

## EXECUTIVE SUMMARY

When seeking an appropriate project for this final assignment, I solicited my past professors from last semester asking if there was something that they wish was automated in their daily work. Every professor that I emailed replied to me with a potential project that would help them. Unfortunately, I was only able to select one for one of the MBA finance professors, Grant McQueen<sup>1</sup>

Professor McQueen has been tracking the economic recovery of the US after the 2009 financial meltdown. He makes presentations on the data including, US unemployment rates, regularly using a PowerPoint presentation. Each time he presents his findings he has to pull data from the Federal Reserve Economic Data (FRED) website into excel, create several charts and put these charts into his PowerPoint slide deck. This process takes about 2 hours. Professor McQueen asked me to automate this process. I was successful in creating a VBA tool that reduces the processing time down to 60 seconds.

My Tools automates the process using 4 sub procedures:

- 1) getData() – Pulls data from the FRED website using the Professor Allen's agent module and inputs the data into an excel worksheet
- 2) user() – the checks who the user is and which API Key to use.
- 3) MakeCharts() – Creates 4 charts of unemployment data
- 4) ChartsAndTitlesToPresentation() – Exports the charts into an existing PowerPoint presentation

This project has increased my comfort level with the topics covered in class, but it has also exposed me to several other topics not covered in class specifically:

- 1) Application Programming interface (API) keys
- 2) Manipulating other Microsoft Office applications other than excel using VBA

<sup>1</sup> I was able to leverage the interest from all of the professors to land a paid VBA consulting project with the director of the MBA program, John Bingham.

## IMPLEMENTATION DOCUMENTATION

To accomplish the task of downloading data, creating charts and exporting those charts to PowerPoint, I created 3 buttons contained in a customized tab in the excel ribbon. I also created an additional sub-procedure to clear the data.

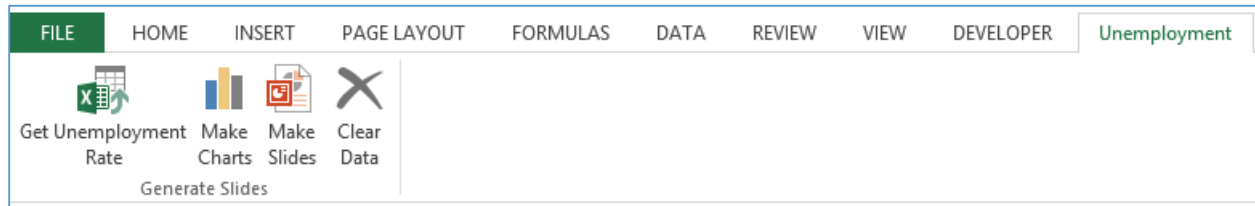


Figure 1 - Customized Tab of Excel ribbon containing buttons for the macros

In order to access the FRED website in an automated fashion, the user will be required to obtain an Application Programmer Interface (API) key (this will only need to be done once).

### Obtain API Key

- 1) Create a research.stlouisfed.org user account –  
<https://research.stlouisfed.org/useraccount/register/step1>
  - a. You will be asked to enter basic information

Register User Account

1. Enter Account Info > 2. Check Your Email > 3. Confirm Account

First Name:

Last Name:

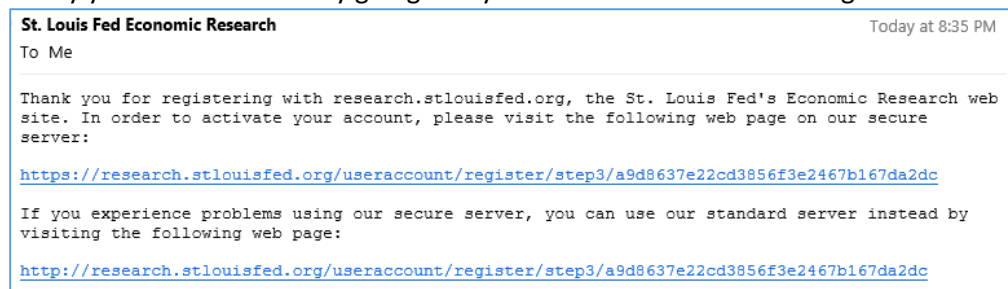
Email Address\*:

Enter a New Password\*:  
(Passwords must be between 4 and 16 characters.)

