Spencer Shumway

MBA 614/IS 520

Gove Allen

Winter 2015

Final Project Write-up

Executive Summary-

I work at a Financial Planning Firm named Merrill Financial Associates. We help people decide how to allocate their money for optimal performance and minimal risk. We are also concerned with making sure that client's investments are liquid so that they can access them when they need them. We are a fee based advisor ship, with our yearly rate being 1.5% of a client's portfolio. We manage over $115 million, with estimated growth in the next 10 years between $200 and $300 million. In order to charge a fee from a client, which we do quarterly, there must be cash in their account. Often this means we must sell a portion of their holdings to pay for the fee. I am tasked with doing this, and it can be very time consuming. This project will probably save me about 3 days or work per quarter. I have automated the process of finding short accounts, finding necessary holdings, and deciding whether those holdings are suitable for sale.

Implementation documentation-

In order to understand this project you must first know that it contains private client information, so I can't show those parts of the project. For the graders ease I have supplied fake account numbers so they can understand the process and output of what will happen when the project is running. Another important element to understand is that at Merrill we have a master model that contains all relevant information in one form or another. The challenge is finding the information and applying it in a way that helps us accomplish our job. This "Fee Trading" Workbook references the "Data" Workbook quite a bit. They are symbiotic, meaning they work together for the greater good! I cannot supply the "Data" model out of concern for privacy, but I will explain the code well enough that you can see what it does.

The most important thing that I did to simplify implementation was define workbook and worksheet objects. I talked to Professor Allen about how to increase the speed of my program, and one of the main things he mentioned was that where possible I should get rid of Sheets("example").activate type syntax. I did that for the most part, especially during the meat of my program, which is contained in the ShortNTF Sub procedure. With that functionality I was able to move around and gather relevant information with much more ease and confidence.

I used a Search to find the column number that an account first appeared

I took the values from a table like this:

B3S3855555	CASH	WMFFX	B3S385555	4707.984	37	WMFFX	09 Large Cap Value/Blend	B3S38956009 Large Cap Value/Blend	41.22	114.216
B3S466666	MARGIN	FOCPX	B3S406666	1777.411	1	FOCPX	10 Large Cap Growth	B3S40022010 Large Cap Growth	85.51	20.786
B3S407777	CASH	HLEMX	B3S407777	834.8473	2	HLEMX	18 Emerging Large Cap	B3S40022018 Emerging Large Cap	49.1	17.003

Instead of looping through every Account Number I could go straight to the account I needed and take the Ticker symbol (third from the left) and the Price (second from the right) and the Share Amount (first from the right). From there I could do a loop that would loop only as long as we were on the correct account number.

I then used another search to find out of the Ticker symbol was a valid NTF. If it was I checked to see whether it had sufficient funds to cover a sale for fees. If it did, I then assigned all the appropriate values to the CSV sheet. The CSV sheet automatically is exported and saved as its own file.

When the process starts again all the Cells on the Accounts sheet are wiped clean, and the CSV sheet is deleted and recreated to ensure it is correct. The NTF sheet is actually pretty static, because the funds we use don't change much.

The only thing that is set up to be viewed is the CSV sheet, which is well organized and unfrilled. It is exactly how our Traders like it, so it is perfect for what it is intended to do.

The final Output looks like this:

B36694517	1	SF	702.71	AMANX	M	D	D
B36748447	2	SF	42.32	BSCFX	M	D	D
B36775231	1	SF	1220.67	PRBLX	M	D	D

It contains:

Account Number that Trade will be placed in

Account Type- Cash or Margin (basically states whether or not this holding can be borrowed against, but that is a discussion for another time)

What action we will take- It will always be SF for Sell Funds

How much to sell

The Ticker Symbol of the NTF we will sell out of

3 other categories used by our traders that are meaningless to outside observers- they detail the sale type, the timeline, and whether or not we have discretion to sell these accounts. For this project we will always have discretion. (non-fee based accounts are the ones we don't have discretion on, and we will never be selling for fees in those accounts)

Here is the URL for the actual Fee Trading Workbook:
http://files.gove.net/shares/files/15w/sshumwa4/Fee_Trading.xlsm

Learning and Conceptual Difficulties Encountered-

Probably the most time consuming problem I faced had to deal with unnecessary looping logic. After I had gathered all the short accounts from the "Data" model and calculated them with my GetShortAccts sub procedure, I then searched the holdings sheet of the "Data" model to find all the holdings. Originally I did this by going to Column G and looping through the entire column until I found an account number that matched the account I was looking for. It looked something like this:

```
Do
If  cells(x,7).value = AccNum then
Holding = cells(x,9).value
Fund = cells(x,10).value
Loop until cells(x,7).value = ""  ' Meaning I would loop through the entire Acct Number list which is like 50,000 lines!
```

Once I found an account that finally matched(which took forever) I had to grab all of the funds the account owned one by one and check to see if they were an NTF account. I did this by then putting that fund String in a comparison with a list of NTF funds and looping through each cell in the NTF list individually till it matched. There are over 2,000 NTF funds that we used, so that was also very time consuming to do for every fund!

This was taking forever, so I asked Gove, and he suggested I use the Find tool and that it would dramatically save time. I did so by implementing this code for the account number, a value that will always exist in the data page and that can't reasonably throw an error:

```
Cells.Find(What:=accNum, After:=ActiveCell, LookIn:=xlFormulas, _
LookAt:=xlPart, SearchOrder:=xlByRows, SearchDirection:=xlNext, _
MatchCase:=False, SearchFormat:=False).Activate
```

This code however did not work when testing to see if a fund was an NTF because if you entered in a fund that wasn't an NTF it would cause an error. I spent probably an hour trying to find my way around this, but Professor Allen suggested I use on of the functions he had created, and it did the trick perfectly!

I also wanted to have the all the trades to automatically output into a CSV file (in other words a file that is compatible with our home office trading system, It's not technically a CSV file), that part wasn't so hard, but I also wanted the CSV sheet within the "Fee Trading" Workbook to automatically export and save in a different folder for ease of organization and sharing. That would enable me to track specific

dates and times when trade sheets were created, and also easily enable me to just share them with my boss or our home office traders. I had a decent amount of trouble with the actual application though. I had originally used the syntax:

```
day = Replace(Date, "/", "-")
Path = "Z:\1. Investing Operations\Trades\Fee Trades\"
FileName = "Fees CSV " & day
ActiveSheet.SaveAs FileName:=Path & FileName & ".xlsx", export xlFileFormat (or something like this)
```

It was almost working, but it would stop on the ActiveSheet.SaveAs and I could not for the life of me figure out why. I experimented with everything, and finally got rid of the final statement

```
ActiveSheet.SaveAs FileName:=Path & FileName & ".xlsx"
```

It worked just fine after that!

Assistance-

I did not receive substantial help from anyone. In fact I didn't receive help from anyone besides Dr. Allen. He gave me some ideas on how to maximize my efficiency through using the Find instead of looping through every cell individually. He also directed me to his LookUp Function that helped me to overcome my problem of searching a value that wasn't contained in a field.