SOFTWARE PACKAGING

Workshop on Advanced Techniques for Scientific Programming and Management of Open Source Software Packages

Beraldo Leal
April 21, 2015
"Talk is cheap. Show me the code."

- Linus Torvalds
INTRODUCTION
IF YOU WANNA SHARE:

∙ You need a package!
IF YOU WANNA SHARE:

- You need a package!
- Installation process should be as painless as possible for your user / contributor;
IF YOU WANNA SHARE:

- You need a package!
- Installation process should be as painless as possible for your user / contributor;
- Use a version control system (choose a modern one);
IF YOU WANNA SHARE:

· You need a package!
· Installation process should be as painless as possible for your user / contributor;
· Use a version control system (choose a modern one);
· Don’t send your code by email (or host at dropbox);
WHAT IS A PACKAGE?

- We are **not** talking about python import package;
- A compressed (or not) file containing other files in a directory structure that will be installed on the target system;
- On steroids:
  - pre/post (un)install scripts;
  - documentation;
  - dependencies and libraries;
  - etc.
WHAT IS A PACKAGE?

· Like a tarball file?
WHAT IS A PACKAGE?

- Like a tarball file?
- Well, let’s use something more sophisticated!
WHAT IS A PACKAGE?

- Like a tarball file?
- Well, let’s use something more sophisticated!
They try to follow the Filesystem Hierarchy Standard (FHS);
They try to follow the Filesystem Hierarchy Standard (FHS);
Can be binary or source package;
· They try to follow the Filesystem Hierarchy Standard (FHS);
· Can be binary or source package;
· Each distro has a different review process;
· They try to follow the Filesystem Hierarchy Standard (FHS);
· Can be binary or source package;
· Each distro has a different review process;
· .deb: any debian-like distro;
They try to follow the Filesystem Hierarchy Standard (FHS);
Can be binary or source package;
Each distro has a different review process;
.deb: any debian-like distro;
.rpm: RedHat Package Manager;
They try to follow the Filesystem Hierarchy Standard (FHS);
Can be binary or source package;
Each distro has a different review process;
.deb: any debian-like distro;
.rpm: RedHat Package Manager;
ABS: Arch Build System;
They try to follow the Filesystem Hierarchy Standard (FHS);
Can be binary or source package;
Each distro has a different review process;
.deb: any debian-like distro;
rpm: RedHat Package Manager;
ABS: Arch Build System;
Ports-like from BSD;
· Did you remember rpmfind.net?
The search service can find package by either name (apache), provides(webserver), absolute file names (/usr/bin/apache), binaries (gprof) or shared libraries (libx11).

The System and Arch are optional added filters, for example System could be "redhat", "redhat-7.2", "mandrake" or "gnome", Arch could be "i386" or "src", etc...

xclock

2 active mirrors: [fr] [fr2]

**RPM resource xclock**

The xclock program displays the time in analog or digital form. The time is continuously updated at a frequency which may be specified by the user.

Found 12 RPM for xclock

<table>
<thead>
<tr>
<th>Package</th>
<th>Summary</th>
<th>Distribution</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>xorg-x11-apps-7.6-5.fc17.i686.html</td>
<td>X.Org X11 applications</td>
<td>Fedora Rawhide for i386</td>
<td>xorg-x11-apps-7.6-5.fc17.i686.rpm</td>
</tr>
<tr>
<td>xorg-x11-apps-7.6-5.fc17.x86_64.html</td>
<td>X.Org X11 applications</td>
<td>Fedora Rawhide for x86_64</td>
<td>xorg-x11-apps-7.6-5.fc17.x86_64.rpm</td>
</tr>
<tr>
<td>xorg-x11-apps-7.6-2.fc15.i686.html</td>
<td>X.Org X11 applications</td>
<td>Fedora 16 for i386</td>
<td>xorg-x11-apps-7.6-2.fc15.i686.rpm</td>
</tr>
<tr>
<td>xorg-x11-apps-7.6-2.fc15.x86_64.html</td>
<td>X.Org X11 applications</td>
<td>Fedora 15 for i386</td>
<td>xorg-x11-apps-7.6-2.fc15.x86_64.rpm</td>
</tr>
<tr>
<td>xorg-x11-apps-7.6-2.fc15.x86_64.html</td>
<td>X.Org X11 applications</td>
<td>Fedora 16 for x86_64</td>
<td>xorg-x11-apps-7.6-2.fc15.x86_64.rpm</td>
</tr>
<tr>
<td>xorg-x11-apps-7.4-14.fc14.i686.html</td>
<td>X.Org X11 applications</td>
<td>Fedora 14 for i386</td>
<td>xorg-x11-apps-7.4-14.fc14.i686.rpm</td>
</tr>
</tbody>
</table>
three releases:
  - **stable**: contains the latest officially released distribution of Debian;
  - **testing**: contains packages that haven’t been accepted into a ”stable” release yet, but they are in the queue for that;
  - **unstable (sid)**: is where active development of Debian occurs.
AND ABOUT PYTHON PACKAGES?
• 2000: `distutils` was added to the Python standard library in Python 1.6;
2000: *distutils* was added to the Python standard library in Python 1.6;

