1. Python 应用发布技术

收集如何 将 Py 应用打包发布的各种技巧:

1.1. 工具

{{{k <yanbo.yuan@gmail.com> reply-to python-cn@googlegroups.com, to python-cn@googlegroups.com, date Tue, Apr 1, 2008 at 2:58 PM subject [CPyUG:45605]}}}[http://groups.google.com/group/python-cn/t/24fbb899f27df30e 将 Python 程序转化为可执行程序~整理]

工欲善其事,必先利其器.python 是解释型的语言,但是在 windows 下如果要执行程序的话还得加个 python shell 的话,未免也太麻烦了.而这里所说的东西就是将 python 程序转换为 exe 文件.下面是一些常用的工具,不过似乎 py2exe 应用的更加广泛一些.

1.1.1. py2exe

- http://py2exe.sf.net

只支持 windows 平台，应该是大家听到最多的一个名字了，用户不少，所以有问题的话在它的 mail list 里面很容易找到答案。文档中提到了"无法找到某某 code"、使用 opengl 等模块的问题

1.1.2. PyPackage

- http://codereactor.net/projects/pypack/index.html

我觉得 py2exe 等等工具还是罗嗦得像唐僧，需要在配置文件中写上需要的数据文件。作者完全无视这样一个事实：我需要发布可执行文件的时候，程序已经完工了，所有的数据文件就在主程序所在目录下，所以多数情况下，根本不用到别的地方搜索。现在终于有人站了出来，PyPackage 实际上并不是一个程序打包的工具，而只是简化 py2exe 的操作，甚至可以自动调用 InnoSetup 5 制作安装文件

不过这个软件并不智能，会打包很多不需要的文件

1.1.3. Installer

- http://www.mcmillan-inc.com/installer_dnld.html

可以产生 windows、linux 平台的可执行文件，现在作者主页连不上去了，但是搜索一下可以在其它地方下载 自带一个小程序写配置文件，如果程序较复杂，还是需要手工修改这个配置文件。支持从 py15 以来的所有 Python 版本

- 2005 年 9 月，冰冻牡蛎更新：Giovanni Bajo 获得 Gordon McMillan's Installer 的授权、版权改变为 GPL，
    - 并在 http://pyinstaller.hpcf.upr.edu/ 继续开发 PYinstaller。
- 2006 年 9 月更新：这里可以看到 Gordon McMillan's 的原始网站的镜像

1.1.4. Python 自带的 freeze.py

（不过 windows 版本不带这个，你可以自己下载 Python 的源程序再找）。 这个是我最不推荐的一种方法（为什么？自己看），不过如果你的 Python 程序要发布到其它工具不支持的平台上，可以考虑这个方法

1.1.5. Pyco

新出来的

- http://www.pythonapocrypha.com/projects/pyco/

还没用过

1.1.6. Squeeze

- http://starship.python.net/crew/fredrik/ipa/squeeze.htm

还没用过，只支持 Python 1.4

1.1.7. cx_Freeze

- http://starship.python.net/crew/atuining/cx_Freeze/

winodws、linux 平台。简单的程序甚至都不需要写配置文件

1.1.8. Stand alone Python for Windows

- http://arctrix.com/nas/python/standalone.html

如果你不介意源程序太过"暴露"的话，用这个吧

会不会觉得 Updated: Sun, 09 Apr 2000 18:39:54 -0600 扎眼？

- 如果你看一看它的 VC 源代码，就不会这么想了
    -

其实这是普遍适用于ｗｉｎ系统的方法，无论是９８、２０００或者ｘｐ。也许也可以用到ｌｉｎｕｘ上

-

我不懂ｌｉｎｕｘ，如果真的可以这么做，还请告诉我。

1.1.9. py2app

- http://undefined.org/python/

支持 linux 平台的工具可能也支持 mac os，或者直接使用这个 py2app。具体就不知道了，只吃过苹果，还没玩过苹果呢

1.1.10. Movable Python

- http://www.voidspace.org.uk/python/movpy/

这个其实是使用 py2exe 制作的、可以放在 U 盘上的绿色 Python。有使用 py2app 制作苹果版 movpy 和用 cx_Freeze 制作 Linux 版 movpy 的计划。懒到都不愿意学习 py2exe、py2app 或者 cx_Freeze 的人可以看看。

1.1.11. Shed Skin

- A Python-to-C++ Compiler:

- 试验项目，windows 上，连他的例子我都没有编译成功 :(。

1.1.12. Jungle

：　使用 GNU 工具（as、ld 和 winres）把 Python 程序编译到 windows 的 exe 可执行文件。

- 该可执行文件只使用基于 python24 的的 pythonic.dll。
- 猜测它支持的模块仅限于内部模块以及 jungle.jgl 列出的模块。
- 只有可执行文件下载，而这个可执行文件也是用 Jungle 自己编译的。
    - 目前版本号都到 1.10 了，经常看 0.xx 的版本号，这个数字好大啊，娃哈哈。

1.1.13. 另类的方法

，对 Python 语言特性都还不是 100％支持，众多的 CPython 模块也不可以使用，还有，我也没有试过：

for .NET 的 Python 编译器

- （如 Visual Python、IronPython），不过我可不喜欢为了一个芝麻大的软件安装.NET framework
- 用 jython，然后用 jbuilder、jsmooth、NativeJ 之类的包裹一下，或者用 gcj 编译成本地代码

