

Budget Tool

VBA Final Project

Trevor Paulsen

Executive Summary

Budget Tool is a very useful VBA application that allows a Zions Bank customer to download their bank statement and automatically as well as manually categorize their expenses into a budget spreadsheet. Once the bank statement has been downloaded and each expense has been categorized, the application automatically fills out a budget sheet with those expenses. This saves a lot of time for Zions Bank customers who traditionally have to download their statement and manually enter these categories by hand.

Implementation

The purpose of Budget Tool is to automatically populate a monthly transaction sheet from an online Zions Bank statement, automatically categorize statement entries as an expense category or income, give the user a useful way to categorized the remaining entries, and complete the “actual” expenses column of the budget sheet. To understand this project implementation, it is necessary to understand the components of a budget Workbook that the Budget Tool operates on.

The Budget Sheet

Below is an example of the budget sheet:

Cash Inflows			Cash Outflows					
	Estimated	Actual		Estimated	Actual			
Caryn's job	\$1,100	\$589.46	Tithing and Fast Offerings			March 2011		
Trevor's job	\$100		Tithing	\$250.00	\$162.80			
				\$250.00	\$162.80			
Other	\$0		House					
Interest	\$0		Rent	\$430.00	\$420.00			
Total	\$1,200	\$589.46	Electric Bill	\$100.00				
			Misc					
				\$530.00	\$420.00	Budget vs. Actual		
			Auto			Net Cash		
			Car Repairs			\$451.72		
			Car inspection/emission			-\$625.63		
			Oil Changes	\$20.00				
			Insurance (6 months)					
			Misc					
			Gasoline	\$60.00				
				\$80.00	\$0.00			
			Necessities					
			Groceries	\$350.00	\$122.74			
			Dining out	\$100.00	\$67.83			
			Dry Cleaning					
			Clothes					
				\$450.00	\$190.57			
			Total	\$1,310.00	\$773.37			

The sheet is organized into income and expense categories and includes the current budget period as well as columns for estimated expenses (filled out at the beginning of the period) and

Budget Tool then reads the entire user defined auto-categorization criteria from an xml file:

```
<?xml version="1.0"?>
<BudgetCategories>
  <Category name="Groceries" type="Expense">
    <Text value="MACEYS"></Text>
    <Text value="COSTCO WHS"></Text>
    <Text value="WAL-MART"></Text>
  </Category>
  <Category name="Rent" type="Expense">
    <Text value="CHECK">
      <MoreThan>400</MoreThan>
      <LessThan>500</LessThan>
    </Text>
  </Category>
  <Category name="Electric Bill" type="Expense">
    <Text value="PROVO CITY"></Text>
  </Category>
  <Category name="Gasoline" type="Expense">
    <Text value="COSTCO GAS"></Text>
    <Text value="MAVERIK">
      <MoreThan>20</MoreThan>
    </Text>
    <Text value="FLYING J"></Text>
  </Category>
  <Category name="Dining Out" type="Expense">
    <Text value="ARBY'S"></Text>
    <Text value="MARKETPLACE CAFE"></Text>
    <Text value="MAVERIK">
      <LessThan>10</LessThan>
    </Text>
    <Text value="PANDA EXPRESS"></Text>
    <Text value="WENDYS"></Text>
    <Text value="EL AZTECA"></Text>
  </Category>
</BudgetCategories>
```

The xml file allows the user to define as many predefined categories as he/she chooses. Each category has a "name" field, whether it's an expense or income category, text that it looks for in the vendor information, and any dollar amounts (less than or more than or both) that a transaction must be within to fall into that category.

