# MBA 614 – Final Project
# Automated Stock Screener Using Graham's Ten Hurdles

Tony Liu

April 15, 2014

## Executive Summary:

In the asset management industry, portfolio managers who actively manage stock portfolios spend much of their time performing cursory evaluations of many different companies to find stocks which may be good candidates for deeper analysis. There are many different methods that portfolio managers will use to perform this first-pass evaluation of different companies. One such method is the "Graham Screener".

Benjamin Graham is widely regarded as one of the most successful and influential value investors of all time. His book, *Security Analysis*, describes the guiding principles of Value Investing, many of which are still followed to this day. As part of his work, Graham came up with a list of ten "points" or "hurdles" which a stock should pass in order to be considered a good value. The Ten Hurdles are defined as follows:

1. An earnings-to-price yield of twice the triple-A bond.
2. A P/E ratio down to four-tenths of the highest average P/E ratio the stock attained in the most recent five years.
3. A dividend yield of two-thirds the triple-A bond yield.
4. A stock price down to two-thirds of tangible book value per share.
5. A stock price down to two-thirds of "net current asset value" (defined as current assets less total debt).
6. Total debt less than tangible book value.
7. Current ratio of two or more.
8. Total debt equal or less than twice the net quick liquidation value.
9. Earnings growth over the most recent ten years of 7% compounded.
10. No more than two declines of 5% or greater in year-end earnings (relative to the previous year) in the most recent ten years

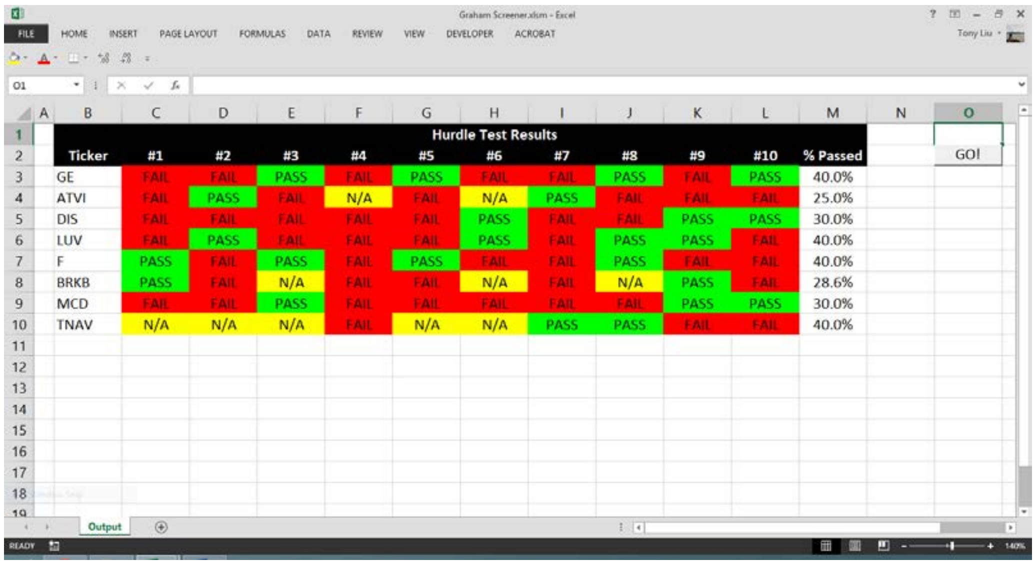
These criteria are very data intensive and it would take quite a long time for a fund manager to manually look up the numbers and perform the comparisons on them one by one. In addition, these ten criteria are very rigorous and it is difficult for the vast majority of stocks to pass all ten. This project is an attempt to automate the process of applying the Graham Screener to a number of stocks based on a list of ticker symbols provided by the user.

## Implementation and Documentation:

The intended audience of this software is primarily non-technical stock portfolio managers who need fast and accurate information that is easy to understand. Keeping in mind the primary audience, the requirements defined for this software were primarily speed, accuracy, and ease of use. Comparable systems that I found often used web queries and multiple user-facing tabs of data that appeared both cumbersome and slow. I sought to improve upon the performance of these systems by stripping the user interface down to its bare minimum components and speeding up the data retrieval process as much as possible.

