Directions for Use

I have included three files. The first file is the actual Add In that I created that you will want to download and install. The second file is an example commission statement that will need to be downloaded to see the functionality of the project. The third file is a copy of the workbook that includes the code as well as the XML in case you need to review it for grading purposes.

1. Download and Install the – Final Project Custom Ribbon (add in).
2. Download and open the – Final Sample Commission Statement (excel file).
3. Under the AdminAssist Tab in the ribbon, click the “Format All” button.
    a. This is a combination of Print Ranges, Name Sheets, and other settings.
    b. This will change all the settings for however number of sheets there are.
4. Click “Print Menu”
    a. This launches the custom print options dialg box.
    b. Select a few names in the multi select box, then click “Print Selection”. A dialog box appears to confirm. If you click yes it will begin printing.
    c. WARNING: If you click Print All – it will start printing immediately.
5. Click “Statement”
    a. This automates the statement access. Again, we only did it for Bank of American Fork due to their desire to have me focus my time in other areas.
6. Highlight some numbers and click “Accn. Format”
    a. This is a custom format that the accounting firm uses.
7. There are other functions included in the ribbon that the accounting firm uses.

Executive Summary

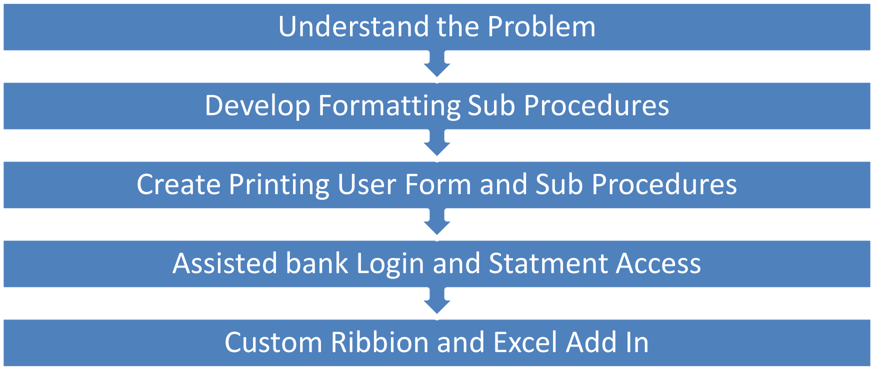
For my VBA project I spoke with my brother-in-law who is a partner in an accounting firm that specializes in payroll, accounting, benefits, hr, and corporate design. I have spoken to him multiple times helping him with his questions regarding excel.

At the accounting firm, they have multiple tasks that are very repetitive and time consuming. These issues include renaming up to 30 sheets in one workbook, setting the print areas for each sheet, and getting bank statement information.

The solution I created for the firm was creating a custom AdminAssist ribbon that stores their most used excel features. Likewise, the custom ribbon has buttons that will initiate VBA sub procedures that assist in formatting and printing their commission statements created monthly for clients. These sub procedures include renaming sheets, defining print areas, and launching a custom print dialog box.   Finally, the ribbon has a button that will log into a client's bank and download their monthly bank statements.

Implementation

To complete the project I accomplished the following steps:



Understand the Problem

Develop Formatting Sub Procedures

Create Printing User Form and Sub Procedures

Assisted bank Login and Statment Access

Custom Ribbion and Excel Add In

Understand the Problem

An important factor in completing my project was to develop a firm understanding of key issues before jumping into writing the actual code. The firm has an accounting system that keeps track of the information for clients.   They can then export any of the information they need into excel for analysis and editing.

The biggest issue in the project was their client's commission statements. When exported from the system, each company has its own workbook for the month.   Each worksheet in the client's workbook holds payout information for an employee. The exporter does not standardize the information very well. Although the appearance is professional, some of the columns and rows will change on each export. These variations needed to be accommodated in the project.

Lastly, it was important to understand that they would not need a lot of visual formatting i.e. creating headers, sub totals, custom colors. However, they did need sub procedures that could handle a lot of the settings and printing. Therefore, when the sub procedures are ran, it is not an “Ah Ha” moment for the typical observer. Simply, it is a lot of time saved for the accountant assigned to the project.

