

Ethan Lindstrom's Personal Finance Program

Executive Summary

Over the last several years I have had accounts in 4 different banks and have tried 3-4 personal budget software programs. Each of them has had features I like but I have not been completely satisfied with any of the packages, or have run into incompatibility problems with data coming from one of my financial institutions. I decided that building my own personal finance program would alleviate those problems and give me control to build the features that I want.

My new personal finance program imports transactions, checks for overlap in the existing and incoming data, categorizes the incoming transactions, has a user form to allow categorization of items that were not categorized under existing rules, and sums the categories over the specified date range to compare against the budgeted amounts. It also pulls all data for the house I own and rent out to a separate sheet so I can keep track of my gains/losses for that property.

Personal Finance Program Implementation

All the functionality of my program is accessed on the ribbon:



An overview of each function is given below, with list headings being the name of the sub procedures called for each function:

Import New Data

1. ImportData

While I noticed that the majority of other personal finance or budget programs written by students in the past had the ability to automatically pull in data from their banks, I did not want to spend time adding in an automated import system that downloaded the data from my bank as I will be changing banks after I move in a month. Instead my program allows me to choose a CSV or XLS file that contains the transactions I want to import and pulls all the data from that file into an 'import' tab.

2. CategorizeData

This sub takes the imported data, prepares it, and then categorizes it using the subs below:

a. FormatData

Gets the imported data into the right format to be categorized and moved to the 'All Transactions' sheet that is the main repository for the data.

b. CheckDates

Looks for overlap between the existing data and incoming data. Both sets of data are first sorted so the most recent transaction appears first. Transactions in the overlap period are checked for duplication and duplicate transactions removed from the incoming data.

c. AddCategories

Checks each row of incoming data against categorization rules defined previously by user and assigns the transaction to the appropriate category. If the transaction is not categorized by current rules it is labeled uncategorized. This is what the Categories/Rules page looks like:

Categories	Rules
Car	DMV, JIFFY LUBE, AAA, AUTOZONE
Clothes	DRESSING YOUR TRUTH, KID TO KID, SISTER'S CLOSET
Donations	INDIEGOGO, PUBLIC RADIO
Eating Out	BAR-B-Q, BBQ, NOODLES, THAI, OLIVE GAR, PAPA JOHN, DENNY, IHOP
Education	BYU STORE, BRIGHAM YOUNG
Fast Food	JAMBA, TACO, YOGURT, MCDONALD, BURGER, WENDY, ARCTIC, LITTLE CAESARS, EINSTEIN BR, PITA, SUBS, CARL'S
Gas	CHEVRON, MAVERIK
Gifts	FTD
Groceries	SPROUTS, SWEET TOMATOES, CREAMERY, RANCHO, HARMONS, REAL FOOD, VITAMIN, MACEY, COSTCO, GROCERS, MILK, WINCO

3. MoveCatData

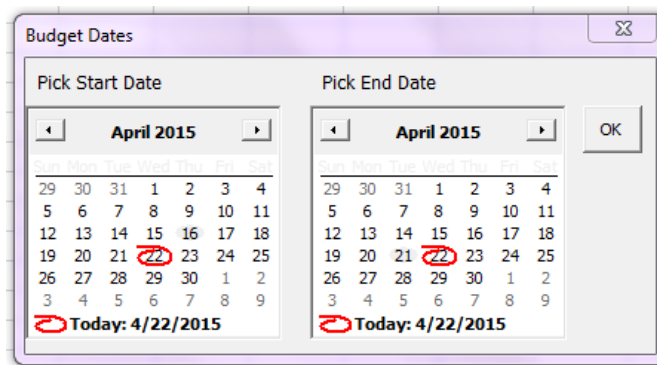
Inserts the now formatted and categorized data above the existing data in the "All Transactions" sheet.

4. UpdateBudgetActuals

Now that I have the most recent transactions imported I can see how I am doing vs. my budget.

a. FrmPickDate

This form allows the user to pick the date ranges that will be used to populate the actuals section of the budget.



Actuals are then computed (in VBA, not using a formula in the cell) for each category using the date range specified. A chart is available to show the budget vs actuals, but it just takes the output of my code and is not created or modified in any way by the code.

That is the end of my main procedure to import and categorize the data and populate the budget sheet and graph.

