
whey-conda

Release 0.2.0

Whey extension for creating Conda packages for Python projects.

Dominic Davis-Foster

Apr 26, 2024

Contents

1	Installation	1
1.1	from PyPI	1
1.2	from Anaconda	1
1.3	from GitHub	1
2	Configuration	3
2.1	[tool.whey-conda]	3
3	Public API	5
3.1	CondaBuilder	5
3.2	whey_conda.config	7
4	Downloading source code	9
4.1	Building from source	10
5	License	11
	Python Module Index	13
	Index	15

Installation

1.1 from PyPI

```
$ python3 -m pip install whey-conda --user
```

1.2 from Anaconda

First add the required channels

```
$ conda config --add channels https://conda.anaconda.org/conda-forge  
$ conda config --add channels https://conda.anaconda.org/domdfcoding
```

Then install

```
$ conda install whey-conda
```

1.3 from GitHub

```
$ python3 -m pip install git+https://github.com/repo-helper/whey-conda@master --user
```


Configuration

wh^{ey}-conda is configured in the `pyproject.toml` file defined in [PEP 517](#) and [PEP 518](#).

See also: The [wh^{ey} documentation](#) contains instructions for configuring wh^{ey} itself.

To enable wh^{ey}-conda, add the following lines to your `pyproject.toml` file:

```
[tool.whey.builders]
binary = "whey_conda"
```

The wh^{ey}-conda-specific configuration is defined in the `tool.whey-conda` table. `conda-channels` and `conda-extras` can instead be defined in the `tool.mkrecipe` table if you also use `mkrecipe`.

2.1 [tool.wh^{ey}-conda]

All keys are optional.

conda-description

Type: [String](#)

The description of the package.

You can use a single `%s` in the description, which will be substituted with the value of the [description](#) key from `pyproject.toml`.

The default value is `'%s'`.

Example:

```
[tool.whey-conda]
conda-description = "Fantastic Spam!"
```

conda-channels

Type: [Array of strings](#)

A list of required conda channels to build and use the package.

The default value is `[]`.

Example:

```
[tool.whey-conda]
conda-channels = [ "domdfcoding", "conda-forge", "bioconda", ]
```

conda-extras

Type: Array of strings *or* the strings 'all' or 'none'.

A list of extras (see [optional-dependencies](#)) to include as requirements in the Conda package.

- The special keyword 'all' indicates all extras should be included.
- The special keyword 'none' indicates no extras should be included.

The default value is 'none'.

Examples:

```
[tool.whey-conda]
conda-extras = [ "test", "doc", ]

[tool.whey-conda]
conda-extras = "all"
```


Public API

Whey extension for creating Conda packages for Python projects.

Classes:

<code>CondaBuilder(project_dir, config[, ...])</code>	Builds Conda packages using metadata read from <code>pyproject.toml</code> .
---	--

```
class CondaBuilder (project_dir, config, build_dir=None, out_dir=None, *, verbose=False,
                    colour=None)
```

Bases: `WheelBuilder`

Builds Conda packages using metadata read from `pyproject.toml`.

Parameters

- **project_dir** (`PathPlus`) – The project to build the distribution for.
- **build_dir** (`Union[str, Path, PathLike, None]`) – The (temporary) build directory. Default `<project_dir>/build/wheel`.
- **out_dir** (`Union[str, Path, PathLike, None]`) – The output directory. Default `<project_dir>/dist`.
- **verbose** (`bool`) – Enable verbose output. Default `False`.

Methods:

<code>build()</code>	Build the Conda distribution.
<code>build_conda()</code>	Build the Conda distribution.
<code>create_conda_archive(wheel_contents_dir[, ...])</code>	Create the conda archive.
<code>get_runtime_requirements()</code>	Returns a list of the project's runtime requirements.
<code>write_conda_about()</code>	Write the conda <code>about.json</code> file.
<code>write_conda_index([build_number])</code>	Write the conda <code>index.json</code> file.
<code>write_license(dest_dir[, dest_filename])</code>	Write the <code>LICENSE</code> file.

Attributes:

<code>default_build_dir</code>	Provides a default for the <code>build_dir</code> argument.
<code>info_dir</code>	The <code>info</code> directory in the build directory for Conda builds.

build()

Build the Conda distribution.

Return type `str`

Returns The filename of the created archive.

build_conda()

Build the Conda distribution.

Return type `str`

Returns The filename of the created archive.

create_conda_archive (*wheel_contents_dir*, *build_number=1*)

Create the conda archive.

Parameters

- **wheel_contents_dir** (`Union[str, Path, PathLike]`) – The directory containing the installed contents of the wheel.
- **build_number** (`int`) – Default 1.

Return type `str`

Returns The filename of the created archive.

property default_build_dir

Provides a default for the `build_dir` argument.

Return type `PathPlus`

get_runtime_requirements()

Returns a list of the project's runtime requirements.

