

Extending Testing Week 1 Challenge (Phase 2)

Contents:

1. Basic plan
 2. Program Understanding (from tester's perspective)
 3. Logged expected errors
 4. Diagram
 5. Initial States assessment and logging of possible file routes
 1. expected test routes (prior to testing)
 6. Basic Exploratory Test notes
 1. Note - bugs may be detailed here, but see bottom for bug report
 7. Test data generation (python) overview
 1. please see actual files for more info
 8. Test cases using generated data
 9. Bug Reports (and classification rules)
 10. Feedback Conclusion
-

Basic plan

1. exploratory test of the program
2. and 3. -- Generate data to test the program with
 1. files with filenames
 1. containing names/addresses
 2. distributed into each directory
 1. ALL into originals
 1. some of which into drop list
 2. some of which also into allowlist
 2. smaller duplicate set to be in updates
3. use the en_UK locale
 1. 'You should be using Faker-generated surnames as filenames (also in `droplist` / `allowlist` content) and Faker-generated addresses as the

content of the documents in **originals** and **updates**.'

3. Run another test session
4. create a feedback document

Your coaches would like to see you submit:

- Details of your test planning, especially any techniques from the course that you thought could be valuable to use here.
- At least two bug reports on bugs you've identified; carefully consider what those bugs need to contain from the perspective of developers *and* stakeholders.

Program Understanding:

Lists

- **Originals** - directory of docs
- **Droplist** - originals to get rid of
- **updates** - use instead of original
- **allowlist** - those from originals, that are not in the other lists
- **Finals** - this is the final working list
 - nothing from **dropped**
 - nothing from original that is also in updates
 - will contain
 - originals in no other list - and therefore in allowlist
 - updates (without their originals)

So, if functioning correctly the final list should just be a group of:

allowlist + updates

but from the eg we have to assume that the allow list could have docs that are 'already updated' - so it will be more like

updates + (allowlist-updates)

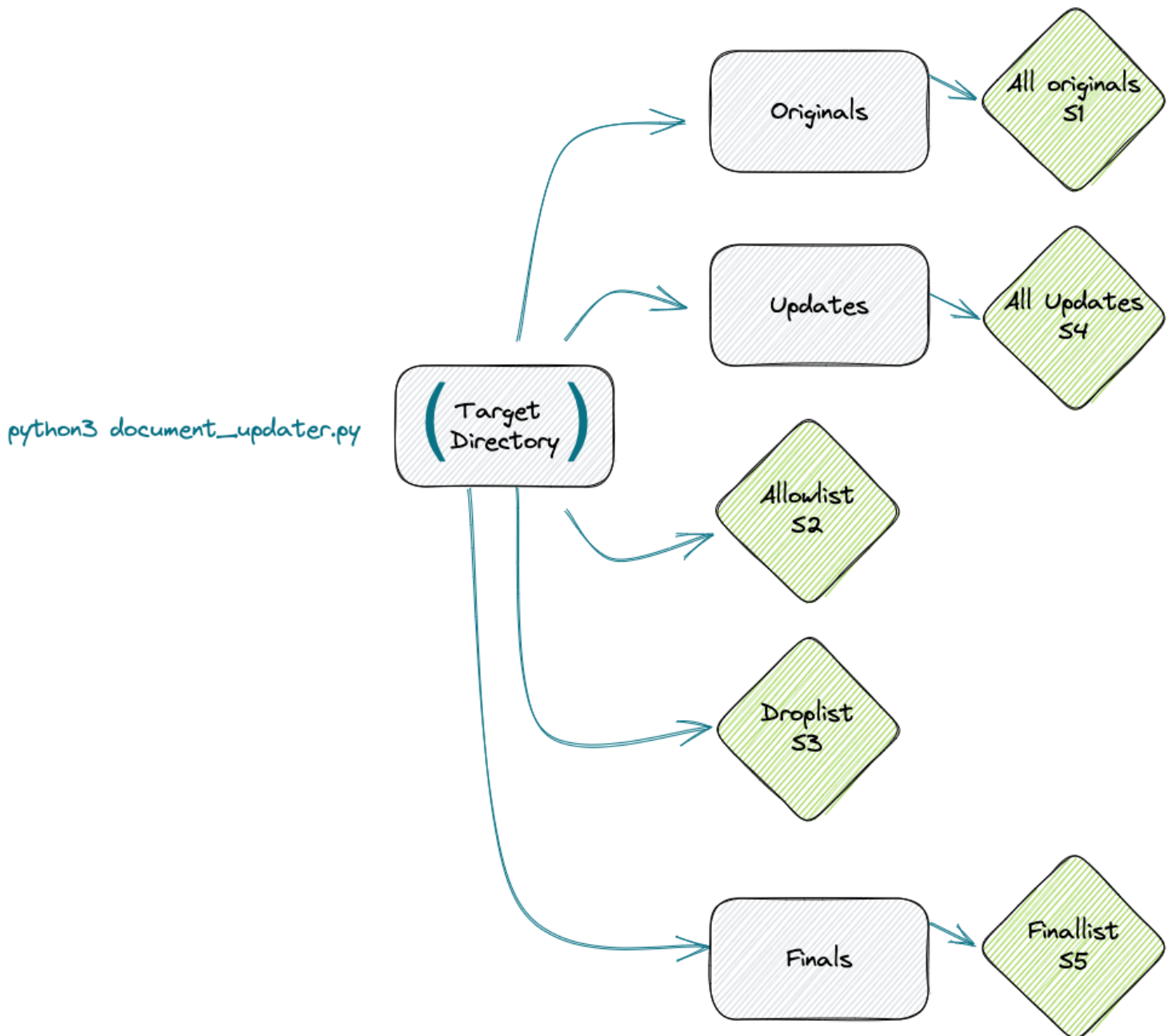
Expected errors:

