

Python Web Installer

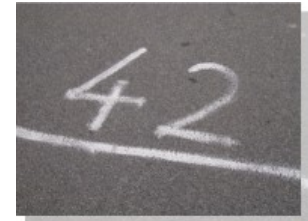
PyCon UK 2014
Coventry, UK

Marc-André Lemburg

Speaker Introduction

Marc-André Lemburg

- Python since 1993/1994
- Studied Mathematics
- eGenix.com GmbH
- Python Core Developer
- Python Software Foundation
- EuroPython Society
- Based in Düsseldorf, Germany
- Available for Python Coaching, Training and Consulting



Agenda

- Introduction
- Finding a Solution
- Implementation Design
- Solution: mxSetup.py
- Questions



Agenda

- Introduction
- Finding a Solution
- Implementation Design
- Solution: mxSetup.py
- Questions



Introduction

- Why yet another installation mechanism ?
 - Waited too long for a solution
 - Found the existing ones not addressing the problems our customers have
 - Python packaging community not really open to the needs of commercial vendors
 - Lock-in on PyPI



Introduction: What are we looking for ?

- **Distribute binary Python packages**
(packages with C extensions)
 - Have the installer find the right binaries to install automatically
 - Ideally: Have added features in the installer such as automatic eval license generation



Existing Tool Support

- Tools
 - `setuptools`, `easy_install`, `zc.buildout`, `pip`
- **Pure Python** packages
 - **well supported**
- Packages with **C-Source Code**
 - **mostly supported**, as long as a compiler and the needed external libraries are available
 - **almost never the case on Windows**



Existing Tool Support: Binary Packages

- Tools
 - setuptools, easy_install, zc.buildout, pip
- Binary packages on **Windows**
 - **limited support** via eggs and wheels
- Binary packages on **Linux**
 - **limited support**, but eggs only
- Binary packages on **other platforms**
(FreeBSD, Solaris, Mac OS X)
 - **poor or no support**



Secure Downloads

- Secure distribution of Python packages
 - User should be sure to get unmodified packages
 - Prevent Man-in-the-middle (MITM) attacks
- `setuptools`, `easy_install`, `zc.buildout`, `pip`:
 - Check the MD5 checksums, if present
 - Always available on PyPI download URLs
 - MD5 is not really secure anymore
 - Many other download URLs don't include checksums
 - Older Python versions don't verify HTTPS certificates



Goals: Summary

- Easy to use and secure package installation
- Automatic selection of the needed package download for the platform
- Checksum verification with all Python versions
- Ideally: HTTPS certificate verification for all Python versions (at least for Python 2.5, 2.6, 2.7)



Agenda

- Introduction
- Finding a Solution
- Implementation Design
- Solution: mxSetup.py
- Questions



Non-Solution

- **Waiting** for PyPI, pip, setuptools, zc.buildout, etc. to implement all this any time soon
- **Which is fine**, since we have a particular need, which others may not have



LCD – Least Common Denominator

- Let's *try to reuse what's already there ...*
- All installers support the “python setup.py install” command line interface
- All installers support “source” packages



Solution: Python web installer

- Create **web installer packages** which:

look like **Python source packages**

- Upload these to PyPI

implement the “**python setup.py install**” interface

- Web installers can do their magic when running setup.py



Solution: Compatibility with existing installers

- Let the **existing installers** take care of ...
 - managing the installations
 - dependency checks
 - meta data handling
 - installation database
 - implementing PEPs



Solution: Requirements checks

- Web installers can then **automate the package download selection**
 - in a package dependent way
 - with knowledge about platform variants and requirements
- Web installers can **add checksum and certificate checks** to the mix
 - by monkey patching the Python StdLib
 - using their own download routines



Solution: Requirements checks

- Web installers can **add extra functionality**
 - setup local configurations
 - search for libraries, header files
 - inform the user of missing libraries and how to install them
 - generate licenses
 - etc.



Agenda

- Introduction
- Finding a Solution
- Implementation Design
- Solution: mxSetup.py
- Questions



Web Installer: Implementation

- Packages are uploaded as **minimal source packages** to PyPI
 - only contain top-level Python modules
 - setup.py
 - **Web installer support module**
 - README
 - license information
 - etc.



Web Installer: Implementation

- Use [special code in setup.py](#) (the web installer) to detect whether it is running as web installer or not
 - detection **based on landmarks** (files, directories)
 - web installation only runs in case these landmarks are **missing**



Web Installer: Implementation

- Web installer takes care of all **platform checks** directly on the platform
 - **programmatically, not declarative**
 - can add additional checks such as for: external libraries, CPU variants, etc.
 - encodes the found information using **platform tags**



Web Installer: Implementation

- Web installer **loads the missing parts** of the package into the installation directory
 - selection **based on package tags** and
 - a **matching algorithm** to match platform to package tags



Agenda

- Introduction
- Finding a Solution
- Implementation Design
- Solution: mxSetup.py
- Questions



