BYU MBA 614 - SECTION 1

Update FX Spreadsheet User’s guide

By

David Tate

4/9/2012

Executive Summary

The Company:

Unicity International is headquartered in Orem, UT and has operations and offices located throughout the world. It sells dietary and cosmetic products through direct marketing channels (Multi-Level Marketing).

The Problem:

Unicity International deals extensively with foreign exchange rates (fx rates) as the majority of its business is overseas. It is critical that Unicity regularly update the fx rates for each currency that it has transactions in. The updating process is done on the last business day of every month by one employee. This employee will manually look up the average monthly rate and the spot rate on the last business day of the month then input that information into a spreadsheet. The employee will then calculate the inverse rates and update the corresponding sheets as well. As part of updating the spreadsheets, the employee will have to add rows, columns, and format the new cells to match the past format. At this point the employee will then copy and paste the information found on the webpage into a word doc and create a PDF file that will be used by the auditors to verify that the FX rates Unicity used were the correct ones. This process is repeated for each relevant currency and takes between 6-8 hours to complete and is often inaccurate.

The Solution:

This workbook removes all of the manual processing described above. Specifically, it does the following based on input gathered from the user:

- It looks up the spot rates on the date provided by the user
- It looks up the average monthly spot rate for each currency
- It inputs those values into the appropriate worksheets
- It calculates the inverse spot and average rates and updates the appropriate worksheets
- It updates the worksheets to match the prior formatting
- It creates, saves, and closes one support document (PDF) that contains the necessary information for each fx rate
- It than creates and saves a non vba copy of itself (management does not want vba on their computers)

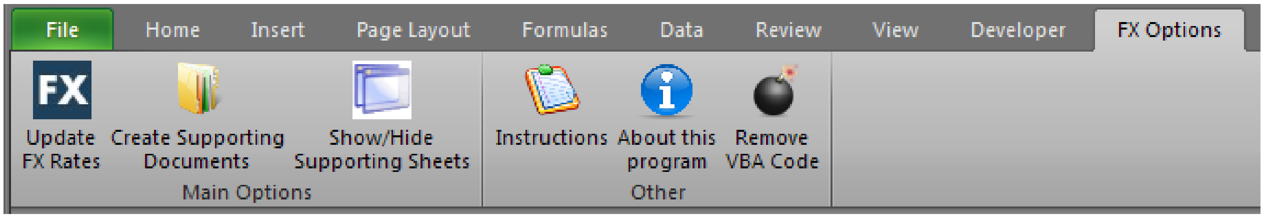
Additional Features:

- Basic error trapping for the most likely errors
- Allows for the user to add or remove currencies from the matrix
- All user interaction is conducted through the ribbon and subsequent forms
- Has a toggle button that hides/shows the supporting tabs for each currency
- Has progress bars indicating how the program is progressing while running
- Information about the program (who created it, the version, and how to contact me)
- Sends me an email that alerts me after the program has been run

Conclusion: This process once took between 6-8 hours to complete and had a high degree of inaccuracy. It now can be completed in about 5 minutes with perfect accuracy assuming the user doesn’t make a mistake I did not account for.

Table of Contents

Users Guide (Implementation)



The features of this program are accessed through buttons on the ribbon located in the FX Options tab as shown above. All user interaction with the program will be accomplished through these buttons. From right to left, the button functions and how they are to be used are discussed in the following section.

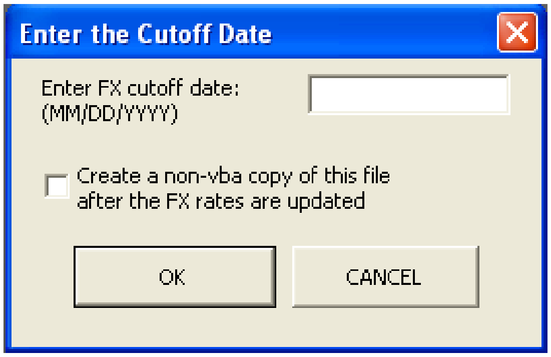
Button: Update FX Rates

The purpose of this button is to gather data from the user then update all of the FX rates. This is accomplished by using two user forms:

Form 1: Enter the Cutoff Date

This form is used to gather 2 pieces of input from the user, (1) the cutoff date and (2) whether or not the user wants to create a non-VBA version of this file.

The date input field is programmed to accept only a valid date in the form mm//dd/yyyy. This date must be after the date the last time the program ran but not by more than one month. Various error messages will appear if an invalid date is entered. This date validation was created because the FX rates are updated only once a month on the last business day of the month.


