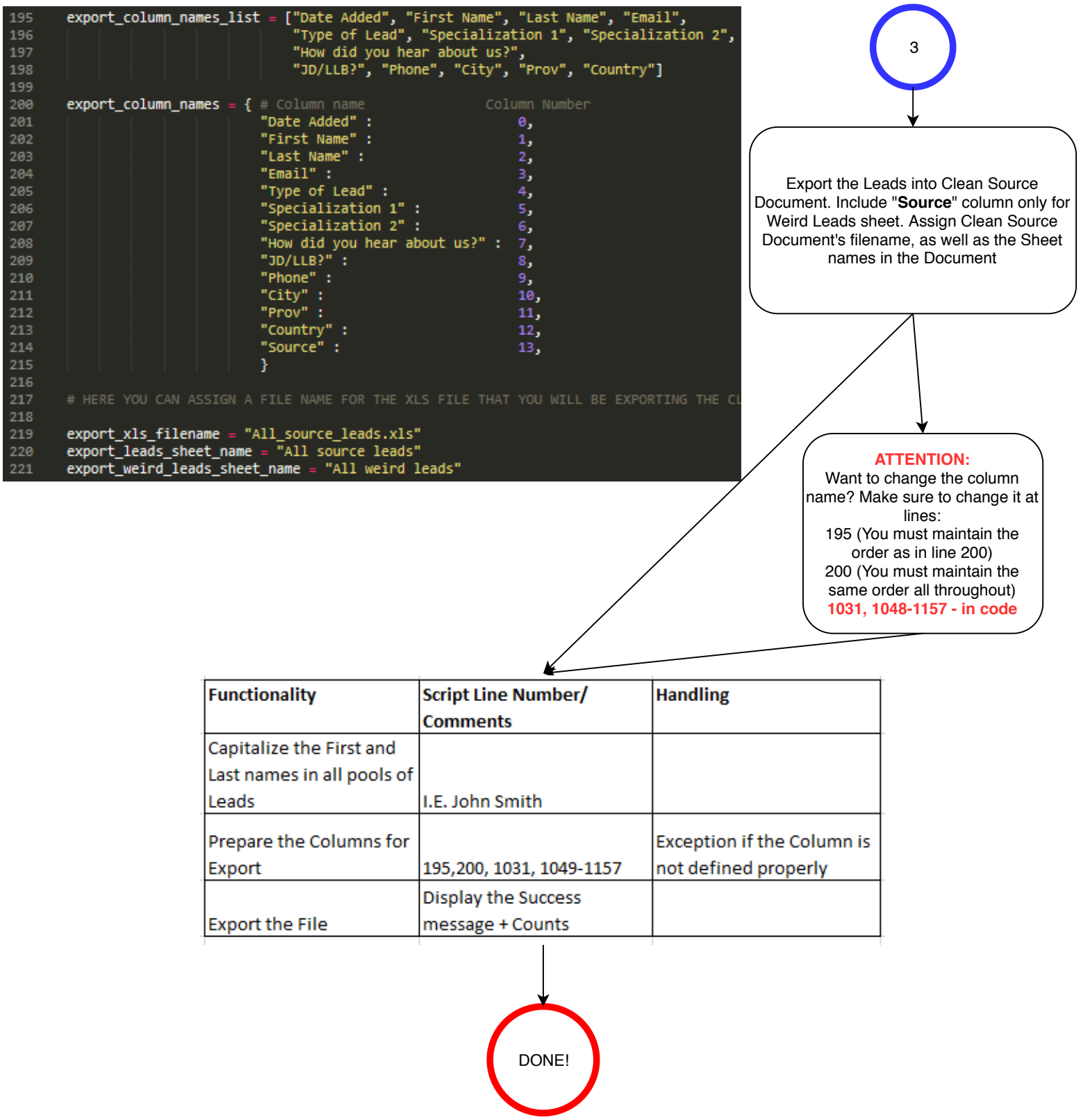
```
79
80      machform_date_column = "Date Created"
81      machform_first_name_column = "Name - First"
82      machform_last_name_column = "Name - Last"
83      machform_email_column = "Email"
84      machform_specialization_column1 = "Program of Interest 1"
85      machform_specialization_column2 = "Program of Interest 2"
86      machform_how_you_found_about_us1 = "How did you first learn about the program?"
87      machform_type_of_lead_column = "Type of Lead"
88      machform_jd_llb_column = "Do you have a JD/LLB?"
89      machform_phone = "Phone Number"
90      machform_city = "City"
91      machform_province = "State/Province/Region"
92      machform_country = "Country"
93      machform_recieve_updates = "I would like to receive updates about my" +
94              + " program of interest from Osgoode Professional Development. -- "
95
```

```
27      #-----------------------------------------------------------------------------
28      # ADD THE POSSIBLE MARKETING LEAD'S NAME HERE. NOTE: WHEN A RECORD WITH THIS NAME
29      # Weird records will be exported separately for manual verification
30      osgoode_leads = ["stewart laszlo", "laszlo stewart",
31                       "test ", "asd ", "test test", "asd asd",
32                       "farzana crocco", "crocco farzana", " ",
33                       "patricia pazos", "andrea chau",
34                       "sarah alexander"]
35
36      # ADD THE POSSIBLE MARKETING LEAD'S EMAIL HERE. NOTE: WHEN A RECORD WITH THIS EMAI
37      # Weird records will be exported separately for manual verification
38      osgoode_leads_emails = ["achau@osgoode.yorku.ca","sambutt@yorku.ca",
39                              "sambutt@osgoode.yorku.ca","rbahrami@yorku.ca",
40                              "rbahrami@osgoode.yorku.ca","ttendean@osgoode.yorku.ca",
41                              "ttendean@yorku.ca","slaszlo@osgoode.yorku.ca",
42                              "jmelancon@osgoode.yorku.ca",
43                              "eshallerauslander@osgoode.yorku.ca",
44                              "ppazos@osgoode.yorku.ca", "salexander@osgoode.yorku.ca",
45                               "speakto@gmail.com", "chaumein@gmail.com",
46                                "speaktosam@gmail.com", "test@gmail.com",
47                               "pdabek@gmail.com"]
```

