A decorative graphic on the right side of the page. It features three overlapping circles of varying sizes, each composed of concentric rings in shades of blue. Two thin, light blue lines intersect at the top right and extend diagonally across the page, framing the circles.

Program to Generate Past Due Letters from an Age List

A Microsoft Excel VBA macro that generates a
letter in Microsoft Word

Brent Hendrickson
12/8/2011

Executive Summary

Entities that invoice need an efficient way to follow-up on unpaid invoices. The entity that I had in mind when I was creating this program was the university regulatory accounting office. This office invoices various sponsors for research related expenses and tracks the receipt of the corresponding payments. This organization in the past has had difficulty collecting in a timely manner. This program was designed to speed up the process of generating letters that urge clients to pay their invoices.

This program starts by bringing in all of the necessary information to generate these letters. This includes an age list and a list of addresses for the invoices that have been sent. The program also brings in other information so that the recipient of the letter can contact the proper administrator in charge of their account. The user will have the option to generate the letters using a user form that has been populated with inputs from the excel spreadsheets. Each of these inputs into the letter may be edited. The program repeats the process for each past-due invoice in the age list, but still allows the user to be in control of the letter creation process.

Occasionally, one invoice needs to be sent to multiple locations. The program has the ability to check the address list after an invoice has been generated, to see if a copy of the letter needs to go to an additional location. If no additional address could be found, a message box appears asking if they want to send the letter to another location by typing in a new address into the user form.

