

Executive Summary

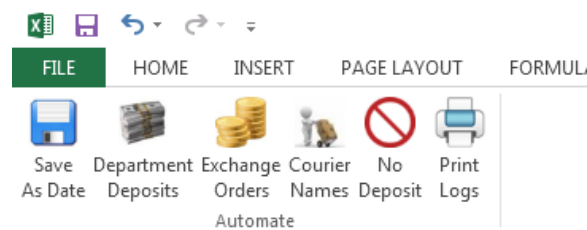
The business I did my project for was BYU Treasury Services, where I currently work. In order to protect confidential information, I will intentionally be somewhat vague in my description of what the business does. BYU Treasury Services provides deposit processing and change delivery services for many departments around campus. These departments email us their change orders every morning, and they also enter their deposit information into our online cashiering system. Every morning, we spend the better part of an hour getting that information from these email attachments and our online system and consolidating that information onto our spreadsheet by hand. This is a long and tedious process, and one that I decided I could make easier through automation.

The system I built consists of six different procedures. There are two main procedures. The first signs onto our online cashiering system, imports some data into Excel, sorts it, and puts it into the appropriate spots on our spreadsheet. The other takes all of the email attachments we get, prints them, imports the data from them into our spreadsheets, and closes them. The other four procedures automate simpler tasks, such as saving the template in the correct folder with the current date as the name, selecting from a dropdown list which departments have no deposit that day, and printing out the several sheets from the workbook at once.

Because of this project, the amount of time that we spend filling in our spreadsheet every morning has decreased significantly. What used to take more than 30 minutes now just requires clicking a few buttons.

Implementation

My project had seven main parts, six of them corresponding to one of the buttons seen in my customized ribbon below. I will go through each of these in the order they are seen here.

