

# pystrap - Project Specifications

Max Weise

October 22, 2023

## Contents

<b>1</b>	<b>File Specifications</b>	<b>1</b>
1.1	pyproject.toml File . . . . .	1
1.2	setup.py File . . . . .	2
1.3	package-init Files . . . . .	3

## 1 File Specifications

The following section shows how the config files will be created by the program. It will also contain a minimal example, which is used as a reference for developers.

### 1.1 pyproject.toml File

The `pyproject.toml` file is used to configure the project in a standardized format. We can define dependencies, metadata and even entry points for console scripts here. Many third party tools use the `pyproject.toml` file for configuration, that's why this file is the default output of the `pystrap` script. The following table 1 shows a list of all supported fields and the default values associated with them.

Field	Description	Type	Default Value
[project]	Nonoptional section-identifier. Define metadata of the project here.	Top-level section	-
name	The name of the project.	String	-
version	The version of the project. pystrap uses semantic versioning as default scheme	String	0.0.1
authors	List of authors of the project. An author consists of a name and an email adress.	List[String, String]	-
maintainers	List of maintainers of the project. pystrap will fill in the same data as in the authors field.	List[String, String]	-
requires-python	Specify the minimal version of python the application needs to run on.	String	$\geq 3.10$
[build-system]	Specifies the system used to build the python application.	Top-level section	-
requires	Define the dependencies for the build system	List[String, String]	["setuptools $\geq 42$ ", "wheel"]

Table 1: Supported data fields.

## 1.2 setup.py File

The `setup.py` file can be used as a way to define requirements and other metadata for a python project. As per PEP 621, the standardized way of storing metadata is the `pyproject.toml` file. `setup.py` is only needed, if the project should be distributed to PyPi. Because of this, the file created by the `pystrap` script is a relatively empty one and there are no extra user inputs needed to construct the file. Codelisting 1 shows the contents generated by the script.

Listing 1: The `setup.py` file

```

1  from setuptools import setup
2
3  if __name__ == '__main__':
4      setup()
```

### 1.3 package-init Files

To define package hierarchies python uses init files, which are allways called `__init__.py` and sit in the directory. While it is *technically* possible to write python code inside of these files that gets executed at import-time, there are currently no benefits in doing so. Pystrap will only create files in the apropriated positions, but will not insert any text or configuration inside of these files. If a user desires to do so, they can manipulate the generated file.