PyInstaller Manual

PyInstaller 3.2

http://www.pyinstaller.org

pyinstaller@googlegroups.com

David Cortesi, based on structure by Giovanni Bajo & William Caban,
based on Gordon McMillan’s manual

This document has been placed in the public domain.

PyInstaller bundles a Python application and all its dependencies into a single package. The user can run the packaged app without installing a Python interpreter or any modules. *PyInstaller* supports Python 2.7 and Python 3.3+, and correctly bundles the major Python packages such as numpy, PyQt, Django, wxPython, and others.

PyInstaller is tested against Windows, Mac OS X, and Linux. However, it is not a cross-compiler: to make a Windows app you run *PyInstaller* in Windows; to make a Linux app you run it in Linux, etc. *PyInstaller* has been used successfully with AIX, Solaris, and FreeBSD, but is not tested against them.

What’s New This Release

Release 3.0 is a major rewrite that adds Python 3 support, better code quality through use of automated testing, and resolutions for many old issues.

Functional changes include removal of support for Python prior to 2.7, an easier way to include data files in the bundle (Adding Files to the Bundle), and changes to the “hook” API (Understanding PyInstaller Hooks).

Contents:

- Requirements
    - Windows
    - Mac OS X
    - Linux
    - AIX, Solaris, and FreeBSD
- License
- How To Contribute
- How to Install *PyInstaller*
    - Installing in Windows

  - Installing in Mac OS X
  - Installing from the archive
  - Verifying the installation
  - Installed commands
- What *PyInstaller* Does and How It Does It
  - Analysis: Finding the Files Your Program Needs
  - Bundling to One Folder
  - How the One-Folder Program Works
  - Bundling to One File
  - How the One-File Program Works
  - Using a Console Window
  - Hiding the Source Code
- Using PyInstaller
  - Options
  - Shortening the Command
  - Using UPX
  - Encrypting Python Bytecode
  - Supporting Multiple Platforms
  - Making Linux Apps Forward-Compatible
  - Capturing Windows Version Data
  - Building Mac OS X App Bundles
- Run-time Information
  - Using `__file__` and `sys._MEIPASS`
  - Using `sys.executable` and `sys.argv[0]`
- Using Spec Files
  - Spec File Operation
  - Adding Files to the Bundle
  - Giving Run-time Python Options
  - Spec File Options for a Mac OS X Bundle
  - Multipackage Bundles
  - Globals Available to the Spec File
- When Things Go Wrong
  - Recipes and Examples for Specific Problems
  - Finding out What Went Wrong
  - Helping PyInstaller Find Modules
  - Getting the Latest Version
  - Asking for Help
- Advanced Topics
  - The Bootstrap Process in Detail
  - The TOC and Tree Classes
  - Inspecting Archives
  - Inspecting Executables
  - Creating a Reproducible Build
- Understanding PyInstaller Hooks
  - How a Hook Is Loaded

- Building the Bootloader
  - Development tools
  - Building for Windows
  - Building for LINUX
- Changelog for PyInstaller
  - 3.2 (2016-05-03)
  - 3.1.1 (2016-01-31)
  - 3.1 (2016-01-09)
  - 3.0 (2015-10-04)
  - 2.1 (2013-09-27)
  - 2.0 (2012-08-08)
  - 1.5.1 (2011-08-01)
  - 1.5 (2011-05-05)
  - 1.4 (2010-03-22)
  - 1.3 (2006-12-20)
  - 1.2 (2006-06-29)
  - 1.1 (2006-02-13)
  - 1.0 (2005-09-19) with respect to McMillan's Python Installer 5b5
- Credits
  - Contributions to PyInstaller 3.2
  - Contributions to PyInstaller 3.1.1
  - Contributions to PyInstaller 3.1
  - Contributions to PyInstaller 3.0
  - Contributions to PyInstaller 2.1 and older
- Man Pages
  - pyinstaller
  - pyi-makespec

Indices and tables

- Index
- Module Index
- Search Page