- Program run without specifying a directory on which to operate
- "originals" folder not found

- "updates" folder not found
- Missing both `droplist` and `allowlist`
- Both `droplist` and `allowlist` are specified
- Document(s) don't contain correctly formatted addresses

1. it may be worth testing that these do throw errors

Diagram



The Coach example would be present in the ORIGINALS, the ALLOWLIST and the UPDATES
 Not present in the DROPLIST
 as a result the program should move it to the FINALS

States

State code	'Location'
S1	All originals
S2	Allow list
S3	Droplist
S4	All Updates
S5	Finallist

*Note - it appears from the way the program is set up either S3 **OR** S2 can be present, not both*

Possible file routes and test cases

Type Code	File type	present in	expected destination
T1	Valid - Original	S1, S2	S5
T2	Invalid - Original (dropped/not allowed)	S1, S3	S3
T3	Valid - Updated Original	S1, S2, S4	S5

*Note - it appears from the way the program is set up either S3 **OR** S2 can be present, not both*

Basic Exploratory Test

Running the coach file and the program as is

- Coach is the file name and surname
- It appears to be a T3 file - in the original list, the allow list and the update list
- The program runs with no error and the new folder 'finals' contains the file 'Coach' with the 'updated' version
 - *Note at this point will adjust diagram to represent a folder containing files rather than a list file*

Running with one of each type of file (Drop list present)

- Coach is T3
- Added file 'Billy' (T1)
 - Miss Milly Billy

- 123 Billy street
- Awesometown
- HR17 7PL
- Added file 'Smith' (T3)
 - Mr John Smith
 - 10 fake street
 - definitely not real town
 - EX17 9LJ
- the program correctly sorted Billy and Coach into **Finals**
- Smith was dropped and therefore not added

Running with one of each type of file (Allowlist present)

- Added file 'Billy'
 - Miss Milly Billy
 - 123 Billy street
 - Awesometown
 - HR17 7PL
- Added file 'Smith'
 - Mr John Smith
 - 10 fake street
 - definitely not real town
 - EX17 9LJ
- the program correctly sorted Coach into **Finals**
- But there was no Billy sorted

Bug B1

With the program running with the 'allowlist' stating Coach and Billy in this case, only Coach was added to finals

Would expect - both to have been added to Finals

Steps to reproduce (running with allowlist):

1. download original program in current state
2. add Billy as above to a file in originals
3. add Smith as above to file in originals
4. add billy to the allowlist
5. run the program
6. notice that only coach is added

Further, when completing the steps as above but also adding Smith to the allow list the result is the same

However, if 'Coach' is then removed from the allow list - other two are added, Billy and Smith

Speculation - when files are in update list - they 'block' the allow list - and only 'updates' are added to finals

Additional update to Billy

- Added file 'Billy'
 - Miss Milly Billy
 - 123 Billy street
 - Awesometown
 - HR17 7PL
- Added file 'Smith'
 - Mr John Smith
 - 10 fake street
 - definitely not real town
 - EX17 9LJ
- Further an updated version of Billy is added

