



**Employee Efficiency Report Generator**

**By Brett Bishopp**

**April 2015**



## **Boostability Employee Efficiency Report Generator**

### **Executive Summary**

Boostability is an internet marketing company. A large part of what they do is create internet content to draw appropriate attention towards their clients' websites. Efficiency Reports are key in understanding both the effectiveness of an employee's time in being able to generate internet traffic for a client. It is the heart of what the business does for the client.

This program will automate the creation of these Efficiency Reports. Because the user base for implementation of these reports will be changing, foresight into possible user errors are key into the making of this program work smoothly.

Due to potential for high employee turnover, this program integrates the option to update employee data while the program is running. User-entered data will be retained, minimalizing the user-interface required in the future.

Overall, these Efficiency Reports are a great tool but take time to create, especially in training the frequently changing user base. Daily, weekly, and monthly reports are created to track progress and trends. The time demanded to generate these reports adds up. This program automates the creation of these reports, both giving back time to the user base as well as ensuring quality data due to decreased risk of user error.

## Implementation Documentation

### *Company and Project Background*

Boostability is an SEO internet marketing company employing over 200 employees. They work with small and medium business to increase the client's rankings on Google Search. They drive traffic towards the client's website on relevant search terms thus driving business to the client website and business.

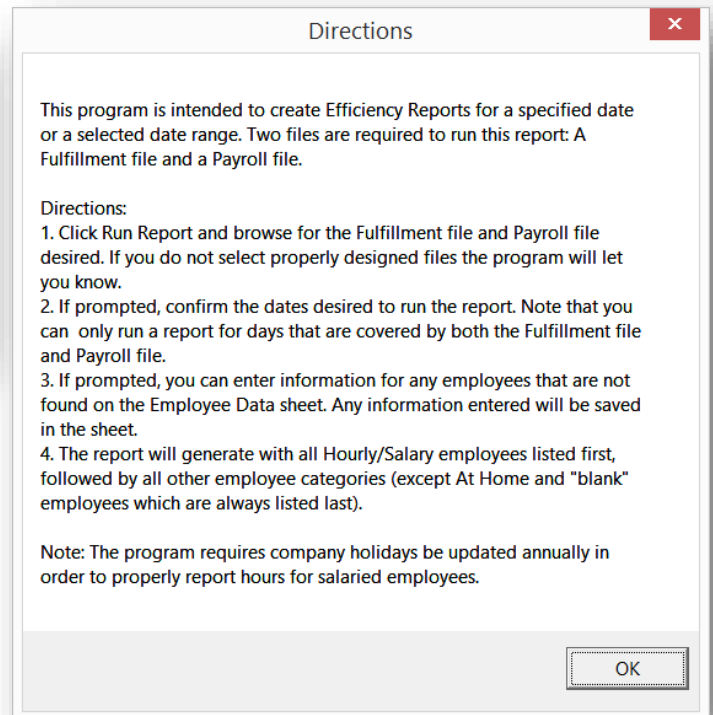
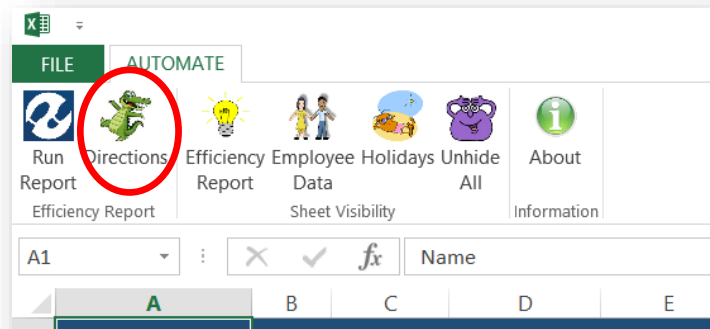
My wife is currently the Director of Fulfillment at Boostability. She is responsible for reporting the work and efficiency of over 100 employees. To enable greater visibility into the productivity of her various teams, she runs numerous reports on various different measures. One of these reports is an Efficiency Report which reports the efficiency of an employee and a team on selected tasks. These tasks include multiple types of Blog Reportings and Onsite Copies. It also includes various types of Review tasks.

Prior to this program, my wife would take time every morning to run a daily version of this report. Every morning for 10-20 minutes she would prepare this report (depending on how many categories and employees were in a given report). Periodically as needed she would also run consolidated reports (similar reports but over various time periods such as a week, two-weeks, and/or a month). One of the biggest challenges for her was incorporating the various company holidays in her reports. This required her to remember when these occurred and manually account for them when she would prepare her reports.

As my wife will be leaving her current position soon, this added to the need to get these reports automated as soon as possible. She felt that being able to create reports in the same manner that they have always been created would highly benefit the new Director (replacing my wife). The new Director would be able to have a parallel reporting structure to compare newly prepared reports against. The 10-20 minute daily report prepared by my wife would take much longer for a newly appointed Director to reproduce.

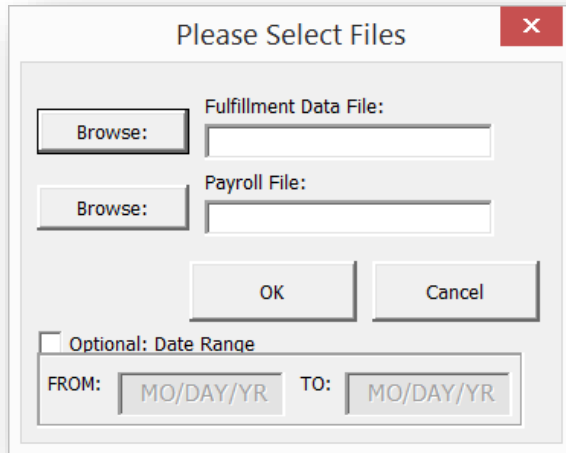
## Running the VBA Program – Details

This program enables Efficiency Reports to be pulled automatically. Let's walk through how this report is pulled. The Directions button (found on the Ribbon) will provide an excellent starting point.



