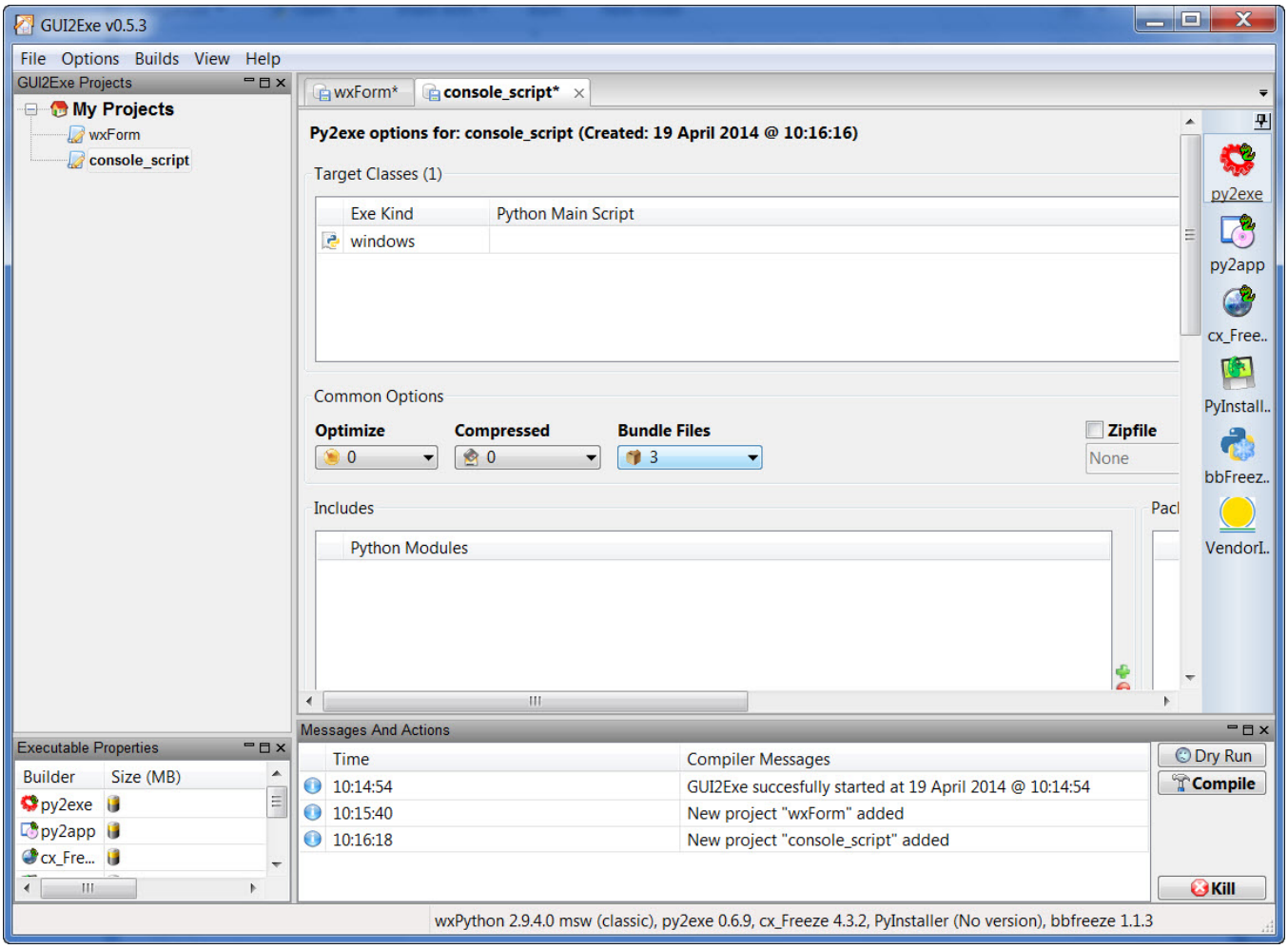
Getting Started with GUI2Exe

To use GUI2Exe, you just have to go to its website (http://code.google.com/p/gui2exe/) and download a release. Then you unzip it and run the script that's called **GUI2Exe.py**. The GUI2Exe project is based on wxPython, so make sure you have that installed as well. I ran mine successfully with wxPython 2.9. Here's how you would call it:

```
python GUI2Exe.py
```

If that executed successfully, you should see a screen similar to this one:



Now go to File -> New Project and give your project a name. In this case, I called the project **wxForm**. If you want to, you can add a fake Company Name, Copyright and give it a Program Name. Be sure to also browse for your main

Python script (i.e. **sampleApp.py**). According to Andrea's website, you should

set **Optimize** to **2**, **Compressed** to **2** and **Bundled Files** to **1**. This seems to work most of the time, but I've had some screwy errors that seem to stem from setting the last one to **1**. In fact, according to one of my contacts on the py2exe mailing list, the **bundle** option should be set to **3** to minimize errors. The nice thing about setting bundle to "1" is that you end up with just one file, but since I'm going to roll it up with Inno I'm going to go with option 3 to make sure my program works well.

Once you have everything the way you want it, click the **Compile** button in the lower right-hand corner. This will create all the files you want to distribute in the **dist** folder unless you have changed the name by checking the **dist checkbox** and editing the subsequent textbox. When it's done compiling, GUI2Exe will ask you if you want to test your executable. Go ahead and hit **Yes**. If you receive any errors about missing modules, you can add them in the **Python Modules** or **Python Packages** section as appropriate. For this example, you shouldn't have that issue though.

Now we're ready to learn about creating the installer!