

## **VBA Final Project – Automating Studio Emails**

### **Executive Summary**

My wife, Shirley, has been teaching piano for almost fifteen years. It has only been the last four years that she has asked me to help with billing her students. This includes figuring out how much each student owes, keeping a ledger of the balance due for each student, and sending out emails to students parents reminding them of the amount due for the coming month. Shirley's studio has ranged anywhere from 15-30 students. Sending emails to each of the parents became a very tedious and time consuming process. Automating the process and making it such that special messages could be included if desired became a very attractive task. I estimate that by automating the email process I will save approximately 30 minutes each month.

## The Template Email

The first step in creating this VBA application was to create a template email that could be modified based on who the email is sent to, the amount that is due for the month, a summary of the charges for the month, the first name of the parent, and any special message that might need to accompany the monthly email. Below is a picture of the email:

---

*<p>Dear <NAME>,</p>  
<p><SPECIALMESSAGE></p>  
<p>Below is a summary of this month's lessons. Please let me know if you have any questions.</p>  
<p>Balance Due: \$<AMOUNT> (Please pay this amount). Payments are due by the 10th of each month.</p>  
<PASTE>  
<p>Please remember that payments will now be accepted via PayPal or Venmo. If possible, please use either of these services rather than a check. For PayPal please send your payment to shirley.washenko@gmail.com and for Venmo please send payment to Shirley Jacob Washenko (user name: Shirley-Washenko). Please remember to pay with your checking account when using these services to avoid charges. Cash will no longer be accepted.</p>  
<p>Thank you,</p>  
<p>Steve Washenko.</p>*

Initially the email was drafted as standard text but due to complications needed to be changed to an HTML format to be able to incorporate a table. This was my first time to ever work with HTML.

As you can see from the template email there are certain tags such as <NAME> that call out what should be replaced. These tags will be replaced each time an email is sent with the appropriate information.

### Email Subject Line

The function sendmail asks for a subject line which is provided in cell A4. The code looks to see what is in this cell and supplies its value for the subject line for each email sent. Below is an example of a typical subject line:

	A
1	\$495.00
2	
3	Email Subject Line:
4	May Piano Lessons

### Username:

A userform was created to input the username and password for the gmail account where the emails will be sent from. The userform auto populates the username from cell A7. This was provided to convenience when sending out emails. Below is a snapchat of the userform.

The screenshot shows a dialog box titled "Google Credentials" with a close button (X) in the top right corner. The dialog contains two input fields: "GMail account:" with the text "shirleystudio1@gmail.com" and "GMail password:" with an empty field. At the bottom of the dialog are two buttons: "Cancel" and "OK".

## Special Message

Sometimes a special message needs to be sent with an email to let all the students know about something special going on. The template email has a space for this identified as <SPECIALMESSAGE>. The special message can be drafted in a section of merged cells at A10. The code looks at cell A10 and modifies the template email to insert the text of the special message. The code also needed to check whether A10 is blank and if is blank it deletes <p><SPECIALMESSAGE></p> from the template email thus eliminating those lines rather than leaving in extra space in the final email. Below is where the special message is drafted:

9	<b>Special Message:</b>
10	
11	
12	I look forward to seeing you
13	tomorrow.
14	
15	

## Send Email Button

A button was needed to activate the procedure for sending the email after selecting the lines that need to be included in the message. The subject line text, username text, special text, and the send email button are conveniently located in column A:A which is frozen so it scrolls with you as you move across sending out emails. Below is a view of these elements:

	A	AD	AE	AF	AG	AH
1	\$495.00	Washenko, Steve & Shirley	\$135.00	11	Washenko, Steve & Shirley	\$145.25
2		Transaction	Amount	Date	Details	Amount
3	Email Subject Line:	August Lessons 1	-\$20.00	4/1/2016	April Lessons: 4	-\$130.00
4	May Piano Lessons	Fall Semester Expenses	-\$10.00	4/1/2016	Encore	-\$15.25
5		Sep Lessons 4	-\$80.00			
6	Username:	Payment	\$110.00			
7	shirleystudio1	Oct Lessons 4	-\$80.00			
8		Payment	\$80.00			
9	Special Message:	4 Lessons	-\$80.00			
10		Payment	\$80.00			
11		4 Lessons	-\$80.00			
12	I look forward to seeing you tomorrow.	4 Lessons	-\$80.00			
13		Fall Semester Fee	-\$5.00			
14		Payment	\$165.00			
15		Mar Lessons 4	-\$90.00			
16	Send Email	Apr Lessons 2	-\$45.00			
17						
18						
19						

### Input a Selection

One of the most time consuming parts of sending out the monthly emails came from copying and pasting a selection of text from the ledger of the student into an email. For example, if a student had 4 lessons for the month of April and they were charged \$130 and there was a separate charge for a competition the student entered it would be copied into an email with a picture of the selection below:

4/1/2016	April Lessons: 4	-\$130.00
4/1/2016	Encore	-\$15.25

Figuring out how to get VBA to look through a selection of cells was by far the most difficult part of the project. I wasted a considerable amount of time trying to send the email by copying the selection using the following code "message.TextBody = body" in the sendgmail function. However, I did not realize that TextBody does not allow for pictures. After speaking with Professor Allen this line was commented out and the line "message.HTMLBody = body" was uncommented. Although the message is not able to include a picture using this method it did

allow for graphs to be created based on the selection. The code takes the above selection of text and goes through each cell starting in the top left and creates a table. The table appears in an email like the email below:

Dear Steve & Shirley,

I look forward to seeing you [tomorrow](#).

