
dottorrent Documentation

Release 1.9.2

Kevin Zhang

Jul 22, 2017

Contents

1	Features	3
1.1	Installation	3
1.1.1	Requirements	3
1.2	Library	4
1.2.1	Example Usage	5
1.3	Command-line tool	5
1.4	dottorrent-gui	6
1.5	Changelog	6
1.5.1	1.9.2	6
1.5.2	1.9.1	6
1.5.3	1.9.0	6
1.5.4	1.8.0	6
1.5.5	1.7.0	7
1.5.6	1.6.0	7
1.5.7	1.5.3	7
1.5.8	1.5.2	7
1.5.9	1.5.1	7
1.5.10	1.5.0	7
1.5.11	1.4.3	7
1.5.12	1.4.2	7
1.5.13	1.4.1	7
1.5.14	1.4.0	8
1.5.15	1.3.0	8
1.5.16	1.2.2	8
1.5.17	1.2.0	8
1.5.18	1.1.1	8
1.5.19	1.1.0	8
1.5.20	1.0.2	8
1.5.21	1.0.1	8
1.5.22	1.0.0	8
2	Indices and tables	9

dottorrent is a high-level Python 3 library and command-line interface (CLI) tool for creating .torrent files.

CHAPTER 1

Features

- Fast (capable of several hundred MB/s)
- Full Unicode support
- Automatic and manual piece size selection
- HTTP/web seeds support (BEP 19)
- Private flag support (BEP 27)
- Source string support
- Info hash generation for created torrents
- User-definable comment field, creation date, creator, etc.
- Filename pattern exclusion
- Per-file MD5 hash inclusion

Contents:

Installation

Requirements

- Python 3.3+ (Python 3.5+ recommended)
- See `requirements.txt` for additional dependencies.

Stable releases are available on PyPI and can be installed using `pip`.

```
pip install dottorrent
```

To install from the latest development sources, clone the git repo and run `pip install .`

Library

The dottorrent library can be used in a Python script or program to create .torrent files.

```
class dottorrent.Torrent (path, trackers=None, web_seeds=None, piece_size=None, private=False,
                          source=None, creation_date=None, comment=None, created_by=None, include_md5=False, exclude=None)
```

Parameters

- **path** – path to a file or directory from which to create the torrent
- **trackers** – list/iterable of tracker URLs
- **web_seeds** – list/iterable of HTTP/FTP seed URLs
- **piece_size** – Piece size in bytes. Must be ≥ 16 KB and a power of 2. If None, `get_info()` will be used to automatically select a piece size.
- **private** – The private flag. If True, DHT/PEX will be disabled.
- **source** – An optional source string for the torrent.
- **exclude** – A list of filename patterns that should be excluded from the torrent.
- **creation_date** – An optional datetime object representing the torrent creation date.
- **comment** – An optional comment string for the torrent.
- **created_by** – name/version of the program used to create the .torrent. If None, defaults to the value of `DEFAULT_CREATOR`.
- **include_md5** – If True, also computes and stores MD5 hashes for each file.

dump()

Returns The bencoded torrent data as a byte string.

Note: `generate()` must be called first.

generate (*callback=None*)

Computes and stores piece data. Returns `True` on success, `False` otherwise.

Parameters **callback** – progress/cancellation callable with method signature (`filename, pieces_completed, pieces_total`). Useful for reporting progress if dottorrent is used in a GUI/threaded context, and if torrent generation needs to be cancelled. The callable's return value should evaluate to `True` to trigger cancellation.

get_info()

Scans the input path and automatically determines the optimal piece size based on ~1500 pieces (up to `MAX_PIECE_SIZE`) along with other basic info, including total size (in bytes), the total number of files, piece size (in bytes), and resulting number of pieces. If `piece_size` has already been set, the custom value will be used instead.

Returns (`total_size, total_files, piece_size, num_pieces`)

info_hash

Returns The SHA-1 info hash of the torrent. Useful for generating magnet links.

Note: `generate()` must be called first.

info_hash_base32

Returns the base32 info hash of the torrent. Useful for generating magnet links.

Note: `generate()` must be called first.

piece_size**save** (*fp*)

Saves the torrent to *fp*, a file(-like) object opened in binary writing (*wb*) mode.

