

## Executive Summary

The company that this workbook is defined for specializes in selling metal cutting and inspection equipment made by other manufacturers. As a distributor, it has salespeople who demonstrate how the equipment will enhance the operations of the potential customer. An example of the inspection equipment is equipment that flawlessly measures whatever surfaces it touches. The measuring “arm” moves easily, conforming to different angles and surfaces, producing flawless measurement. This might seem simple or even useless but these measurements are recorded and can be automatically input into CAD or other design programs. One practical use of these measurements in CAD is reverse engineering other equipment if necessary to know how they can customize the equipment to meet the specific needs of their business.

This company also provides support services including ordering spare parts, providing engineering resources, as well as helping buyers learn to use the equipment and troubleshooting any problems. In order to track one division’s quotes, demonstrations, and the resulting purchase orders, one salesperson asked me to automate a workbook they use to measure progress toward the division’s goals. The workbook now has the needed functionality.

The first user form allows the user to easily add potential transactions with corresponding names, salesperson, dates, products, categories, etc. Another user form allows the user to find previously entered transactions and update the progress of the transaction clear to the point of sale. Another function of the workbook is generating a report that counts how many demonstrations, quotes, and purchase orders were done each month and the total sales per month, per quarter, and for the year. These totals are then compared to the company goals automatically and a chart previously built updates to show the company’s progress so far.

Both of these user forms are accessible by clicking a button on the custom ribbon bearing the company’s name. Another button generates the report mentioned previously. Finally, the last button automatically saves a copy of the workbook under a name specified by the user and automatically emails each salesperson an attachment of the saved workbook copy as well as any message the user desires to be sent.

Discussion about learning and difficulties involved in the completion of the project are also included in this report. The company’s name has been omitted from this project completely and the data has been scrubbed to protect the identity of the company and its customers. The products and companies listed in the workbook have nothing to do with this company.

## Implementation

*Adding New Potential Transactions:* I created user forms that allow the user to easily add new transactions. As can be seen by the screenshot on the left, the user can easily add a date, the company name, location, potential contract amount, and product. Also, the sales-personnel can select their name

The 'Add' form is titled 'Demo Date' and features a calendar for 'Apr 2013'. Below the calendar are several input fields: 'Company' (text box), 'Salesperson' (dropdown), 'Location - City' (text box), 'State' (dropdown), 'Contract Amount' (text box), 'Product' (list box with items: Rifle, Shotgun, Machine Gun, Missiles, Gernade Launcher), and 'Category' (dropdown). At the bottom are buttons for 'Add Product', 'Save', 'Delete Product', and 'Cancel'.

from the dropdown box that populates based on a list of salespeople in the company that is on a separate sheet. This allows the user to easily add or delete salespeople as they are hired or terminated. The dropdown box automatically populates. The same is true with the state and category dropdown boxes (which is used to track the categories of products). This way the user does not have to change the VBA code at all avoiding errors. The list box under product is populated by a list on a separate sheet as well. As can be seen, products can be easily deleted or added. From what I understand of the business model of this company, once a potential customer or transaction is identified, a demonstration of the product and its capabilities is done at the customer's location. Sometimes the customer orders the product right at the demonstration. Sometimes the purchase comes later. Whenever a quote is given, this second user form allows the user to update the original entry adding a quote date, purchase order date, and follow up date.

The 'Record Found' form displays the same fields as the 'Add' form but with pre-filled data: 'Salesperson' is 'Scott', 'Category' is 'Tracker', 'Amount' is '9,876.66', 'Company' is 'John's Hut', 'Location - City' is 'John's Hut', 'State' is 'South Jordan, AZ', and 'Product' is 'Shotgun'. It includes buttons for 'Find', 'Next Record', 'Add Quote Date', 'Add Purchase Order Date', 'Add Follow up Date', 'Save', and 'Cancel'.

