

Creating an Advanced setup.py File

WE'LL COVER THE FOLLOWING ^

- Wrapping Up

Let's see what other options py2exe gives us for creating binaries by creating a more complex **setup.py** file.

```
from distutils.core import setup
import py2exe

includes = []
excludes = ['_gtkagg', '_tkagg', 'bsddb', 'curses', 'email', 'pywin.debugger',
            'pywin.debugger.dbgcon', 'pywin.dialogs', 'tcl',
            'Tkconstants', 'Tkinter']
packages = []
dll_excludes = ['libgdk-win32-2.0-0.dll', 'libgobject-2.0-0.dll', 'tcl84.dll',
                'tk84.dll']

setup(
    options = {"py2exe": {"compressed": 2,
                          "optimize": 2,
                          "includes": includes,
                          "excludes": excludes,
                          "packages": packages,
                          "dll_excludes": dll_excludes,
                          "bundle_files": 3,
                          "dist_dir": "dist",
                          "xref": False,
                          "skip_archive": False,
                          "ascii": False,
                          "custom_boot_script": '',
                          }
              },
    windows=['sampleApp.py']
)
```

This is pretty self-explanatory, but let's unpack it anyway. First we set up a few lists that we pass to the options parameter of the setup function.

- The **includes** list is for special modules that you need to specifically

include. Sometimes py2exe can't find certain modules, so you get to manually specify them here.

- The **excludes** list is a list of which modules to exclude from your program. In this case, we don't need Tkinter since we're using wxPython. This list of excludes is what GUI2Exe will exclude by default.
- The **packages** list is a list of specific packages to include. Again, sometimes py2exe just can't find something. I've had to include email, PyCrypto, or lxml here before. Note that if the excludes list contains something you're trying to include in the packages or includes lists, py2exe may continue to exclude them.
- **dll_excludes** - excludes dlls that we don't need in our project.

In the **options** dictionary, we have a few other options to look at. The **compressed** key tells py2exe whether or not to compress the zipfile, if it's set. The **optimize** key sets the optimization level. Zero is no optimization and 2 is the highest. By setting **optimize** to 2, we can reduce the size of folder by about one megabyte. The **bundle_files** key bundles dlls in the zipfile or the exe. Valid values for **bundle_files** are:

- 1 = bundle everything, including the Python interpreter.
- 2 = bundle everything but the Python interpreter
- 3 = don't bundle (default)

A few years ago, when I was first learning py2exe, I asked on their mailing list what the best option was because I was having issues with bundle option 1. I was told that 3 was probably the most stable. I went with that and stopped having random problems, so that's what I currently recommend. If you don't like distributing more than one file, zip them up or create an installer. The only other option I use in this list is the **dist_dir** one. I use it to experiment with different build options or to create custom builds when I don't want to overwrite my main good build. You can read about all the other options on the py2exe website.

The py2exe package does not support including Python eggs in its binaries, so if you have installed a package that your application depends on as an egg,

when you go to create the executable, it won't work. You will have to make sure your dependencies are installed normally.

Wrapping Up

At this point, you should know enough to get started using py2exe yourself. You can get busy and start distributing your latest creation now. It should be noted that there are several alternatives to py2exe, such as **bbfreeze**, **cx_freeze** and **PyInstaller**. You should try at least a couple of the others to see how they compare. Creating executables can be frustrating, but have patience and persevere through it. The Python packaging community is quite willing to help. All you need to do is ask.