# Cody Nolden

# Executive Summary

Blog Manager is a VBA application that allows posting to and making backups from an unlimited number of Blogger and Posterous blogs.

Many avid bloggers, including me, have difficulty maintaining several blogs across multiple platforms. I frequently need to post the same content in multiple places (once in my public blog, once in my personal, private blog, etc). I also want to back up my blogs so that I can have them for offline use, or even store them in another format.

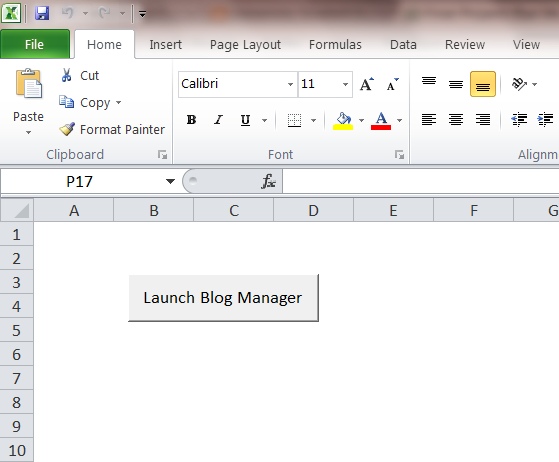
Blog Manager accomplishes these tasks. Blog manager allows you to:

* Store account details and usernames for an unlimited number of Blogger or Posterous blogs
* Compose a new blog post
* Send a blog post to any or all of your blogs
* Back up all blog posts in an easy-to-read HTML format
* Back up all blog posts to a native XML or JSON format for use elsewhere

Blog Manager is useful because it unites two blogging platforms into one interface. Multiple other platforms such as Wordpress could be easily added to the program, as described in the Implementation Documentation below.

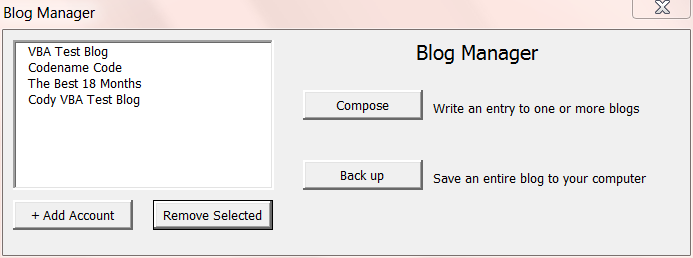
# Implementation Documentation

Blog Manager consists of three main workflows: adding accounts, posting to accounts, and backing up accounts. Blog Manager is easily launched when opening the Excel workbook:

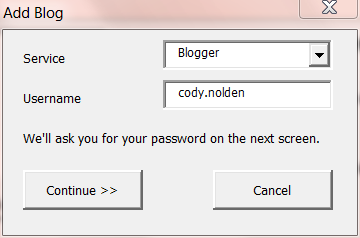


## Adding Accounts

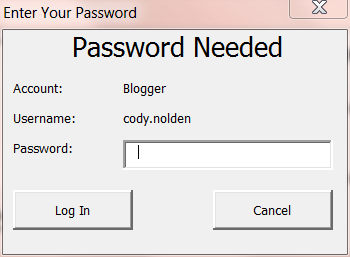
Accounts from Blogger or Posterous are supported. All authentication is performed via the Blogger and Posterous APIs, and no passwords or API tokens are stored in the Excel spreadsheet (for security).



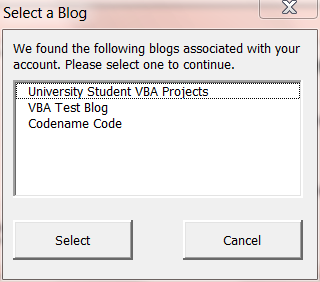
Adding an account is performed by clicking “Add Account.” This opens a window for you to enter basic account information:



Choices in the “Service” dialog include Blogger and Posterous. On the next screen you enter your password. This password is not stored in the Excel spreadsheet. Each time a transaction with a blog occurs, the user is prompted for the password again.

