

How to use Python distutils?

Asked 16 years, 3 months ago Modified 6 years ago Viewed 11k times



25



I wrote a quick program in python to add a gtk GUI to a cli program. I was wondering how I can create an installer using distutils. Since it's just a GUI frontend for a command line app it only works in *nix anyway so I'm not worried about it being cross platform.



my main goal is to create a .deb package for debian/ubuntu users, but I don't understand make/configure files. I've primarily been a web developer up until now.

edit: Does anyone know of a project that uses distutils so I could see it in action and, you know, actually try building it?

Here are a few useful links

- [Ubuntu Python Packaging Guide](#)

This Guide is **very** helpful. I don't know how I missed it during my initial wave of googling. It even walks you through packaging up an existing python application

- [The Ubuntu MOTU Project](#)

This is the official package maintaining project at ubuntu. Anyone can join, and there are lots of

tutorials and info about creating packages, of all types, which include the above 'python packaging guide'.

- ["Python distutils to deb?" - Ars Technica Forum discussion](#)

According to this conversation, you can't just use distutils. It doesn't follow the debian packaging format (or something like that). I guess that's why you need dh_make as seen in the Ubuntu Packaging guide

- ["A bdist_deb command for distutils](#)

This one has some interesting discussion (it's also how I found the ubuntu guide) about concatenating a zip-file and a shell script to create some kind of universal executable (anything with python and bash that is). weird. Let me know if anyone finds more info on this practice because I've never heard of it.

- [Description of the deb format and how distutils fit in - python mailing list](#)

python

linux

installation

debian

distutils

Share

Improve this question

Follow

edited Dec 6, 2018 at 13:50



Cœur

38.6k ● 26 ● 202 ● 276

asked Aug 27, 2008 at 5:03



Jiaaro

76.8k ● 43 ● 174 ● 191

5 Answers

Sorted by:

Highest score (default)



14



See the [distutils simple example](#). That's basically what it is like, except real install scripts usually contain a bit more information. I have not seen any that are fundamentally more complicated, though. In essence, you just give it a list of what needs to be installed. Sometimes you need to give it some mapping dicts since the source and installed trees might not be the same.

Here is a real-life (anonymized) example:

```
#!/usr/bin/python

from distutils.core import setup

setup (name = 'Initech Package 3',
       description = "Services and libraries ABC, D",
       author = "That Guy, Initech Ltd",
       author_email = "that.guy@initech.com",
       version = '1.0.5',
       package_dir = {'Package3' : 'site-packages/P'},
       packages = ['Package3', 'Package3.Queries'],
       data_files = [
           ('/etc/Package3',
            ['etc/Package3/ExternalResources.conf'])
       ])
```

Share Improve this answer

answered Aug 27, 2008 at 5:12

Follow



Sander

26.3k ● 3 ● 54 ● 88

can the data_files path be absolute ? I've got ValueError: path '/etc/Package3' cannot be absolute – [Sake](#) May 24, 2011 at 11:13

Plus 1 for "Initech" - the only thing that would have been better is "Plas-trol-tech" – [Ben DeMott](#) Sep 26, 2011 at 20:37



`apt-get install python-stdeb`

6

Python to Debian source package conversion utility



This package provides some tools to produce Debian packages from Python packages via a new distutils command, `sdist_dsc`. Automatic defaults are provided for the Debian package, but many aspects of the resulting package can be customized via a configuration file.

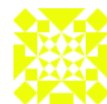


- `pypi-install` will query the Python Package Index (PyPI) for a package, download it, create a `.deb` from it, and then install the `.deb`.
- `py2dsc` will convert a distutils-built source tarball into a Debian source package.

Share Improve this answer

answered Jan 17, 2011 at 19:00

Follow



[Stuart Cardall](#)

2,417 ● 29 ● 22



Most Python programs will use distutils. [Django](#) is a one - see

2

<http://code.djangoproject.com/svn/django/trunk/setup.py>



You should also read [the documentation](#), as it's very comprehensive and has some good examples.



Share Improve this answer

answered Aug 27, 2008 at 10:04

Follow



[Andrew Wilkinson](#)

10.8k ● 3 ● 36 ● 38



I found the following [tutorial](#) to be very helpful. It's shorter than the distutils documentation and explains how to setup a typical project step by step.

2



Share Improve this answer

answered Nov 15, 2010 at 21:57

Follow



[Rusty Shackelford](#)

81 ● 4



1

distutils really isn't all that difficult once you get the hang of it. It's really just a matter of putting in some meta-information (program name, author, version, etc) and then selecting what files you want to include. For example, here's a sample distutils setup.py module from a decently complex python library:



[Kamaelia setup.py](#)

Note that this doesn't deal with any data files or or whatnot, so YMMV.

On another note, I agree that the distutils documentation is probably some of python's worst documentation. It is extremely inclusive in some areas, but neglects some really important information in others.

Share Improve this answer

answered Aug 27, 2008 at 20:42

Follow



Jason Baker

198k ● 138 ● 382 ● 520
