

Police Analyst

A new kind of law enforcement analysis

Using the Google Geocoding API in Excel

Posted on [May 24, 2012](#) by [josephglover](#)

Like all crime analysts I work with street addresses in Excel every day and I frequently want to determine the latitude and longitude coordinates for an address. Turning addresses into coordinates is known as [Geocoding](#) and modern web services make it easy to do without specialized (and expensive) tools.

For example, using [Google's geocoding API](#) it is possible to visit a URL that contains an address and have Google return an XML file that contains the latitude and longitude of the location. Say that I wanted to find the coordinates for "1 Yonge Street in Toronto, Ontario, Canada" I could put the following URL into my browser and get the XML results:

<http://maps.googleapis.com/maps/api/geocode/xml?address=1%20yonge%20street%20toronto%20ontario%20canada&sensor=false>

Click the link above and, if you have a modern browser, you should see a bunch of XML text that contains, buried within a bunch of other address data, the latitude and longitude of 1 Yonge Street.

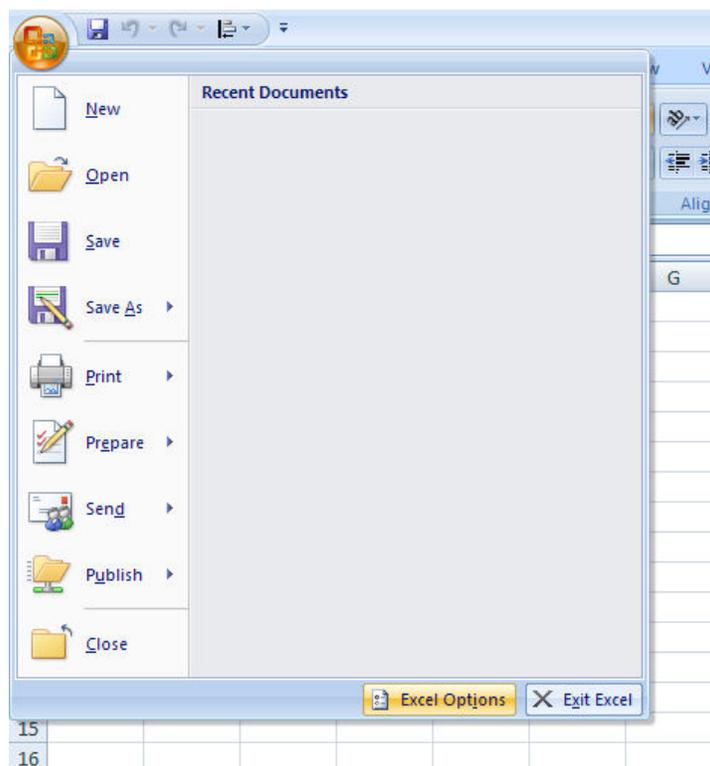
While this is impressive in a "we live in the future" kind of way, it is still a labour-intensive, manual process. To make this technique more useful I have written an Excel function that performs the work behind the scenes and returns the coordinates in Excel.

The rest of this post will demonstrate how to create the function.

