

---

# **HuaweiCrawler Documentation**

*Release unknown*

**Quan Pan**

**Oct 07, 2019**



---

## Contents

---

<b>1</b>	<b>Contents</b>	<b>3</b>
1.1	License	3
1.2	Contributors	6
1.3	Contributing	6
1.3.1	Code of Conduct	6
1.3.2	Rights	6
1.3.3	Issue Conventions	6
1.3.4	Design Principles	7
1.3.5	Dataset Objects	7
1.3.6	Git Conventions	7
1.3.7	Code Conventions	7
1.3.8	Development Environment	7
1.3.8.1	Initial Setup	8
1.3.8.2	Installing GDAL	8
1.3.8.3	Python build requirements	8
1.3.8.4	Installing Rasterio	8
1.3.8.5	Running the tests	9
1.4	Changelog	9
1.4.1	Version 0.1	9
1.5	HuaweiCrawler	9
1.5.1	HuaweiCrawler package	9
1.5.1.1	Subpackages	9
1.5.1.2	Submodules	10
1.5.1.3	HuaweiCrawler.skeleton module	10
1.5.1.4	Module contents	11
<b>2</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>
	<b>Index</b>	<b>17</b>



This is the documentation of **HuaweiCrawler**.

From the root of the project, run:

```
python setup.py --version
```

Read the Docs, run:

```
python setup.py doctest
```

```
python setup.py docs
```

Unit test, run:

```
python setup.py test
```

PyPI upload, run `setup.py`:

```
1. Commit -> Git - tag - add - v0.0.1 -> ``setup.py`` -> push
2. Github - Release - new release v0.0.1

python setup.py sdist bdist_wheel
twine upload dist/*
```

---

**Note:** This is the main page of your project's [Sphinx](#) documentation. It is formatted in [reStructuredText](#). Add additional pages by creating `rst`-files in `docs` and adding them to the [toctree](#) below. Use then [references](#) in order to link them from this page, e.g. [Contributors](#) and [Changelog](#).

It is also possible to refer to the documentation of other Python packages with the [Python domain syntax](#). By default you can reference the documentation of [Sphinx](#), [Python](#), [NumPy](#), [SciPy](#), [matplotlib](#), [Pandas](#), [Scikit-Learn](#). You can add more by extending the `intersphinx_mapping` in your Sphinx's `conf.py`.

The pretty useful extension [autodoc](#) is activated by default and lets you include documentation from docstrings. Docstrings can be written in [Google style](#) (recommended!), [NumPy style](#) and [classical style](#).

---



### 1.1 License

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

“License” shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

“Licensor” shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

“Legal Entity” shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, “control” means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

“You” (or “Your”) shall mean an individual or Legal Entity exercising permissions granted by this License.

“Source” form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

“Object” form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

“Work” shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

“Derivative Works” shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

“Contribution” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as “Not a Contribution.”

“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a “NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying



the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets “[ ]” replaced with your own identifying information. (Don’t include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same “printed page” as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 1.2 Contributors

- Quan Pan <quanpan302@hotmail.com>

## 1.3 Contributing

Welcome to the Rasterio project. Here’s how we work.

### 1.3.1 Code of Conduct

First of all: the Rasterio project has a code of conduct. Please read the `CODE_OF_CONDUCT.txt` file, it’s important to all of us.

### 1.3.2 Rights

The BSD license (see `LICENSE.txt`) applies to all contributions.

### 1.3.3 Issue Conventions

The Rasterio issue tracker is for actionable issues.

Questions about installation, distribution, and usage should be taken to the project’s [general discussion group](#). Opened issues which fall into one of these three categories may be perfunctorily closed.

Questions about development of Rasterio, brainstorming, requests for comment, and not-yet-actionable proposals are welcome in the project’s [developers discussion group](#). Issues opened in Rasterio’s GitHub repo which haven’t been socialized there may be perfunctorily closed.

Rasterio is a relatively new project and highly active. We have bugs, both known and unknown.

Please search existing issues, open and closed, before creating a new one.

Rasterio employs C extension modules, so bug reports very often hinge on the following details:

- Operating system type and version (Windows? Ubuntu 12.04? 14.04?)
- The version and source of Rasterio (PyPI, Anaconda, or somewhere else?)
- The version and source of GDAL (UbuntuGIS? Homebrew?)

Please provide these details as well as tracebacks and relevant logs. When using the `$ rio` CLI logging can be enabled with `$ rio -v` and verbosity can be increased with `-vvv`. Short scripts and datasets demonstrating the issue are especially helpful!