1.1.13.1. Psyco

：　给 Python 程序加速的东西，看不出对发布 Python 程序的直接好处，并且作者以后将致力于 PyPy。

1.1.13.2. PyPy

：　项目目标是纯 Python 实现的 Python、速度比 CPython 快，将来可以帮助实现编译 Python。

1.1.13.3. pyc

：　Python compiler in Python，一个用纯 Python 写的 Python 的 bytecode 编译器，可以优化输出的 pyc 文件。

- 和 PyPy 一样，现在还看不出对发布 Python 程序的直接好处。只有 py24 的 bytecode。
- pyc 是 pyvm 这个新的 python 虚拟机的一部分。

1.1.13.4. Pyrex

使用 Pyrex

- 直接将 Python 应用编译成 C 代码，然后，自然就生成.exe 了

1.2. 体验

- 手动制作 python 的 exe 可执行程序 ~ by LeoJay

1.2.1. PyInstaller

{{{nEO (a.k.a. gentoo.cn) <gentoo.cn@gmail.com> reply-to python-cn@googlegroups.com, to python-cn@googlegroups.com, date Wed, Apr 2, 2008 at 12:34 AM }}} http://pyinstaller.python-hosting.com/

吸取了 py2exe 的优点，支持打包成一个可执行文件，支持 upx 壳，支持多平台
体积比 py2exe 生成的小 我现在用这个替代 py2exe 了

1.2.2. NSIS

{{{Gerald Lee <leejd80@gmail.com> reply-to python-cn@googlegroups.com, to python-cn@googlegroups.com, date Tue, Apr 1, 2008 at 8:19 PM subject [CPyUG:45664] Re: 将 Python 程序转化为可执行程序[整理] }}}

- 这两天一直在写 NSIS 脚本做安装程序，因为是 N 个模块定制安装的，所以我用 python 写，然后输出 NSI 脚本文件，再逐个编译，以产生需要的安装文件。
- 写脚本的过程突然发现一个问题，是不是可以使用 nsis 脚本来引导 python 程序呢？以前一直是用 bat 文件引导的。顺手写了一个，测试感觉可行，生成的一个 exe 文件 33K，NSIS 代码如下：

```
Name "Python Launcher"
Icon "images\shipping.ico"
OutFile "GridOK.exe"

SilentInstall silent
AutoCloseWindow true
ShowInstDetails nevershow

Section ""
  Exec "pythonw GridOK.py";这里可能需要变动一下
SectionEnd
```

参考资料：

- http://www.blogjava.net/xilaile/archive/2007/05/13/117039.html

1.2.3. 实例 Py2exe

在最后,给一个人学习 py2exe 的文章,帮助学习：最近学了一点 PYTHON，想把 PYTHON 写的程序转换成 EXE 文件，在网上查到了资料后发现了这个东东 写下来做一下记录。 英文教程：

- http://www.py2exe.org/index.cgi/Tutorial

Python 2.5 + Py2exe 工作目录：c:\python25

首先随便写一个程序

```
hello.py
print "Hello World!"
```

测试一下是否能运行

```
python hello.py
结果：Hello World
```

- 到 www.py2exe.org 下载 PY2exe ,或者在 SF 上下载

接下来直接安装 PY2EXE 包

- 。。它是一个安装文件。。直接装就行了。
- 下在编写一个设置的 PY 文件 setup.py

```
from distutils.core import setup
import py2exe
```

```
setup(console=['hello.py'])
```

运行：python setup.py py2exe 出现以下信息后，在 DIST 目录里，就会有一个 hello.exe 即成功。

```
running py2exe
*** searching for required modules ***
*** parsing results ***
creating python loader for extension 'zlib'
creating python loader for extension 'unicodedata'
creating python loader for extension 'bz2'
*** finding dlls needed ***
*** create binaries ***
*** byte compile python files ***
```

```
byte-compiling
C:\Tutorial\build\bdist.win32\winexe\temp\bz2.py to
bz2.pyc
byte-compiling
C:\Tutorial\build\bdist.win32\winexe\temp\unicodedata.py
to unicodedata.pyc
byte-compiling
C:\Tutorial\build\bdist.win32\winexe\temp\zlib.py to
zlib.pyc
skipping byte-compilation of c:\Python24\lib\StringIO.py to
StringIO.pyc
[skipping many lines for brevity]
skipping byte-compilation of c:\Python24\lib\warnings.py to
warnings.pyc
*** copy extensions ***
*** copy dlls ***
copying c:\Python24\lib\site-packages\py2exe\run.exe ->
C:\Tutorial\dist\hello.exe
*** binary dependencies ***
Your executable(s) also depend on these dlls which are not
included,
you may or may not need to distribute them.
Make sure you have the license if you distribute any of them,
and
make sure you don't distribute files belonging to the
operating system.
  ADVAPI32.dll - C:\WINDOWS\system32\ADVAPI32.dll
  USER32.dll - C:\WINDOWS\system32\USER32.dll
  SHELL32.dll - C:\WINDOWS\system32\SHELL32.dll
  KERNEL32.dll - C:\WINDOWS\system32\KERNEL32.dll
```