The state of packaging

Tarek Ziadé

tarek@critsend.com

@tarek_ziade
Google Moderator
http://tiny.cc/packaging13
Packaging sucks
We need to package and release our Python apps! on all systems!

I’ll work on it!

Packaging? mm.. Distutils or Setuptools should work

Friday, February 19, 2010
So how do we install our product?

easy_install foo

Sweet! How do we remove it?

Manually

Are you kidding?

But boss, you want a portable installer!
I’ll use virtualenv!

And create local installation with everything

For every app

sysadmins will hate me
But I want to improve packaging!

This will be my contribution to Python

Let's try to fix things from the root
pip install foo

python setup.py install

easy_install foo

site-packages

User PoV

Friday, February 19, 2010
Standardization
What is presented

1. Metadata standardization
2. Version schemas
3. Installed distributions index
What is not presented

• build/compiling flaws in Distutils
• target-specific binary distributions problems
• trillions of things we are working on...
Distutils is not cmake
Part I

Metadata standardization
PEP 314
Metadata for Python Software Package 1.1
PKG-INFO example

Metadata-Version: 1.0
Name: pip
Version: 0.6
Summary: pip installs packages. Python packages. An easy_install replacement
Home-page: http://pip.openplans.org
Author: The Open Planning Project
Author-email: python-virtualenv@groups.google.com
License: MIT
Keywords: easy_install distutils setuptools egg virtualenv
Platform: UNKNOWN
Classifier: Development Status :: 4 - Beta
Classifier: Intended Audience :: Developers
Classifier: License :: OSI Approved :: MIT License
Classifier: Topic :: Software Development :: Build Tools
Distutils metadata cycle

Project
  setup.py
  setup(options)

python setup.py sdist

Distribution
  PKG-INFO
    (PEP 314)

PyPI
  PKG-INFO
    (PEP 314)

Target system
  PKG-INFO
    (PEP 314)
Setuptools metadata cycle

**Project**
- setup.py
  - setup(options)

**python setup.py sdist**

**Distribution**
- PKG-INFO
  - (PEP 314)
  - + requires.txt
  - + ....

**PyPI**
- PKG-INFO
  - (PEP 314)
  - + ?????

**Target system**
- PKG-INFO
  - (PEP 314)
  - + requires.txt
  - + ....
PEP 345
Metadata for Python Software Package 1.2

Accepted
Requires-Dist

Requires-Dist: zope.interface (>3.5)

Requires-Dist: PasteDeploy
Version specifiers

Requires-Dist: zope.interface (>3.5, <4.0)

<, >, <=, >=, == and !=
Version specifiers

- > 2.6, < 3
- 3.5 means 3.5.x
- 2.5.0 means >=2.5.0,<2.5.1
- >=3.1,!=3.1.3,<3.2
Requires-Python

Requires-Python: >2.4

Requires-Python: >=2.5, <2.7
Requires-External

Requires-External: libxslt

Requires-External: libpng (>=1.5)
Provides-Dist

Provides-Dist: transaction

Provides-Dist: ZODB

Zodb

transaction
Obsoletes-Dist

Obsoletes-Dist: Setuptools
Project-URL

Description: This project provides powerful math functions. For example, you can use `sum()` to sum numbers:

Example:

```python
>>> sum(1, 2)
3
```
PEP 345 - Summary

• Requires-Dist *
• Obsoletes-Dist *
• Provides-Dist *
• Requires-External
• Requires-Python *
• Project-URL
• Description

* with version specifier
PEP 345 for developers

```python
from setuptools import setup

setup(name='MyProject',
      version='1.0',
      install_requires=['zope.interface'])
```
from distutils.core import setup

setup(name='MyProject',
      version='1.0',
      requires_dist=['zope.interface'],
      project_urls=[('bug tracker',
                     'http://project.org/issues'),
                     ('mailing list',
                      'http://project.org/lists')])
Environment markers
import sys

requires_dist=['zope.interface']

if sys.platform == 'win32':
    requires_dist.append('pywin32')

setup(name='MyProject',
      version='1.0',
      requires_dist=requires_dist)
Dynamic metadata

PyPI

MyProject.tar.gz

local

Pip

python setup.py egg_info

Metadata

get dependencies

python setup.py install

Friday, February 19, 2010
Environment Markers

Requires-Dist: value (version); marker

- sys.platform == 'win32' and platform.machine == 'i386'
- 'linux' in sys.platform
- python_version == '2.4'
Environment Markers