### User Interface:

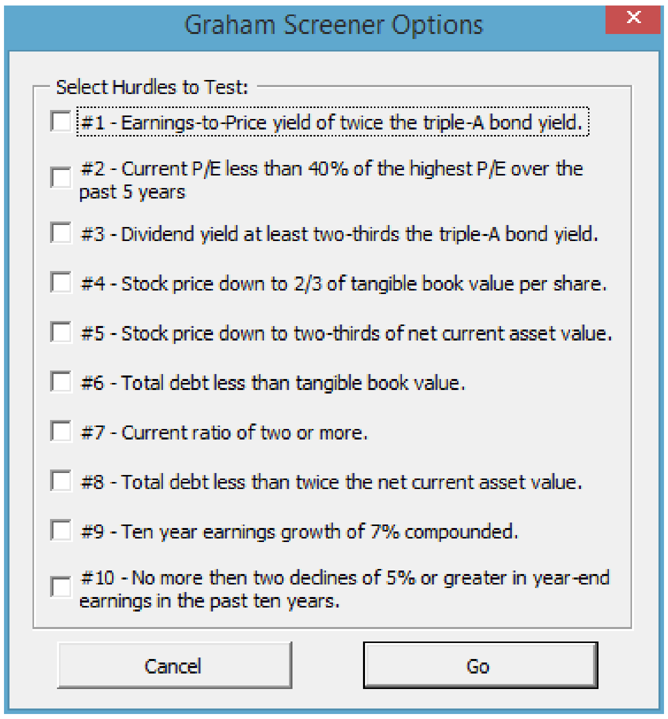
The user interface consists of just two portions: the main interface and the hurdle selection menu. On the main interface, the user simply inputs a list of ticker symbols in the first column of the table, then clicks on the 'GO!' button to the right of the table. The user is then brought to a menu which they can use to select which hurdle(s) they would like to test using the provided checkboxes. Once the desired hurdle(s) are selected, the user simply selects 'GO' from this menu and the software will retrieve the data from the internet and perform the tests. If no hurdles are selected, the system will prompt the user to select a hurdle before proceeding.



Hurdle Test Results

| Ticker | #1 | #2 | #3 | #4 | #5 | #6 | #7 | #8 | #9 | #10 | % Passed |
|---|---|---|---|---|---|---|---|---|---|---|---|
| GE | FAIL | FAIL | PASS | FAIL | PASS | FAIL | FAIL | PASS | FAIL | PASS | 40.0% |
| ATVI | FAIL | PASS | FAIL | N/A | FAIL | N/A | PASS | FAIL | FAIL | FAIL | 25.0% |
| DIS | FAIL | FAIL | FAIL | FAIL | FAIL | PASS | FAIL | FAIL | PASS | PASS | 30.0% |
| LUV | FAIL | PASS | FAIL | FAIL | FAIL | PASS | FAIL | PASS | PASS | FAIL | 40.0% |
| F | PASS | FAIL | PASS | FAIL | PASS | FAIL | FAIL | PASS | FAIL | FAIL | 40.0% |
| BRKB | PASS | FAIL | N/A | FAIL | FAIL | N/A | FAIL | N/A | PASS | FAIL | 28.6% |
| MCD | FAIL | FAIL | PASS | FAIL | FAIL | FAIL | FAIL | FAIL | PASS | PASS | 30.0% |
| TNAV | N/A | N/A | N/A | FAIL | N/A | N/A | PASS | PASS | FAIL | FAIL | 40.0% |

*Figure 1: Main User Interface*



Graham Screener Options

Select Hurdles to Test:

- ☐ #1 - Earnings-to-Price yield of twice the triple-A bond yield.
- ☐ #2 - Current P/E less than 40% of the highest P/E over the past 5 years
- ☐ #3 - Dividend yield at least two-thirds the triple-A bond yield.
- ☐ #4 - Stock price down to 2/3 of tangible book value per share.
- ☐ #5 - Stock price down to two-thirds of net current asset value.
- ☐ #6 - Total debt less than tangible book value.
- ☐ #7 - Current ratio of two or more.
- ☐ #8 - Total debt less than twice the net current asset value.
- ☐ #9 - Ten year earnings growth of 7% compounded.
- ☐ #10 - No more then two declines of 5% or greater in year-end earnings in the past ten years.

