

---

# **friendly\_dist\_manager**

***Release 0.0.1***

**Kevin S. Phillips**

**Mar 23, 2021**



**CONTENTS:**

|          |                              |           |
|----------|------------------------------|-----------|
| <b>1</b> | <b>friendly_dist_manager</b> | <b>3</b>  |
| <b>2</b> | <b>Overview</b>              | <b>13</b> |
| <b>3</b> | <b>Indices and tables</b>    | <b>15</b> |
|          | <b>Python Module Index</b>   | <b>17</b> |
|          | <b>Index</b>                 | <b>19</b> |







## FRIENDLY\_DIST\_MANAGER

### 1.1 friendly\_dist\_manager package

#### 1.1.1 Subpackages

`friendly_dist_manager.package_formats` package

Subpackages

`friendly_dist_manager.package_formats.wheel` package

Submodules

`friendly_dist_manager.package_formats.wheel.metadata_file` module

Primitives for manipulating distutils metadata files

**class** `friendly_dist_manager.package_formats.wheel.metadata_file.ExtraRequirement` (*label*,  
*req*)

Bases: `tuple`

Create new instance of `ExtraRequirement(label, req)`

**property** `label`

Alias for field number 0

**property** `req`

Alias for field number 1

**class** `friendly_dist_manager.package_formats.wheel.metadata_file.MetadataFile` (*dist\_name*,  
*ver-*  
*sion*)

Bases: `object`

Abstraction around a distutils metadata file

## References

- (Latest Spec): <https://packaging.python.org/specifications/core-metadata/>
- (v2.1 Proposed PEP) <https://www.python.org/dev/peps/pep-0566/>
- (v1.2 Accepted PEP) <https://www.python.org/dev/peps/pep-0345/>
- (V1.1 Accepted PEP) <https://www.python.org/dev/peps/pep-0314/>
- (V1.0 Accepted PEP) <https://www.python.org/dev/peps/pep-0241/>

### Parameters

- **dist\_name** (*str*) – Name of the distribution being described
- **version** (*str*) – Version of the distribution being described

### **property authors**

authors of the project

Type *list (Person)*

### **property classifiers**

list of distribution classifiers

Type *list (str)*

### **property distribution\_name**

name of the distribution being built

Type *str*

### **property distribution\_version**

version of the package being built

Type *str*

### **property download\_url**

URL where the distribution package can be downloaded

Type *str*

### **property extra\_requirements**

list of optional requirements that users of the distribution may select to enable additional features

Type *list (ExtraRequirement)*

### **property file\_version**

the metadata file version / schema version used by this file

Type *str*

### **property homepage**

URL of the project homepage

Type *str*

### **property keywords**

labels users can search for when looking for distributions like this one

Type *list (str)*

### **property license**

text describing the licensing terms for the project

Type `str`

**property maintainers**  
maintainers of the project

Type `list (Person)`

**property project\_urls**  
Support URLs associated with the distribution

Type `list (ProjectURL)`

**property python\_requirements**  
Python version identifiers describing the supported Python runtime versions supported by this distribution package

Type `list(str)`

**property raw**  
the raw text content of the metadata file

Type `str`

**property requirements**  
package definitions describing other distributions this one depends on

Type `list(str)`

**property summary**  
description of the package

Type `str`

**class** `friendly_dist_manager.package_formats.wheel.metadata_file.Person` (*name*,  
*email*)

Bases: `tuple`

Create new instance of `Person(name, email)`

**property email**  
Alias for field number 1

**property name**  
Alias for field number 0

**class** `friendly_dist_manager.package_formats.wheel.metadata_file.ProjectURL` (*label*,  
*url*)

Bases: `tuple`

Create new instance of `ProjectURL(label, url)`

**property label**  
Alias for field number 0

**property url**  
Alias for field number 1

## friendly\_dist\_manager.package\_formats.wheel.wheel\_file module

Primitives for manipulating Python wheel files

**class** friendly\_dist\_manager.package\_formats.wheel.wheel\_file.**WheelFile** (*dist\_name*,  
*version*)

Bases: `object`

Abstraction around a Python wheel file

### References

- (Latest Spec) <https://packaging.python.org/specifications/binary-distribution-format/>
- (Accepted PEP) <https://www.python.org/dev/peps/pep-0427/>
- (Proposed PEP) <https://www.python.org/dev/peps/pep-0491/>

#### Parameters

- **dist\_name** (*str*) – name of the distribution package being built
- **version** (*str*) – version number associated with the new distribution package

