



At the time of printing, cx\_Freeze for Python 3.5 is not released, but is expected soon!



The cx\_Freeze tool can also create executable files for Mac and Linux systems. Discover more on cx\_Freeze online at [cx\\_freeze.readthedocs.org](http://cx_freeze.readthedocs.org)



setup.py

# Deploying applications

Your apps developed in the Python language can be deployed on Windows systems where the Python interpreter is not installed. To do so, all the program files must be “frozen” into a bundle that includes an executable (.exe) file to create a simple MSI installer. The “cx\_Freeze” tool is a free set of scripts and modules for freezing Python programs available at [cx-freeze.sourceforge.net](http://cx-freeze.sourceforge.net)

The cx\_Freeze tool uses Python’s “distutils” package and requires a setup script to describe your module distribution in order to bundle appropriate support for your application. The setup script is traditionally named **setup.py** and consists mainly of a call to a cx\_Freeze **setup()** function – supplying information as parameter pairs. This specifies any required build options, such as image files or modules to be included, and identifies the executable script and system platform type. For example, the setup script for the application developed throughout this chapter must include the logo image file **lotto.gif** and specify the final code script named **lotto.py** as the executable script. Once cx\_Freeze is installed, a setup script can be executed from a Windows Command Prompt with the argument **bdist-msi** – to create a sub-directory named “dist” containing a distributable MSI installer for your app.

1

In IDLE start a Python setup script by making available the “sys” module and items from the “cx\_Freeze” module

```
import sys
from cx_Freeze import setup, Executable
```

2

Next, add statements to identify the base platform in use

```
base = None
if sys.platform == 'win32' : base = 'Win32GUI'
```

3

Now, add a statement listing options to be included

```
opts = { 'include_files' : [ 'lotto.gif' ] , 'includes' : [ 're' ] }
```

4

Finally, add a call to the **setup()** function passing all information as parameter pairs

```
setup( name = 'Lotto' ,
       version = '1.0' ,
       description = 'Lottery Number Picker' ,
       author = 'Mike McGrath' ,
       options = { 'build_exe' : opts } ,
       executables = [ Executable( 'lotto.py' , base= base ) ] )
```