

Re: Wheel–reinvention with Python

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2005-08/msg00629.html>

- *From:* Cliff Wells <cliff@xxxxxxxxxxx>
 - *Date:* Mon, 01 Aug 2005 21:22:15 -0700
-

On Mon, 2005-08-01 at 23:49 -0400, Mike Meyer wrote:

> Cliff Wells <cliff@xxxxxxxxxxx> writes:

>

>> On Sun, 2005-07-31 at 14:58 -0400, Mike Meyer wrote:

>>> And what do I use to bundle my application for Unix? Most of the

>>> things I build get installed on Unix servers.

>> You install GUI apps on Unix *servers*?

>

> Yup. Thanks to the wonders of X, I can run GUI apps on servers and

> have them display on my desktop. Or my OS X laptop. I normally leave a

> gkrellm running on most of my servers.

Yes, yes. The "wonders of X" have been known to me since around 1992. Personally I avoid running GUI apps on my servers since they are usually unnecessary, waste memory, and if not used properly, open the door for security issues (for that matter, any extra piece of software installed opens the door for potential security issues so I tend to run stripped-down servers that only provide needed services – no fluff. Then again "server" is a pretty broad term. The bulk of my servers are shared and dedicated hosting environments where paranoia is rewarded. Your situation may be different).

> Of course, the problem under discussion isn't restricted to GUI

> apps. Anytime I use a third party library, I have to deal with how end

> users are going to get it.

Absolutely. However GUI libraries tend to stick out a bit more since they tend to have a larger number of dependencies plus sheer code size and complexity makes platform-specific bugs more likely.

>> Regardless, when you say "Unix", what do you mean? You may as well say

>> "OS" as this term has little meaning. Linux (which flavor)? BSD? SCO?

>> HPUNIX? OS/X? Minix? Whichever way, I suspect that a bit of distutils

>> hacking would solve your problem.

>

> I think the post I replied to covered the OS X case – there's an

> application bundler available for it already.

Yes, I've used it to bundle wxPython apps on 10.3.

Re: Wheel–reinvention with Python

- > I want distributions that will work on all three major BSD variants
- > and most Linux distributions – in particular, anything that uses deb's,
- > rpm's or emerge. If you want to choose one in particular, try ubuntu
- > Linux.

OpenBSD: in portage

FreeBSD: in portage

386BSD: don't know... does anyone still use this?

NetBSD: included AFAICT

Fedora: rpms available

Debian: debs available

Ubuntu: included

Gentoo: in portage

OS/X: available on wxpython.org

FWIW, much of this info I found in less than 5 minutes using Google. I suppose with a little persistence you could have discovered it for yourself.

Here's your Ubuntu instructions:

http://wiki.wxpython.org/index.cgi/wxPython_20with_20Ubuntu

Of course, if you are distributing end–user software by requiring users to install 3rd party packages themselves, then I hope you are distributing open source software. If it's a commercial/closed–source app you'll do far better by simply packaging all the requirements into the installer and installing it within the application's directory tree (this solves a lot of problems with version conflicts and user confusion).

In my experience, binary packages work best for Windows and OS/X, and source distributions work best for Linux (mostly because building packages for Linux is a varied and sometimes painful process. Further Linux truly works best as a source–based system: you cannot always depend on ABI compatibility across versions of libraries – *usually* you can but... caveat emptor).

Regards,
Cliff

—

cliff@xxxxxxxxxxxx

<http://www.develix.com> :: Web applications and hosting :: Linux, PostgreSQL and Python specialists ::

.

• *Follow–Ups:*

◆ **Re: Wheel–reinvention with Python**

◇ From: Mike Meyer

• **References:**

◆ **Re: Ten Essential Development Practices**

◇ From: Dark Cowherd

◆ **Re: Ten Essential Development Practices**

◇ From: Michael Hoffman

◆ **Wheel–reinvention with Python (was: Ten Essential Development Practices)**

◇ From: Torsten Bronger

◆ **Re: Wheel–reinvention with Python (was: Ten Essential Development Practices)**

◇ From: Calvin Spealman

◆ **Re: Wheel–reinvention with Python**

◇ From: Torsten Bronger

◆ **Re: Wheel–reinvention with Python**

◇ From: Mike Meyer

◆ **Re: Wheel–reinvention with Python**

◇ From: Cliff Wells

◆ **Re: Wheel–reinvention with Python**

◇ From: Mike Meyer

◆ **Re: Wheel–reinvention with Python**

◇ From: Cliff Wells

◆ **Re: Wheel–reinvention with Python**

◇ From: Mike Meyer

• Prev by Date: **Re: Wheel–reinvention with Python**

• Next by Date: **Re: Path as a dictionary tree key? (was Re: PEP on path module for standard library)**

• Previous by thread: **Re: Wheel–reinvention with Python**

• Next by thread: **Re: Wheel–reinvention with Python**

• Index(es):

◆ **Date**

◆ **Thread**