

MBA 614 Cumulative Project

Automating Columbia Physical Therapy Profit-Sharing

Kramer Holle

Executive Summary

Columbia Physical Therapy (CPT) is a regional physical therapy company that operates in various towns in Washington, Oregon, and Idaho. The Company has grown at a steady rate over the last 25 years (approximately how long the company has been in existence). Each clinic has a clinic manager, who is a physical therapist. While the therapists are very good at managing clinics, the current accounting system is archaic.

At the end of each quarter, the company accountant calculates profit-sharing to determine what the company should distribute to each therapist. Thus, it is important that the profit-sharing numbers are calculated accurately. Because the accounting system is fairly limited, the profit-sharing is not calculated through the accounting system. Instead, the accountant uses an Excel spreadsheet to keep track of all the items used to compute the profit-sharing. In the past, the billing office manager has emailed the accountant 25 Word documents with all the relevant data (the formatting on the exported Excel documents is very difficult to use). The accountant has then copied those documents to an Excel file, and then used that data to determine profit-sharing numbers. This is a time-consuming process, sometimes taking hours or all day. The goal of my project was to create a template spreadsheet that would import the files and calculate profit-sharing for each clinic in a matter of minutes.

Project Overview

The following bullets give an overview of how I would like the macros I have written to be used by the company accountant (with the associated subprocedures):

- Import the workbook files with profit-sharing data by clicking on the custom button “Import Clinic Results” on the “CPT” tab. As the files are imported, the program runs a macro that automatically reformats the data.
 - *Sub Callback (used to connect the following files to the custom ribbon)*
 - *Sub ImportFiles*
 - *CleanUpWorksheet*
 - *Deleteallshapes*
 - *Unmerge*
 - *Trim*
 - *DeleteBlue*
 - *Removerows*
 - *Removecolumns*
 - *Sub Autofit*
- Import the administrative files by clicking on the custom button “Import Admin”. Then reformat the data by clicking the custom button “Clean Up Admin”
 - *Sub Callback1*
 - *Sub ImportAdmin*
 - *CleanUpAdmin*
 - *Deleteallshapes*
 - *Unmerge*
 - *Trim*
 - *Cleanup2*
 - *RemoveBlankRows*
 - *RemoveBlankColumns*

- Autofit
- Sub Callback2
 - Sub CleanupAdmin2
 - RemoveBlankRows
 - RemoveBlankColumns
 - Removerows
 - Removecolumns
 - DeleteIncomplete
- Once the data has been imported, run subprocedures and functions that calculate key numbers needed to produce profit sharing.
 - Sub Callback3
 - FindCollections
 - CompanyCollections
 - Function FindProfit
 - Function FindInsurance
 - Function FindCommissions
 - Function AccruedRetirement

Implementation

Objective #1: Clean up the formatting

As you can see from the following pictures, the formatting on the files exported from the accounting system is really tough to use by other procedures or functions. Merged cells, shapes, and page numbers, as well as extra rows and columns, make it difficult to easily manipulate the data.

COLUMBIA PHYSICAL THERAPY, INC						
Income Statement-Cash Basis						
Quarter-to-Date Variance, December 2014 - 1 quarter back, Department 30						
		3 Months Ended	3 Months Ended	Variance		
		December 31, 2014	December 31, 2014	Fav/<Unf>	% Var	
		Budget				
Revenue-Cash Basis						
Professional Fees-Kennewick		104,230.78	137,283.28	(33,052.50)	#####	
Other Income-Kennewick		1,825.00	2,316.47	(491.47)	#####	
Report Fees-Kennewick		506.21	283.78	222.43	78.4 %	

88	Educ Exp - Matthew Patislo	570.00	862.50	292.50	33.9 %	
89						
90						
91						
92	4/7/2015 10:07 AM					Page 1
93						
94						
95						
96						
97						
98						
99						
100						
101						
102	DUES&SUBSCRIPTIONS-KENNEWICK	0.00	67.78	67.78	#####	
103	Insurance-General-Kennewick	770.60	123.75	(646.85)	#####	
104	Insurance-malpractice-Kennewic	0.00	195.75	195.75	#####	






Therefore, my first goal was to develop code that would clean up the spreadsheets so the data could be manipulated. This took some time because I had to clean up the formatting on the sheets, and also clean up bad formatting that had been built-in by the systems administrator. For example, many of the cells had leading and trailing spaces. Accordingly, I wrote a trimming procedure that kept the formatting by using indentation while getting rid of leading spaces. This shows what the page looks like once the formatting procedures have been run.

