<u>**ID440: Honors Capstone**</u>

Financial Programming: Koios | an Excel Add-in in C#

April 23rd, 2010

Submitted by: Lubka A. Dagorova
Faculty Advisor: Professor Nathan Carter

Introduction to Excel Add-ins

With its first launch in 1984, Microsoft Excel was originally created for the Macintosh Operating System. However, through subsequent developments and the launch of the Microsoft Windows Operating System in 1987, Excel quickly became the most widely used spreadsheet application by both private users and business professionals (Sharon Parq Associates, Inc.) In addition to the fact that applications within the Microsoft Office Suite are entirely compatible with one another, one of the most significant program features that made this possible was a macro programming language known as Visual Basic for Applications (VBA) (Sharon Parq Associates, Inc.). Included in 1993, VBA made it possible for different users to customize Excel for specific needs by creating user-defined functions and automated processes.

In order to allow developers to share their customizations with other users, Excel created new file formats such as XLA and XLL that allowed these customizations to be “added on” to the software’s standard features. Hence, the term “Excel add-in” was created, and developers have since then created myriad add-ins that carry out very specific tasks, such as statistical analyses and hypothesis testing. While some of these have been created by Microsoft and are therefore automatically included in Excel, the vast majority has been created by third-party developers and must be installed separately (Microsoft Support).

Currently, there are two different types of add-ins that can be used: COM add-ins and automation add-ins. The primary difference between the two lies in the method in which users access the functionality that each offers. Simply put, automation add-ins are accessed through cell formulas in worksheets while COM add-ins, like Koios, are accessed through the graphical user interface, such as buttons on the Excel ribbon (Microsoft Support). For more advanced users, Microsoft explains the underlying technical differences in the following manner:

> COM Add-ins must be in-process COM servers that support the IDTExtensibility2 interface; however, Automation Add-ins can be in-process or out-of-process COM

> servers and implementation of IDTExtensibility2 is optional… All COM Add-ins must implement each of the five methods of this [IDTExtensibility2] interface: OnConnection, OnStartupComplete, OnAddinsUpdate, OnBeginShutDown, and OnDisconnection (Microsoft Support).

Both of these add-in types use the XLA and XLL file format. However, the primary different here is that XLA add-ins are typically written in VBA while XLL add-ins are typically written in some other programming language, such as C, C++, Visual Basic, and FORTRAN. In many cases, like Koios, C# is also used.

Benefits of Add-ins: How are they useful?

Because add-ins can be tailored to accommodate an arbitrary number of tasks, they offer many benefits to users:

- **Increase user efficiency:** Using add-ins allows users to spend less time on menial tasks, such as formatting and more time on more important tasks, such as data analysis. This may be of particular importance to professional users who use Excel to analyze spreadsheets that contain enormous amounts of data.
- **Reduce the learning curve:** Excel has become such a popular application in the business world that most professionals are already familiar with how to use the software. Therefore, add-ins simply build on that knowledge without requiring users to learn how to use an entirely new software platform.
- **New user capabilities:** Add-ins are created to perform specific tasks, many of which users may not otherwise be able to do in Excel. The wealth of options available when it comes to add-ins gives users the opportunity to analyze data in ways that they may not have thought of before. Thus, add-ins are also useful in that they can sometimes give a user a fresh perspective from which to evaluate information.

- **Exploit interoperability:** The various applications within the Microsoft Office Suite (Word, Excel, PowerPoint, Access, etc.) are built to be compatible with one another, allowing users to move data from one application to another with minimal effort. Therefore, data created through the use of add-ins can still be shared across organizations not only in Excel but in the rest of the Microsoft Office Suite as well, thereby encouraging collaboration.