2


2
CSV
XLS
Functionality
Script Line Number/ Comments
Handling
Go over Unbounce Entries
ALL Entries are collected at this point
Clean Type of Lead
168-182
Matching Keywords are replaced; Non-matching - added to the Weird Leads pool and marked as Type of Lead is weird
Clean Specialization 1
110-154
Matching Keywords are replaced; Non-matching - added to the Weird Leads pool and marked as Specialization 1 is weird
XLS Entries come with clean Type of Lead and Specialization 1
Go over Weird Leads for the CSV Entries only. Clean the Type of Lead, Specialization 1 and 2
Type of Lead is not Empty
Keep the Entry as is
Type of Lead is Empty OR is part of the keywords
type_of_lead_keywords = {
"aggregate" : "Aggregate",
"cpc" : "JMH",
"digital" : "JMH",
"email" : "Email",
"enewsletter" : "Email",
"events" : "Events",
"link-posts" : "TBD: link-posts",
"outdoor" : "Ads",
"print" : "Ads",
"social+advertising" : "JMH",
"sponsored" : "JMH",
"video" : "Video",
"web+ad" : "Ads",
}
program_keywords = {
"admin" : "Administrative Law",
"bank" : "Banking and Financial Services Law",
"canad" : "Canadian Common Law",
"cert" : "Certificate in Foundations for Graduate Legal Studies",
"consti" : "Constitutional Law",
"crimi" : "Criminal Law and Procedure",
"disp" : "Dispute Resolution",
"ener" : "Energy and Infrastructure Law",
"enfor" : "Gradute Diploma in Law for Law Enforcement",
"heal" : "Health Law",
"intel" : "Intellectual Property Law",
"labo" : "Labour Relations and Employment Law",
"onlin" : "Online NCA Exam Prep",
"priva" : "Privacy and Cybersecurity Law",
"securities" : "Securities Law",
"yuel" : "YUELI's Intensive Advanced Legal English Program"
}
# detection of the Specializations that may appear in the future, you may require more develop
# The workaround with overlapping keywords is to come up with unique keywords that charecteri
# is drawn from page_variant_name, which is created internally by marketing team. It is easier
business_key = "busi"
international_key = "interna"
full_time_key = "key"
llm_key = "llm"
professional_key = "profess"
program_key = "prog"
part_time_key = "part"
tax_key = "tax"
study_portal_key = "study"
single_course_key = "enrol"
business_law = "Business Law"
international_business_law = "International Business Law"
full_time_law = "Full-time LLM"
single_course_pt = "Single Course Enrollment - PT"
single_course_ft = "Single Course Enrollment - FT"
international_programs_law = "International Programs"
part_time_law = "Part-time LLM"
tax_ft_law = "Tax Law - FT"
tax_pt_law = "Tax Law - PT"
(interna+ llm) or full = Full-Time LLM
(profess + llm) or part = Part-Time LLM
busi (only) = Business Law
busi + interna = International Business
interna + prog = International Programs
tax (how you heard about us? = study portals) = Tax-Law - FT
tax (how you heard about us? ≠ study portals) = Tax-Law - PT
enrol (country = Canada) = Single Course Enrollment - PT
enrol (country ≠ Canada) = Single Course Enrollment - FT
3

```
195  export_column_names_list = ["Date Added", "First Name", "Last Name", "Email",
196                              "Type of Lead", "Specialization 1", "Specialization 2",
197                              "How did you hear about us?",
198                              "JD/LLB?", "Phone", "City", "Prov", "Country"]
199
200  export_column_names = { # Column name                Column Number
201                          "Date Added" :                  0,
202                          "First Name" :                  1,
203                          "Last Name" :                   2,
204                          "Email" :                       3,
205                          "Type of Lead" :                4,
206                          "Specialization 1" :            5,
207                          "Specialization 2" :            6,
208                          "How did you hear about us?" :  7,
209                          "JD/LLB?" :                     8,
210                          "Phone" :                       9,
211                          "City" :                        10,
212                          "Prov" :                        11,
213                          "Country" :                     12,
214                          "Source" :                      13,
215                          }
216
217  # HERE YOU CAN ASSIGN A FILE NAME FOR THE XLS FILE THAT YOU WILL BE EXPORTING THE CL
218
219  export_xls_filename = "All_source_leads.xls"
220  export_leads_sheet_name = "All source leads"
221  export_weird_leads_sheet_name = "All weird leads"
```

