

Final Project  
David Parry  
December 6, 2012

## **Executive Summary**

I work as an Accounts Payable clerk in the BYU Bookstore. There are many vendors with which the bookstore is in a credit balance. The AP department keeps track of all the monthly credit balances throughout the year. At the beginning of each month, one of the tasks is to download the end of month data from the bookstore's accounting software. This data is then inputted into a summary file that shows the vendors and the credit balances throughout the year. This can be a challenge because first, the vendors with a zero balance during the month do not have a balance in the downloaded data, yet they exist in the year summary file. Second, some credit balances are new; vendors which previously did not have a credit balance now do. However, they do not exist in the year summary file. It is a tedious process to find which vendors exist and add rows of data to the file in the correct location. Previously, it would take my supervisor many valuable hours to input this data. My project saves her this time and helps her to avoid this menial task.

My project performs the following key steps:

- Creates a connection to the downloaded text file and imports the data to a new sheet in the file.
- Formats the data for improved readability and enhanced transferability to the summary sheet.
- Transfers the monthly data to the correct month location in the summary file based on the modified date of the downloaded text file.
- Adds rows of data for the new vendors as needed.
- Moves and saves the month data to the same directory as the year summary file.

At the end of the year, a new file will need to be created for 2013. When the end of month file is downloaded next year and the program is run, it will generate a new file, clear out the old balances, create a new folder for the next year, and save the file in that new year folder. Thus, this program will continue for as long as the bookstore stands.

## **Implementation Documentation**

Note1: It is very fortunate that when the accounting software downloads the end of month data, it automatically saves the data to the same location on the computer every time—C:\dell\excel.txt. This way, the program can always read the same file path. However, for the purposes of this demonstration, I have saved the file in another location—C:\Users\David\Documents\.....Fall 2012\VBA EXCEL\Final Project\Copy of credit balance.txt.

### **\*\*\*Step 1: Determine what month we are planning to input.**

The first step is to determine what date the text file was downloaded. This will help us determine what month we are inputting. The AP department has a strict policy of not downloading end of month data until at least the first day of the next month. Thus, the date of the download will always correspond to the previous month's data. For example, if the file was downloaded anytime in May, the month of the data corresponds to April.

```
al Basic - AP Credit Balances.xlsm - [Module1 (Code)]
View Insert Format Debug Run Tools Add-Ins Window Help Type a question for help
Ln 38, Col 1
(General) InputData

Dim LastModified As Date
LastModified = FileDateTime("C:\Users\David\Documents\.....Fall 2012\VBA EXCEL\Final Project\Copy of credit balance.txt")

Dim WhatMonth(1 To 12) As String
Dim MonthName As Variant
Dim MonthNum As Integer
Dim YearNum As Integer

WhatMonth(1) = "January"
WhatMonth(2) = "February"
WhatMonth(3) = "March"
WhatMonth(4) = "April"
WhatMonth(5) = "May"
WhatMonth(6) = "June"
WhatMonth(7) = "July"
WhatMonth(8) = "August"
WhatMonth(9) = "September"
WhatMonth(10) = "October"
WhatMonth(11) = "November"
WhatMonth(12) = "December"

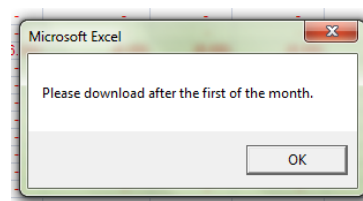
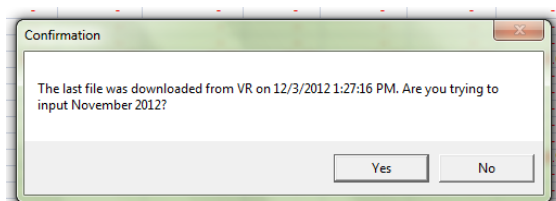
If Month(LastModified) = 1 Then
    MonthNum = 12
    YearNum = Year(LastModified) - 1
Else
    MonthNum = Month(LastModified) - 1
    YearNum = Year(LastModified)
End If

MonthName = WhatMonth(MonthNum)
```

In the preceding screenshot, the variable LastModified returns the date and time of the last modification of the text file. Using the Month(LastModified) function, we can determine the number of the month. This is stored as the variable MonthNum. To determine the name of the corresponding month number, I created an array from 1 to 12 called WhatMonth. One challenge is what happens if the file was modified in January. One minus the month number of January (1 - 1) is equal to zero. The If statement in the code addresses this challenge. If the file was downloaded in January, then the month number is automatically assigned as 12 (December) and the year number (defined as the variable YearNum) is the previous year from the download year. Otherwise, the month is simply the previous month and the year is simply the same year.

### \*\*\*Step 2: Obtain user's confirmation.

We don't want the user to accidentally run the program without previously downloading the file from the accounting software (which automatically saves the data as a text file as explained in Note1). To address this, I created a series of message boxes to appear. The first box (on the left) gives the date of the text file download and asks for confirmation that the user is attempting.



Upon selecting "Yes", the program will continue to run. If "No" is selected, then another message box (on the right) will appear, indicating to the user to wait until after the first of the month to run the program.

These message boxes were created using the following code:

```
result = MsgBox("The last file was downloaded from VR on " & LastModified & ". Are you trying to input " & MonthName & " " & YearNum & "?", vbYesNo, "Confirmation")
```

```
If result = vbNo Then  
MsgBox "Please download after the first of the month."  
Exit Sub  
End If
```

I also created a message box that will appear and cause the sub procedure to close if there already exists a saved file for that specific month. The code for this message box is:

```
Dim filename As String  
filename = ThisWorkbook.Path  
If Dir(filename & "\" & MonthName & YearNum & ".xlsm") <> vbNullString Then  
MsgBox "Error. This month has already been inputted. Please wait until next month."  
Exit Sub  
End If
```

Note2: If you are following the code that appears in Exhibit A, you will notice that the next part of the code executes according to the year (If it is a new year, a new file will be created). I will address that challenge forthcoming.

