Paver: the build tool you missed

Lukas Linhart
Centrum Holdings

About Me

- Regular pythonista since '05
- Testing and automation obsession
- Centrum Holdings, mostly Django-related work
- Lots of smallish projects

Stumbling upon Paver

- "Setuptools no more"
- Kevin Dangoor To The Rescue

Survey

Who would use a build tool...

. presenter notes: yeeah

Build tool in an interpreted world

- Generated Content In Repository (TM)
- Moving (files) around

Single Point of Entry

As in "Console API"

```
1 paver bootstrap
2 paver prepare
3 paver run
```

Paver vs. rest

Getting started (First Task!)

`pavement.py`:

```
1 from paver.easy import *
2
3 @task
4 def install_dependencies():
5     sh('pip install --upgrade -r requirements.txt')
```

Embrace distutils/setuptools

```
1 from paver.easy import *
2 from paver.setuputils import setup
3
4 setup(**same_args_as_in_setup)
```

setup.py compatibility

```
paver minilib

paver generate_setup

(pip install -e worky)
```

Dependencies

```
1 @task
2 @needs('install_dependencies')
3 def prepare():
4     """ Prepare complete environment """
5     sh("python setup.py develop")
```

Overwriting distutils commands

```
1 @task
2 @needs('html', "minilib", "generate_setup", "distutils.command.sdist")
3 def sdist():
4     pass
```

Command line arguments (positional)

```
1 @task
2 @consume_args
3 def unit(args):
4     import nose
5     nose.run_exit(
6         argv = ["nosetests"] + args
7     )
```

Command line arguments (GNU style)

```
1 @task
2 @cmdopts([
3     ('domain-username=', 'd', 'Domain username'),
4     ('upload-url=', 'u', 'URL to upload to')
5 ])
6 @needs('download_diff_packages')
7 def upload_packages(options):
8     # censored
```

Oh, options

```
1 options(
2     minilib=Bunch(
3         extra_files=['doctools', 'virtual']
4     )
5 )
```

Options (cont.)

```
1 options.get('extra_files', [])
```

Namespace search

```
1 options(setup=Bunch(version="1.1"))
2 options.version
3
4 options.order('minilib')
5 options.version
```

sh

```
1 myval = sh("cat /etc/fstab", capture=True)
```

dry

```
1 # prepare
2 dry("Modify", do_fs_mumbo_jumbo)
```

path.py

```
1 @task
2 def publish_docs():
3     builtdocs = path("docs") / options.sphinx.builddir / "html"
4     destdir = options.docroot
5     destdir.rmtree()
6     builtdocs.move(destdir)
```

Documentation (sphinx)

```
options(
    sphinx=Bunch(
        builddir="build",
        sourcedir="source"
    )
)

!sh
paver html
```

Documentation (cog)

```
1 [[[ replace ]]]
```

Virtualenv

```
options(
    virtualenv=Bunch(
        packages_to_install=["nose", "virtualenv"],
        install_paver=True,
        script_name='bootstrap.py',
        paver_command_line=None,
        dest_dir="virtualenv"
    )
)
```

Discovery

Setting up django discovery

```
 1 from paver.discovery import discover_django
 2
 3 options(
 4     discovery = Bunch(
 5         django = Bunch(
 6             settings_path = "subdir"
 7         )
 8     )
 9 )
10
11 discover_django(options)
```

Happy Django command panda

```
paver django.validate
```

Future

- virtualenv improvements
- VIRTUAL_ENV context autodetection
- more integration (first-class fabric, ...)
- what would you like?

Q & (some) A

Fork us on github
http://github.com/paver/paver/
paver@googlegroups.com