Graham Orme

4/13/10

VBA Final Project

**Querying a Purchase Price Index Database from the Bureau of Labor Statistics Website to**

**Analyze and Anticipate Changes in Commodity Prices of Cement Aggregates, Cement, and Concrete**

**EXECUTIVE SUMMARY**

My internship this summer will be in the sourcing department for Union Pacific Railroad. I will most likely be assigned to be a sourcing analyst to purchase commodities or other materials that Union Pacific (UP) purchases to build track or run their operations. I reached out to a BYU MBA alum who works in the same department to see if I could do my final VBA project on something that could either benefit me or the department in purchasing materials. My friend mentioned that he and his department are continuously opening the Bureau of Labor Statistics (BLS) website and pulling the Purchase Price Index (PPI) for various commodities to analyze and anticipate their price changes. Thus, was spawned my idea and you have this project before you; VBA code that automatically does exactly what my friend at UP does by hand every month.

The three commodities I am searching for in this project are sand/gravel aggregates for cement, cement itself, and concrete. All of which go into the foundation of UP’s train tracks. At the push of a button my code accesses the PPI database for each commodity, one by one, on the BLS site and loads them onto three tabs in a prepared worksheet. Two input boxes then pop up asking for the month and year of the price one would like to see from the index and loads that price and the price from exactly a year prior onto a table that calculates the percentage change in price from year to year. At the same time, the projected price for the subsequent month of that which was selected, along with the projected price of the commodity exactly a year from then is put into another table below to calculate the projected change in price.

**VBA CODE STEPS**

The following are the steps that are taken in the code along with descriptions and photos illustrating what is being done:

**Step 1 – Begin Data Search at the press of a button**

**Begin Data Search.tiff**

The above image shows what the table looks like when the prices for December 2007 were searched. The button named “Begin Data Search” to the right of the table initiates the code that searches for the new commodity prices. Unfortunately the PPI database goes only back to 2000 and the most recent update shows the index price of October 2009.

**Step 2 – Open an Internet Explorer box to access the BLS website**

**BLS Screen 1.tiff**

This is the new Internet Explorer box that opens to the page where the commodity PPI search begins. Pulling the data off the PPI was much more difficult to do than previously anticipated. The website does not have a complete URL which means one can’t simply write a macro to do a web search for the correct URL and pull the data. Instead, each step that is needed to access the PPI must be written into the code, made even more difficult by the fact that none of the forms have any names, only numbers. These numbers were found by counting the order in which they appeared to find the right form to which the code needed to refer. A tedious process.

Once this page has opened the code selects the form “Not Seasonally Adjusted” and clicks the “Next Form” button.

\*\*DISCLAIMER: As the forms are not named the entire code could potentially be made void at any time if the BLS site programmer changes the order in which the forms appear. This actually happened to me while I was working on the project. One form number that worked one day did not work the next and the code was useless.\*\*

**Step 3 – Select the commodity group desired**

**BLS Screen 2.tiff**

Here is a picture of the list box containing all of the different commodity groups. The code then selects group #13 (Nonmetallic mineral products) which is the group of all three commodities being searched. The following picture shows commodity group #13 being selected. Once the commodity group is selected the code clicks on “Next form”.

BLS Screen 2.1.tiff

**Step 4 – Select commodity type being requested**

BLS Screen 3.tiff

Once Screen 3 of 4 pops up the code then selects the commodity type being requested. Each commodity is searched for through the BLS site one at a time through a loop in the code. The commodity types selected are #21, #22, and #3 for aggregates, cement and concrete, respectively. The following image shows #21 – Sand/gravel aggregates being selected. Once the desired commodity has been selected the button “Next form” is clicked.

BLS Screen 3.1.tiff

**Step 5 – Request retrieval of the data**

The next step is to cause the site to retrieve the PPI for the commodity selected. This is simply done by having the code click on “Next form” on Screen 4 of 4.

BLS Screen 4.tiff

**Step 6 – Save page data to an HTML document**

Once the PPI database has been accessed it is still not possible to simply scrape the data on the page and place it on excel. Everything will transfer over but the table that you actually need. Below is an image of the PPI database as seen on the BLS website.

BLS Screen 5 - Database.tiff

When the above data is sent to an HTML document the page data is now seen in its unmodified form as shown below.

Source.tiff

It is in this form that the data can now be imported to Excel.

**Step 7 – Import PPI database to an Excel worksheet**

The code then imports the table from the HTML document into the worksheet “localQuery” and formats the table into a neat and tidy form as shown below

Sheet 1.tiff

As you can see in the image above, the PPI table for each of the three commodities, in turn, is then sent to pre-prepared worksheets. Once all three tables are loaded into all three specified worksheets an input box appears on the original, starting place worksheet, “PPI Increases”.

**Step 8 – Request month and year of the desired price**

Below is an image of the input box that appears, ready to receive a request for the month of the price that is being sought.

Input Box 1.tiff

The first three digits of the desired month can then by entered into the input box. If cancel is selected the user can start the entire process over again by clicking the button “Begin Data Search”. This is a bit of a pain, I know, but it is the best code that I can give at my current level of expertise.

Once the “OK” button is pressed either by mouse or by the “enter” or “return” key a second input box appears as shown below.

Input Box 2.tiff

This box allows the user to enter the year of the price desired. Once the year has been entered and the “OK” button has been clicked the price data for the year and month desired will be pulled from the PPI tables and put into the calculation tables. In this case that I am illustrating through images the prices for September 2008 will be pulled and compared to commodity price from September 2007 to get the price increase or decrease from year to year. Also, simultaneously, the price for October 2008 will be pulled as well as the projected, or known, data from October 2009, which is the last known price available to this date on the PPI. These prices are then used to calculate the percentage of the price change for the next year. Below is an image of the completed table.

Worksheet Final 2.tiff

What used to be a tedious, time-consuming process for a sourcing analyst to complete at Union Pacific is now a 3 step, automated process through this code. Not only will this code allow the user to pull the most recent commodity price data available on the BLS but it will allow the user to look at any price historically, back to the year 2000.

I learned a ton from putting this code together. In fact it stretched me, surprisingly enough to the more experienced programmer, beyond what I thought was capable and I’m extremely proud of it. Beyond the difficult of dealing with incomplete URL’s as discussed above, the main lesson I learned were the following tasks that I completed for the first time, unaided:

* How to perform a loop that repeated the same task for different variables
  + This entailed defining multiple variables and making sure the code that called the variables was able to go through each one seamlessly.
* How to write conditional statements that will look at numerous scenarios and pull out the right result when the condition is met.
* Building input boxes that feed information into the code and then gets written back into the spreadsheet

My hope is that you enjoyed seeing how my code works and that maybe even some, stressed-out sourcing analyst somewhere may find it useful. Thank you.