

# Binomial Option Pricing Model for Employee Stock Options

## Executive Summary

### Business Problem

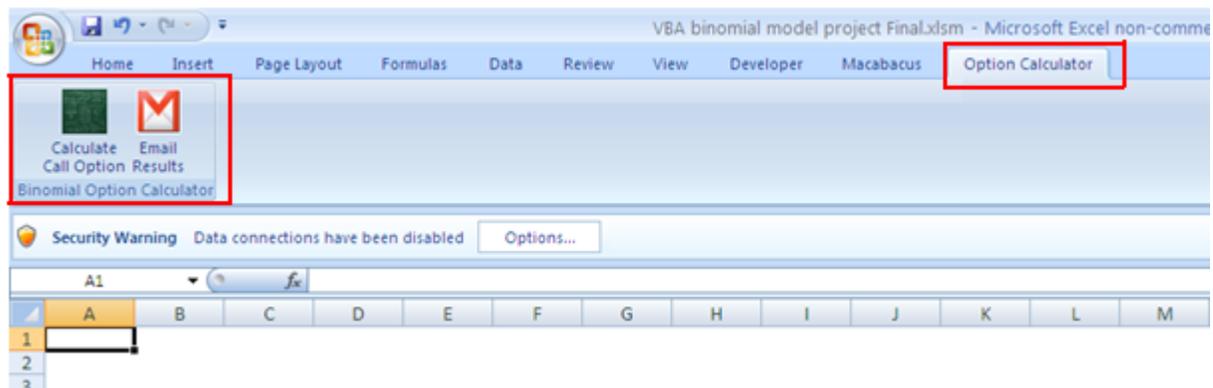
A common problem for accountants in today's world is determining an item's 'fair market value.' This is especially difficult for assets that are not traded on an exchange (such as the New York Stock Exchange). One example of this dilemma is an employee stock option. Many employees of public companies receive stock options as part of their compensation package. If these employees pass away before exercising the options, the fair value of the options must be included in the decedent's estate for tax purposes. The issue is that these options have different terms than those which *are* traded on the stock market. So how do we value these employee stock options? Fortunately, there are a few methods available for calculating the fair value of an employee stock option, one of which is the binomial option pricing method.

### System Description

The problem is that a binomial option pricing model requires some tedious calculations. In this project, I have utilized excel VBA to automate the calculation, formatting, and reporting involved with using a binomial option pricing model to value employee stock options. All that is required of the user are the relevant financial inputs for the option pricing model (gathered using a user form) and the program will calculate the fair value using (loops and arrays) and subsequently email the results to a colleague to review the work.

### Implementation Documentation

First, I added two buttons into the excel ribbon to facilitate the use of the program. The first button calculates the fair value of the employee stock option, while the second button will email the workbook to the desired third party. The screen shot below shows the ribbon modifications.





The stock tree shows the possible price movements of Facebook's common stock according to the inputs of the user (volatility, number of steps, time to maturity). Thus, the program uses a loop to populate the stock tree array with the values calculated by the binomial formula. These prices are then pasted and formatted into an exhibit using another loop. Once the stock tree is formed, the program uses another loop and array to calculate the various call option prices at each node (or index) in the array as seen by Exhibit C below.

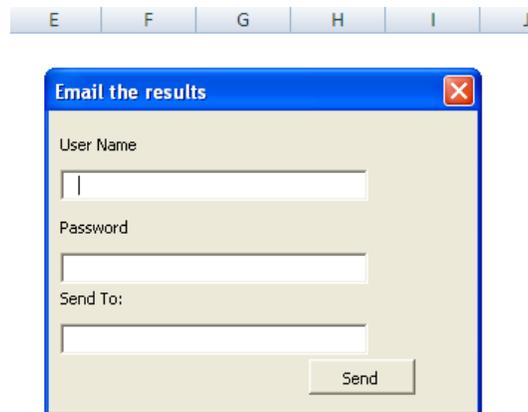
	A	B	C	D	E	F	G	H	I	J	K
1											
2	<b>Facebook, Inc.</b>										<b>Exhibit C</b>
3	<i>Valuation of Facebook, Inc. Employee Stock Options</i>										
4	<i>Valuation Date: 03/27/2013</i>										
5	<i>Call Tree</i>										
6											
7	\$4.56	\$6.85	\$10.01	\$14.23	\$19.62	\$26.03	\$33.35	\$41.73	\$51.31	\$62.27	\$74.80
8		\$2.63	\$4.18	\$6.47	\$9.71	\$14.07	\$19.62	\$26.03	\$33.35	\$41.73	\$51.31
9			\$1.33	\$2.26	\$3.75	\$6.04	\$9.40	\$14.02	\$19.62	\$26.03	\$33.35
10				\$0.54	\$1.0	\$1.81	\$3.21	\$5.52	\$9.12	\$14.02	\$19.62
11					\$0.16	\$0.32	\$0.63	\$1.26	\$2.49	\$4.84	\$9.12
12						\$0.02	\$0.05	\$0.10	\$0.23	\$0.50	\$1.09
13							\$0	\$0	\$0	\$0	\$0
14								\$0	\$0	\$0	\$0
15									\$0	\$0	\$0
16										\$0	\$0
17											\$0
18											
19											

