Github Page on Using Dropbox as a means of sharing executable files with others… two way research for data sharing

About me?

My name is Megan DuPriest and I am an undergraduate research assistant in the computer science department of the University of North Carolina at Chapel Hill, working under the guidance of Dr. Gary Bishop. I am planning to graduate in May 2017 with a Bachelor of Science in Computer Science.

The idea

Use Dropbox as a means of A) delivering software to users B) gathering data from user

How EyeReader demonstrates this

The first step to this development was using cxfreeze in order to convert the python scripts of EyeReader into an executable file. You can get it from here [hyperlink]. This will create a dir folder that contains almost everything necessary to run the program. Note however, that if you program requires other python files, dlls, etc. in order to run, those must be added to the dir folder. And programs which explicitly refer to certain directories and locations in order to find these files must be written in way such that it works within the dir folder.

An easy way to work around this is to take advantage of os.getcwd() from the os module which will return the location of the file that called it. And from there usually you can navigate to find your other necessary files.

Another way to work around this can be built into the creation of the shortcut to this file. See ‘how to make a shortcut’

How to Make a Shortcut

There is no installation process involved with using this method of Dropbox… save for one thing. Your users may not be particularly tech savvy, and are not necessarily going to know how to locate the exe file in your dir in order to run the program; not to mention that’s just tedious.

Instead, I’ve found that its simple to create a small secondary executable program that I usually name mkshortcut. Its sole purpose is to utilize os and winshell to create a shortcut to your executable file. The benefits of this is that it appears right on your user’s desktop without any hassle; and there’s some customization to be had in how your user runs the program.

[Include the basic code for the EyeReader’s mkshortcut]

1. Being able to alter the working directory of your program can make things run much simpler for the user and also for you. Cxfreeze creates the executable file inside of a dir folder… which as mentioned before can bug up your code that ran at a different file level previously. But in this case, all you have to do is change the working directory of the executable to be wherever you desire (perhaps at the same level your original python script did!)
2. You can add an icon to the shortcut, and therefore have your logo on it when the program starts up. A small detail, but it certainly makes your program look a lot more professional.

But then the issue is… well, doesn’t the user then have to search the program files to find *this* executable that creates the shortcut?

No. The simplest solution I’ve found is just to create a bat file that’s sole purpose is to locate the mkshortcut executable and run it. That way, the bat file can be located higher up in the program, and the user doesn’t touch the actual program files.

[include the code for the bat file here]

And the fun thing about the mkshortcut is that even after the user has run the bat file and created the shortcut, even if they do it again accidentally, it doesn’t keep creating more and more new shortcuts. It just checks to see if the shortcut already exists (and if it does) and simply returns the same shortcut. Then if you really want, the user can right click the shortcut and pin it to the taskbar or wherever they like.