Note: `generate()` must be called first.

trackers**web_seeds**

Example Usage

```
from dottorrent import Torrent

t = Torrent('/my/data/', trackers=['http://tracker.openbittorrent.com:80/announce'])
t.generate()
with open('mydata.torrent', 'wb') as f:
    t.save(f)
```

Command-line tool

If you installed dottorrent using pip, dottorrent should be available in your system path.

```
usage: dottorrent [-h] [--tracker TRACKER] [--web_seed WEB_SEED]
                 [--piece_size PIECE_SIZE] [--private] [--source SOURCE]
                 [--exclude RE] [--comment COMMENT] [--date DATE] [--md5]
                 [--verbose]
                 path output_path

Create a .torrent file

positional arguments:
path                  path to file/directory to create torrent from
output_path          Output path for created .torrent file. If a directory
                    is provided, the filename will be
↳ automatically      generated based on the input.

optional arguments:
-h, --help            show this help message and exit
--tracker TRACKER, -t TRACKER
                    tracker URL (can be specified
↳ multiple times)
--web_seed WEB_SEED, -w WEB_SEED
                    web seed URL (can be specified
↳ multiple times)
```

```
--piece_size PIECE_SIZE, -s PIECE_SIZE           piece size, e.g. 16KB, 1M. Leave
↳ unspecified for                               automatic piece size
--private, -p          set private flag (useful for private trackers)
--source SOURCE        source string (useful for private trackers)
--exclude RE, -x RE    filename patterns that should be excluded (can be
                        specified multiple times)
--comment COMMENT, -c COMMENT                    string for the torrent comment field
--date DATE, -d DATE  Torrent creation date. Valid values: unix
                        timestamp/none/now (default: now)
--md5                  Add per-file MD5 hashes
--verbose, -v          verbose mode

dottorrent/1.9.2 (https://github.com/kz26/dottorrent)
```

When creating a torrent, all dotfiles (filenames beginning with a ‘.’) are excluded. On Windows systems running Python 3.5+, all hidden files are excluded as well.

To add multiple trackers, web seeds and/or exclusion patterns, repeat the `-t`, `-w`, or `-x` as many times as necessary, e.g. `-t tracker1 -t tracker2` or `-x "*.jpg" --exclude "*.png"`.

dottorrent-gui

An official GUI for dottorrent is available. Please see <https://github.com/kz26/dottorrent-gui> for more information.

Changelog

1.9.2

- Always set announce key, even for multi-tracker torrents (<https://github.com/kz26/dottorrent-gui/issues/15>)

1.9.1

- Fix unreferenced piece size error in CLI

1.9.0

- Increase `MAX_PIECE_SIZE` to 64 MiB
- Minor docstring improvements

1.8.0

- Human-friendly piece size specification in CLI
- Minor string formatting improvements

1.7.0

- Implement filename pattern exclusion from PR #7 (`--exclude`, `-x`)

1.6.0

- Add support for source strings (`--source` option)
- Exclude hidden dotfiles from being added
- Exclude hidden files on Windows (requires Python 3.5+)
- Refactor - CLI tool is now simply *dottorrent*

1.5.3

- Use relative paths instead of absolute paths for directory mode path generation
- Add invalid input path check in `get_info()` and `generate()`

1.5.2

- Use `humanfriendly.format_size` in binary mode
- Pin major versions of dependencies in `requirements.txt` and `setup.py`

1.5.1

- Fix filename clobbering when filename contains parent path string (issue #2)

1.5.0

- Allow both file and directory names to be specified for `output_path` (PR #1, @mahkitah)

1.4.3

- `dottorrent_cli`: Change `output_path` to `output_file` for clarity

1.4.2

- Rename `dottorrent_cli.py` to `dottorrent_cli`

1.4.1

- Ignore empty files in `get_info()` and `generate()`

1.4.0

- Use custom exceptions (see exceptions.py)
- Add check for empty input

1.3.0

- Unicode support

1.2.2

- Fix auto piece sizing in generate() with small files

1.2.0

- Switch to BEP 19 (GetRight style) web seeds

1.1.1

- Return True/False in generate()

1.1.0

- Move include_md5 option to Torrent constructor
- Add cancellation functionality to generate function via existing callback
- get_info() now always scans/rescans the input path instead of checking for cached data
- Documentation improvements

1.0.2

- Raise exception when path is a directory and no files exist

1.0.1

- Change bencoder.pyx minimum version dependency to 1.1.1
- Add none/now to CLI date option
- Minor tweaks

1.0.0

- Initial release.

CHAPTER 2

Indices and tables

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

D

`dump()` (`dottorrent.Torrent` method), 4

G

`generate()` (`dottorrent.Torrent` method), 4

`get_info()` (`dottorrent.Torrent` method), 4

I

`info_hash` (`dottorrent.Torrent` attribute), 4

`info_hash_base32` (`dottorrent.Torrent` attribute), 4

P

`piece_size` (`dottorrent.Torrent` attribute), 5

S

`save()` (`dottorrent.Torrent` method), 5

T

`Torrent` (class in `dottorrent`), 4

`trackers` (`dottorrent.Torrent` attribute), 5

W

`web_seeds` (`dottorrent.Torrent` attribute), 5