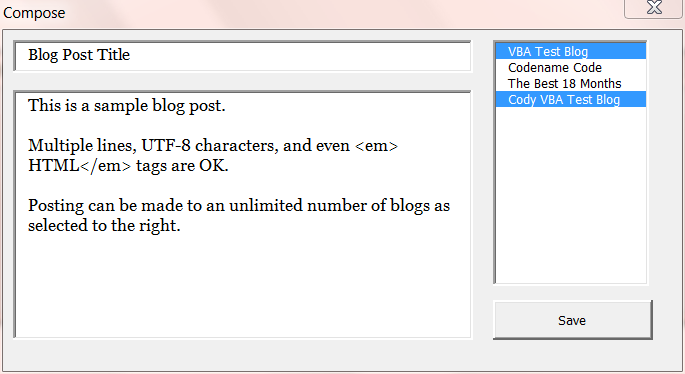


The application will then obtain a list of blogs associated with your account. Both Blogger and Posterous support multiple blogs per account. Only one blog can be selected at this time -- if you’d like to add multiple blogs, you can repeat the process again.

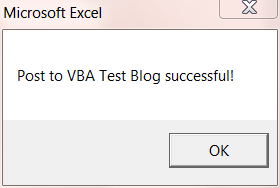


Once you’ve selected your blog, the application will store this information so you don’t have to enter it again.

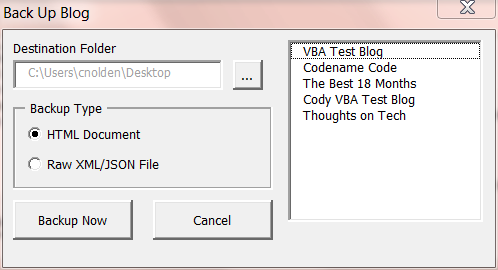
Composing new blog posts is easy. Just click “Compose” on the main menu and select which blogs you’d like to post to. You can post to any or all of the blogs.



Feedback is given for each blog post that completes successfully. If any of the posts fails to transmit, this window remains open so you can save your post elsewhere or diagnose your internet connection.

