

VBA Final Project Written Report: Business Firm's Special Project P&L

Cole Moffat
MBA 614, Sec 001

Executive Summary

A Consulting Firm has implemented a new Special Project and hired interns to work on it. This new project does not earn any revenues, but when there is a surge of billable work, the Firm takes some of the interns to fill in the gaps. This way, they can bill clients at the standard rate, pay the interns the same as they always do, keep the difference, and not have to constantly hire and fire their workforce to meet the demand. Interns submit weekly timecards, detailing the things they work on. Paychecks to interns is cash out from the Project, and amounts billed to clients from interns' work is cash in from the project. The Business Firm wants to know how much this Special Project is making them or costing them, updated weekly.

The VBA program, complete with custom buttons on a new ribbon tab, prompts the user to select a raw data file with the updated timesheet information; imports the data, configures it to fit the template, and appends it to the existing timesheet data; updates/ refreshes pivot tables; uses that updated information to recalculate a Project P&L; displays the last 4 weeks' data and the grand total since project inception; and saves the updated data as a new file with the data processing date. The Business Firm will know the profitability of their new Special Project in near- real time.

*This is based on a real business with a real project. Due to the sensitive nature of the data, all dollar amounts, hours, employee names, company names, or other personally identifying information has been changed. The Consulting Firm is referred to as the Business Firm (BIZ) and the new project is referred to as Special Project (SP).

Implementation Documentation

The various macros can be accessed via the new buttons in the customized ribbon:



The new ribbon tab "BUDGET TOOLS" has 4 buttons in 2 groups: "Update Budget" in the "Automate Budget" group; and "Expand Tables," "Add Timecard Data," and "Update Tables" in the "Individual Steps" group.

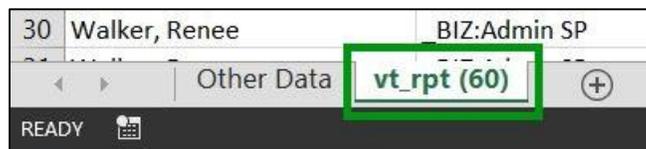
I wrote four macros to automate the whole process, which can be accessed via the buttons in the ribbon. The four macros are called:

- (1) insertNewData
- (2) expandTables
- (3) updateTables
- (4) SaveAsToday

The “Add Timecard Data” button runs the insertNewData macro, the “Expand Tables” button runs the expandTables macro, and the “Update Tables” button runs the updateTables macro. As you can see, the first 3 macros can be done independently, but the last macro (SaveAsToday) is only done when the user chooses to run the whole macro as one fluid procedure by pressing the “Update Budget” button with a green check. This button calls each macro necessary to complete the entire task. We’ll look at these 4 macros individually to see how the whole process is automated, then discuss a few user-set up items to run the macros properly.

1. insertNewData Macro

This is the first macro that the “Update Budget” button calls. This sub first prompts the user to select the file that has the raw time data that needs to be added to the “SpringAheadData” worksheet. The “SpringAheadData” worksheet is where all timesheet information since the inception of Special Project is stored. We’ll call the newly selected file the RawDataWorkbook, because it is exported directly from a data warehouse. Once the RawDataWorkbook is selected, excel opens it and searches for the correct worksheet with the new timesheet data. We’ll call the new data NewData. Typically, there are other worksheets in the RawDataWorkbook with other, unnecessary information. The NewData is always in the worksheet that has “rpt” in its name, and it is the only worksheet that has “rpt” in its name. This part of the macro reads in the NewData from this sheet.



The macro will find the sheet with “rpt” somewhere in it and pull the NewData from that worksheet.

When the NewData first comes into the “SpringAheadData” tab, it is not formatted correctly. The RawDataWorkbook has been exported from the data warehouse (a) with a few extra columns of information we don’t need (usually blank), and (b) without a “Week Ended” column, which is key to the functioning of the pivot tables and P&L. The insertNewData macro also reformats the NewData so that it matches with the formatting that already exists in the “SpringAheadData” worksheet. Different formulas are used to compute the “Week Ended” date, Special Projects vs. Business Firm Billable in the IS Mapping column, and a few others. The following screenshots show the transformation of the data from the time of import to when it finally becomes usable. Above the dotted line is the existing data, and below the line is the new data.

