**Cd How to Create Financials from AAII**

1. Create the following directory

~~C:\sundeep\Stocks\_Automation\Downloaded\_from\_AAII\2020\_08\_28(yyy\_mm\_dd)~~

This file location has changed to

C:\sundeep\Private\Investing\Automation\AAII\AAII\_Download\_Every\_Quarter

and the script has been updated to use the new pointer. Maybe I can cleanup the information above

1. Then copy over all the files from the previously created directory into this directory. It should have the following subdirectories:
   * 1. Analysis
     2. Financials - Quarterly
     3. Financials – Yearly
     4. Key Statistics

Note – The best way to create these can be to copy all the files (and subdirs) from the previously existing directory. So for e.g. while creating AAII Financials for 2021\_05\_27, I can copy all the files (and subdirs) from 2021\_03\_28. Then do step #3 (remove all the files from the subdirs)

1. Remove all the files from the 4 subdirectories – They should be empty
2. Next download the AAII data in xlsm file in the above directory. It is just easier to rename the xlsm file from the previous step copy. The name format of the file should be 2020\_08\_28\_AAII\_DATA (yyyy\_mm\_dd\_AAII\_DATA). The file should have to following 9 tabs
   * 1. DateAndPeriod
     2. IncomeSheet-QTR
     3. IncomeSheet-YR
     4. BalanceSheet-QTR
     5. BalanceSheet-YR
     6. CashFlow-QTR
     7. CashFlow-YR
     8. Misc-QTR
     9. Misc-YR
3. Go ahead and remove all the data from the 9 Tabs above – This is needed b/c the number of companies can be different from the previous run and removing all the data before copying new data keeps the run clean
4. Then copy the data into the tabs from AAII Investor Pro window – as listed below
5. After the data is copied over from AAII Investor Pro window – make sure that the last columns matches what is listed below – that is a sanity check for the downloaded data. Also note that out of 9 tabs – there are 5 tabs that use Custom view created by me and 4 tabs that use Standard AAII views.
6. **Last but not the Least –**
   1. **Change the Company name in the col A from TRUE 🡪 TRUEE for all the tabs otherwise excel gets confused and dies when it encounters TRUE – it takes that as a Boolean rather than a Company Name (String)**
   2. **Remove DDMX2\* (yes, there is a ticker with an “ \* ” in the name) – For now I have changed the script to skip that ticker**
7. Then change the date\_str in the macro - **Create\_Financials\_from\_AAII\_v1** - and run
   1. The macro copies the template files from C:\sundeep\Stocks\_Automation\User\_Files\AAII\_Templates
   2. The way I use to align everything on the screen to open the vba sript and make it to a line (using the “Debug” down and select -> “Run to cursor” :
      1. I choose the line “For ascii = 65 To 90” (line number xx)
      2. Then from the Debug drop down, I choose “Run to Cursor”
      3. This will open up the yyyy\_mm\_dd\_AAII\_Data file.
      4. You can “resize that file” and put is where you want in the screen
      5. This also gives to option to put your git bash and windows explorer on the screen where you want.
      6. Generally, I like to have the git bash and windows explorer on the left screen and the open vba code and the excel files on the right monitor
8. The progress of the script can be monitor from the debug file that is created in the Logs directory

~~C:\sundeep\Stocks\_Automation\Logs\yyyy-mm-dd-Create\_Financials\_from\_AAII\_v1.txt~~

C:\sundeep\Private\Investing\Automation\_Not\_in\_Git\Logs\YYYY-mm-dd-Create\_Financials\_from\_AAII\_v1

