

# Python Packaging

# setup.py

- This is a convention in Python: usually when you download Python source, there will be a file called “setup.py”
- When you see this file, you can run the command:

*python setup.py install*

to automatically build and install the package.

# Dependencies

- `setup.py` contains all of the information needed to install one Python package.
- What if that package has dependencies? You'll need to download those packages and install them with their `setup.py` scripts as well.
- What if those dependencies have dependencies? Etc., etc....

# Setuptools

- Solution: you can use the “easy\_install” tool, found in setuptools

Download (for Windows):

<http://pypi.python.org/pypi/setuptools#files>

# easy\_install

- Python keeps an online repository of common packages
- `easy_install` will search this repository to download the package you want and automatically handle any dependencies.
- Example:

*`easy_install scipy`*

# py2exe

- py2exe is an extension that allows you to use a setup.py script to create a windows executable (.exe) from a Python module or package.

Website:

<http://www.py2exe.org/>

# py2exe

- py2exe can package a custom Python installation containing your package and any other packages that are needed into a single file.
- It will automatically read through your Python scripts looking for import statements and include those modules.

# Other distutils extensions

- You can use the same `setup.py` script to package your Python application on multiple platforms
  - Debian-based Linux systems (such as Ubuntu): **stdeb**  
<https://github.com/astraw/stdeb>
  - Mac: **py2app**  
<http://svn.pythonmac.org/py2app/py2app/trunk/doc/index.html>

# Once again...

- The EcoData Retriever is an example of how to package a graphical application for Windows, Ubuntu, and Mac using a single setup.py script

Source:

<https://github.com/weecology/retriever/blob/master/setup.py>