As seen from the Directions, there are multiple steps shown in this program. It is important to note that for the most part (especially after multiple reports are generated), steps 2 and 3 will probably be bypassed, resulting in a quickly generated report. Part of the challenge of creating this program was having to take into account possible user error or lack of information. For the sake of clarity, I will walk through various possible error messages and what impact they will have on the user interface with the program.

After clicking Run Report (from the Ribbon) a pop-up box will display itself:

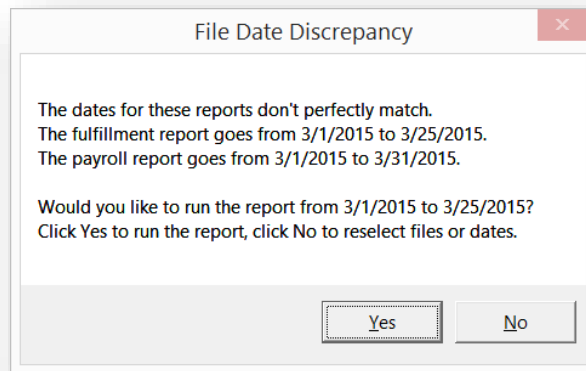


The dialog box titled "Please Select Files" contains the following elements:

- A "Browse:" button next to a text field labeled "Fulfillment Data File:".
- A "Browse:" button next to a text field labeled "Payroll File:".
- "OK" and "Cancel" buttons.
- An unchecked checkbox labeled "Optional: Date Range".
- Below the checkbox, "FROM:" and "TO:" labels with corresponding date input fields showing the format "MO/DAY/YR".

Upon selecting the appropriate Fulfillment and Payroll files (note that an error message will pop up if non-Fulfillment files or non-Payroll files are selected at this point) the user has the option to select a date range to run the report. If no date range is selected and the Fulfillment file dates do not correlate perfectly with the Payroll file dates then error messages like the following could apply:

(If the dates from the two files do not match)



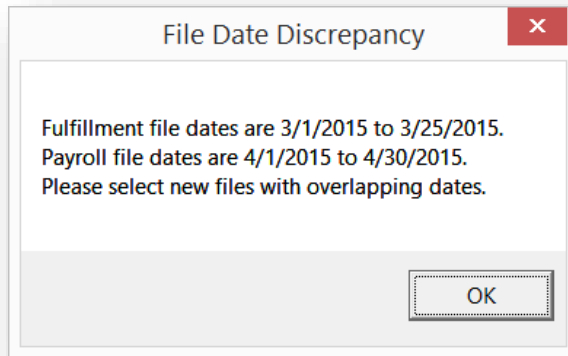
The dialog box titled "File Date Discrepancy" contains the following text:

The dates for these reports don't perfectly match.  
The fulfillment report goes from 3/1/2015 to 3/25/2015.  
The payroll report goes from 3/1/2015 to 3/31/2015.

Would you like to run the report from 3/1/2015 to 3/25/2015?  
Click Yes to run the report, click No to reselect files or dates.

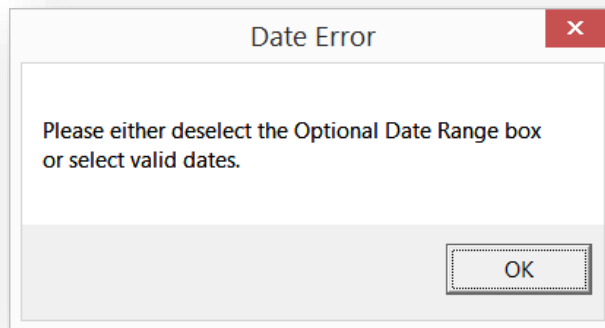
Buttons: Yes, No

(If the dates from the two files do not overlap)

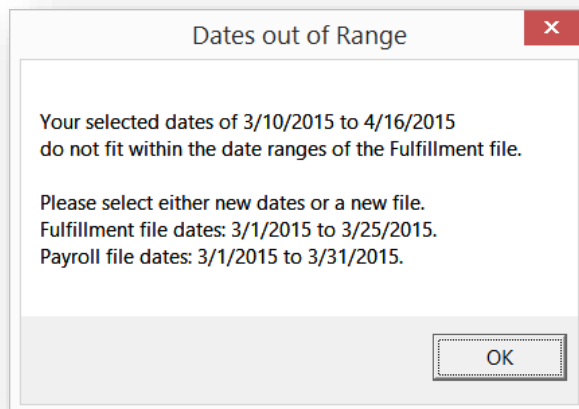


The following error messages could apply if the user opts to manually select dates to run the report:

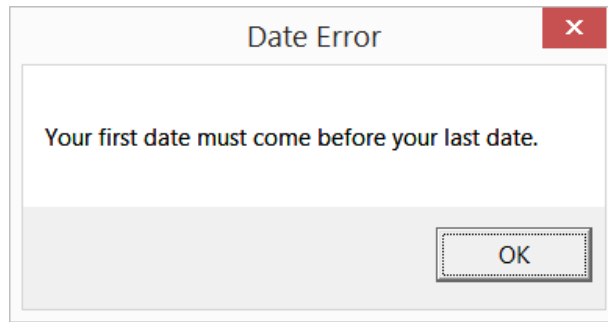
(If the Optional Date Range box is selected and a non-date-formatted text is entered)



(If one or both files do not sufficiently cover the dates that were selected)



(If the first date comes after the second date)



Once the user selects valid dates (or utilizes the system-driven date range) the report will begin to generate. During this time, the previous report in the workbook will be deleted (which will eventually be replaced with the newly generated report). The new Fulfillment file and new Payroll file (which were selected when browsing) will be copied as sheets to the workbook (replacing the old versions if they exist). Note that both .CSV and .XLSX files are allowed to be selected and copied in this step.