Table of Contents

Executive Summary..... 2

Implementation 4

 Setting up the program..... 4

 The flow of the program 4

 Generate Letter..... 5

 Next (Don't Create) 7

 Cancel..... 7

 When there are no more past due letters to be created 7

Learning and Obstacles Encountered 7

 Creating a Word document 8

 Formating the Word document 8

 Writing back to the database..... 9

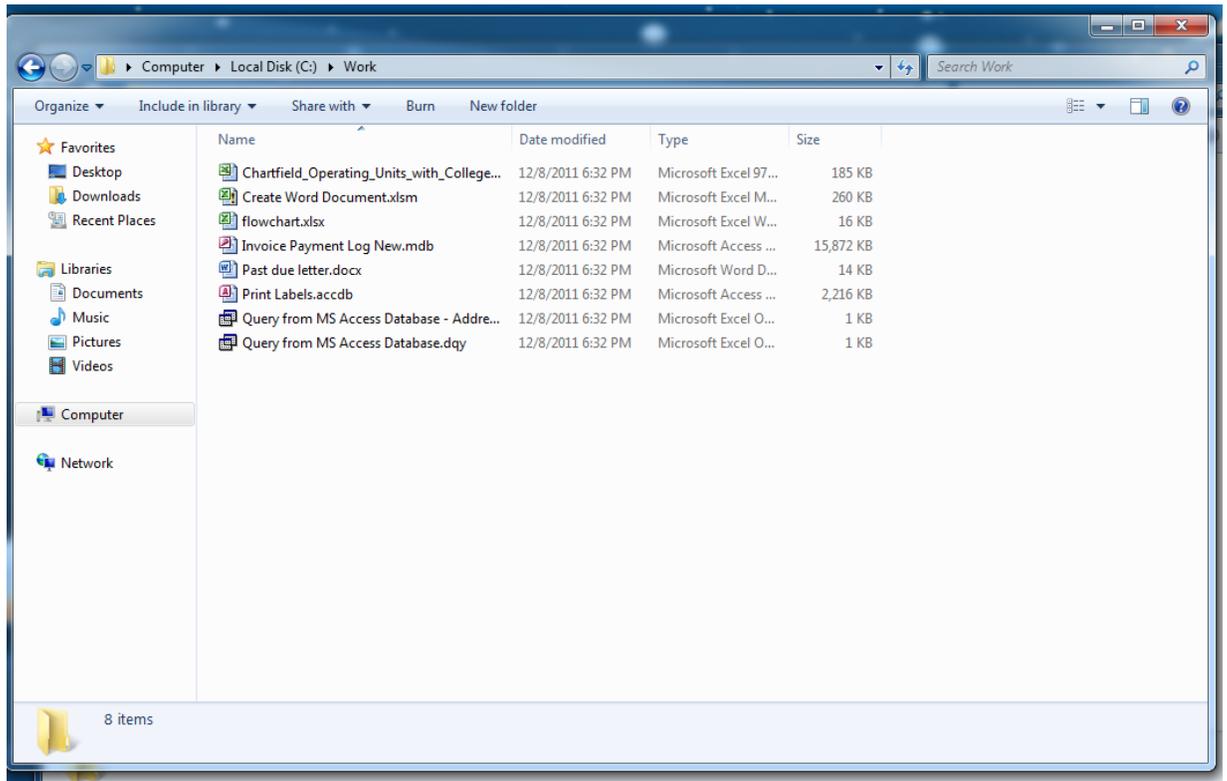
Letter with contract number 10

Letter without contract number 11

Implementation

Setting up the program

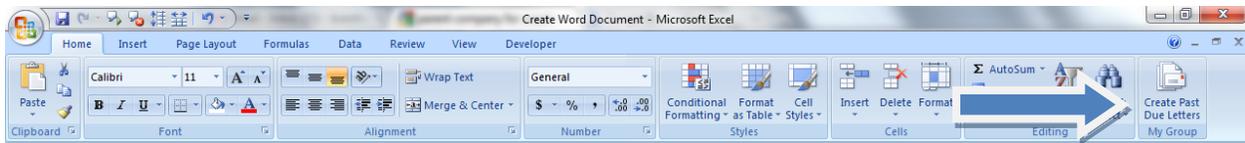
1. Download all the files and put them into a folder entitled “Work” on the “C-drive”. (C:\Work) The references in the macros assume that all of these files are in the above location.



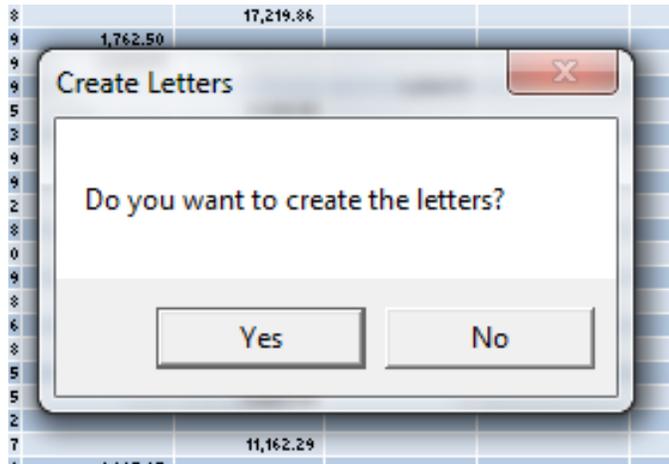
2. Open the excel document entitled “Create Word Document”
3. Enable all content and enable all data connections. This action allows access to the databases and Excel documents.
4. Add a reference to the Word-library if one does not already exist
5. Also create a folder in the C-drive named “Foldername”. The other option is you could edit the “filePath” variable in both the “withContractNo” and “withoutContractNo” macros. This action creates a location for the Word documents to be saved to.

The flow of the program

For easy access to initiate the program, I created a special button within the home ribbon entitled “Create Past Due Letters.”