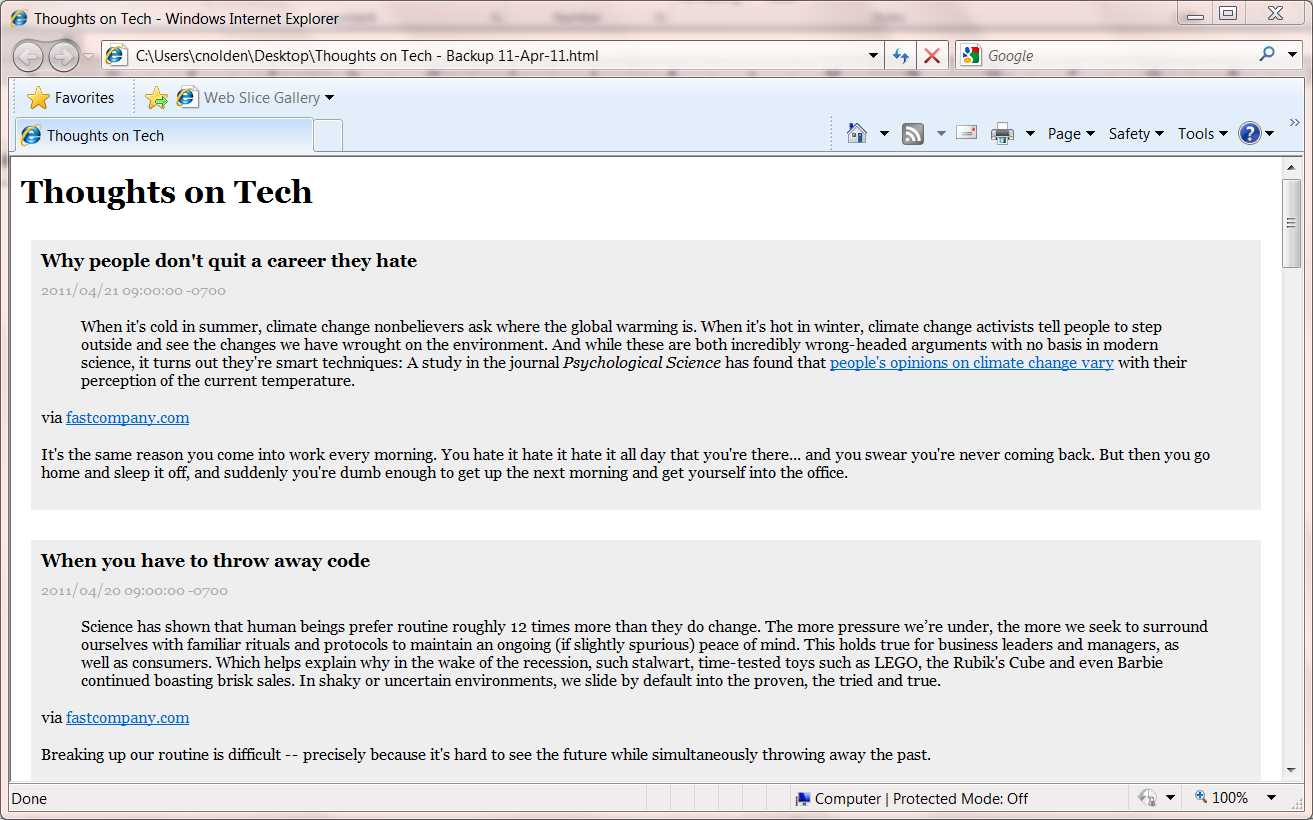


Backing up your blog is just as easy. Two formats are available for backup: a human-friendly HTML file and a native JSON or XML file (JSON for Posterous and XML for Blogger).