### 1.3.4 Design Principles

Rasterio's API is different from GDAL's API and this is intentional.

- Rasterio is a library for reading and writing raster datasets. Rasterio uses GDAL but is not a “Python binding for GDAL.”
- Rasterio always prefers Python's built-in protocols and types or Numpy protocols and types over concepts from GDAL's data model.
- Rasterio keeps I/O separate from other operations. `rasterio.open()` is the only library function that operates on filenames and URIs. `dataset.read()`, `dataset.write()`, and their mask counterparts are the methods that perform I/O.
- Rasterio methods and functions should be free of side-effects and hidden inputs. This is challenging in practice because GDAL embraces global variables.

### 1.3.5 Dataset Objects

Our term for the kind of object that allows read and write access to raster data is *dataset object*. A dataset object might be an instance of *DatasetReader* or *DatasetWriter*. The canonical way to create a dataset object is by using the `rasterio.open()` function.

This is analogous to Python's use of [file object](#).

### 1.3.6 Git Conventions

We use a variant of centralized workflow described in the [Git Book](#). We have no 1.0 release for Rasterio yet and we are tagging and releasing from the master branch. Our post-1.0 workflow is to be decided.

Work on features in a new branch of the mapbox/rasterio repo or in a branch on a fork. Create a [GitHub pull request](#) when the changes are ready for review. We recommend creating a pull request as early as possible to give other developers a heads up and to provide an opportunity for valuable early feedback.

### 1.3.7 Code Conventions

The `rasterio` namespace contains both Python and C extension modules. All C extension modules are written using [Cython](#). The Cython language is a superset of Python. Cython files end with `.pyx` and `.pxd` and are where we keep all the code that calls GDAL's C functions.

Rasterio supports Python 2 and Python 3 in the same code base, which is aided by an internal compatibility module named `compat.py`. It functions similarly to the more widely known [six](#) but we only use a small portion of the features so it eliminates a dependency.

We strongly prefer code adhering to [PEP8](#).

Tests are mandatory for new features. We use [pytest](#).

We aspire to 100% coverage for Python modules but coverage of the Cython code is a future aspiration ([#515](#)).

### 1.3.8 Development Environment

Developing Rasterio requires Python 2.7 or any final release after and including 3.4. We prefer developing with the most recent version of Python but recognize this is not possible for all contributors. A C compiler is also required to leverage [existing protocols](#) for extending Python with C or C++. See the Windows install instructions in the [readme](#) for more information about building on Windows.

### 1.3.8.1 Initial Setup

First, clone Rasterio's git repo:

```
$ git clone https://github.com/mapbox/rasterio
```

Development should occur within a [virtual environment](#) to better isolate development work from custom environments.

In some cases installing a library with an accompanying executable inside a virtual environment causes the shell to initially look outside the environment for the executable. If this occurs try deactivating and reactivating the environment.

### 1.3.8.2 Installing GDAL

The GDAL library and its headers are required to build Rasterio. We do not have currently have guidance for any platforms other than Linux and OS X.

On Linux, GDAL and its headers should be available through your distro's package manager. For Ubuntu the commands are:

```
$ sudo add-apt-repository ppa:ubuntugis/ppa
$ sudo apt-get update
$ sudo apt-get install gdal-bin libgdal-dev
```

On OS X, Homebrew is a reliable way to get GDAL.

```
$ brew install gdal
```

### 1.3.8.3 Python build requirements

Provision a virtualenv with Rasterio's build requirements. Rasterio's `setup.py` script will not run unless Cython and Numpy are installed, so do this first from the Rasterio repo directory.

Linux users may need to install some additional Numpy dependencies:

```
$ sudo apt-get install libatlas-dev libatlas-base-dev gfortran
```

then:

```
$ pip install -U pip
$ pip install -r requirements-dev.txt
```

### 1.3.8.4 Installing Rasterio

Rasterio, its Cython extensions, normal dependencies, and dev dependencies can be installed with `$ pip`. Installing Rasterio in editable mode while developing is very convenient but only affects the Python files. Specifying the `[test]` extra in the command below tells `$ pip` to also install Rasterio's dev dependencies.

```
$ pip install -e .[test]
```

Any time a Cython (`.pyx` or `.pxd`) file is edited the extension modules need to be recompiled, which is most easily achieved with:

```
$ pip install -e .
```

When switching between Python versions the extension modules must be recompiled, which can be forced with `$ touch rasterio/*.pyx` and then re-installing with the command above. If this is not done an error claiming that an object has the wrong size, try recompiling is raised.

The dependencies required to build the docs can be installed with:

```
$ pip install -e .[docs]
```

### 1.3.8.5 Running the tests

Rasterio's tests live in `tests` <tests/> and generally match the main package layout.