- amending the street to 123456 Billy Street
- all three are in the allow list and the originals folder
- would expect that all three are added to finals, but with 2 updated versions, coach and billy

Bug 1.1

As with bug B1 - only those in the updates folder are added to finals, not 'Smith'. This appears to confirm the speculation (**Speculation** - when files are in update list - they 'block' the allow list - and only 'updates' are added to finals)

Droplist - Coach, Smith, Billy with updated versions of Coach and Billy

As previously:

- Added file 'Billy'
 - Miss Milly Billy
 - 123 Billy street
 - Awesometown
 - HR17 7PL
- Added file 'Smith'
 - Mr John Smith
 - 10 fake street
 - definitely not real town
 - EX17 9LJ
- Further an updated version of Billy is added
 - amending the street to 123456 Billy Street

Ran the program with just Billy in the droplist:

- Runs incorrectly, Billy and Coach are added
- See bug 1 and 1.1

Ran the program with just Coach in the droplist:

- Runs incorrectly, Billy and Coach are added
- See bug 1 and 1.1

Ran the program with just Smith in the droplist:

- Runs incorrectly

Bug B2

When running with droplist and specifying to drop a file, whilst updating 2 other files, rather than just adding the 2 updated files to the Finals - Finals remains empty

The following error is given:

```
→ 03_resources git:(main) x python3 document_updater.py
target_directory
Traceback (most recent call last):
  File "/Users/adamlittle/Documents/extending-testing/extending-
testing/phase2/03_resources/document_updater.py", line 125, in
<module>
    os.makedirs("blends")
  File "<frozen os>", line 225, in makedirs
FileExistsError: [Errno 17] File exists: 'blends'
```

and a folder called 'blends' is created - this contains just the updated version of 'Billy'

Steps to reproduce (running with droplist):

1. download original program in current state
2. add Billy as above to a file in originals
3. Add the update to Billy in updates
4. add Smith as above to file in originals
5. add Smith to the droplist
6. run the program
7. Notice that
 1. error is given
 2. blends folder is made
 1. containing just the updated 'Billy'

Both **droplist** and **allowlist** are specified

- Correctly throws an error


```
ERROR: both droplist and allowlist found in target directory
```

Missing both **droplist** and **allowlist**

- Correctly throws an error

```
ERROR: no droplist nor allowlist found in target directory
```

Program run without specifying a directory on which to operate

- Correctly throws an error

```
ERROR: must supply a target directory as an argument
```

"originals" folder not found

- Correctly throws an error

```
ERROR: originals folder not found in target directory
```

"updates" folder not found

- Correctly throws an error

```
ERROR: updates folder not found in target directory
```

Document(s) don't contain correctly formatted addresses

- Added file 'Billy'
 - Miss Milly Billy
 - this is definitely not how to format an address
- Added file 'Smith'
 - Mr John Smith
 - 😄 😄 😄 😄 😄 😄 😄 😄 😄 😄
- Correctly throws error

ERROR: document " Billy " doesn't contain an appropriately formatted address

Then amended Billy back to

123 Billy street

Awesometown

HR17 7PL

again:

ERROR: document " Smith " doesn't contain an appropriately formatted address

Test data generation overview:

extending-testing > phase2 > Challenge_address_maker.py > ...

```
1 import shutil
2 from faker import Faker
3 import os
4 fake_list = []
5
6 # This will create the directory structure for ORIGINALS
7 > def make_originals(): ...
14
15 # This will create the directory structure for UPDATES
16 > def make_updates(): ...
23
24 > def make_allow_list(): ...
29
30 # This will populate the Contents of the above directories with a DROPLIST (can adjust the
    numbers on lines inside to adjust the amount dropped, and updated)
31 > def faker_generator_drop_list(): ...
60
61 # This will populate the Contents of the above directories with an ALLOWLIST (can adjust the
    numbers on lines inside to adjust the amount allowed, and updated)
62 > def faker_generator_allow_list(): ...
91
92
93 # The two below just order the above functions to make the final commands at the bottom more
    concise
94 > def make_THE_WORLD_drop_list(): ...
98
99 > def make_THE_WORLD_allow_list(): ...
103
104 #####
105
106
107 # Comment out one of the below to determine which you want to populate with
108
109
110 # make_THE_WORLD_drop_list()
111 make_THE_WORLD_allow_list()
112 |
```

- Code was written in python, within vscode
- as above there are functions to create directories and populate them with data, you can either run with an allowlist OR with a drop list
- Adjustments can be fine-tuned within the 2 generator functions to determine how many files are made, how many of them are updated, and any many of them are dropped/allowed