*Finding and Updating the Customer Record:* The user can search for the original record by entering the company name, contract amount, or location of the customer. If the company is found, the values for each field as currently recorded is listed to the right as can be seen in the screenshot to the left. Scott is listed to the right of the salesperson dropdown box. If the record is not found, a message box pops up saying that the record is not found. If

The 'Add Purchase Order Date' form is titled 'Purchase Order Date' and features a calendar for 'May 2013'. It has a 'Save' button and a 'Cancel' button.

find is clicked before search terms have been entered, a message box appears alerting the user of the need of search terms. The user can keep the values already recorded by leaving the text boxes empty or update the values by filling in the text boxes or selecting new values in the dropdown boxes or in the list box. A calendar

showing the date that was originally selected by the user appears as well, which can be updated easily simply by selecting a new date.

The user can also add dates by clicking on one of the “Add \* Date” buttons on the Find/Update user form. When a button is clicked, this user form or one just like it for a different date displays allowing the user to easily select a date. Once the user clicks on the date they want and click save, the date shows up



on the Find/Update user form as can be seen on the second screen shot. Then, once the user clicks save, this updates the record in Excel,

adding a date and replacing any old values with the updates made by the user. I really like the functionality of the calendars. It makes choosing a date easy and intuitive and avoids having to enter dates in the date format (m/dd/yyyy) typing slashes, which I find annoying. This is done automatically once save is clicked.

4/1/2013									
J	K	L	M	N	O	P	Q	R	
View up	Performance	#Demos	#quotes	#sale	Total				
	Jan-13	12	9	0	\$ -				
	Feb-13	9	9	3	\$ 456,303.00				
	Mar-13	18	15	17	\$ 2,257,613.74		\$2,713,916.74		
	Apr-13	6	3	5	\$ 348,841.49				
	May-13	0	0	0	\$ -				
	Jun-13	0	0	0	\$ -		\$348,841.49		
	Jul-13	0	0	0	\$ -				
	Aug-13	0	0	0	\$ -				
	Sep-13	0	0	0	\$ -			\$0.00	
	Oct-13	0	0	0	\$ -				
	Nov-13	0	0	0	\$ -				
	Dec-13	0	0	0	\$ -			\$0.00	
	<b>totals</b>	<b>45</b>	<b>36</b>	<b>25</b>	<b>\$ 3,062,758.23</b>			<b>159%</b>	
		<b>4</b>	<b>3</b>	<b>2</b>	<b>per month</b>			<b>109%</b>	
					<b>2013</b>	<b>\$ 1,925,440.65</b>			
					<b>Total Sales</b>	<b>23</b>			
<b>R</b>	Romer Arms	9			<b>Hexagon Goals</b>				
<b>C</b>	CMM's	9			<b>\$ 12,000,000.00</b>	<b>Total Sales</b>			
<b>A</b>	After Market	0			<b>\$ 3,120,000.00</b>	<b>Sales per Quarter</b>			
<b>V</b>	Vision	3							
<b>T</b>	Tracker	4			<b>Mountain Team Goals</b>				
	<b>Total:</b>	<b>25</b>			<b>58 Hexagon sales</b>				
					<b>\$ 3,120,000.00</b>				
<b>A</b>	Andrew	3							
<b>B</b>	Brad	12							
<b>J</b>	Jim	0							
<b>R</b>	Bob	3							
<b>S</b>	Scott	7							
	<b>Total:</b>	<b>25</b>							

As can be seen here on the ribbon screen shot above, I added a new ribbon (named Dartgig on this spreadsheet to scrub the data and protect company information). This contains buttons for the add and find/update user forms already described as well as buttons that generate a report and a button that emails the report.

*Generate Report:* When the user clicks on generate report, the report shown in the screenshot to the left is updated based on the entries recorded in the spreadsheet. For each month, the number of quotes, demonstrations, sales, and total sales are calculated based on the respective dates of demos, quotes, and

purchase orders recorded for each transaction. Totals are then calculated at the bottom of each column, and averages are calculated for the number of quotes, demos, and sales. Sales are then divided by Total sales goals and the total dollar amount in sales is divided by the 2013 goal. As can be seen in column “Q”, totals for quarters are also calculated. There is a chart that is tied to these numbers that is not shown in the screen shot. It updates automatically without automation because it is tied to the “Total” column range as well as the months range. This report allows the user to instantly see how the company is doing in reaching its goals based on all the transaction records. In addition, all information in the transaction record is highlighted so it stands apart from the transactions still in process. This button also updates totals for each salesperson as well as each category of product as can be seen in the small boxes in the bottom left of the screen shot.

