

All revenue, company, and client data has been changed to try to ensure company and client information is kept confidential

EXECUTIVE SUMMARY

I work for a local company that develops and licenses software for academic institutions and corporations throughout the world. The software is hosted online meaning that users purchase access to the tool for a specified period of time. Most transactions occur annually and the user pays for their access at the beginning of its license. The organization has experienced intense growth over the last few years. A key part of its business model is recurring or renewal licensing. The accounting department is responsible to keeping track of upcoming renewals and responsible to confirm renewals with sales representatives before sending invoices to clients. In prior years the task was simple and required relatively little time. However over the last year the number of transactions and the revenue generated from renewals has grown substantially. Additionally the workforce has more than doubled and there are no longer only a few sales representatives to speak with.

My VBA project reduces the amount of time for the accounting team to generate and deliver individual reports for all sales representatives through automating the process. Each sales representative has its own worksheet automatically generated with all relevant information presented in an organized matter. The solution is meant to be simple and intuitive as individuals with varying degrees of Excel experience will use the Workbook. The project has already been implemented and has reduced the time required to complete the task from two-three hours each week to just a few minutes.

THE PROBLEM

In order to implement a solution that could provide benefits to the accounting team it is necessary to understand the problem and task the accounting team was charged with. As sales representatives built their customer base their amount of renewals naturally grew. Because sales representatives predominantly focus on acquiring new clients renewing customers were sometimes neglected or not contacted about their upcoming renewal until the last possible moment. Consequently renewal invoices were often sent late or not sent at all. This was not acceptable because recurring revenue is what makes the business models for internet, subscription-based companies. The company's CEO asked the accounting team to print and tape a list of upcoming renewals to each sales representatives computer monitor each week. One of the accountants is then charged with meeting with and discussing upcoming renewals with the sales representatives. This has helped us send most invoices on time and ensure continued growth for the company. However because the company has doubled for each of the past three years the renewal list has grown to represent a very large number of transactions.

To complicate matters the accounting software does not easily allow users to see necessary information about upcoming renewals. Sales orders and invoices can be 'memorized' and scheduled to automatically generate in a certain time period. However we are unable to generate a report to know which sales representatives are responsible for which memorized transactions. Consequently we have had to create a report through Excel that combines a few different reports from our accounting system so that we can have the information needed in one place. An accountant would then essentially copy and paste the

different upcoming transactions for each sales representative into different worksheets and then print out each of those sheets and deliver them to the sales representative. This process was repeated each week and required at least two-three hours to get everything organized to print. Obviously this was tedious and required too much time, but it also left a lot of room for error. Because most transactions are for several thousands of dollars mistakes were expensive and needed to be limited as much as possible.

THE SOLUTION

I have broken up my solution to the problem in several parts. Each of these parts is meant to represent a logical process necessary to deliver the solution. These parts include the following:

- 1. The Summary Sheet and Launching Pad
- 2. Creating New Sheets for Each Sales Representative
- 3. Populating Information into the Individual Sheets
- 4. Computing all of the Information on the Home Page
- 5. Printing the Reports
- 6. Clearing all of the tabs except for the Summary and Export worksheets

THE SUMMARY SHEET AND LAUNCHING PAD

I knew it was important to have the home screen be intuitive and adaptable to accommodate the change the company would continue to experience. Consequently, the solution only requires two sets of information, [1] the individual sales representatives and [2] all the information regarding upcoming transactions. As mentioned earlier I did not want to make the process more complicated than necessary – so the Summary worksheet begins with just a list of sales representatives. This list can be updated by adding, deleting and editing the list of sales representatives. The summary sheet also provides summary information regarding the amount of upcoming renewals and the portion of the upcoming renewals that are confirmed for invoicing. This provides a valuable report to present to management each week.

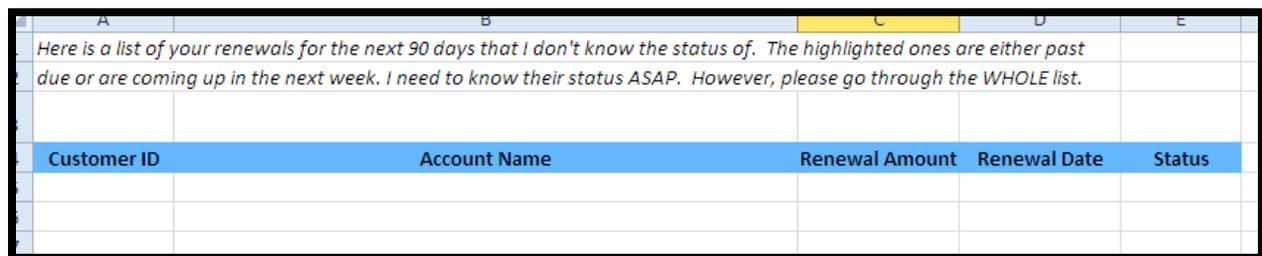
	A	B	C	D	E	F	G	H
		Renewals Upcoming	Renewals Confirmed	% Confirmed		Start Over	Run Report	Print Reports
1	SEANS							
2	MARCOL							
3	BRYCEW							
4	RYANS							
5	STUARTO							
6	CHRISA							
7	DWATKINS							
8	JHARVEY							
9	MICHAELM							
10	SCOTTA							
11	ANDREW P							
12	KENK							
13	RYANC							

