Donor Report Scraping System

Data Processing Subsystem: Template Generation

Version 0.9

Document Number: TG-001

Project Team Members: Joseph Bieselin JAB975

Errol Elbasan EE634

Robert Ryszewski RR1854

Shunman Tse ST1637

**1 Where to Begin**

Included with this document there is a sample template file called **sample\_readTemplate.py**. The programming language used to create the template files is Python (version 2.7).

The sample template file, **sample\_readTemplate.py**, is a functional template that was designed to work with National Fish and Wildlife Foundation’s (NFWF) 2014 and 2015 reports.

The sample template can be used as the starting point for a new template or as a guide for creating new template structures in the future. As long as the output of the template is in a format that the rest of the system can work with, there will not be any issues with changing the template.

**2. Creating a New Template (based on the sample template)**

To create a new template, modifications can be made to the file, **sample\_readTemplate.py**.

First, you need to open the sample template file in a text editor or a Python IDE. If you have Python installed on your system, you can use Python’s IDLE or the Integrated Development Environment. Otherwise, use the text editor that you prefer.

**3. Lists**

Upon opening the file, the first few lines of code before the first function should by default be:

# Global Lists

#>>> The following lines have to be changed to fit the document that you are working with <<<

# Ignore list containing keywords for unwanted lines

ignore\_list = ["restore wildlife", "2015", "46", "47", "Anonymous"]

# List of specific lines that are merged with the previous upon detection

merge\_list = ["Family Foundation\n", "Insurance Company\n", "Owners Master Association, Inc.\n"]

# List of keywords to look for in the start and end of lines that would signify merging

se\_merge\_list = ['of', 'the', 'for', 'and']

In the code sample above, you can see three different lists:

* ignore\_list
* merge\_list
* se\_merge\_list

**ignore\_list**

The ignore\_list contains keywords to identify lines that we do not want to appear in the output file. If the ignore\_list was empty (i.e. ignore\_list = [ ]) and the nonprofit file being read was the NFWF 2015 report, then we would get these lines in the output file:

47

46

2015 DONORS

In 2015, generous gifts from these donors furthered NFWF's actions to protect and restore wildlife

If the above lines are in the output file, they will become donors in the current category in the DRSS database. While it is possible for the database to be cleaned up, there could be far more than 4 rogue entries in a single nonprofit report. It’s better in the long run for the template to filter out as many of the bad entries as possible. By using the ignore\_list defined in sample file, we can get rid of the undesirable entries in the output file.

**merge\_list**

Before going into merge\_list, it’s important to note that the template works by reading one line ahead and considering whether there is a need to combine the two lines (i.e., the current line that the system is working with and the following line). This is because nonprofit reports may not write out the entire donor name in a single line.

If the merge\_list was empty, we would get these lines in the NFWF 2014 report:

Paul and Patricia Kuehner

Family Foundation

Those two lines are supposed to be a single donor entry. By adding ‘Family Foundation\n’ into the merge\_list, the template will now be able to combine the two above lines into a single entry. The output file will now have the single entry:

Paul and Patricia Kuehner Family Foundation

The merge\_list can be created by going through the output file and looking for cases that needs to be merged or it can be created by looking at the nonprofit report for the donor names that are split over multiple lines.

**se\_merge\_list**

se\_merge\_list has a functionality similar to merge\_list in that they both help identify lines that should be combined. The difference between the two is that merge\_list contains exact entries to combine, while the entries in se\_merge\_list are general things to look for at the beginning or end of a line to see if there is a need for merging.

If se\_merge\_list was empty (i.e., contains no entries), then in the 2014 NFWF report, we would get this in the output file:

Mitsubishi Corporation Foundation

for the Americas

By adding the entry ‘for’ to se\_merge\_list and using the template on the 2014 report, we get:

Mitsubishi Corporation Foundation for the Americas

Entries in se\_merge\_list are checked for at the end of the current line and at the beginning of the next line to identify merge cases.

**4. Starting and Ending Points for Donor Lists**

Annual reports come with large amounts of content that are not relevant to the DRSS. The system only needs the donors and nothing else. In the template, we must make modifications to identify where donor lists start and when they end.

In the template file, we see:

# Find the starting point of the list of donors in the document

# >>> The following lines have to be changed to fit the document that you are working with <<<

if ('$500,000+' in line):

In the NFWF reports, the first category of donors is $500,000 and so that will be our starting point. In the above code statement, it’s saying that if we see $500,000 in the current line, then the program is staring at the starting point of the donor list.

Further down in the template file, we see:

# Check for the ending condition

# >>> The following line has to be changed to fit the document that you are working with <<<

if ("2 0 1 4 d o n o r s" in line) or ("A scarlet tanager sings in" in line):

wfile.close()

return

In the 2014 NFWF file, the ending condition was reaching a line that said 2 0 1 4 d o n o r s.

In the 2015 NFWF file, the ending condition was reaching a line that said A scarlet tanager sings in.

**5. Issue**

A template cannot be completed by someone who knows nothing about Python code. When the system encounters a new category such as:

$500,000+

The system must convert that line of text into 500000.

Otherwise, the database will not know how much a donor gave to the nonprofit.