2644	Dunphy, Phil	_BIZ Admin SP	Intern	1099	1.25	03/02/2016	03/02/2016 Thu	NonBill	22	Pending	Special Projects	12	15	D
2645	Dunphy, Phil	_BIZ Admin SP	Intern	1099	3	03/02/2016	03/02/2016 Fri	NonBill	22	Pending	Special Projects	12	36	D
2646	Dunphy, Phil	_BIZ Administrative	Intern	1099	1	03/02/2016	03/02/2016 Mon	NonBill	22	Pending		12	12	D
2647	Dunphy, Phil	_BIZ Administrative	Intern	1099	2	03/02/2016	03/02/2016 Tue	NonBill	22	Pending		12	24	D
2648	Dunphy, Phil	_BIZ Administrative	Intern	1099	1.25	03/02/2016	03/02/2016 Wed	NonBill	22	Pending		12	15	D
2649	Dunphy, Phil	Some Client Equity Assista	Intern	1099	2	03/02/2016	03/02/2016 Fri	Admin 5%	22	03/02/2016	Business Firm Billable	66.25	132.5	D
2650	Dunphy, Phil	Some Client Equity Assista	Intern	1099	1.25	03/02/2016	03/02/2016 Sat	Admin 5%	22	03/02/2016	Business Firm Billable	66.25	82.8125	D
2651	Dunphy, Phil	Some Client Equity Assista	Intern	1099	1.25	03/02/2016	03/02/2016 Mon	Admin 5%	22	03/02/2016	Business Firm Billable	66.25	82.8125	D
2652	Dunphy, Phil	Some Client Equity Assista	Intern	1099	1.25	03/02/2016	03/02/2016 Thu	Admin 5%	22	03/02/2016	Business Firm Billable	66.25	82.8125	D
2653	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.25	0	0	4245 Mon	NonBill	22	Pending			
2654	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.25	0	0	4245 Tue	NonBill	22	Pending			
2655	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.25	0	0	4245 Wed	NonBill	22	Pending			
2656	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.25	0	0	4245 Thu	NonBill	22	Pending			
2657	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.75	0	0	4245 Fri	NonBill	22	Pending			
2658	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.75	0	0	4245 Sat	NonBill	22	Pending			
2659	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.25	0	0	4245 Mon	NonBill	22	Pending			

This is how the NewData appears when it's first imported

650	Dunphy, Phil	Some Client Equity Assista	Intern	1099	1.25	01/03/2016	01/03/2016 Sat	Admin 5%	22					
651	Dunphy, Phil	Some Client Equity Assista	Intern	1099	1.25	01/11/2016	01/16/2016 Mon	Admin 5%	22					
652	Dunphy, Phil	Some Client Equity Assista	Intern	1099	1.25	01/14/2016	01/16/2016 Thu	Admin 5%	22					
653	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.25			4245 Mon	NonBill	22	Pending			
654	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.25			4245 Tue	NonBill	22	Pending			
655	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.25			4245 Wed	NonBill	22	Pending			
656	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.25			4245 Thu	NonBill	22	Pending			
657	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.75			4245 Fri	NonBill	22	Pending			
658	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.75			4245 Sat	NonBill	22	Pending			

651	Dunphy, Phil	Some Client Equity Assista	Intern	1099	1.25	01/11/2016	01/16/2016 Mon	Admin 5%	22					
652	Dunphy, Phil	Some Client Equity Assista	Intern	1099	1.25	01/14/2016	01/16/2016 Thu	Admin 5%	22					
653	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.25	03/21/2016	03/26/2016 Mon	NonBill	22	Pending				
654	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.25	03/22/2016	03/26/2016 Tue	NonBill	22	Pending				
655	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.25	03/23/2016	03/26/2016 Wed	NonBill	22	Pending				
656	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.25	03/24/2016	03/26/2016 Thu	NonBill	22	Pending				
657	Delgado, Manny	_BIZ Admin SP	Intern	1099	1.75	03/25/2016	03/26/2016 Fri	NonBill	22	Pending				

The dates and days of the weeks needs to be shifted over and formatted as a date, not a number. The "Week Ended" column needs to be populated with a formula to calculate the correct week ended date.

