

---

# MBA 614 Final Project

---

Brian Clegg  
April 13, 2011

# Table of Contents

Executive Summary.....	Page 3
Overview.....	Page 3
Business Description.....	Page 3
System Overview.....	Page 3
Implementation and Documentation.....	Page 4
Learning Discussion.....	Page 8
Omissions / Future Projects.....	Page 9

# Executive Summary

## Business Description

I've been trying my hand at investing in companies via the stock market for the last three years. I try to take a long-term, ownership view – investing as though I was actually buying the entire company in a private transaction with the intention of holding for a multi-year period. Despite my intention of taking this long-term view, the reality is that the stock market provides a price at which I can buy or sell my positions any time that the market is open at the click of a button and with very low transaction costs. This has the potential to be either a blessing or a curse. If I've done a reasonable job of placing a value on the company, the market fluctuations allow me to occasionally purchase more of the company at low prices and sell when prices are attractively high. On the other hand, the fluctuations in the market also send a signal to me on what others think my business investments are worth. If these prices fall outside of my valuation, it has the tendency to make me question my valuation. This can be a good thing, but it also has a tendency to make me complacent when the market agrees with me and very nervous when the market disagrees. In this case the market can create emotions that lead to poor decision-making – buying high and selling low – rather being an effective tool to allow me to do the opposite.

Recognizing this potential problem, I've tried to set up a process that helps remove the emotion from my investment decisions. One of these is that I try to trade 1-2% of the portfolio every week – no more and no less. This forces me to move methodically. I can't come across an idea and get really excited and buy a big position all at once, nor can I get all nervous and sell out all at once. Instead, I try to be disciplined to slowly add to the positions in which I have the most confidence in falling markets and trim positions slowly as the market rises.

One of the problems I've had in executing this weekly portfolio culling process is that I haven't had a good "dashboard" view of the overall portfolio to help me compare my positions in a non-emotional way so that each week I'm actually adding to the most attractive positions and selling the least attractive based on my evaluation and current market prices rather than what I "like" at the moment. I've created this dashboard for my VBA project.

## System Overview

I found it easier include the overview in the Implementation section below.

## Implementation and Documentation

The first step was to create a dashboard template with the relevant columns for the information I needed for my decision-making process. Following is a snapshot of the dashboard template. It has been populated with several rows of data to show how the final product will look.

Ticker	Shares Owned	Current Price	Target Price	% NAV	Sigma Score	Timing Score	Quality Score	Liquidity Score	Total Score	Dollars Invested	% Portfolio	Average Basis	Low Basis	High Basis
FSIN	1100	7.97	11.50	69.3%	3	3	3	3	3.0	\$8,767	7.9%	9.75	7.69	14.00
HOFD.OB	250	28.00	35.00	80.0%	3	2	4	1	2.7	\$7,000	6.3%	21.85	21.50	23.25
SKYW	400	15.79	24.00	65.8%	4	3	4	3	3.6	\$6,316	5.7%	16.39	15.77	17.00
FFNW	1000	5.83	9.00	64.8%	2	3	3	2	2.5	\$5,830	5.3%	4.82	4.50	4.90
AON	90	52.21	55.00	94.9%	4	3	4	5	4.0	\$4,699	4.2%	40.17	40.61	45.06
CNQ	80	46.07	55.00	83.8%	2	3	4	5	3.4	\$3,686	3.3%	34.29	24.91	37.42
WMT	60	53.52	65.00	82.3%	4	3	5	5	4.3	\$3,211	2.9%	55.95	55.00	56.97
GEN	750	3.70	7.00	52.9%	2	2	3	3	2.5	\$2,775	2.5%	3.90	3.85	4.00
APC	30	77.43	95.00	81.5%	3	3	4	5	3.7	\$2,323	2.1%	32.80	31.80	38.60
DSWL	700	3.18	5.00	63.6%	3	2	2	2	2.3	\$2,226	2.0%	3.50	3.50	3.50
LUK	60	36.40	40.00	91.0%	3	3	4	3	3.3	\$2,184	2.0%	27.03	23.50	28.80
HERO	400	5.20	9.00	57.8%	1	4	3	2	2.4	\$2,080	1.9%	2.50	2.50	2.50

Most of the columns are self explanatory, but I'll describe several of them just in case. "Target Price" is a somewhat subjective judgment of what I believe a share of the company is worth. "% NAV" is the current price of a share divided by the target price, showing market value as a percent of my fair value. The various "Scores" are subjective inputs. "Sigma" refers to variability in potential investment outcomes. This variability can be based on the business model, the industry, or the quality of the information on which I have based my valuation analysis. The others scores are more self explanatory. The three columns on the left have to do with cost basis. They provide important information regarding the tax consequences of selling a position. Since most positions were accumulated at several different prices, I've included average a "high" basis and "low" basis which correspond to the highest and lowest price at which a stock was purchased.