The program will then account for all the Blog, Copy, and Review tasks to the right of the Fulfillment data as shown here. Note that this is all running automatically and the user will not see this:

(Columns M:Q are automatically filled by the program)

The screenshot shows an Excel spreadsheet titled "Efficiency Report Generator - Excel". The spreadsheet has a ribbon with "FILE" and "AUTOMATE" tabs. The "AUTOMATE" tab is active, showing options like "Run Report", "Directions", "Efficiency Report", "Employee Data", "Holidays", "Unhide All", and "About". The spreadsheet data is as follows:

1	UserName	StatusDate	time	am/pm	TaskTyp	TaskSta	Task Ac	ActualS	Partner	Custom	Langua	Textbo	Blogging	Copy	Review	Sum Total	Within Date Ran
2	Employee67	3/4/2015 0:00	5:17:02	AM	Onsite Co	COMPLETE						Yes	0	0	0	0	YES
3	Employee67	3/4/2015 0:00	5:17:02	AM	Onsite Co	COMPLETE						Yes	0	0	0	0	YES
4	Employee147	3/4/2015 0:00	5:16:59	AM	Onsite Co	COMPLETE						Yes	0	0	0	0	YES
5	Employee147	3/4/2015 0:00	5:17:00	AM	Onsite Co	COMPLETE						Yes	0	0	0	0	YES
6	Employee14	3/4/2015 0:00	9:15:58	AM	Onsite Blo	COMPLETE						Yes	1	0	0	1	YES
7	Employee14	3/4/2015 0:00	10:29:08	AM	Onsite Blo	COMPLETE						Yes	1	0	0	1	YES

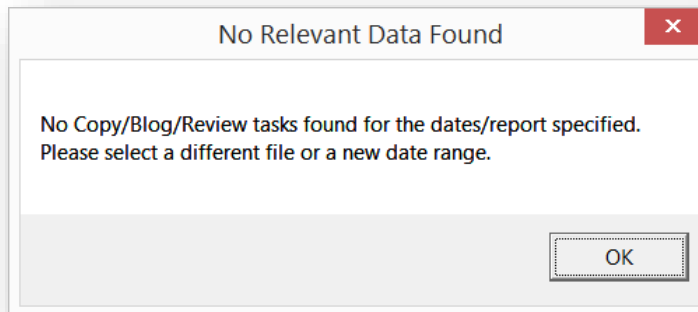
The program will then sum the appropriate payroll hours (for the dates specified) on the Payroll sheet. Note that this is all running automatically and the user will not see this either.

(The last unused column – In this case *Column AW* – has a formula automated to it from VBA to sum the hours inclusively within the selected date range. The selected dates are pasted to the right for convenience.)

	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY
1	3/25/2015	3/26/2015	3/27/2015	3/28/2015	3/29/2015	3/30/2015	3/31/2015	Total Non-	Total OT	Total Hour	Home Dep	Home Cha	Home Mar	Home Des	Home Nev	Home Nev	Applicable	3/1/2015	3/25/2015
2	8	8	8	0	0	7.88	8	159.26	0	159.26	Home Dep	Home				Home Nev	127.38		
3	7.91	8.15	7.95	0	0	8.33	7.71	167.36	0	167.36	Home Dep	Home				Home Nev	135.22		
4	0	5.97	0	0	0	0	5.9	53.98	0	53.98	Home Dep	Home				Home Nev	42.11		
5	5	6.5	3.5	0	0	5	5	105.15	0	105.15	Home Dep	Home				Home Nev	85.15		
6	0	0	0	0	0	0	0	56.86	0	56.86	Home Dep	Home				Home Nev	56.86		
7	4.42	4.1	4.42	0	0	4.68	2.02	92.26	0	92.26	Home Dep	Home				Home Nev	76.22		

At this point in the automation, the following error may appear. This is due to no Blog, Copy, or Review tasks being recorded during the selected dates (this is rare and was implemented to cover all incidences):

(After user is notified they will be redirected back to the Select Files prompt)



After the program makes all appropriate sums and categorical references, a pivot is then automated. This pivot will replace a previous pivot if one exists. If one doesn't exist then a new pivot sheet will be added.

Once a pivot is generated, the program will paste formulas to the right. These formulas will reference the Employee Data sheet. This is how the program knows if an employee is Salary, Hourly, etc., who their manager is, and if they have a specific payroll name.