Test cases using generated data

Titles will contain:

- Allowlist/ Droplist - this determines which function was run within 'Challenge_address_maker.py'
- how many files in originals
- how many files in allowed/dropped
- how many updated

Allowlist - 10 files, all allowed, 5 updated

- **Bug 1**
 - updates folder takes precedence over all else

Did not run as expected - only those in the update folder ended up in finals

Allowlist - 10 files, 3 allowed, 0 updated

- ran as expected - the 3 allowed ended up in finals

Droplist - 10 files, all dropped, 5 updated

- **Bug 1**
 - updates folder takes precedence over all else

Did not run as expected - only those in the update folder ended up in finals

Droplist - 10 files, 3 dropped, 0 updated

- ran as expected - the 7 not dropped ended in finals

Allowlist - 100 files, 1 allowed, 0 updated

- ran as expected - 1 in finals

Allowlist - 10 files, 3 allowed, 0 updated - But with one file of the same name created prior to running app

- unable to complete - based on file structures there can only be one file of one name

Speculation - only one entry of a single surname is currently 'accepted'. This will likely apply to both entries, but also updates

Allowlist - 1 files, 1 allowed, 1 updated (different details completely, including name)

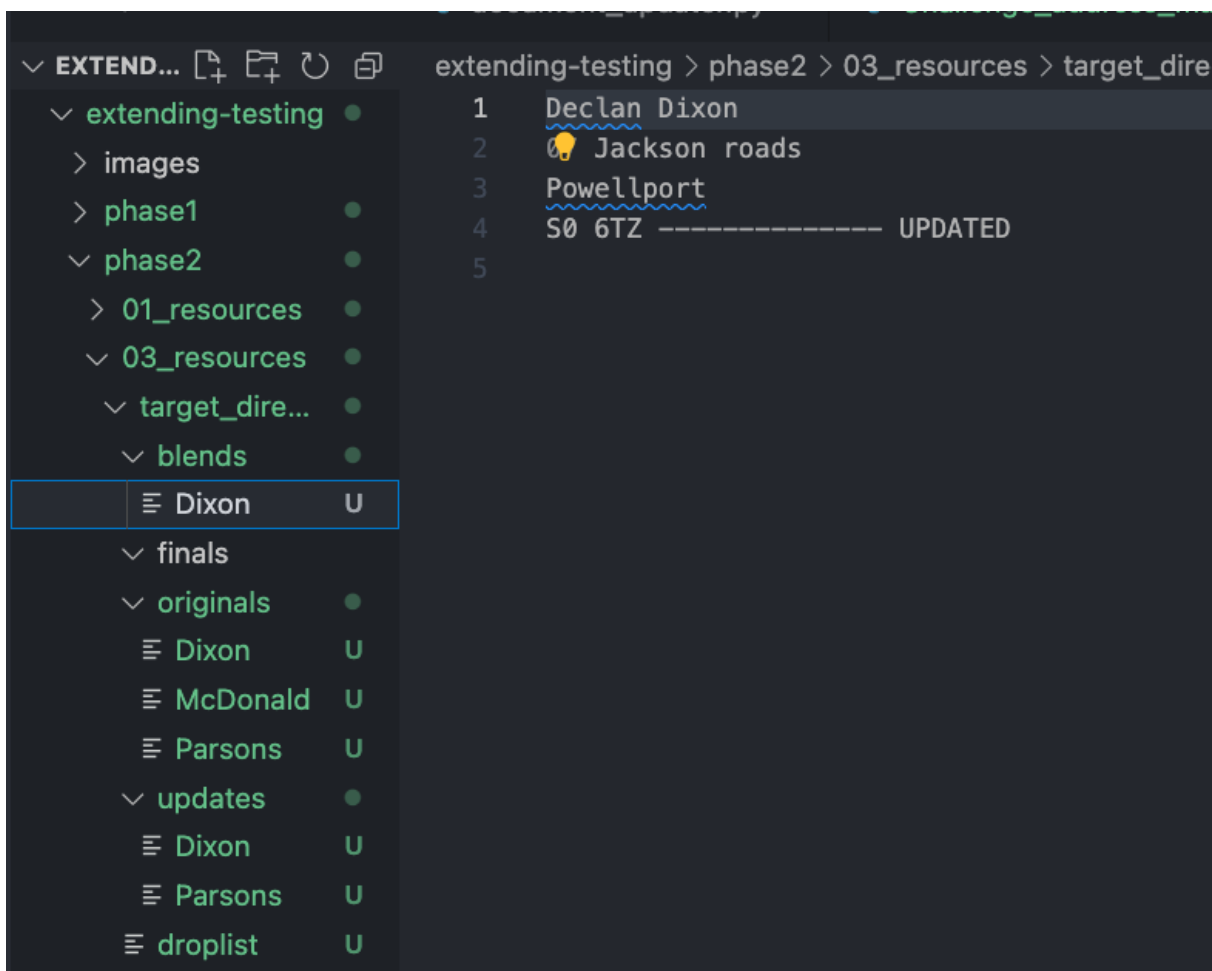
- Ran in a problematic way - confirming the above speculation