**add\_file** (*src\_file*, *target\_path*)

Adds a new file to the package

#### Parameters

- **src\_file** (*pathlib.Path*) – reference to the source file to be packaged
- **target\_path** (*pathlib.Path*) – location, relative to the root of the wheel file, where this new file should be deployed

**build** (*output\_path*)

Constructs a wheel file from the metadata stored in this class

**Parameters** **output\_path** (*pathlib.Path*) – folder where the wheel file should be generated

**Returns** Reference to the newly generated wheel file

**Return type** *pathlib.Path*

**property filename**

gets the fully qualified file name of the wheel file to be generated

**Type** *str*

**classmethod from\_pyproject** (*pyproject\_config*)

Factory method that instantiates an instance of this class from data stored in a pyproject.toml configuration file

**Parameters** **pyproject\_config** (*PyProjectParser*) – config file for a PEP518 compliant Python package

**Returns** Instance of the class pre-initialized from data provided by the given config file

**Return type** *WheelFile*

**property metadata**

properties describing this distribution package

Type *MetadataFile*

## Module contents

Primitives for manipulating Python wheel files

## Module contents

This sub-package contains modules for managing various package formats supported by the build tool

### friendly\_dist\_manager.pyproject package

#### Submodules

#### friendly\_dist\_manager.pyproject.build\_system\_table module

Primitives for manipulating the ‘build-system’ table in a pyproject.toml file

**class** friendly\_dist\_manager.pyproject.build\_system\_table.**BuildSystemTable**(*data*)  
Bases: *object*

Abstraction around the ‘build-system’ table from a pyproject.toml file

#### References

- (Add support custom build backends) <https://www.python.org/dev/peps/pep-0518/>
- (Original Proposal) <https://www.python.org/dev/peps/pep-0517/>

**Parameters** *data* (*dict*) – TOML data parsed from a pyproject.toml config file

#### **property backend**

module name for the build backend to use for this project

**Type** *str*

#### **property requirements**

list of packages required to run the build system backend

**Type** *list (str)*

#### friendly\_dist\_manager.pyproject.parser module

Primitives for operating on pyproject.toml files

**class** friendly\_dist\_manager.pyproject.parser.**PyProjectParser**(*toml\_data*)  
Bases: *object*

**Interface for parsing and manipulating data stored in a TOML formatted** configuration file

Each section, or table, within the TOML configuration file is defined by it’s own independent PEP standard. As such this primary entry point defers the handling of specific tables to supporting classes.

## References

- (TOML Spec) <https://toml.io/en/>

**Parameters** `toml_data` (*str*) – Raw text loaded from a TOML formatted configuration file

### **property** `build_system`

returns the ‘build-system’ table from the config file

**Type** *BuildSystemTable*

### **classmethod** `from_file` (*file\_path*)

Factory method used to instantiate instances of this class from a disk-based file

**Parameters** `file_path` (*pathlib.Path*) – path to the TOML file to load

**Returns** reference to the class instance created

**Return type** *PyProjectParser*

### **property** `project`

returns the ‘project’ table from the config file

**Type** *ProjectTable*

## friendly\_dist\_manager.pyproject.project\_table module

Primitives for manipulating the ‘project’ table in a pyproject.toml file

**class** `friendly_dist_manager.pyproject.project_table.Entrypoint` (*name, ref*)

Bases: *tuple*

Create new instance of Entrypoint(name, ref)

### **property** `name`

Alias for field number 0

### **property** `ref`

Alias for field number 1

**class** `friendly_dist_manager.pyproject.project_table.Person` (*name, email*)

Bases: *tuple*

Create new instance of Person(name, email)

### **property** `email`

Alias for field number 1

### **property** `name`

Alias for field number 0

**class** `friendly_dist_manager.pyproject.project_table.ProjectTable` (*data*)

Bases: *object*

Abstraction around the ‘project’ table from a pyproject.toml file

## References

- <https://www.python.org/dev/peps/pep-0621/>

**Parameters** **data** (*dict*) – TOML data parsed from a pyproject.toml config file

### **property authors**

list of people who are considered ‘authors’ of the project

Note: each person may have a name and/or an email address, but either one may be omitted

