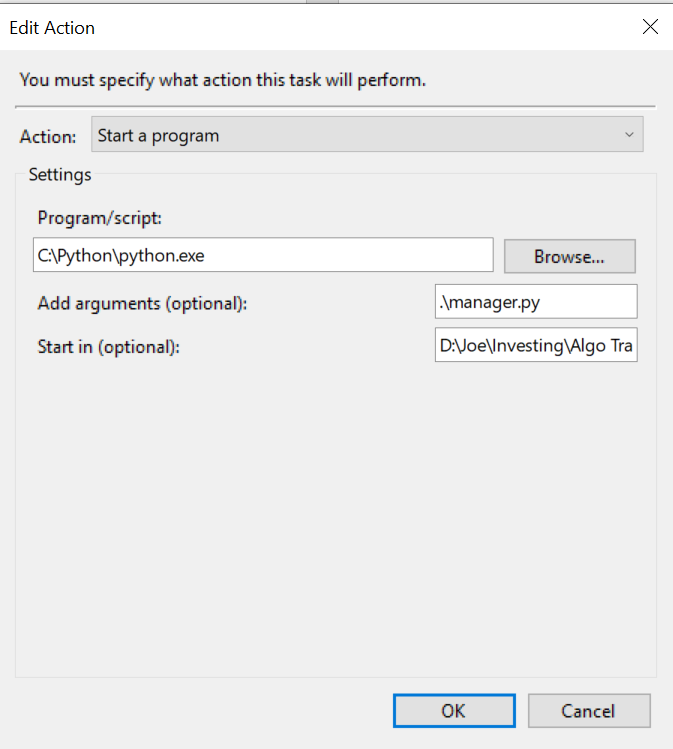
SEC Scrape Run Instructions

1. Manually download stock list csv from NASDAQ stock screener: <https://www.nasdaq.com/market-activity/stocks/screener>
2. downloadYF.py: download historical prices for each stock from Yahoo Finance; rate of ~500 symbols/hr
3. getCIK.py: generate a map of ticker symbol to CIK for downloading from SEC and save to dictionary structure
   1. <https://www.sec.gov/include/ticker.txt>
4. downloadSummaries.py: use CIK map to download Filing Summary.xml for each ticker from SEC; rate of ~50 symbols/hr
5. getReportLinks.py: from each filing summary, get all report links and save all URLs to dict. struct.
6. checkReportLinks.py: for each ticker, find a URL for each type of report (ex. ‘balance’, ‘income’, etc.); save specific report URLs to dict. struct.
7. downloadReports.py: download reports (financial statements) from URLs, saved as .html
8. parseReports.py: parse each table in report (html format) and save to csv

Finpie Run Instructions

1. filterYF.py: filter yahoo finance price data for stocks which has data for most trading days of the year
2. initialize.py: set up download directories, log and ticker position text files
3. manager.py: calls multiple instances (4-5 most stable) of getStatements.py to download in parallel. getstatements.py takes in a copy number and ticker stop position as arguments.
4. Use windows scheduler to run manager.py on restart (in case of losing internet connection). Create basic task, follow dialog box and under actions, enter folder location of python.exe, file name with “.\” before it in arguments, and file location in “Start in”. All entered without quotes.
   1. 
5. checkIncome.py: after finish downloading need to check all income statements are downloaded for every cashflow statement downloaded
6. findMissing.py: find tickers who don’t have cashflow or income statements downloaded. Also update “No Download.txt” with tickers who have no statements available. Outputs “Missing Symbols.txt” (make sure no such file exists before running this because this will append to that file).
7. Use manager.py to download any missing symbols.

Notes:

* In base.py from Python yfinance package, edit to look like below to avoid json parsing errors when downloading from Yahoo Finance:

        try:

            data = data.json()

        except ValueError:

            print('Decoding JSON has failed')

* In Finpie package changed base.py, \_load\_driver function to suppress chrome output to console:
  + options.add\_experimental\_option("excludeSwitches", ["enable-logging"])
  + options.add\_argument("--log-level=3")