The program has a few other features:

Categorize Transactions

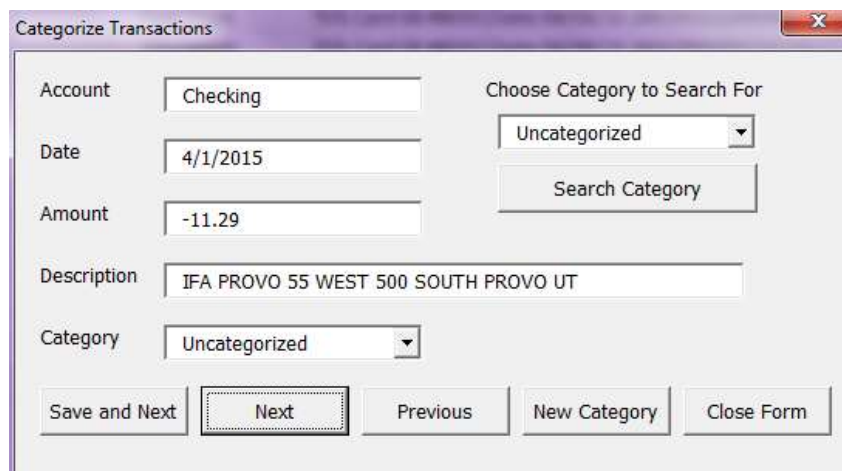
The auto categorization done during import leaves some transactions uncategorized. This is the opportunity for the user to categorize them using a form.

NOTE: Because I am still building this, there are LOTS of uncategorized transactions, however this is not due to the code, but because I haven't built in the rules yet (I don't view that as part of the assignment, since the code works I will add more later as I use the program).

1. FrmCategorizeTransactions

Displays all the relevant information for each transaction to allow the user to determine which category the transaction belongs to and save that category to the "All Transactions" sheet. New categories can be added from this form.

The form only displays transactions in the category chosen in the upper right hand corner, with uncategorized being the default. Thus the user can choose any category and quickly move through all transactions in just that category.



Re-Categorize Transactions

If you add budget categories or rules, this allows you to run all uncategorized transactions on the "All Transactions" sheet through the categorization algorithm again to see if they can now be categorized.

Update Budget Actuals

This button on the ribbon allows you to update the actuals section of the budget sheet. This allows you to update it in case you made a change to a budget category, changed budget categorization rules, or want to see a different date range (the first step in updating is to again choose the date range you want as shown above (frmPickDate)).

Update Rental Data

This button looks at the most recent transaction on the Rental Data sheet and then looks for newer transactions in "All Transactions" and inserts them into Rental Data sheet. These are used to populate a Yearly Income vs Costs table and graph.

Other Functionality

Finally, I browsed through many other projects and found a few tools that others had written and made note to copy them into my file so that I have them available in the future. The one that I added already does loan sensitivity and amortization calculations, which as I'm thinking of buying a car in the near future, I found very useful and want to have at my fingertips. Others I have bookmarked for future incorporation, but have not taken the time to add them in yet (stock screener, portfolio analysis, etc.).

Discussion of learning

My initial plan was to find an existing budgeting program written by a previous student and update and modify it to make it fit my needs and then add functionality. I found this to be very difficult. There were small parts I liked in several previous projects, but much of the code was structured in ways that made me believe it would be hard to modify it to fit my needs. So, taking the best ideas I saw as inspiration for what the functionality could be, I ended writing almost all my own code.

Several times in class Dr. Allen admonished us to write code that was more general, so it would be easier to recycle it. I also found this to be difficult, as it is hard to make my brain take a step back from the problem at hand and think about generic solutions, but I did end up with several sub procedures that are somewhat general that I call in multiple other subs. Twice (or more) I found myself copying a large section of a sub procedure into another sub to use with only minor modification. This is both good and bad, it means that I have gained the ability to remember similar code I have written and reuse it, but it also means that my code could still be improved and simplified by making that chunk into a general sub procedure.

I did not have any elements that I was unable to add because they didn't work. Everything that I planned to include I was able to get working. However, there are many things I would like to add (and will add over the next few weeks/months/year-I plan for this to be a continual work in progress as my needs evolve):

- Import data from web (once I decide which bank I'm going to use after I move)
- Option to add rules on manual categorization form (augment current ability to add categories)
- Option to re-categorize all transactions, not just uncategorized transactions
 - Didn't think of this until I was doing the write up
- Split budget up into two categories: fixed and variable
 - Have variable 'budget' numbers update with date range so I can see how we are tracking on variable expenses for the month so far, the current quarter, etc..
- Net worth page
 - Import data from all my accounts to show current net worth

Assistance

While I looked at many previous projects to gain inspiration, and used some code found on the internet for specific tasks I didn't know how to do (sources cited in each module), I did not receive assistance from anyone.