If a name (for which no matching employee from the Employee Data sheet) has had activity, then "(blank)" will show up next to the pivot and will trigger the below pop up box to show:



(Pop up box informing the user that information is needed for "Employee25")

Notice information is missing for "Employee25"

The screenshot shows an Excel spreadsheet titled 'Efficiency Report Generator - Excel'. The spreadsheet contains a table with columns for employee information and summary statistics. A red arrow points from the text 'Notice information is missing for "Employee25"' to the row for Employee25 in the spreadsheet. A pop-up dialog box titled 'Employee Information Needed (1 of 4)' is overlaid on the spreadsheet, showing the form for Employee25. The form includes fields for Employee Type (Hourly, Salary, At Home, Other), Manager (a dropdown menu), and Optional: Payroll Name (I will enter time manually, or a text input field). The form also has buttons for 'Save & Next', 'Skip One', and 'Skip Remaining'.

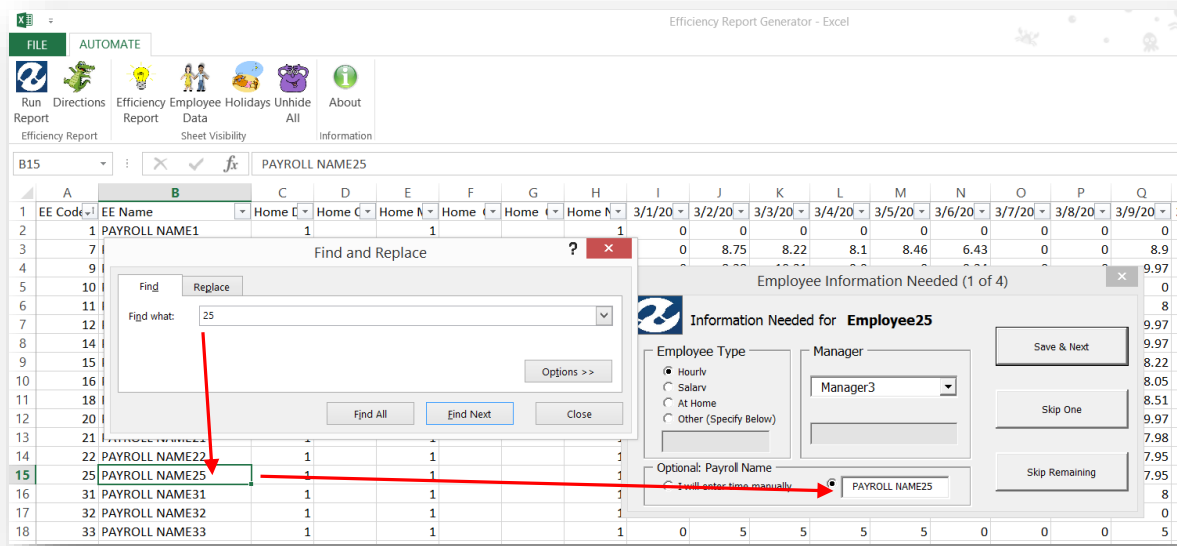
Row Labels	Sum of Copy	Sum of Blogging	Sum of Review	employee	manager	payroll nar	total hour	abrev e type	
Employee1	0	19	73	Salary	Manager1	A	Employee	132.5	SalHour
Employee10	26	26	28	Salary	Manager1	J	Employee	128	SalHour
Employee11	27	14	0	Hourly	Manager2	K	Employee	135.22	SalHour
Employee12	0	18	74	Hourly	Manager2	L	Employee	147.23	SalHour
Employee13	4	1	0	Salary	Manager1	M	Employee	136	SalHour
Employee14	27	52	196	Hourly	Manager1	N	Employee	147.23	SalHour
Employee15	0	2	0	Hourly	Manager3	O	Employee	143.23	SalHour
Employee16	20	13	0	Hourly	Manager2	P	Employee	139.59	SalHour
Employee17	11	1	0	At Home	0	0	0	0	At Home
Employee18	29	18	0	Hourly	Manager3	R	Employee	139.81	SalHour
Employee19	4	0	0	At Home	0	0	0	0	At Home
Employee2	7	3	0	Salary	Manager1	B	Employee	136	SalHour
Employee20	1	26	116	Hourly	Manager2	T	Employee	147.23	SalHour
Employee21	24	32	0	Hourly	Manager2	U	Employee	143.52	SalHour
Employee22	0	30	47	Hourly	Manager2	V	Employee	143.39	SalHour
Employee23	2	3	0	At Home	0	0	0	0	At Home
Employee24	47	23	55	Salary	Manager3	X	Employee	136	SalHour
Employee25	19	21	28	(blank)	(blank)	(blank)	0	(blank)	(blank)
Employee26	5	6	0	Language	0	0	0	0	Language
Employee27	2	0	0	(blank)	(blank)	(blank)	0	(blank)	(blank)

(The user has the option to select the Employee Type, the Manager, and a Payroll Name. If the user specifies a new manager, the new manager's name will show up on the manager drop down menu for the next request.)

The close-up shows the 'Employee Information Needed (1 of 4)' dialog box. The 'Manager' dropdown menu is open, showing a list of managers: Manager3, Manager1, Manager3, Manager2, and Other (Specify Below). The 'Employee Type' section has radio buttons for Hourly, Salary, At Home, and Other (Specify Below). The 'Optional: Payroll Name' section has radio buttons for 'I will enter time manually' and a text input field. The 'Save & Next' button is visible.

For ease of use, the user is allowed to click off this form and reference the Payroll file (or any Payroll file) if the wish to copy and paste the exact Payroll Name for the employee. This was done at request of my wife.