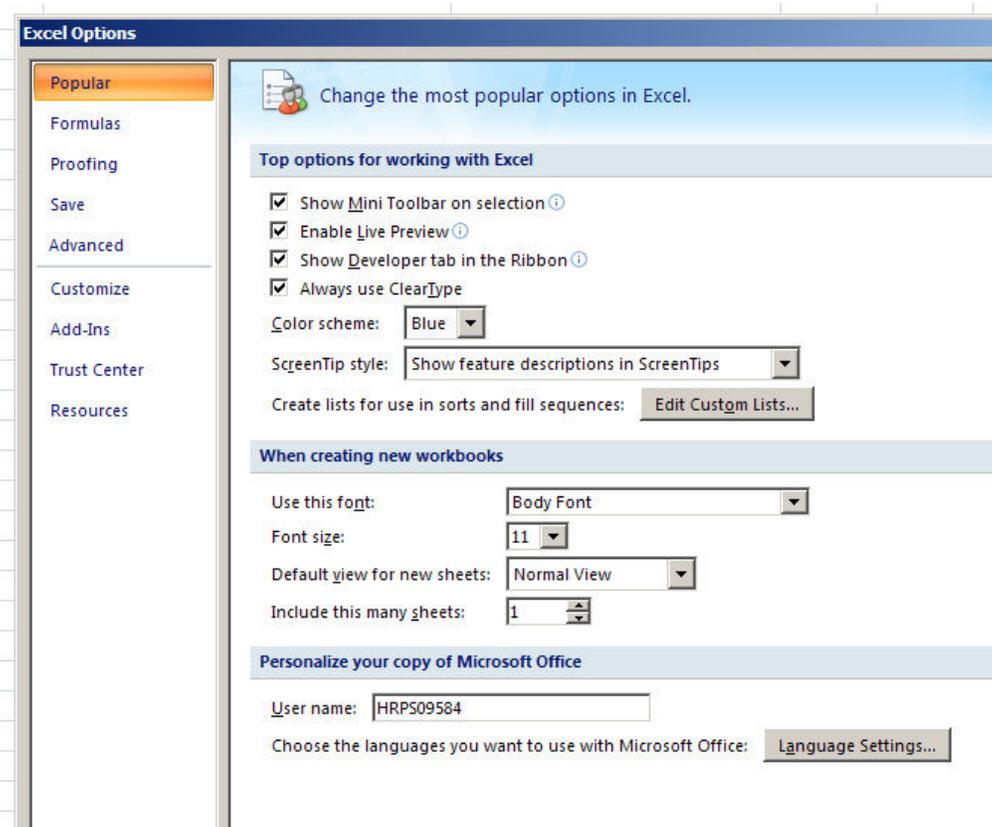
How to create an Excel function

These instructions are particular to Excel 2007 because that's the version I use. They should also work for Excel 2010 but you may need to poke around to find the corresponding buttons.

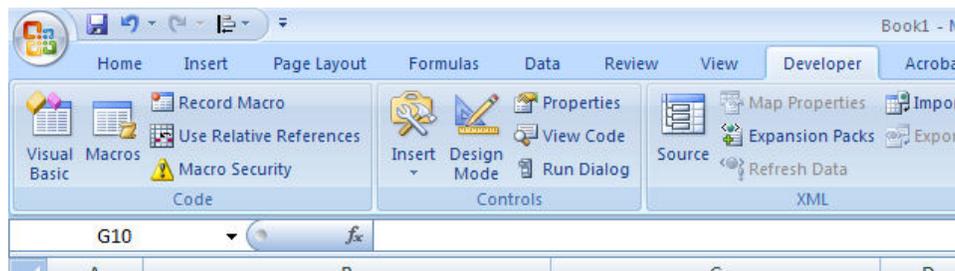
The very first thing that we need to do is to turn on the "Developer" tab in the main ribbon. To do that, click the the Office button in the top left corner (the same one that is used to access save or print).



Click the “Excel Options” button at the bottom. This will open a new dialog window. Click the “Popular” menu option (it should be the default menu option) and then check the third checkbox under “Top options for working with Excel”, the one that says “Show Developer tab in Ribbon”.



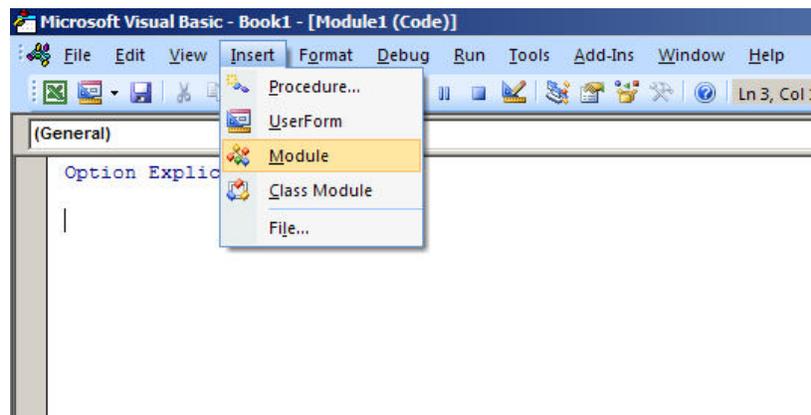
Now the developer tab will show in the ribbon. It looks like this:



This tab will now be available any time you open Excel. Close Excel.

