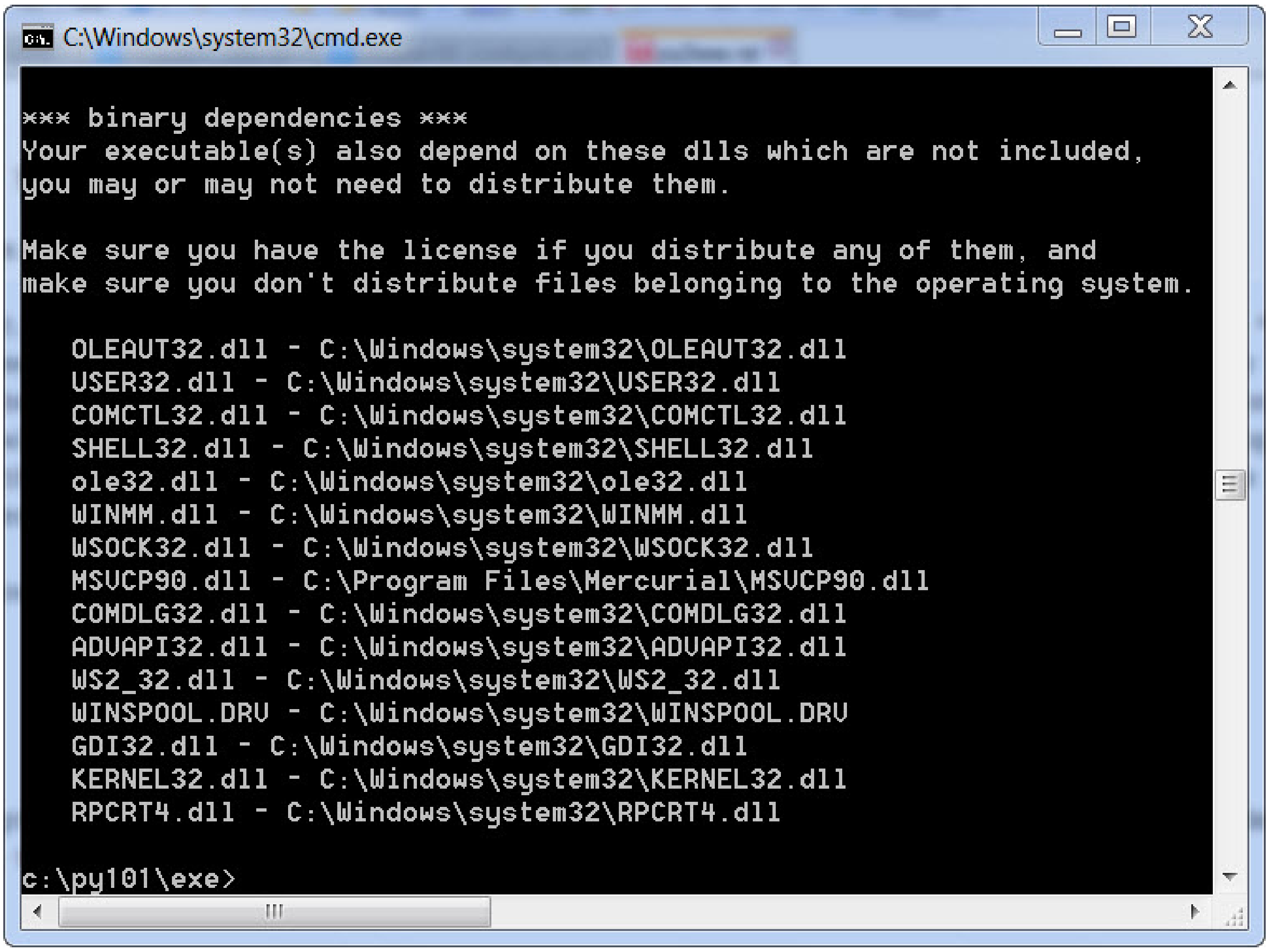
The py2exe setup.py file

The key to any py2exe script is the **setup.py** file. This file controls what gets included or excluded, how much we compress and bundle, and much more! Here is the simplest setup that we can use with the wx script above:

```
from distutils.core import setup
import py2exe

setup(windows=['sampleApp.py'])
```

As you can see, we import the **setup** method from **distutils.core** and then we import **py2exe**. Next we call setup with a **windows** keyword parameter and pass it the name of the main file inside a python list object. If you were creating a non-GUI project, than you would use the **console** key instead of **windows**. To run this snippet, save it into the same folder as your wxPython script, open up a command prompt and navigate to the location that you saved the two files. Then type **python setup.py py2exe** to run it. If all goes well, you will see a lot of output ending with something like this:



```
*** binary dependencies ***
Your executable(s) also depend on these dlls which are not included,
you may or may not need to distribute them.

Make sure you have the license if you distribute any of them, and
make sure you don't distribute files belonging to the operating system.

   OLEAUT32.dll - C:\Windows\system32\OLEAUT32.dll
   USER32.dll - C:\Windows\system32\USER32.dll
   COMCTL32.dll - C:\Windows\system32\COMCTL32.dll
   SHELL32.dll - C:\Windows\system32\SHELL32.dll
   ole32.dll - C:\Windows\system32\ole32.dll
   WINMM.dll - C:\Windows\system32\WINMM.dll
   WSOCK32.dll - C:\Windows\system32\WSOCK32.dll
   MSVCP90.dll - C:\Program Files\Mercurial\MSVCP90.dll
   COMDLG32.dll - C:\Windows\system32\COMDLG32.dll
   ADVAPI32.dll - C:\Windows\system32\ADVAPI32.dll
   WS2_32.dll - C:\Windows\system32\WS2_32.dll
   WINSPOOL.DRV - C:\Windows\system32\WINSPOOL.DRV
   GDI32.dll - C:\Windows\system32\GDI32.dll
   KERNEL32.dll - C:\Windows\system32\KERNEL32.dll
   RPCRT4.dll - C:\Windows\system32\RPCRT4.dll

c:\py101\exe>
```

If you happen to use Python 2.6, you might get an error about **MSVCP90.dll** not being found. Should you see that error, you will probably need to go find the **Microsoft Visual C++ 2008 Redistributable Package** and install it to make the DLL available on your system. Occasionally you will create the executable and then when you run it, it just won't load correctly. A log file is normally created when this happens that you can use to try to figure out what happened. I have also found a tool called **Dependency Walker** that you can run against your executable and it can tell you about non-Python items that are missing (like DLLs, etc).

I would like to point out that the **setup.py** file doesn't explicitly include wxPython. That means that py2exe was smart enough to include the wxPython package automatically. Let's spend some time learning a bit more about including and excluding packages.