

# Spreadsheet Automation

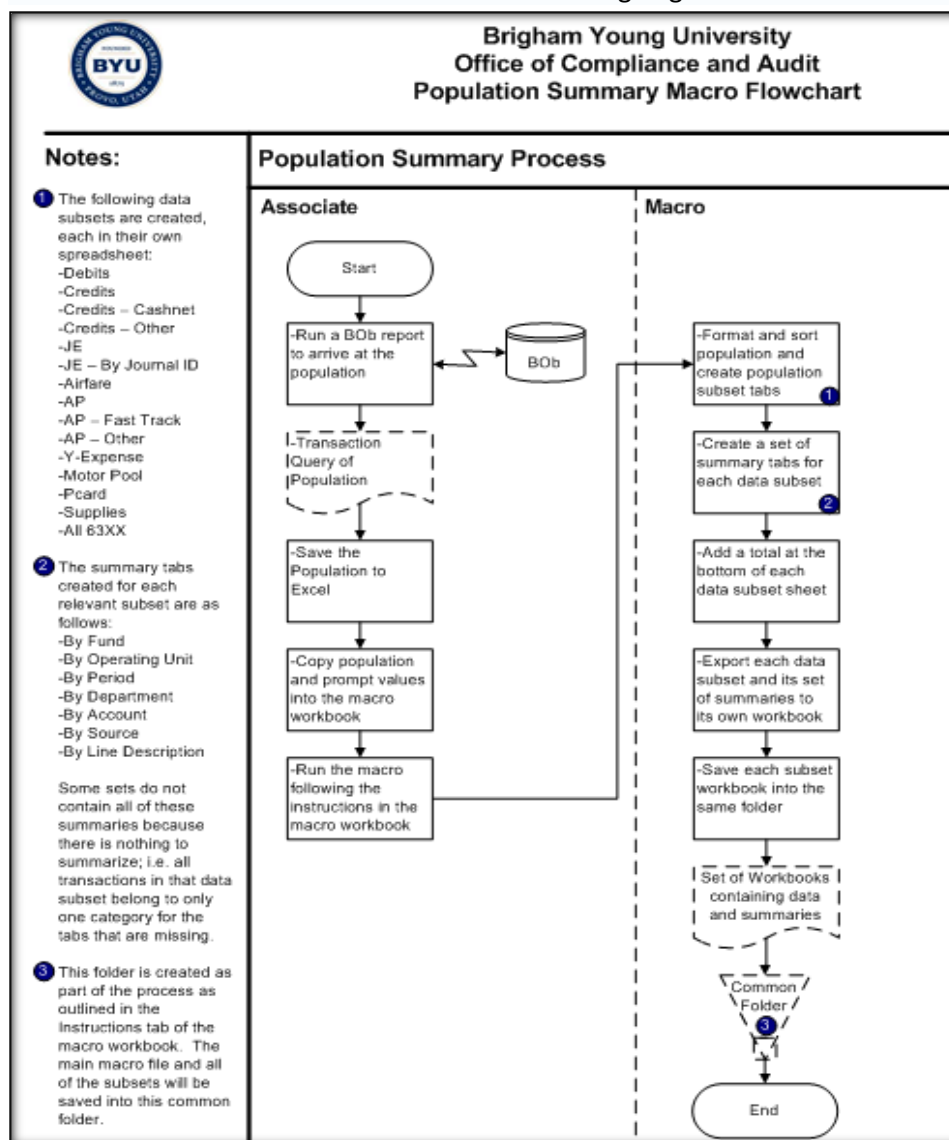
Jacob Stonehocker

Final Project Write-up

12/6/2012

## Executive Summary

I currently work for the BYU Office of Compliance and Audit and have built a solution to meet a business need within the office. The need for my solution arose because of changes being made to the overall audit strategy within our office. Audits will now be done using broader populations, which necessitates a way to dissect the population and analyze it across several different categories of transactions based on several different attributes. There will be an ongoing need for an extensive analysis of the



population for each audit; for this reason, I was asked to create a macro that will automate the otherwise long, arduous and divergent process. My solution is a set of templates (one for each category of transactions), each one housing a macro, into which one can drop the current year population following the instructions in the template, run the macros, and within minutes have a set of workbooks in a common folder that contain all of the desired analyses, complete with formatted summary tables and graphs. This will save our department many hours of student employee labor time and will save the full-time staff from having

to explain what analysis they want to each new student employee that is given the task. The flowchart included here outlines the process of using my solution to create the desired analyses.