### **\*\*\*Step 3: Make a connection to text file.**

The next step is to add a new sheet and create a connection in order to import the text file into the sheet. This is done by the following code:

```
Sheets.Add
```

```
With ActiveSheet.QueryTables.Add(Connection:= _  
    "TEXT;C:\Users\David\Documents\.....Fall 2012\VBA EXCEL\Final Project\Copy of credit balance.txt", _  
    Destination:=Range("$A$1"))  
    .TextFileColumnDataTypes = Array(2, 2, 2, 1)  
    .Refresh BackgroundQuery:=False  
End With
```

```
ActiveWorkbook.Connections("Copy of credit balance").Delete
```

```
ActiveSheet.Name = "CB"
```

The code also deletes the connection after importing the data to ensure that the data does not get “refreshed” or changed. Then the code changes the name of the sheet to “CB”. The following screenshot shows the data in the text file and then the data after it has been imported into a spreadsheet in the workbook.

Copy of credit balance.txt - Notepad

File Edit Format View Help			
No heading specified			
AP.VEND	VENDOR	Name	BALANCE.DUE
	NUMBER		
3288	3288	FOLLETT HIGHER EDUCATION	-73554.95
5473	5473	GROUP	
		MBS TEXTBOOK EXCHANGE, "	-69647.13
		INC.	
5085	5085	MCGRAW - HILL COMPANIES	-53400.22
5410	5410	MPS	-36788.05
0704	0704	APPLE	-23061.00
4728	4728	KENDALL HUNT PUBLISHING	-16034.67
		COMPANY	
8824	8824	WEST GROUP PAYMENT CENTER	-13461.30
7535	7535	SOUNDS OF ZION	-4739.12
0705	0705	"APPLE COMPUTER, INC SVC "	-4696.80
7090	7090	SAGE PUBLICATIONS INC	-4344.86
3974	3974	HAY GROUP	-3500.00
7012	7012	ROWMAN & LITTLEFIELD	-3221.66
		PUBLISHING GROUP	
0698	0698	APHA PUBLICATIONS	-2082.50
1040	1040	BOOKBYTE DIRECT	-1957.08
7600	7600	SPRINGER	-1852.73
7473	7473	SINAUER ASSOCIATES INC	-1612.14
7608	7608	STACKPOLE BOOKS	-1513.28
7093	7093	SAGAMORE PUBLISHING, LLC "	-1510.96
6348	6348	PERSEUS DISTRIBUTION INC.	-1501.69
0898	0898	ATHENA SCIENTIFIC	-1335.42
8753	8753	"WAVELAND PRESS, INC. "	-1167.56
1708	1708	CALIF. PRINCETON	-1001.10
		FULFILLMENT SERVICES	
3844	3844	HACKETT PUBLISHING	-553.76
		COMPANY	
0904	0904	ATWOOD PUBLISHING, LLC "	-538.92
7592	7592	SPRINGER PUBLISHING	-528.00
		COMPANY	
5374	5374	MERIDIAN MAGAZINE	-526.68
5820	5820	NAXOS OF AMERICA INC.	-484.74



AP Credit Balances.xlsm - Microsoft Excel

Home Insert Page Layout Formulas Data Review View Developer Project							
C5		GROUP					
A	B	C	D	E	F	G	H
1	No heading specified						
2	AP.VEND	VENDOR	Name	BALANCE.DUE			
3		NUMBER					
4	3288	3288	FOLLETT HIGHER EDUCATION	-73554.95			
5			GROUP				
6	5473	5473	MBS TEXTBOOK EXCHANGE,	-69647.13			
7			INC.				
8	5085	5085	MCGRAW - HILL COMPANIES	-53400.22			
9	5410	5410	MPS	-36788.05			
10	0704	0704	APPLE	-23061.00			
11	4728	4728	KENDALL HUNT PUBLISHING	-16034.67			
12			COMPANY				
13	8824	8824	WEST GROUP PAYMENT CENTER	-13461.30			
14	7535	7535	SOUNDS OF ZION	-4739.12			
15	0705	0705	APPLE COMPUTER, INC SVC	-4696.80			
16	7090	7090	SAGE PUBLICATIONS INC	-4344.86			
17	3974	3974	HAY GROUP	-3500.00			
18	7012	7012	ROWMAN & LITTLEFIELD	-3221.66			
19			PUBLISHING GROUP				
20	0698	0698	APHA PUBLICATIONS	-2082.50			
21	1040	1040	BOOKBYTE DIRECT	-1957.08			
22	7600	7600	SPRINGER	-1852.73			
23	7473	7473	SINAUER ASSOCIATES INC	-1612.14			
24	7608	7608	STACKPOLE BOOKS	-1513.28			
25	7093	7093	SAGAMORE PUBLISHING, LLC	-1510.96			
26	6348	6348	PERSEUS DISTRIBUTION INC.	-1501.69			
27	0898	0898	ATHENA SCIENTIFIC	-1335.42			
28	8753	8753	WAVELAND PRESS, INC.	-1167.56			
29	1708	1708	CALIF. PRINCETON	-1001.10			
30			FULFILLMENT SERVICES				
31	3844	3844	HACKETT PUBLISHING	-553.76			
32			COMPANY				
33	0904	0904	ATWOOD PUBLISHING, LLC	-538.92			
34	7592	7592	SPRINGER PUBLISHING	-528.00			
35			COMPANY				
36	5374	5374	MERIDIAN MAGAZINE	-526.68			

### \*\*\*Step 4: Format the imported data.

There exist various flaws in the imported data. First, there are additional spaces within each cell that must be trimmed. Also, if the vendor is longer than a specified length, then parts of the vendor get cut off and moved to the next row. We need to concatenate that portion of the name and delete the blank rows. We also need to delete the extra vendor names that exist in column two. Additionally, we need to sort the data to make it easier to transfer into the year summary sheet. Finally, the heading will be renamed to reflect the month and year of the data. These types of formatting require relatively simple code:

Dim FinalRow As Long

Dim Col As Long

```
FinalRow = Sheets("CB").Cells(Rows.Count, 1).End(xlUp).Row
```

Loop to remove all unnecessary spaces from the cells

```
For Row = 1 To FinalRow
```

```
For Col = 1 To 4
```

```
Cells(Row, Col).Value = Trim(Cells(Row, Col).Value)
```

```
Next
```

To concatenate the cut off part of the vendor name

```
If Cells(Row, 1).Value = "" And Cells(Row, 2).Value = "" And Cells(Row, 4).Value = "" Then
```

```
Cells(Row - 1, 3).Value = Cells(Row - 1, 3).Value & " " & Cells(Row, 3).Value
```

```
End If
```

```
Next
```