Retype Password\*:

By submitting your registration information, you indicate that you understand the St. Louis Fed Privacy Policy.

- b. verify your email address by going into your email account and clicking onto a link.



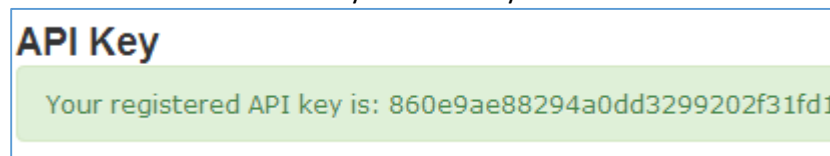
- 2) Request an API Key
  - a. <https://research.stlouisfed.org/useraccount/apikeys/>
  - b. Click on "Create API key"



- c. Describe the application or program you intend to write
- d. Read the Terms of use, privacy policy, and legal notices and Disclaimers and check that you have
- e. Click on Request API Key



- f. You should receive an API key immediately



## Get Unemployment Rates



This button calls the getData sub-procedure. This sub-procedure does the following:

- 1) Calls the User sub-procedure (This will be described in the next section)
- 2) Creates an array of the state codes used in the FRED URL and state names associated with those codes from a list found in the "Codes" worksheet.
- 3) Within a for loop it creates and uses a web agent to open the FRED website using the API key provided and the state codes within the state codes array. The figure below shows the XML of the webpage which is not intended to be visual.

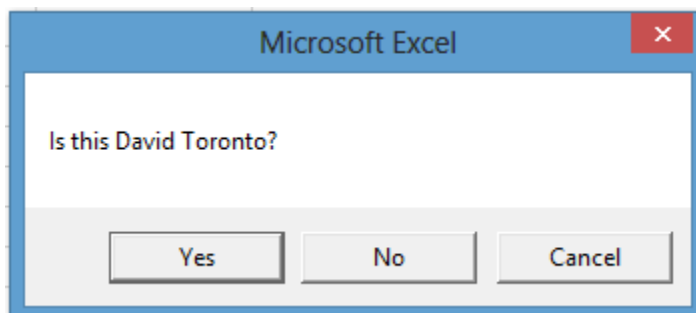
```
<?xml version="1.0" encoding="UTF-8"?>
- <observations limit="100000" offset="0" count="795" sort_order="asc" order_by="observation_date" file_type="xml" output_type="1" units="lin" observation_end="9999-12-31"
  observation_start="1776-07-04" realtime_end="2014-04-15" realtime_start="2014-04-15">
  <observation realtime_end="2014-04-15" realtime_start="2014-04-15" value="3.4" date="1948-01-01"/>
  <observation realtime_end="2014-04-15" realtime_start="2014-04-15" value="3.8" date="1948-02-01"/>
  <observation realtime_end="2014-04-15" realtime_start="2014-04-15" value="4.0" date="1948-03-01"/>
  <observation realtime_end="2014-04-15" realtime_start="2014-04-15" value="3.9" date="1948-04-01"/>
  <observation realtime_end="2014-04-15" realtime_start="2014-04-15" value="3.5" date="1948-05-01"/>
  <observation realtime_end="2014-04-15" realtime_start="2014-04-15" value="3.6" date="1948-06-01"/>
  <observation realtime_end="2014-04-15" realtime_start="2014-04-15" value="3.6" date="1948-07-01"/>
  <observation realtime_end="2014-04-15" realtime_start="2014-04-15" value="3.9" date="1948-08-01"/>
  <observation realtime_end="2014-04-15" realtime_start="2014-04-15" value="3.8" date="1948-09-01"/>
  <observation realtime_end="2014-04-15" realtime_start="2014-04-15" value="3.7" date="1948-10-01"/>
  <observation realtime_end="2014-04-15" realtime_start="2014-04-15" value="3.8" date="1948-11-01"/>
```

- 4) Since the only data of interest is the most recent, the Moveto function is used in a Do while loop to find the last instance of "Date="
- 5) An element in the stateDate array is set and an element in the stateRate array is set.
- 6) After the for statement loops through all of the state codes in "Codes" worksheet until the stateCodes and StateNames array is filled
- 7) The StateCodes and StateNames arrays are then imported into "Codes" worksheet and the numbers are properly formatted.