While Koios takes advantage of all of these benefits to a certain extent, it is mostly concerned with enhancing user knowledge because it performs a function that gives the user access to data that would otherwise take a significant amount of time to accumulate. Koios is a finance add-in that extracts stock price data from different stock exchanges around the world and, taking into account commission charges and current exchange rates, informs the user if market mispricing exists. Hence, the fundamental finance principle around which KOIOS was built is what is known as the "Law of One Price," which states that, "in an efficient market, all identical goods must have only one price." While this theory should hold true in financial markets, this is not always the case. Thus, theoretically, it should not matter whether KOIOS is extracting data from a stock exchange in the United States, Canada, or China. Once the data from each is converted to a base currency, the prices should be equal. If they are not, there is an arbitrage opportunity, whereby the user can buy the stock at the lower price and sell the stock at the higher price to make a financial profit.

Methodology for Creating Add-in

Concept Phase

As is the case with most projects, the concept phase usually involves brainstorming and planning. Going into this project, only one thing was certain: a finance add-in would be created. However, the specifics in terms of what it would do and how it could be created (i.e. what programming language would be used) were uncertain.

Initially, the goal was to create an add-in that could suggest hedging strategies that limit a corporation's exposure to the foreign exchange market based on the user's risk preference and speculations about the market. After receiving inputs from the user, the add-in would have suggested short- and long-term strategies such as forward contracts, futures contracts, options contracts, swap, and money market hedges. However, Professor Jahangir Sultan of the Finance Department at Bentley University suggested that such an application might not be useful to finance professionals because advanced applications that perform this function already exist. So the target audience of the project then expanded to include the average user as well, since obviously many advanced finance applications exist but few of them are available at low or no cost. Thus, the idea for Koios was created since international arbitrage was a simple enough concept that it would be of interest to both finance professionals and to private home users.

Furthermore, the project was originally intended as an opportunity to learn the C++ programming language since preliminary research showed that this was the most widely used programming language for finance applications. However, given that project was specifically designed to be compatible with Excel, it quickly became apparent that C# or Visual Basic (VB), which is closely related to VBA, were more logical choices for this specific type of application since both were developed by Microsoft and were therefore entirely compatible with Excel. Consequently, they are better supported for this kind of work by Microsoft's developer tools. The differences between C# and VB are syntactic sugar, which simply means that some tasks are easier for a human to write in one language compared to another. However, since both C# and VB are both extensively used to program the .NET framework, the choice to use C# was based entirely on the fact that it shares syntax similarities with Java, which is a programming language that I already know.

Development Phase

Koios was developed with Microsoft Visual Studio 2008 (Professional Edition) on an HP6930p Elitebook running a 64-bit version of Windows 7 Professional.

Writing the code for the add-in involved the following steps:

1. Adding a tab to the Excel ribbon through which Koios could be accessed
2. Creating a window that would prompt the user to enter the required inputs
3. Fetching the data from the World Wide Web and importing it into an Excel Workbook
4. Performing the necessary calculations (i.e. converting the price to the base currency and adding commission charges) and generating a conclusion table with a suggested trading strategy

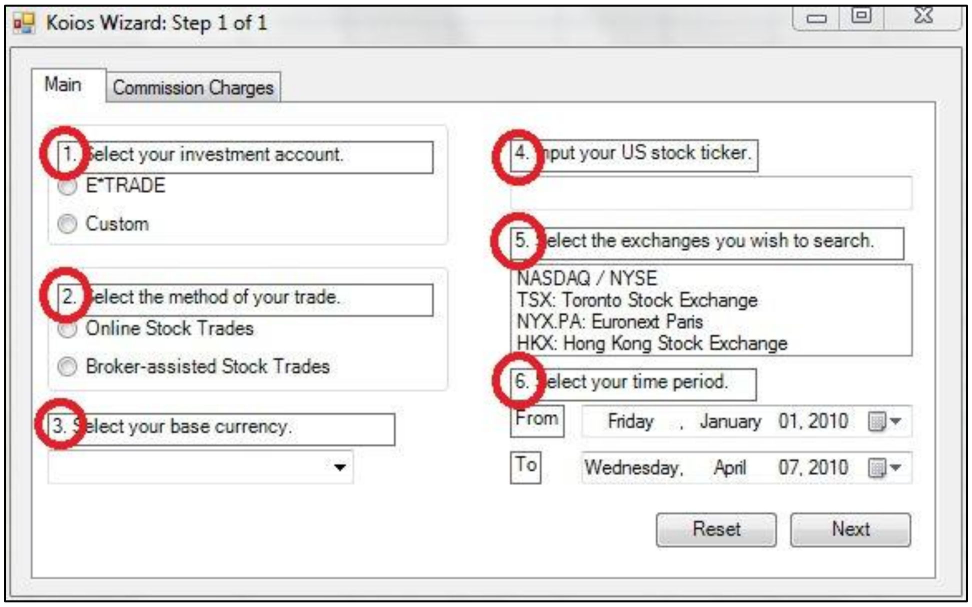
For more information about user workflow in Koios, see the add-in documentation available at http://web.bentley.edu/empl/c/ncarter/student_capstone.html.