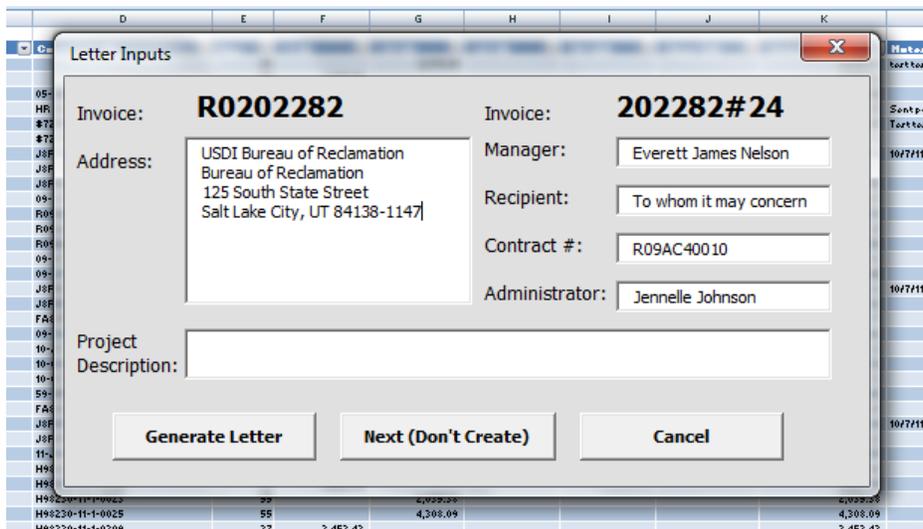


When the button is pressed, the macro creates data connections from multiple databases in order to bring in the age list and addresses. A message box is then displayed which gives the option to generate past due letters (if “Yes” is selected) or to simply just go to the age list (if “No is selected”).



Assuming the user presses the “Yes” button to create past due letters, this takes the user to a form which displays all of the inputs.

At this point, the user can edit any of the text boxes so that the letter inputs are entirely accurate and so that the user can get a sense of what will be going into the letter. I wanted the user to be in complete control of the letter generating process but still take advantage of all of



the pre-existing data that we already have access to. On the form, there are also three buttons entitled “Generate Letter”, “Next (Don’t Create)”, and “Cancel”. Each of these provides the user with unique options.

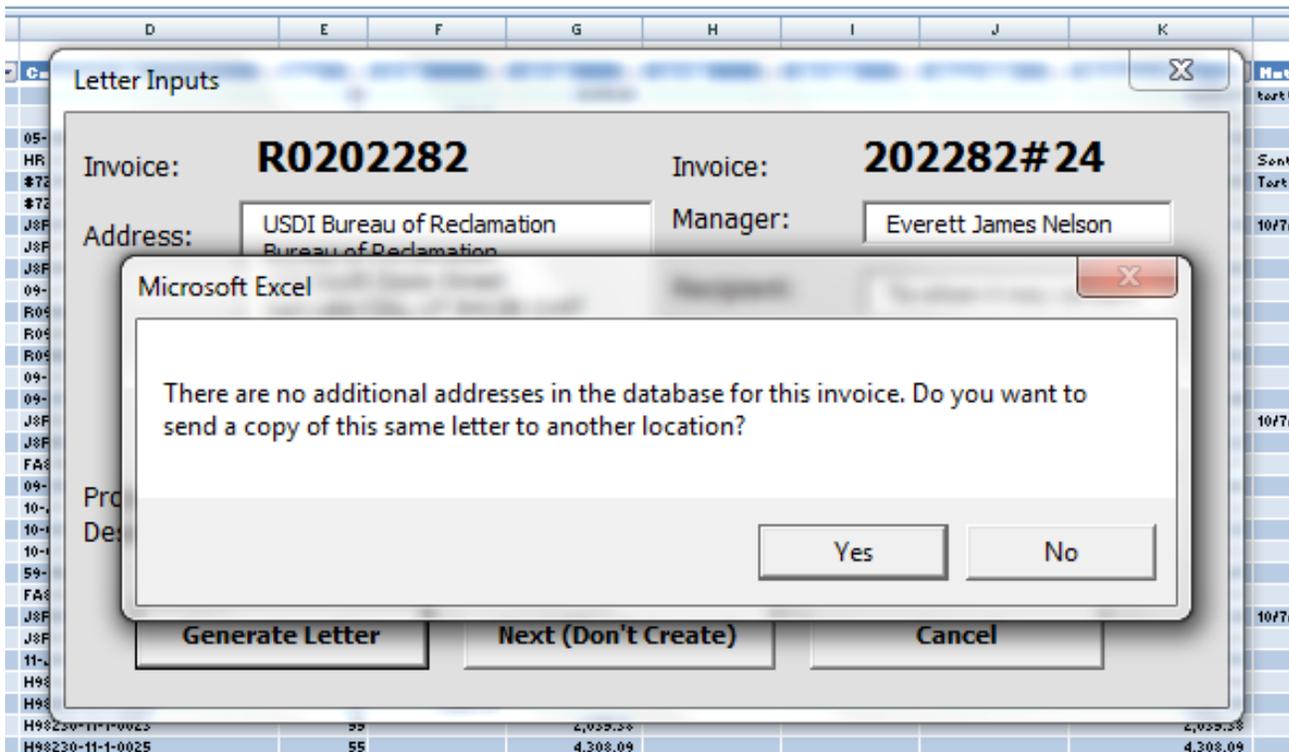
Generate Letter

Before pressing the “Generate Letter” button, there are several required inputs to create a letter. The address, manager, recipient, and administrator are all required. Purpose description is only required if there is no contract number (often there are no contract numbers). In this case where an invoice does not have a contact number associated with it, the wording of the letter will be changed. A simple IF statement will determine which letter needs to be generated. The letters could become as complicated as the logic that creates them. Below you can find the wording of both past due letters.