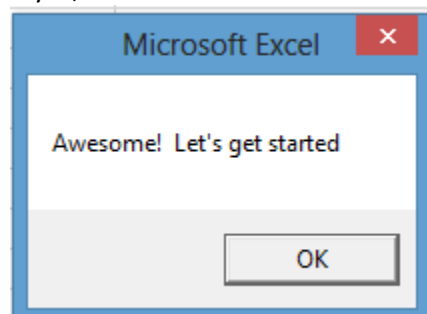
### User Sub-procedure

Now that the user has an API key, they will be guided on how to enter the API code using a series of message and input boxes. If the User has already used the form before and has already entered their APIkey, they can select it from a dropdown list. This is done with the user sub-procedure. This sub-procedure does the following:

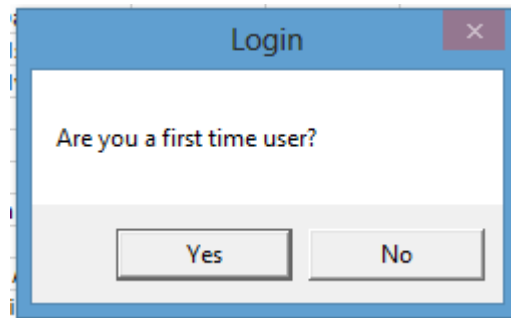
1. Asks if the user is the current default user.



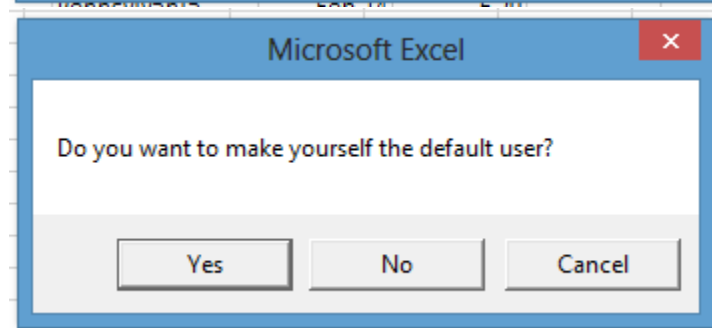
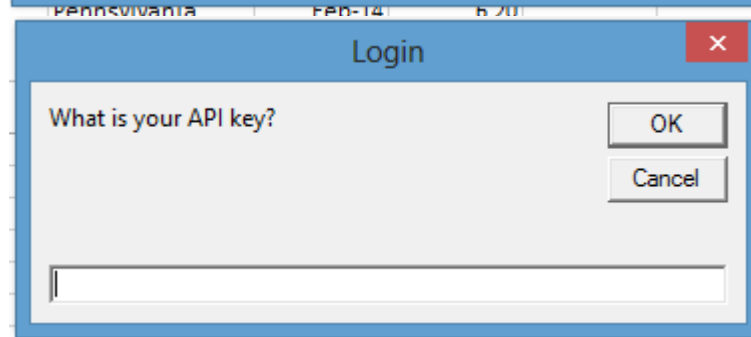
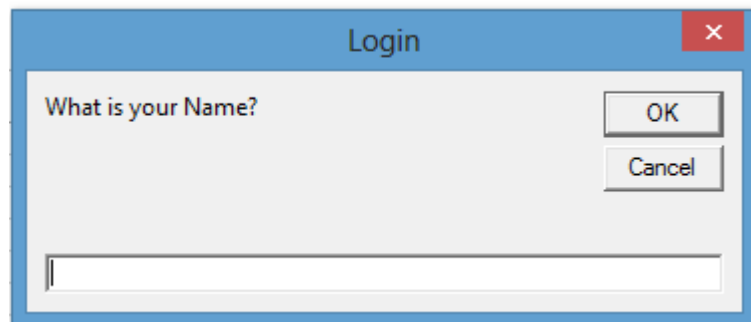
- a. If yes, then the current Default User APIKey is used to query the FRED Website