Below is a summary of this month's lessons.  
Please let me know if you have any questions.

Balance Due: \$145.25 (Please pay this amount).  
Payments are due by the 10th of each month.

Date	Details	Ammount
4/1/2016	April Lessons: 4	-130
4/1/2016	Encore	-15.25

This clearly shows the student what is being charged. This table replaces the text <PASTE> in the form email and is based on the selection the user makes when creating the email.

### Amount Due

After the above table is created the application automatically moves upwards to find the value that has the total amount due (see below circled in red). This amount is negative and shows - \$145.25 in excel. The application replaces the negative sign so the final result is \$145.25 and this amount replaces the <AMOUNT> tag in the template email.

11	Washenko, Steve & Shirley	\$145.25
Date	Details	Amount
4/1/2016	April Lessons: 4	-\$130.00
4/1/2016	Encore	-\$15.25

## First Name Input

After finishing the amount due the application offsets one column to the parent name and parses the name removing the last name leaving only the first name(s) of the parent(s). This value is then substituted into the template email using the tag <NAME>.

11	Washenko, Steve & Shirley	-\$145.25
Date	Details	Amount
4/1/2016	April Lessons: 4	-\$130.00
4/1/2016	Encore	-\$15.25

## The Email List and Email Address

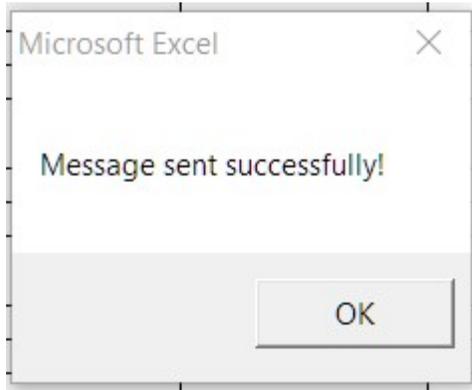
In order to identify the email address for the parent where the email will be sent it was necessary to create a separate worksheet called ContactInfo. ContactInfo contains the names of the parents with their email next to them as pictured below:

Parent/Family	Email
Washenko, Steve & Shirley	<a href="mailto:steve.washenko@gmail.com">steve.washenko@gmail.com</a>
Washenko, Steve & Shirley	<a href="mailto:steve.washenko@gmail.com">steve.washenko@gmail.com</a>
Washenko, Steve & Shirley	<a href="mailto:steve.washenko@gmail.com">steve.washenko@gmail.com</a>
Washenko, Steve & Shirley	<a href="mailto:steve.washenko@gmail.com">steve.washenko@gmail.com</a>
Washenko, Steve & Shirley	<a href="mailto:steve.washenko@gmail.com">steve.washenko@gmail.com</a>
Washenko, Steve & Shirley	<a href="mailto:steve.washenko@gmail.com">steve.washenko@gmail.com</a>
Washenko, Steve & Shirley	<a href="mailto:steve.washenko@gmail.com">steve.washenko@gmail.com</a>
Washenko, Steve & Shirley	<a href="mailto:steve.washenko@gmail.com">steve.washenko@gmail.com</a>
Washenko, Steve & Shirley	<a href="mailto:steve.washenko@gmail.com">steve.washenko@gmail.com</a>
Washenko, Steve & Shirley	<a href="mailto:steve.washenko@gmail.com">steve.washenko@gmail.com</a>
Washenko, Steve & Shirley	<a href="mailto:steve.washenko@gmail.com">steve.washenko@gmail.com</a>

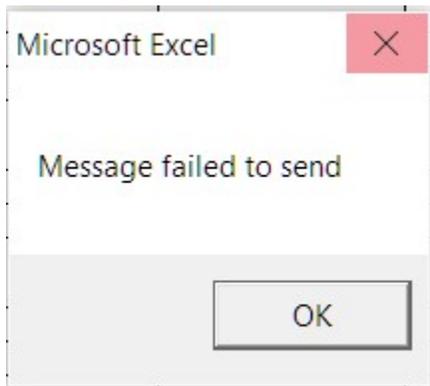
The code takes the current selection which is the parents name and matches it with the name in column A:A and then takes the adjacent value and inputs it into the email variable. It does this by using an excel function called VLookup. After returning a value for the email that value is stored in a variable called email which is used in the sendGmail function.

## The Email is Successful or Failed

By this point the email template has been built and modified and the sendgmail function is ready to go to work. After the function runs and returns a true Boolean a message box appears with the following message:



If the message failed to send a message box appears with the following message:



## Conclusion

I learned a lot from this project. The biggest lesson I learned came from understanding how to use HTML and that only HTML can contain pictures and tables. One thing I also learned is the importance of making the code flow together. VBA programming follows a list of items and it's imperative that the list is in the correct order to make everything work correctly. I estimate the project will save me about six hours a year in sending out emails to students.