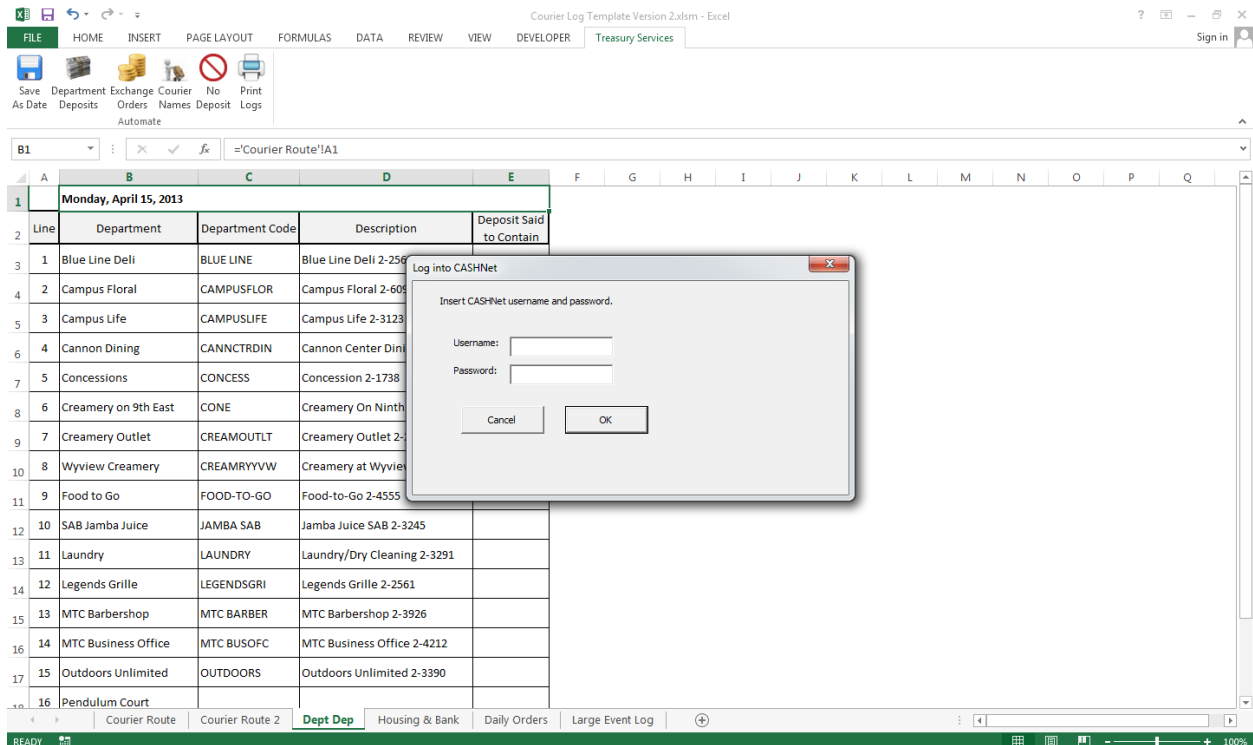


Save As Date

The Save As Date button does just what it sounds like. This was a simple subroutine consisting of just a few lines of code. Every morning we open the template and save it in the archive folder with the current date as the name. If the current name of the workbook is 'Courier Log Template,' this button will save it as the current date (in the format m-dd-yy) in the archive folder. If the workbook is already saved as the current day's date, the button will simply save the workbook. This simple button saves a few seconds of searching for the correct folder and typing the name.

Department Deposits

The next button, Department Deposits, was probably the most difficult one to do. It uses the agent class that Dr. Allen gave us to log onto our online cashiering system, cashnet.com, and get the deposit amount for every department that we service. It uses a userform to ask for a username and password, as shown below.



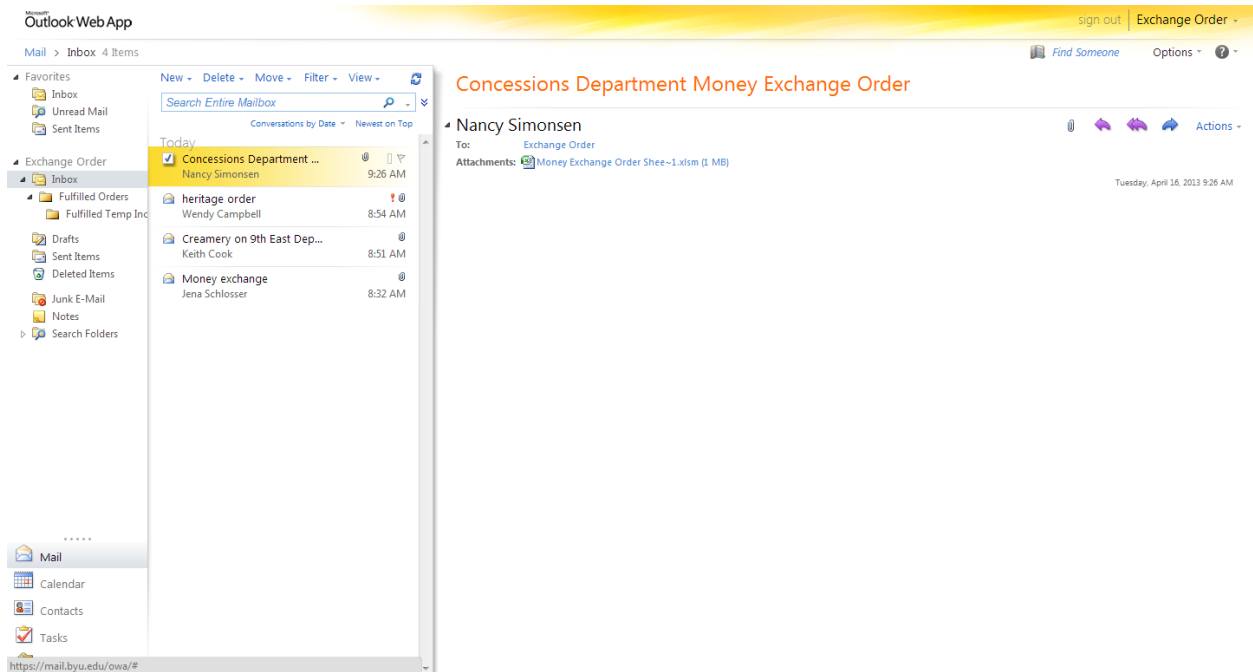
After it logs in, it downloads the entire page into a new worksheet. It then searches in that worksheet for the text that begins the table I am looking for. After finding the correct table, it uses a nested for loop to match the name of the department from the website with its name on our spreadsheet. When it finds a match, it enters the deposit amount in the correct column (column E above).