Based on the underlying mathematical principles of binomial option pricing, we arrive at the true value of the employee stock option as shown by the value in cell A7.

This value along with the other inputs are then calculated and formatted in an exhibit for others to review as seen by Exhibit A below.

	A	B	C
1			
2	<b>Facebook, Inc.</b>		<b>Exhibit A</b>
3	<i>Valuation of Facebook, Inc. Employee Stock Options</i>		
4	<i>Valuation Date: 03/27/2013</i>		
5	<i>Inputs</i>		
6			
7	FB Stock Price as of the Valuation Date		\$26.1
8	Exercise Price		\$25.0
9	Valuation Date	3/27/2013	
10	Expected Life of the Option	2	
11	Number of Periods	10	
12	Dividend Yield	2.0%	
13	Volatility	30.0%	
14	Risk-free Rate	1.0%	
15			
16	<i>Calculated Inputs</i>		
17	u		1.14
18	d		0.874447
19	p		0.459085
20	1-p		0.540915
21			
22	Fair Market Value per Option		\$4.56
23			

Once you are comfortable with the analysis, you can email the workbook to the client or a colleague by clicking on the 'Email Results' button in the custom ribbon. This will bring up a user form that requires the sender's email and password as well as the recipient's email address as shown below.



The image shows a portion of an Excel spreadsheet with columns labeled E, F, G, H, I, and J. Below the spreadsheet is a dialog box titled "Email the results". The dialog box has a blue title bar with a close button (X) in the top right corner. It contains three text input fields: "User Name", "Password", and "Send To:". A "Send" button is located at the bottom right of the dialog box.

### Learning and Conceptual Difficulties

- Formatting syntax – Although the concepts behind excel formatting aren't very difficult; I had no prior knowledge regarding how to actually implement formatting syntax into VBA code. This was extremely useful to me because in many financial projects (especially in spreadsheets), formatting is key. Thus, getting to know more formatting syntax was helpful.
- Web query – Despite their seemingly simple nature, I have had trouble with web queries as I have received many errors when trying to run the code. I think this had to do with the exact way in which the web query is initiated with VBA syntax.
- Manipulating Gmail – Initially I had a problem being able to attach a copy of the workbook to an email to be sent out. The difficulty came around saving a copy of the file and calling its path in the sendGmail function. I was initially trying to attach the file that was still in use by the program, which was not allowed. To overcome this, I saved a copy of the workbook with a different file name, and then used that file as the email attachment.
- Arrays – I think that arrays and loops are essential parts in the logic of almost any program. A fundamental part of the binomial option pricing model (the stock tree that held the possible movements of the stock price) was easily handled by the computing concept of the array. This made me appreciate the usefulness of arrays and examine how I can use them to solve other computational problems.
- Modifying the ribbon – This piece of the project was particularly fun for me. When a program can be run from excel's ribbon, the average user will be astonished that such a thing can be done. It is also a quick way of calling out sub procedures that have been written. One thing that made the implementation of the ribbon modification very easy was Professor Gove Allen's ribbon wizard file.

### **Assistance Received**

The only substantial assistance that I received was the instruction I received from Professor Gove Allen. As previously mentioned, his ribbon modification wizard file was extremely helpful in modifying the excel ribbon instead of changing the xml code of the excel file. Thank you for the class and your help.