2003: *PyPI* was up and running;
∙ 2000: distutils was added to the Python standard library in Python 1.6;
∙ 2003: PyPI was up and running;
∙ To not be confuse: ”distribution” vs ”python package”.
PyPI - the Python Package Index

The Python Package Index is a repository of software for the Python programming language. There are currently 58091 packages here.

To contact the PyPI admins, please use the Support or Bug reports links.

Get Packages
To use a package from this index either "pip install package" (get pip) or download, unpack and "python setup.py install" it.

Package Authors
Submit packages with "python setup.py upload": The index hosts package docs. You may also use the web form. You must register. Testing? Use testpypi.

Infrastructure
To interoperate with the index use the JSON, OAuth, XML-RPC or HTTP interfaces. Use local mirroring or caching to make installation more robust.

<table>
<thead>
<tr>
<th>Updated</th>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-04-16</td>
<td>trollius_redis 0.1.0</td>
<td>PEP 3156 implementation of the redis protocol.</td>
</tr>
<tr>
<td>2015-04-16</td>
<td>Themer 1.1</td>
<td>Themer is a colorscheme generator and manager for your desktop.</td>
</tr>
<tr>
<td>2015-04-16</td>
<td>agms 0.1.0</td>
<td>Agms Python Library</td>
</tr>
<tr>
<td>2015-04-16</td>
<td>circlator 0.2.0</td>
<td>circlator: a tool to circularise bacterial genome assemblies</td>
</tr>
<tr>
<td>2015-04-16</td>
<td>pyper 0.1b.10</td>
<td>A toolkit for numerical simulations to allow easy parameter exploration and storage of results.</td>
</tr>
<tr>
<td>2015-04-16</td>
<td>yoyo-migrations 4.2.5</td>
<td>Database schema migration tool using SQL and DB-API</td>
</tr>
<tr>
<td>2015-04-16</td>
<td>ebayasdk 2.1.1</td>
<td>eBay SDK for Python</td>
</tr>
<tr>
<td>2015-04-16</td>
<td>rodeo 0.1.0</td>
<td>an ide for data analysis in python</td>
</tr>
<tr>
<td>2015-04-16</td>
<td>django-user-guide 0.8.1</td>
<td>Show configurable HTML guides to users.</td>
</tr>
<tr>
<td>2015-04-16</td>
<td>topsky 1.0.1</td>
<td>A simple wrapper library for the Topsy APIs</td>
</tr>
<tr>
<td>2015-04-16</td>
<td>django-celery-model 0.1</td>
<td>django-celery-model is an extension to Celery and django-celery which adds support for tracking Celery tasks assigned to Django model instances.</td>
</tr>
<tr>
<td>2015-04-16</td>
<td>zymbit 0.4.0</td>
<td>Zymbit cloud library</td>
</tr>
<tr>
<td>2015-04-16</td>
<td>dlc-sdk 2.12</td>
<td>CenturyLink Cloud SDK and CLI</td>
</tr>
</tbody>
</table>
from distutils.core import setup
setup(name='foobar',
    version='1.0',
    author='tarek',
    author_email='foo@bar.org',
    url='http://example.com',
    packages=[
        'foobar',
        'foobar.subfoo',
    ],
)
# Build source distribution
$ python setup.py sdist
$ ls -l dist/
 total 4
 -rw-r--r-- 1 user user 491 Apr 17 14:22 hello-1.0.tar.gz
# Manual installation
$ tar zxvf hello-1.0.tar.gz
$ cd hello-1.0
$ sudo python setup.py install
# Register at PyPi and upload

$ python setup.py register  
$ python setup.py sdist upload
# Install from PyPi
$ sudo pip install hello
But...

