Daily/Weekly Scripts

Team Name: ASIC

Ben Lammers
Spencer Venable

Daily Script

Daily script can be run from the terminal while in the Quinterac folder with a call to the module. For example:

```
python -m daily_script valid_account_list.txt master_account.txt new_valid_account.txt new_master_account.txt
```

The script will ask for the user to indicate the amount of front end sessions that will happen that day and then run through each session with the user allowing them to manually input the transactions for that day

Parameters:

1. Valid account list file
    a. Must exist and be a .txt file
    b. Contains list of valid accounts from previous day
2. Master account file
    a. Must exist and be a .txt file
    b. Contains master account information from previous day
3. New valid account list file
    a. Does not need to exist, can be name of file to be created
    b. Must be a .txt
    c. Will contain valid account list at the end of the day
4. New master account list file
    a. Does not need to exist, can be name of file to be created
    b. Must be a .txt
    c. Will contain master account information that is generated after the day ends

Output:

- A transaction summary file for each front end session instantiated
    - The transaction summary file is moved to /transaction_archive folder after being merged to merged transaction file
- Merged transaction file that contains all transactions from other files
- Writes information about the accounts in the provided new master account file
- Writes list of valid account numbers in the provided new valid account list file

Weekly Script

Weekly script can be run from the terminal while in the Quinterac folder with a call to the module. For example:

```
python -m weekly_script valid_account_list.txt master_account.txt
```

The script will ask for the user to indicate the amount of days to simulate and then run through each day with the user to choose how many sessions to instantiate that day and allowing them to manually input the transactions for that day

Parameters:

1. Valid account list file
    a. Must exist and be a .txt file
    b. Can be empty or contain list of valid accounts from previous transactions
2. Master account file
    a. Must exist and be a .txt file
    b. Can be empty or contain information from previous account transactions

Output:

- Generates all output indicated above from the daily script
- Outputs a valid account list and master account text file at the end of each day
    - Suffixed with the number for that given day

Simulation of running the Weekly Script for 5 days

Days	Session One Inputs	Session Two Inputs	Session Three Inputs	Merge Transaction Summary	Master Accounts File
1	Login Agent Createacct 1000327 Spencer Venable Createacct 1000111 Ben Lammers Logout	NA	NA	NEW 1000327 000 0000000 Spencer Venable NEW 1000111 000 0000000 Ben Lammers EOS 0000000 000 0000000 ***	1000111 0 Ben Lammers 1000327 0 Spencer Venable
2	Login Machine Deposit 1000327 10000 Logout	Login Machine Transfer 1000111 1000327 500 logout	Login Agent Deposit 1000000 logout	DEP 1000327 10000 0000000 *** XFR 1000111 500 1000327 *** DEP 1000111 1000000 0000000 *** EOS 0000000 000 0000000 ***	1000111 1000500 Ben Lammers 1000327 9500 Spencer Venable
3	Login Agent Createacct 1111111 Skylar Jane Logout	Login Machine Withdraw 1000111 1000	NA	NEW 1111111 000 0000000 Skylar Jane WDR 1000111 1000 0000000 *** EOS 0000000 000 0000000 ***	1000111 999500 Ben Lammers 1000327 9500 Spencer Venable 1111111 0 Skylar Jane
4	Login Machine Deposit 123214 Transfer 1111111 1000111 1232 Logout	NA	NA	DEP 1111111 123214 0000000 *** XFR 1111111 1232 1000111 *** EOS 0000000 000 0000000 ***	1000111 998268 Ben Lammers 1000327 9500 Spencer Venable 1111111 124446 Skylar Jane

5	Login Machine Withdraw 1000327 5432 Logout	Login Agent Transfer 1000327 1000111 4334 Logout	NA	WDR 1000327 5432 0000000 *** XFR 1000327 4334 1000111 *** EOS 0000000 000 0000000 ***	1000111 993934 Ben Lammers 1000327 8402 Spencer Venable 1111111 124446 Skylar Jane

Integration Defect Report

In our previous release we implemented the creation of the merged transaction file in the back end. While creating the daily script we pulled that function out of the back end and put it in the daily script run file. We then refactored the back end such that it only took in the merged transaction as input and not multiple transaction summary files.

Issue	**Cause**	**Fix**
Daily Script - causing error when given incorrect system arguments	In run.py of daily script in lines 14-17 the script was accessing indices 0-3 of sys.argv where it should have been 1-4	Incremented each index by 1
Front End - Multiple sessions could be written to same transaction summary file	Front end sessions were given option to exit or login again after a logout	Got rid of the option to exit or re-login on logout. Now the front end session exits directly after logout
Back End/Front End Testing Suite - all tests failing based on expected terminal output	Slight adjustments made to print statements in the back and front ends to allow for a better user experience through the daily/weekly scripts	Updated the expected terminal input in the testing suite to match adjusted print statements in the front and back ends