Formatting Sub Procedures

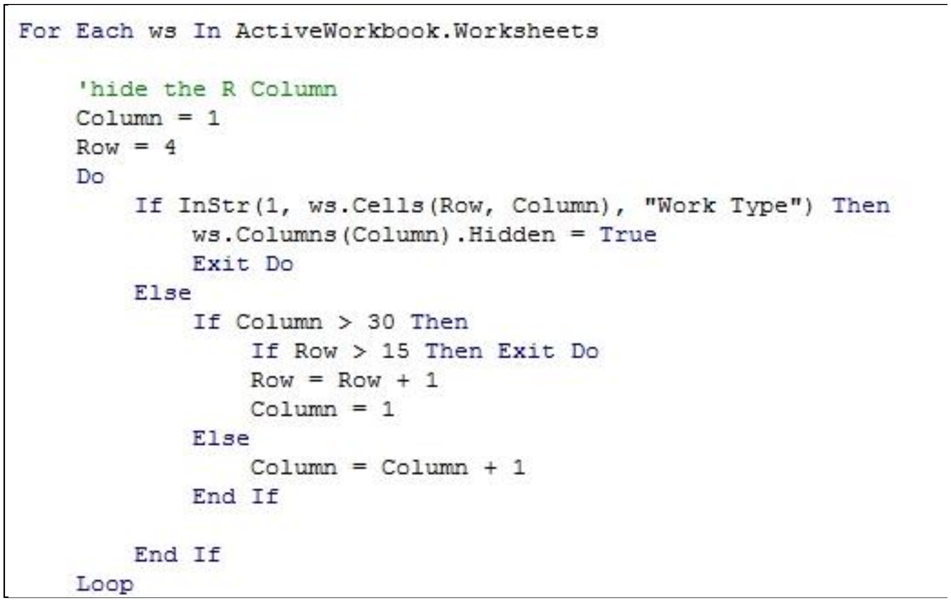
1. Name the Sheet

   As mentioned, each client will have workbook that contains commission statements for each employee. The last name of the employee would be the sheet name. The first step was to find the word “Contract” in the sheet. This was done using an instr() function. Once that word was found, offset one cell down and that is the location of the cell holding the individuals name.

   By using a combination of the Len(), Mid(), and Instr() function, I could use VBA to locate the last characters after the space in the cell. This takes out the last name of the individual. I then use a loop to rename the sheet for each sheet in the workbook. If there is a sheet that does not include a name, it sends a message back to the user that a “Specific Sheet” was not named. Typically this will happen on the last sheet of the workbook to inform the user if there was an issue in the export or that it has reached the summary tab.

2. Set Print Areas and Titles

   The next step in formatting is to set the print areas. When printing, the user will want to hide a column that contains the words “Work Type.” However, this is not always on the same row or column, but typically in the same block of data. ***Figure 1*** illustrates the code I wrote to accommodate this issue.