**Type** *list(Person)*

### **property classifiers**

trove classifiers describing properties of the project

<https://pypi.org/classifiers/>

**Type** *list(str)*

### **property console\_scripts**

list of entry points for console / shell scripts exposed by the project

**Type** *list(Entrypoint)*

### **property dependencies**

list of package dependencies associated with this project

## References

- <https://www.python.org/dev/peps/pep-0508/>

**Type** *list(str)*

### **property description**

descriptive name summarizing the purpose of the distribution

**Type** *str*

### **property entrypoint\_identifiers**

list of custom entrypoint identifiers associated with the project

**Type** *list(str)*

### **get\_entrypoint** (*entrypoint\_id*)

Gets definition for a custom entrypoint associated with the project

**Parameters** **entrypoint\_id** (*str*) – ID of the entrypoint to retrieve. See *entrypoint\_identifiers()* for supported values

**Returns** list of entrypoints associated with the entrypoint identifier

**Return type** *list(Entrypoint)*

### **get\_optional\_dependencies** (*dependency\_id*)

Set of dependencies associated with a specific group ID

## References

- <https://www.python.org/dev/peps/pep-0508/>

**Parameters** `dependency_id` (*str*) – identifier for a specific group of optional dependencies associated with the project. See `optional_dependency_identifiers()` for list of valid options

**Returns** list of dependency definitions for the optional dependencies associated with the specified ID

**Return type** `list` (*str*)

**property** `gui_scripts`

list of application entry points for GUI based projects

**Type** `list` (*Entrypoint*)

**property** `keywords`

descriptive keywords used when searching for project on pypi

**Type** `list` (*str*)

**property** `license`

text explaining the licensing details associated with the project

**Type** `str`

**property** `maintainers`

list of people who are considered ‘maintainers’ of the project

Note: each person may have a name and/or an email address, but either one may be omitted

**Type** `list` (*Person*)

**property** `name`

name of the distribution being built

**Type** `str`

**property** `optional_dependency_identifiers`

list of IDs associated with groups of optional package dependencies associated with the project

**Type** `list` (*str*)

**property** `python_requirement`

the Python runtime version identifier indicating the range of versions supported by this project

**Type** `str`

**property** `readme`

text formatted data loaded from the mentioned readme file associated with the project

**Type** `str`

**property** `readme_file`

reference to the readme file associated with the project, or None if no readme file provided

**Type** `pathlib.Path`

**property** `urls`

URLs providing additional information about the distribution package

**Type** `list` (*ProjectURL*)

**property version**

version of the distribution being built

**Type** `str`**class** `friendly_dist_manager.pyproject.project_table.ProjectURL` (*label*, *url*)Bases: `tuple`Create new instance of ProjectURL(*label*, *url*)**property label**

Alias for field number 0

**property url**

Alias for field number 1

## Module contents

Primitives for manipulating the TOML based pyproject configuration file

### 1.1.2 Submodules

#### friendly\_dist\_manager.hooks module

PIP hooks for triggering build operations

**class** `friendly_dist_manager.hooks.PEP517`Bases: `object`

Hooks related to the PEP517 standard

<https://www.python.org/dev/peps/pep-0517/>**static build\_wheel** (*wheel\_directory*, *\_config\_settings=None*, *\_metadata\_directory=None*)

Hook triggered when user has requested a Python Wheel to be generated

<https://www.python.org/dev/peps/pep-0517/#build-wheel>**Parameters**

- **wheel\_directory** (*str*) – Path where the generated wheel file should be placed
- **\_config\_settings** (*dict*) – optional settings provided by the caller to customize the behavior of the wheel file creation
- **\_metadata\_directory** (*str*) – Path to folder where metadata describing the contents of the wheel file can be found

**Returns** Path to the generated wheel file**Return type** `str`

**friendly\_dist\_manager.version module**

### **1.1.3 Module contents**

User friendly Python distribution package manager

## OVERVIEW

User friendly Python distribution package manager

Comments, suggestions and bugs may be reported to the project [maintainer](#)



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### f

- friendly\_dist\_manager, [12](#)
- friendly\_dist\_manager.hooks, [11](#)
- friendly\_dist\_manager.package\_formats,  
    [7](#)
- friendly\_dist\_manager.package\_formats.wheel,  
    [7](#)
- friendly\_dist\_manager.package\_formats.wheel.metadata\_file,  
    [3](#)
- friendly\_dist\_manager.package\_formats.wheel.wheel\_file,  
    [6](#)
- friendly\_dist\_manager.pyproject, [11](#)
- friendly\_dist\_manager.pyproject.build\_system\_table,  
    [7](#)
- friendly\_dist\_manager.pyproject.parser,  
    [7](#)
- friendly\_dist\_manager.pyproject.project\_table,  
    [8](#)
- friendly\_dist\_manager.version, [12](#)