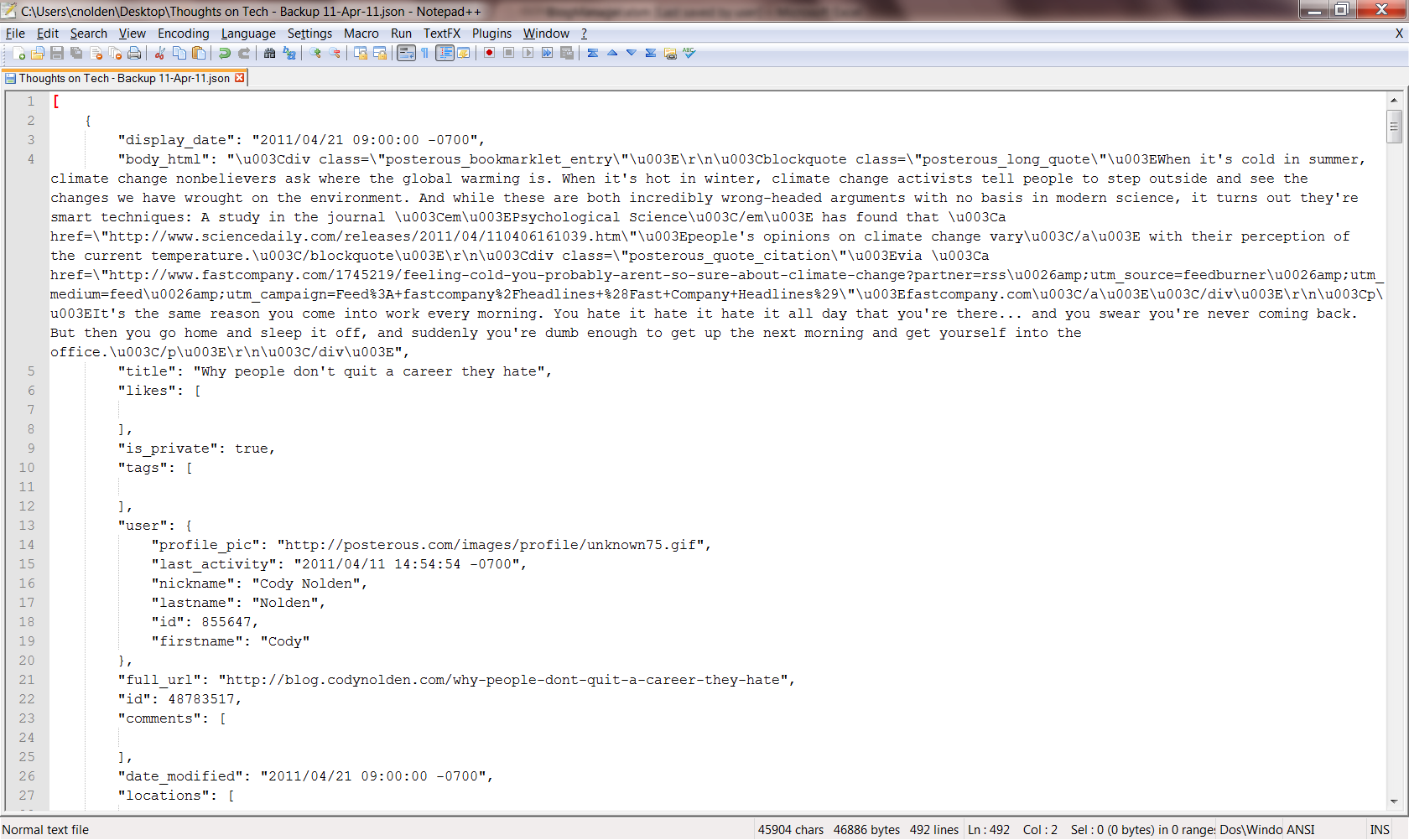


One blog may be backed up at a time. Clicking the “…” button reveals a folder picker for you to select the destination of the backup. Today’s date is automatically added at the end of the filename, so you can back up your blog as often as you would like.

HTML backups are opened automatically in Internet Explorer.



Raw backups are not as easy to read, but are great if you want to take the output and process it further in another program.



Overall, this is a very useful and expandable tool for managing your blogs on multiple platforms.

# Learnings and Difficulties

I learned a great deal about VBA with this project, namely:

* **The Importance of User-defined Classes**. Because this application is meant to support multiple blog platforms, I had to make it as extensible as possible. That’s why I made each platform its own user-defined class (cBloggerBlog and cPosterousBlog). By implementing the same methods in each class, I was able to customize the data layer logic without having to change my User Forms or Modules files.
* **Why You Should Isolate View Logic from Application Logic.** This application performs the same tasks in multiple places. Collecting a password, for example, happens over and over. I did not want to create a User Form for every time I wanted to collect a password. So I created a generic collectPasswordForm and passed data to it from another user form according to the circumstances. Separating the view logic in this way helped save a lot of code.
* **How to Process XML.** Blogger uses XML for its data library. I learned how to use the MSXML2 libraries to iterate through the nodes of a document and find the needed information. This was one of the most difficult parts of the project.
* **How to Modify a VB 6.0 Library to Work in VBA.** The Posterous API uses JSON and nothing else. VBA does not have native support for JSON, and there are no readily available add-ins (similar to MSXML2) that support JSON parsing. The only working parsers that I could find were written in VB 6.0 -- compatible with VBA, but still needing a few tweaks.
* **How to Implement Basic HTTP Authentication using the WinHTTP object.** Posterous requires a flavor of Basic HTTP Authentication that didn’t seem to work with the built-in WinHTTP authentication features. So I had to find a module that could do Base64 encoding to do my own version of Basic HTTP authentication. Once I had done this, adding an Authorization header to the WinHTTP object was a breeze.
* **How to Implement a Folder Selection Dialog.** The backup screen allows the user to select a folder via a convenient selection dialog. I learned how to harness the power of the shell to use this dialog.
* **How to Write a File.** Writing HTML files and text files is not as difficult as it sounds. I wrote two user-defined classes to handle these operations in a reusable manner.

My biggest difficulties with this project were 1) parsing XML and JSON responses from the Google and Posterous APIs, 2) implementing the authentication for the Google and Posterous APIs, and 3) coming up with an application design that is both flexible and easy to maintain. I feel that I did very well on all of these things.

Overall this project was a great experience for me, and I’m very confident that my VBA skills will be of use in the workforce.