## Implementation Documentation

I will describe how one of the macro templates works; all of the templates are very similar, with minor tweaks for each different data set. The following is a screen shot of the template with instructions for the user for analyzing a population of general expenses for a given college:

Account 6XXX Population Summary Template - Microsoft Excel non-commercial use

File Home Insert Page Layout Formulas Data Review View Developer Add-Ins

H5

1 **Name of College**

2 **Series 6XXX Population Summary**

3 **Scope Period**

4 Created by: Jacob Stonehocker

5 Date Created: 10/25/2012

6

7 **Purpose**

8 The purpose of this workbook is to automate the summary of 6XXX series transactions on several different factors in order to get a feel for the makeup of the population.

9

10 **Instructions**

11 1 Create a new folder in the appropriate location named "Series 6XXX Summaries"

12 2 Save this file as "Name of College - Series 6XXX Population Summary" in the folder created in step 1.

13 3 Run a BOB report for the college under review and save it as an Excel file.

14 a) use "Transactions - Optional Prompts Query". The query values to be used are as follows:

15

Field	Value
Beginning Journal Date	Beginning of the Scope Period
Ending Journal Date	End of the Scope Period
Dept ID	4-digit Dept ID of the college
Account ID (Start)	6000
Account ID (End)	6999

22

23 4 Copy the "AddlDetail" tab into this workbook.

24 **Note:** to copy a worksheet tab, right click on the tab and select "Move or Copy". A box will appear. Select the check box at the bottom left that says "Create a Copy".

25 Then use the dropdown to select this Summary workbook to copy the worksheet into. Select "(move to end)" and click OK. The tab will now appear in the summary

26 workbook to run the macros on.

27 5 Delete the last row of the data containing the total. Any additional values in the worksheet may cause the summaries to be incorrect.

28 **Note:** The "Replace WS Totals" macro will replace/add the total to the bottom of each data tab.

29 6 Name the tab where the data is located "All Transactions".

30 7 Copy the "Prompt Values" tab into this workbook.

31 8 Run the following macros in order (click each button only once). The last macro will export each subset of tabs to its own workbook within the folder created in step 1.

32 **Note:** the only tabs in this workbook before the macros are run should be "Instructions", "All Transactions", and "Prompt Values".

33 **Note:** the first two macros may take up to 10-20 minutes each to run, depending on the size of the population.

34

35 Create Data Tabs

36

37 Create Summary Tabs

38

39 Replace WS Totals

40

41 Export Tabs

42

There are several steps in the instructions of the template that could be written into the macro, but are not, for different practical reasons. For example, step 5, delete the total row. This could be automated, but is not, because it causes problems if you need to re-run some macros, but not all of them. If this was automated, and you needed to make an adjustment on one sheet, you would have to delete all of the output spreadsheets and start over. Also, the macro is split up into four buttons instead of just one. These and other items that could be automated or combined and are not were done purposely that way to add flexibility to the solution. The formatting of the template worksheet follows the format of other templates used by our department. This includes the command buttons being on the spreadsheet instead of in the ribbon; it is done for consistency with other templates and it flowed better that way because of the way it is set up.

The function of the macros is as follows:

#### **Create Data Tabs macro:**

This macro creates an “absolute value of amount” column in the population worksheet and then sorts the population in descending order by absolute value of amount. This is done so that amounts that net out to zero will be right next to each other in the data and so that the largest transactions, which are the most important, are at the top. For the general expenses analysis, this macro then creates the following data subset worksheets:

"Debits", "Credits", "Credits - Cashnet", "Credits - Other", "Motor Pool", "AP", "AP - FastTrack", "AP - Other", "Pcard", "Air Fare", "JE", "JE - By Journal ID", "Supplies", "Y-Expense", "All 63XX".

These data subsets are each created by copying the population worksheet, renaming it, and then deleting all of the rows that do not meet the criteria of that particular data subset. Screen updating is turned off for this and the other macros, which saves a considerable amount of time in the execution of the code.

#### **Create Summary Tabs macro:**

This macro, for each data subset that will be analyzed, creates the following summary tabs:

"By Fund" & ref, "By Operating Unit" & ref, "By Period" & ref, "By Account" & ref, "By Dept" & ref, "By Line Descr" & ref, "By Source" & ref,

Where “ref” = the name of the data subset to be analyzed, i.e. “By Fund – Debits” would be a summary worksheet by fund based on the data subset of debit (positive) amounts only.

