Explanation of transaction importation from Mountain America Credit Union’s website into a budget managed by Microsoft Excel

Trevor Hoth

I Sys 540 Section 1

Mountain America Credit Union to Budget Automation

Table of Contents

[Executive Summary 2](#_Toc290297041)

[Implementation Documentation 3](#_Toc290297042)

[Transaction Retrieval 3](#_Toc290297043)

[Retrieve data 3](#_Toc290297044)

[Format data for use 4](#_Toc290297045)

[Insert data into “Transactions” worksheet 4](#_Toc290297046)

[Transaction Display to User in Form 4](#_Toc290297047)

[Saving Formatted and Categorized Transaction 5](#_Toc290297048)

[Automated Transaction Formatting and Categorization 5](#_Toc290297049)

[Budget Amount Editing for the Current Month 5](#_Toc290297050)

[Conceptual Difficulties and Learning 6](#_Toc290297051)

[Write-up (Project Details) 7](#_Toc290297052)

[Transaction Retrieval 7](#_Toc290297053)

[Transaction Display to User in Form 8](#_Toc290297054)

[Saving Formatted and Categorized Transaction 9](#_Toc290297055)

[Automated Transaction Formatting and Categorization 9](#_Toc290297056)

[Budget Amount Editing for the Current Month 10](#_Toc290297057)

[Conclusion 11](#_Toc290297058)

# Executive Summary

For the past few years I have kept a budget in Microsoft Excel. This budget is a static document that I add to by either typing each transaction or logging into my credit union, Mountain America, copying the transactions that have been posted, pasting them into Excel, then formatting the text into the desired format. After years of doing this exact procedure I have gotten fairly fast but it is always a tedious task. I have been looking for a way to automate this process to increase efficiency.

This system which I have called “Update Finances” has been written to automate the above described process. Running the system will go to https://MACU.com and retrieve all transactions since the last import date up to today’s date. After automated formatting of the imported transactions, it will then present each transaction to the user for further formatting and categorization. The user may then save or delete the transaction. The following is a detailed description of how this system saves time and energy in the budgeting process.

# Implementation Documentation

In order to run this program a new tab in the Excel Ribbon has been added named “Update Finances”. This tab has a single button in it named “Update Finances.” Once clicking this button the program will start. There are essentially five parts to this program. These five parts facilitate the transaction manipulation and saving and the managing of the budget for the current month. The specific five parts are:

1. Transaction retrieval
2. Transaction display to user in form
3. Saving formatted and categorized transaction
4. Automated transaction formatting and categorization
5. Budget amount editing for the current month

The following is a brief description of the above five parts of the system, their roles in the overall task, and the processes in each of the four parts.

## Transaction Retrieval

Transaction retrieval is a three step process which consists of retrieving the data, formatting the data, and inserting the data into the “Transactions” worksheet for later use. This process reads the last import date and retrieves the transactions between that date and today’s date. Following are the three steps in this process.

### Retrieve data

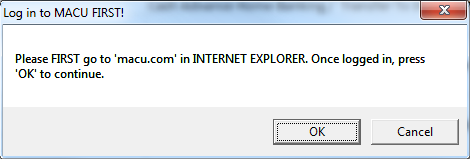


Figure 1

Figure 1 shows the first thing the user will see once clicking on the “Update Finances” button. The user will then go out to Internet Explorer, log in to Mountain America Credit Union, and click “OK”. (Note: It is important for the user to manually log in to the website since Mountain America’s website is unfriendly to automation.)

After clicking “OK” the program will manipulate the already open Internet Explorer to retrieve all transactions since the last time the program was run until today’s date. After retrieving the data, tomorrow’s date is written to the worksheet as the “Last Import Date”. To keep from duplicating transactions the program will use the day after the last day the program was run as the start date and today’s date as the end date.

### Format data for use

|  |  |
| --- | --- |
| Figure 2 | Figure 3 |

The raw imported data (example shown in Figure 2) is unformatted and unreadable to the program. The raw imported data will be automatically formatted in order for proper use (example shown in Figure 3).

### Insert data into “Transactions” worksheet

At this point a simple copy/paste is done to insert the newly formatted data after any existing transactions in the “Transactions” worksheet for use by the program. The worksheet that was created for the imported raw data is discarded and the program is ready to display the transaction for user input.