The workbook also opens with one other sheet entitled 'Export' which contains the exported data from our accounting system about all upcoming renewal transactions for the next several years. This data is prepared by an accountant on a weekly basis and only involved copying and pasting information obtained from a few exports from the company's accounting system.

CREATING NEW SHEETS FOR EACH SALES REPRESENTATIVE

I needed to create a different worksheet for each sales representative listed on the home screen and name each worksheet according to the sales representative's name. In order to do this I coded the macro to begin at the first sales representative in the list and then it began the process of naming and formatting each sheet until it reached an empty cell in the list. The selected sales representative was defined as 'SheetName' and then I used the code `Sheets.Add.Name=SheetName` to create and name each sheet. The code then formats each sheet to ensure the proper column width, headings, font formats and number formats.

The output of the loop would be a blank sheet that looked like the following:



The screenshot shows an Excel worksheet with a header row and a body of empty rows. The header row has the following columns: Customer ID, Account Name, Renewal Amount, Renewal Date, and Status. The first row of the body contains a message: "Here is a list of your renewals for the next 90 days that I don't know the status of. The highlighted ones are either past due or are coming up in the next week. I need to know their status ASAP. However, please go through the WHOLE list." The rest of the rows are empty.

Customer ID	Account Name	Renewal Amount	Renewal Date	Status
Here is a list of your renewals for the next 90 days that I don't know the status of. The highlighted ones are either past due or are coming up in the next week. I need to know their status ASAP. However, please go through the WHOLE list.				

POPULATING INFORMATION INTO INDIVIDUAL SHEETS

Populating the information into individual sheets was not as simple as using a sales representative search function and then populating all the necessary information into each individual worksheet. I was not interested in showing all renewals for the next several years for sales representatives. Instead I wanted to only display renewals that are either past due or upcoming in the next 90 days that have not been confirmed. I used a Do Until Loop within a For Loop to search and populate each of the sheets.

- *For SummRow = 2 To 300* – This started the For Loop at the first sales representative listed on the Summary worksheet and would work its way through the entire list until it reached a blank cell. At that point there was an Exit For statement.
- *Do Until EndList = True* – I started the search at the first row of the Export worksheet and then wanted it to search through each row until it reached the end of list, at which point *EndList=True*. At this point the search would return to the top of the Export worksheet and begin the next sales representative search.

I used three different `Instr()` functions within the Do Until Loop to look for applicable data to bring into each sheet.

1. *If Instr(1, UCase(Sheets("Export").Cells(r, 6)), UCase(SheetName)) > 0 Then*
 - a. This searched for the sales representative name in the Export worksheet

2. *If Sheets("Export").Cells(r, 7).Value <= EDate Then*
 - a. We only want information for past due renewals and renewals upcoming in the next 90 days to appear on each sheet. In order to do this I declared the variable *EDate* as a Date and defined it as *EDate = DateAdd("d", 90, Date)*
3. *If Sheets("Export").Cells(r, 10).Value = "" And Sheets("Export").Cells(r, 9).Value <> "Void" And Sheets("Export").Cells(r, 9).Value <> "Confirmed" And Sheets("Export").Cells(r, 9).Value <> "Need PO" Then*
 - a. We only want information about renewals that have not been invoiced, cancelled or confirmed. In other words once a sales representative lets accounting know whether a renewal needs to be invoiced it will no longer appear on their respective renewal sheet.

Once all of the criteria were met then the individual row was populated with the required information and the search continued until it found all of the upcoming renewals for each sales representative. A typical sales representative sheet would look like the following.