Since Visual Studio includes templates for Excel add-ins written in C# and VB, the first and second steps were relatively simple. Adding the ribbon item was entirely automated and creating the window was done through a graphical user interface, which allowed user interface elements to be dragged and dropped into the window. From there, the properties of each element could be set through a properties window.

The third and fourth steps in the process, however, were the most time consuming and involved a significant amount of research. Koios was written in such a way that the entire process of creating a new workbook, fetching the data, performing the calculations, and generating the conclusion table all occur immediately when the user clicks the “next” button.

The following is a basic walkthrough that describes each section of code and what it does (please refer to Figure 1.1 for each "section" of code):

Figure 1.1



Koios Wizard: Step 1 of 1

Main | Commission Charges

1. Select your investment account.
E*TRADE
Custom

2. Select the method of your trade.
Online Stock Trades
Broker-assisted Stock Trades

3. Select your base currency.

4. Input your US stock ticker.

5. Select the exchanges you wish to search.
NASDAQ / NYSE
TSX: Toronto Stock Exchange
NYX.PA: Euronext Paris
HKX: Hong Kong Stock Exchange

6. Select your time period.
From Friday , January 01, 2010
To Wednesday, April 07, 2010

- Once the user has opened Excel and accessed Koios (for more information on this process, see the accompanying documentation found at *http://web.bentley.edu/empl/c/ncarter/student_capstone.html*, the Excel application is launched and a new workbook containing the default number of worksheets is created.
- Koios moves through successive rows of the worksheet for each new line of output.

***NOTE: Since this happens at many points throughout the code, the walkthrough will not identify each occurrence. In addition, since cell formatting also occurs extensively throughout the code, and this will also not be explicitly stated.

1. Koios provides a summary of the user inputs as shown in (a) – (e).

   (a) Koios determines which of the two investment accounts is selected in question 1 from Figure 1.1 and outputs the value in the current row.

(b) Koios determines which of the two trade methods is selected in question 2 from Figure 1.1 and outputs the value in the current row.

(c) Koios determines which item in the base currency ComboBox is selected and extracts a substring containing the first three letters of text (i.e. "EUR: Euro" becomes a new string that contains "EUR"). This value is outputted in the currency row.

(d) Koios simply returns the text from question 4 from Figure 1.1 in the current row.

2. Koios loops through each selected item in the exchange list box from question 5 in Figure 1.1 and outputs a string that contains all of the selected items, separated by "||". Within this loop, several sub-processes occur:
    a. Now that Koios has information on the investment account, the trade method, the base currency, and the exchange, it can determine the commissions that will be charged for each trade. A second loop goes through each selected item and determines which country the exchange is associated with as well as its base currency (i.e. NASDAQ/NYSE is associated with "USA" and "USD"). Within this second loop, the following steps occur:
        1. Behind the scenes, Koios goes to www.xe.com and saves the exchange rate from the selected base currency to the exchange's local currency.
        2. Depending on the investment account and type of trade, the appropriate commission charge is selected (these are hard-coded into the add-in) and converted to the base currency.
        3. A table is created containing the commission charge for each country, and the counter is incremented.
        4. A second table is created containing the exchange rate data for each local currency compared to the base currency.
3. Koios saves the "from" and "to" dates selected by the user, but it does not output them anywhere.

a. In addition, since Koios now has the stock ticker, it can search for the stock on multiple exchanges. However, since difference exchanges use different ticker symbols for the same company, Koios uses the US ticker to search Yahoo! Finance (UK) for the corresponding ISIN number. Once that's found, it performs a second search on Yahoo! Finance (UK) to find all the tickers on all the exchanges for the corresponding ISIN.