The most difficult part of this subprocedure was getting it to change the name of the department, in column B above. When a department has more than one deposit, we note that in parentheses after the name. For example, if Campus Floral had two deposits one day, we would change their name on our spreadsheet for that day to "Campus Floral (2)." My program does this automatically by concatenating the "(2)" to the end of the name. However, if I ran the program multiple times (since departments enter their deposits throughout the morning), it would end up looking like this: "Campus Floral (2) (2)." To fix this, I had to look at the name first and make sure it didn't end with ")". If it did, I changed the number within the parentheses instead of adding more parentheses to the end.

Exchange Orders

The next button, Exchange Orders, was the other big part of my project. I struggled with this one for a while, trying to automate the whole thing. We use the Outlook Web App at mail.byu.edu to receive emails that have departments' change orders attached. We download these attachments, copy their change order over to our spreadsheet, and print out their order. This used to be the most tedious and

time-consuming process of our job every morning. I hit a roadblock while trying to automate the process of downloading each attachment. Using the agent class, I was able to login and download the attachment of the email that was showing. But I couldn't figure out how to automatically click down to the next email and download its attachment (see image below).



I was also having an issue regarding the box that pops up when you click an attachment and asks if you want to open or save the file. It wasn't displaying correctly when using Windows XP, which is what we use on my work computer.

After trying to fix this problem for a while, I decided that it probably wasn't in the best interest of the company to fix it. If I were to automate the downloading of email attachments, it would eliminate the need to look at the emails that were sent to us. And those emails will sometimes contain important messages along with the attachments, which would go unread if I were to automate this process. After discussing this issue with my boss, I came up with a partial solution instead.

The Exchange Orders button now requires that you first download all of the change orders that you want to copy over and have those workbooks open. I have included a blank change order with my project submission so you can see how it works. A blank change order looks like this:

Money Exchange Order Sheet ver 2.xlsm - Excel

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DEVELOPER

Clipboard Font Alignment Number Styles

H1 : X ✓ fx =B5

A B C D E F G

1 **Money Exchange Order Form** Brigham Young University
Treasury Services

2 version 2

3

4

5 **Date**

6 **Ordered By**

7 **Location/Department** (Select From List)

8

Currency Needed	Denomination	Bills Requested	Amount Requested
Twenties	\$20		\$0.00
Tens	\$10		\$0.00
Fives	\$5		\$0.00
Ones	\$1		\$0.00
Total currency requested			\$0.00

Coin Needed	Roll Value	Rolls Requested	Amount Requested
Dollars (Vending)	\$25		\$0.00
Half-Dollars (Concessions)	\$10		\$0.00
Quarters	\$10		\$0.00
Dimes	\$5		\$0.00
Nickels	\$2		\$0.00
Pennies	\$1		\$0.00
Total coin requested			\$0.00

Cash Order Sheet

READY

The button loops through each open workbook. If the workbook is a change order, it will copy the necessary information, find the department name on our spreadsheet, and paste that information onto its line. It will then print out the order and close the workbook. Before it starts this process, a message box will pop up that warns the user that every open change order will be printed, and gives them an option to cancel it.

Courier Names

This button is a simple procedure that brings up a userform and asks the user to enter the names of the two couriers for the day.

