使用PyInstaller(USING PYINSTALLER)

PyInstaller的命令格式为：

pyinstaller [options] script [script…] | specfile

在大多数不复杂的情况下，你都可以使用下面这条简单的命令在当前目录打包你的代码：



然后PyInstaller就会分析myscript.py并做以下工作：

在相同的目录下，以脚本的形式编写myscript.spec文件

在相同的目录下，创建build目录

在build目录下写一些日志文件

创建dist目录

完成myscript可执行文件

最后打包的可执行文件会放置在dist目录中。通常而言，你仅需要在命令行指定一个代码文件。如果你指定多个代码文件的话，它们都会被分析并反馈于输出信息中。但是最后生成的spec文件与可执行文件都会按照第一个代码文件的名称进行命名。

在一些特殊的使用中，你或许需要编辑myscript.spec的内容。之后你需要执行

pyinstaller myscript.spec

或许还需要给要执行的代码或者spec文件指定路径，例如（虽然觉得有点罗嗦，但不得不说老外写的文档考虑的情况很多）：

pyinstaller options… ~/myproject/source/myscript.py

哦，如果你是在windows系统上操作，则需要输入:

pyinstaller “C:\Documents and Settings\project\myscript.spec”

选项(Options)

一般选项(General Options)

-h,--help

-v,--version

--distpath DIR 放置打包后程序的目录，默认./dist

--workpath WORKPATH 放置log,pyz等临时文件的目录

-y,--noconfirm 替换输出目录而不向用户发出确认，默认SPECPATH/dist/SPECNAME

--upx-dir UPX\_DIR UPX用途的路径，默认 搜索可执行的路径

-a,--ascii 不包含unicode编码支持，默认 如果可用则包含

--clean 在打包之前清理PyInstaller的缓存文件

--log-level LEVEL 将带有具体时间的控制台信息写成日志，可设置日志级别，默认为INFO级别

What to generate

-D,--onedir 生成包含可执行文件的单目录包（默认）

-F,--onefile 生成单个可执行文件的包

--spechpath DIR 指定存放产生的spec文件的目录（默认当前目录）

-n NAME, --name NAME 指定生成打包文件与spec文件的名称（默认为第一个代码文件名去除后缀的内容）

What to bundle,where to search

--add-data <SRC;DEST or SRC:DEST> 指定额外的需要添加进可执行文件的非二进制文件或目录，可添加多个路径，路径的分隔符由所在平台决定。Windwos上是； 大多数unix系统是： 这个选项可以使用多次

--add-binary <SRC;DEST or SRC:DEST> 指定额外的需要添加进可执行文件的二进制文件。和上面的用法一致。

-p DIR, --path DIR 搜索py脚本引用的内容，类似使用 PYTHONPATH,允许输入多个路径，用：分隔开便可，或者多次使用这个选项

--hiden-import MODULENAME，--hiddenimport MODULENAME

指定在代码中不可见的导入(这个我不是很明白，原文是an import not visible）

--additional-hooks-dir HOOKSPATH

指定一个搜索挂钩（hooks）的路径，选项可以多次使用

--runtime-hook RUNTIME\_HOOKS

Path to a custom runtime hook file. A runtime hook is code that is bundled with the executable and is executed before any other code or module to set up special features of the runtime environment. This option can be used multiple times.

--exclude-module EXCLUDES

Optional module or package (the Python name, not the path name) that will be ignored (as though it was not found). This option can be used multiple times.

--key KEY

用户加密python字节码的密钥

如何产生(How to Generate)

-d,--debug 让bootloader在初始化与开始打包的时候显示处理信息，用于诊断丢失导入相关的问题

-s.—strip 应用一张用于剥去共享库与可执行文件的符号表，不推荐在windows系统上使用

--noupx 即便UPX可用也不是用，此选项在Windows系统在\*nix上会有不同的使用效果

Windows与Mac OS系统共有的特殊选项

-c,--console,--nowindowed 给标准I/O开启一个控制台窗口（默认）

-w,--windowed,--noconsole 上面命令的相反内容，在Mac OS X上还会创建一个 OS X.app包。这个选项在linux系统上会被无视

-I <FILE.ico or FILE.exe,ID or FILE.icns>, --icon <FILE.ico or FILE.exe,ID or FILE.icns>

FILE.ico 给windows下的可执行文件指定一个文件图标 FILE.exe,ID 从一个exe文件上取出图标 FILE.icns在Mac OS X系统上给打包的.app文件添加图标

Windows系统特有选项

--version-file FILE 给生成的exe文件添加版本信息

-m <FILE or XML>,--manifest <FILE or XML> 给exe文件添加XML等信息文件

-r RESOURCE,--resource RESOURCE

Add or update a resource to a Windows executable. The RESOURCE is one to four items, FILE[,TYPE[,NAME[,LANGUAGE]]]. FILE can be a data file or an exe/dll. For data files, at least TYPE and NAME must be specified. LANGUAGE defaults to 0 or may be specified as wildcard \* to update all resources of the given TYPE and NAME. For exe/dll files, all resources from FILE will be added/updated to the final executable if TYPE, NAME and LANGUAGE are omitted or specified as wildcard \*.This option can be used multiple times.

--uac-admin

Using this option creates a Manifest which will request elevation upon application restart.

--uac-uiaccess

Using this option allows an elevated application to work with Remote Desktop.