- `distutils` is not really designed for compiled packages: Ex: `numpy.distutils`
- No uninstalling;
- You cannot use built-in dependencies!
- Only possible with `requirements.txt` file:
  ```
  $ cat requirements.txt
  FooProject >= 1.2
  SomethingWhichVersionIDontCareAbout
  BarProject == 1.0
  # Install requirements
  $ sudo pip install -r requirements.txt
  ```
But...

- distutils is not really designed for compiled packages: Ex: `numpy.distutils` fork;

...
PYTHON PACKAGES: DISTUTILS PROBLEMS

· But...
· distutils is not really designed for compiled packages: Ex: numpy.distutils fork;
· no uninstalling;
· But...

· distutils is not really designed for compiled packages: Ex: `numpy.distutils` fork;
· no uninstalling;
· You cannot use built in dependencies!
But...

- distutils is not really designed for compiled packages: Ex: `numpy.distutils` fork;
- no uninstalling;
- You cannot use built in dependencies!
- Only possible with `requirements.txt` file:
But...

- distutils is not really designed for compiled packages: Ex: `numpy.distutils` fork;
- no uninstalling;
- You cannot use built in dependencies!
- Only possible with `requirements.txt` file:

```
$ cat requirements.txt
FooProject >= 1.2
SomethingWhichVersionIDontCareAbout
BarProject == 1.0

# Install requirements
$ sudo pip install -r requirements.txt
```
2004: setuptools was introduced by Phillip Eby, which included the Egg format, and the ability to declare and automatically install dependencies;
· 2004: setuptools was introduced by Phillip Eby, which included the Egg format, and the ability to declare and automatically install dependencies;

```python
from ez_setup import use_setuptools
use_setuptools()

from setuptools import setup
from setuptools import find_packages

setup(name='smr2763hello',
      version='0.2',
      author='me',
      author_email='foo@bar.org',
      url='http://example.com',
      packages = find_packages(),
      )
```
2007: virtualenv was introduced by Ian Bicking, which allowed users to create isolated Python environments based on a central system installation of Python;
2007: *virtualenv* was introduced by Ian Bicking, which allowed users to create isolated Python environments based on a central system installation of Python;
· 2011: The PyPA is created to take over the maintenance of pip and virtualenv from Ian Bicking, led by Carl Meyer, Brian Rosner and Jannis Leidel. Other proposed names were “ianb-ng”, “cabal”, “pack” and “Ministry of Installation”.


· **2011**: The PyPA is created to take over the maintenance of pip and virtualenv from Ian Bicking, led by Carl Meyer, Brian Rosner and Jannis Leidel. Other proposed names were ”ianb-ng”, ”cabal”, ”pack” and ”Ministry of Installation”.

· **2012**: The effort to include ”Distutils2/Packaging” in Python 3.3 was abandoned due lack of involvement;
2011: The PyPA is created to take over the maintenance of pip and virtualenv from Ian Bicking, led by Carl Meyer, Brian Rosner and Jannis Leidel. Other proposed names were ”ianb-ng”, ”cabal”, ”pack” and ”Ministry of Installation”.

2012: The effort to include ”Distutils2/Packaging” in Python 3.3 was abandoned due lack of involvement;

2013: PEP425 and PEP427 were accepted. Together, they specify a built-package format for Python called Wheel;
DEMO
SUMMARIZING
TOOL RECOMMENDATIONS:

- **Always** try to install from your distro repository;

---

1 from Python Packaging User Guide
TOOL RECOMMENDATIONS:

- Always try to install from your distro repository;
- But if you need to use PyPI \(^1\)

\(^1\)from Python Packaging User Guide
TOOL RECOMMENDATIONS:

- **Always** try to install from your distro repository;
- But if you need to use PyPI \(^1\)
  - Use `pip` to install Python packages from PyPI;

\(^1\)from Python Packaging User Guide
TOOL RECOMMENDATIONS:

- **Always** try to install from your distro repository;
- But if you need to use PyPI:
  - Use **pip** to install Python packages from **PyPI**;
  - Use **virtualenv**, or **pyvenv** to isolate application specific dependencies from a shared Python installation;

---

¹from Python Packaging User Guide
TOOL RECOMMENDATIONS:

- **Always** try to install from your distro repository;
- But if you need to use PyPI
  - Use `pip` to install Python packages from PyPI;
  - Use `virtualenv`, or `pyvenv` to isolate application specific dependencies from a shared Python installation;
  - Use `setuptools` to define projects and create Source Distributions.

---

1from Python Packaging User Guide
QUESTIONS?