## INDEX

### A

`add_file()` (`friendly_dist_manager.package_formats.wheel.wheel_file.WheelFile`  
*method*), 6  
`authors()` (`friendly_dist_manager.package_formats.wheel.metadata_file.MetadataFile`  
*property*), 4  
`authors()` (`friendly_dist_manager.pyproject.project_table.ProjectTable`  
*property*), 9

### B

`backend()` (`friendly_dist_manager.pyproject.build_system_table.BuildSystemTable`  
*property*), 7  
`build()` (`friendly_dist_manager.package_formats.wheel.wheel_file.WheelFile`  
*method*), 6  
`build_system()` (`friendly_dist_manager.pyproject.parser.PyProjectParser`  
*property*), 8  
`build_wheel()` (`friendly_dist_manager.hooks.PEP517`  
*static method*), 11  
`BuildSystemTable` (class in `ExtraRequirement` (class in  
`friendly_dist_manager.pyproject.build_system_table`), `friendly_dist_manager.package_formats.wheel.metadata_file`),  
7 3

### C

`classifiers()` (`friendly_dist_manager.package_formats.wheel.metadata_file.MetadataFile`  
*property*), 4  
`classifiers()` (`friendly_dist_manager.pyproject.project_table.ProjectTable`  
*property*), 9  
`console_scripts()`  
(`friendly_dist_manager.pyproject.project_table.ProjectTable`  
*property*), 9

### D

`dependencies()` (`friendly_dist_manager.pyproject.project_table.ProjectTable`  
*property*), 9  
`description()` (`friendly_dist_manager.pyproject.project_table.ProjectTable`  
*property*), 9  
`distribution_name()`  
(`friendly_dist_manager.package_formats.wheel.metadata_file.MetadataFile`  
*property*), 4  
`distribution_version()`  
(`friendly_dist_manager.package_formats.wheel.metadata_file.MetadataFile`  
*property*), 4

`download_url()` (`friendly_dist_manager.package_formats.wheel.metadata_file.MetadataFile`  
*property*), 4  
`email()` (`friendly_dist_manager.package_formats.wheel.metadata_file.MetadataFile`  
*property*), 5  
`email()` (`friendly_dist_manager.pyproject.project_table.Person`  
*property*), 8

### E

`Entrypoint` (class in `friendly_dist_manager.pyproject.project_table`),  
8  
`entrypoint_identifiers()`  
(`friendly_dist_manager.pyproject.project_table.ProjectTable`  
*property*), 9  
`extra_requirements()`  
(`friendly_dist_manager.package_formats.wheel.metadata_file.MetadataFile`  
*property*), 4

### F

`file_version()` (`friendly_dist_manager.package_formats.wheel.metadata_file.MetadataFile`  
*property*), 4  
`filename()` (`friendly_dist_manager.package_formats.wheel.wheel_file.WheelFile`  
*property*), 6  
`friendly_dist_manager`  
module, 12  
`friendly_dist_manager.hooks`  
module, 11  
`friendly_dist_manager.package_formats`  
module, 7  
`friendly_dist_manager.package_formats.wheel`  
module, 7  
`friendly_dist_manager.package_formats.wheel.metadata_file`  
module, 3  
`friendly_dist_manager.package_formats.wheel.wheel_file`  
module, 6  
`friendly_dist_manager.pyproject`  
module, 11  
`friendly_dist_manager.pyproject.build_system_table`  
module, 7

friendly\_dist\_manager.pyproject.parser module, 7

friendly\_dist\_manager.pyproject.project\_table module, 8

friendly\_dist\_manager.version module, 12

from\_file() (friendly\_dist\_manager.pyproject.parser.PyProjectParser class method), 8

from\_pyproject() (friendly\_dist\_manager.package\_formats.wheel\_wheel\_file.WheelFile class method), 6

## G

get\_entrypoint() (friendly\_dist\_manager.pyproject.project\_table.ProjectTable method), 9

get\_optional\_dependencies() (friendly\_dist\_manager.pyproject.project\_table.ProjectTable method), 9

gui\_scripts() (friendly\_dist\_manager.pyproject.project\_table.ProjectTable property), 10

## H

homepage() (friendly\_dist\_manager.package\_formats.wheel.metadata\_file.MetadataFile property), 4

## K

keywords() (friendly\_dist\_manager.package\_formats.wheel.metadata\_file.MetadataFile property), 4

keywords() (friendly\_dist\_manager.pyproject.project\_table.ProjectTable property), 10