Once all of the relevant transactions have been automatically classified, the user is presented with the following user form:

Expense Classification

Expenses Categorized Entries Count: 6 Subtotal: \$167.32 Average: \$27.89

Groceries	3/7/2011	\$20.52	P.O.S. PURCHASE COSTCO WHS 648 EAST 80 OREM UT
Rent	3/7/2011	\$56.54	P.O.S. PURCHASE MACEYS PRO 1400 N STAT PROVO UT
Dining Out	3/14/2011	\$20.49	P.O.S. PURCHASE COSTCO WHS 648 EAST 80 OREM UT
Dates	3/14/2011	\$25.19	P.O.S. PURCHASE MACEYS PRO 1400 N STAT PROVO UT
Netflix	3/21/2011	\$15.86	P.O.S. PURCHASE COSTCO WHS 648 EAST 80 OREM UT
Tithing	3/21/2011	\$28.72	P.O.S. PURCHASE MACEYS PRO 1400 N STAT PROVO UT

Add Delete Move Up Move Down Next >>

Income

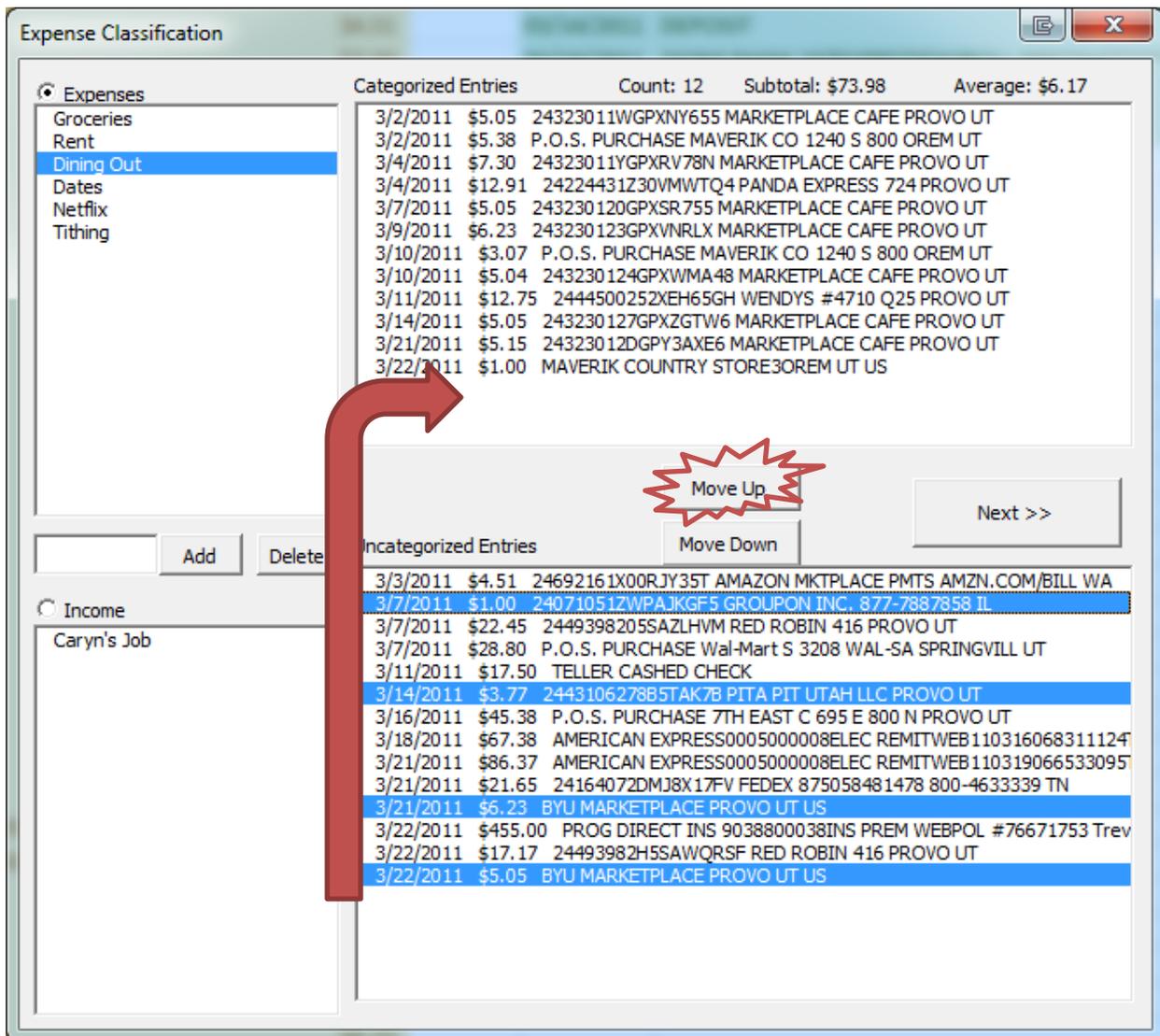
Caryn's Job	3/3/2011	\$4.51	24692161X00RJY35T AMAZON MKTPLACE PMTS AMZN.COM/BILL WA
	3/7/2011	\$1.00	24071051ZWPAJKGF5 GROUPON INC. 877-7887858 IL
	3/7/2011	\$22.45	2449398205SAZLHVM RED ROBIN 416 PROVO UT
	3/7/2011	\$28.80	P.O.S. PURCHASE Wal-Mart S 3208 WAL-SA SPRINGVILL UT
	3/11/2011	\$17.50	TELLER CASHED CHECK
	3/14/2011	\$3.77	2443106278B5TAK7B PITA PIT UTAH LLC PROVO UT
	3/16/2011	\$45.38	P.O.S. PURCHASE 7TH EAST C 695 E 800 N PROVO UT
	3/18/2011	\$67.38	AMERICAN EXPRESS0005000008ELEC REMITWEB110316068311124
	3/21/2011	\$86.37	AMERICAN EXPRESS0005000008ELEC REMITWEB110319066533095
	3/21/2011	\$21.65	24164072DMJ8X17FV FEDEX 875058481478 800-4633339 TN
	3/21/2011	\$6.23	BYU MARKETPLACE PROVO UT US
	3/22/2011	\$455.00	PROG DIRECT INS 9038800038INS PREM WEBPOL #76671753 Trev
	3/22/2011	\$17.17	24493982H5SAWQRSF RED ROBIN 416 PROVO UT
	3/22/2011	\$5.05	BYU MARKETPLACE PROVO UT US

