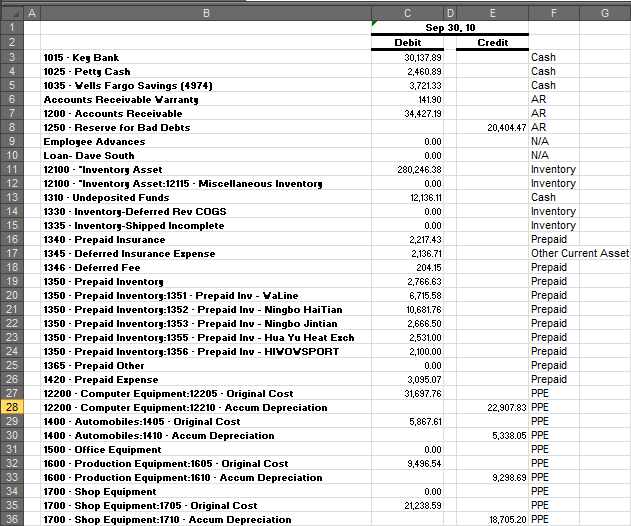
**Executive Summary**

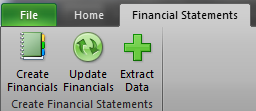
I am currently the Accounting Manager at my company, and one of my responsibilities entails creating the financial statements in excel. We are currently using QuickBooks, but it was improperly structured; so I must export the trial balances of the company into an excel spreadsheet, and then manually create the income statement, balance sheet, and cash flow statement (with accompanying spreadsheets that were used to build the cash flow statement). The auditors have also asked that I color code the trial balances so they are easy to look at and understand. In addition to the colors, the sides of each line item will have some type of descriptor (e.g. cash, accounts receivable, etc.). I want this process, the descriptors, to remain manual. VBA code will use these descriptors to create the financial statements, and it will be much easier to have the user classify each line item manually instead of having VBA look at the trial balance to determine if anything new was added and how to classify it.

This quarterly task usually takes me several days to create and adjust so that the resulting financial statements can gain the stamp of approval from the auditors. My desire is to automate most of this process so at most I have to click on a few buttons and the majority of the financial statements will “magically” appear. The code also needs to make appropriate changes based on the year inputted and the quarter. For example, if “2” was entered as the quarter, then the income statement should have “Quarter Ended June 30” instead of “September 30” for example.

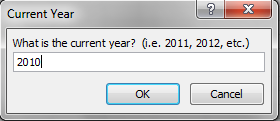
It took 2,676 lines of code to be able to implement my solution

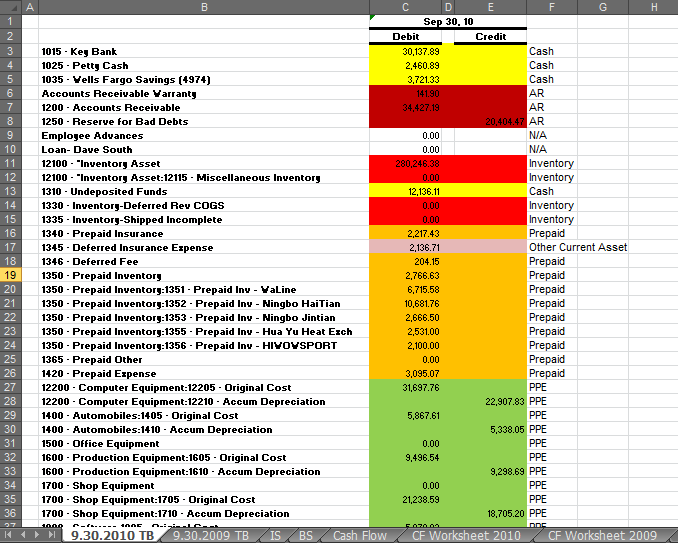
**Implementation Discussion**

The user will start out with a simple exported trial balance, and each line item will be manually classified as a specific type of asset, liability, expense, etc. The user will do this for the current year trial balance and the previous year’s trial balance. Currently, the company is still small so there are not a lot of line items to classify. The snapshot to the right shows part of the completed trial balance.