Enter the Cutoff Date

Enter FX cutoff date:
(MM/DD/YYYY)

☐ Create a non-vba copy of this file after the FX rates are updated

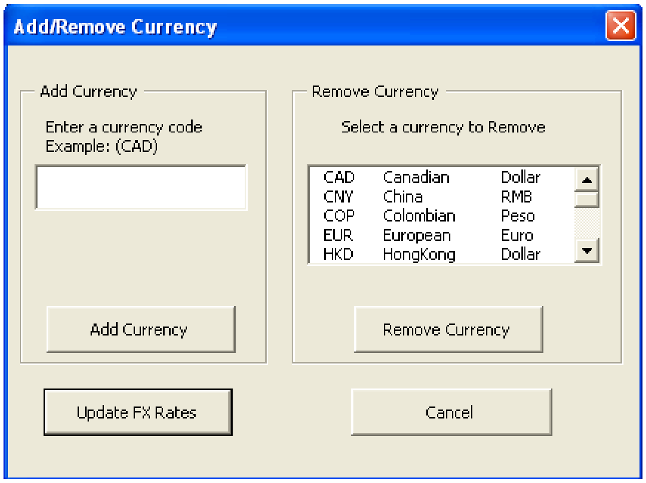
If the user checks the box, “create a non-VBA copy of this file after the FX rates are updated”, a separate file containing the necessary sheets and no VBA code will be created when the entire process is complete. This feature was added because management did not want VBA files sent to them.

If the user clicks “ok”, the program will progress if an acceptable date is entered or will display a message box with additional instructions. The program will close if the user clicks “cancel”.

Form 2: Add/Remove Currency

The purpose of form two is to make sure the desired FX rates will be looked up. It has two parts, (1) Add Currency and (2) Remove Currency.

The form is loaded and initially displayed as shown to the right. The 3 column list box is populated based on the currency codes and names located on 1st, 4th, and 5th rows in sheet BS Rates.



CAD	Canadian	Dollar
CNY	China	RMB
COP	Colombian	Peso
EUR	European	Euro
HKD	HongKong	Dollar

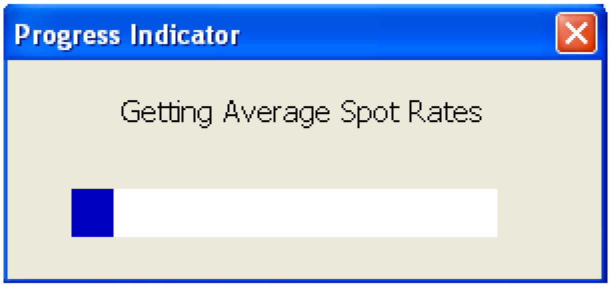
Add Currency

The add currency textbox provides the user an option for adding additional FX rates to be looked up. When the currency code is entered and the "add currency" button is clicked, the new currency will be added to the list in the list box. The added currency code must be a valid 3 digit code. The user should double check this code.

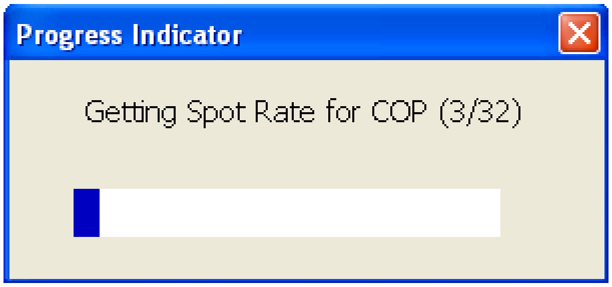
Remove Currency

The remove currency combo box allows the user to select a currency and remove it from the list. This action will also delete all data related to the selected currency on the main FX sheets (main FX sheets refers to the collection of sheets and includes the BS rates, IS Rates, Inventory Rates, Inverse IS Rates, and Inverse BS Rates sheets). Before deleting the information, a warning message is displayed that asked if the user wants to continue or not and based on the users input, will continue or exit the sub program.

When the codes in the list are the ones the user wants to look up, click Update FX rates. This button will then call several sub procedures that will prepare the workbook to receive data, retrieve the new FX rates, add them to the correct locations, format all of the main FX sheets, and then create and save a non VBA version of the file if the user selected this option on form 1. When the program is running, two progress bars will appear showing the progress of the average and spot FX rate retrieval process.



Getting Average Spot Rates



Getting Spot Rate for COP (3/32)

Useful notes about how the program works

Formatting:

The program reads the format in column C of the main FX sheets and copies that format to all of the columns in the sheet that contain FX codes. The user may change the format of the entire sheet by updated the format in column C.