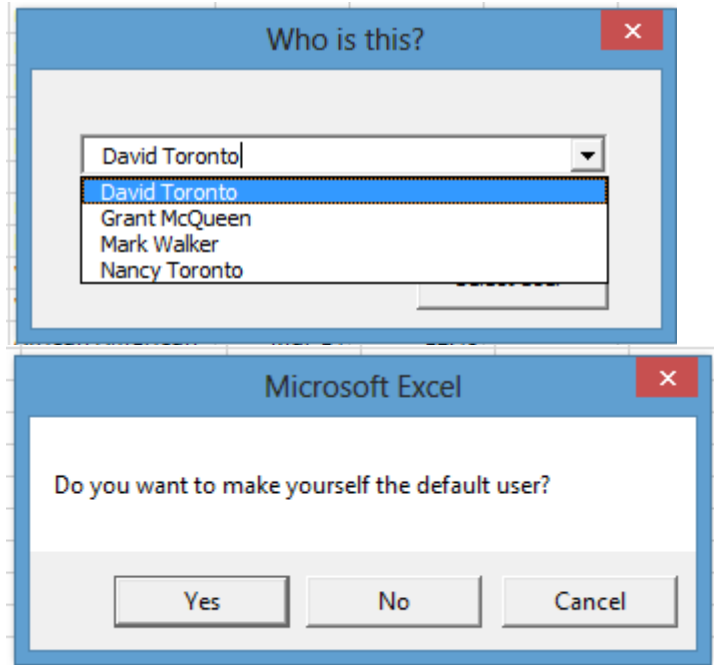
b. If No



i. If Yes, the user will be prompted to enter their name and API Key and asked if they would like to be made the default user.



ii. If No, the user will be ask to select who this is from a drop down menu of previous users and the user will be asked if they would like to be made the default user.



c. If the user hits cancel, both the user sub-procedure and the getData sub-procedure will be exited.

### Make Charts



This button calls the makeCharts sub-procedure. This sub-procedure does the following:

- 1) Clears all the charts in the "Codes" worksheet to ensure prevent a buildup of old charts.
- 2) Loops through all of the unemployment rates in column D to find which rows contain the unemployment rates of interest in the US specifically:
  - a. USA
  - b. Utah
  - c. State with the highest unemployment in the US
  - d. State with the lowest unemployment in the US
  - e. State with the median unemployment in the US
- 3) Loops through Column D to find which rows contain the unemployment rates for different demographics specifically:
  - a. Men
  - b. Women
  - c. Caucasian
  - d. African American
  - e. Hispanic or Latino
  - f. Teens (16-19)

Note: the row with the men data and teen data are used to bound the demographic codes. In other words, it is assumed that the men data will be the first demographic code and the teen data will be the last demographic data.

- 4) Loops through Column D to find which rows contain the unemployment rates for the emerging economies specifically:
  - a. Brazil
  - b. Russia
  - c. China
- 5) Loops through Column D to find which rows contain the unemployment rates for the European economies specifically:
  - a. EuroZone
  - b. Portugal
  - c. Italy
  - d. Ireland
  - e. Germany
  - f. Spain
  - g. Greece
  - h. Great Britain

Note: the row with the Brazil data and EuroZone data are used to bound the European codes. In other words, it is assumed that the Brazil data will be the first emerging economies code, the European codes will immediately follow the Emerging economies data, and the EuroZone data will be the first of the European economies data.

- 6) Copies and formats the data of interest into Columns F through I on the “Codes” worksheet to make it easier to create charts from the data.

