

CountTiffPagVer1.py Program Manual

About

CountTiffPagVer1.py is a program developed to assist in the process of verification of loaded information (in this case invoice data from truck hauls) into an ERP. The idea is to make sure that all pages (freight bill <FB> and bill of loading <BOL> sheets) are loaded into the system and none are left out. To accomplish that, the program goes through all the files inside a given folder and selects for processing only TIFF image files. Each selected file gets scanned to determine the amount of pages it contains. Finally a report is made with the amount of pages in each file and miscellaneous information pertinent to the processing of those files.

Program Licensing

CountTiffPagVer1.py is licensed under LGPL version 2.1 and all limitations to that license apply to it.

Program Requirements

This program has the following minimal requirements in order to correctly run in a computer system:

1. A PC or laptop with at least the following characteristics: i5 CPU, 4 GB RAM, 250 GB HDD / SSD.
2. OS: Windows 10 pro / Linux 6.x (any distro).
3. Python 3.10 installed. Important to check and upgrade to this version if necessary.
4. Python PyMuPDF library version 1.23.3 (needs to be installed separately).
5. A terminal application. PowerShell ISE in Windows or any Terminal shell in Linux.
6. A text editor to open and modify *tiffFolderPath.json*.

Program Design

The following is a representation of the components and relations of this program:

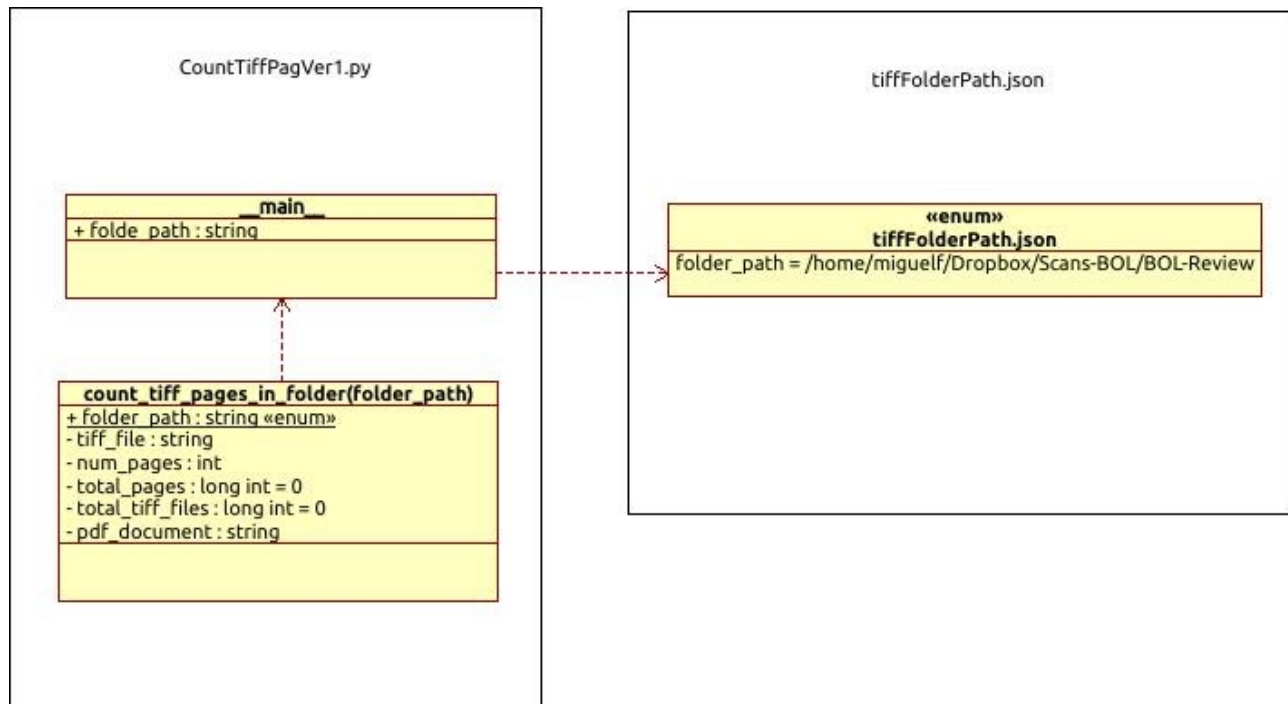


fig. 1

The program consists of 2 separate entities. The first entity is the program code itself (file named *CountTiffPagVer1.py*) and the second entity is a configuration file written with Json syntax (file named *tiffFolderPath.json*) that contains the path to the folder where the TIFF files reside. At program start it will first look for this path to configure the search for the TIFF files and assign it to the variable *folder_path* as a string of characters, so it has to be setup properly with the correct files path before starting the program. The following is a diagram of the program flow:

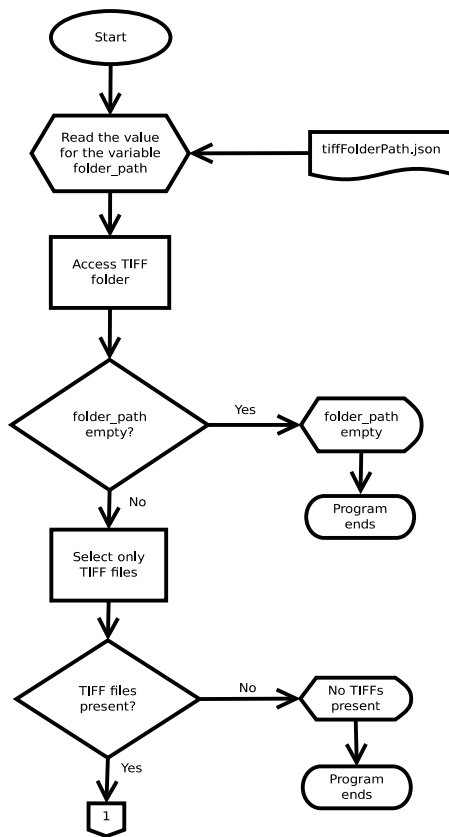


Fig. 2

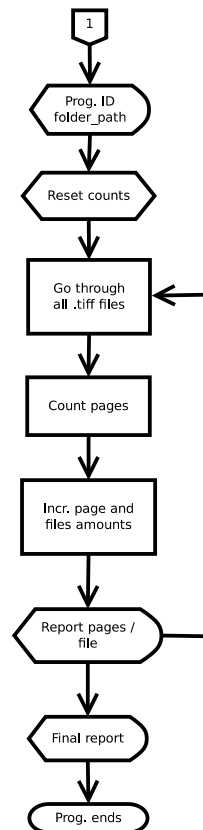


Fig. 3

The output of the program will be redirected as a text file that has to be named during the program's call for running, so its final name is decided by the user, but it's recommended to end this filename with the extension .txt to reduce any problems with the Windows OS file handling policy. Here is an example of such a file in figure 4.

```

CountPagEx.txt
~/Documents/Burningham/Proyectos/BT-scripts/Ti...
Open  Save  [Menu]  [Close]

1 TIFF pages count per .tiff file
2 Folder reviewed: /home/miguelf/Dropbox/Scans-BOL/BOL-Review/Agosto
3
4 File: 644635_BOL.tif has 2 pages.
5 File: 742584_BOL.tif has 3 pages.
6 File: 724981_BOL.tif has 1 pages.
7 File: 699685_BOL.tif has 2 pages.
8 File: M01668_BOL.tif has 1 pages.
9 File: 722546_BOL.tif has 2 pages.
10 File: 748283_BOL.tif has 2 pages.
11 File: 747388_BOL.tif has 5 pages.
12 File: 745903_BOL.tif has 10 pages.
13 File: 726793_BOL.tif has 2 pages.
14 File: 644630_BOL.tif has 2 pages.
15 File: 747099_BOL.tif has 5 pages.
16 File: 722823_BOL.tif has 1 pages.
17 ...
18 ...
19 ...
20 Process Report:
21 Total TIFF files processed: 2458
22 Total Pages in Folder: 6522
  
```

Fig. 4

A print out of the program's code will not be included in this manual because it's accessible directly using any editor or python programming IDE of choice. The internal structure of the file *tiffFolderPath.json* is as follows:

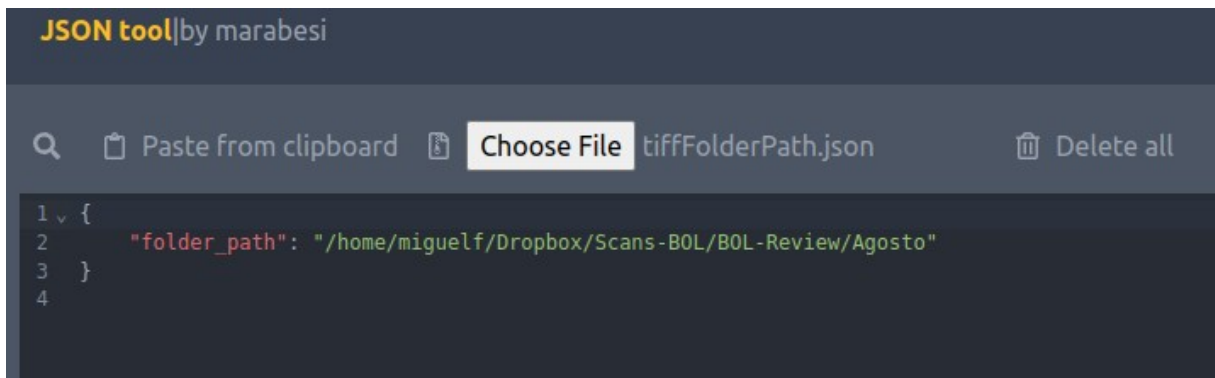


Fig. 5

Program Execution

Before program execution it is necessary to specify the folder path where the TIFF files reside. That is done by opening *tiffFolderPath.json* with any text editor and replacing the *folder_path* variable value inside quotes (green text in figure 5) with the actual folder path that will be analyzed. Make sure to save the file with the new folder path before running the program. The program handles seamlessly the syntax of the folder path in Windows or in Linux, so you only need to copy the path into *tiffFolderPath.json* with whatever format it comes.

Running the program requires to redirect the output of the program to a file in order to have a hard copy of the results of the program execution. Otherwise we'll only have a screen output of the program. In order to implement the output redirection we'll need to type the run command in one of the following ways depending on the OS being used:

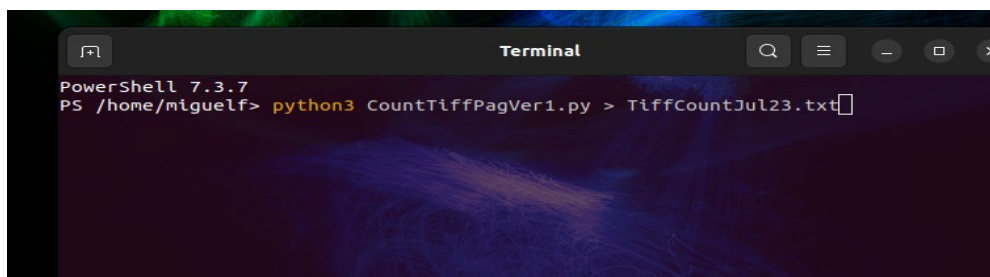


Fig. 6 (Windows PowerShell ISE)

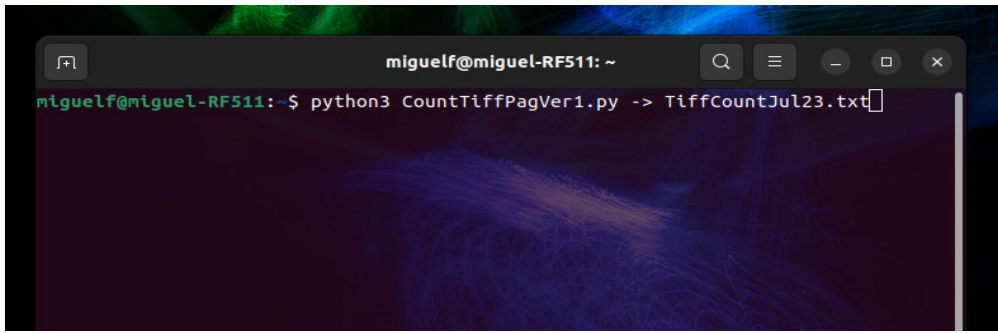


Fig. 7 (Linux Terminal Shell)

Note that it is at this stage that the output text file's name is selected.