The top left list box details each of the budget expense categories that have been created. The bottom left list box shows the income categories. The top right list box shows the categorized expenses that have been binned into the selected expense or income category. The bottom right list box shows all of the uncategorized expenses or uncategorized income (depending on whether the "Expenses" or "Income" radio buttons are selected). When a user selects an expense category, the "Expenses" radio button is automatically selected and any income categories that were previously selected become unselected and vice versa. Only one category may be selected at a time.

The user can add a new budget category using the "Add" button after typing in the name of the new category in the text box next to the "Add" button. Pressing the "Delete" button removes

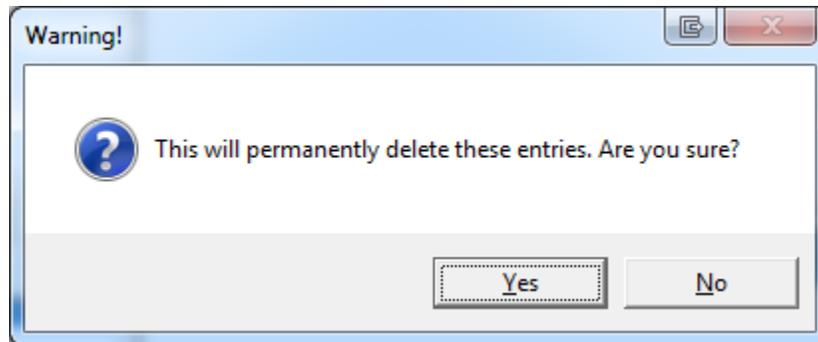
the selected category and places any categorized expenses belonging to that category into the “Uncategorized Entries” list box.

In order to move a bank entry into a category, the user can select as many uncategorized entries as he/she wants from the “Uncategorized Entries” box and click the “Move Up” button to move them into whichever category is currently selected. The count, subtotal, and average labels are all updated to reflect the information in the categorized entry box, and those entries are removed from the uncategorized box.