*Figure 2: Hurdle Selection Menu*

Once the program finishes running, the main interface will update to reflect the results of the tests with either 'PASS', 'FAIL', or, if that particular piece of data was not available for that company, 'N/A'. Additionally, the software will also compute a total '% PASSED' value based on the number of passed hurdles vs successfully run hurdles. Hurdles which were not successfully run are not included within this calculation.

## Data Retrieval:

Once the user clicks 'Go' on the hurdle select menu, the code clears the previous results and then begins retrieving data from the internet. The first thing it does, after making sure the user has selected at least one hurdle to test, is use the *agent* class to go to Yahoo Finance's bond page to retrieve the latest 10-yr AAA Corporate Bond Yield. It takes this data and stores it into a Public variable named 'bondYld' (declared as type Double).

Next, it begins at the top of the tickers list and passes each ticker symbol to another sub procedure named *processTicker* which uses it to gather all the relevant data from the internet. The *processTicker* sub procedure gets all the necessary data from three sources: Reuters.com, and two different Morningstar pages.

To get the appropriate data from Reuters, the *agent* class is used once again. The code goes through and locates the Price per Share, P/E, 5 Year High P/E, Tangible Book Value per Share, and the Dividend Yield. It then takes each piece of data and stores it into an appropriately named Public variable.

For the two Morningstar pages, I was able to figure out how to automate the csv export function, which sped up the data retrieval process considerably. From these two pages, I downloaded and opened the csv files directly from Morningstar, copied them into the local workbook, and closed the downloaded copies. I was able to search for certain key words, such as "Total current assets", within these files and grab the correct data due to the standardized format that Morningstar uses for all of its financials. All in all, I was able to get the rest of the data that I needed from these two pages. After all the data is assigned to the variables, the code automatically deletes the two sheets it used to pull this data in.

To prevent errors when the code tries to assign non-numeric values to the Public variables, I made sure to precede the code with [On Error Resume Next]. In these cases, I assumed that the website had no data for those particular metrics, so I initialized all of my variables to a value of -1 so that if the *processTickers* sub procedure finished and the variable was still -1, then the hurdle result would return 'N/A'.

After retrieving the data for each tickers, the code checks to see which checkboxes are checked using a series of If…Then statements. When a particular checkbox is checked, the code will call another sub procedure named *hurdle**X***, with X being equal to the number of the corresponding hurdle. Each sub procedure performs the comparison required for its respective hurdle and then assigns the 'PASS'/'FAIL'/'N/A' value into the appropriate cell. After all the selected hurdles are complete, the system calls a *summarize* sub procedure which counts the number of passed hurdles and divides it by the total number of successfully run hurdles. The code then moves on to the next ticker symbol and repeats this process until all tickers have been looked at.

## Discussion of Learning:

The biggest thing that I learned while working on this project was that I needed to keep my objectives in mind when building my program. I was temped many times to use a web query or open several different websites to pull in data. While these methods would have provided functional code, they would have been far too slow to be of any practical use to me or anyone else.

By keeping my objectives of simplicity and speed in mind, I was able to minimize the number of page calls by only visiting websites with the most useable data for my project. Additionally, after figuring out that importing a csv from the web is much faster than navigating to the same data on a web page, I did my best to maximize my usage of that particular mode of data retrieval.

A second major skill that I honed through doing this project was the ability to really understand the code that I was looking at and be able to debug problems much more quickly using the debugging tools built into the VBA Editor. An example of this is when I was having some trouble with the *waitForLoad* method in the *agent* class. I reached a point in my project where the data retrieval process was hanging indefinitely. I traced the problem line by line by stepping through the code in the VBA editor. I quickly realized that the code was hanging in the *waitForLoad* method as the ie object would not return that it was ready for some reason. Once I figured this out, I was able to bypass this code and just have the application wait for a few seconds before proceeding. While this solution wasn't elegant, it certainly worked for my purposes and I was able to get my program working again.

## Assistance:

I did not receive substantial help from another person on this project.