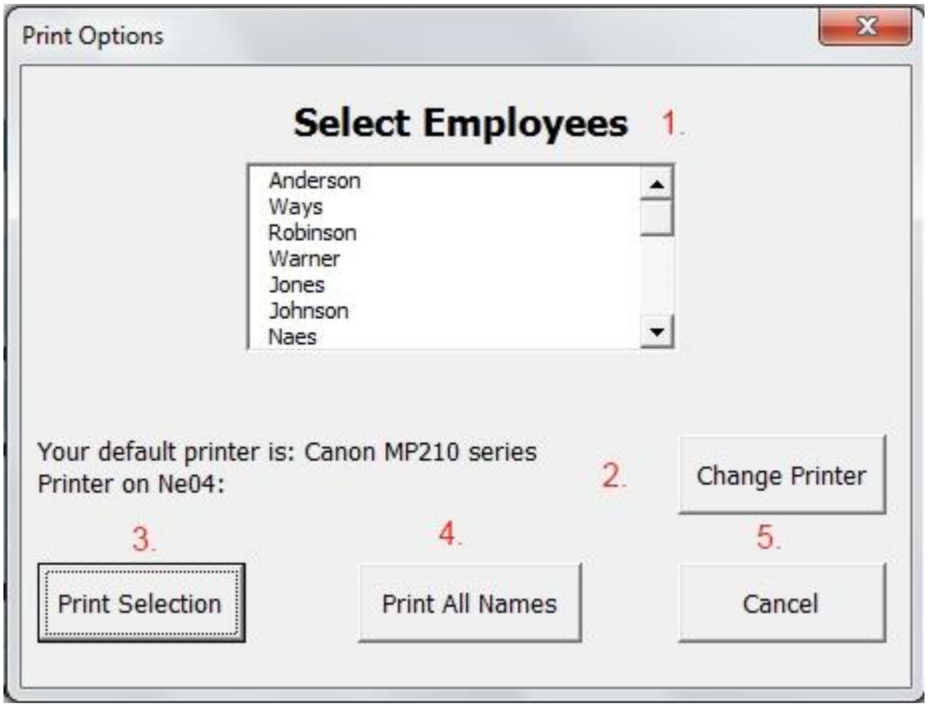
```
For Each ws In ActiveWorkbook.Worksheets

    'hide the R Column
    Column = 1
    Row = 4
    Do
        If InStr(1, ws.Cells(Row, Column), "Work Type") Then
            ws.Columns(Column).Hidden = True
            Exit Do
        Else
            If Column > 30 Then
                If Row > 15 Then Exit Do
                Row = Row + 1
                Column = 1
            Else
                Column = Column + 1
            End If

        End If
    Loop
```

Figure 1

In order to create to code to set the print area and create rolling print titles, I simply recorded a macro while I made those changes. I then removed the unnecessary information and set the print area and titles using the instr() function to locate specific information. A for each loop was then used to do this for each worksheet in the workbook.

Print User Forms and Sub Procedures

In order to make the sub procedures understandable and accessible by the user, I set up a User From to handle printing of the documents. The idea was generated when I was playing with the print options dialog box. I simply though it would be useful if I could customize my own. This is when I started creating the user form. The form is displayed in ***Figure 2***.



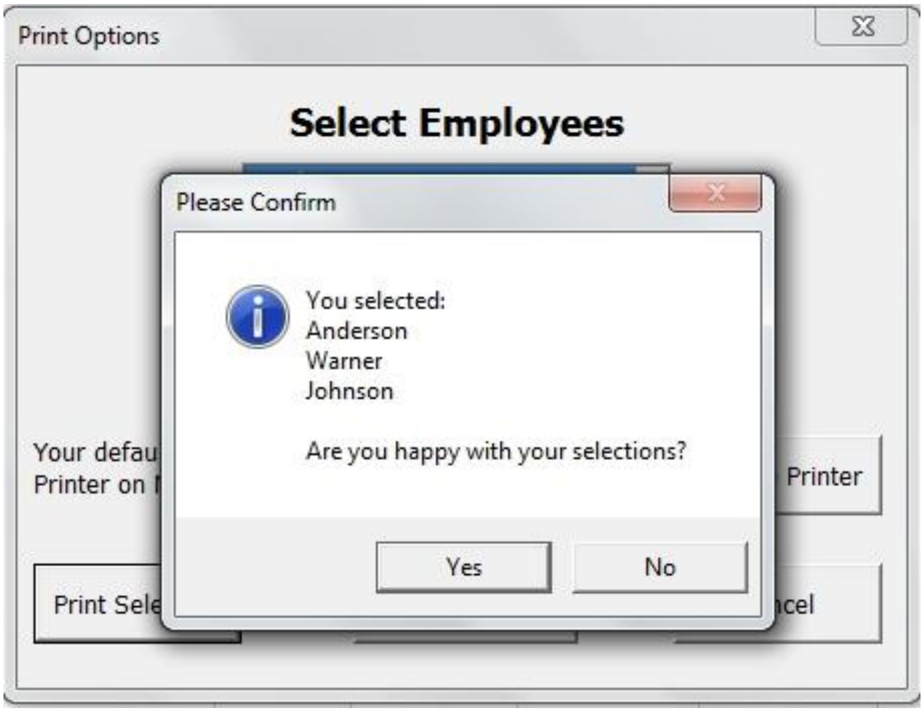
Your default printer is: Canon MP210 series Printer on Ne04:

Figure 2

1. Select Employees

   A main problem that I came across in the project is that the user had to manually print each sheet if they did not want to print all of them. Therefore, I created a multi select dialog box where the accountant can select multiple sheets (employee's last name) when they print the documents. I had to use Google to discover how to use a multi select box.

Once you have selected multiple employees and click "Print Selection" a dialog box will appear where you can confirm your selections ***(Figure 3)***. If you click "Yes" it will print the documents, if you click "No," it will take you back to the selection page.



Please Confirm

You selected:
Anderson
Warner
Johnson

Are you happy with your selections?

Figure 3

2. **Change Printer:** This was a more difficult part of the project. I created this information in case the user wants to change their default printer for these pages. I researched on Google for the VBA code that will allow you to launch the dialog box to display your default printer and allow you to change it. This allows the user to quickly make these changes without using the default Print Options dialog box.

3. **Print Selection:** This button will allow you to print only the highlighted employees that you have included in the box above. It will confirm your selection before it prints.

4. **Print All Names:** When clicked, this button will print all the sheets in the workbook.

5. **Cancel:** Unloads the User Form.

The code written behind Print All Names was simple to write using a for each loop. However, the Print Selection code was much more difficult to write. I spent a fair amount of time playing with the user form before I figured out how to write a for each loop that incorporates the selected names. In the end, this function of the project is the most useful and the accounting firm was amazed by its functionality.

Bank Login

In order to start this section, I asked the accounting firm for a bank they use and the credentials for login. Once they gave that information to me, I wrote the code to access the page of the bank statement information for a particular bank. I then created the login User Form displayed in ***Figure 4***.