- `python_version = '%s.%s' % (sys.version_info[0], sys.version_info[1])`
- `python_full_version = sys.version.split()[0]`
- `os.name = os.name`
- `sys.platform = sys.platform`
- `platform.version = platform.version()`
- `platform.machine = platform.machine()`
- a free string, like '2.4', or 'win32'
Environment Markers

• Requires-Python
• Requires-External
• Requires-Dist
• Provides-Dist
• Obsoletes-Dist
• Classifier
Pseudo-static metadata fields

```python
from distutils.core import setup

setup(name='MyProject',
      version='1.0',
      requires_dist=[
        'zope.interface',
        'pywin32; sys.platform == 'win32''])
```
Pseudo-static metadata

PyPI

MyProject.tar.gz

+ metadata

local

Pip

python setup.py install

get dependencies

Friday, February 19, 2010
I love PEP 345!
Part 2

Version schemas
Brought to you by PyPI

20090105

RC

unreleased.unofficialdev

1.*
Version 1 > Version 2?

Distutils’
LooseVersion
StrictVersion

Setupools’
parse_version()

Result vary....
Distutils

```python
>>> import distutils.version
>>> V = distutils.version.LooseVersion

>>> V('1.2rc1') < V('1.2')
False

>>> V('1.2-a1') < V('1.2')
False

>>> V('1.2a1') < V('1.2')
False
```

Setuptools

```python
>>> import pkg_resources
>>> V = pkg_resources.parse_version

>>> V('1.2rc1') < V('1.2')
True

>>> V('1.2-a1') < V('1.2')
True

>>> V('1.2a1') < V('1.2')
True
```
Version scheme requirements

• final version
• pre-final version
• post-release versions
• development versions at all stages
Version scheme requirements

• final
  • MAJOR.MINOR.MICRO
• pre-final
  • a1, a2, b1...c1, rc2, ...
• post-release
  • +r145, -r1256, ...
• development
  • dev12, ~dev12, -dev12, ...
PEP 386
Interoperable version comparison scheme
Accepted
major, minor, micro


pre-tag

extra numbers

post-tag

dev tag

N = integer
1.0a1 < 1.0a2 < 1.0b2 < 1.0c1 < 1.0 < 1.1
development versions

1.0a1 < 1.0a2.dev345 < 1.0b1 < 1.0.dev432 < 1.0
suggest_normalized_version(your scheme)

PEP 386 compatible scheme
>>> from verlib import NormalizedVersion
>>> NormalizedVersion('1.2-a1') < NormalizedVersion('1.2')
Traceback (most recent call last):
...
    raise IrrationalVersionError(s)
verlib.IrrationalVersionError: 1.2-a1
>>> from verlib import suggest_normalized_version
>>> suggest_normalized_version('1.2-a1')
'1.2a1'
>>> NormalizedVersion('1.2a1') < NormalizedVersion('1.2')
True
PEP 386 @ PyPI distributions

- Direct match: 90%
- suggest_normalized_version: 8%
- No match: 3%

Friday, February 19, 2010
We love PEP 386
Part 3

Installed distributions

index
PEP 262

A database of installed packages
Distutils standard

MyProject

PKG-INFO

(PEP 314)

python setup.py install

Target system

site-packages

MyProject.egg-info

Directory

file

Friday, February 19, 2010
Setuptools’ own standard

MyProject

PKG-INFO

(PEP 314)
+ requires.txt
+ ....

easy_install MyProject

Target system

site-packages

MyProject.egg-info

PKG-INFO

requires.txt

SOURCES.txt

Directory
file

Friday, February 19, 2010
Pip’s own standard

MyProject

PKG-INFO

(PEP 314)

+ requires.txt
+ ....

pip install MyProject

Target system

site-packages

PKG-INFO

installed_files.txt

requires.txt

SOURCES.txt

MyProject.egg-info
PEP 376

Interoperable .egg-info structure
PEP 376

- .egg-info becomes a directory
- INSTALLER
- RECORD
- REQUESTED
- query functions in Distutils
- uninstall function in Distutils
More after the sprints
Conclusion

Summit output
Q/A
Google Moderator
Thanks!

http://ziade.org
#distutils @ freenode
@tarek_ziade