To delete the useless column of data

```
Cells(2, 2).EntireColumn.Delete
```

```
For Row = FinalRow To 2 Step -1
```

```
If Cells(Row, 1).Value = "" And Cells(Row, 3).Value = "" Then
```

```
Cells(Row, 1).EntireRow.Delete
```

```
End If
```

```
Next
```

Loop from bottom of data to top deleting blank rows

```
Columns("A:C").EntireColumn.AutoFit
```

```
Range("A2").Select
```

```
Range(Selection, Selection.End(xlToRight)).Select
```

```
Range(Selection, Selection.End(xlDown)).Select
```

```
Sheets("CB").Sort.SortFields.Clear
```

```
Sheets("CB").Sort.SortFields.Add Key:=Range(Cells(3, 1), Cells(3, 1).End(xlDown)) _  
    , SortOn:=xlSortOnValues, Order:=xlAscending
```

Selects the range of the data and sorts the data ascending based on vendor

```
With Sheets("CB").Sort
```

```
.SetRange Selection
```

```
.Header = xlYes
```

```
.MatchCase = False
```

```
.Orientation = xlTopToBottom
```

```
.SortMethod = xlPinYin
```

```
.Apply
```

```
End With
```

```
FinalRow = Sheets("CB").Cells(Rows.Count, 1).End(xlUp).Row
```

```
Range(Cells(3, 1), Cells(FinalRow, 1)).Select
```

```
Cells(3, 7).Value = 1
```

```
Cells(3, 7).Copy
```

```
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlMultiply
```

```
Application.CutCopyMode = False
```

```
Cells(3, 7).ClearContents
```

```
Selection.HorizontalAlignment = xlLeft
```

The vendor numbers are stored as text. Changing the cells to number format does not correct this. To change to numbers, I copied 1 and opted to paste, special, multiply the range of vendor numbers.

```
Range(Cells(3, 3), Cells(FinalRow, 3)).Select
```

```
Selection.NumberFormat = "#,##0.00_);[Red](#,##0.00)"
```

Changes the credit balances to number format and turns them red

```
Sheets("CB").Cells(1, 1).Value = MonthName & " " & YearNum & " Credit Balances"
```

Gives the data a heading which specifies the month and year

### \*\*\*Step 5: Transfer data to year summary file.

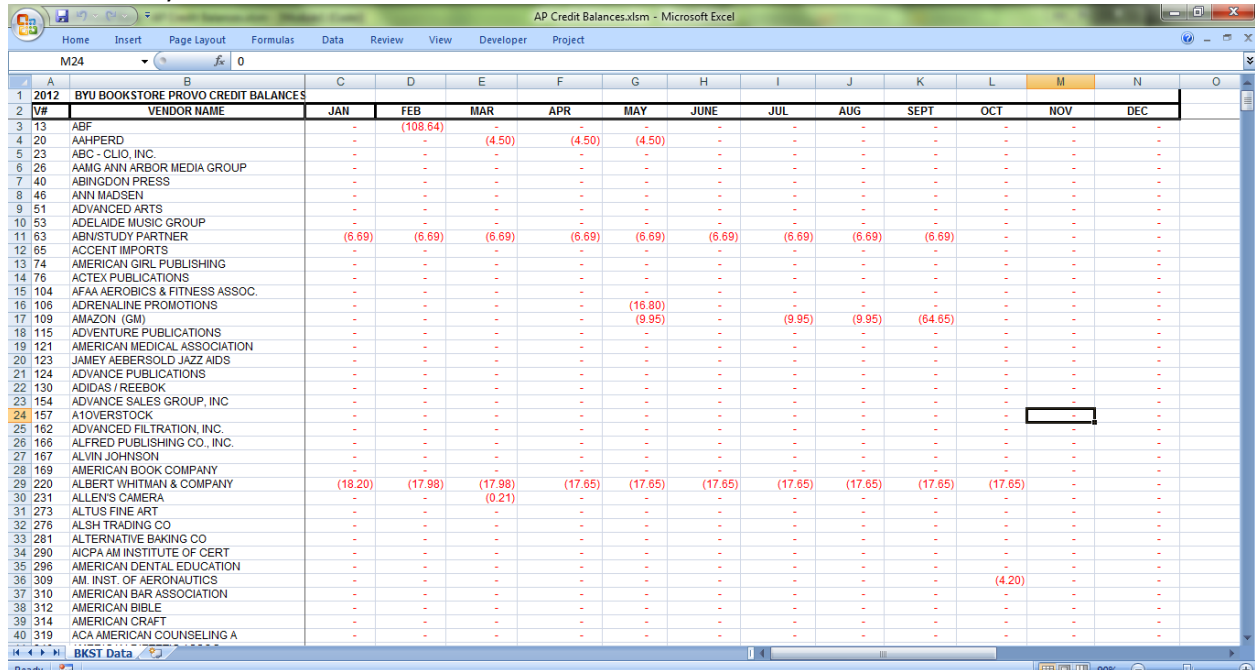
Now that the month data has been formatted, it is ready to be entered into the year summary file. The two sheets look like this:

#### Month Data



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	November 2012 Credit Balances														
2	AP VENDOR	Name	BALANCE DUE												
3	23	ABC - CLIO, INC.	(38.16)												
4	109	AMAZON (GM)	(398.30)												
5	220	ALBERT WHITMAN & COMPANY	(352.96)												
6	309	AM. INST. OF AERONAUTICS & ASTRONAUTICS	(83.96)												
7	405	AMERICAN MATHEMATICAL SOCIETY	(403.20)												
8	698	APHA PUBLICATIONS	(2,082.50)												
9	704	APPLE	(23,061.00)												
10	705	APPLE COMPUTER, INC SVC	(4,696.80)												
11	788	ARTECH HOUSE	(309.60)												
12	823	ASHGATE PUBLISHING COMPANY	(423.92)												
13	898	ATHENA SCIENTIFIC	(1,335.42)												
14	904	ATWOOD PUBLISHING, LLC	(538.92)												
15	937	AUNT LUTE BOOKS	(122.04)												
16	970	AUTOMATED BUSINESS PRODUCTS	(280.80)												
17	995	BADEN SPORTS, INC.	(447.70)												
18	1040	BOOKBYTE DIRECT	(1,957.08)												
19	1335	BOXERCRAFT INCORPORATED	(17.65)												
20	1385	BRANDEN BOOKS	(163.95)												
21	1458	BUSINESS LEARNING SOFTWARE, INC	(178.00)												
22	1593	BRUZER SPORTSGEAR LTD	(91.02)												
23	1701	CALICO COTTAGE, INC.	(15.25)												
24	1708	CALIF. PRINCETON FULFILLMENT SERVICES	(1,001.10)												
25	1743	CAREER PRESS / NEW PAGE BOOKS	(98.98)												
26	1772	CAROLINA ACADEMIC PRESS	(280.00)												
27	1900	CHELSEA GREEN PUBLISHING CO.	(16.38)												
28	1928	CHINA BOOKS & PERIODICALS	(439.76)												
29	2033	CLAITOR'S LAW BOOKS & PUBLISHING INC.	(43.00)												
30	2170	CONNIE SOKOL	(17.97)												
31	2256	CUP SERVICES	(300.34)												
32	2515	THE DAVID BROWN BOOK COMPANY	(46.65)												
33	2548	DEER STAGS	(30.00)												
34	3288	FOLLETT HIGHER EDUCATION GROUP	(73,554.95)												
35	3357	FOUNDATION ARTS	(2.03)												
36	3497	GALF GROU IP - 344394TR	(146.35)												

#### Year Summary



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	2012	BYU BOOKSTORE PROVISO CREDIT BALANCES													
2	#	VENDOR NAME	JAN	FEB	MAR	APR	MAY	JUNE	JUL	AUG	SEPT	OCT	NOV	DEC	
3	13	ABF	-	(108.64)											
4	20	A&H PERD	-	-	(4.50)	(4.50)	(4.50)	-	-	-	-	-	-	-	
5	23	ABC - CLIO, INC.	-	-	-	-	-	-	-	-	-	-	-	-	
6	26	A&M AN ARBOR MEDIA GROUP	-	-	-	-	-	-	-	-	-	-	-	-	
7	40	ABINGDON PRESS	-	-	-	-	-	-	-	-	-	-	-	-	
8	46	ANN MADSEN	-	-	-	-	-	-	-	-	-	-	-	-	
9	51	ADVANCED ARTS	-	-	-	-	-	-	-	-	-	-	-	-	
10	53	ADELAIDE MUSIC GROUP	-	-	-	-	-	-	-	-	-	-	-	-	
11	63	ABIN STUDY PARTNER	(6.69)	(6.69)	(6.69)	(6.69)	(6.69)	(6.69)	(6.69)	(6.69)	(6.69)	-	-	-	
12	65	ACCENT IMPORTS	-	-	-	-	-	-	-	-	-	-	-	-	
13	74	AMERICAN GIRL PUBLISHING	-	-	-	-	-	-	-	-	-	-	-	-	
14	76	ACTEX PUBLICATIONS	-	-	-	-	-	-	-	-	-	-	-	-	
15	104	AFAA AEROBICS & FITNESS ASSOC.	-	-	-	-	-	-	-	-	-	-	-	-	
16	106	ADRENALINE PROMOTIONS	-	-	-	-	(16.80)	-	-	-	-	-	-	-	
17	109	AMAZON (GM)	-	-	-	-	(9.95)	-	(9.95)	(9.95)	(64.65)	-	-	-	
18	115	ADVENTURE PUBLICATIONS	-	-	-	-	-	-	-	-	-	-	-	-	
19	121	AMERICAN MEDICAL ASSOCIATION	-	-	-	-	-	-	-	-	-	-	-	-	
20	123	JAMEY AEBERSOLD JAZZ AIDS	-	-	-	-	-	-	-	-	-	-	-	-	
21	124	ADVANCE PUBLICATIONS	-	-	-	-	-	-	-	-	-	-	-	-	
22	130	ADIDAS / REEBOK	-	-	-	-	-	-	-	-	-	-	-	-	
23	154	ADVANCE SALES GROUP, INC	-	-	-	-	-	-	-	-	-	-	-	-	
24	157	A1OVERSTOCK	-	-	-	-	-	-	-	-	-	-	-	-	
25	162	ADVANCED FILTRATION, INC.	-	-	-	-	-	-	-	-	-	-	-	-	
26	166	ALFRED PUBLISHING CO., INC.	-	-	-	-	-	-	-	-	-	-	-	-	
27	167	ALVIN JOHNSON	-	-	-	-	-	-	-	-	-	-	-	-	
28	169	AMERICAN BOOK COMPANY	-	-	-	-	-	-	-	-	-	-	-	-	
29	220	ALBERT WHITMAN & COMPANY	(18.20)	(17.98)	(17.98)	(17.65)	(17.65)	(17.65)	(17.65)	(17.65)	(17.65)	(17.65)	-	-	
30	231	ALLEN'S CAMERA	-	-	(0.21)	-	-	-	-	-	-	-	-	-	
31	273	ALTUS FINE ART	-	-	-	-	-	-	-	-	-	-	-	-	
32	276	ALSH TRADING CO	-	-	-	-	-	-	-	-	-	-	-	-	
33	281	ALTERNATIVE BAKING CO	-	-	-	-	-	-	-	-	-	-	-	-	
34	290	AICPA AM INSTITUTE OF CERT	-	-	-	-	-	-	-	-	-	-	-	-	
35	296	AMERICAN DENTAL EDUCATION	-	-	-	-	-	-	-	-	-	-	-	-	
36	309	AM. INST. OF AERONAUTICS	-	-	-	-	-	-	-	-	-	(4.20)	-	-	
37	310	AMERICAN BAR ASSOCIATION	-	-	-	-	-	-	-	-	-	-	-	-	
38	312	AMERICAN BIBLE	-	-	-	-	-	-	-	-	-	-	-	-	
39	314	AMERICAN CRAFT	-	-	-	-	-	-	-	-	-	-	-	-	
40	319	ACA AMERICAN COUNSELING A	-	-	-	-	-	-	-	-	-	-	-	-	

As you can see above, many vendors in the year summary do not appear in the month data because they have a credit balance of zero. Also, there exist vendors in the month data that do not appear in the

The outside loop goes through each row of data on the month sheet (the data to be transferred). The first inside loop selects each row in the year summary sheet. The purpose of the inside loop is to find a vendor match in the year summary sheet. Within the first inside loop, if the value in column a of the month sheet equals the value of column a of the year summary sheet, then the value in the relative month in the year file (measured by MonthNum + 2 columns) is equal to the value in column c (the credit balance amount) of the month file. The next step is to note on the month file next to the credit balance that this vendor and balance existed previously in the year file ("Existing Balance" will appear in column e). Also, if the statement is true, then the start x value (where to start on the row for the inside loop) increases by one. That way the loop does not have to return and loop through rows that have already been inputted. If not for this, then the first inside loop would start at 3 for every single row in the outside loop. The loop also exits so that the loop does not continue pointlessly.