Open a new, empty Excel spreadsheet and verify that the “Developer” tab is still there. Click the tab and then click the “Visual Basic” button on the far left. This will open up the Visual Basic Editor.

Visual Basic will open with a main content area that looks something like this.



As in the image, we want to add a new module file to store our new function. So click “Insert” and select “Module”. This will create the typeable area you see above that says “Option Explicit” at the top. (Option Explicit is just some default code, don’t worry about it for now).

Copy the code below into the typeable area. Put it below the “Option Explicit” line.

```
Function GoogleGeocode(address As String) As String
    Dim strAddress As String
    Dim strQuery As String
    Dim strLatitude As String
    Dim strLongitude As String

    strAddress = URLEncode(address)

    'Assemble the query string
    strQuery = "http://maps.googleapis.com/maps/api/geocode/xml?"
    strQuery = strQuery & "address=" & strAddress
    strQuery = strQuery & "&sensor=false"

    'define XML and HTTP components
    Dim googleResult As New MSXML2.DOMDocument
    Dim googleService As New MSXML2.XMLHTTP
    Dim oNodes As MSXML2.IXMLDOMNodeList
    Dim oNode As MSXML2.IXMLDOMNode

    'create HTTP request to query URL - make sure to have
    'that last "False" there for synchronous operation

    googleService.Open "GET", strQuery, False
    googleService.send
    googleResult.LoadXML (googleService.responseText)

    Set oNodes = googleResult.getElementsByTagName("geometry")

    If oNodes.Length = 1 Then
```

```

For Each oNode In oNodes
    strLatitude = oNode.ChildNodes(0).ChildNodes(0).Text
    strLongitude = oNode.ChildNodes(0).ChildNodes(1).Text
    GoogleGeocode = strLatitude & "," & strLongitude
Next oNode
Else
    GoogleGeocode = "Not Found (try again, you may have done too many too fast)"
End If
End Function

Public Function URLEncode(StringVal As String, Optional SpaceAsPlus As Boolean = False) As String
    Dim StringLen As Long: StringLen = Len(StringVal)

    If StringLen>0 Then
        ReDim result(StringLen) As String
        Dim i As Long, CharCode As Integer
        Dim Char As String, Space As String

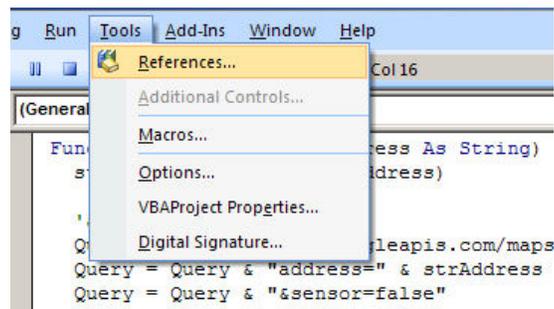
        If SpaceAsPlus Then Space = "+" Else Space = "%20"

        For i = 1 To StringLen
            Char = Mid$(StringVal, i, 1)
            CharCode = Asc(Char)

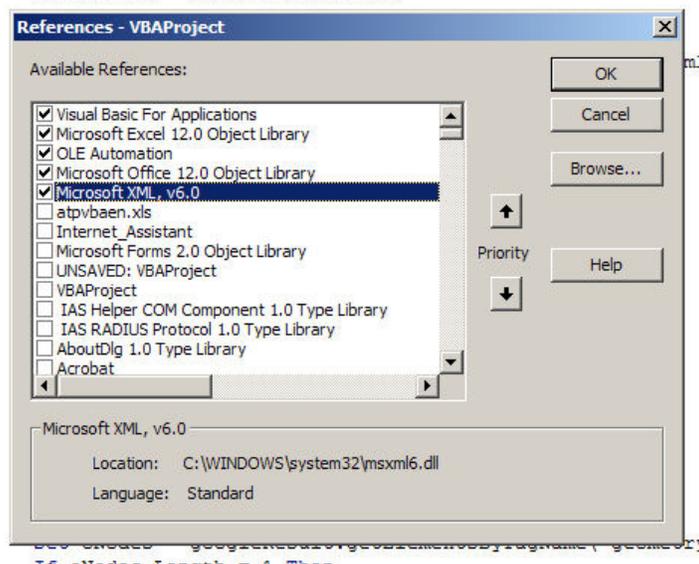
            Select Case CharCode
                Case 97 To 122, 65 To 90, 48 To 57, 45, 46, 95, 126
                    result(i) = Char
                Case 32
                    result(i) = Space
                Case 0 To 15
                    result(i) = "%0" & Hex(CharCode)
                Case Else
                    result(i) = "%" & Hex(CharCode)
            End Select
        Next i
        URLEncode = Join(result, "")
    End If
End Function