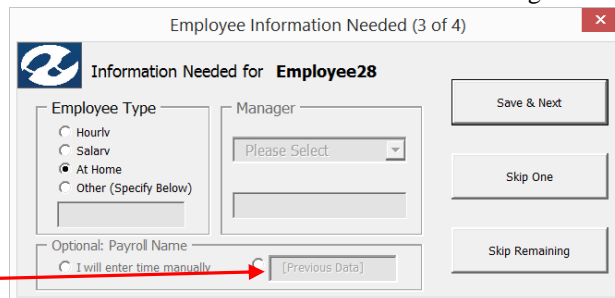
(User is able to click off form and search data if they wish)



Due to the nature of the business, *At Home* employees and *Other* employees do not report to a manager (these employees are actually contract piece-rate workers and are not employed directly by the company). When these options are selected the Manager and Payroll options are no longer selectable. If previously entered data (previous to the selection of *At Home* or *Other* option) then this will be grayed out as well and will not upload to the Employee Data sheet upon clicking save.

(In the case of selecting *At Home*, the [Previously Data] is retained incase the user changes Employee Type to *Hourly* or *Salary*. If *At Home* remains selected when the user clicks Save then whatever was entered for the Manager Name will not upload.)

Data retained (but grayed out) if *At Home* or *Other* selected as Employee Type



If the user does not want to enter this information, they can skip this request. They also have the option to skip all remaining requests. Note on the final request for information, the “Save & Next” button changes to “Save & Continue” as well as the “Skip One” button becomes inactive.

Employee Information Needed (4 of 4)

Information Needed for **Employee30**

Employee Type

- Hourly
- Salary
- At Home
- Other (Specify Below)

Manager

Please Select

Optional: Payroll Name

- I will enter time manually

Buttons: Save & Continue, Skip One, Skip Remaining (circled in red)

(Once the User clicks Save, the data is automatically added to the Employee Data report so the user won't have to repeat this for the next time they pull a report with this employee.)

	A	B	C	D
9	Employee8	At Home		
10	Employee9	Hourly	Manager2	PAYROLL NAME9
11	Employee10	Salary	Manager1	PAYROLL NAME10
12	Employee11	Hourly	Manager2	PAYROLL NAME11
13	Employee12	Hourly	Manager2	PAYROLL NAME12
14	Employee13	Salary	Manager1	PAYROLL NAME13
15	Employee14	Hourly	Manager1	PAYROLL NAME14
16	Employee15	Hourly	Manager3	PAYROLL NAME15
17	Employee16	Hourly	Manager2	PAYROLL NAME16
18	Employee17	At Home		
19	Employee18	Hourly	Manager3	PAYROLL NAME18
20	Employee19	At Home		
21	Employee20	Hourly	Manager2	PAYROLL NAME20
22	Employee21	Hourly	Manager2	PAYROLL NAME21
23	Employee22	Hourly	Manager2	PAYROLL NAME22
24	Employee23	At Home		
25	Employee24	Salary	Manager3	PAYROLL NAME24
26	Employee26	Language		
27	Employee29	At Home		
28	Employee25	Hourly	Manager3	PAYROLL NAME25
29	Employee27	Salary	Manager4	PAYROLL NAME27
30	Employee28	At Home		
31				
32				
33				
34				
35				
36				

After the last employee information is gathered the data to the right of the pivot will regenerate itself based on the updated information in the Employee Data sheet and a final report will be generated off of this data.

Notice this data is automatically repulled if updates are made to the Employee Data during the program.

Row Labels	Sum of Copy	Sum of Blogging	Sum of Review	employee	manager	payroll nar	total hour	abbrev type
Employee1	0	19	73	Salary	Manager1	PAYROLL	132.5	SalHour
Employee10	26	26	28	Salary	Manager1	PAYROLL	128	SalHour
Employee11	27	14	0	Hourly	Manager2	PAYROLL	135.22	SalHour
Employee12	0	18	74	Hourly	Manager2	PAYROLL	147.23	SalHour
Employee13	4	1	0	Salary	Manager1	PAYROLL	136	SalHour
Employee14	27	52	196	Hourly	Manager1	PAYROLL	147.23	SalHour
Employee15	0	2	0	Hourly	Manager3	PAYROLL	143.23	SalHour
Employee16	29	13	0	Hourly	Manager2	PAYROLL	139.59	SalHour
Employee17	11	1	0	At Home			0	At Home
Employee18	20	18	0	Hourly	Manager3	PAYROLL	139.81	SalHour
Employee19	4	0	0	At Home			0	At Home
Employee2	7	3	0	Salary	Manager1	PAYROLL	136	SalHour
Employee20	1	36	116	Hourly	Manager2	PAYROLL	147.23	SalHour
Employee21	24	32	0	Hourly	Manager2	PAYROLL	143.52	SalHour
Employee22	0	30	47	Hourly	Manager2	PAYROLL	143.39	SalHour
Employee23	2	3	0	At Home			0	At Home
Employee24	47	23	55	Salary	Manager3	PAYROLL	136	SalHour
Employee25	19	21	28	Hourly	Manager3	PAYROLL	132.9	SalHour
Employee26	5	6	0	Language			0	Language
Employee27	2	0	0	Salary	Manager4	PAYROLL	136	SalHour
Employee28	9	4	0	At Home			0	At Home
Employee29	2	0	0	At Home			0	At Home
Employee3	2	0	0	At Home			0	At Home

The order of presentation of employee categories was very clearly specified (as this is how they were created historically). All Hourly/Salary employees are grouped together and are presented first. If they have a manager listed they are presented first (in groups with their manager). If no manager is listed then they are listed after the employees with a manager.