Windows 并行组合搜索选项（高级使用）

Windows Side-by-side Assembly searching options(advanced)

--win-private-assemblies

Any Shared Assemblies bundled into the application will be changed into Private Assemblies. This means the exact versions of these assemblies will always be used, and any newer versions installed on user machines at the system level will be ignored.

--win-no-prefer-redirects

While searching for Shared or Private Assemblies to bundle into the application, PyInstaller will prefer not to follow policies that redirect to newer versions, and will try to bundle the exact versions of the assembly.

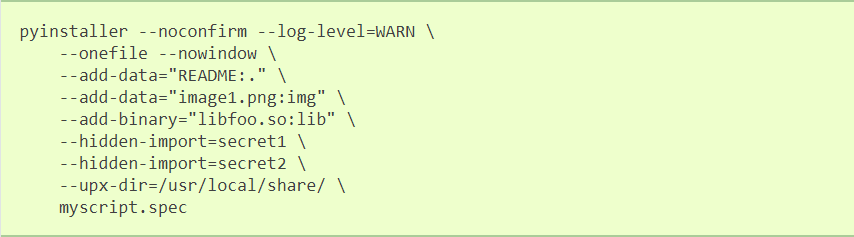
Mac OS X 特有选项

--osx-bundle-identifier BUNDLE\_IDENTIFIER

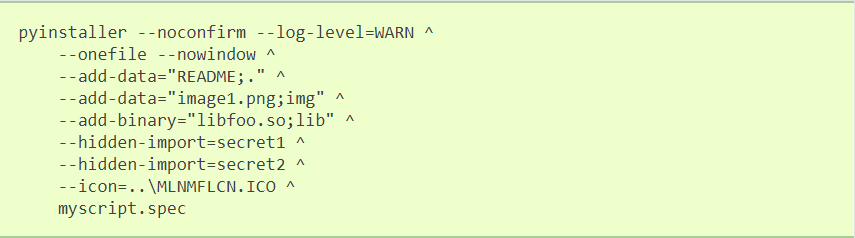
Mac OS X .app bundle identifier is used as the default unique program name for code signing purposes. The usual form is a hierarchical name in reverse DNS notation. For example: com.mycompany.department.appname (default: first script’s basename)

缩短命令(Shortening the Command)

看过上文你也了解到PyInstaller拥有这么多参数可以设定，一个完整的打包命令长度在命令行上会换行多次。而且打包工作常常不会一次性完成，意味着这么长的命令是需要输入多次的。那么可以使用换行符号让你的命令变得更可读和优美。



或者在windows系统上，可以编写成BAT文件的格式：



使用UPX(Using UPX)

UPX对于大多数操作系统而言是免费可用的。UPX可以把可执行文件与二进制文件压缩成更小的体积。查看UPX的主页来下载，并且可以在网页上查看可以压缩得可执行文件列表。UPX的开发似乎在2013年9月就结束了。除了在Mac OS X上的64位二进制文件上是不起效果的，UPX支持很多可执行文件格式。

UPX的启动代码可以包裹一个压缩过的可执行程序，当启动该程序的时候对其进行动态解压缩。当解压缩之后，程序就可以正常地运行。如果设置打包成单个文件，那么PyInstaller就会对其进行UPX压缩。

PyInstaller是通过命令中--upx-dir选项来寻找或者修改UPX的压缩路径。如果系统上UPX存在并且—nohpx参数不生效，PyInstaller就会应用它。UPX经常被用于PyInstaller的输出，大多数情况都是正常的。

加密Python字节码(Encrypting Python Bytecode)

如果你想给存放于打包文件中的Python字节码进行加密，那么打包命令行中的--key=参数是必不可少的。

如果想实现加密的功能，PyCrypto模组必须存在且可用。而你在—key选项所跟的参数应该是16位长的字符串。

多系统支持(Supporting Multiple Platforms)

如果你仅在同一套Python与OS下分发你的应用，那就像安装其他包一样安装PyInstaller，然后在开发中正常使用它。

多Python环境的支持(Supporting Multiple Python Environment)

当你所打包的程序的使用环境在同一操作系统，但在不同python版本与支持库的情况时，这里的叙述或许会对你有帮助。举例来说，打包称程序需要在Python3与python2.7两种不同的环境下使用，或一个python版本使用Qt4而另一个版本使用Qt5.这样的话，我只能推荐你是用，virtualenv。在virtualenv下你可以维持不同环境进行开发，而且可以在其中进行轻松地切换。当然，如果你是用的是3.4之后较新的版本，则可以使用内建的pyvenv做相同的事情。

当然你需要注意：如果应用的使用环境很复杂，你需要虚拟出可能的所有环境；在每个环境中都别忘记安装PyInstaller。

还有请了解，如果使用virtualenv，那么调用PyInstaller命令的路径会变成：

windows系统 ENV\_ROOT\Scripts

其他系统 ENV\_ROOT\bin

在windows系统下，使用pip-Win包会让virtualenv的安装变得十分简单。而在\*ix系统下，你需要通过命令行来切换。

支持多重操作系统(Supporting Multiple Operating Systems)

让Linux应用向前兼容(Making Linux Apps Forward-Compatible)

捕获Windows版本数据(Capturing Windows Version Data)

创建Mac OS X应用包(Building Mac OS X App Bundles)

让Mac OS X 应用向前兼容(Making Mac OS X apps Forward-Compatible)

在Mac OS X中创建32位应用(Building 32-bit Apps in Mac OS X)

获取开放文档名称(Getting the Opened Document Names)