grep –i “iteration number :” yyyy-mm-dd-Create\_Financials\_from\_AAII\_v1.txt

Generally I just do a tail –f yyyy-mm-dd-Create\_Financials\_from\_AAII\_v1.txt

**How the data is downloaded from AAII for various Tabs:**

The size of the file after the download should be around 32-35MB

1. DateAndPeriod
   * Uses AAII **Custom view “Date and Periods**”
   * Last Column - BJ - Update Type Y7
2. IncomeSheet-QTR
   * Uses AAII **Standard view “Income Statement-Qtr”**
   * Last Column is GG – Dividend Q8
3. IncomeSheet-YR
   * Uses AAII **Standard view “Income Statement-Ann’l”**
   * Last Column is GG – Dividend Y7
4. BalanceSheet-QTR
   * Uses AAII **Standard view “Balance Sheet-Qtr”**
   * Last Column is FY – Book value/share Q8
5. BalanceSheet-YR
   * Uses AAII **Standard view “Balance Sheet-Ann’l”**
   * Last Column is FC – Book value/share Y7
6. CashFlow-QTR
   * Uses AAII **Custom view “Cash Flow Qtr”**
   * Last Column is CP – Free Cash flow/share Q8 – Make sure to correct the formatting in the template file
7. CashFlow-Yr
   * Uses AAII **Custom** **view** “**Cash Flow YR**”
   * Last Column is CD – Free Cash flow/share Y7
8. Misc - QTR
   * Uses AAII **Custom view “Misc-QTR”**
   * Last Column is Z – Retained Earnings Q8
9. Misc - YR
   * Uses AAII **Custom view “Misc-YR”**
   * Last Column is X – Retained Earnings Y7

**Once you have run the script and it has completed, then**

1. Sanity check a file to make sure it has the correct financials. I double check IBM financials by comparing them to Yahoo Finance financials
2. Once a sanity check is done - The whole directory needs to be git added
   1. Run from Stock\_Automation directory

git add Downloaded\_from\_AAII/2021\_03\_28

* 1. Then go to the directory Downloaded\_from\_AAII and change the value of the variable dir\_name in the script .git\_commit.sh to the date that corresponds to the date when AAII data was downloaded – in this case

dir\_name='2021\_03\_28'

* 1. Then run the .git\_commit.sh script – The script will recursively go into the 4 sub-directories and commit the files
  2. ~~Then commit the file 2021\_03\_28/2021\_03\_28\_AAII\_DATA.xlsm – as somehow this does not get committed (Probably need to modify the .git\_commit.sh to include the commit for this file too).~~

The script, git\_commit.sh,cd .. now handles it, but make sure that the AAII Downloaded file is committed too.

* 1. Don’t forget to commit the file .git\_commit.sh itself as that file has the date changed
  2. Don’t forget to commit AAMSC\_Macros.xlsm from the scripts directory, (and push it if you don’t plan to run merge\_financials (that would be odd but just saying ☺, why would you create/extract the financials and not merge them ??)
  3. Finally push everything

**How to Merge Financials from Just created Financials**

The last step is to merge the freshly created financials with the already existing (past history) of the financials.   
The (running) historical financials (with past history) are available in the directory

~~C:\sundeep\Stocks\_Automation\AAII\_Financials~~

C:\sundeep\Private\Investing\Automation\AAII\AAII\_Financials

In order to merge the freshly created financials –

1. Change the date\_str in the macro **aaii\_merge\_financials** in AAMSC\_Macros.xlsm file and run it.
2. After the script runs, do a sanity check for some of the files updated that you know will have a column added for quarterly and yearly.   
     
   For e.g I can look at IBM\_QTR\_FIN and IBM\_YR\_FIN.   
     
   Most of the tickers should have a quarterly added.   
   For Yearly, that ticker is can be found from the logfile   
     
   ~~C:\sundeep\Stocks\_Automation\Logs\2021-05-28-aaii\_merge\_financials~~  
   C:\sundeep\Private\Investing\Automation\_Not\_in\_Git\Logs\

by grepping   
  
grep -i "Yearly" 2021-05-28-aaii\_merge\_financials.txt | grep -i " will be inserted"

This will give you a full list of tickers displaying if there were new columns inserted for not. Pick a couple for which 1 column was inserted and check the quality of the data (check the numbers from either C:\sundeep\Stocks\_Automation\Downloaded\_from\_AAII\2021\_05\_27\Financials – Yearly or for that financial Year from Yahoo Finance)  
  
Once the quality of data check is done, then go to AAII\_Financials directory and run the script:  
./git\_commit\_financials.sh

There is nothing that needed to be changed in that script

1. Once the script is run, then do a git push
2. Finally – don’t forget to commit and push AAMSC\_Macros.xlsm from the scripts directory. That is the file that made all the AAII financials to be extracted and updated possible.
3. At this point, you are all done.