Next comes the Language employees (and any category that will be added in the future). Finally the At Home writers come next as well as any uncategorized employees (noted with “(blank)” category). None of the Language, At Home, or uncategorized employees are tracked by hours worked so that data is not included in this report.

Finally, if there was a holiday during the reported time period, then it is noted at the bottom of the report. This is important as it negates hours for salaried employees on the holidays. Salaried employees are also negated for any weekends as well as any time off they take (which information comes through in the Payroll file).

(An example of a final report. Notice columns H:O are hidden. There are minor calculations occurring here. This was done so the format would stay consistent with the manual reports pulled previously.)

Notice "Employee25" is part of the final report.

Because we opted to not include any data on "Employee30" they are in a separate (blank) category.

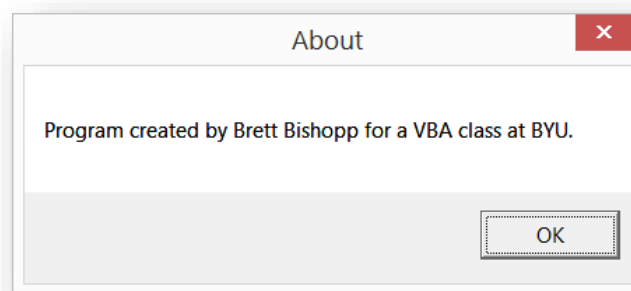
Notice the holiday that this period accounts for (March 17<sup>th</sup>)

	A	B	C	D	E	F	G	P	
	Name	Team	Copy Tasks Completed	Blogging Tasks Completed	Review Tasks Completed	Total Tasks Completed	Hours Worked	Productive Minutes	
1									
2	Employee1	Manager1	0	19	73	92	132.5	56.23%	
3	Employee10	Manager1	26	26	28	80	128	71.88%	
4	Employee13	Manager1	4	1	0	5	136	4.41%	
5	Employee14	Manager1	27	52	196	275	147.23	155.54%	
6	Employee2	Manager1	7	3	0	10	136	9.56%	
7	<b>Team Total</b>		<b>64</b>	<b>101</b>	<b>297</b>	<b>462</b>	<b>679.73</b>	<b>60.98%</b>	
8									
9	Employee11	Manager2	27	14	0	41	135.22	40.67%	
10	Employee12	Manager2	0	18	74	92	147.23	49.58%	
11	Employee16	Manager2	29	13	0	42	139.59	39.40%	
12	Employee20	Manager2	1	36	116	153	147.23	88.98%	
13	Employee21	Manager2	24	32	0	56	143.52	61.32%	
14	Employee22	Manager2	0	30	47	77	143.39	58.23%	
15	Employee9	Manager2	0	33	46	79	147.23	60.45%	
16	<b>Team Total</b>		<b>81</b>	<b>176</b>	<b>283</b>	<b>540</b>	<b>1003.41</b>	<b>57.25%</b>	
17									
18	Employee15	Manager3	0	2	0	2	143.23	2.79%	
19	Employee18	Manager3	29	18	0	47	139.81	46.49%	
20	Employee24	Manager3	47	23	55	125	136	88.60%	
21	Employee25	Manager3	19	21	28	68	132.9	56.43%	
22	Employee7	Manager3	0	23	57	80	144.83	51.44%	
23	<b>Team Total</b>		<b>95</b>	<b>87</b>	<b>140</b>	<b>322</b>	<b>696.77</b>	<b>48.65%</b>	
24									
25	Employee27	Manager4	2	0	0	2	136	1.47%	
26	<b>Team Total</b>		<b>2</b>	<b>0</b>	<b>0</b>	<b>2</b>	<b>136</b>	<b>1.47%</b>	
27									
28	<b>Language</b>								
29	Employee26		5	6	0	11	0	-	
30	Employee4		0	0	27	27	0	-	
31	<b>Language Total</b>		<b>5</b>	<b>6</b>	<b>27</b>	<b>38</b>	<b>0</b>	<b>-</b>	
32									
33	<b>At Home</b>								
34	Employee17		11	1	0	12	0	-	
35	Employee19		4	0	0	4	0	-	
36	Employee23		2	3	0	5	0	-	
37	Employee28		9	4	0	13	0	-	
38	Employee29		2	0	0	2	0	-	
39	Employee3		2	0	0	2	0	-	
40	Employee5		4	5	0	9	0	-	
41	Employee6		1	12	58	71	0	-	
42	Employee8		8	9	0	17	0	-	
43	<b>At Home Total</b>		<b>43</b>	<b>34</b>	<b>58</b>	<b>135</b>	<b>0</b>	<b>-</b>	
44									
45	<b>(blank)</b>								
46	Employee30		2	24	0	26	0	-	
47	<b>(blank) Total</b>		<b>2</b>	<b>24</b>	<b>0</b>	<b>26</b>	<b>0</b>	<b>-</b>	
48									
49	<b>Total Internal</b>		<b>247</b>	<b>370</b>	<b>747</b>	<b>1364</b>	<b>2515.91</b>	<b>54.08%</b>	
50	<b>Total</b>		<b>292</b>	<b>428</b>	<b>805</b>	<b>1525</b>	<b>2515.91</b>	<b>61.63%</b>	
51	Note: This report is for Mar 01 - Mar 25 and accounts for 1 holiday(s) during this period.								

Finally there are some other important items to note. Of the two files selected to generate this report (the Fulfillment file and Payroll file), the program will not leave open a file if it was not already open prior to the program executing. If a file was open prior to the program executing, the program will leave this file open upon completion of the program.