## New Month Sheet

AP Credit Balances.xlsx - Microsoft Excel					
Home Insert Page Layout Formulas Data Review View Developer Project					
November 2012 Credit Balances					
A	B	C	D	E	F
1	November 2012 Credit Balances				
2	AP VEND Name	BALANCE DUE			
3	ABC - CLIO, INC.	(38.16)	Existing Balance		
4	AMAZON (GM)	(398.30)	Existing Balance		
5	ALBERT WHITMAN & COMPANY	(352.96)	Existing Balance		
6	AM. INST. OF AERONAUTICS & ASTRONAUTICS	(83.96)	Existing Balance		
7	AMERICAN MATHEMATICAL SOCIETY	(403.20)	Existing Balance		
8	APHIA PUBLICATIONS	(2,082.50)	Existing Balance		
9	APPLE	(23,061.00)	Existing Balance		
10	APPLE COMPUTER, INC SVC	(4,696.80)	Existing Balance		
11	ARTECH HOUSE	(309.60)	Existing Balance		
12	ASHGATE PUBLISHING COMPANY	(423.92)	Existing Balance		
13	ATHENA SCIENTIFIC	(1,335.42)	Existing Balance		
14	ATWOOD PUBLISHING, LLC	(538.92)	Existing Balance		
15	AUNT LUTE BOOKS	(122.04)	Existing Balance		
16	AUTOMATED BUSINESS PRODUCTS	(290.90)	Existing Balance		
17	BADEN SPORTS INC.	(447.70)	Existing Balance		
18	BOOKBYTE DIRECT	(1,957.08)	Existing Balance		
19	BOXERCRAFT INCORPORATED	(17.65)	New Balance		
20	BRANDEN BOOKS	(163.95)	Existing Balance		
21	BUSINESS LEARNING SOFTWARE , INC	(178.00)	Existing Balance		
22	BRUZER SPORTSGEAR LTD	(91.02)	New Balance		
23	CALICO COTTAGE, INC.	(15.25)	Existing Balance		
24	CALIF. PRINCETON FULFILLMENT SERVICES	(1,001.10)	Existing Balance		
25	CAREER PRESS / NEW PAGE BOOKS	(98.98)	Existing Balance		
26	CAROLINA ACADEMIC PRESS	(280.00)	Existing Balance		
27	CHELSEA GREEN PUBLISHING CO.	(16.38)	Existing Balance		
28	CHINA BOOKS & PERIODICALS	(439.76)	Existing Balance		
29	CLAITOR'S LAW BOOKS & PUBLISHING INC.	(43.00)	Existing Balance		
30	CONNIE SOKOL	(17.97)	Existing Balance		
31	CUP SERVICES	(300.34)	Existing Balance		
32	THE DAVID BROWN BOOK COMPANY	(46.65)	Existing Balance		
33	DEER STAGS	(30.00)	Existing Balance		
34	FOLLETT HIGHER EDUCATION GROUP	(73,554.95)	Existing Balance		
35	FOUNDATION ARTS	(2.03)	Existing Balance		
36	GAI F. GROUP - 344394TR	(146.35)	Existing Balance		

## New Year Summary Sheet

AP Credit Balances.xlsm - Microsoft Excel														
M100														
1	A	B	C	D	E	F	G	H	I	J	K	L	M	N
2	2012	BYU BOOKSTORE PROVO CREDIT BALANCES												
3		VENDOR NAME	JAN	FEB	MAR	APR	MAY	JUNE	JUL	AUG	SEPT	OCT	NOV	DEC
79	938	AUGSBURG FORTRESS PUB	-	-	-	-	-	-	-	-	-	-	-	-
80	948	AUGUSTA SPORTSWEAR	-	-	-	-	-	-	-	-	-	-	-	-
81	957	AVANTI	-	-	-	-	-	-	-	-	-	-	-	-
82	970	AUTOMATED BUSINESS	-	-	-	-	-	-	-	-	(14.04)	(14.04)	(280.80)	-
83	991	AYER COMPANY PUBLISHERS	(5.00)	(5.00)	(5.00)	(5.00)	(5.00)	(5.00)	(5.00)	(5.00)	(5.00)	-	-	-
84	993	AZTEC	-	-	-	-	-	-	-	-	-	-	-	-
85	995	BADEN SPORTS, INC.	-	-	-	-	-	-	-	-	-	-	(447.70)	-
86	997	BABALU INC	-	-	-	-	-	-	-	-	-	-	-	-
87	1008	BAKER & TAYLOR	-	-	-	-	-	-	-	-	-	-	-	-
88	1010	BAKER BOOK HOUSE	-	(1.57)	-	-	-	(8.48)	(12.53)	(12.54)	(4.16)	-	-	-
89	1036	BARBARIAN	-	-	-	-	-	(2.20)	-	-	-	-	-	-
90	1037	BAREFOOT BOOKS	(2.67)	(2.67)	-	-	-	-	-	-	-	-	-	-
91	1040	BOOKBYTE DIRECT	-	-	(9.43)	-	-	(0.17)	(0.17)	-	(19.16)	(97.85)	(1,957.08)	-
92	1044	BARBOUR PUBLISHING, INC.	-	-	-	-	-	-	-	-	-	-	-	-
93	1047	BARCHARTS	-	-	-	-	-	-	-	-	-	-	-	-
94	1079	BRUCE BARTON'S SCRIPT KIT	-	-	-	-	-	-	-	-	-	-	-	-
95	1080	BARRON'S EDUCATIONAL	-	-	-	-	-	-	-	-	-	-	-	-
96	1093	BLACK CLOVER	(21.25)	(21.25)	-	-	-	-	-	-	-	-	-	-
97	1115	STEPHEN BEHUNIN	-	-	-	-	-	-	-	-	-	-	-	-
98	1120	BEHRMAN HOUSE INC	(50.38)	-	-	-	-	-	-	-	-	-	-	-
99	1134	BENCHMARK BOOKS	-	-	-	-	-	-	-	-	-	-	-	-
100	1139	BM BOOKSHOP	-	-	-	-	-	-	-	-	-	-	-	-
101	1152	BRODER BROS	-	-	-	-	(24.69)	(30.11)	(20.49)	-	-	-	-	-
102	1156	BUZTRONICS	-	-	-	-	(4.65)	(4.65)	-	-	-	-	-	-
103	1198	BIC GRAPHIC USA	-	-	-	-	-	-	-	-	-	-	-	-
104	1205	BERGHAIN BOOKS	-	-	-	-	-	-	-	-	-	-	-	-
105	1226	BIBLICAL ARCHAEOLOGY SOCIETY	-	-	-	-	-	-	-	-	-	-	-	-
106	1233	BISOTUN	-	-	-	-	-	-	-	-	-	-	-	-
107	1235	BIC CORPORATION	-	-	-	-	-	-	-	-	-	-	-	-
108	1276	BOOK COUNTRY	-	-	-	-	-	-	-	-	-	-	-	-
109	1284	BNA BOOKS	-	-	-	-	-	-	-	-	-	-	-	-
110	1335	BOXERCRAFT INCORPORATED	-	-	-	-	-	-	-	-	-	-	(17.65)	-
111	1340	ISLANDER GROUP	-	-	-	-	-	-	-	-	-	-	-	-
112	1348	BOOK SALES INC	-	-	-	-	-	-	-	-	-	-	-	-
113	1352	BOOKS INTERNATIONAL	-	-	-	-	-	-	-	-	-	-	-	-
114	1373	BOOKS FOR LESS LLC	-	-	-	-	-	-	-	-	-	-	-	-
115	1381	PATIENCE BREWSTER, INC.	-	-	-	-	-	-	-	-	-	-	-	-
116	1385	BRANDEN BOOKS	(8.20)	(8.20)	(8.20)	(8.20)	(8.20)	(8.20)	(8.20)	(8.20)	(8.20)	(8.20)	(163.95)	-