Since the inputs for "Target Price" and the various "Scores" are based on judgment rather than calculations, these data are entered manually and have been formatted in blue to distinguish them from cells that populate automatically via VBA code or worksheet formulas. There is one exception. The tickers are entered manually, but are in black. I just didn't like the way it looked when I put them in blue, and I since they're letters, I didn't think I'd get them mixed up with all the numerical data.

The dashboard ties to trade data that is contained in a table on a separate worksheet in the workbook labeled "Trades." A snapshot of the first several rows of this table is found below.

Date	Buy/Sell	Ticker	Shares	Price	Commission	WA Cost Basis
	Shares In/Shares Out					
2/19/2008	Buy	FISV	30	52.35	0	1570.5
2/29/2008	Buy	AON	80	40.61	1	3248.8
2/29/2008	Buy	BR	125	19.15	0	2393.75
2/29/2008	Buy	BRUFX	12.84	339.8	0	4363.032
2/29/2008	Shares in	Cash	73505.86	1	0	73505.86
2/29/2008	Buy	CNQ	60	37.42	0	2245.2
2/29/2008	Buy	FAIRX	588.125	31.97	0	18802.3563
2/29/2008	Buy	FAST	35	40.84	0	1429.4
2/29/2008	Buy	JCI	30	32.86	0	985.8

This table isn't particularly pretty, but it gets the job done. I enter each trade manually into the table in the next available row. I suppose I could have made a fancy form in which to enter the data, but I think it's just as easy to enter data into spreadsheet as a form. I may have been able to pull the data from my brokerage account, except for the fact that my brokerage requires me to enter a randomly generated code to login. Also, entering a couple trades a week is not very difficult and I like having a hand-made record I can reconcile with my brokerage records.

The only other data needed to populate the dashboard are current market prices, which are pulled from Yahoo! finance using a web query.

## Implementation

I found it easier to break the VBA code into several small steps and write a sub procedure for each. Following are the steps and the corresponding code.

### 1. Update current stock price via web query

```
Sub updatePrices()
'Updates prices only for the sheet starting from the active cell
ClearImportData
Do Until ActiveCell.Value = ""
    Application.ScreenUpdating = False
    getPrice
    ActiveCell.Offset(1, 0).Range("A1").Select
    Application.ScreenUpdating = True
Loop
Range("B4").Select
End Sub
```

Similar to what we did in class, I wrote the following code that includes two other sub procedures: ClearImportData, which clears tab to which the data is downloaded; and getPrice, which is a web query pulling data from the web. The getPrice sub

procedure was written primarily using the macro recorder.

## 2. Tally up the number of shares owned

I summed the trades using a SumIf worksheet function as seen below. Interestingly, I had previously just used this formula in the dashboard table (no VBA), but it would get all messed up if I sorted the table, thus necessitating the VBA code.

```
Sub sumPositions()  
    ' Sums up the number of shares owned from the Trades worksheet  
  
    Dim sumTrades As Double  
  
    sumTrades = Application.SumIf(Sheets("Trades").Range("d4:d1000"), ActiveCell.Value, Sheets("Trades").Range("E4:e10000"))  
    ActiveCell.Offset(0, 1).Range("A1") = sumTrades  
    'ActiveCell.Offset(1, 0).Range("A1").Select  
  
End Sub
```

## 3. Calculate average cost basis

The following sub procedure calculates the average cost basis. I was able to use the sumTrades sub procedure from the previous step as part of this procedure.

```
Sub averageBasis()  
    'Finds the average cost basis for a given position  
  
    Dim sumTrades As Double  
    Dim avgBasis As Double  
  
    sumTrades = Application.SumIf(Sheets("Trades").Range("d4:d1000"), ActiveCell.Value, Sheets("Trades").Range("E4:e10000"))  
  
    If sumTrades <> 0 Then  
        avgBasis = Application.SumIf(Sheets("Trades").Range("d4:d1000"), ActiveCell.Value, Sheets("Trades").Range("h4:h10000")) / sumTrades  
    End If  
  
    ActiveCell.Offset(0, 14).Range("A1") = avgBasis  
    ' ActiveCell.Offset(1, 0).Range("A1").Select  
  
End Sub
```

## 4. Calculate low cost basis

```
Sub lowBasis()  
    ' Finds the lowest price paid for a given position  
  
    Dim low As Double  
    Dim r As Integer  
    low = 500000  
  
    For r = 1 To 10000  
        If Sheets("Trades").Cells(r, 4).Value = ActiveCell.Value Then  
            If Sheets("Trades").Cells(r, 6).Value < low Then  
                low = Sheets("trades").Cells(r, 6).Value  
            End If  
        End If  
    Next  
  
    If low = 500000 Then  
        low = 0  
    End If  
  
    ActiveCell.Offset(0, 15).Range("A1") = low  
    'ActiveCell.Offset(1, 0).Range("A1").Select  
  
End Sub
```

This was a little trickier for me. I couldn't figure out a way to either record the macro or use a worksheet function. I wrote the following code that looped to find the lowest trade value per corresponding ticker.

