

JASON JORDAN, FINAL PROJECT WRITE-UP

Project #1

Executive Summary

I chose to automate a mundane daily task at by job for my VBA project. Every day I receive a list of individuals whose refunds must be reversed. It is my job to use this list to create the journal entry that will actually reverse the refunds. To do this, I have to copy and paste the information over one piece at a time and change a variety of items so they are in the correct format for the journal entry. I will use screenshots to help illustrate this (names and ID numbers have been changed to protect the innocent).

Here is what I receive every day. Notice that the ID numbers all start with a “B” which must be removed. Also, the names will need to be changed to “last name, first name middle” format.:

G	H	I	J	K	L	M	N	O	P	Q
rn Type Code	Return Type Desc	Currency	Credit Amount	Debit Amount	Individual ID	Individual Name	Effective Date	Descriptive Date	Receiving RDFI	Account
10	Return		2350	0	B123456789	Mickey Mouse	3/23/2011	110323	123456789	12345
10	Return		1176	0	B234567890	Charles In Charge	3/23/2011	110323	123456789	12345
10	Return		625	0	B345678901	John Dillinger	3/23/2011	110323	123456789	12345
40	NOC		45	0	B456789012	Mean Mister Mustard	3/24/2011	110324	123456789	12345
40	Return		862.5	0	B567890123	Jimi Hendrix	3/23/2011	110324	123456789	12345
49	Return		1622.6	0	B678901234	Just Joe King	3/23/2011	110324.3	123456789	12345
58	Return		2382.7	0	B789012345	Jon Bon Jovi	3/23/2011	110324.6	123456789	12345
67	NOC		3142.8	0	B890123456	Harriet Beecher Stowe	3/24/2011	110324.9	123456789	12345
76	Return		3902.9	0	B901234567	Tom Petty	3/23/2011	110325.2	123456789	12345
85	Return		4663	0	B012345678	Paul Bunyon	3/23/2011	110325.5	123456789	12345

Here is what it will look like when my program runs it with the click of the “Create JE” button.

B	C	D	E	F	G	H	I	J	K	L	M
Journal Id	Journal Header (254)/ Line Description (30)	OperUnit	Acct	Class	Debit	Credit	Reference	mm-dd-yyyy Date			
	Mickey Mouse	49890305	1000	09799	2,350.00		123456789	03-25-2011			
	Mouse, Mickey	11890077	1820	REFND		2,350.00	123456789	03-25-2011			
	Charles In Charge	49890305	1000	09799	1,176.00		234567890	03-25-2011			
	Charge, Charles In	11890077	1820	REFND		1,176.00	234567890	03-25-2011			
	John Dillinger	49890305	1000	09799	625.00		345678901	03-25-2011			
	Dillinger, John	11890077	1820	REFND		625.00	345678901	03-25-2011			
	Mean Mister Mustard	49890305	1000	09799	45.00		456789012	03-25-2011			
	Mustard, Mean Mister	11890077	1820	REFND		45.00	456789012	03-25-2011			
	Jimi Hendrix	49890305	1000	09799	862.00		567890123	03-25-2011			
	Hendrix, Jimi	11890077	1820	REFND		862.00	567890123	03-25-2011			
	Just Joe King	49890305	1000	09799	1,623.00		678901234	03-25-2011			
	King, Just Joe	11890077	1820	REFND		1,623.00	678901234	03-25-2011			
	Jon Bon Jovi	49890305	1000	09799	2383.00		789012345	03-25-2011			
	Jovi, Jon Bon	11890077	1820	REFND		2383.00	789012345	03-25-2011			
	Harriet Beecher Stowe	49890305	1000	9799	3143.00		890123456	03-25-2011			
	Stowe, Harriet Beecher	11890077	1820	REFND		3143.00	890123456	03-25-2011			
	Tom Petty	49890305	1000	9799	3903.00		901234567	03-25-2011			
	Petty, Tom	11890077	1820	REFND		3903.00	901234567	03-25-2011			
	Paul Bunyon	49890305	1000	9799	4663.00		012345678	03-25-2011			
	Bunyon, Paul	11890077	1820	REFND		4663.00	012345678	03-25-2011			

Create JE

Implementation Documentation

To create my sub procedure I selected different items from the source document and assigned them to variables (temporary memory locations). With the information I needed set up into convenient variables, I just inserted the values of the variables where they needed to be in the journal entry. A couple of the pieces of information I needed were in the incorrect format and this is how I fixed them:

- ID numbers are listed starting with a B, so I fixed this by using the following code:
 - `ID = Right((Sheets("CSV_File").Cells(X, 12).Value), 9)`
 - This code takes the right-most 9 digits of the given ID number which is the length of a student ID.
- The names I must use in a journal entry must follow different conventions depending on the account they are going to. In this case I had to take the full name given for the first line of the journal entry for a given student, then I had to use a “last name, first name middle” format for the second line. To do this, I created 2 variables – “Name” and “modName”. “modName” was the most difficult to figure out and I created it as follows:
 - `modName = Right(Name, Len(Name) - InStrRev(Name, " ")) & ", " & Left(Name, (Len(Name) - Len(Right(Name, Len(Name) - InStrRev(Name, " "))) - 1))`
 - This code determines the length of the last name by using “InStrRev” measuring back to the space before the last name. Next, a comma is placed and the rest of the name is placed after that.

The rest of the data I needed were the account codes for the debits and credits. I already know what these need to be so I just have the program write them into the journal entry where they need to be.

Some of the mechanics of making this program work were also tricky, here are a couple of them:

- To erase the information from the last journal entry, I used this code:
 - `Worksheets("JE_Template").Range("C2:J2").Select
Range(Selection, Selection.End(xlDown)).ClearContents`
 - This uses “xlDown” to select all fields with data in them then clear the contents.
- Because the journal entry takes 1 row of data from the source document and writes 2 lines with it in the journal entry, I had to have the program find the next empty cell to write in using this code:
 - `Sheets("JE_Template").Range("C1").Select
Do While Not IsEmpty(ActiveCell)
ActiveCell.Offset(1, 0).Select
Loop`

Since the data from the source document is provided in rows, I just used “X” as an integer to represent the current row number and added one to it each time the “do loop” ran. However, included in the source document are students who are receiving notice from the bank that they need to fix their account information, but who do not need to be included in the journal entry. These students are

always listed at the bottom of the report and are listed with a 0 amount, so to end the sub procedure I have it stop when it finds a student with a 0 balance. I used the following code in the do loop to do this:

- Do Until Sheets("CSV_File").Cells(X, 10).Value = 0

Learning and Conceptual Difficulties Encountered

I am very proud of my program! I have been telling everyone at work about it and they are proud of me too. I am someone who struggled in this VBA class, especially during the last portion, because I have limited confidence in my programming skills. Although this program may seem simple, it took me 6 hours to write it all and I really had to put some elbow grease into it. It now saves me time every day at work and I now get excited when I create this journal entry.

This activity helped me solidify some of the basic concepts such as “do loops” and variable usage. I also feel that I am getting better at knowing what is out there with VBA so I can go online and grab what I need to make a successful automated spreadsheet.

Another important thing I learned was the importance of getting your VBA code to function properly. In my case, had I not tested this code multiple times to look for glitches, I might have submitted incorrect journal entries which could have caused a lot of problems. VBA coding can be fun, but one must ensure it functions properly before using it in an employment setting.

Project #2

Executive Summary

In order to meet the scope requirements for my project (per Dr. Allen’s specific request to me that I do so), I added a second project. I decided that with all of the time I will save with the first project doing all of this work for me I would have time to play “Yahtzee” from time to time. I have always loved this game and I wondered how hard it might be to make it work in Excel, so I went for it. I have used it quite a few times already and it is fantastic. There is no need to make noise with loud dice, harbor doubts that you shook the dice incorrectly, or worry about that stray dice that occasionally makes its way to the floor causing a re-shake controversy.

Implementation Documentation

Displaying, randomizing, and holding the dice was my biggest challenge. I decided to display the numbers of the dice rather than try to bring up an image of the traditional dot-numerated dice. I want to use this Yahtzee game in the future and I do not want to have to worry about bringing the images along with it. I used the excel function “randbetween” to generate the random number between 1 and 6.

Learning and Conceptual Difficulties Encountered

Although this is a fun game, this still took me a while to get it working correctly and make it aesthetically pleasing. The biggest learning outcome for this game was probably learning how to implement rules into VBA code. If I had not addressed the problem of using more than the allowed 3 rolls, the user would surely not keep track and would abuse their roll privileges. I designed the code so the button would be useless after 3 rolls until the player was ready to start another turn.

Another thing I learned was how to prevent gross player error in using my Yahtzee game. I decided to place the “clear scorecard” button far away from the other buttons to prevent users from accidentally hitting this when going to roll or take another turn.

I also learned the value of simple code with this second project. I decided not to use graphical pictures of dice to show the values rolled because that would not add much to the game for the time and potential future problems associated with carrying around the pictures of the dice to get this to work. I also chose to let users manually input their scores on the scorecard because there is a choice element involved in the game that I didn't want to eliminate.