Return type `List[ComparableRequirement]`

property info_dir

The `info` directory in the build directory for Conda builds.

Return type `PathPlus`

write_conda_about()

Write the conda about `.json` file.

See also: <https://docs.conda.io/projects/conda-build/en/latest/resources/package-spec.html#info-about-json>

write_conda_index (*build_number=1*)

Write the conda index `.json` file.

See also: <https://docs.conda.io/projects/conda-build/en/latest/resources/package-spec.html#info-index-json>

Parameters **build_number** (`int`) – Default 1.

write_license (*dest_dir*, *dest_filename*='LICENSE')

Write the LICENSE file.

Parameters

- **dest_dir** (*PathPlus*) – The directory to write the file into.
- **dest_filename** (*str*) – The name of the file to write in *dest_dir*. Default 'LICENSE'.

3.2 whey_conda.config

Configuration for whey-conda.

Classes:

<i>WheyCondaParser</i> ()	Parser for the [tool.whey-conda] table from pyproject.toml.
---------------------------	---

class WheyCondaParser

Bases: *AbstractConfigParser*

Parser for the [tool.whey-conda] table from pyproject.toml.

Attributes:

<i>keys</i>	The keys to parse from the TOML file.
-------------	---------------------------------------

Methods:

<i>parse</i> (<i>config</i> [, <i>set_defaults</i>])	Parse the TOML configuration.
<i>parse_conda_channels</i> (<i>config</i>)	Parse the conda-channels key, giving a list of required conda channels to build and use the package.
<i>parse_conda_description</i> (<i>config</i>)	Parse the conda-description key, giving the description of the package.
<i>parse_conda_extras</i> (<i>config</i>)	Parse the conda-extras key, giving a list of extras (see optional-dependencies) to include as requirements in the Conda package.

property keys

The keys to parse from the TOML file.

Return type *List[str]*

parse (*config*, *set_defaults*=False)

Parse the TOML configuration.

Parameters

- **config** (*Dict[str, Any]*) – The unparsed TOML config for the [tool.whey-conda] table.
- **set_defaults** (*bool*) – If *True*, the values in *self.defaults* and *self.factories* will be set as defaults for the returned mapping. Default *False*.

Return type *Dict[str, Any]*

parse_conda_channels (*config*)

Parse the `conda-channels` key, giving a list of required conda channels to build and use the package.

The default value is `[]`.

Example:

```
[tool.whey-conda]
conda-channels = [
    "domdfcoding",
    "conda-forge",
    "bioconda",
]
```

Parameters `config` (`Dict[str, Any]`) – The unparsed TOML config for the `[tool.whey-conda]` table.

Return type `List[str]`

parse_conda_description (*config*)

Parse the `conda-description` key, giving the description of the package.

You can use a single `%s` in the description, which will be substituted with the value of the `description` key from `pyproject.toml`.

The default value is `'%s'`.

Example:

```
[tool.whey-conda]
conda-description = "Fantastic Spam!"
```

Parameters `config` (`Dict[str, Any]`) – The unparsed TOML config for the `[tool.whey-conda]` table.

Return type `str`

parse_conda_extras (*config*)

Parse the `conda-extras` key, giving a list of extras (see [optional-dependencies](#)) to include as requirements in the Conda package.

- The special keyword `'all'` indicates all extras should be included.
- The special keyword `'none'` indicates no extras should be included.

The default value is `'none'`.

Examples:

```
[tool.whey-conda]
conda-extras = [ "test", "doc", ]

[tool.whey-conda]
conda-extras = "all"
```

Return type `Union[Literal['all'], Literal['none'], List[str]]`

Downloading source code

The whey-conda source code is available on GitHub, and can be accessed from the following URL: <https://github.com/repo-helper/whey-conda>

If you have git installed, you can clone the repository with the following command:

```
$ git clone https://github.com/repo-helper/whey-conda
```

```
Cloning into 'whey-conda'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 173 (delta 16), reused 17 (delta 6), pack-reused 126
Receiving objects: 100% (173/173), 126.56 KiB | 678.00 KiB/s, done.
Resolving deltas: 100% (66/66), done.
```

Alternatively, the code can be downloaded in a 'zip' file by clicking:

Clone or download -> Download Zip

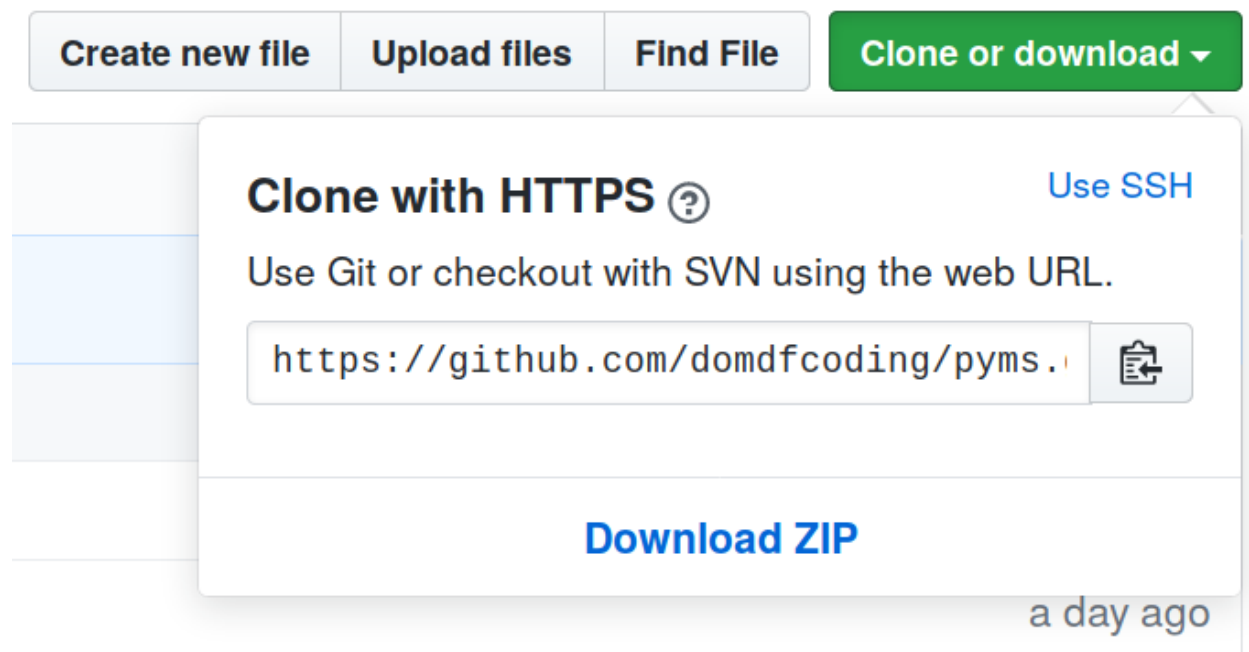


Fig. 1: Downloading a 'zip' file of the source code

4.1 Building from source

The recommended way to build `whey-conda` is to use `tox`:

```
$ tox -e build
```

The source and wheel distributions will be in the directory `dist`.

If you wish, you may also use `pep517.build` or another **PEP 517**-compatible build tool.

License

whely-conda is licensed under the [MIT License](#)

A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Permissions

- Commercial use – The licensed material and derivatives may be used for commercial purposes.
- Modification – The licensed material may be modified.
- Distribution – The licensed material may be distributed.
- Private use – The licensed material may be used and modified in private.

Conditions

- License and copyright notice – A copy of the license and copyright notice must be included with the licensed material.

Limitations

- Liability – This license includes a limitation of liability.
- Warranty – This license explicitly states that it does NOT provide any warranty.

[See more information on choosealicense.com](#) ⇒

```
Copyright (c) 2021 Dominic Davis-Foster
```

```
Permission is hereby granted, free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished to do so, subject to the following conditions:
```

```
The above copyright notice and this permission notice shall be included in all  
copies or substantial portions of the Software.
```

```
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,  
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR  
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE  
OR OTHER DEALINGS IN THE SOFTWARE.
```


Python Module Index

W

`whey_conda`, [5](#)

`whey_conda.config`, [7](#)

B

`build()` (*CondaBuilder method*), 5
`build_conda()` (*CondaBuilder method*), 6

C

`conda-channels`
 configuration value, 3
`conda-description`
 configuration value, 3
`conda-extras`
 configuration value, 3
`CondaBuilder` (*class in whey_conda*), 5
 configuration value
 `conda-channels`, 3
 `conda-description`, 3
 `conda-extras`, 3
`create_conda_archive()` (*CondaBuilder method*), 6

D

`default_build_dir()` (*CondaBuilder property*), 6

G

`get_runtime_requirements()` (*CondaBuilder method*), 6

I

`info_dir()` (*CondaBuilder property*), 6

K

`keys()` (*WheyCondaParser property*), 7

M

MIT License, 11
 module
 `whey_conda`, 5
 `whey_conda.config`, 7

P

`parse()` (*WheyCondaParser method*), 7
`parse_conda_channels()` (*WheyCondaParser method*), 8

`parse_conda_description()`
 (*WheyCondaParser method*), 8
`parse_conda_extras()` (*WheyCondaParser method*), 8
 Python Enhancement Proposals
 PEP 517, 3, 10
 PEP 518, 3
 PEP 621#dependencies-optional-dependencies, 4, 7, 8
 PEP 621#description, 3, 8

T

TOML: Array, 3, 4
 TOML: String, 3, 4

W

`whey_conda`
 module, 5
`whey_conda.config`
 module, 7
`WheyCondaParser` (*class in whey_conda.config*), 7
`write_conda_about()` (*CondaBuilder method*), 6
`write_conda_index()` (*CondaBuilder method*), 6
`write_license()` (*CondaBuilder method*), 6