Getting Started with cx_Freeze

As mentioned on the cx_Freeze website, there are three ways to use this script. The first is to just use the included cxfreeze script; the second is to create a distutils setup script (think py2exe) which you can save for future use; and the third is to work with the internals of cxfreeze. We will focus on the first two ways of using cx_Freeze. We'll begin with the console script:

```
# config_1.py
import configobj

def createConfig(configFile):
    """
    Create the configuration file
    """
    config = configobj.ConfigObj()
    inifile = configFile
    config.filename = inifile
    config['server'] = "http://www.google.com"
    config['username'] = "mike"
    config['password'] = "dingbat"
    config['update interval'] = 2
    config.write()

def getConfig(configFile):
    """
    Open the config file and return a configobj
    """
    return configobj.ConfigObj(configFile)

def createConfig2(path):
    """
    Create a config file
    """
    config = configobj.ConfigObj()
    config.filename = path
    config["Sony"] = {}
    config["Sony"]["product"] = "Sony PS3"
    config["Sony"]["accessories"] = ['controller', 'eye', 'memory stick']
    config["Sony"]["retail price"] = "$400"
    config.write()

if __name__ == "__main__":
    createConfig2("sampleConfig2.ini")
```

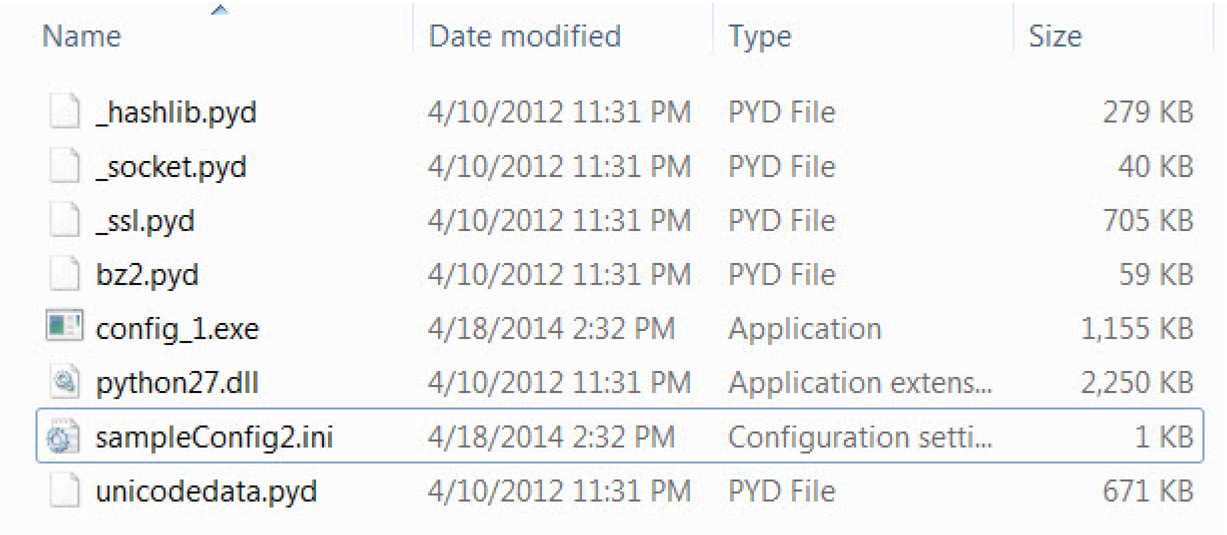
All this script does is create a really simple configuration file using Michael

All this script does is create a really simple configuration file using Michael Foord's **configobj** module. You can set it up to read the config too, but for this

example, we'll skip that. Let's find out how to build a binary with cx_Freeze! According to the documentation, all it should take is the following string on the command line (assuming you are in the correct directory):

```
cxfreeze config_1.py --target-dir dirName
```

This assumes that you have **C:\PythonXX\Scripts** on your path. If not, you'll either have to fix that or type out the fully qualified path. Anyway, if the cxfreeze script runs correctly, you should have a folder with the following contents:



Name	Date modified	Type	Size
_hashlib.pyd	4/10/2012 11:31 PM	PYD File	279 KB
_socket.pyd	4/10/2012 11:31 PM	PYD File	40 KB
_ssl.pyd	4/10/2012 11:31 PM	PYD File	705 KB
bz2.pyd	4/10/2012 11:31 PM	PYD File	59 KB
config_1.exe	4/18/2014 2:32 PM	Application	1,155 KB
python27.dll	4/10/2012 11:31 PM	Application extens...	2,250 KB
sampleConfig2.ini	4/18/2014 2:32 PM	Configuration setti...	1 KB
unicodedata.pyd	4/10/2012 11:31 PM	PYD File	671 KB

As you can see, the total file size should around 5 megabytes. That was pretty easy. It even picked up the configobj module without our having to tell it to. There are 18 command line arguments you can pass to cx_Freeze to control how it does things. These range from what modules to include or exclude, optimization, compression, include a zip file, path manipulation and more.

Now let's try something a little more advanced.