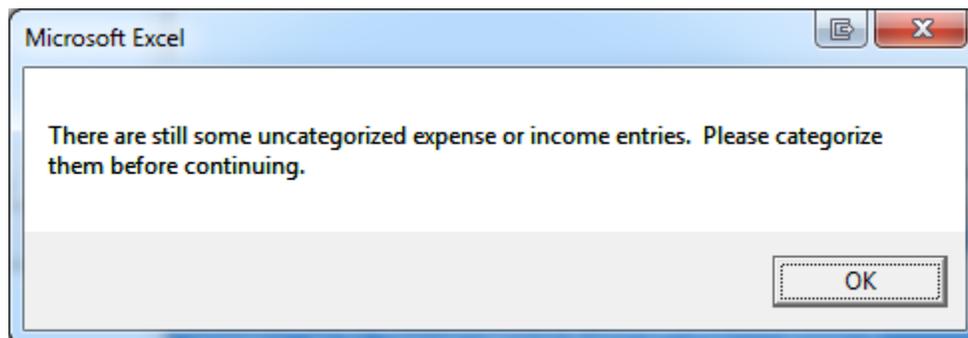
If the user wants to remove certain entries from a category, he/she can select as many as is wanted and click the “Move Down” button to move them from the categorized box into the uncategorized box.

If the user wants to permanently delete an entry without categorizing it, he/she can highlight either the categorized or uncategorized entries and just hit the delete key. This will trigger the following warning:



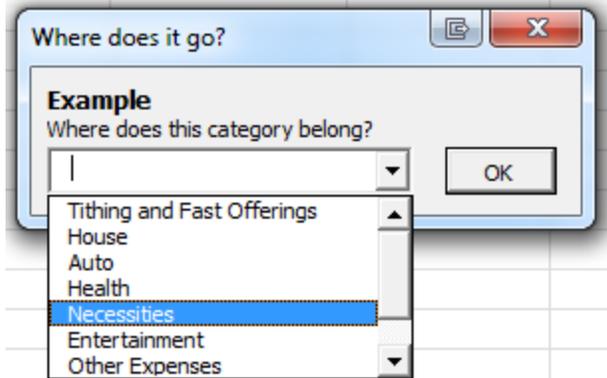
This warning helps the user not to accidentally delete transaction records that he/she doesn't mean to.

Once the user is finished placing all of the entries into various categories, he/she can push the next button to continue. If there are still uncategorized entries remaining, Budget Tool throws an error:



This prevents the user from accidentally forgetting to include all of their expenses in the budget. If all of the entries have been properly put into categories, the user can safely press "Next" and Budget Tool will enter each of the subtotals from each category into the budget sheet if that category is already listed on the budget sheet.

If a category is not listed on the budget sheet, Budget Tool will display the following:



This form allows a user to select which super-category that sub-category belongs to. For this sub-category "Example", the user has the choice of placing it in one of the automatically populated super-categories already found on the budget sheet. Once a super-category is selected, Budget Tool automatically inserts that sub-category into that super-category section on the budget sheet with its associated subtotal.

After all of these steps are completed, Budget Tool has done its job! This is a huge time saver if you're used to having to manually download and categorize all of your expenditures every single month.