FX list creation:

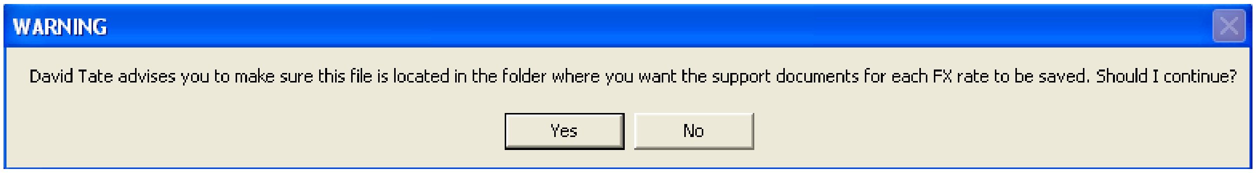
The FX list shown in the listbox is originally populated from the first row on sheet BS Rates. The user may manually add a currency by adding the 3 digit code at the end of the list of codes on row one. The user may manually remove a code by deleting the entire column for the currency. The FX codes must be contained on row one of sheet BS Rates for the program to work. The user may also add/remove currency codes using form 2 as discussed above.

FX sheet creation:

The program will import the data for each spot rate onto its own sheet and the average rates into one sheet. These sheets are then hidden and cannot be access unless the user uses the Show/Hide button discussed later.

Button: Create Supporting Documents

When this button is clicked, a warning message appears that informs the user that the support documents will be created in the same folder that the workbook is located in. Based on the users input, the program will continue or end.



David Tate advises you to make sure this file is located in the folder where you want the support documents for each FX rate to be saved. Should I continue?

If the user clicks yes, then the program will read the data from each of the individual currency sheets and will make a PDF file of each. This is accomplished by inserting the data into a Word document, printing a PDF, saving the PDF, and then closing the Word document. Not all the data in the currency sheets will be in the PDF version. Non-relevant information on the sheets will be removed and only the relevant data will be inserted into the PDFs.

Button: Show/Hide Supporting Sheets

This button is used to toggle between showing and hiding all sheets in the workbook other than the main FX sheets and the Matrix sheet. This button allows the user to quickly access the currency sheets as needed then quickly hide them when done. This option was included so that the user could quickly view or hide the source data. These sheets can only be accessed through this button. Attempting to unhide these sheets through conventional methods will not be successful.

Button: Notes

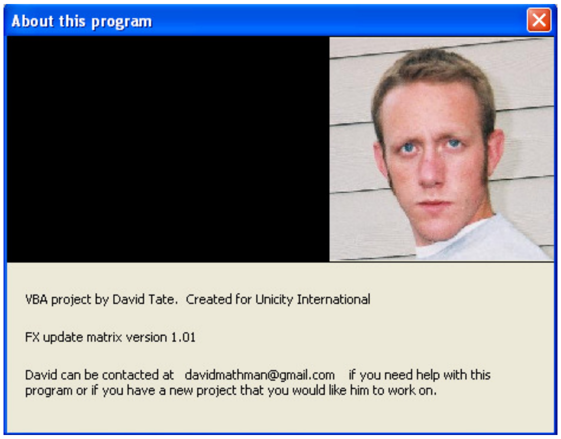
This button allows the user to toggle between showing and hiding the notes sheet. This sheet can also be revealed by using the Show/Hide button discussed above.

I provided the user with a personal one-on-one training that replaces a detailed instruction page. Therefore, the notes sheet is not meant to inform the user of all of the nuances of the program but rather to give the high level purpose of each button and a note or two that I consider important.

It was my intention that the forms, buttons, and imbedded message boxes provide enough information and error trapping so that the user should be able to operate the program.

Button: About this program

This button reveals a form that contains information about the program. The information includes my name, my picture, the program's version, and my contact information. It also hints that I would be open to working on additional projects in the future.



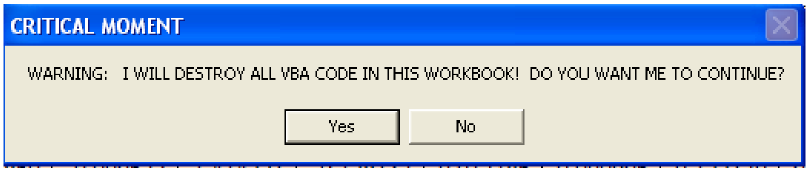
VBA project by David Tate. Created for Unicity International