## 5. Calculate high cost basis

```
Sub hiBasis()  
  
'Finds the highest price paid for a given position  
  
Dim hi As Double  
Dim r As Integer  
hi = 0.001  
  
For r = 1 To 10000  
If Sheets("Trades").Cells(r, 4).Value = ActiveCell.Value Then  
If Sheets("Trades").Cells(r, 6).Value > hi Then  
hi = Sheets("trades").Cells(r, 6).Value  
End If  
End If  
Next  
|  
If hi = 0.001 Then  
hi = 0  
End If  
  
ActiveCell.Offset(0, 16).Range("A1") = hi  
'ActiveCell.Offset(1, 0).Range("A1").Select  
  
End Sub
```

I did this just like calculating the low basis, only in reverse.

## 6. Sort the data by largest position

```
Sub Sort()  
  
'This sorts the data by in descending order by Dollars Invested  
  
Range("B4:R4").Select  
Range(Selection, Selection.End(xlDown)).Select  
ActiveWorkbook.Worksheets("Portfolio").Sort.SortFields.Clear  
ActiveWorkbook.Worksheets("Portfolio").Sort.SortFields.Add Key:=Range( _  
    "N4:N37"), SortOn:=xlSortOnValues, Order:=xlDescending, DataOption:= _  
    xlSortNormal  
With ActiveWorkbook.Worksheets("Portfolio").Sort  
    .SetRange Range("B4:R37")  
    .Header = xlGuess  
    .MatchCase = False  
    .Orientation = xlTopToBottom  
    .SortMethod = xlPinYin  
    .Apply  
End With  
  
End Sub
```

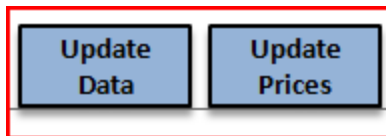
This sub sorts the table by position size once it has been loaded into the dashboard.

## 7. Put it all together

```
Sub updateData()  
    ClearImportData  
    Sheets("Portfolio").Range("B4").Select  
    Do Until ActiveCell.Value = ""  
        Application.ScreenUpdating = False  
        getPrice  
        sumPositions  
        averageBasis  
        lowBasis  
        hiBasis  
        ActiveCell.Offset(1, 0).Range("A1").Select  
        Application.ScreenUpdating = True  
    Loop  
    Range("B4").Select  
    Sort  
    Range("B4").Select  
End Sub
```

This sub procedure is made up primarily of the other subs and makes it all happen.

## 8. Created buttons to run the procedure



These two buttons sit next to the Dashboard. The Update Data runs all of the sub procedures necessary to update all the data. The Update Price button only updates prices, beginning with

the ticker of the active cell. I wanted to be able to update the prices alone since new trades are not added nearly as often as I'd like to see new prices.

# Learning Discussion

I actually learned conceptually how VBA works, how to use VBA for a web query, how to use the macro recorder (including how to interpret results and edit them as needed), and how to write some pretty simple codes such as loops. Frankly, I didn't really understand any of this very well until working on this project – despite a wonderful teacher and having done my homework. While I'm still a long way from being able to use VBA extensively, I now feel comfortable tackling relatively simple tasks that I can use to make my spreadsheets more efficient. I also believe I could get started a start on more complex issues and finish them with some effort and help from google.

There were some specific issues I came across that gave me some trouble, but I was able to work around.

### 1. getPrice web query

I would occasionally have trouble with the imported data getting bungled. I was able to clear this up by writing a simple sub procedure to clear import tab of any data that might have been left there before running the web query. Also, the prices data would



populate the dashboard in a random format. I added a line of code to center the data in a number format with 2 decimals. I recorded a macro to do it.

2. Summing the number of shares owned

I initially had this as just a sumif formula in the workbook (no VBA), but when I sorted the data, the references would get all messed up. It took me a while to remember that I can use most of the worksheet functions in VBA. Once I figured out how to use this on a single ticker reference, I was able to apply a loop to go through the whole sheet. I used this same method to calculate my average cost basis.

3. High and low average cost basis

I thought I could do this easily using a worksheet function, but couldn't find a function for finding conditional minimum or maximum values. I solved the problem by defining a variable with a very high number then using a "For loop" (is that what it's called?) to cycle through the trade price data and replace it with any smaller trade value (for the low basis). The smallest number became the new min value. I did the reverse to find the max.

4. Sorting

I thought sorting the data was going to be really easy using the macro recorder, which I was getting pretty good at. It turned out to be harder than I thought if I wanted my range of data being sorted to expanded with new rows. The sorter would always select a static range, and my attempts to stick a variable for the number of the last row into the code were fruitless. I worked on this for good long while, and even found a few examples of others that accomplished the same task on the internet. It didn't seem hard and made sense conceptually, but I just couldn't get this detail to work in time. This is something I plan to fix sometime after April 13<sup>th</sup>.

That's all I've got. If you've read this far I'm impressed. Before this project, I was a bit frustrated because I really didn't feel like I really understood much of what we did in class. Doing the project brought a lot of the pieces together for me. Thanks!