```

The code makes use of Microsoft’s XML library and you’ll need to add a reference to it. To do that click the “Tools” menu option and select “References”.



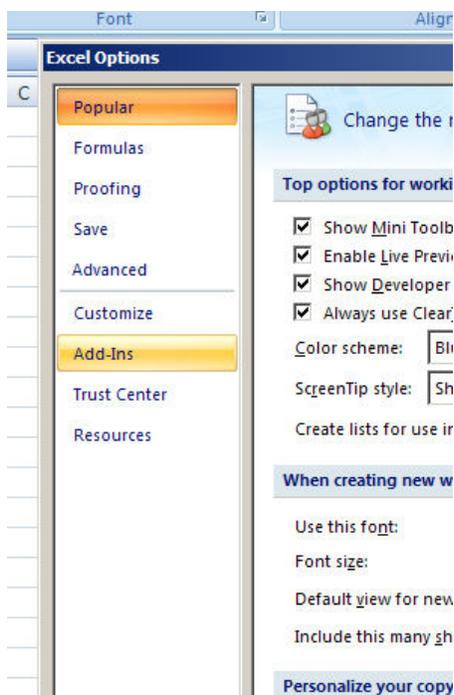
A list of checkboxes will pop up. You need to locate “Microsoft XML v6.0” from the list and check it. Thankfully, the libraries are in alphabetical order. Mine looks like this, of course, I’ve already checked the reference so the library is at the top of the list. When you’re done, click “OK”.



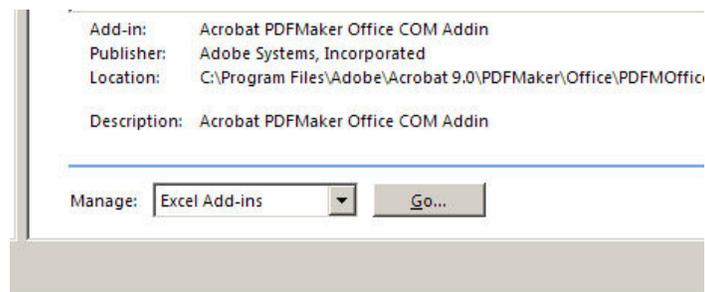
Finally, click the little Floppy Disc icon in the Visual Basic Editor to save. In the “Save as Type:” drop down select “Excel Add-in (*.xlam)” and name the file PoliceAnalyst.xlam.

An *.xlam is a type of Excel file used for saving code. Once you’ve created it you can reference it in Excel and the functions within will be available to every Excel document that you open.

Close all your Excel files and open a fresh one. As above, click the “Excel Options” button under the main Office button in the top left corner. This time, however, click the “Add-Ins” menu option.



And at the bottom of the options area click “Go” beside the “Manage: Excel Add-Ins” drop down.

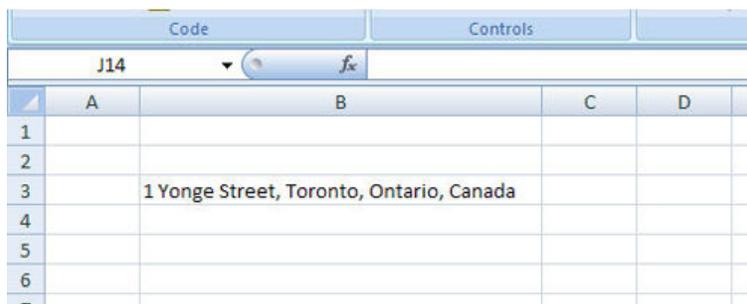


This will open a window that will list all your available add-ins along with checkboxes. If you created the PoliceAnalyst.xlam add-in as described above (and saved it in the default location offered by Excel) it should be right there. Click the checkbox and click “OK”. Otherwise, click “Browse” and browse to the location where you saved the file.