FX update matrix version 1.01

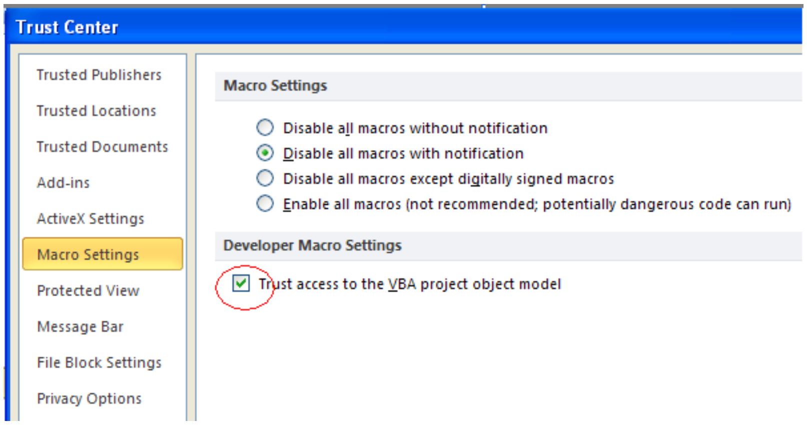
David can be contacted at   davidmathman@gmail.com   if you need help with this program or if you have a new project that you would like him to work on.

Button: Remove VBA code

This button will remove all VBA code within the file. When the button is clicked, a message box appears that warns the user that this sub will remove all VBA code within the workbook. If the user proceeds, all VBA code, including the program doing the removal, will be deleted and cannot be undone. This function was originally designed with the intention of creating a VBA free workbook for management, but it was subsequently replaced by the option on Form 1 discussed above. However, I think the code is handy so I left it active in the ribbon.



WARNING: I WILL DESTROY ALL VBA CODE IN THIS WORKBOOK! DO YOU WANT ME TO CONTINUE?



Macro Settings

○ Disable all macros without notification
● Disable all macros with notification
○ Disable all macros except digitally signed macros
○ Enable all macros (not recommended; potentially dangerous code can run)

Developer Macro Settings

☑ Trust access to the VBA project object model

This sub will only work if the user has enabled "Trust access to the VBA project object model" in the security settings.

Conceptual Difficulties

This project presented me with several difficulties that I had to overcome. Eventually I was able to overcome all of my difficulties or find a work around. In order of most difficult to least difficult, my main issues were as follows:

Navigating Oanda.com: this website proved especially difficult to navigate using vba because of all of the Java running in the back ground. I was unable to navigate it myself and sought help from Professor Allen. He too was unable to successfully navigate the page. However, I created a workaround by using a non-Java version of the page and successfully navigating it.

Exporting to Word and Creating a PDF: this action took quite a bit of work just to learn a few basic Word specific objects, methods, and properties. It appears that there is very little information on the internet regarding Word VBA. The most difficult part of this process was finding the needed export format verbiage: wdExportFormatPDF.

VBA Code Destruction: creating a VBA sub that would eliminate all VBA code within a workbook presented quite a challenge. I first had to learn how to get access to a reference library that allowed me to directly alter VBA and second I had to create code that would do what I wanted. Every time I made a mistake or just executed one line of this code, I had to close the file and open it again so as to undo any actions the sub may have taken. This sub was very frustrating, but it is something I am proud of.

Ribbon: I added several buttons and corresponding Microsoft pictures as well as personal pictures to the ribbon before we were taught how to do it in class. This required me to discover what the Custom UI editor was, how to use it, and the structure of the necessary language. Most of the learning came from trial and error as well as opening Professor Allen's files and examining how he coded his ribbon. It was unfortunate that I could not wait until the ribbon topic was taught in class to add these features to my program.

Progress Bars: creating the progress bas was a struggle. Creating labels and text that changed as the program looped was easy. The hard part was discovering how to make it so that the program did not pause every time the form updated. Eventually, I was able to find the vbModeless setting and was able to get it to work.

Parsing a string: parsing the strings that were downloaded from OAnda.com were particularly difficult. This difficulty stemmed from the fact that the number I needed within the string varied from 7 to 11 digits long and neither the left nor the right sides of the string were a consistent number of characters. In the end, the only constant in the strings was a decimal point and so I used that as a starting point to build the left and right sides of the fx rates.

Assistance from others

I do not believe I received substantial assistance on this project. Given the scope and sheer size of the number of subs I created, the help I did get was mostly centered on small tweaks to improve efficiency or utility of my code. Most of the challenges I faced while creating this program I was able to resolve by using the textbook or internet searches. When faced with a problem, I create a method that worked then sought help from Professor Allen to refine the process or learn a different way to perform the same task. For example, I researched and created a way to export my data from excel to word, but it was cumbersome and inefficient. I approach Professor Allen and he showed me how to directly insert into word rather than copying and pasting the data.

User Warnings

The user is warned again not to use the remove VBA button unless he really wants all VBA deleted from the workbook. Also, allowing access to the "Trust access to the VBA project model" is inherently dangerous.

The user is warned that the program will send the programmer (me) an email every time the program is run. The information emailed to me will include who ran the program and what day and time it was ran. No other information will be gathered.

The program does not distinguish between a valid or invalid 3 digit currency code.