## Transaction Display to User in Form

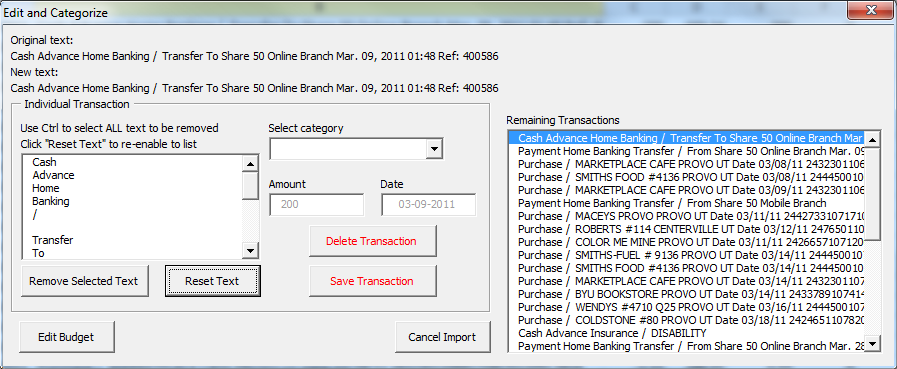


Figure 4

Once the transactions have been imported into Excel, formatted, and pasted into the proper location, the “Edit and Categorize” form (Figure 4) will appear and display all transactions. In order for this to happen the form will retrieve all transactions from the “Transactions” worksheet and start working through the transactions from the oldest transaction (formatting of oldest to newest done on import).

The imported transactions are broken up into six columns. The important columns are columns 1, 2, and 5. These columns contain all pertinent data to the transaction. The form will grab these columns and insert the contents into their proper locations: “Date”, “Original text”, and “Amount”.

## Saving Formatted and Categorized Transaction

In order to save the transaction the user must first assign it a category from the dropdown menu. The user may also choose to format the text in a special way (removing some pieces of the description of the transaction). When the user clicks the “Save Transaction” button the formatted transaction will be written out to the proper worksheet (according to the date of the transaction) and the next transaction will be loaded in to be formatted and saved.

Another aspect of the save button is the saving of the history of the transaction formatting. The program will record how the text was formatted and what category this transaction was given. This will facilitate for later time savings and program learning.

## Automated Transaction Formatting and Categorization

The history discussed in the “Saving Formatted and Categorized Transaction” section facilitates the automation of transaction formatting and categorization. When a transaction is read from the “Transactions” worksheet the program will first check to see if this transaction has been seen by the program before. If it has it will automatically use the formatting of the last seen similar transaction to format the text and categorize the transaction. This formatting is then displayed to the user.

Once the transaction is displayed, the user may choose to accept the proposed formatting or reformat the transaction according to the either proper or special circumstances of that transaction (e.g. Wendys was categorized as “Food” but for the current transaction should be categorized as “Dates”). This process saves the user, in the long run, hours of formatting and categorization which allows the user to see the final outcome of his/her expenses compared to the set budget more quickly.

## Budget Amount Editing for the Current Month

The “Edit Budget” button (see Figure 4) displays the budget editing form (Figure 5). This form will allow the user to edit the budget for the current month which allows for a dynamic budget from month to month. The budgeting portion of this program is the most important aspect since the original purpose of this file was to maintain a budget. This form is very important to the overall program.

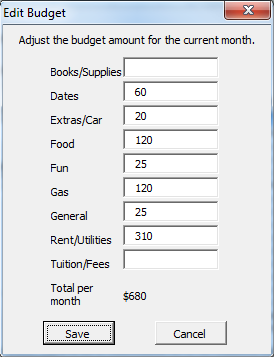


Figure 5

# Conceptual Difficulties and Learning

This project, of course, turned out to be a longer project than I expected. There were many stumbling blocks to the success of this project which resulted in much learning. Some stumbling blocks include the proper communication with and manipulation of the Mountain America website, displaying and hiding of forms, formatting of text going into and being retrieved from the worksheet, and proper functioning of the history and automatic formatting options.

One interesting thing I learned is that .value and .text are very different. I retrieved the .text of a cell one time and got “######”. The text of the cell was like this because the full date that was stored there was too wide for the column it was in, therefore the .text of the column was different than the .value. Also, by retrieving the .value of the column you were generally guaranteed to get the information that you wanted, not so with .text.