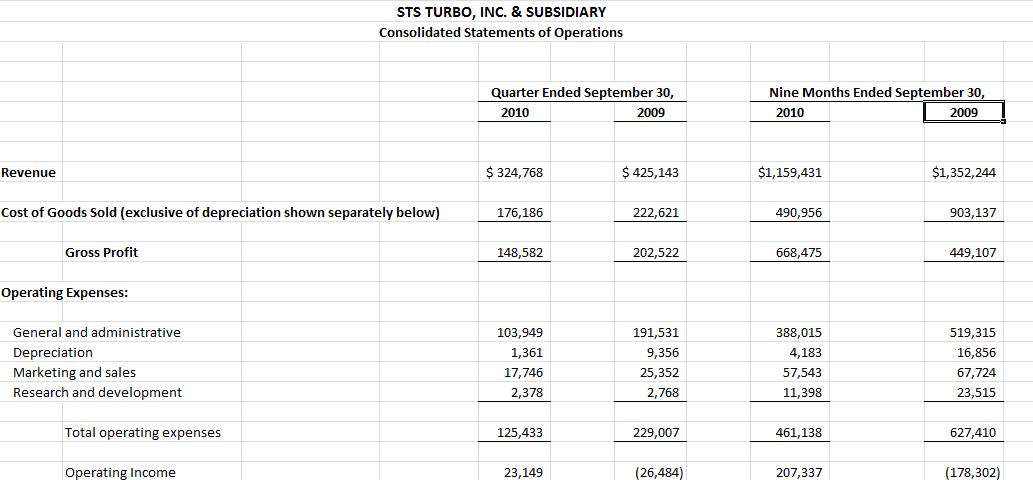
I have added a new ribbon tab called “Financial Statements” with three new buttons that link to different subprocedures within the module. The three new buttons are “Create Financials,” “Update Financials”, and “Extract Data.” The screen snapshot to the right shows these new buttons.

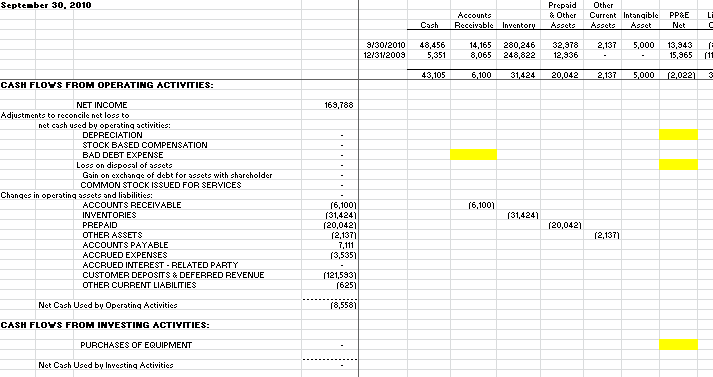
After the user has manually classified the different line items, they will come to this new tab and click on the “Create Financials” button. They will be prompted with two input boxes asking for the current year and the current quarter which the user must input for the code to work. The code has been made “user friendly” allowing the user to hit cancel or ok (without inputting any numbers) without showing any errors or taking them to an error within the code. Certain message boxes have been added to let the user know why the financials statements were not populated.

An input box has been shown here as an example. The VBA code will then search the current year trial balance and last year’s trial balance which has also been exported into the excel workbook looking at the manually entered classifications.

The code will place the user at the current year’s trial balance with several new worksheets created, formatted, and filled in. An example of the newly updated trial balance with its appropriate color coordination and the new tabs is shown to the side.

