

Daniel Holbrook  
(DanMHolbrook@yahoo.com)

---

## Executive Summary

I have been serving in the supply section of a unit in the Utah Army National Guard for the past 8 years. The battalion I serve in is divided up into three different companies. Each company within the battalion has a supply sergeant with a debit card for purchases. Most items are ordered in advance, but periodically a supply sergeant may need to make an emergency purchase at a local store. The supply sergeant for the battalion is periodically allocated money; he then allocates that money out to the supply sergeants, who then can spend the money using their cards. The battalion needed a spreadsheet that would keep track of the allocation of money to each company, the transactions occurring for each card, and an up to date balance for each card.

On the first sheet of the spreadsheet the battalion sergeant can view the balance on the cards for each of the three companies (BSC, B, and C companies). On the first sheet there is also a button connected to a form to allocate or remove money from any of the companies. The next three sheets are for each of the company supply sergeants. On each sheet there is a button to input a new transaction including the date, price, purchase description, and quantity. That information is input onto the sheet along with a new transaction number and a purchase total. Another button is on each sheet shows a form to delete a transaction. After any type of entry the balances are updated.

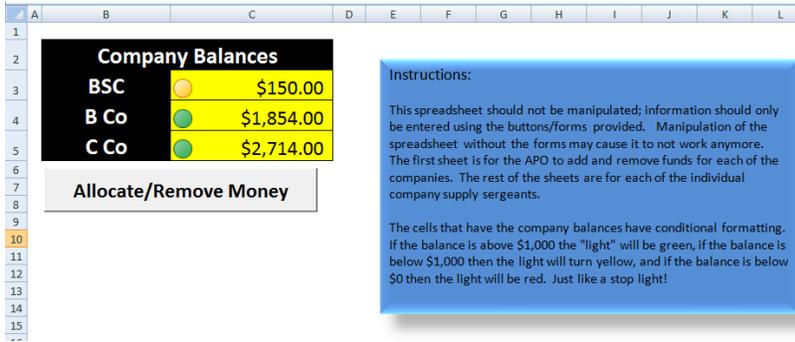
Only one workbook is necessary because all supply sergeants can have access to the workbook on a shared drive on the unit network.

And that's my spreadsheet. I hope you enjoy it.

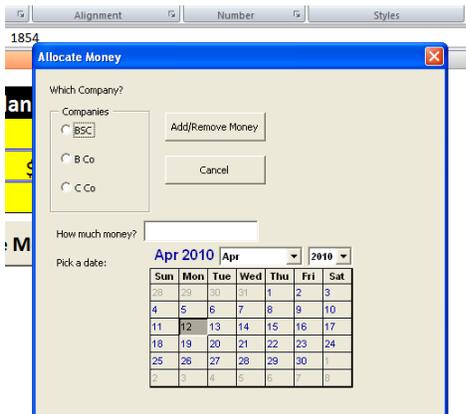
## Implementation Documentation

### Sheet 1

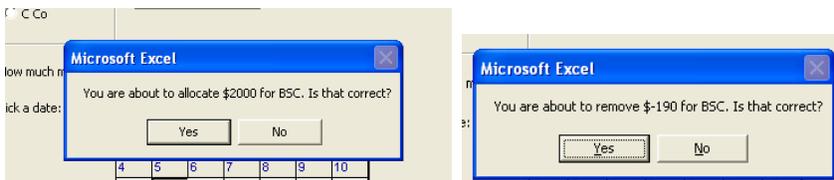
Below is a screenshot of the first sheet in the workbook. As shown there is a balance for each of the three companies. Code exists to update the balances whenever data is manipulated. Next to each of the balances there is a green light, yellow light, or red light. The "lights" are based on conditional formatting. If the balance is above \$100 then the light is green, if it falls below \$100 it turns yellow, and if the balance is below \$0 then the light is red for a credit balance. This first sheet also contains brief instructions for the user.



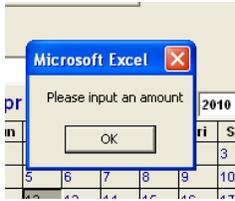
If the "Allocate/Remove Money" is clicked then the following form will appear.



The user can select a company, input an amount, and select a date. The calendar automatically starts with the current date. If the user selects "Cancel" then the form is unloaded, closes, and no data is entered. If the user selects "Add/Remove Money" then a message box with the information from the form appears. The code recognizes if the transaction is an addition or removal depending on if the user input a positive or negative number and changes the wording. The user is asked if the information is correct.



If a company is not selected or an amount is not entered the following message box will appear requesting the user enter the information, and the user will be returned to the form to enter the information.



If the user selects yes then the sheet that has the data for the selected company is activated. A new transaction is input into the table on a new row with a new transaction number, the date, amount, and transaction description. Whenever a new transaction of any type is put into one of the tables the code finds the highest transaction number used and adds 1 to it to create a new number that has never been used. Following is a screenshot of the table with a recently added transaction for an addition and a removal of funds.