Other items to note on the ribbon include Sheet Visibility which will unhide a specified sheet and activate it. There is also a button that will unhide all sheets at once. Upon creation of a new Efficiency Report, all sheets will be hidden except for the Employee Data, Holiday Schedule, and the regenerated report.

There is also an About button which, when clicked, shows a brief note about the author of this program.



## Learning and Discussion of Difficulties Encountered

One of the biggest challenges in creating this report was considering the vast opportunities for user error. As stated earlier, this report will be utilized by various users with presumably no VBA experience. Taking into account the various possible errors was a major challenge in building this program.

Another large challenge was forcing a listed format for Employee Type in the final report (*Hourly/Salary* first – grouped by manager, followed by the rest of the categories except for *At Home* and *(blank)* which were always listed at the end). Taking into account the possibility that one of these groups may not exist when this report was formed was a challenge as well. I used multiple arrays to complete the execution of this.

Implementing blank rows and row sums in the final report was challenging as well. Working with a versatile amount of employees, managers, and employee categories was quite challenging. This was completed through assigning different variables *rowCount* and *bigRowCount* when writing to various rows.

```
For y = 1 To eTypeAquantity
  If eTypeA(y) <> "SalHour" Then
    ws.Cells(bigRowCount + 1, 1).Value = eTypeA(y)
    ws.Cells(bigRowCount + 1, 1).Font.Bold = True
    bigRowCount = bigRowCount + 1
  End If
  For z = 1 To managerQuantity2
    For x = 1 To sht.Cells(sht.Rows.Count, "A").End(xlUp).row - 6
      If finalArray(x, 9) = eTypeA(y) Then
        If finalArray(x, 6) = managerArray(z) Then
          ws.Cells(rowCount + bigRowCount, 1).Value = finalArray(x, 1)
          If managerArray(z) = "0" Then
            ws.Cells(rowCount + bigRowCount, 2).Value = ""
          ElseIf managerArray(z) = "(blank)" Then
            ws.Cells(rowCount + bigRowCount, 2).Value = ""
          Else: ws.Cells(rowCount + bigRowCount, 2).Value = managerArray(z)
          End If
          ws.Cells(rowCount + bigRowCount, 3).Value = finalArray(x, 2)
          ws.Cells(rowCount + bigRowCount, 4).Value = finalArray(x, 3)
          ws.Cells(rowCount + bigRowCount, 5).Value = finalArray(x, 4)
          ws.Cells(rowCount + bigRowCount, 7).Value = finalArray(x, 8)
          ws.Cells(rowCount + bigRowCount, 6).FormulaR1C1 = "=SUM(RC[-3]:RC[-1])"
          ws.Cells(rowCount + bigRowCount, 8).FormulaR1C1 = "=RC[-5]*R1C9"
          ws.Cells(rowCount + bigRowCount, 10).FormulaR1C1 = "=RC[-6]*R1C11"
          ws.Cells(rowCount + bigRowCount, 12).FormulaR1C1 = "=RC[-7]*R1C13"
          ws.Cells(rowCount + bigRowCount, 14).FormulaR1C1 = "=SUM(RC[-6]:RC[-1])"
          ws.Cells(rowCount + bigRowCount, 15).FormulaR1C1 = "=RC[-8]*60"

          rowCount = rowCount + 1
          dataPresent = True
        End If
      End If
    Next
  End If
  If dataPresent = True Then
    ws.Cells(rowCount + bigRowCount, 6).FormulaR1C1 = "=SUM(RC[-3]:RC[-1])"
    ws.Cells(rowCount + bigRowCount, 8).FormulaR1C1 = "=RC[-5]*R1C9"
    ws.Cells(rowCount + bigRowCount, 10).FormulaR1C1 = "=RC[-6]*R1C11"
    ws.Cells(rowCount + bigRowCount, 12).FormulaR1C1 = "=RC[-7]*R1C13"
```

Another challenge in creating this program was forming a list of unique Manager name for the below combo box:

The screenshot shows a dialog box titled "Employee Information Needed (1 of 4)" with a close button (X) in the top right corner. The main heading is "Information Needed for Employee25".

On the left, under "Employee Type", there are four radio buttons: "Hourly" (selected), "Salary", "At Home", and "Other (Specify Below)". Below this is an empty text input field.

On the right, under "Manager", there is a dropdown menu. The dropdown is open, showing a list of unique manager names: "Manager3", "Manager1", "Manager3", "Manager2", and "Other (Specify Below)". The "Manager3" entry is currently selected and highlighted in blue.

At the bottom left, under "Optional: Payroll Name", there are two radio buttons: "I will enter time manually" (selected) and an empty radio button. To the right of the second radio button is an empty text input field.

On the right side of the dialog, there are three buttons: "Save & Next", "Skip One", and "Skip Remaining".



I did not want to force the user to create and maintain a separate list of managers. Working off of the current Employee Detail list would ensure an up-to-date manager list. The list of managers, however, was full of duplications as they are attributes to the list unique identifier (employee name). Once I had an array of all the manager names (including duplicated names), I used the following line of code to trim down the array to include only unique names:

```
'this sub makes sure that each name in the array is unique
Sub doubleCheckManager1()
Dim name As String
Dim x As Integer
Dim y As Integer
Dim n As Integer

For x = 1 To managerQuantity
If x > managerQuantity Then
Exit For
Else
name = managerList(x)
For y = x + 1 To managerQuantity
If y > managerQuantity Then
Exit For
Else
If name = managerList(y) Then
n = y
Do
If y = managerQuantity Then
n = n + 1
Else
managerList(n) = managerList(n + 1)
n = n + 1
End If
Loop Until n = managerQuantity Or n > managerQuantity
ReDim Preserve managerList(1 To managerQuantity - 1)
managerQuantity = managerQuantity - 1
y = y - 1
End If
End If
Next
End If
Next

End Sub
```

Another challenge was implementing the date parameters in the reports. Working with three various date ranges (the Fulfillment file dates, the Payroll file dates, and possibly manually-entered dates) I had to be able to read the dates available from each file and compare them against each other.