1.25	01/03/2016	01/03/2016 Fri	Admin 5%	22	02/02/2016	Business
1.25	01/09/2016	01/09/2016 Sat	Admin 5%	22	02/02/2016	Business
1.25	01/11/2016	01/16/2016 Mon	Admin 5%	22	02/02/2016	Business
1.25	01/14/2016	01/16/2016 Thu	Admin 5%	22	02/02/2016	Business
1.25	03/21/2016	03/26/2016 Mon	NonBill	22	Pending	
1.25	03/22/2016	03/26/2016 Tue	NonBill	22	Pending	
1.25	03/23/2016	03/26/2016 Wed	NonBill	22	Pending	
1.25	03/24/2016	03/26/2016 Thu	NonBill	22	Pending	
1.75	03/25/2016	03/26/2016 Fri	NonBill	22	Pending	
1.75	03/26/2016	03/26/2016 Sat	NonBill	22	Pending	

1.25	01/03/2016	01/03/2016 Fri	Admin 5%	22	02/02/2016	Business
1.25	01/09/2016	01/09/2016 Sat	Admin 5%	22	02/02/2016	Business
1.25	01/11/2016	01/16/2016 Mon	Admin 5%	22	02/02/2016	Business
1.25	01/14/2016	01/16/2016 Thu	Admin 5%	22	02/02/2016	Business
1.25	03/21/2016	03/26/2016 Mon	NonBill	22	Pending	
1.25	03/22/2016	03/26/2016 Tue	NonBill	22	Pending	
1.25	03/23/2016	03/26/2016 Wed	NonBill	22	Pending	
1.25	03/24/2016	03/26/2016 Thu	NonBill	22	Pending	

The "Type," "Approver," and "Status" data needed to be shifted to the appropriate column.

Admin 5%	22	02/02/2016	Business Firm Billable	66.25	82.8125	D
NonBill	22	Pending	Special Projects	12	15	D
NonBill	22	Pending	Special Projects	12	15	D
NonBill	22	Pending	Special Projects	12	15	D
NonBill	22	Pending	Special Projects	12	15	D
NonBill	22	Pending	Special Projects	12	21	D

The rest of the columns needed to be populated with their appropriate formulas.

It is important to note that for all but one column, the new formulas are not autofilled from the last row of the existing data. Instead the formulas are programmed with their own relative references. The reason for this is because there are some fields that may need to be edited after the macro has completed its task. For example, if a custom hourly rate is used for unique billable client situations, that new hourly rate can be entered in and overwrite the complex formula. If this happens in the last row of the previously existing data, that number should not be autofilled as

the NewData is brought in. Instead, a consistent formula is used for every new row in a column, regardless of what previous rows' formulas are.

2. expandTables Macro

While not the first macro called in “Update Budget,” it is the first individual step listed in the Ribbon because it’s relatively simple: The macro only adjusts the view of the data. All columns between Grand Total and the fourth most recent week of data are usually hidden and grouped. The “Expand Tables” macro unhides and ungroups these columns so that all data since project inception is available to view. This view is helpful if the user wants to see summary details from any week since project inception. The macro is useful if you are not familiar with the “1,” “2,” or “+” buttons above the column headers. Also, other macros require all columns to be visible to function, and this macro is called as part of those macros. To regroup all the columns with data older than 4 weeks, press the “Refresh Tables” button to run the macro that refreshes the pivot and P&L tables and regroups the necessary columns.

1									
2									
	A	B	C	D	E	F	G	CS	
1	BIZ Intern Program								
2	1. FINAL VBA project with Dummy Data								
3									
4									
5		Sum of Extended Costs							
6		Row Labels		03/26/2016	03/19/2016	03/12/2016	03/05/2016	Grand Total	
7		Business Firm billable		-	-	-	-	63,087	
8		Special Projects		42	225	429	456	44,539	
9		Grand Total		42	225	429	456	107,627	
10									
11									

Grouping the data is useful when looking at the newer data and understanding the bottom line. Columns H through CR are grouped and hidden (about 90 columns!).

The screenshot shows a very wide Excel spreadsheet with numerous columns and rows. The data is organized into a grid, with many columns containing numerical values and some containing text. The spreadsheet is densely packed with data, illustrating the result of ungrouping and unhidden data.