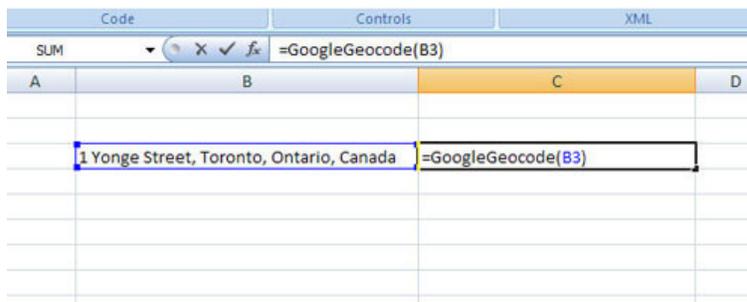
Phew. We just created an add-in, filled it with code, and installed it in Excel.

How to use the function

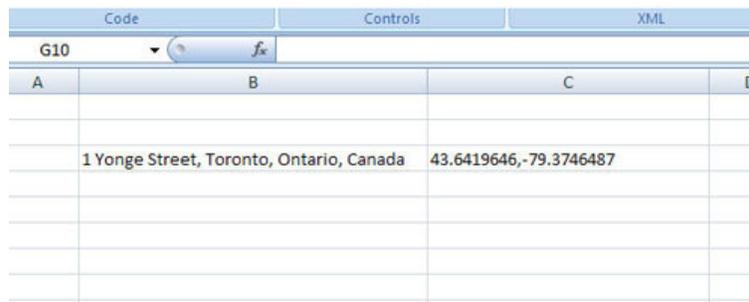
Ok, this is the fun part. Open up a new Excel spreadsheet and in a cell type in an address you’d like to geocode. Here is the Yonge Street example from above.



In the cell beside it, reference the new function in the same manner as you’d reference other Excel functions, passing in the address cell.



And hit return, give it a second, and you should, fingers crossed, have your latitude and longitude returned. Both values occupy a single cell but they are comma separated so it’s easy to split them off for later manipulation.



| Code | Controls | XML | |
|------|--|------------------------|---|
| G10 | | | |
| A | B | C | D |
| | | | |
| | 1 Yonge Street, Toronto, Ontario, Canada | 43.6419646,-79.3746487 | |
| | | | |
| | | | |
| | | | |
| | | | |

Now, there are some caveats. You can fill this function down or across just like other Excel functions but keep in mind that the function is calling out to the Internet every time it is executed and that takes a toll in terms of processing power and performance. The best practice is to run the function on your addresses and then copy and “paste values” to save just the coordinates in order to prevent the function from running every time the spreadsheet is opened or updated.

There is another reason to do this: Google is nice enough to geocode for free but they won’t put up with thousands (or even hundreds) of attempts per minute from the same computer. After a certain number (about 10 per second or so) they will stop providing coordinates for a brief period. If you have, say, 50 addresses and you fill the function down you’ll notice that after 10 or so rows you’ll start to get back a “Not Found” message. It is possible that Google can’t find your address but more likely Google cut you off because you asked for too many addresses too quickly. The solution is to wait a few seconds and try the “Not Found” ones again; if they still come back “Not Found” then Google really can’t locate your address but the chances are good that you’ll get your coordinates the second time around.

If you persistently get the “Not Found” message check to make sure that your address is as complete as it can be. Adding things like the province and the country can help Google narrow its search and provide a better answer.

I realize that this blog post was a little involved but I find the function very useful and I use it everyday. I hope you find it useful too.

This entry was posted in [Excel Tricks](#) and tagged [.xlam](#), [geocoding](#), [Google](#). Bookmark the [permalink](#).

Police Analyst

Proudly powered by WordPress.