To run the entire suite and the code coverage report:

```
$ py.test --cov rasterio --cov-report term-missing
```

A single test file:

```
$ py.test tests/test_band.py
```

A single test:

```
$ py.test tests/test_band.py::test_band
```

## 1.4 Changelog

### 1.4.1 Version 0.1

- Feature A added
- FIX: nasty bug #1729 fixed
- add your changes here!

## 1.5 HuaweiCrawler

### 1.5.1 HuaweiCrawler package

#### 1.5.1.1 Subpackages

**HuaweiCrawler.core package**

**Submodules**

**HuaweiCrawler.core.core module**

core

**example** In the Docker Image `quanpan302/huawei-crawler`:

```
scrapy startproject tutorial
```

```
scrapy runspider /notebooks/src/HuaweiCrawler/core/core.py -o mobile.csv -t csv
```

```
class HuaweiCrawler.core.core.TmobileSpider(name=None, **kwargs)
    Bases: scrapy.spiders.Spider

    fieldnames = ['url', 'name']

    file_name = <_io.TextIOWrapper name='tmobile_spider.csv' mode='w' encoding='UTF-8'>
    name = 'tmobile_spider'

    parse(response)

    parse_detail(response)

    start_urls = ['https://www.t-mobile.nl/shop/alle-telefoons?ch=es&cc=con&sc=acq']
    writer = <csv.DictWriter object>
```

## Module contents

core

### 1.5.1.2 Submodules

#### 1.5.1.3 HuaweiCrawler.skeleton module

This is a skeleton file that can serve as a starting point for a Python console script. To run this script uncomment the following lines in the `[options.entry_points]` section in `setup.cfg`:

```
console_scripts = fibonacci = HuaweiCrawler.skeleton:run
```

Then run `python setup.py install` which will install the command `fibonacci` inside your current environment. Besides console scripts, the header (i.e. until `_logger...`) of this file can also be used as HuaweiCrawler for Python modules.

Note: This skeleton file can be safely removed if not needed!

`HuaweiCrawler.skeleton.fib(n)`

Fibonacci example function

**Parameters** `n` (`int`) – integer

**Returns** `n`-th Fibonacci number

**Return type** `int`

`HuaweiCrawler.skeleton.main(args)`

Main entry point allowing external calls

**Parameters** `args` (`[str]`) – command line parameter list

`HuaweiCrawler.skeleton.parse_args(args)`

Parse command line parameters

**Parameters** `args` (`[str]`) – command line parameters as list of strings

**Returns** command line parameters namespace

**Return type** `argparse.Namespace`

`HuaweiCrawler.skeleton.run()`

Entry point for console\_scripts

`HuaweiCrawler.skeleton.setup_logging(loglevel)`

Setup basic logging

**Parameters** `loglevel` (*int*) – minimum loglevel for emitting messages

#### 1.5.1.4 Module contents

HuaweiCrawler





## CHAPTER 2

---

### Indices and tables

---

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



### h

`HuaweiCrawler`, [11](#)

`HuaweiCrawler.core`, [10](#)

`HuaweiCrawler.core.core`, [9](#)

`HuaweiCrawler.skeleton`, [10](#)



## F

`fib()` (in module *HuaweiCrawler.skeleton*), 10  
`fieldnames` (*HuaweiCrawler.core.core.TmobileSpider* attribute), 10  
`file_name` (*HuaweiCrawler.core.core.TmobileSpider* attribute), 10

## H

*HuaweiCrawler* (module), 11  
*HuaweiCrawler.core* (module), 10  
*HuaweiCrawler.core.core* (module), 9  
*HuaweiCrawler.skeleton* (module), 10

## M

`main()` (in module *HuaweiCrawler.skeleton*), 10

## N

`name` (*HuaweiCrawler.core.core.TmobileSpider* attribute), 10

## P

`parse()` (*HuaweiCrawler.core.core.TmobileSpider* method), 10  
`parse_args()` (in module *HuaweiCrawler.skeleton*), 10  
`parse_detail()` (*HuaweiCrawler.core.core.TmobileSpider* method), 10

## R

`run()` (in module *HuaweiCrawler.skeleton*), 10

## S

`setup_logging()` (in module *HuaweiCrawler.skeleton*), 11  
`start_urls` (*HuaweiCrawler.core.core.TmobileSpider* attribute), 10

## T

*TmobileSpider* (class in *HuaweiCrawler.core.core*), 10

## W

`writer` (*HuaweiCrawler.core.core.TmobileSpider* attribute), 10