C Company Balance:		(\$286.00)		Input New Transactions		Delete Transaction	
Transaction #	Date	Item Description	Price	Quantity	Vendor Info	Transaction Amount	
7	Monday, April 12, 2010	Removed Funds	NA	NA	NA	\$	(1,000.00)
6	Monday, April 12, 2010	New Funds	NA	NA	NA	\$	2,000.00
5	Monday, April 12, 2010	Boots	\$ 90.00	25	REI	\$	(2,250.00)
4	Tuesday, April 13, 2010	New Funds	NA	NA	NA	\$	1,000.00
3	Friday, April 09, 2010	Hats	\$ 3.00	12	Walmart	\$	(36.00)



### Sheets 2-4

The following three sheets are labeled for each of the companies. Each sheet functions and is formatted identically except for the fun graphic to the right. Following is a screenshot of the sheet for the first company (BSC).

BSC Company Balance:		\$150.00		Input New Transactions		Delete Transaction	
Transaction #	Date	Item Description	Price	Quantit	Vendor Info	Transaction Amount	
7	Friday, April 09, 2010	Tent	\$ 320.00	2	REI	\$	(640.00)
6	Thursday, April 08, 2010	Paper	\$ 5.00	20	WalMart	\$	(100.00)
5	Tuesday, April 06, 2010	Rope	\$ 20.00	3	WalMart	\$	(60.00)
4	Monday, April 12, 2010	New Funds	NA	NA	NA	\$	3,000.00
3	Monday, April 12, 2010	Boots	\$ 90.00	25	REI	\$	(2,250.00)
1	Friday, April 09, 2010	New Funds	NA	NA	NA	\$	200.00



At the top is the current balance which is calculated through the code and is just the sum of all the transaction amount totals in column "H". Similar to the first sheets the balance cell is formatted with a green, yellow, or red stoplight to warn the supply sergeant if a balance is getting low. The data can be sorted by transaction #, date, item description, price quantity, vender, or total amount.

When a supply sergeant needs to input a new transaction he can click the "Input New Transaction" button. A form like the one in the screenshot below will appear.

The 'New Transaction' dialog box contains the following information:

- Item Description: Tent
- Quantity: 2
- Price: 320
- Vendor: REI
- Transaction Date: Apr 2010

The calendar shows the month of April 2010 with the 9th selected.

The user inputs the item description, price, quantity, and vendor. The calendar automatically has the current date. If the user selects save and any information is missing then one of the following message boxes will appear requesting the user input the data, and the user will be returned to the form.

Two error message boxes are shown:

- Left: "Please input a Quantity."
- Right: "Please input a Vendor."

If all information is entered correctly when the user clicks save, then a message box like the one below will appear. The user is prompted to verify that the information entered is correct. If the user selects yes then the data is entered into the table on the sheet, otherwise the user is returned to the form. A new transaction number is created for that transaction.

The confirmation dialog box displays the following information:

- You have entered the following information:
- Item: Tent
- Price: \$320
- Quantity: 2
- Vendor: REI
- Is that correct?

Buttons: Yes, No

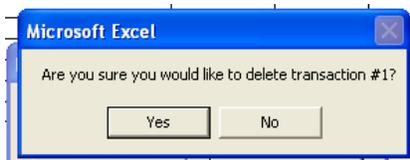
If the user needs to delete a transaction then he only needs to click the "Delete Transaction" button. The following form appears.



The user is then prompted to enter a transaction number. If a number is not entered, or a number that does not exist is entered, then the following message box appears.



If a number is entered and it is valid then the following message box appears and the user is prompted to validate the deletion.



If the user selects yes then the code searches for that transaction number and deletes the entire row that corresponds to the transaction number.

That is what this spreadsheet does. The sheet will be extremely useful to the unit in helping them keep track of the transaction and balances for each of the companies. The battalion supply sergeant can know at any time how much money is available to the supply sergeants, and if it is necessary to allocate more money or move money from one company to another.

### Challenges/Difficulties

I only ran into two major difficulties while doing my work. First, I had originally put the data in actual tables but found a lot of difficulty in manipulating those tables and extracting data from them so I decided that it wasn't necessary and simplified it so the data isn't contained in tables. The second difficulty was finding a way to create a transaction number that doesn't exist. I wanted the code to always look through the current transaction numbers and find one that wasn't already being used. In the end I used a code that went through the numbers and found the highest number using the "max" Excel function, and then created a number that was one integer larger. Below is the code I used to solve the problem.

```
'Find a new transaction number
Dim transnumber As Integer
Range("B5").Offset.End(xlDown).Select
transnumber = Application.WorksheetFunction.max(Selection)
```