The code to transfer month data to the year summary sheet:

Col = MonthNum + 2

Dim x As Long

x = 3

Outside Loop

For Row = 3 To FinalRow

Inside Loop #1

For BKSTRow = x To FinalRowBKSTData

If Sheets("CB").Cells(Row, 1).Value = Sheets("BKST Data").Cells(BKSTRow, 1).Value Then

Sheets("BKST Data").Cells(BKSTRow, Col).Value = Sheets("CB").Cells(Row, 3).Value

Sheets("BKST Data").Cells(BKSTRow, Col).NumberFormat = "#,##0.00\_);[Red](#,##0.00)"

Sheets("CB").Cells(Row, 5).Value = "Existing Balance"

x = BKSTRow + 1

Exit For

End If

Next

Inside Loop #2

If Sheets("CB").Cells(Row, 5).Value = "" Then

For BKSTRow = x To FinalRowBKSTData

If Sheets("BKST Data").Cells(BKSTRow, 1).Value > Sheets("CB").Cells(Row, 1).Value Then

Sheets("BKST Data").Cells(BKSTRow, 1).EntireRow.Insert

Sheets("BKST Data").Cells(BKSTRow, Col).Value = Sheets("CB").Cells(Row, 3).Value

Sheets("BKST Data").Cells(BKSTRow, Col).NumberFormat = "#,##0.00\_);[Red](#,##0.00)"

Sheets("BKST Data").Cells(BKSTRow, 1).Value = Sheets("CB").Cells(Row, 1).Value

Sheets("BKST Data").Cells(BKSTRow, 2).Value = Sheets("CB").Cells(Row, 2).Value

Sheets("CB").Cells(Row, 5).Value = "New Balance"

x = BKSTRow + 1

FinalRowBKSTData = Sheets("BKST Data").Cells(Rows.Count, 1).End(xlUp).Row

Exit For

End If

Next

End If

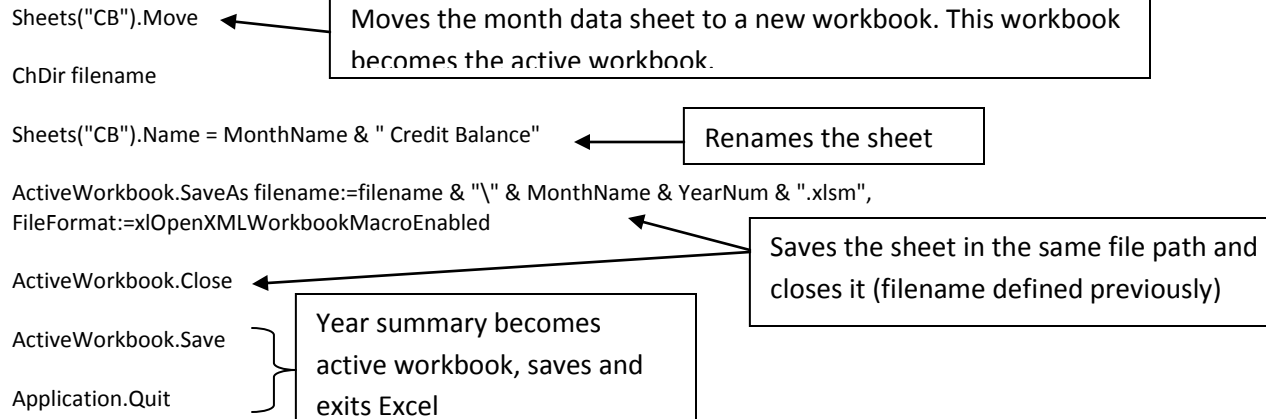
Next

See discussion on  
previous page



**\*\*\*Step 6: Move month data to new workbook, save in same directory, and save year summary.**

After looping through all the rows of data on the month sheet, the update of the year summary sheet is complete. The month sheet is ready to be moved to a new workbook and saved in the same directory as backup. The year summary will then be saved. The following code executes this step:



**\*\*\*Step7: Create new file when next year has arrived.**

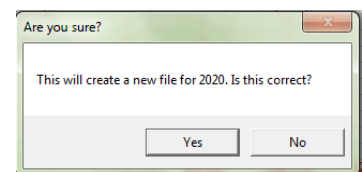
Note2 states that between step 2 and step 3, additional code determines whether a new file needs to be created for the next year. We will now return to discuss this step.