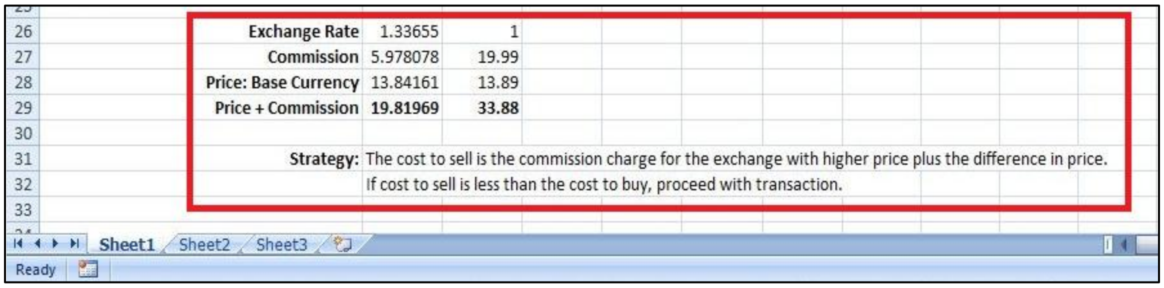
   i. Yahoo! Finance appends ticker symbols with suffixes for each exchange. For example, a stock traded on the Toronto Stock Exchange will have a ".TO" suffix. Since Koios only follows a few exchanges, it goes through the string that includes all possible ticker symbols for a given company to extract the ones that are of interest to the user.

   ii. A URL with the location of the historical stock price data is saved into the program, and each required piece of information is substituted with the user's own inputs. More specifically, in the URL below, each item in bold is substituted: http://ichart.finance.yahoo.com/table.csv?s=**[symbol]**&a=**[startMonth]**&b=**[startDay]**&c=**[startYear]**&d=**[endMonth]**&e=**[endDay]**&f=**[endYear]**&g=d&ignore=.csv. The code to initiate the web client and to replace parts of this particular URL was taken from a tutorial by Peter Bromberg, which can be found at the following URL: http://www.eggheadcafe.com/tutorials/aspnet/f651bcc7-1256-4e4d-bbc1-95f88500d86b/c-net-yahoo-stock-downl.aspx.

      1. Since the URL points to a *.csv file, it is relatively easy to read into Excel. Thus, a Stream is created with the data from the URL, and a Streamreader "reads" each line of the *.csv file through the use of a loop. Since such files are usually comma delimited, another loop goes through each line that gets read and splits the line at each comma. The resulting substrings are saved to a new column (i.e. Date, Open, High, Low, Close, Volume, and Adj Close).

iii. However, since not all stocks are traded on all exchanges, Koios determines whether the number of exchanges selected corresponds to the number of exchanges found. If they are equal, it simply outputs the stock price data. If they are not equal, it informs the user that the stock they are searching for was not found on all of the exchanges.

Figure 1.2



Exchange Rate	1.33655	1
Commission	5.978078	19.99
Price: Base Currency	13.84161	13.89
Price + Commission	**19.81969**	**33.88**

Strategy: The cost to sell is the commission charge for the exchange with higher price plus the difference in price. If cost to sell is less than the cost to buy, proceed with transaction.