Revenue-Cash Basis				
Professional Fees-Richland	156,213.44	152,788.50	3,424.94	2.2 %
Other Income-Richland	244.00	0.00	244.00	
Report Fees-Richland	692.03	571.50	120.53	21.1 %
Retail Sales-Richland	0.00	45.00	(45.00)	-100.0 %
Refunds(Overpayments)-Richland	(519.28)	(106.50)	(412.78)	-387.6 %
TOTAL Revenue-Cash Basis	156,630.19	153,298.50	3,331.69	2.2 %
NET REVENUE	156,630.19	153,298.50	3,331.69	2.2 %
GROSS PROFIT	156,630.19	153,298.50	3,331.69	2.2 %
Expenses				
Payroll, taxes and benefits				
Salaries-PT-Richland	31,923.10	33,069.69	1,146.59	3.5 %
COMM - PT	22,054.00	13,842.00	(8,212.00)	-59.3 %
COMM - OFF	2,417.00	1,268.85	(1,148.15)	-90.5 %
Salaries-Office-Richland	16,127.58	14,555.79	(1,571.79)	-10.8 %
Payroll Taxes-Richland	3,248.26	5,332.41	2,084.15	39.1 %
Medical Insurance-Richland	3,450.00	3,450.00	0.00	
Retirement Plan Contribution - R	7,642.76	6,706.23	(936.53)	-14.0 %

Objective #2: Import the Files

Once I developed programming that would clean up the formatting, I then developed a subprocedure that would quickly import the files. My hope was that the company accountant could download all the raw data into a folder, and then the program I created would extract the data into the master spreadsheet. With the help of Nathan, our TA, I was able to develop a code that looped through all of the files in the “Books” Folder to import the relevant worksheets from each workbook. The code also renamed the worksheets to the file names so the files are easily identifiable.

I then wrote a program that performed the same function for the administrative files. I wanted to delineate the two types of files because the formatting was different, and the subs required to clean up the files were different. You can see example of the folder and the code below.

Name	Date modified	Type	Size
 Admin	4/7/2015 11:36 AM	File folder	
 Books	3/30/2015 3:28 PM	File folder	
 Recovered Files	4/9/2015 12:37 PM	File folder	
 TestFile_Template_RecoveredData.xlsm	4/11/2015 3:05 PM	Microsoft Excel M...	291 KB
 Miscellaneous	4/11/2015 1:21 PM	File folder	

```

Sub ImportFiles()

Dim filename As String
Dim folderPath As String
Dim dataWb As Workbook
Dim Name As String

'Get the file name
folderPath = ThisWorkbook.Path & "\Books"
filename = Dir(folderPath & "\*.xls")

'Import the worksheet
Do Until filename = ""
    Debug.Print filename
    Set dataWb = Workbooks.Open(folderPath & "\" & filename)
    dataWb.Sheets(1).Copy After:=ThisWorkbook.Sheets(ThisWorkbook.Sheets.Count)
    Name = Left(filename, Len(filename) - 4)
    ThisWorkbook.Sheets(ThisWorkbook.Sheets.Count).Name = Name
    CleanUpWorksheet
    dataWb.Close False
    filename = Dir
    Call autofit
Loop
End Sub

```

The code opens the files, imports them into the master workbook, and names them based upon the file extension. Once the files are imported to the master file, they are cleaned up by calling the “CleanUpWorksheet” sub (which calls all of the other formatting subs such as trimming, getting rid of shapes, and extra rows/columns). Then, to make the data more readable, the sub calls an Autofit sub that makes all the columns and rows the right height or width.