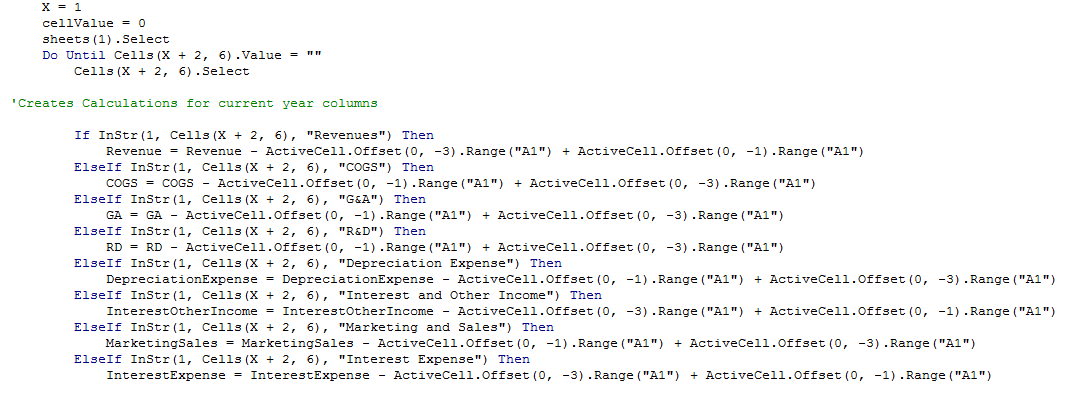
Each tab has been properly formatted and created to match previous financial statement formats. The income statement will be partially completed, the balance sheet will be finished, and the remaining tabs will not have any calculations as of yet.

The next step is to extract data from a previous set of financial statements. The user will select the “Extract Data” button from the Financial Statements tab. This will open a browse box to allow the user to select the previous quarter’s financial statements. The VBA code will then open the document selected, check to make sure certain spreadsheets are in the excel file, and it will then comb through the needed sheets taking the data it needs to complete the income statement (completed income statement shown to the right) and most of the cash flow statement. The code will then close the newly opened workbook.

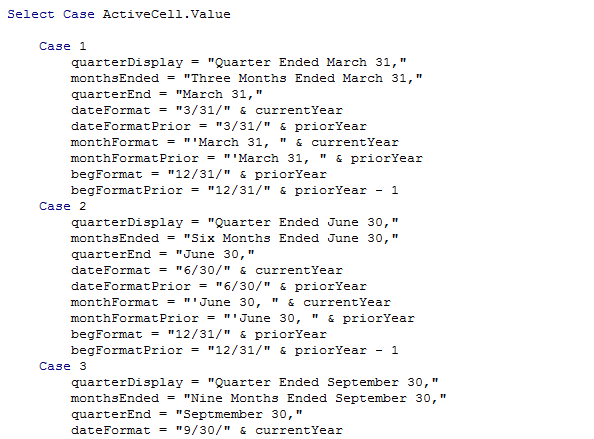
The only remaining items that the user must complete are the statement of equity (which is manual) and certain line items in the cash flow worksheets must be manually calculated and entered. These cells have been highlighted (shown to the right) to let the user know what they still must do to complete the cash flow statement. As they enter the data, the cash flow statement is automatically updated.

The final sub procedure copies the income statement, balance sheet, and cash flow statements, renames them, and then rounds each number to zero decimal places. These final statements will be what the investors see. They will enable the user to adjust the statements so that they are mathematically correct. Excel is wonderful, but rounding will alter the totals to be off a few dollars. These new statements will allow the user to adjust them to be mathematically correct while maintaining the integrity of the original calculations.

**Learning Discussion**

This project has allowed me to see that I am very capable of finding solutions to questions that I have with regards to VBA code. I do not consider myself a strong programmer, but I was able to come to each problem that I had and overcome them. The first major problem I encountered was how to use the manually entered classifications for the line items on the trial balance to create the income statement. I had troubles writing a code that would allow me to look up a string such as cash , then count the debit entries as positive amounts, the credit entries as negative amounts, and then sum them all together and place them on the appropriate line in the income statement or balance sheet. The resulting code is only a part of what I needed to be able to do the above.

Another major problem I encountered was how to appropriately capture the quarter entered in the input box and then use this to correctly populate any cells with reference to dates, years, quarters, etc. I tried to use if statements, but I gave up rather quickly and tried my hand at a select case statements. The code below is an excerpt of the written code.



Another problem I ran into was determining how to select a new spreadsheet when its name incorporated a year. For example, “CF Worksheet 2010” could not be hardcoded into the code. I needed the year part of the name to be updated with the year that was entered into the input box. I figured out how to do this through the use of a variable. I used code such as the following, *sheets("CF Worksheet " & currentYear)*, to help me navigate through the different spreadsheets. This would allow me to call on the worksheet that’s named CF Worksheet 2010 if 2010 had been entered into the input box mentioned above. These same variables will allow the code to gather data from previous quarterly statements because they are named accordingly and they will change from year to year - as will the number entered into the input box.