Courier Log Template Version 2.xlsm - Excel

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DEVELOPER Treasury Services

Save As Date Deposits Department Exchange Orders Courier Names No Deposit Logs Automate Print

A1 : X ✓ fx =TODAY()

1 Tuesday, April 16, 2013 Courier: Person A & Person B

2	Department	Change Fund Temporary	Dept Initials	Exchange Order	Dept Initials	Deposit Said to Contain	Bag # (last 4 digits)	Time	Dept Initials	Department Signature	Courier Initials	T.S. Initials	T.S. Initials
3	Creamery on 9th East												
4	Heritage Housing												
5	Helaman Housing												
6	Cannon Dining												
7	Wyview Creamery												
8	Wyview Housing												
9	Outdoors Unlimited												
10	Concessions												
11	Marriott Center (check for deposit every day)												
12	Continuing Education (check for deposit every day)												
13	Wymount Housing												
14	Laundry												
15	Print and Mail Services (check for deposit every day)												
16	Vending												
17	Campus Floral												

Enter Names of Couriers

Enter the names of the couriers who will be doing the routes today.

Courier 1: Person A

Courier 2: Person B

Cancel OK

Courier Route Courier Route 2 Dept Dep Housing & Bank Daily Orders Large Event Log

READY

It then puts those names in the appropriate spots in our workbook (on the sheet above, the names are in the top right corner of the sheet).

No Deposit

This is another simple procedure that is useful when a department does not have a deposit on a given day. They will call or email us to let us know, and we need to make a note of it in our workbook. So the user simply clicks this button and selects the department from a drop-down box. The program will then enter "none" into the appropriate spots in our log.

Courier Log Template Version 2.xlsm - Excel

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DEVELOPER Treasury Services

Save As Date Deposits Department Exchange Orders Courier Names No Deposit Logs Automate Print

D4 : X ✓ fx

1 Date: Tuesday, April 16, 2013

2	Line	Department	Department Code	Deposit Said to Contain
3	Housing Deposits			
4	1	Campus Accomodations	CAMPUS ACC	
5	2	Helaman Housing	HELAMAN	
6	3	Heritage Housing	HERITAGE	
7	4	Wymount Housing	WYMOUNT	
8	5	Wyview Housing	WYVIEW	
9	Direct Bank Deposits			
10	1	Marriott Center		
11	2	Print and Mail Services		
12	3	Continuing Education		
13				

No Deposit Today

Select the department that has no deposit today.

Campus Accomodations

Cancel OK

Print Logs

The final button is another simple one. When we are done filling out the log every morning, we print out three of the worksheets. This button simply prints out those three sheets with one click instead of several.