- Once all of this data has been saved, where necessary, and outputted into the Excel sheet, Koios is ready to form a conclusion table with the suggested strategy as shown in Figure 1.2 above. First, it prints out the exchange rate and commission (in the base currency) for each country again. Second, it converts the most recent stock price available on each exchange into the base currency. Finally, it adds the commission to the stock price. The final strategy is a result of a comparison between the cost to buy the stock and the cost to sell the stock. For example, Figure 1.2 shows the conclusion table for GE stock on the NYSE and Euronext Paris. Since it is less expensive on the NYSE the user would want to buy it in the US and sell it in France. The price to buy the stock is equal to the price + commission, which is 19.82. The cost to sell the stock is equal to the commission charge for Euronext Paris plus the difference in the price. More specifically, it is 19.99 + (13.89 – 13.84), which is equal to 20.04. Therefore, since it would cost more to sell it than it would to buy it, there is no opportunity to make a financial profit. If the situation was reversed (i.e. the cost to buy was less than the cost to sell), the user could proceed with the transaction. (***NOTE: prices on Yahoo! Finance, and consequently in the spreadsheet, are delayed by approximately 15 minutes.)
- Control of the Excel sheet is returned to the user.

Implementation/Testing Phase

The implementation and testing phase is perhaps one of the most important because this is when bugs in the code are identified and resolved. Therefore, there is a strong relationship between this phase and the development phase.

The first part of the implementation/testing phase involved verifying that the correct data is extracted from www.xe.com and Yahoo! Finance. A few companies from each exchange were chosen, and the resulting information in the Excel spreadsheets was checked against the data available on the websites. Overall, Koios works seamlessly in this regard.

However, a few problems were encountered with the Yahoo! Finance service. First, it was initially unclear what currency Yahoo! Finance lists stock price data for each exchange as it was not explicitly stated on the stock's page or in the help and support pages. However, through comparisons with Google Finance and stock price data from the exchanges' websites, it quickly became apparent that prices are listed in the exchange's local currency. A second problem was that not all desired exchanges were available through Yahoo! Finance. For example, Koios is intended to be entirely compatible with E*TRADE, which trades on five international exchanges: Toronto Stock Exchange, Euronext Paris, Hong Kong Stock Exchange, Tokyo Stock Exchange, and London Stock Exchange (LSE) (E*TRADE). However, Yahoo! Finance does not include data on the Tokyo Stock Exchange, and the London Stock Exchange appears to trade certain stocks infrequently (London Stock Exchange). For example, one test case was for JPMorgan Chase (Ticker: JPM) stock on the NYSE and the London Stock Exchange (LSE). However, first, LSE appears to list stock price data in pence, which, on its own, is an easy problem to fix. Unfortunately, upon further inspection of their website, it appears that some stocks are traded in Euros and others in Dollars (London Stock Exchange). Therefore, it becomes unclear which exchange rate Koios should use for each company. Furthermore, since Yahoo! Finance is a free source of data, in order to ensure that it is in fact accurate, price data was compared to Bloomberg at the Hughey Center for

Financial Services at Bentley University. Regrettably, they did not have much historical price data for the two primary test cases: JPMorgan Chase and Bank of America. Thus, the decision was made to eliminate LSE from the add-in altogether.

The implementation/testing phase also relates to the finalization phase in that once the program was completed, it had to be tested on various machines with different operating systems. The Koios installation was tested on three operating systems: Windows XP Professional, Windows Vista Enterprise, and Windows 7 Basic and Professional. Once it installs the required prerequisites, which occurs automatically during installation, Koios works as intended on Windows Vista Enterprise and on both versions of Windows 7. Unfortunately, for reasons that are unclear, Koios does not appear to be compatible with Windows XP Professional. The installation process proceeds as intended, but the add-in itself does not appear to function correctly as it does not consistently search all the exchanges selected by the user. In addition, there were times when no output was generated at all but no error message was shown. The installation requires the following other prerequisites to proceed: Microsoft Office Excel 2007, Windows Installer 3.1, .NET Framework 3.5, Visual Studio Tools for the Office System 3.0 Runtime. All of the machines already have .NET Framework 3.5 and Microsoft Office Enterprise. The installation settings are such that any required prerequisites that are missing from the user's machine are automatically downloaded and installed from the Microsoft website, and this feature works flawlessly when installing the Visual Studio Tools for the Office System 3.0 Runtime. In addition, while there should be no problems running Koios on a standard Microsoft Office edition, this was not tested.

