Thank you

I hope that you are **completely satisfied**

rfv.io

Thank you Brian for using my services. I truly hope that I have created a solution to your problem.

Your Specification

Need to have developed a command line utility in Python for OS X.

The utility will take as input (source) as list of files and/or folders, or a file containing same, and search another (target) list of file and/or folders, or a file containing same, to see if the source file(s) are found anywhere in the target location(s).

The determination of a match includes name, create timestamp, modification timestamp, file size, file hash, and file attributes. Source and/or target files may begin with a period "." meaning that they are normally invisible in the Finder.

The output is a list with tab separated fields as follows: source file location, target file location. A source file may be found in more than one target file location. If the source file is not found in any target file location, then the default text of "not found" should be used in place of the target file location.

An optional argument to the utility will move source file(s) to the Finder Trash (not delete). Source file(s) and target location(s) may be on remotely mounted drives, so the optional "move to trash" feature will need to determine if this is the case, and, if so, "move to trash" on the remote volume and not "move to trash" on the target volume which would force a cross-volume move.

Sample arguments:

```
utilityname -i filename1 filename2 -o location1 location2
utilityname -if filelist_i -o location1 location2
utilityname -i filename1 -of filelist_o
utilityname -if filelist_i -of filelist_o
utilityname -if filelist_i -of filelist_o -trash
```

-i indicates that one or more source filenames and/or directories follow

-if indicates that the filename that follows contains a list of one or more source filenames and/or directories, and should be used as if they were provided via -i

-o indicates that one or more target filenames and/or directories follow

-of indicates that the filename that follows contains a list of one or more target filenames and/or directories, and should be used as if they were provided via -o

-trash indicates that matching source filename and/or directories should be moved to Finder Trash

-noempty indicates that directories left empty as a result of -trash (ignoring the .DS_Store file) should also be moved to the Finder Trash

Execution statistics should also be provided: Time from start to finish, total number of source files checked, total number of target locations checked, total number of matches, total number of non-matches, percentage of matches to total number of source files checked.

Platform is running OS X 10.11.5 El Capitan.

Results

Before you begin you well need to install one python utility module used for moving files to Trash and to make sure that it stays on the correct volume.

Please install “Send2Trash” with the following command: **pip install Send2Trash**

Then I have also added the following command line parameters so that you may specify which parameters to switch off. All search parameters are on by default, but a delete will only occur if all parameters are matched (as per you last message). For example, only if name and modified timestamp and creation timestamp and hash and size all match exactly will the file be deleted. To switch off a parameter here are the options:

-noname

-nosize

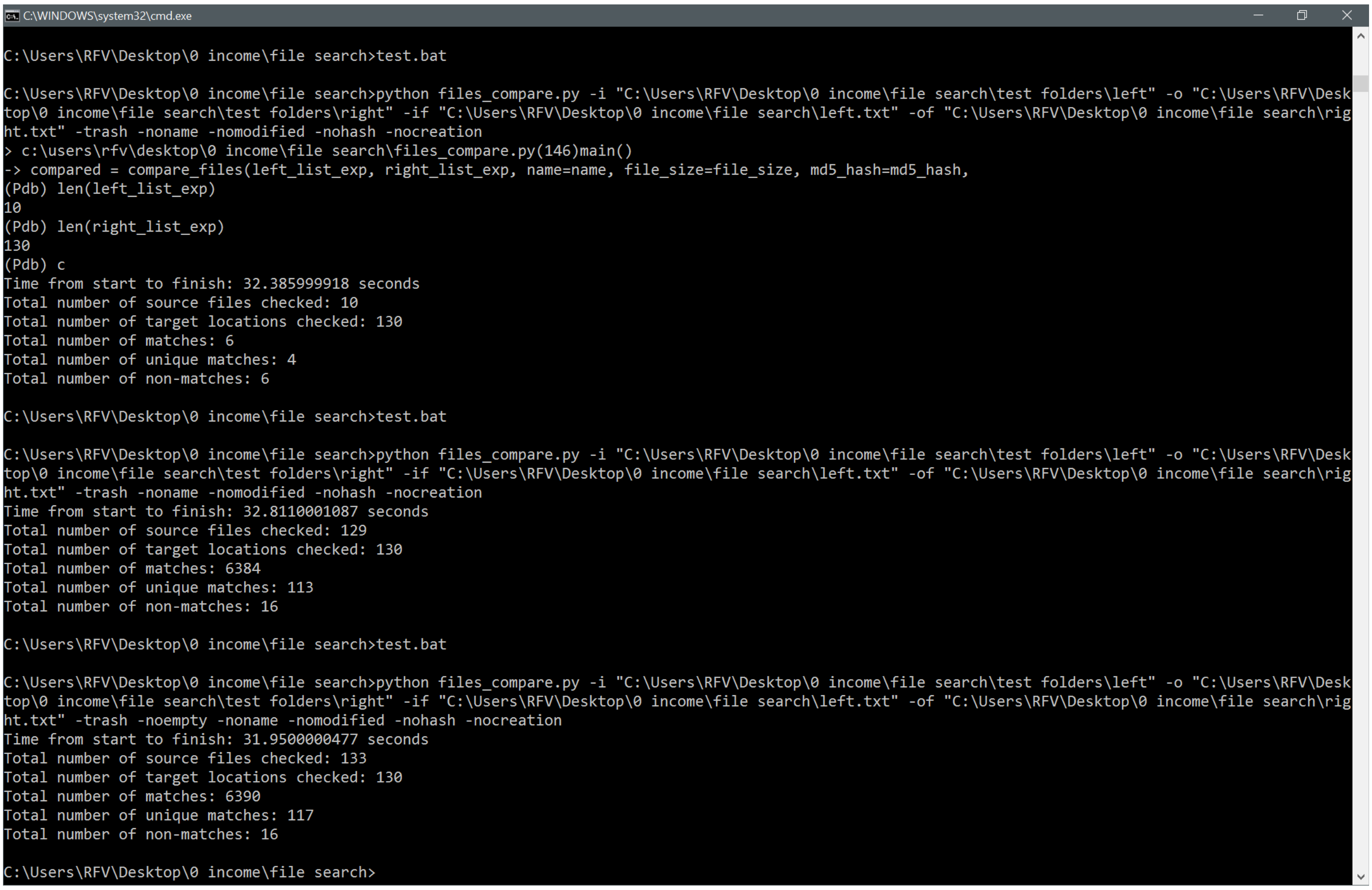
-nohash

-nocreation

-nomodified

I have created a csv output for the time being, if you need another tabed output just let me know. There are 4 columns [input files, output files, matched parameters, all possible parameters found (True if all parameters where found for that particular input/output files pair).

On my system (windows) I have created some test cases with a bat file and they run correctly.



```
C:\Users\RFV\Desktop\0 income\file search>test.bat

C:\Users\RFV\Desktop\0 income\file search>python files_compare.py -i "C:\Users\RFV\Desktop\0 income\file search\test folders\left" -o "C:\Users\RFV\Desk
top\0 income\file search\test folders\right" -if "C:\Users\RFV\Desktop\0 income\file search\left.txt" -of "C:\Users\RFV\Desktop\0 income\file search\rig
ht.txt" -trash -noname -nomodified -nohash -nocreation
> c:\users\rfv\desktop\0 income\file search\files_compare.py(146)main()
-> compared = compare_files(left_list_exp, right_list_exp, name=name, file_size=file_size, md5_hash=md5_hash,
(Pdb) len(left_list_exp)
10
(Pdb) len(right_list_exp)
130
(Pdb) c
Time from start to finish: 32.385999918 seconds
Total number of source files checked: 10
Total number of target locations checked: 130
Total number of matches: 6
Total number of unique matches: 4
Total number of non-matches: 6

C:\Users\RFV\Desktop\0 income\file search>test.bat

C:\Users\RFV\Desktop\0 income\file search>python files_compare.py -i "C:\Users\RFV\Desktop\0 income\file search\test folders\left" -o "C:\Users\RFV\Desk
top\0 income\file search\test folders\right" -if "C:\Users\RFV\Desktop\0 income\file search\left.txt" -of "C:\Users\RFV\Desktop\0 income\file search\rig
ht.txt" -trash -noname -nomodified -nohash -nocreation
Time from start to finish: 32.8110001087 seconds
Total number of source files checked: 129
Total number of target locations checked: 130
Total number of matches: 6384
Total number of unique matches: 113
Total number of non-matches: 16

C:\Users\RFV\Desktop\0 income\file search>test.bat

C:\Users\RFV\Desktop\0 income\file search>python files_compare.py -i "C:\Users\RFV\Desktop\0 income\file search\test folders\left" -o "C:\Users\RFV\Desk
top\0 income\file search\test folders\right" -if "C:\Users\RFV\Desktop\0 income\file search\left.txt" -of "C:\Users\RFV\Desktop\0 income\file search\rig
ht.txt" -trash -noempty -noname -nomodified -nohash -nocreation
Time from start to finish: 31.9500000477 seconds
Total number of source files checked: 133
Total number of target locations checked: 130
Total number of matches: 6390
Total number of unique matches: 117
Total number of non-matches: 16

C:\Users\RFV\Desktop\0 income\file search>
```

The program takes all the input files + folders + an input files list and concatenates it all into one list of files, then it gets each files attributes. This is done for both input (left) and output (right). Then it compares all on the left to the right and will return the matched properties (and not properties that have been turned of at command prompt)

After all it displays the following:

"Time from start to finish: XXX seconds”

"Total number of source files checked: XXX”

"Total number of target locations checked: XXX”

"Total number of matches: XXX”

“Total number of unique matches: XXX”

"Total number of non-matches: XXX”

Final word

If all is well then I would like to make a suggestion towards creating a GUI for you, if you feel that this may be beneficial to you – just let me know, and there is a very basic mockup included in the attached files.

If you need any assistance in the future or follow up work, you are more then welcome to contact me.

Also, please remember to add your review about my work – this will help me to help others because good reviews help employers to find quality workers.

Take care.