First, if the year (YearNum) is not equal to the year noted in Cell A1 of the year summary sheet, then this month data does not belong in this year's sheet and a new file needs to be created (Note: this will occur in February of each year). A message box will appear to confirm the user's intention. Then, a new folder will be created to house the new year file using the function `MkDir`. Next, we need to use the `SaveAs` function instead of the `SaveCopyAs` function. If we use the `SaveCopyAs` function, then the previous year summary file will remain the active workbook. Using the `SaveAs` function will be equivalent of making a copy of the current file in a new folder location. In addition, this will allow us to maintain VBA code across different years of files.

```
Dim NewYear As Boolean
NewYear = False
```

```
If YearNum <> Cells(1, 1).Value Then
result2 = MsgBox("This will create a new file for " & YearNum & ". Is this correct?", vbYesNo, "Are you sure?")
If result2 = vbYes Then
NewYear = True
```

```
On Error Resume Next
MkDir "C:\Users\David\Documents\.....Fall 2012\VBA EXCEL\Final Project\" & YearNum
On Error GoTo 0
```



```
ActiveWorkbook.Save
ActiveWorkbook.SaveAs filename:="C:\Users\David\Documents\.....Fall 2012\VBA EXCEL\Final Project\" _
& YearNum & "\AP Credit Blancs" & YearNum & ".xlsm", FileFormat:=xlOpenXMLWorkbookMacroEnabled
```

```
Else
Exit Sub
End If
End If
```

This following code is then executed to format the new year sheet. We first need to change the year of the file in Cell A1. Then, the code deletes all rows except for two. These will be the lower and upper bounds of the vendor numbers so that when the loops are executed to bring in the month data, the rows will be inserted between these bounds. After the loops are executed, the final code at the bottom is executed. This deletes those lower and upper bounds.

```
Dim Row As Long
Dim BKSTRow As Long
Dim FinalRowBKSTData As Long
```

```
If NewYear = True Then
```

```
Cells(1, 1).Value = YearNum
```

```
FinalRowBKSTData = Sheets("BKST Data").Cells(Rows.Count, 1).End(xlUp).Row
```

```
For BKSTRow = FinalRowBKSTData To 5 Step -1
```

```
Sheets("BKST Data").Cells(BKSTRow, 1).EntireRow.Delete
```

```
Next
```

```
Cells(3, 1).Value = "0"
```

```
Cells(4, 1).Value = "99999"
```

Creates lower and upper bounds for  
inserting vendor numbers in Step 5

```
End If
```

```
If NewYear = True Then
```

```
Sheets("BKST Data").Cells(FinalRowBKSTData, 1).EntireRow.Delete
```

```
Sheets("BKST Data").Cells(3, 1).EntireRow.Delete
```

Code executed after Step 5 to  
delete lower and upper bounds

```
End If
```

## Discussion of Learning and Difficulties Encountered

This project was very fulfilling. It challenged my understanding of Visual Basic and I was really tested in creating a solution for this business problem. The main difficulty I encountered was the code created for step 5—inputting the month data into the year sheet. Many loops were performed that inserted rows and vendors in wrong locations before I developed a loop that would correctly input the data from the month sheet to the year sheet. One possibility that I considered in solving step 5 was to simply skip the new vendors and insert rows at the end of the year data. Then, I considered using the sort function to return the vendors to ascending order. Conceptually, I did not believe that doing it that way was as efficient as using two loops inside of another loop. I believe that using the loops is concise and more professional. I also experienced a little difficulty creating a new workbook for the next year. The challenge was how to save the file to a new folder while also maintaining the VBA code and keeping open the newly created file. I believe that the way I addressed the problem was the correct one.

## Assistance

I did not receive substantial assistance from another person or persons. I did receive some help via online searches of the problems I was encountering. I also received semester-long tutelage from the amazing Professor Gove Allen.

## Exhibit A: Full VBA Code

```
Sub callGetData(control As IRibbonControl)
    Call InputData
End Sub

Sub InputData()

Dim LastModified As Date
LastModified = FileDateTime("C:\Users\David\Documents\.....Fall 2012\VBA EXCEL\Final Project\Copy of credit balance.txt")

Dim WhatMonth(1 To 12) As String
Dim MonthName As Variant
Dim MonthNum As Integer
Dim YearNum As Integer

WhatMonth(1) = "January"
WhatMonth(2) = "February"
WhatMonth(3) = "March"
WhatMonth(4) = "April"
WhatMonth(5) = "May"
WhatMonth(6) = "June"
WhatMonth(7) = "July"
WhatMonth(8) = "August"
WhatMonth(9) = "September"
WhatMonth(10) = "October"
WhatMonth(11) = "November"
WhatMonth(12) = "December"

If Month(LastModified) = 1 Then
    MonthNum = 12
    YearNum = Year(LastModified) - 1
Else
    MonthNum = Month(LastModified) - 1
    YearNum = Year(LastModified)
End If

MonthName = WhatMonth(MonthNum)

result = MsgBox("The last file was downloaded from VR on " & LastModified & ". Are you trying to input " _
& MonthName & " " & YearNum & "?", vbYesNo, "Confirmation")

If result = vbNo Then

MsgBox "Please download after the first of the month."

Exit Sub

End If

Dim filename As String
```

```

filename = ThisWorkbook.Path

If Dir(filename & "\" & MonthName & YearNum & ".xlsm") <> vbNullString Then

MsgBox "Error. This month has already been inputted. Please wait until next month."

Exit Sub

End If


Dim NewYear As Boolean
NewYear = False

If YearNum <> Cells(1, 1).Value Then
result2 = MsgBox("This will create a new file for " & YearNum & ". Is this correct?", vbYesNo, "Are you sure?")
    If result2 = vbYes Then
        NewYear = True

        On Error Resume Next
        MkDir "C:\Users\David\Documents\.....Fall 2012\VBA EXCEL\Final Project\" & YearNum
        On Error GoTo 0

        ActiveWorkbook.Save
        ActiveWorkbook.SaveAs filename:="C:\Users\David\Documents\.....Fall 2012\VBA EXCEL\Final Project\" _
        & YearNum & "\AP Credit Blancs" & YearNum & ".xlsm", FileFormat:=xlOpenXMLWorkbookMacroEnabled

    Else
        Exit Sub
    End If
End If


Dim Row As Long
Dim BKSTRow As Long
Dim FinalRowBKSTData As Long

If NewYear = True Then

    Cells(1, 1).Value = YearNum

    FinalRowBKSTData = Sheets("BKST Data").Cells(Rows.Count, 1).End(xlUp).Row

    For BKSTRow = FinalRowBKSTData To 5 Step -1

        Sheets("BKST Data").Cells(BKSTRow, 1).EntireRow.Delete

    Next

    Cells(3, 1).Value = "0"
    Cells(4, 1).Value = "99999"

End If

Sheets.Add

With ActiveSheet.QueryTables.Add(Connection:= _
    "TEXT;C:\Users\David\Documents\.....Fall 2012\VBA EXCEL\Final Project\Copy of credit balance.txt", _