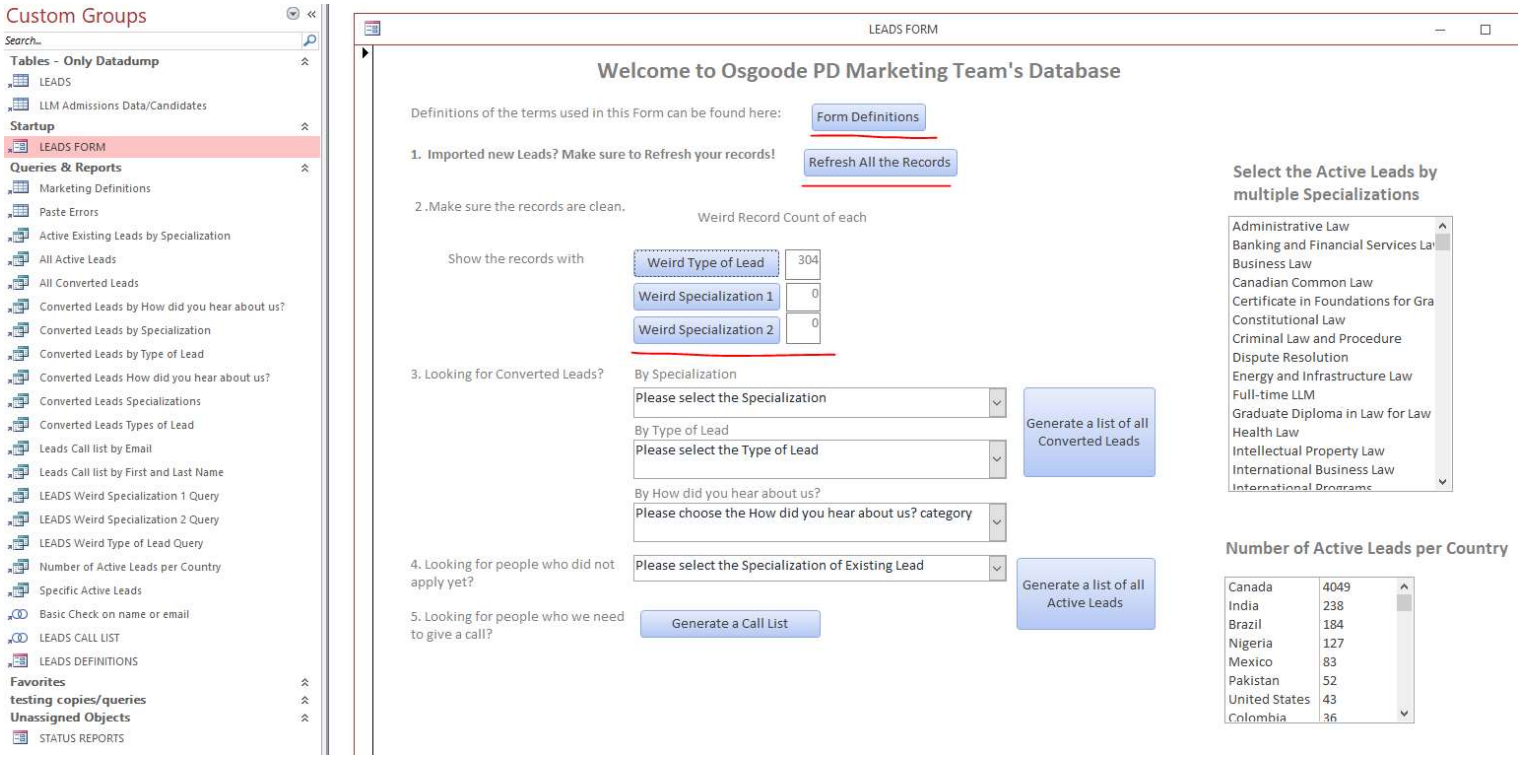


Functionality	Script Line Number/ Comments	Handling
Capitalize the First and Last names in all pools of Leads	I.E. John Smith	
Prepare the Columns for Export	195,200, 1031, 1049-1157	Exception if the Column is not defined properly
Export the File	Display the Success message + Counts	

MS Access Marketing Database

MS Access – a well-known database software solution that relies on SQL language. MS Access is very intuitive and most of the time does not require knowledge of SQL to build the Queries. This allows a faster lookup of required records with a single push of a button. The workflow in MS Access is presumed to be as follows:

1) Update the LEADS and LLM Admissions tables. LEADS updates come from the Clean Source Documents, LLM Admissions - from Admissions Team. The new records *must come in Excel document format, which are then "Appended" to the respective tables in Access.*
2) Navigate to the LEADS FORM and hit Refresh "All the Records" button. Reassure that the Records are clean by checking the Weird Leads who have Weird Specialization or Type of Lead. Clean the records, if needed.
3) Check Form Definitions to learn more about the terms used in this Form
4) Feel Free to add **new** Buttons/Queries! **DO NOT MODIFY THE EXISTING QUERIES!**



Appendix A – Python Script installation and operating instructions

Installation and System requirements

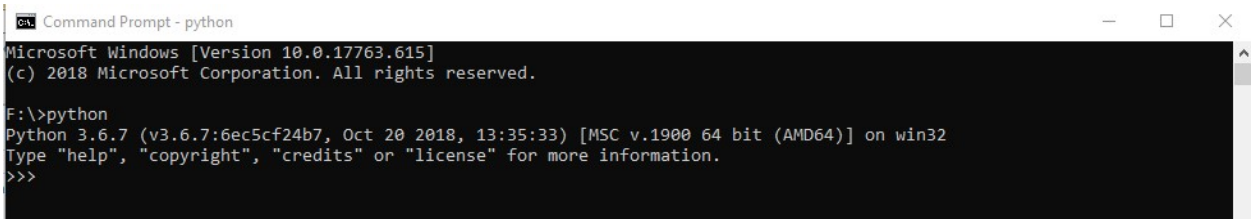
The installation of the components and Software Modules required by the Script are better to be performed by an IT specialist; however, the installation steps are not as difficult as they may seem. There are a few software products that go together to run the Script and modify it, if needed.

