
py3createtorrent Documentation

Release 1.0.1

Robert Nitsch

Jan 05, 2022

Contents

1	py3createtorrent	3
1.1	Features	3
1.2	Motivation	3
1.3	Requirements	4
1.4	Installation	4
1.5	Usage	4
1.6	Examples	9
1.7	Configuration	12
2	Changelog	15
2.1	Version 1.0.1	15
2.2	Version 1.0.0	15
2.3	Version 0.9.7	17
2.4	Version 0.9.6	17
2.5	Version 0.9.5	17
2.6	Version 0.9.4	17
2.7	Version 0.9.3	17
2.8	Version 0.9.2	18
2.9	Version 0.9.1	18
2.10	Version 0.9	18
2.11	Version 0.8	18
3	Developer documentation	19
3.1	Todo	19
3.2	Future	19
4	Indices and tables	21

Contents:

Create torrents via command line!

py3createtorrent is a comprehensive shell/commandline utility for creating torrents (Linux & Windows).

1.1 Features

Some of the features:

- you can create **huge torrents** for any amount of data
- you can add a **comment** to the torrent file
- you can create **private torrents** (disabled DHT, ...)
- you can create torrents with **multiple trackers**
- you can create **trackerless torrents**
- you can add **webseeds** to torrents
- you can **exclude specific files/folders**
- you can exclude files/folders based on **regular expressions**
- you can specify **custom piece sizes**
- you can specify custom creation dates

1.2 Motivation

There already is rTorrent, but unfortunately it does not support creating torrents. Thus, it is often a pain to seed torrents from your servers directly.

py3createtorrent is intended to fill this gap.

1.3 Requirements

py3createtorrent requires at least Python 3.5 and the `bencode.py` module.

Note: It may be possible to use the script with older Python versions. For Python 3.2, 3.3, 3.4 you need to install the backport of Python's `typing` module: <https://pypi.org/project/typing/>. For Python 3.1 you need to additionally install the backport of Python's `argparse` module: <https://pypi.org/project/argparse/>.

This has not been tested, though. Feedback is welcome.

1.4 Installation

1.4.1 Recommended: Installation using pip

pip is the easiest and recommended way of installing py3createtorrent.

Just execute:

```
pip3 install py3createtorrent
```

After that, you can use py3createtorrent on your commandline.

1.4.2 Alternative: Manual installation

Alternatively, you can download py3createtorrent manually. Download the desired version from here: <https://github.com/rsnitsch/py3createtorrent/releases>

And then install the dependencies:

```
pip install bencode.py
```

Then you can execute py3createtorrent.py (it is in the `src` folder).

Alternatively, use pipenv:

```
pipenv install
```

Then you can execute py3createtorrent.py with pipenv as follows:

```
pipenv run src/py3createtorrent.py
```

1.5 Usage

Syntax:

```
usage: py3createtorrent.py [-h] [-p PIECE_LENGTH] [-P] [-c COMMENT] [-s SOURCE] [-f]   
→ [-v] [-q] [-o PATH] [-e PATH] [--exclude-pattern REGEXP] [--exclude-pattern-ci   
→ REGEXP] [-d TIMESTAMP] [-n NAME] [--md5] [--config CONFIG]   
[-t TRACKER_URL] [--node HOST,PORT] [--webseed WEBSEED_URL]   
path
```

(continues on next page)