```

```

Destination:=Range("$A$1"))
.TextFileColumnDataTypes = Array(2, 2, 2, 1)
.Refresh BackgroundQuery:=False
End With

Range("C5").Select

ActiveWorkbook.Connections("Copy of credit balance").Delete

ActiveSheet.Name = "CB"

Dim FinalRow As Long
Dim Col As Long

FinalRow = Sheets("CB").Cells(Rows.Count, 1).End(xlUp).Row

For Row = 1 To FinalRow

    For Col = 1 To 4

        Cells(Row, Col).Value = Trim(Cells(Row, Col).Value)

    Next

    If Cells(Row, 1).Value = "" And Cells(Row, 2).Value = "" And Cells(Row, 4).Value = "" Then

        Cells(Row - 1, 3).Value = Cells(Row - 1, 3).Value & " " & Cells(Row, 3).Value

    End If

Next

Cells(2, 2).EntireColumn.Delete

For Row = FinalRow To 2 Step -1

If Cells(Row, 1).Value = "" And Cells(Row, 3).Value = "" Then

Cells(Row, 1).EntireRow.Delete

End If

Next

Columns("A:C").EntireColumn.AutoFit

Range("A2").Select
Range(Selection, Selection.End(xlToRight)).Select
Range(Selection, Selection.End(xlDown)).Select
Sheets("CB").Sort.SortFields.Clear
Sheets("CB").Sort.SortFields.Add Key:=Range(Cells(3, 1), Cells(3, 1).End(xlDown)) _
, SortOn:=xlSortOnValues, Order:=xlAscending

With Sheets("CB").Sort
.SetRange Selection
.Header = xlYes
.MatchCase = False
.Orientation = xlTopToBottom

```

```
.SortMethod = xlPinYin
.Apply
End With
```

```
FinalRow = Sheets("CB").Cells(Rows.Count, 1).End(xlUp).Row
```

```
Range(Cells(3, 1), Cells(FinalRow, 1)).Select
Cells(3, 7).Value = 1
Cells(3, 7).Copy
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlMultiply
Application.CutCopyMode = False
Cells(3, 7).ClearContents
Selection.HorizontalAlignment = xlLeft
Range(Cells(3, 3), Cells(FinalRow, 3)).Select
Selection.NumberFormat = "#,##0.00_);[Red](#,##0.00)"
```

```
Sheets("CB").Cells(1, 1).Value = MonthName & " " & YearNum & " Credit Balances"
```

```
FinalRowBKSTData = Sheets("BKST Data").Cells(Rows.Count, 1).End(xlUp).Row
```

```
Col = MonthNum + 2
```

```
Dim x As Long
```

```
x = 3
```

```
For Row = 3 To FinalRow
```

```
For BKSTRow = x To FinalRowBKSTData
```

```
If Sheets("CB").Cells(Row, 1).Value = Sheets("BKST Data").Cells(BKSTRow, 1).Value Then
```

```
Sheets("BKST Data").Cells(BKSTRow, Col).Value = Sheets("CB").Cells(Row, 3).Value
```

```
Sheets("BKST Data").Cells(BKSTRow, Col).NumberFormat = "#,##0.00_);[Red](#,##0.00)"
```

```
Sheets("CB").Cells(Row, 5).Value = "Existing Balance"
```

```
x = BKSTRow + 1
```

```
Exit For
```

```
End If
```

```
Next
```

```
If Sheets("CB").Cells(Row, 5).Value = "" Then
```

```
For BKSTRow = x To FinalRowBKSTData
```

```
If Sheets("BKST Data").Cells(BKSTRow, 1).Value > Sheets("CB").Cells(Row, 1).Value Then
```

```
Sheets("BKST Data").Cells(BKSTRow, 1).EntireRow.Insert
```

```
Sheets("BKST Data").Cells(BKSTRow, Col).Value = Sheets("CB").Cells(Row, 3).Value
```

```
Sheets("BKST Data").Cells(BKSTRow, Col).NumberFormat = "#,##0.00_);[Red](#,##0.00)"
```

```
Sheets("BKST Data").Cells(BKSTRow, 1).Value = Sheets("CB").Cells(Row, 1).Value
```

```
Sheets("BKST Data").Cells(BKSTRow, 2).Value = Sheets("CB").Cells(Row, 2).Value
```

```
Sheets("CB").Cells(Row, 5).Value = "New Balance"
```

```

        x = BKSTRow + 1

        FinalRowBKSTData = Sheets("BKST Data").Cells(Rows.Count, 1).End(xlUp).Row

        Exit For
    End If
Next

End If
Next

If NewYear = True Then

    Sheets("BKST Data").Cells(FinalRowBKSTData, 1).EntireRow.Delete

    Sheets("BKST Data").Cells(3, 1).EntireRow.Delete

End If

Sheets("CB").Move

ChDir filename

Sheets("CB").Name = MonthName & " Credit Balance"

ActiveWorkbook.SaveAs filename:=filename & "\" & MonthName & YearNum & ".xlsm",
FileFormat:=xlOpenXMLWorkbookMacroEnabled

ActiveWorkbook.Close

ActiveWorkbook.Save

Application.Quit

End Sub

```