Displaying and hiding forms turned out to be a small stumbling block. I have a “Cancel” button on multiple of the forms in the program and the desired result after pressing “Cancel” is for the program to completely stop. Me.Hide does not stop the program. After some searching I found that the End command gave the desired result. I also found that, for my purposes, making the form non-modal was the best way to get the desired result out of this program of being able to close certain forms without stopping the program. If the form was modal and the red X was pressed then my program would shut down.

The proper use of the history tab was also challenging. I learned the power and drawbacks of using the replace command. In my program I eventually had to use the split command to make sure that I was only replacing the correct data. This took some time to figure out and learn.

There turned out to be only one thing that I couldn’t get to work. From what I could gather this obstacle was intentional. Mountain America’s website would apparently block some forms of automated manipulation. After having run this program multiple times on their website the program would no longer work, even though I could manipulate the website by hand just fine. In fact, my computer ran a virus scan and found a suspicious cookie from Mountain America, most likely preventing automated manipulation. The portion of the process that caused this error was the login portion. However, I found that after opening internet explorer and logging in, I was able to manipulate their website with no hassle. Therefore, in order to fix this, I went around the automated login portion of this program and required the user to log in to <https://MACU.com> in Internet Explorer first. This fixed the problem. Therefore, the automated login process was not included in the final program.

# Write-up (Project Details)

This program is both very involved and fairly straight forward. Some portions of the project took some ingenuity and creativity while other portions took thoughtful and meticulous planning. I am particularly proud of the history tab which will be explained later on. This write-up will go into moderately deep detail on how each of the five parts of this project runs namely:

1. Transaction retrieval
2. Transaction display to user in form
3. Saving formatted and categorized transaction
4. Automated transaction formatting and categorization
5. Budget amount editing for the current month

The process of transaction retrieval, categorization, and formatting starts when the user clicks the custom ribbon control entitled “Update Finances”.

## Transaction Retrieval

To retrieve the transactions I used the “agent” module that was given us in class. This module helped me manipulate Mountain America’s website to retrieve the desired data in the desired format. To use this process with Mountain America, the user must first open Internet Explorer and log in to Mountain America’s site. Automation of the login process proved problematic due to some sort of counter automation tool that they used.

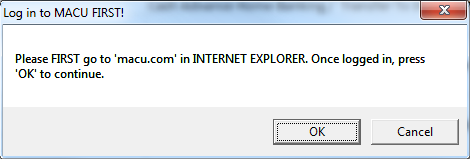


Figure 6

Once logged in to their website the user clicks “OK” on the dialog box that is displayed after starting the program (Figure 6). Clicking “OK” will start the “agent” module to copy the desired data into a new spreadsheet.

The desired data is determined by today’s date and the last time the program was run. The latter date is recorded in a tab named “Transaction History” as the day after the last time the program was run. This ensures no duplication of transactions in the budget. The data will be retrieved in ascending order (oldest to newest).

|  |  |
| --- | --- |
| Figure 7 | Figure 8 |

After the new, raw data is retrieved from Mountain America it must be formatted to be used by the program. The raw data comes in with extra columns and rows (see Figure 7) and must be reduced to one continuous table of transactions with no extra or non-transaction text (see Figure 8). This is done by first deleting the extra columns, then deleting the first 7 rows (unwanted text), then deleting the two rows after the current cell as long as the current cell is not blank. Once the current selected cell is blank the program will select the next 200 rows and delete them (to be sure all data on the worksheet is a single continuous table of transactions only).

This continuous table is then copied over to the “Transactions” worksheet where it will be used in the rest of the program. The new worksheet that was inserted to store the new, raw data from Mountain America is disposed to keep the workbook clean.

## Transaction Display to User in Form

After all the transactions are inserted into the “Transactions” worksheet the “Edit and Categorize” form will display. This form is populated from the “Transactions” worksheet by selecting the first cell in the second column of the first row (cell B1) and populating the text of each cell in that column into the “Remaining Transactions” list box. This box is only meant as a viewable box.

Once the “Remaining Transactions” list is populated the form then takes the first transactions (oldest) and populates the form with the appropriate information: date, text, and amount. The date is found in the first column, the text in the second, and the amount in the fifth. The date and amount are not allowed to be changed in the form.

The text is shown in two different ways. It is shown as the original text of the transaction (the way it was retrieved from Mountain America) and the “New text:” which is how the user has now formatted the text.