(continued from previous page)

```

py3createtorrent is a comprehensive command line utility for creating torrents.

positional arguments:
  path                file or folder for which to create a torrent

optional arguments:
  -h, --help          show this help message and exit
  -p PIECE_LENGTH, --piece-length PIECE_LENGTH
                    piece size in KiB. 0 = automatic selection (default).
  -P, --private       create private torrent
  -c COMMENT, --comment COMMENT
                    include comment
  -s SOURCE, --source SOURCE
                    include source
  -f, --force         do not ask anything, just do it
  -v, --verbose       verbose mode
  -q, --quiet         be quiet, e.g. don't print summary
  -o PATH, --output PATH
                    custom output location (directory or complete path). default ↵
↵= current directory.
  -e PATH, --exclude PATH
                    exclude path (can be repeated)
  --exclude-pattern REGEXP
                    exclude paths matching the regular expression (can be ↵
↵repeated)
  --exclude-pattern-ci REGEXP
                    exclude paths matching the case-insensitive regular ↵
↵expression (can be repeated)
  -d TIMESTAMP, --date TIMESTAMP
                    set creation date (unix timestamp). -1 = now (default). -2 = ↵
↵disable.
  -n NAME, --name NAME
                    use this file (or directory) name instead of the real one
  --md5               include MD5 hashes in torrent file
  --config CONFIG     use another config file instead of the default one from the ↵
↵home directory
  -t TRACKER_URL, --tracker TRACKER_URL
                    tracker to use for the torrent
  --node HOST,PORT    DHT bootstrap node to use for the torrent
  --webseed WEBSEED_URL
                    webseed URL for the torrent

```

1.5.1 Specifying trackers (-t, --tracker)

One or multiple tracker URLs can be specified using the `-t` or `--tracker` switch. Single tracker example:

```
py3createtorrent -t udp://tracker.opentrackr.org:1337/announce my_data_folder/
```

This is equivalent to the short form using the *tracker abbreviation* for opentrackr.org:

```
py3createtorrent -t opentrackr my_data_folder/
```

For multiple trackers, just use `-t` repeatedly. Multiple tracker example:

```
py3createtorrent -t udp://tracker.opentrackr.org:1337/announce -t udp://tracker.
↪coppersurfer.tk:6969/announce -t udp://tracker.cyberia.is:6969/announce my_data_
↪folder/
```

This is equivalent to the short form using the tracker abbreviations:

```
py3createtorrent -t opentrackr -t coppersurfer -t cyberia my_data_folder/
```

bestN: Automatically add the best trackers

You can use `bestN` to add the best N trackers from <https://github.com/ngosang/trackerslist>. This requires internet access, obviously.

For example:

```
py3createtorrent -t best5 my_data_folder/
```

Trackerless torrents

You can create a trackerless torrent by not specifying any tracker URLs at all (i.e. don't use the `-t` switch at all).

1.5.2 Specifying DHT bootstrap nodes (`--node`)

One or multiple DHT bootstrap nodes can be specified using the `--node` switch. Each bootstrap node must be specified in the form `host,port`. Just like `-t`, the `--node` switch can be used repeatedly in order to specify multiple DHT bootstrap nodes.

Example:

```
py3createtorrent --node router.bittorrent.com,8991 --node second.node.com,1337 my_
↪data_folder/
```

It is recommended to specify some DHT bootstrap nodes for trackerless torrents.

1.5.3 Piece size (`-p`)

This switch allows you to specify a custom piece size. The piece size should be chosen with care, because it affects the following properties:

- size of the `.torrent` file
- network overhead
- cost of fixing corrupted pieces
- time it takes until peers start sharing data

Note: Unless you know what you're doing, please let `py3createtorrent` automatically determine the best piece size for you.

Background

In general, the files for which a .torrent is created are sliced up in **pieces**.

For each piece, a 20-byte checksum (based on SHA-1, the Secure Hash Algorithm 1) is calculated and stored inside the .torrent file - this, by the way, is the time-consuming part of torrent creation. Therefore, the piece size strongly correlates with the size of the created .torrent file: The larger the pieces, the smaller the number of pieces for which a checksum must be stored (and vice versa).

The piece size also affects the **network overhead** involved in the peer-2-peer communication for a torrent. The peers regularly exchange information records that specify the pieces that each peer has finished downloading so that they know where they can get certain pieces from. The greater the number of pieces, the larger these information records need to be and thus the greater the overhead will tend to be.