Every invoice has an administrator that works in our office that is in charge of following-up on the invoices. The letter is also populated with information so that the recipient can get in contact with the administrator. You can see those inputs in the letter. An IF statement determines what goes in these fields.

Once the letter is created, it will be saved as the invoice number concatenated with the date and time it was created. Keep in mind that the format of the date and time needed to be acceptable to be saved (can't save a date with a slash or a colon in it). For example if a letter for invoice 202282#24 was generated on December 8, 2011 at 9:05PM the letter would be saved as "7-December-2011 21-05-00". This effectively provides a time stamp for the creation of the invoice which will be useful for documentation purposes.

After the file is saved, it will check to see if there is another address associated with the invoice because sometimes, but rarely, past due letters need to be sent to multiple offices in order to receive payment. If an address is not found, it will prompt to see if you want to send it to another location anyways. If "Yes" is pressed, it will return you to the user form so that you can press generate with a new address that you inputted manually. If "No" is pressed, it will search for the next invoice that is past due.



Next (Don't Create)

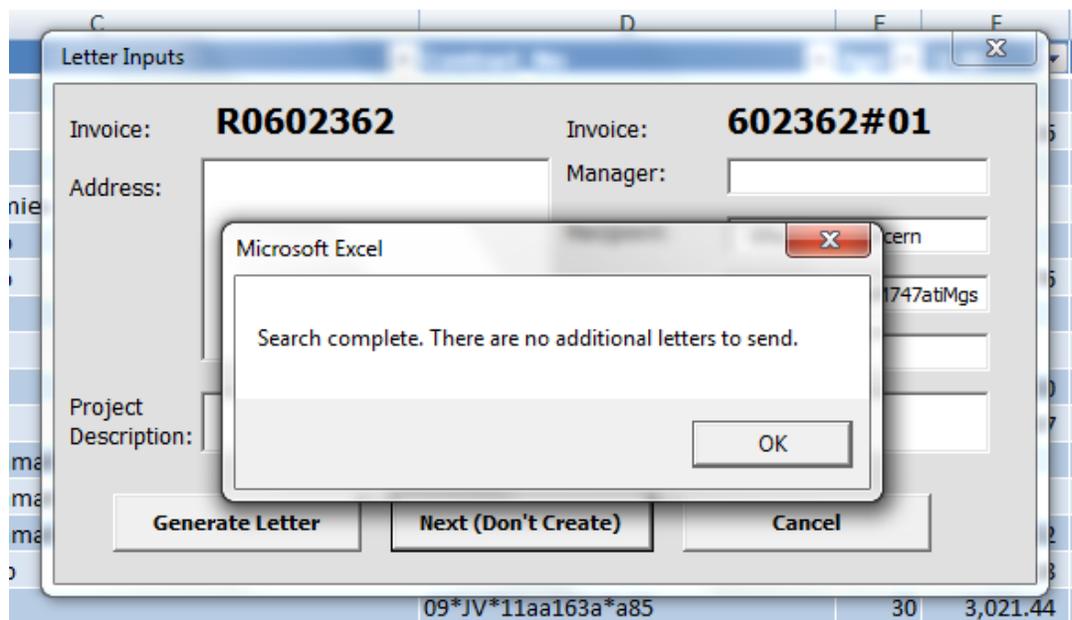
In the case that the “Next (Don't Create)” button is pressed, it will check to see if there is another address associated with the invoice just like at the end of the code for the “Generate Letter” button. If an address is not found, it will prompt to see if you want to send it to another location anyways. If “Yes” is pressed, it will return you to the user form. If “No” is pressed, it will search for the next invoice that is past due. The appearance of the prompt is exactly the same.