Objective #3: Manipulating the Data to Calculate Profit-Sharing

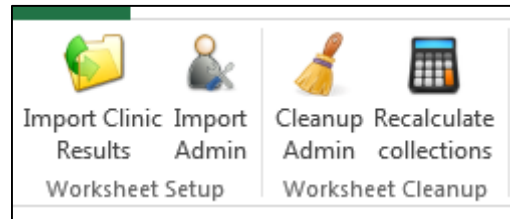
Once the data is imported and usable, the next step is to populate the profit-sharing sheets for clinic with the appropriate data from the imported spreadsheets. To do this, I created an index worksheet that referenced the worksheets related to each clinic. Then, the subs and functions I wrote looped through all the worksheets to pull data from the relevant worksheets to calculate profit-sharing numbers, such as the accrued retirement expense, clinic collections, and total company collections. For procedures that needed an input, I wrote functions; for procedures that did not, I created subprocedures. You can see how some of the functions are used below.

5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					
31					
32					
33					
34					
35					
36					
37					
38					
39					
40					
41					
42					
43					
44					
45					
46					
47					
48					
49					
50					
51					
52					
53					
54					
55					
56					
57					
58					
59					
60					
61					
62					
63					
64					
65					
66					
67					
68					
69					
70					
71					
72					
73					
74					
75					
76					
77					
78					
79					
80					
81					
82					
83					
84					
85					
86					
87					
88					
89					
90					
91					
92					
93					
94					
95					
96					
97					
98					
99					
100					

Objective #4: Making the sub user-friendly through a custom ribbon

One of the most important objectives of this project was to make the macros user-friendly, so the users could use the macros without much confusion. The primary way I accomplished this is by

providing a custom ribbon tab entitled “CPT”. On the custom tab, there are four buttons: “Import Clinic Results”, “Import Admin”, “Cleanup Admin”, and “Recalculate collections”. The first three buttons are used to get the data into the master file. The last button is used to recalculate the profit sharing because several of the subprocedures hard-code the data into the cells.



Learning and Conceptual Difficulties

This project caused me to learn a great deal about programming for end users. This class has been my first significant experience to VBA coding. Accordingly, the learning curve during the project was fairly steep. Unlike the structured projects we have completed before this project, I had to conceptualize what I wanted to accomplish and make adjustments along the way. For example, I wrote functions for every key data point for calculating the profit-sharing. After meeting with Dr. Allen, I learned that some of functions were better accomplished as a sub that hard-coded the data into the cells.

Additionally, I had to adjust some of my code to consider what would be the easiest to use for a user who was not computer savvy. Thus, I added customized buttons, and made subs that called other related subs so the amount of steps needed to complete the profit-sharing process was as easy as possible.

In terms of actual VBA coding, I became much more familiar with how to manipulate objects, such as shapes in a workbook. I also learned how to create For Loops inside of other For Loops. Furthermore, I cemented how to use VBA to import data from other worksheets. I also reinforced how to control the Excel ribbon using the Custom UI editor. Finally, I learned how to create logic that would work well for functions or subprocedures.

The toughest difficulty I encountered was getting enough information from the billing office manager and accountant, because they were both hard to contact. Some of the information used to calculate the profit-sharing was convoluted because it involved multiple accounts, so I did not always understand where all the data should be pulled from. Because the accountant had not explained to me where she finds the therapist charges, I was not able to write code that calculates the individual therapist charges. However, I plan on writing a subprocedure to find that last data as soon as the account responds to my emails.

Assistance

I received some assistance on my project. Nathan, the VBA TA, helped me write code to import the worksheets from other Excel workbooks. He and Dr. Allen also helped me talk through an issue with one of my functions. I learned that the problem with one of the variables in the function (that would not hold a value), was that my function was set to Application.Volatile. This

caused it to recalculate every time and deleted the placeholder variable. I fixed this by writing a subprocedure that did the same thing and hardcoded the data to a cell.