After creating each tab, the macro copies and pastes the column needed for the summary row headers from the data subset worksheet into the summary worksheet, removes duplicates, and creates the column headers (\$ in population, % of population, # in population, % in population, and average \$ in population). The macro then populates the summary table using the “countif” and “sumif” functions on the relevant data subset. This is done for each summary worksheet for each data subset to be analyzed.

After creating the summary table, the macro calls other macros that format the data tables (with cell borders, coloring every other row in the table, etc.) and create the graphs desired for each summary.

The macro does not create all of the summary tabs for some of the data subsets because it doesn't make sense to do so. For example, all AP transactions have the same source code, so it wouldn't be meaningful to summarize that data set by source code. When this is the case, the tab is left blank.











At the end of this process, there are roughly 60 worksheets in the workbook, which consists of the data subsets and a set of summary spreadsheets for each data subset.

#### **Replace Worksheet Totals macro:**

This macro simply adds a total on amount to the bottom of each data tab, the original population as well as each data subset. It also creates a bold "TOTAL" row heading and formats the total cell with a "top and double bottom" border.

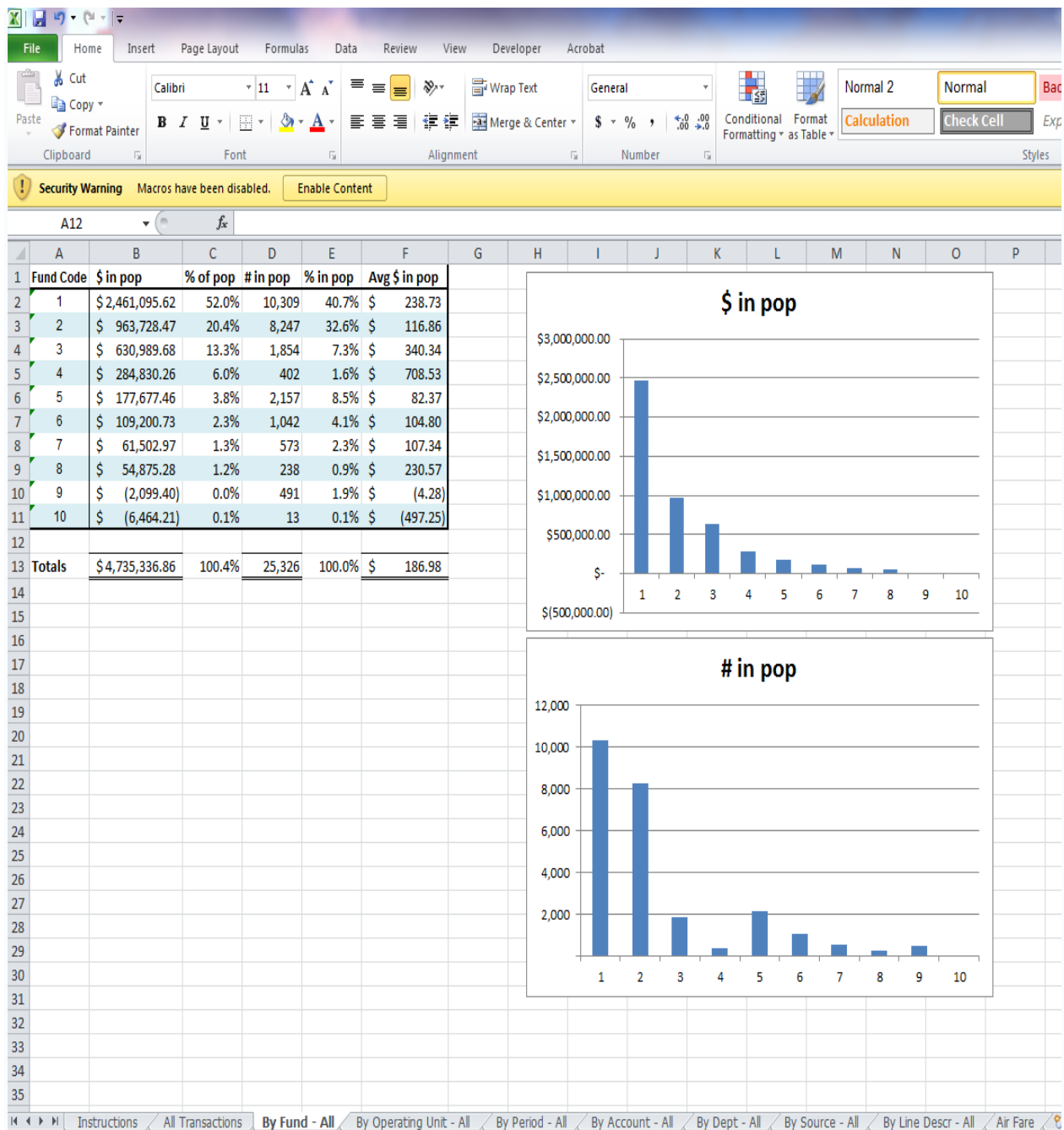
#### **Export Tabs macro:**

This macro exports each data subset and its related summary tabs into a new workbook and saves that workbook in the same location as the template workbook. When finished, the output is as follows:

Name	Date modified
 Account 6XXX Population Summary.xlsm	10/30/2012 5
 All 63XX Summary.xlsx	10/30/2012 5
 AP Summary.xlsx	10/30/2012 5
 Credits Summary.xlsx	10/30/2012 5
 Debits Summary.xlsx	10/30/2012 5
 JE Summary.xlsx	10/30/2012 5
 Motor Pool Summary.xlsx	10/30/2012 5
 Pcard Summary.xlsx	10/30/2012 5
 Supplies Summary.xlsx	10/30/2012 5
 Y-Expense Summary.xlsx	10/30/2012 5

To clarify, each of the workbooks shown here contains a data subset and the relevant summaries on that data subset. For example, the worksheets in the workbook "All 63XX Summary" would be as follows:

"All 63XX" (data subset), "By Fund – All 63XX", "By Operating Unit– All 63XX", "By Period– All 63XX", "By Account– All 63XX", "By Dept– All 63XX", "By Line Descr– All 63XX", and "By Source– All 63XX" (summaries as shown below). The screen shot below is taken from the "Account 63XX Population Summary", but is similar to the "All 63XX Summary" and the other workbooks shown above.



A few of the data subsets that are created are for information only and do not have summaries run on them. These data subsets are saved with the data subsets that they relate to. For example, the data subset “AP – FastTrack” is not summarized; it is just included in the “AP Summary” workbook.

## Learning and Conceptual Difficulties

One of the conceptual difficulties that I encountered during this project was changing the code to accept variables as inputs in order to make the same code work for different data sets. When I was given the project, I was asked to analyze travel transactions. After I finished that, I was asked to do the same thing

for the other categories of transactions. Rather than copy the code over and over and tweak it for every data subset, I wanted to make the code more dynamic so that I could just pass in variables to the code for each data subset and it would work without any alterations. I was able to figure out how to make that work, and in so doing learned a lot more about how to work with variables that are passed in to a sub procedure. To illustrate this point, I include below a sub procedure in which I used this concept that I learned. The variable “sheetSuffix” is the name of the data set being analyzed.

```
Sub exportTabs(sheetSuffix As String)
Application.ScreenUpdating = False
    Sheets(Array(sheetSuffix, "By Fund - " & sheetSuffix, "By Operating Unit - " & sheetSuffix, _
        "By Dept - " & sheetSuffix, "By Period - " & sheetSuffix, "By Account - " & sheetSuffix, _
        "By Source - " & sheetSuffix, "By Line Descr - " & sheetSuffix)).Move
    ActiveWorkbook.SaveAs Filename:=ThisWorkbook.Path & "\" & sheetSuffix & " Summary" _
        & ".xlsx", FileFormat:= xlOpenXMLWorkbook, CreateBackup:=False
    ActiveWorkbook.Close
End Sub
```

The code above illustrates another concept that I learned, that of moving worksheets into a new workbook, naming, saving, and closing the new workbook, and returning to the main workbook to repeat the process, all without any user input. We learned some of these concepts in class, but I really learned to apply them in these instances. I also learned on my own how to write the creation of graphs in VBA. I originally just gave instructions to the user of which graphs to use on each summary spreadsheet, but as the project got larger I decided to automate that part of it, and now the macro creates the desired graphs for each summary spreadsheet, which vary based on summary key (fund, source, etc.).

I thought about having the macro go out to the website to run the population data query and bring it in automatically, but the query takes a while to run, especially for bigger populations, and the user interface would have been difficult to manipulate through VBA. It would have taken quite a while to write the code for something that would save the user very little, if any, time. For these and other reasons I felt like the solution would be more robust, simple, and practical to start with the data in excel.

## Assistance

I did not receive substantial help from anyone on this project. I did turn to the internet from time to time to help me figure out how to write certain pieces of code, but I did not consult with anyone. My biggest sources of help were the Macro Recorder and the internet.