You will need Administrator privileges to install ALL the software below.

Python

1) Python Development Framework **Version 3.6 (not older!)**, which is available on this website: https://www.python.org/downloads/release/python-360/ . Choose the applicable Operating System, in case of Windows 10, you may select Windows x86-64 executable installer, and download it. *You may also download a newer version of Python, you may find the installer online as well.*
2) PIP (or pip) Python installer – **this feature comes in automatically in the above Python 3.6 installer.** If you are installing a different version of Python framework, make sure that pip is also included in the installation.
3) Adding Python to the Environment Variables – **this feature comes automatically in the above Python 3.6 installer.** If you are installing a different version of Python framework, make sure that Environment variables are configured accordingly.

At this point you should have Python installed. You may test it by opening the Command Prompt and typing "python" command – it should look something like this:



Feel free to close this window, once you are done. **If typing "python" gives you an error, restart your machine and try again. If it still does not work – make sure the Environment Variables are set properly.**

Python Software Modules

Now that you have Python and pip installed, you are ready to install the necessary Python modules that help the script throughout the execution.

1) Open Command Prompt and type the following command: pip install _____, where ____ is the name of the Software Package you need to install. The packages that you need to install are as follows, in order:
pprint, xlrd, xlwt, datetime, time, python-dateutil

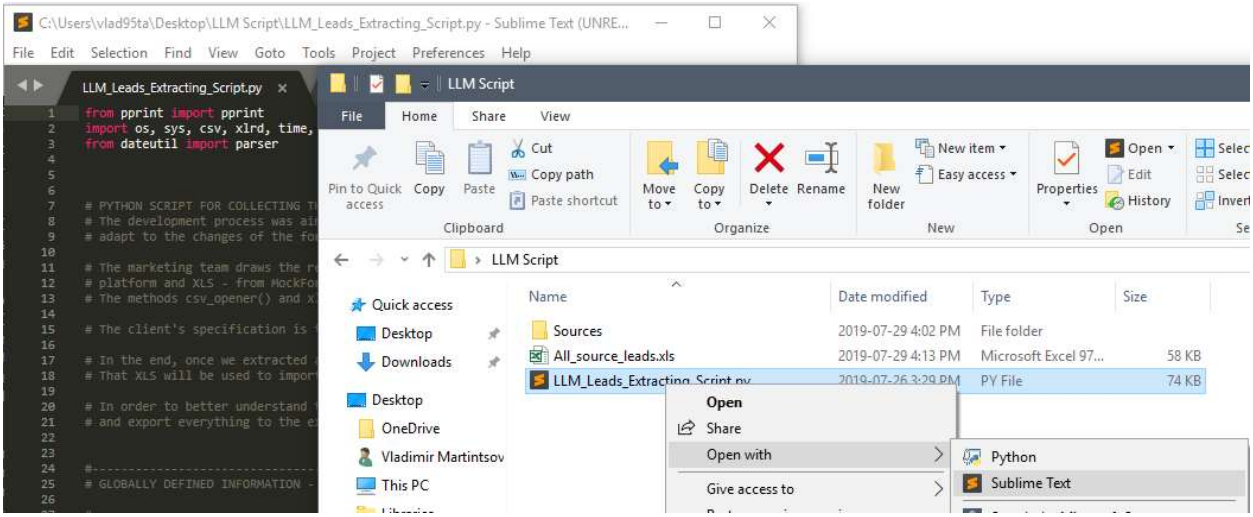
*An example of the **datetime** module installation using Command Prompt and pip can be found on the screenshot on the next page. Repeat the same steps for other packages as well!*

```
C:\Program Files\Sublime Text 3>pip install datetime
Collecting datetime
  Downloading https://files.pythonhosted.org/packages/73/22/a5297f3a1f92468cc737f8ce7ba6e5f245fcfafeae810ba37bd1039ea01c
/DateTime-4.3-py2.py3-none-any.whl (60kB)
    |████████████████████████████████| 61kB 1.9MB/s
Collecting zope.interface (from datetime)
  Downloading https://files.pythonhosted.org/packages/da/08/726e3b0e3bd9912fb530f9864bf9a3af9f9f6a1dfd4cc7854ca14fdab441
/zope.interface-4.6.0-cp36-cp36m-win_amd64.whl (133kB)
    |████████████████████████████████| 143kB 6.4MB/s
Collecting pytz (from datetime)
  Downloading https://files.pythonhosted.org/packages/3d/73/fe30c2daaaa0713420d0382b16fbb761409f532c56bdcc514bf7b6262bb6
/pytz-2019.1-py2.py3-none-any.whl (510kB)
    |████████████████████████████████| 512kB 6.4MB/s
Requirement already satisfied: setuptools in c:\users\vlad95ta\appdata\local\programs\python\python36\lib\site-packages
(from zope.interface->datetime) (39.0.1)
Installing collected packages: zope.interface, pytz, datetime
Successfully installed datetime-4.3 pytz-2019.1 zope.interface-4.6.0
WARNING: You are using pip version 19.1.1, however version 19.2.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

Text Editor installation to modify the Script, if needed