Bug B3

Droplist - 3 files, 1 dropped, 2 updated

- Bug 2

- Error given as in manual exploratory testing



In the interest of time I have called the testing to a stop here

Bug Reports

Classification of severity:

- High - This bug appears to break the functionality relative to the expected
- Medium - This bug impacts on the functioning, but the program still runs, this will impact the usefulness of the program

- Low - the program runs and functionality is either not impacted or is only impacted in a way that is perceived as minimal

As above - will provide further elaboration here:

Bug B1

Description

With the program running with the either allow or droplists, and with files specified in either - the program sorts only what is in the 'udaptes' folder into finals.

Would expect - to have the updates folder files, as well as any 'non dropped' files in the case of the droplist - or any allowed files in the case of allowlist.

This has a severe impact on the functionality of the program. It will likely lead a nearly completely wrong data set being output into finals. As a program, this breaks functionality. As a product it will not provide what is required and will lead to false datasets

Steps to reproduce:

1. download original program in current state
2. Run 'Challenge_address_maker.py' with the following details
 1. Droplist version - 10 files in originals, all dropped, 5 updated
3. run the program
4. notice that only files from the updated folder are added to finals

This was replicated with manual adding as well

Note: in the case of running this manually, with the allowlist, if the files from the update list were not in the allow list then this would 'dissallow' those files from being updated, and therefore the program would push the rest of the allow list to the finals folder, this happened when there was only the 1 file in updates, and that file was not present in the allow list.

Severity Rating

High - this bug will likely lead to false data sets, the functionality is severely impacted, and the usefulness of the product is questionable

Bug B2

Description

When running with Droplist and marking a file to be dropped, whilst having other files to update an error is given and unexpected folder 'blends' is made, containing just one of the 'updates' files (see below screens)