State Chart Data			
Series ID	Name	Date	Rate
UNRATE	USA	Mar-14	6.70
UTUR	Utah	Feb-14	3.90
NDUR	North Dakota	Feb-14	2.60
RIUR	Rhode Island	Feb-14	9.00
PAUR	Pennsylvania	Feb-14	6.20
UNRATE	USA	Mar-14	6.70
LNS14000001	Men	Mar-14	6.80
LNS14000002	Women	Mar-14	6.60
LNS14000003	White	Mar-14	5.80
LNS14000006	African American	Mar-14	12.40
LNS14000009	Hispanic or Latino	Mar-14	7.90
LN04027660	High School Only	Mar-14	6.80
LNS14000012	16 to 19 years years c	Mar-14	20.90
UNRATE	USA	Mar-14	6.70
LRUNTTTBRM156S	Brazil	Dec-13	5.13
LMUNRRITCNQ156S	China	Jul-11	4.10
LMUNRRITRUM156S	Russia	Dec-13	1.21
UNRATE	USA	Mar-14	6.70
LRHUTTTTEZM156S	Euro Area	Dec-13	12.00
PRTURHARMMDSEI	Portugal	Apr-12	15.20
IRLURHARMMDSEI	Ireland	Apr-12	14.20
LRHUTTTTITM156S	Italy	Dec-13	12.70
LRHUTTTTDEM156S	Germany	Dec-13	5.10
ESPURHARMMDSEI	Spain	Apr-12	24.30
GRCURHARMMDSEI	Greece	Feb-12	21.70
GBRURHARMMDSEI	Great Britain	Feb-12	8.10

- 7) Creates 4 charts by
  - a. Assigning 4 chart objects as column clustered type charts

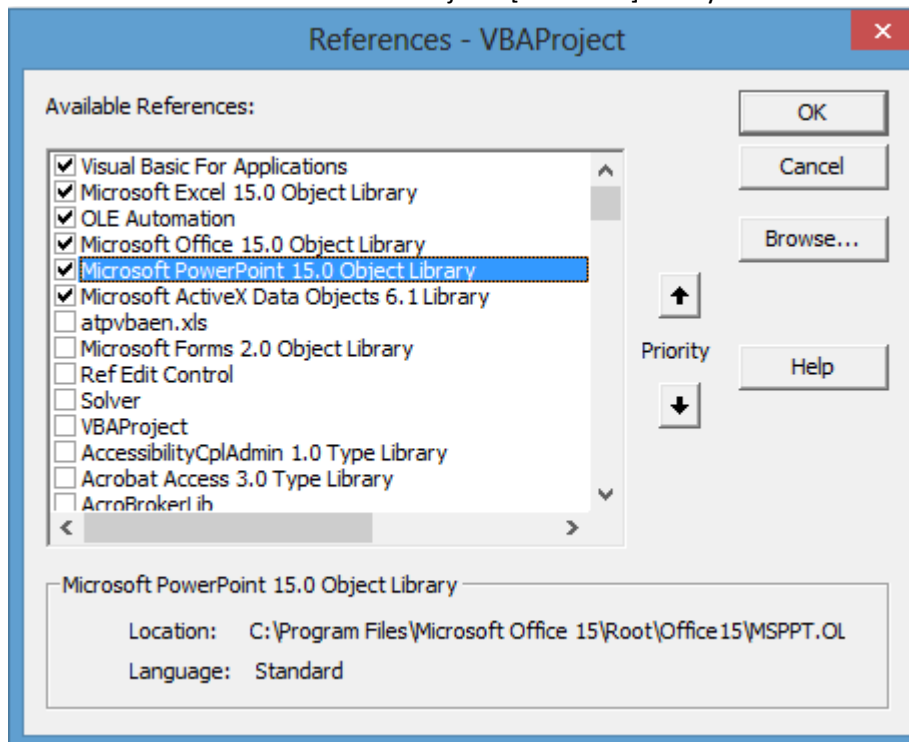
- b. Sets the source range for each
- c. Positions each chart to the left of the data and stacks the charts one on top of the other
- d. Gives a title and an axis title to each