Here is a list of your renewals for the next 90 days that I don't know the status of. The highlighted ones are either past due or are coming up in the next week. I need to know their status ASAP. However, please go through the WHOLE list.					
ANDREW P					
Customer ID	Account Name	Renewal Amount	Renewal Date	Status	
35512	COMPANY 552	48	12/1/2010	Pending	
34451	COMPANY 745	17	1/12/2011	Pending	
35233	COMPANY 918	94	2/23/2011	Pending	
34754	COMPANY 967	9	3/1/2011	Pending	
Christian Johnson - christianj@company.com - ext 6914					

COMPUTING ALL THE INFORMATION ON THE HOME PAGE

I then needed to make sure that all the information was correctly calculated on the Summary worksheet. Because the list of sales representatives is always changing it means that the formulas needed to be able to change based on the name and number of sales representatives. In order to do this I defined what the last row of the sales representative was by using the following code –

```
LastRow = Cells(Rows.Count, "A").End(xlUp).Row
```

This allowed me to make one formula in a cell and then use the AutoFill command to copy the formula down to the last sales representative. Defining the *LastRow* also allowed me to calculate the totals by doing *LastRow + 1* and then inputting the formulas in through the macro. The formulas I used were just variations of the sumifs formula to calculate the needed information. The remainder of my code worked to ensure all of the number and format formats made the report look aesthetically pleasing. After the report ran the Summary worksheet would look like the following.

	A	B	C	D
1		Renewals Upcoming	Renewals Confirmed	% Confirmed
2	SEANS	3,241	72	2.2%
3	MARCOL	716	0	0.0%
4	BRYCEW	2,821	349	12.4%
5	RYANS	2,572	313	12.2%
6	STUARTO	660	82	12.4%
7	CHRISA	2,579	454	17.6%
8	DWATKINS	2,794	2,195	78.6%
9	JHARVEY	1,551	134	8.6%
10	MICHAELM	346	255	73.7%
11	SCOTTA	49	0	0.0%
12	ANDREW P	181	13	7.2%
13	KENK	69	0	0.0%
14	RYANC	70	0	0.0%
15		17,649	3,867	21.9%
16				

PRINTING THE REPORTS

The accountants print off all of the sheets except for the Export worksheet. I used the following code to make sure each worksheet was printed.

For Each Sht In ThisWorkbook.Worksheets

If Sht.Name = "Export" Then Exit Sub

Because the code was already written so that the Export worksheet was last I knew that all sheets necessary would be printed. Because these printed reports are given to individual sales representatives, I included code to make sure the print settings fit to one page tall and wide. I also adjusted the margins for each printed report to eliminate as much white space as possible. I was able to accomplish this by sampling recording a macro, which worked rather well.

CLEARING ALL OF THE WORKSHEETS, EXCEPT FOR THE SUMMARY AND EXPORT WORKSHEETS

I thought it was beneficial to also ensure the user was always working with current data. In order to help facilitate this I wrote so sub procedure that executes when the workbook is first opened. It essentially deletes all of the worksheets, except those named Summary and Export. Additionally there is a button included on the Summary worksheet to run the same macro once the worksheet is opened.

For x = Sheets.Count To 1 Step -1

If Sheets(x).Name <> "Summary" And Sheets(x).Name <> "Export" Then Sheets(x).Delete

SUMMARY OF OPERATIONS

I included three buttons to facilitate the solutions: [1] Start Over, [2] Run Report, and [3] Print Reports. An explanation of what each button does is below.

When the *START OVER* button is clicked then all of the sheets are deleted except for the Summary and Export worksheets. All formulas are also deleted on the Summary worksheet.

When the *RUN REPORT* button is clicked then it first deletes all the worksheets except for the Summary and Export worksheets. Then it adds all of the worksheets for the sales representatives. Then it populates all of the individual sheets with the necessary information. Finally the code executes and populates the Summary worksheet with the necessary formulas.

When the *PRINT REPORTS* button is clicked then all sheets except for the Export worksheet are printed to the local printer. All margins are changed and the page is formatted to fit one page tall and one page wide.

LEARNING AND CONCEPTUAL DIFFICULTIES

Many of the difficulties were addressed previously in the report. But I have included a few additional small issues that I encountered as I worked through this final project.

There were several difficulties encountered as I did my project. At first I was trying to select certain cells in order to name sheets and other processes. However, when I did this I was using the *Activecell* command which ended up causing a lot more problems than it was worth. After consulting with Dr. Allen I abandoned all use of the *Activecell* command and my code was able to work much better.

I also ran into problems when I tried to calculate dates. As far as I remember we never did this in class and I did not know the format needed in order to make it work. However it was quickly addressed with an internet search.

CONCLUSION

I am admittedly not someone with a lot of VBA or Macro experience. This class was really my first introduction to macros of any kind. However, I look at the code written for the final project and can see huge progress made. I know there are much more complex issues that VBA is able to accomplish, however this project helped me solve a very real problem my company was dealing with. The accountant we pay to do this typically spends at least 3 hours a week on this report - this project allows it to be done in just a few minutes. The accountant's salary is \$35,000 and by better allocating his time we are able to save almost \$3,000. However the real savings is found by taking out much of the human element that was present before this solution. Within the first week that we used this report we were able to find over \$10,000 in renewals that had accidentally not been included on previous renewal reports through human error. So far my company loves the report.

If you decide to run the macro on your own computer all of the code should work appropriately, with the exception of the Print command