Solution: mxSetup

- **mxSetup** – Open Source Packaging Module
 - based on and extends [distutils](#)
 - works with [setuptools](#) as well
 - active development since 2001
 - Part of [egenix-mx-base](#)
 - supports [uninstall](#), [prebuilt](#) packages, web installers and many other things
 - currently only Python 2.4 – 2.7, Python 3.4+ support planned



New distutils command: `bdist_prebuilt`

- `bdist_prebuilt` runs the build step, stores the distutils state and packages up everything in a ZIP file
 - allows shipping binary packages
 - does not include the source files
 - includes all build files
 - packages support the standard “python setup.py install” interface

→ Think of prebuilt packages as “frozen” package builds



New distutils command: `sdist_web`

- `sdist_web` creates the web installer packages for upload to PyPI
 - based on distutils `sdist`
 - only adds top-level `.py` files to the archive – without the landmarks
 - includes `mxSetup.py` in the packages
 - builds a `setup.tags` file and adds it to the archive
 - looks and feels like a source code package



Binary package selection

- Based on **tags** which identify the properties of the available binary packages
- Tag examples:
 - `prebuilt, linux, x86, py2.7, ucs2`
 - `prebuilt, linux, x64, py2.7, ucs4`
 - `sdist, source, compiler`
- **Matching algorithm** finds the best fit on the installation platform



Combining URLs and tags: The setup.tags file

- Lists the download URLs with hash checksums (using hash tags)
- Associates each URL with a list of tags
- Simple, but effective
 - Allows secure downloads (checksums)
 - Allows downloads from arbitrary URLs
 - HTTPS is not necessary for secure downloads
 - Included in the web installer package to prevent MITM attacks
 - Doesn't need special file naming conventions



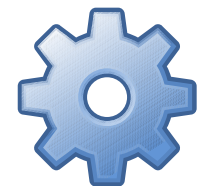
Example without URLs and hash tags

egenix-web-installer-test-0.2.0-py2.4_ucs2-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.4, ucs2
egenix-web-installer-test-0.2.0-py2.4_ucs4-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.4, ucs4
egenix-web-installer-test-0.2.0-py2.5_ucs2-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.5, ucs2
egenix-web-installer-test-0.2.0-py2.5_ucs4-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.5, ucs4
egenix-web-installer-test-0.2.0-py2.6_ucs2-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.6, ucs2
egenix-web-installer-test-0.2.0-py2.6_ucs4-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.6, ucs4
egenix-web-installer-test-0.2.0-py2.7_ucs2-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.7, ucs2
egenix-web-installer-test-0.2.0-py2.7_ucs4-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.7, ucs4
egenix-web-installer-test-0.2.0.zip -> sdist, source, compiler



Example without URLs and hash tags

egenix-web-installer-test-0.2.0-py2.4_ucs2-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.4, ucs2
egenix-web-installer-test-0.2.0-py2.4_ucs4-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.4, ucs4
egenix-web-installer-test-0.2.0-py2.5_ucs2-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.5, ucs2
egenix-web-installer-test-0.2.0-py2.5_ucs4-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.5, ucs4
egenix-web-installer-test-0.2.0-py2.6_ucs2-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.6, ucs2
egenix-web-installer-test-0.2.0-py2.6_ucs4-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.6, ucs4
egenix-web-installer-test-0.2.0-py2.7_ucs2-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.7, ucs2
egenix-web-installer-test-0.2.0-py2.7_ucs4-linux-x86_64-prebuilt.zip -> prebuilt, linux, x64, py2.7, ucs4
egenix-web-installer-test-0.2.0.zip -> sdist, source, compiler



Example without URLs and hash tags: **Close-ups**

egenix-web-installer-test-0.2.0-py2.4_ucs2-linux-x86_64-prebuilt.zip

→prebuilt, linux, x64, py2.4, ucs2

egenix-web-installer-test-0.2.0.zip

→sdist, source, compiler



setup.tags file: Example with hash tags

Without URLs and hash tags:

egenix-web-installer-test-0.2.0-py2.7_ucs4-linux-x86_64-prebuilt.zip -> prebuilt,
linux, x64, py2.7, ucs4

egenix-web-installer-test-0.2.0.zip -> sdist, source, compiler

With hash tags and URLs:

https://downloads.egenix.com/python/egenix-web-installer-test-0.2.0-py2.7_ucs4-linux-x86_64-prebuilt.zip#md5=d61ee73e96f3286ca23ac3ab35846643&sha1=551200ca6c375c67eb061ce64294c9a5966e5d24&sha256=50e18bfdb6dfaabf4efaa215f0e4ae3001479ed4e3a89f545b45a1d6a3ee0904&size=908557 -> prebuilt, linux, x64, py2.7, ucs4

<https://downloads.egenix.com/python/egenix-web-installer-test-0.2.0.zip#md5=ec827fd7200e2cc17fa728093e12045a&sha1=b7954de2cb288d8744e82fcd0e530642cfd7c5f2&sha256=104532b4a2c459a26b25306d43cb3eb9edfd16657e089f78e27d0fa898&size=900616> -> sdist, source, compiler