Python is a “wordy” programming language, and it has many words that are reserved for internal use. A Text Editor that would help distinguish these words is crucial in development and especially Script modification. There are plenty of the Text Editors available, *Sublime Text* – is one of them.

1) Navigate to https://www.sublimetext.com/ and download the latest version of the Sublime Text editor for your Operating System. Install the Software.
2) Once done, you may open the Script file by right clicking on the file and choosing “Open with Sublime Text”. A respective Text Editor will appear.

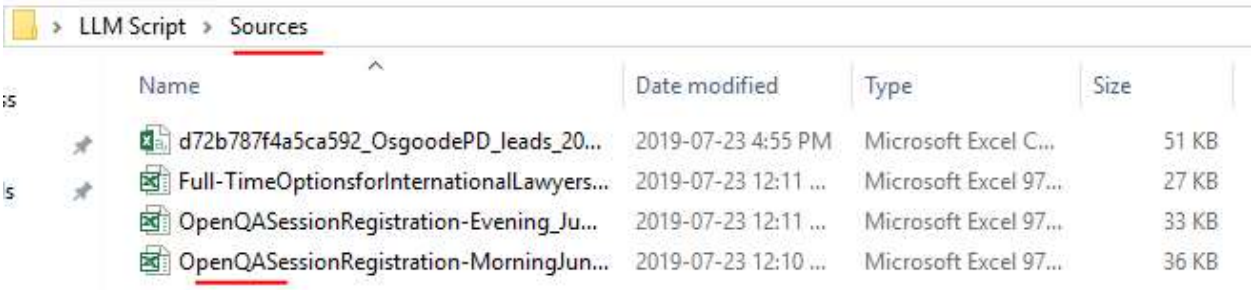


3) Now you should be ready to manipulate the Script manually.

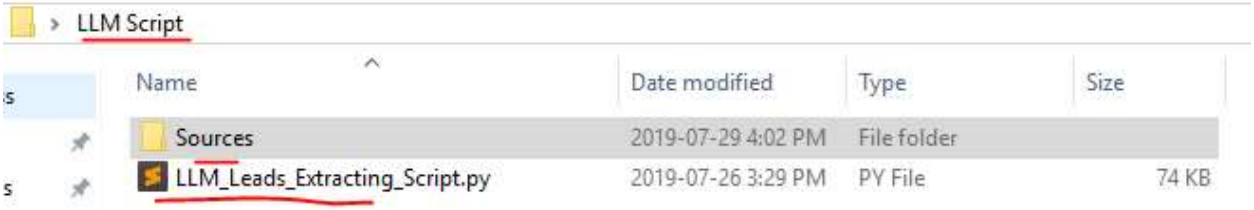
Appropriate Directory Structure for the Script to Run

Now that you have installed Python and all the modules required by the Script, you should set up the directory structure where you can place all the relevant files.

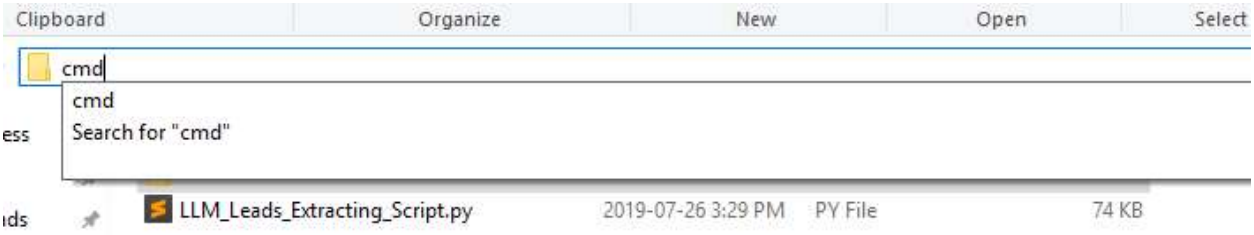
1) Create a folder Sources**(Exact naming is required!),** and put your Raw Source Documents there



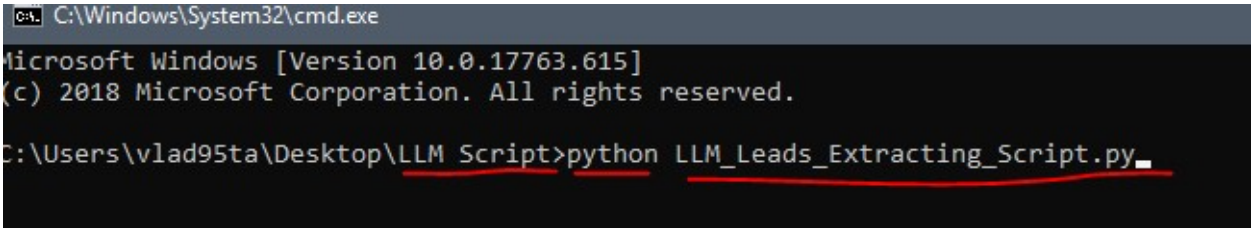
2) Place Sources folder in the same folder as the Script, for example *LLM Script:*



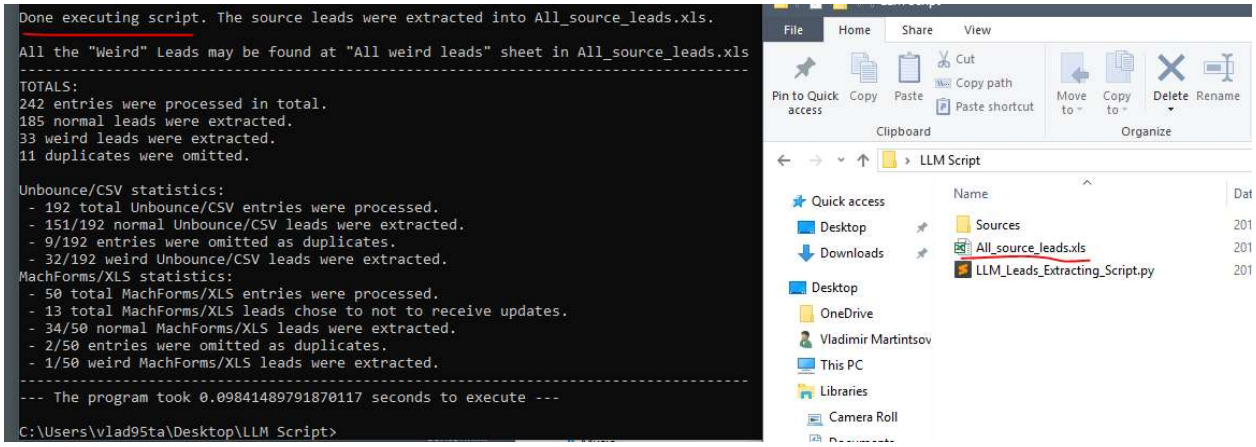
3) Open the Command Prompt in the *LLM Script Directory* by typing "cmd" in address bar of the File Explorer and hitting Enter button:



4) Type in Command Promt the following command: python LLM_Leads_Extracting_Script.py as follows and hit Enter button:



5) As you see the Script executing, wait until it is done. The Clean Source Document will appear in the same directory as the Script:



6) The file All_source_leads.xls is ready to be imported into the Database!