Cancel

The cancel button simply exits the user form and returns the user to the age list.

When there are no more past due letters to be created

When the program has cycled to the bottom of the age list and there are no more letters to be generated, there is simply a message box that informs the user that there are no additional past due letters to send.



Learning and Obstacles Encountered

It was especially enlightening to scour the internet and books to obtain answers to problems that other people were facing in the past. It felt really good to be part of a community that learns and grows from each other. I also got a better sense of what programmers do to find solutions and solve problems. There were several specific learning opportunities that caused a large amount of frustration and an equal amount of excitement when I was able to find the solution.

Creating a Word document

The First problem/learning opportunity that I faced was regarding the creation of a word document. To create a document I used the following code:

```
Set wrdApp = CreateObject("Word.Application")  
wrdApp.Visible = True  
Set wrdDoc = wrdApp.Documents.Add
```

Then to save the document, you simply use the following code:

```
wrdDoc.SaveAs (filePath)  
Set wrdDoc = Nothing  
Set wrdApp = Nothing
```

Formating the Word document

The Second, obstacle that I faced was regarding formatting the letter itself. The objects of a Word document obviously are very different than the objects in an Excel environment. For example, Word has paragraphs that contain writing, whereas Excel has cells. Word was a particularly annoying environment to work in and to format a letter because of line spacing.

Line spacing was an interesting problem because I wanted the address line to be formatted so that there would not be any lines spaces between in the address block. Word, for some reason, often assumes that after a “return” key is pressed that there should be a break in the paragraph by inserting a line, rather than simply breaking the line. I was able to figure out through scouring the internet that you could turn off (or on) the line break below the paragraph with “wrdApp.WordBasic.OpenOrCloseParaBelow”. This wasn’t good enough because I needed to just turn it off. In other words, if the word application was already set to off, this function would just turn it on. I also could not figure out the property associated with the above method.

Eventually I learned that there are many different types of “returns” or “enter keys”. There is a carriage return, a soft return (which does not add a line underneath it), then there is a blank line or some people call it white space.

- Chr(10) = blank line/white space
- Chr(11) = manual line break (shift-enter), or soft enter
- Chr(13) = carriage return

I basically removed Chr(10) and Chr(13) from the code and then inserted a soft enter and it made the letter formatted just like I wanted.

Writing back to the database

The last thing that is outside the scope of my project proposal, but I was trying to get the VBA program write information back to the database to make note of the fact that a past-due letter was generated on a certain date. The purpose of writing back to the database would be purely for documentation purposes so that we would not be sending out the same past due letters over and over. Currently, we manually enter into the database the fact that the letters were sent out and the date they were sent, but it would be nice if that was automated.

To try to implement this I found some very helpful information on the internet regarding writing back to the database. I just could not get it to work. I could set up the database connection with the following code:

```
Dim Cn As ADODB.Connection, Rs As ADODB.Recordset
Dim MyConn, sSQL As String
Set Cn = New ADODB.Connection
Cn.Open sqlQuery
```

“sqlQuery” was the variable that held my SQL query. I think my query had issues. I am confident that I will be able to solve the problem before long.

Letter with contract number

8-December-2011

<ADDRESS>

<RECIPIENT>,

Enclosed you will find a past due invoice for contract number <CONTRACT NUMBER> with our professor, <MANAGER>. For some reason we have not received full payment on invoice <INVOICE NUMBER>. Please send a check or wire transfer immediately for the stated amount. We are anxious to have this matter taken care of.

Please feel free to contact me at <PHONE> with any questions or concerns you may have. You may also email me at <EMAIL>.

Sincerely,

<ADMINISTRATOR>

Letter without contract number

8-December-2011

<ADDRESS>

<RECIPIENT>,

Enclosed you will find a past due invoice for the "<PROJECT DESCRIPTION>" project with our professor, <MANAGER>. For some reason we have not received full payment on invoice <INVOICE NUMBER>. Please send a check or wire transfer immediately for the stated amount. We are anxious to have this matter taken care of.

Please feel free to contact me at <PHONE> with any questions or concerns you may have. You may also email me at <EMAIL>.

Sincerely,

<ADMINISTRATOR>

Program Flowchart