The following lines are a small section of code that is used to evaluate which dates will be assigned to the global variables *CombinedMin* and *CombinedMax*. These global variables will be used later as the date parameters to pull the report against.

```
Set rngP = payrollWB.Sheets(1).Range(Cells(1, 9), Cells(1, col - 1))
fulfillmentWB.Activate
Set rngF = fulfillmentWB.Sheets(1).Range(Cells(2, 2), Cells(rowF, 2))
mainWB.Activate
minimumDateF = Application.WorksheetFunction.Min(rngF)
maximumDateF = Application.WorksheetFunction.Max(rngF)
minimumDateP = Application.WorksheetFunction.Min(rngP)
maximumDateP = Application.WorksheetFunction.Max(rngP)

If minimumDateP >= minimumDateF Then
    If minimumDateP <= maximumDateF Then
        payrollMin = True
    End If
End If

If minimumDateF >= minimumDateP Then
    If minimumDateF <= maximumDateP Then
        fulfillMin = True
    End If
End If

If maximumDateF >= minimumDateP Then
    If maximumDateF <= maximumDateP Then
        fulfillMax = True
    End If
End If

If maximumDateP >= minimumDateF Then
    If maximumDateP <= maximumDateF Then
        payrollMax = True
    End If
End If

If payrollMax = True Then
    combinedMax = maximumDateP
ElseIf fulfillMax = True Then
    combinedMax = maximumDateF
Else
```

Another challenge was working with the Fulfillment file and the Payroll file as these files were organized quite differently. The first file was made from a set amount of columns and varying number of rows (depending on the number of tasks). The Payroll file varied on both the column and row dimension (depending on days and employees). I was able to capture the appropriate data through utilizing `.End(xlToLeft)` and `.End(xlUp)`.

A final challenge encountered was accounting for the various holidays and weekends in this report. As stated earlier, salaried employees need to have holiday and weekend hours negated from their total hours worked. Because company holidays will vary from year to year, I created a separate tab for the user to enter company holidays. This tab will only need to be updated once a year.

To account of the weekends and holidays, I utilized the following lines of code:

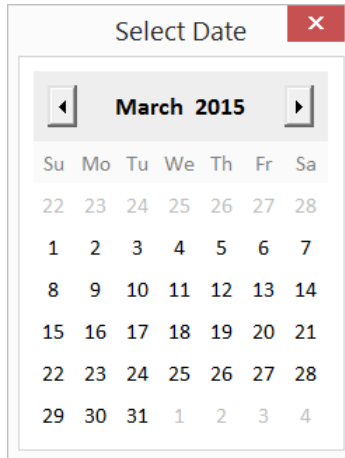
```
Sub holidayCounter()  
    Dim x As Integer  
    Dim y As Byte  
  
    x = 0  
    holidayCount = 0  
    If combinedMin <> combinedMax Then  
        For x = 1 To combinedMax - combinedMin + 1  
            For y = 1 To mainWB.Sheets("Holiday Schedule").Cells(Sheets("Holiday Schedule").Rows.Count, "A").End(xlUp).row  
                If MonthName(Month(combinedMin + x - 1)) = mainWB.Worksheets("Holiday Schedule").Cells(y, 1) Then  
                    If Format(combinedMin + x - 1, "DD") + 0 = mainWB.Worksheets("Holiday Schedule").Cells(y, 2) Then  
                        holidayCount = holidayCount + 1  
                    End If  
                End If  
            Next  
        Next  
    End If  
End Sub  
  
Sub calcDaysWorked()  
  
    Dim x As Integer  
  
    daysWorked = combinedMax - combinedMin + 1  
  
    Do Until combinedMin + x = combinedMax  
        If (WeekdayName(Weekday(combinedMin + x)) = "Saturday") Or (WeekdayName(Weekday(combinedMin + x)) = "Sunday") Then  
            daysWorked = daysWorked - 1  
        End If  
        x = x + 1  
    Loop  
End Sub
```

## Assistance Needed and Used

For the most part this project was done completely by me (with exceptions noted below). There were two instances when I went to speak with the professor. Those questions were mostly clarifying questions about the scope of the project and avoiding use of the “.copy” method. I learned that copying and pasting worksheets are considered okay in the programming community and won’t affect the user’s clipboard.

Two exceptions of where I did seek outside assistance was in the CalendarForm as well as the function to check to see if a file was already open (function IsFileOpen). I utilized the CalendarForm because I felt that it would improve the user-friendly ability of this program. I utilized the IsFileOpen function so the user would not be subject to an open file getting closed because they decided to run my program.

(The CalendarForm used to select dates)



## Conclusion

In conclusion my wife is excited to be able to utilize this program. This versatility of this program allows various reports to be created. Not only will this program save her time but she is eager to share it with her replacement. The addition of this VBA program to the current tools she utilizes in her work setting will enable a smooth transition between the department directors when she changes positions.