Missing Change Orders

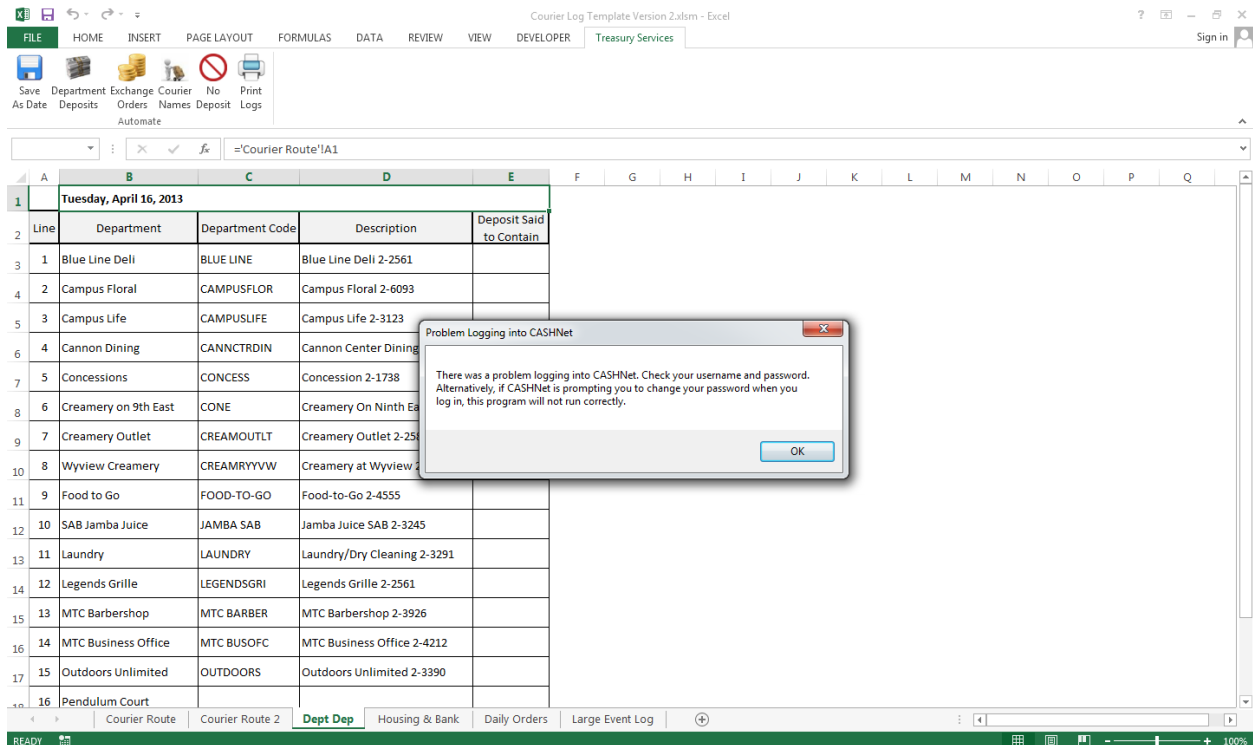
The seventh part of my project does not have a button, because it runs automatically. There are five departments that, with very few exceptions, order change from us every day. Occasionally, one of these departments forgets to send in their order, or there is a technical problem that prevents the email from reaching us. Sometimes we wouldn't notice that we hadn't received an order from one of these departments, and they wouldn't get the change they wanted for the day. To enhance our level of customer service, I wrote a subprocedure that runs automatically five minutes before we start our routes. If one of these five departments doesn't have an order on our log at that time, a message box pops up and notifies us. Otherwise, nothing happens. Now we can call the department and remind them that we haven't gotten their order yet.

Difficulties and Learning

I learned a lot by doing this project. The biggest difficulty I faced was the one I described in the *Exchange Orders* section above, trying to use the agent class to automate Outlook Web App. I got some help from Dr. Allen with the problem, but as I explained above, I ended up deciding it was best to not automate that particular part of the process. (This was the only help I received on the project.)

Another area of difficulty that turned into a great source of learning was handling errors. Error handling was something that we didn't discuss in much detail in class, so I had to learn a lot of it on my own while doing this project. I am happy with what I learned and what I was able to accomplish in this area. I was also lucky to be able to implement my project little by little over the course of a few weeks, giving me time to test it at work and fix the problems I encountered.

For example, if the user enters an incorrect username or password with the Department Deposits button, it used to bring the program to a halt and end with a VBA error. I knew what was going on, but I realized that others using the program wouldn't know what that error meant or how to fix it. So I had to figure out how to catch that error in VBA and make it pop up my own error message. This is now what happens when a user enters an incorrect username or password:



Another error that I had to notify the user of was with the Exchange Orders button. Sometimes the department doesn't select their department name on the order form, leaving it as the default "Select from List." When this happens, my program doesn't find a match on our log, so the information doesn't get copied over. But it wasn't triggering an error, so the user didn't know that it hadn't worked correctly. I had to go into the code and add a couple Booleans that trigger when an order isn't copied over. A message box then comes up at the end saying that one of the orders wasn't copied over correctly. It doesn't close that workbook so that the user can go back and select the correct department name and run it again.

I learned a lot of functions and capabilities of VBA that I didn't know before. I learned how to make a subprocedure run automatically at a specific time every day. I learned how to use the Find method of the Range object. I learned how to use the Trim function to remove trailing spaces from a string. I learned how to use the Format function to format a date the way I want it. I learned how to populate a Combo Box on a userform. I learned how to use the printout method of the worksheets object. And I learned a lot of good uses of message boxes and userforms.

Overall, I learned a lot about what VBA is capable of and how to make things work. My confidence in using VBA to solve simple problems increased greatly, and my grasp of programming concepts is now much stronger.