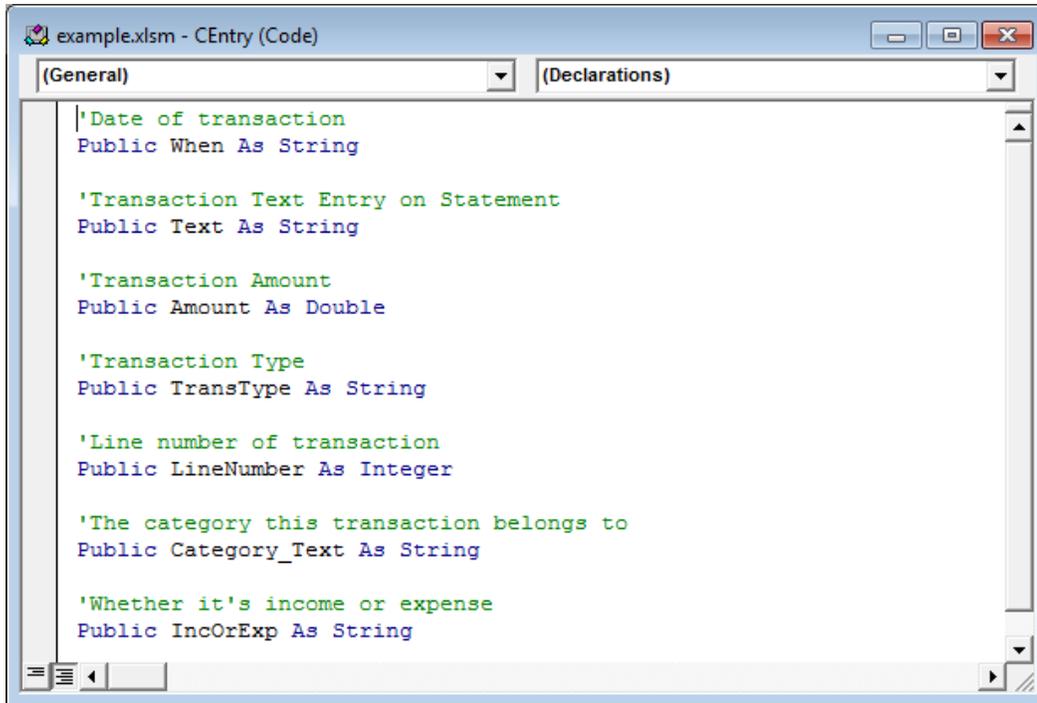
Learning and Conceptual Difficulties

When I started this project, I knew what I wanted it to do, but I didn't know how I would do it. I ran into several challenges while writing this solution:

The first problem I encountered was how to have Budget Tool automatically classify bank statement entries into categories that a user could easily define. At first I hard-coded all of the auto-categorization categories and criteria into the VBA script, but soon realized that no ordinary user (e.g. my wife) would be able to change that, so it wouldn't be very useful. While I was in my Web Development class, we were using an xml file to remember users for an ad-hoc membership database and the idea came to me that I could use an xml file for this project as well. Once I saw how easy it was to open an xml file with Excel, the rest was cake. I really liked how this feature turned out because it lets the user define as many or as few categories as he/she would like, but also classify according to vendor AND amount spent which is very useful.

Another problem I had to deal with was how to set up a data structure that would allow me to store each of the bank statement entries into a data structure that would be easily parsed. This was a big problem because since there are many potential budget categories, I couldn't simply

use the list boxes to store that information alone. Eventually I decided to create a Class Module that would store all of the information for each entry. This would allow me to create collections of entries that would be easily searchable and sortable:



```
example.xlsm - CEntry (Code)
(General) (Declarations)
'Date of transaction
Public When As String

'Transaction Text Entry on Statement
Public Text As String

'Transaction Amount
Public Amount As Double

'Transaction Type
Public TransType As String

'Line number of transaction
Public LineNumber As Integer

'The category this transaction belongs to
Public Category_Text As String

'Whether it's income or expense
Public IncOrExp As String
```

By only having to search through one collection of entries, I avoided having to deal with multiple multi-dimensional arrays that could have been a horrible nightmare to code and keep track of. Plus, by keeping all the entries in this collection, there was no need to sort any of the list boxes when something new was added.

A third hurdle I had to get over was actually downloading the data from Zions Bank website. This was a tricky feat because there were several pages of HTML to navigate through before I finally reached the data that I wanted. In order to make it work, I opened up Firefox with the Firebug extension to help me find the specific inner HTML that I was looking for and was then able to change the value to what I needed. The dates used in the budget spreadsheet are what I pull from to enter the dates I want my bank statement to cover:



This worked out great. The other problem was actually copying the data over. Luckily I found the following code to help me do that:

```
ie.ExecWB OLECMDID_SELECTALL, OLECMDEXECOPT_DONTPROMPTUSER  
ie.ExecWB OLECMDID_COPY, OLECMDEXECOPT_DODEFAULT
```

This code basically selects everything in the IE window and copies it to the clipboard. Since the bank statements are loaded as an HTML table, pasting them into Excel puts every entry into its own cell and after a little formatting, voila!

Conclusion

I was really happy with how this project turned out. It took me quite a while to code it up, but it works great. If I could add some features in the future, I would add a little more robustness to the login code. As it stands, the user must actually have Zions “remember” this computer when logging in outside of this VBA script for the login script to work. Also, there’s probably a few error checking things that a troublesome user might do to mess up the program that I didn’t really check for. I didn’t really mind though since I figured this was going to be most useful only to me and my wife for our personal budgets.