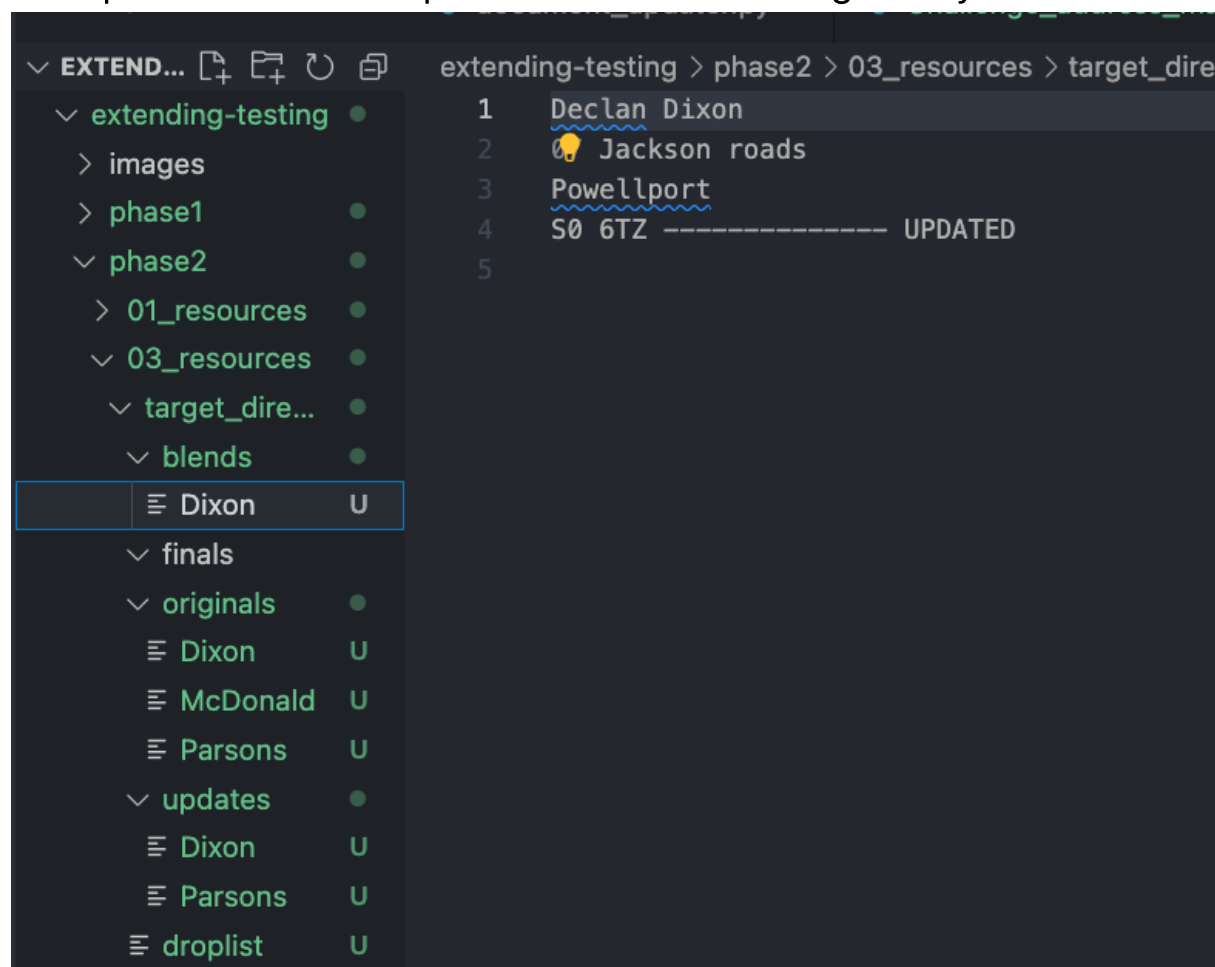
Would have expected that just the files 'not dropped' as well as updated ones to be added to finals

This is very problematic as this breaks the functionality of the program and will likely lead to false data sets.

The following error is given:

```
→ 03_resources git:(main) x python3 document_updater.py
target_directory
Traceback (most recent call last):
  File "/Users/adamlittle/Documents/extending-testing/extending-
testing/phase2/03_resources/document_updater.py", line 125, in
<module>
    os.makedirs("blends")
  File "<frozen os>", line 225, in makedirs
FileExistsError: [Errno 17] File exists: 'blends'
```

Example of the new unexpected folder and the single entry within:



Steps to reproduce (running with droplist):

1. download original program in current state
2. Run 'Challenge_address_maker.py' with the following details
 1. Droplist version - 3 files in originals, 1 dropped, the other 2 updated
 2. (can also be replicated manually)
3. run the program
4. Notice that
 1. error is given
 2. blends folder is made
 1. containing just one of the 'updated' files

Severity Rating

High - this bug will likely lead to false data sets, the functionality is severely impacted, and the usefulness of the product is questionable

Bug B3

Description

A file with the same name will override any previous file. Either in process or in updates.

Tester is aware that this is in part with the labeling of used test data. So long as data of unique names only, is used then this does not necessarily pose a problem.

However, even in a small sample size 2 people with the same last name is completely possible, the probability of which grows as the sample increases.

No error is thrown, but this being an issue or not, is not clear within the spec of the product.

Steps to reproduce

1. download original program in current state
2. try to enter both:
 1. John Smith
 1. 07 Jackson roads
 2. Powellport
 3. S0 6TZ
 2. Jane Smith
 1. Studio 27

2. Bradley fall
3. Lake Mandy
4. EN3 4HG

3. can be done with test generator, but is probably easier done manually
4. notice that both files cannot exist with the name 'Smith'
5. This same issue would occur when using the updates folder

Recommendation if/when implementing a feature to allow multiple people of the same last name, make sure this is correctly synced with the updates folder and functionality

Severity Rating

Medium - this is a perceivably severe issue, however the spec of the product did not comment on whether this is within scope.

Feedback conclusion

The program overall does seem to perform some of it's intended functionality. There are some clear problems with the way some of its functionality throws unexpected errors (B2), does not throw errors, but performs unexpectedly (B1) as well as potentially performs in a problematic way (B3).

I would recommend that the identified bugs are addressed and whilst B3 is potentially not a problem, consider the objective of the product in relation to it's user base.

The expected errors all worked as expected.