Moreover, corrupted pieces need to be redownloaded. Of course, large pieces are more expensive to redownload (both in terms of time and traffic).

Finally, the piece size also affects the time it takes until peers start to share data with each other (only pieces that have been downloaded completely can be shared with other peers). Therefore, if the piece size is large, it will take longer for any peer to finish downloading a piece and to be able to share this piece with other peers.

1.5.4 Private torrents (-P)

Private torrents force the BitTorrent clients to only use the specified trackers for discovering other peers. Advanced peer discovery methods like DHT or peer list exchange are effectively disabled.

1.5.5 Comment (-c)

The comment is a short text stored in the .torrent file and displayed by most BitTorrent clients in the torrent info.

By default py3createtorrent uses “created by py3createtorrent <version>” as comment (to change this behavior, consult the *Configuration* section).

1.5.6 Source (-s)

The source field is a non-standard metainfo field used by private trackers to reduce issues (such as misreported stats) caused by cross-seeding. For private trackers that forbid their torrent files from being uploaded elsewhere, it ensures that torrent files uploaded to the tracker from a different source are unique to the private tracker.

New in 0.9.7.

1.5.7 Force (-f)

Force makes py3createtorrent e.g. overwrite existing .torrent files without asking for your permission.

1.5.8 Verbose (-v)

Verbose mode makes py3createtorrent report about the individual steps it is undertaking while creating the .torrent file.

This is particularly useful for debugging purposes.

1.5.9 Quiet (`-q`)

py3createtorrent will try to stay completely silent on the commandline.

1.5.10 Output path (`-o`)

The output path is either the directory in which the .torrent file should be saved or the complete path to the destination .torrent file. In the former case, the name of the .torrent file is deduced from the input's name (i.e. the input directory's or file's name), unless this name is explicitly overwritten (using the `-n` switch). (In the latter case, the name of the .torrent file is itself specified by the output path.)

By default, py3createtorrent uses the current working directory as the output directory.

1.5.11 Exclude path (`-e`)

This allows for the exclusion of specific files or directories.

The switch may be used repeatedly to exclude multiple files/directories.

On Windows, this is case-insensitive.

1.5.12 Exclude pattern (`--exclude-pattern`, `--exclude-pattern-ci`)

This allows for the exclusion of files or directories that match a certain pattern (regular expression).

The switches may be used repeatedly to specify multiple exclusion patterns.

New in version 0.9.5: The `--exclude-pattern-ci` variant (case-insensitive). On Windows, the `--exclude-pattern` has been made case-sensitive (previously it was case-insensitive on Windows and case-sensitive on UNIX etc.).

1.5.13 Creation date (`-d`)

This switch allows you to overwrite the creation date saved in the .torrent file. You can fake any creation date you like.

The creation date is specified as [UNIX timestamp](#).

You can disable storing a creation date altogether by providing a timestamp of `-2`.

1.5.14 Name (`-n`)

This setting overwrites the file or directory name stored inside the .torrent file. **Thus it affects the file or directory name that will be presented to downloaders as the real name of the data.** You can use it to avoid renaming your input data.

Unless a destination .torrent file is explicitly specified (using the `-o` switch), this name will also be used to deduce the name of the resulting .torrent file.

Note: The name switch is an advanced feature that most users probably don't need. Therefore, please refrain from using this feature, unless you really know what you're doing.

For most intents and purposes, the `-o` switch is probably more suitable.

1.5.15 MD5 hashes (`--md5`)

As of py3createtorrent 0.9.5 the calculation of MD5 hashes must be explicitly requested, because it significantly slows down the torrent creation process (and makes the torrent file a little larger, although this is probably negligible).

New in 0.9.5.

1.5.16 Path to config (`--config`)

By default, py3createtorrent tries to load the config file `.py3createtorrent.cfg` from the user's home directory. To use another config file, specify the path with `--config`. Use `--verbose` for troubleshooting this, if it does not work as expected.