Finalization Phase

The finalization phase of the project involved three things:

1. Creating the Koios installer
2. Writing the accompanying documentation and tutorials
3. Publishing the program and all related documentation to a website.

The Koios installation was created automatically through Visual Studio using the publish feature, and it was later tested as described in the implementation/testing phase section. The documentation and tutorials were then created and published on the following website:

http://web.bentley.edu/empl/c/ncarter/student_capstone.html

Add-in Limitations: How can it be improved?

While Koios incorporates all the core functionality of the original plan, there are still many features that could be included to make it a more comprehensive and user-friendly program. While Koios is complete for the purposes of the Honors Capstone project, I intend to develop newer versions of the add-in that eliminate the weaknesses listed below:

- Koios currently only includes commission charges from E*TRADE and does not allow the user to input commission charges from other trading companies. The E*TRADE commission charges are currently hard-coded into the add-in. If E*TRADE were to change its pricing policies, they would **not** be automatically updated in the add-in. Thus, it would be better if Koios searched for these prices from the E*TRADE website instead so that the user could be assured that the correct commission charges are being applied. Furthermore, it would also be helpful to include a second tab in the wizard that allows the user to input custom commission charges.
- The base currencies that the user can currently select are all hard currencies, which are currencies that are not likely to depreciate suddenly in value. More simply, hard currencies to the most widely quoted, such as USD, EUR, CHF, etc. Thus, it would be better if Koios included soft currencies, which are the opposite of hard currencies, as well.
- Koios can currently only search for one company at a time. This was originally done to reduce the number of loops required in the source code. However, it would be beneficial to users if they could input a list of tickers, separated by commas.

- As mentioned, Koios currently only searches exchanges that can be accessed through E*TRADE. However, there are many other widely-used exchanges that the user may be interested in searching. It would therefore be better if users could search those as well.
- Koios assumes that the user knows where companies are traded. However, it would be much more useful to a user if Koios could suggest companies that a user could look at based on the exchanges they select. For example, if a user is interested in NYSE and TSX, Koios could extract a list of companies traded on each, find which are common to both, and return those to the user.
- Obviously, Koios relies heavily on user inputs. However, the usefulness of the add-in would be substantially increased if the add-in were capable of searching for arbitrage opportunities on its own. For example, once the process in the previous bullet is complete, perhaps Koios could automatically get price data for all the companies that are common to the selected exchanges and highlight the ones where arbitrage opportunities currently exist. This would significantly minimize Koios' dependence on the user, and it would make the add-in almost entirely self-sufficient.

Bibliography

E*TRADE. (n.d.). Retrieved from E*TRADE Financial: https://global.etrade.com/hk/en/home

Hong Kong Stock Exchange. (n.d.). Retrieved from HKEx: http://www.hkex.com.hk/eng/index.htm

London Stock Exchange. (n.d.). Retrieved from London Stock Exchange: http://www.londonstockexchange.com/home/homepage.htm

Microsoft Support. (n.d.). *Excel COM add-ins and Automation add-ins*. Retrieved from http://support.microsoft.com/kb/291392

MSDN - Microsoft Developer Network. (n.d.). *Visual Studio 2010 - Visual C#*. Retrieved from http://msdn.microsoft.com/en-us/library/kx37x362%28v=VS.100%29.aspx

NYSE Euronext. (n.d.). Retrieved from Euronext Paris Stock Exchange: http://www.euronext.com/landing/indexMarket-18812-EN.html

Sharon Parq Associates, Inc. . (n.d.). *Understanding Add-Ins*. Retrieved from http://excel.tips.net/Pages/T002276_Understanding_AddIns.html

TMX Group. (n.d.). Retrieved from Toronto Stock Exchange: http://www.tmx.com/

XE. (n.d.). Retrieved from XE - The World's Favority Currency Site: http://www.xe.com/

Yahoo. (n.d.). Retrieved from Yahoo! Finance: http://finance.yahoo.com/