

---

# Py2app (rewrite)

*Release 0.1*

Apr 03, 2018



---

## Contents

---

<b>1</b>	<b>Release information</b>	<b>3</b>
<b>2</b>	<b>Supported platforms</b>	<b>5</b>
<b>3</b>	<b>General documentation</b>	<b>7</b>
<b>4</b>	<b>Py2app-rewrite Developement</b>	<b>11</b>



The Py2app project provides tooling to create standalone application packages (“`.app`”) for Python programs.

This is a highly experimental and incomplete rewrite of [py2app](#), instead to end up with a clean and testable codebase. This rewrite also switches to a separate configuration file for controlling builds.

**Warning:** It is not possible to install `py2app-rewrite` and [py2app](#) at the same time, both use the python package name “`py2app`”.



# CHAPTER 1

---

## Release information

---

There are no releases of this project at this point in time.





## CHAPTER 2

---

### Supported platforms

---

Py2app will be supported on macOS 10.9 or later, as well as Python 3.6 or later.

Support for earlier versions of Python and macOS might be added later, the need.



## 3.1 Installing Py2app

### 3.1.1 Supported versions

I expect to support macOS 10.9 or later with this project, for 32-bit and 64-bit binaries (including fat binaries).

This will at initially support Python 3.6 and later, support for older versions will be added later when this can be done without cluttering source code (Python 2.7 may not be supported at ever).

### 3.1.2 Installation

The preferred way to install py2app will be using [pip](#):

```
$ pip install -U py2app
```

## 3.2 What's new in Py2app

An overview of the relevant changes in new, and older, releases.

### 3.2.1 Version 0.90

- To be done

## 3.3 Running py2app

The basic idea:

```
$ python -m py2app build
```

That is, the interface follows the common pattern of a main executable with subcommands (like `pip`, `git`, `Mercurial`, ...). For now I expect there to be one subcommand: “build”.

## 3.4 Configuration for py2app

This page will describe the configuration file for `py2app`.

Basic ideas:

- The file will be a “TOML” file, with a default name of “`py2app.toml`”

---

**Todo:** determine if TOML is the right choice, it appears to use too much quoting to be a nice language, YAML appears to be friendlier.

---

- The file contains arrays of tables describing application and plugin bundles
- There may be global configuration, which can be overridden
- Application bundles may contain an array table for plugins
- In the long run it might be interesting to try to generalize the format to be usable on other platforms (windows, linux flatpacks, ...)
- The format should support all features of macOS applications/bundles, and should have a gentle learning curve (“easy things should be easy, hard things should be possible”)

---

**Todo:** What about conditional “logic” (version dependent settings etc.)

---

### 3.4.1 Introduction

The configuration file for `py2app` is a [TOML](#) file containing information on what, and how to build. This document is a reference for that configuration, see *the to be written tutorial* for a gentler introduction to this file.

The default name for this file is “`py2app.toml`”, located in the root of the source tree for a project.

A basic example of a `py2app` configuration is included below:

```
[py2app]
build_style = "alias"

[[py2app.application]]
name = "Hello World"
entrypoint = "main.py"
```

This defines an “alias” build for an application named “Hello World” that has “`main.py`” as its entry point.

### 3.4.2 Specification

The configuration for `py2app` is stored in a [TOML](#) file with a table named “`py2app`”. All other data in the file is ignored by `py2app`.

This table contains global and shared configuration as well as arrays of tables for applications and bundles. The configuration should define at least one application or bundle.

Global configuration in the table `[py2app]`:

- *build\_style*: The type of bundle to build

Values values are:

- “standalone” (default): Create a fully standalone bundle that can be copied to other machines without further dependencies.
- “semi-standalone”: Create a standalone bundle that only relies on a Python framework. Can be run on all machines where a Python framework for the correct version of Python is installed.
- “alias”: Create a bundle that includes references to source code. These bundles cannot be moved and are primarily usefull during debugging.

Table arrays `[py2app.application]` and `[py2app.bundle]` are used to describe applications and plugin bundles and contain the following keys and tables:

- *entrypoint*: Name of the source file that is used to start the application.
- *name*: Name of the bundle, defaults to the basename of the entrypoint file.
- *suffix*: Name of the suffix for the bundle. This defaults to “.app” for application bundles, and “.bundle” for plugin bundles.

---

**Note:** The suffix should be changed for plugin bundles.

---



---

**Todo:** Document the suffix for common types of plugin bundles.

---

- *archs*: Either a single string value, or an array of strings. Values values are:

- “i386”: 32-bit Intel binary
- “x86\_64”: 64-bit Intel binary

This defaults to the architectures of the python binary.

- *deployment\_target*: The minimal macOS version where the output can be used on. This defaults to the version of macOS on the build machine.
- *iconfile*: Name of a icon file. This must be an existing file in the ICNS format. When no icon is provided py2app will use a generic icon.
- *build\_type*: Type of build. Valid values are:
  - “python”: a python program
  - “native”: use native (Objective-C) code to build this bundle. The bundle will not have a Python dependency.
- *native\_source*: List of Objective-C sources for this bundle.

For “python” builds these sources are used to specify a custom launcher (instead of the launcher provided by py2app).

- *native\_cflags*: Compiler flags to be used when building native sources
- *native\_ldflags*: Compiler flags to be used when linking the native binary

---

**Todo:** Add more options to specify information that will be placed in the Info.plist file (other than iconfile), to make the info table a poweruser tool.

---

The table `[py2app.application.info]` contains information to be included in the [Info.plist file](#) for the bundle. See Apple's document for the [Info.plist file](#) for information that can be included. The information in this table is copied as is into the [Info.plist file](#).

For application bundles the table array `[py2app.application.plugins]` can be used to add plugins into the application bundle. These bundles will be placed in the right location in the application bundle to be found by the system. The structure of these table arrays is the same as the toplevel `[py2app.plugins]` table array.

## CHAPTER 4

---

### Py2app-rewrite Developement

---

Py2app development is hosted at bickbucket, in particular at <<https://bitbucket.org/ronaldoussoren/py2app-rewrite/>>.

Important resources:

- [Issue tracker](#)
- [Mailing list for the PythonMac SIG](#)

A mailing list for anyone developing with Python on macOS.