Ungrouping the data is useful to get to the details of each week's data, but can be overwhelming if you don't need that much information. It also is needed in order to refresh the table.

3. updateTables macro

This macro starts by calling the expandTables macro described above. If the tables weren't expanded, some parts of the procedure would only operate on visible cells, while others would

operate on any cell, regardless of its visibility. To avoid any issues, all columns are ungrouped and made visible.

Excel already has a built-in button to refresh pivot table data; in fact, there's a button to refresh all pivot tables on a worksheet. However, there are some limitations to using the built-in button. A cell in a pivot table must be selected, and the built-in button only updates a pivot table to reflect changes within the existing data source. Because NewData was just added to the "SpringAheadData" tab, we want the pivot tables to incorporate this additional data. The updateTables macro is used to do all of this. The current region of the relevant data in "SpringAheadData" is entered in as the new data source for each of the three pivot tables in "Projects P&L." For each new week with data in it, the pivot table expands one column to the right when it is refreshed. Business Firm also wants to include only the data as far back as the week ended 14 Jun 2014; this filter isn't carried through the refresh correctly when the new weeks are added, and needs to be adjusted as well. After the pivot tables are updated, the "Grand Total" section of the P&L needs to be transferred to the right place, without using the clipboard. These adjustments are shown in the screenshots on the following page.

	CS	CT
2014 06/14/2014	Grand Tot	
245	1,826	63,585
306	243	45,448
551	2,069	109,034
Grand Total		
2014	Total	
306	(243)	
306	(243)	
245	1,826	
245	1,826	
939	1,583	
2014 Grand Total		
20	3,711	
22	761	
42	4,472	
2014 Grand Total		
-	283	
-	185	
-	98	
-	12	

Each of the pivot tables is adjusted, and the "Grand Total" section of the P&L is put into place

Next, the whole P&L section (the navy header) needs to be updated. The columns up through the column before "Grand Total" need to be updated to pull the relevant data. If multiple new weeks were added, there may be various blank columns. The formulas in the "Grand Total" section need to be updated to accurately reflect the actual totals, not just the totals for the week ending 06/14/2014. This transformation is shown below:

06/21/2014	Grand Total	Grand Total
(306)	(243)	(243)
(306)	(243)	(243)
1,245	1,826	1,826
1,245	1,826	1,826
939	1,583	1,583

06/21/2014	06/14/2014	Grand Total
(306)	(243)	(243)
(306)	(243)	(243)
1,245	1,826	1,826
1,245	1,826	1,826
939	1,583	1,583

6/21/2014	06/14/2014	Grand Total
(306)	(243)	(45,448)
(306)	(243)	(45,448)
1,245	1,826	63,585
1,245	1,826	63,585
939	1,583	18,137

Finally, the columns between column G and “Grand Total” need to be grouped and hidden so the final P&L spreadsheet is readable and manageable. You’ll notice in the screenshot below that the columns are now hidden, and the most recent week of data is “04/02/2016”; before it was “03/26/2016.” The four most recent weeks are shown together with the grand total.

	A	B	C	D	E	F	G	CS
1	BIZ Intern Program							
2	1. FINAL VBA project with Dummy Data							
3								
4								
5			Sum of Extended Costs					
6			Row Labels	03/26/2016	03/19/2016	03/12/2016	03/05/2016	Grand Total
7			Business Firm billable	-	-	-	-	63,087
8			Special Projects	42	225	429	456	44,539
9			Grand Total	42	225	429	456	107,627
10								
11								
12				03/26/2016	03/19/2016	03/12/2016	03/05/2016	Grand Total
13			Cash outflows:					
14			Special Project	(42)	(225)	(429)	(456)	(44,539)
15			Total outflows	(42)	(225)	(429)	(456)	(44,539)
16								
17			Cash inflows:					
18			Business Firm Billable	-	-	-	-	63,087
19			Total inflows	-	-	-	-	63,087
20								
21			Net cash inflow (outflow)	(42)	(225)	(429)	(456)	18,548
22								
23								
24								
25			Sum of Hours (by type)					
26			Row Labels	03/26/2016	03/19/2016	03/12/2016	03/05/2016	Grand Total
27			Special Projects	4	19	36	38	3,711
28			Business Firm billable	-	-	-	-	761
29			Grand Total	4	19	36	38	4,472
30								
31								
32								
33			Sum of Hours (by user)					

Columns H through CS are hidden (about 90 columns), and we see that the NewData is now summarized in the various tables.