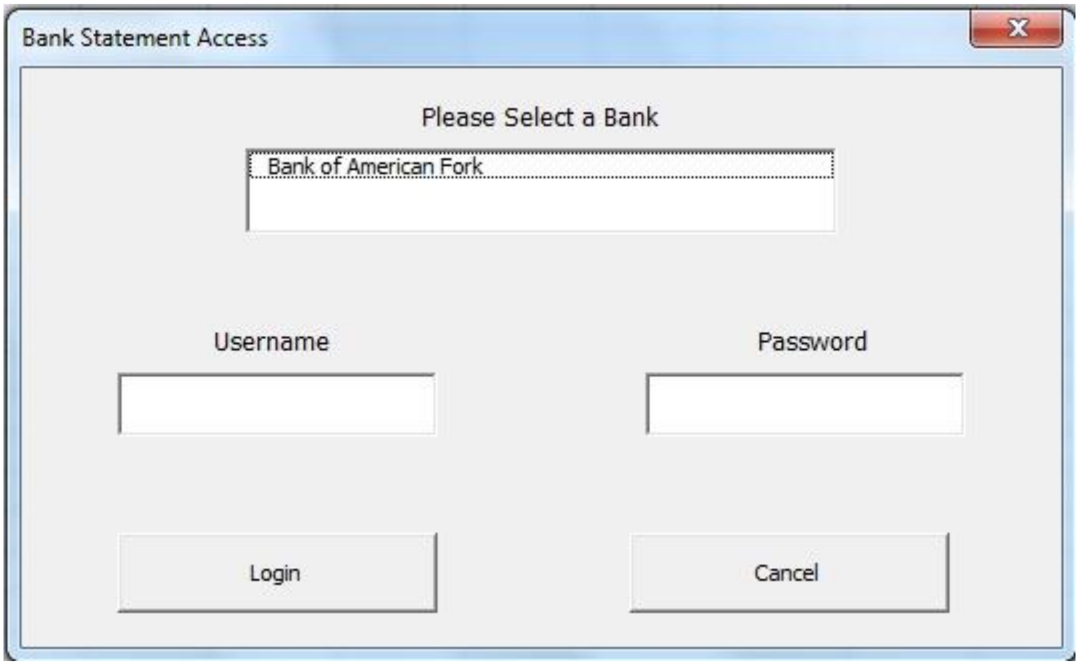


Figure 4

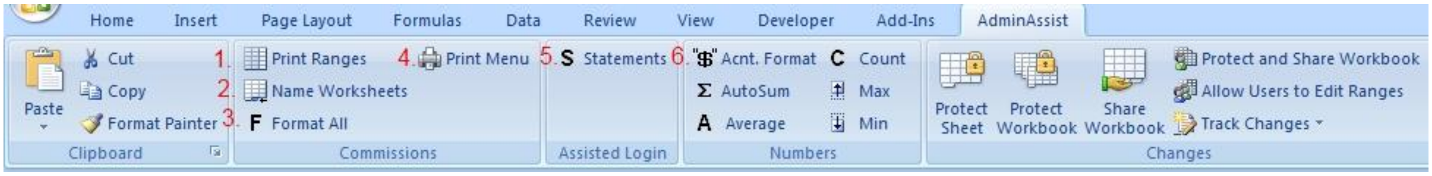
This allows the user to select a bank, and then insert the credentials for login. The code executes when "Login" is clicked and takes you to the statement download page.

After creating this assisted login function, I took it to the accounting firm. Although they thought it was very useful, there were some issues that arose when dealing with their clients' other banks. Some of their clients' passwords had to be accessed through a remote server and those passwords changed very frequently. Likewise, the login process was different each time with pictures, questions, and rotating dials. At times, the user would need to click in the password using their mouse and a screen displayed keyboard.

Another difficulty was that with other banks, the download statements function was not very easy to do using internet explorer. The accountant would need to manually select a location for download and therefore, the automation does not speed things up dramatically.

This was an area of my project that I completed first. Although it saved time, I decided to start focusing my project more on the commissions and the Custom Ribbon which would save substantially more time than assisted login for each bank. Simply put, there were better returns if I spent my time somewhere else instead of repeating this processes for all of their banks.

Custom AdminAssist Ribbon



As you can see in the AdminAssit Ribbon, I have included the Clipboard and Changes group. Along with the Numbers group, these are different functions that get used very frequently by the accounting firm. Below I have discussed some of custom buttons included.

1. Print Ranges - This function defines the print ranges and print titles for monthly commission statements.
2. Name Worksheets – This function will name all the worksheets according to the Last name of the individual who is on that commission statement.
3. Format All – This will accomplish all of the formatting needs for the commission statement. It includes Print Ranges, Name Worksheets, as well as other functionality.
4. Print Menu – This launches the printing process for the monthly commission statements. It brings up the User Form and will initiate printing for you.
5. Statements – This is the assisted login User Form. It will launch the process of automated bank login. I have included the one bank that I have completed before altering my project.
6. Accnt. Format – When clicked, this will run a VBA sub procedure that will change the selected cells to the custom number format that the accounting firm prefers to use.

The custom ribbon took a substantial amount of time put together. However, after saving it as an Add In, it makes it a lot easier for the firm to access the VBA sub procedures and further expedite the processes. The firm really appreciated this functionality.

Complications

As I discussed the implementation steps, I have covered many of the complications that arose. Ultimately, I had some issues understanding the Login function using the Agent1. I came to you for help which got my project moving. After that, many of the issues that I came across I was able to solve using Google. It was not always the easiest, but after some research, I was always able to find a form or post that generated some ideas. I would appreciate if for your classes you would send out a document or excel spreadsheet with a list of your favorite VBA links. This would be useful in helping us be able to access information online more quickly.

Conclusion

The final project was the biggest resource for the class. It was evident that I didn’t remember everything since the first day of class, but the foundation was sufficient to discover the solution on my own. I truly see the value in this course. As for my final project, the accounting firm loves it. They feel like I am an expert programmer for putting this together. Although this project took many hours, I thoroughly enjoyed it.