Formatting of the text is done by selecting ALL of the desired text to be removed from the list (using ctrl if necessary) then clicking “Remove Selected Text”. All Text must be selected at one time. This will remove the selected text from the list and populate the “New text:” area with the appropriate modified text. The text can be reset if a mistake is made.

In order to save the transaction a category must be selected first. The user will use the dropdown menu to select the appropriate category. These categories are stored in a hidden worksheet named “Categories”. This sheet will be queried by the “Edit and Categorize” form when initialized to add all the proper categories to the list. Once the category is selected the user may click the “Save Transaction” button and the transaction will be saved. If a category is not selected a message will appear instructing the user to select a category before saving.

## Saving Formatted and Categorized Transaction

Once the “Save Transaction” button is clicked the form will check to see if a worksheet exists for the month of the current transaction. If the worksheet does not exist the form will create a worksheet with the name being the month of the transaction in a two number format. This new worksheet is created from a hidden worksheet named “Template”. The template worksheet has all the formatting necessary to maintain the budget. The title of the template tab is formatted as a date in the “Month-Year” format so that, as the sheet is being created, the form can update the title to be the date of the current transaction. This will be displayed as “Month-Year”.

After a worksheet is either created or found in the workbook the transaction will be written to the first empty row (transaction date, new text, and amount in the proper column [determined by the category chosen]). The original text, category index, and text pieces (explained later) values are inserted into the top row (new row inserted to make space for new history) of the “Transaction History” worksheet.

The text pieces value is a comma-delimitated list consisting of the remaining indexes of the original text list after all undesired text is removed. In other words, if the transaction text were “Hi Doctor Gove” and I only wanted to use the first word and the last word the test pieces value would read “0,2,”. This facilitates the later predictive formatting of like transactions.

After the formatted and categorized transaction is saved the first row of the “Transactions” worksheet is deleted and the next transaction is populated into the form. The “Remaining Transactions” list is also updated with the remaining information in the “Transactions” worksheet.

## Automated Transaction Formatting and Categorization

In order to automate transaction formatting and categorization a “Transaction History” worksheet needed to be created. This worksheet contains the original text, the category index, and text pieces (as previously described) of each transaction. When a new transaction is about to be displayed on the “Edit and Categorize” form the first word after the “/” in the original transaction text is first searched for in the “Transaction History” worksheet. If found, the form will format the current transaction’s text the same way the user formatted the historical transaction’s text by using the comma-delimitated text pieces text.

The text pieces process is simple. First, the form will separate the original text into separate pieces using a space as a delimiter (this is the same process which populates the original text into the text editing list box for formatting). The form will then remove all pieces of text that match the indexes found in the text pieces text. This process is what I feel is the most creative of this program simply because I did not search it online or hear it in class, this process was 100% home spun or inspiration.

After the text is reduced to the before-used format, the new text, new text list, and predicted category are written to the form for user approval or adjustment. In order to adjust the user must click the “Reset Text” button unless the only adjustment is the category.

Through this process the transaction could then be saved with one click as opposed to formatting each new transaction. This saves time and reduces stress.

## Budget Amount Editing for the Current Month

Since this project was undergone for the sake of making a budget easier to record and keep, editing the current month’s budget was specifically written into this application. This functionality only allows for current month budget editing.

|  |  |
| --- | --- |
| Figure | Figure 10 |

To edit the current month’s budget the user would click on the “Edit Budget” button in the “Edit and Categorize” form. This will bring up a new form named “Edit Budget” (see Figure 9). This form is populated from the “Budget Amounts” row of the current month’s worksheet (see Figure 10). As values are changed the form will check to see if the values entered are numeric. If they are non-numeric the form will clear that text box. Also, as the amounts are updated the total amount at the bottom will update automatically, always returning the current sum.

Clicking “Save” will write out the modified amounts to the current month’s worksheet to reflect the new budget for that month. Clicking “Cancel” will leave all budget amounts the same.

# Conclusion

This project has been very insightful and challenging. I have learned many things while doing this project that I hope will help me in my later jobs, careers, and home use. Excel is a very powerful program that has almost limitless possibilities for projects and automation. This project has taught me that it is possible to automate even the most complex (or so I felt) processes in a fairly short amount of time, all you need is a little bit of creativity.