Email Management

2020-10-28 - created by Ethan Hunter

# Intro

This script offers tools for working with lists of emails and names in Excel files. To run the command, copy the files you wish to manipulate, the files “main.py” and “functs.py”, and the config.txt file into a folder (hereafter called your Workspace.). Double click the file main.py to execute the macro.

I use bold to highlight the instructions given to each command, and italics to highlight the actual commands. You should not use italics in the actual commands.

Contents

[Intro 1](#_Toc54780600)

[Writing Macros 1](#_Toc54780601)

[Intro 1](#_Toc54780602)

[The Commands 1](#_Toc54780603)

[new\_table 1](#_Toc54780604)

[delete 1](#_Toc54780605)

[open 1](#_Toc54780606)

[save\_as 1](#_Toc54780607)

[copy\_over 1](#_Toc54780608)

[append\_to 2](#_Toc54780609)

[remove\_duplicates 2](#_Toc54780610)

[keep\_only\_duplicates 2](#_Toc54780611)

[remove\_rows\_matching 2](#_Toc54780612)

[The Actual Macro 2](#_Toc54780613)

[Examples 2](#_Toc54780614)

[Appending Two Files 2](#_Toc54780615)

[Removing Duplicates in a File 2](#_Toc54780616)

# Writing Macros

## Intro

Macros are stored in the config.txt folder. The macro language operates on opening excel files as tables, modifying them, and then saving them as new excel files.

## The Commands

### new\_table

New table creates a new, empty, table in memory. The syntax is “*nameOfTable new\_table*”. For instance, the command “*allMyStuff new\_table*” would create a new table named **allMyStuff**.

### delete

The delete command has two usages. First, you can use it to delete a table in memory. The syntax for this is “*nameOfTable delete.*” For instance, to remove our old table “**allMyStuff**”, we would type “*allMyStuff delete*”.

It can also delete an excel file. This one is slightly more dangerous in terms of your data, so make sure to back up all excel sheets in your workspace using if you plan to make use of this command. The syntax is *“excelFileToDelete.xlsx delete*”. For instance, “*allMyData.xlsx delete*” would delete the excel spreadsheet titled “**allMyData.xlsx**”.

### open

The open command creates a new table in memory if one doesn’t already exist with the name you give it, and then opens an excel spreadsheet into it. The syntax is “*nameOfTable open nameOfSpreadsheet.xlsx”*. For instance, entering *“allMyStuff open allMyData.xlsx”*  
would create a new table called “**allMyStuff”** (if it doesn’t exist) and put the data from “**allMyData.xlsx**” into it.

### save\_as

The save\_as command saves a table into your workspace. The syntax is “*nameOfTableToSave save\_as excelSpreadsheet.xlsx”*. For instance, to save the table “**allMyStuff**” as **“newData.xlsx**”, you would type “*allMyStuff save\_as newData.xlsx”*.

### copy\_over

The copy\_over command copies the data from one table and pastes it over another table. The syntax for this is *“tableToCopyFrom copy\_over tableToCopyTo*”. For instance, to copy the data from “**allMyStuff**” to “**copyOfStuff**”, you’d type “*allMyStuff copy\_over copyOfStuff*”.

Only the table the data is copied from must exist, the table the data is pasted to will be created if it doesn’t.

### append\_to

This command copies data from one table and pastes it at the end of another. The syntax is *“fileToCopyFrom append\_to fileToAppendTo*”. For instance, to append “**allMyStuff**” to “**copyOfStuff**”, you’d type “*allMyStuff append\_to copyOfStuff*”. This command only modifies the table being pasted to.

### remove\_duplicates

This command takes a table and removes all the duplicates from it. The syntax is “*fileToRemoveDuplicates remove\_duplicates*”. For instance, to remove all duplicates from “**allMyStuff**”, you’d type “*allMyStuff remove\_duplicates*”.

### keep\_only\_duplicates

This command does the opposite of remove\_duplicates. Rather than removing any duplicate rows, it removes all rows that aren’t duplicated, and then keeps only one copy of each duplicate line. For instance, to keep only the duplicates, you’d type “*allMyStuff keep\_only\_duplicates*”.

### remove\_rows\_matching

This command takes two tables. It goes through the first table line by line, and if the second table has the same row, removes it from the first table. For example, to remove any rows that are in both “**allMyStuff”** and “**copyOfStuff**” from “**allMyStuff**”, you’d type “*allMyStuff remove\_rows\_matching copyOfStuff*”.

## The Actual Macro

Type the commands you want the script to do in order, from top to bottom, in config.txt.

## Examples

### Appending Two Files

mainFile open emailsForTest.xlsx  
toAppend open testDuplicates.xlsx  
toAppend append\_to mainFile  
mainFile save\_as appendedFile.xlsx

### Removing Duplicates in a File

mainFile open emailsForTest.xlsx  
mainFile remove\_duplicates  
mainFile save\_as appendedFile.xlsx