## L

label() (friendly\_dist\_manager.package\_formats.wheel.metadata\_file.ExtraRequirement property), 3

label() (friendly\_dist\_manager.package\_formats.wheel.metadata\_file.ProjectURL property), 5

label() (friendly\_dist\_manager.pyproject.project\_table.ProjectURL property), 11

license() (friendly\_dist\_manager.package\_formats.wheel.metadata\_file.MetadataFile property), 4

license() (friendly\_dist\_manager.pyproject.project\_table.ProjectTable property), 10

## M

maintainers() (friendly\_dist\_manager.package\_formats.wheel.metadata\_file.MetadataFile property), 5

maintainers() (friendly\_dist\_manager.pyproject.project\_table.ProjectTable property), 10

metadata() (friendly\_dist\_manager.package\_formats.wheel\_wheel\_file.WheelFile property), 6

MetadataFile (class in friendly\_dist\_manager.package\_formats.wheel.metadata\_file), 3

module

friendly\_dist\_manager, 12

friendly\_dist\_manager.hooks, 11

friendly\_dist\_manager.package\_formats, 7

friendly\_dist\_manager.package\_formats.wheel, 7

friendly\_dist\_manager.package\_formats.wheel.metadata\_file, 3

friendly\_dist\_manager.package\_formats.wheel\_wheel\_file.WheelFile, 6

friendly\_dist\_manager.pyproject, 11

friendly\_dist\_manager.pyproject.build\_system\_table, 7

friendly\_dist\_manager.pyproject.parser, 7

friendly\_dist\_manager.pyproject.project\_table, 8

friendly\_dist\_manager.version, 12

## N

name() (friendly\_dist\_manager.package\_formats.wheel.metadata\_file.MetadataFile property), 5

name() (friendly\_dist\_manager.pyproject.project\_table.Entrypoint property), 8

name() (friendly\_dist\_manager.pyproject.project\_table.Person property), 8

name() (friendly\_dist\_manager.pyproject.project\_table.ProjectTable property), 10

## O

optional\_dependency\_identifiers() (friendly\_dist\_manager.pyproject.project\_table.ProjectTable property), 10

## P

PEP517 (class in friendly\_dist\_manager.hooks), 11

Person (class in friendly\_dist\_manager.package\_formats.wheel.metadata\_file), 5

Person (class in friendly\_dist\_manager.pyproject.project\_table), 8

project() (friendly\_dist\_manager.pyproject.parser.PyProjectParser property), 8

project\_urls() (friendly\_dist\_manager.package\_formats.wheel.metadata\_file.MetadataFile property), 5

ProjectTable (class in friendly\_dist\_manager.pyproject.project\_table), 8

ProjectURL (class in friendly\_dist\_manager.pyproject.project\_table), 11

ProjectURL (class in friendly\_dist\_manager.pyproject.project\_table), 11

WheelFile (class in friendly\_dist\_manager.package\_formats.wheel.metadata\_file), 5

WheelFile (class in friendly\_dist\_manager.pyproject.project\_table), 11

```

PyProjectParser          (class          in
    friendly_dist_manager.pyproject.parser),
7
python_requirement()
    (friendly_dist_manager.pyproject.project_table.ProjectTable
    property), 10
python_requirements()
    (friendly_dist_manager.package_formats.wheel.metadata_file.MetadataFile
    property), 5

```

## R

```

raw() (friendly_dist_manager.package_formats.wheel.metadata_file.MetadataFile
    property), 5
readme() (friendly_dist_manager.pyproject.project_table.ProjectTable
    property), 10
readme_file() (friendly_dist_manager.pyproject.project_table.ProjectTable
    property), 10
ref() (friendly_dist_manager.pyproject.project_table.EntryPoint
    property), 8
req() (friendly_dist_manager.package_formats.wheel.metadata_file.ExtraRequirement
    property), 3
requirements() (friendly_dist_manager.package_formats.wheel.metadata_file.MetadataFile
    property), 5
requirements() (friendly_dist_manager.pyproject.build_system_table.BuildSystemTable
    property), 7

```

## S

```

summary() (friendly_dist_manager.package_formats.wheel.metadata_file.MetadataFile
    property), 5

```

## U

```

url() (friendly_dist_manager.package_formats.wheel.metadata_file.ProjectURL
    property), 5
url() (friendly_dist_manager.pyproject.project_table.ProjectURL
    property), 11
urls() (friendly_dist_manager.pyproject.project_table.ProjectTable
    property), 10

```

## V

```

version() (friendly_dist_manager.pyproject.project_table.ProjectTable
    property), 10

```

## W

```

WheelFile                (class          in
    friendly_dist_manager.package_formats.wheel.wheel_file),
6

```