## Make Slides



In order for this code to work a PowerPoint reference must be added to excel VBA. This can be done by:

- 1) Going to Tools → References
- 2) Select the Microsoft PowerPoint [Version #] Object Library
- 3) Select the Microsoft ActiveX Data Objects [Version #] Library



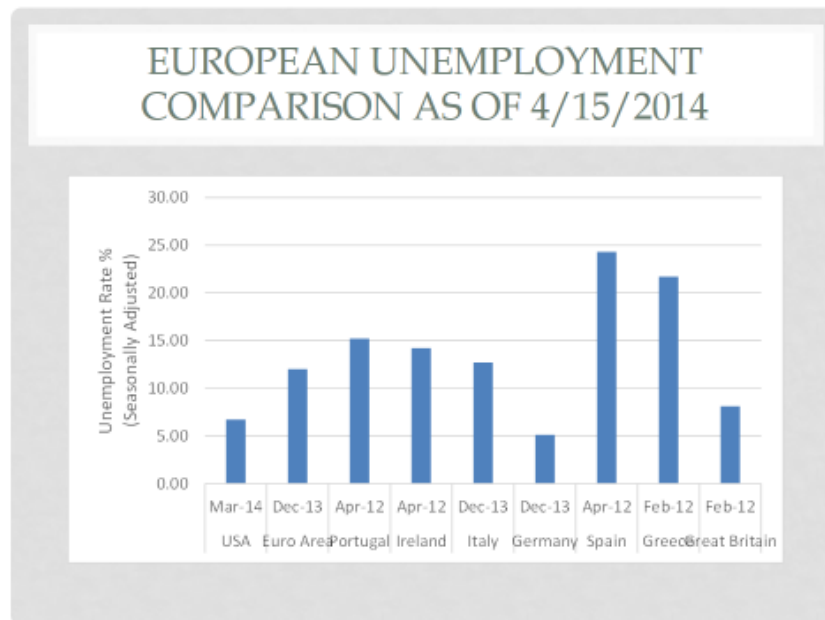
In addition, in order for this button to work the Existing PowerPoint presentation that is desired to add the slides to must be open.

The Make Slides button calls the ChartsAndTitlesToPresentation sub-procedure which does the following:

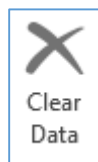
- 1) Loops through each of the 4 charts in the activesheet of the workbook and
  - a. Extracts and saves the chart title as a string variable and removes the title from the chart
  - b. Copies the chart as a picture



- c. Adds the picture of the chart to the end of the powerpoint presentation
- d. Aligns and sizes the chart to properly fit onto slide
- e. Creates a label of the slide as the name of the chart



#### Clear Data



This button calls the clearCharts sub-procedure. This sub-procedure does the following:

- 1) Deletes the existing charts on the sheet
- 2) Deletes the chart data in columns F through I

#### DISCUSSION OF LEARNING AND DIFFICULTIES

Other than gaining experience in using the concepts discussed in class, I also learned two new concepts not covered in class. API Keys and manipulating Power Point from VBA Excel:

API Keys are necessary to create tools that automatically pull data from websites. My API key that I requested from their website allows FRED to track my usage of their website. If I begin to pull data excessively, the FRED has the ability to revoke my key. This helps them in their server management.

Manipulating PowerPoint in VBA is not as difficult as I thought it was going to be. The key to manipulating an office application is to set the appropriate reference in VBA. Once this is done, excel VBA now has access to PowerPoint VBA functions.

The one thing I did not get to figuring out a way to open powerpoint within VBA. I am sure it is relatively simple, but I ran out of time and Professor McQueen was okay with opening ppt manually. It would have been a nice-to-have. In my next update, I will add that feature.

### **ASSISTANCE**

The only assistance that I received with this project was from Professor Allen regarding the how to use API keys. He also sent me a link to a website that contained code that allowed for excel charts to be copy as pictures into PowerPoint. My only other source of assistance came from Google.