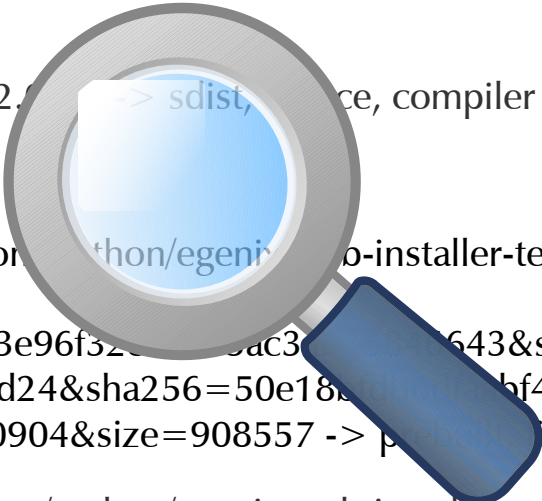
setup.tags file: Example with hash tags

Without URLs and hash tags:

egenix-web-installer-test-0.2.0-py2.7_ucs4-linux-x86_64-prebuilt.zip -> prebuilt,
linux, x64, py2.7, ucs4

egenix-web-installer-test-0.2.0-py2.7_ucs4-linux-x86_64-prebuilt.zip -> sdist, source, compiler

With hash tags and URLs:


https://downloads.egenix.com/python/egenix-web-installer-test-0.2.0-py2.7_ucs4-linux-x86_64-prebuilt.zip#md5=d61ee73e96f32c5ac3b1e643&sha1=551200ca6c375c67eb061ce64294c9a5966e5d24&sha256=50e18a1bf4efaa215f0e4ae3001479ed4e3a89f545b45a1d6a3ee0904&size=908557 -> prebuilt, linux, x64, py2.7, ucs4

https://downloads.egenix.com/python/egenix-web-installer-test-0.2.0-py2.7_ucs4-linux-x86_64-prebuilt.zip#md5=ec827fd7200e2cc17fa728093e12045a&sha1=b7954de2cb288d8744e82fcd0e530642cfd7c5f2&sha256=104532b4a2c459a26b25306d43cb3eb1edfd16657e089f78e27d0fa898&size=900616 -> sdist, source, compiler 

setup.tags file: **Close-up** of an entry

https://downloads.eGenix.com/python/egenix-web-installer-test-0.2.0-py2.7_ucs4-linux-x86_64-prebuilt.zip

#

md5=d61ee73e96f3286ca23ac3ab35846643&

sha1=551200ca6c375c67eb061ce64294c9a5966e5d24&

sha256=50e18bfdb6dfaabf4efaa215f0e4ae3001479ed4e3a89f545b45a1d6a3ee0904&

size=908557

→ prebuilt, linux, x64, py2.7, ucs4



Create setup.tags

- Manually using a text editor
- Automatic using mxSetup (as <distfile>.tags) for each package file; then concatenate using cat:

```
dist.tags:    dist/${PACKAGENAME}-${VERSION}*.tags  
cat dist/${PACKAGENAME}-${VERSION}*.tags > dist.tags
```

```
sdist-web:    dist.tags  
$(PYTHON) setup.py sdist_web \  
    --package-tags=dist.tags \  
    --base-url=${DOWNLOAD_BASE_URL} \  
    --dist-dir=dist/web/
```



Python Web Installer in action

- Add the following snippet to setup.py:

```
#  
# Run web installer, if needed  
#  
import mxSetup, os  
mxSetup.run_web_installer(  
    os.path.dirname(os.path.abspath(__file__)),  
    landmarks=('mx', 'PREBUILT'))
```

... that's all



What's behind `mxSetup.run_web_installer()` ? 1/2

- `run_web_installer()` adds the web installer functionality to `setup.py`:
- Checks for landmarks

`if landmarks_present:`

 no actions taken

`else:`

 # run the web installation code...



What's behind `mxSetup.run_web_installer()` ? 2/2

else:

run the web installation code...

- Loads `setup.tags` file
- Determines the `platform tags`
- Uses a `matching algorithm` to find the right download package for the platform
- **Downloads the package** and **unpacks it**
 - Checks the checksums
 - Unpacks the package in the installer directory
 - Double-checks for landmarks



Python Web Installer: Current status

- Create tags: **works**
- Verify checksums: **works**
- Build web installer: **works**
- Deploy web installer: **works**
- HTTPS certificate checks: **TBD**
- Documentation: **TBD**



Present & Future

- **Availability:**
 - **egenix-mx-base 3.2.8** includes the first working version
 - **egenix-mx-base 3.2.8** ships with it
 - **egenix-mxodbc 3.3.1** ships with it
 - Zero complaints so far ...
- **Roadmap:**
 - Add documentation, more tools
 - Add more binary formats
 - Check GPG signatures
 - Check HTTPS certificates



Agenda

- Introduction
- Finding a Solution
- Implementation Design
- Solution: mxSetup.py
- Questions

Questions



Thank you for listening



Beautiful is better than ugly.

Contact

eGenix.com Software, Skills and Services GmbH

Marc-André Lemburg

Pastor-Löh-Str. 48

D-40764 Langenfeld

Germany

eMail: mal@egenix.com

Phone: +49 211 9304112

Fax: +49 211 3005250

Web: <http://www.egenix.com/>