	A	B	C	D	E	F	G	H	I	J	K	L
1	<b>When Sent?</b>	<b>Name</b>	<b>Email address</b>	<b>Subject</b>		<b>Message</b>						
2	4/17/2013 9:26	Andrew	<a href="mailto:engliskevlar@gmail.com">engliskevlar@gmail.com</a>	Sales report as of April 16, 2013		Hey <person>,						
3	4/17/2013 9:26	Brad	<a href="mailto:engliskevlar@gmail.com">engliskevlar@gmail.com</a>			Here is our sales report as of today. See the attached file and enable macros.						
4	4/17/2013 9:26	Jim	<a href="mailto:engliskevlar@gmail.com">engliskevlar@gmail.com</a>	<b>File Name</b>								
5	4/17/2013 9:26	Bob	<a href="mailto:engliskevlar@gmail.com">engliskevlar@gmail.com</a>	April 17 Sales Report								
6	4/17/2013 9:26	Scott	<a href="mailto:engliskevlar@gmail.com">engliskevlar@gmail.com</a>									
7												
8												

*Emailing the Report:* When the user clicks the Email button, a copy of the workbook is saved in the same folder on the user's hard drive that the active workbook is saved in. Excel names the copy of the workbook whatever is in the cell under **File Name** as shown in the screen shot above. Then a user form is displayed prompting the user to enter his or her email username and password. Before clicking the Email button on the ribbon, the user can enter whatever message he or she wants in the body of the email under **Message** and whatever he or she wants the subject to be in the appropriate range. The emails are automatically sent to the email address of each salesperson. The time the message was successfully sent is listed or if the email fails to send, the box displays "Failed." I used the code from the workbook for the class where this concept was covered, modifying it to meet the needs of this workbook.

### Discussion of Learning and Conceptual Difficulties Encountered

To complete this project, I had to learn how to work with the date functions in VBA. I had not worked with these before but figured it was the best way to accomplish the task of counting the total number of dates for each category (demo, quote, and purchase order) for each month. I first started with the month function. I used the INSTR function to compare the month in the report to the month for each respective date type. After finishing this code, I noticed on the report that several quotes and purchase orders were showing up for December. That did not make any sense to me because none of the dates were in December. The only cells that could have been counted that were not under different months were blank cells. I used the immediate window to see if blank cells, when entered into the function, would return a value for month. I was surprised to see that blank cells in the month function produced values of 12 (that of December). Understanding this made the fix easy. I was able to add another constraint using the year function (year(date)) along with the month function to get my code to execute in the manner I desired.

Another section that I had not experienced before was figuring out how to save my workbook before I attached it to my email so that I could include any changes made to the workbook. It saved correctly but then would not send because apparently using the .Save method does not allow a workbook currently being used to be attached. So then I tried using the .SaveAs method and saving it under another file name. However, when I did this, I had the same error as before because it saved my current workbook under that name so it was in use when it was being attached and would cause an error. Finally, I tried the .SaveAsCopy method that allowed me to save a copy of the workbook in the same folder as the active workbook using the same path (thisworkbook.path). One argument of this method is the file type. I could not figure out which file type from the index of file types I found online was a macro enabled workbook. Not knowing this and not specifying the file type created a file of an unknown type. Finally, I tried making the file name include .xlsm at the end and that actually worked. It created a file name that I guess told the interpreter what type of file to save the copy as. That was a relief.

It is always amazing to me how these projects can be never-ending black holes. Improvements can always be made. Doing this and the other projects increased my respect for software engineers designing major software such as Windows. As I discovered, predicting everything the user to do is hard and time-consuming but is necessary to make sure the program works as intended

### **Assistance**

I did not receive substantial help from anyone on my project. I asked a few questions but mainly just went to the internet with my questions because people did not know how to answer my questions most of the time.

### **Conclusion**

This project definitely showed me the importance of thinking through the logic behind the code I was writing. At one point I could not figure out a way to do something without copying and pasting the code over and over again. Later as I was going to bed, I realized what I needed to make the loop I needed work and avoid all the redundant code. I believe this project is a good demonstration of principles I learned in this class.