4. SaveAsToday Macro

This macro saves the updated spreadsheet as a new file with today’s date, and is only called when the “Update Budget” button is pressed and the entire process is run. It may so happen that

a user runs the complete “Update Budget” macro various times in a single day, but wants to keep each version that is saved. If a file entitled “SpecProj Budget” with today’s date already exists in the folder where the current workbook is saved, the macro does not overwrite the file, but adds a “(1)”, “(2)”, “(3)”, etc. after the file name. If there is no file of the same name, the number and parentheses are omitted.



As previously mentioned, the “Update Budget” button calls the insertNewData macro, the updateTables macro (which includes the expandTables macro), and the SaveAsToday macro. “Update Budget” runs the entire process, and the only extra user interaction is simply selecting the RawDataWorkbook file.

Ideally, a user presses “Update Budget,” selects the new file, and has an updated budget saved as a new file with today’s date. No further work necessary. However, this is not always the case, and the extra buttons to run the separate macros are required. As mentioned before, the actual billable rate may not be populated correctly because a particular client may have a unique situation and be billed at a different rate. This realization can happen at any point. If the user knows data will need to be adjusted from the start, the user should press “Add Timesheet Data,” make the changes to the data after it’s imported, then press “Update Budget” without selecting a file. This will complete the process. If the user realizes that changes need to be made after “Update Budget” runs all the macros, the user can still modify the timesheet data, press “Update Tables” to reflect the changes in the P&L worksheet, and save the file (because it was already saved as a new file with today’s date). We have previously discussed the convenience of the “Expand Tables” button as well. Thus, we see that these individual steps are necessary for the full functionality of the automation process.

Finally, when adding new data every week, the most recent copy of the workbook should be used. Using an older copy will result in lost data, as only the most recent copy will contain all the timesheet data up to that point.

Discussion of learning and conceptual difficulties encountered

I really enjoyed working on this project. Not only did I learn a lot about VBA, programming logic, and debugging (definitely a lot of debugging), but I was doing all of this for a real company with real needs, managing real money and finances. I added value! When I turned this in a few weeks ago, they were very impressed with my work, and definitely want a weekly update on the budget (previously, it was done every few months). Because everything is automated, it will no longer be prone to human errors, either.

One big difficulty that I had to work through was programming formulas with relative references into 100s of cells using VBA. I eventually learned about the FormulaR1C1 property of the range or cells object that you can write using VBA. Using this style of formula, both absolute and

relative references can be used. Within a formula, R5C7 is the absolute reference to cell G5 (5th row, 7th column). R[5]C[7] is a relative reference, referring to the cell 5 rows below and 7 columns to the right of the cell containing the formula. This was absolutely necessary to adding all the necessary formulas to the NewData when it was imported.

I also needed to essentially copy and paste the final “Grand Totals” column of the P&L to its new place, but I wanted to do it without the clipboard. My immediate thought was using Range.Value to move it, but that would only bring over the actual values, not the formatting, too. After a lot of digging, I found that the .value property of the range object can actually take in an optional argument: “RangeValueDataType.” This can be a built-in excel constant, for example, “xlRangeValueXMLSpreadsheet,” which is a constant for the number 11. Range.Value(10) is the same as Range.Value, and will just return the values in a cell. However, Range.Value(11) doesn’t just input the same values, but it returns the values, formatting, formulas, and names of the specified Range object in the XML Spreadsheet format. This is exactly what we wanted! Success.

Assistance

The only assistance I received was with the pivot table update. I loosely used another person’s “update all pivot tables on a sheet” loop macro, but that was before we learned about pivot tables in class. That part of the code required quite a bit of modification anyway, and I certainly would have been able to do it myself if I had gotten to that part of the code after we talked about pivot tables in class. Other than the pivot tables and a few random things like Range.Value(11) and FormulaR1C1, I used only my own notes and the MacroRecord function in excel.