New in 1.0.0.

1.6 Examples

Assume there is a folder “example” with the following contents:

```
example/
  subfolder/
    10_more_minutes_please.JPG
    image.rar
  anotherimage.jpg
  image.zip
```

Assume, we're currently inside the parent directory.

1.6.1 Example 1 - from directory, no options, default behaviour

Command:

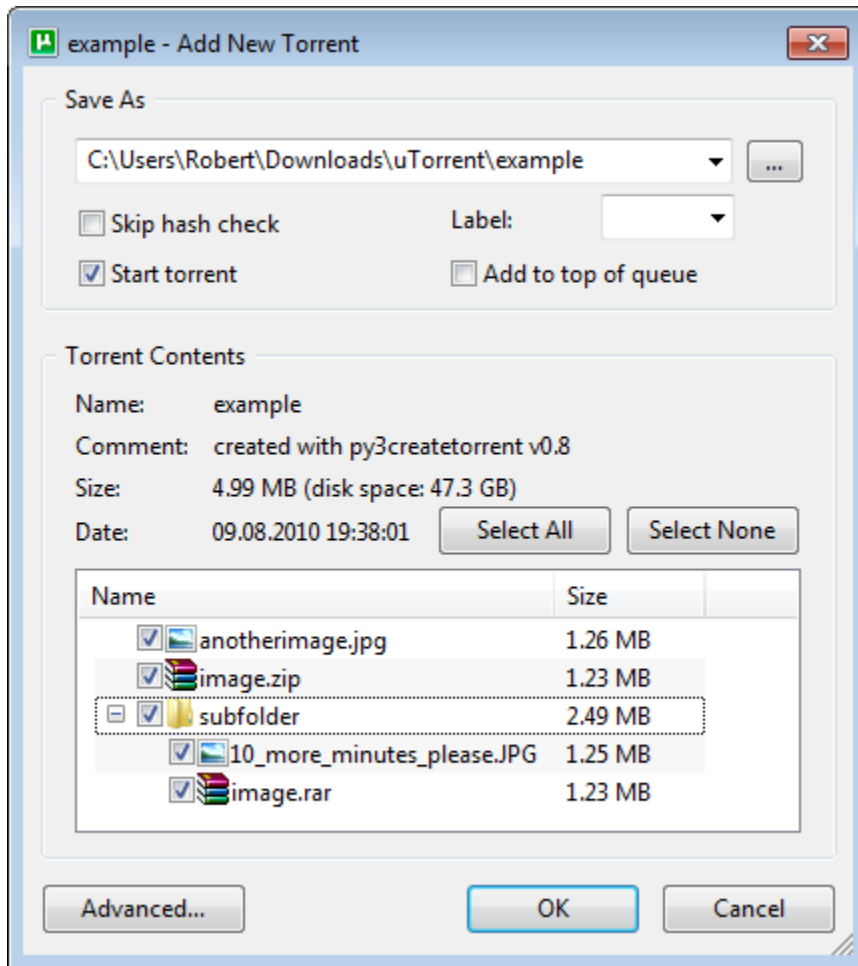
```
C:\Users\Robert\Desktop\Python\createtorrent>py3createtorrent example -t udp://
↪tracker.opentrackr.org:1337/announce
```

Alternative, equivalent command using a tracker abbreviation for convenience:

```
C:\Users\Robert\Desktop\Python\createtorrent>py3createtorrent example -t opentrackr
```

Effect: Creates `example.torrent` inside the current directory.

In μ Torrent it will look like this:



Note: Please note: If you do not specify a comment yourself using the `-c / --comment` option, py3createtorrent will advertise itself through the comment field, as you can see in the screenshot (Torrent Contents -> Comment: *created with py3createtorrent v0.8*).

To change this behavior, consult the [Configuration](#) section.

1.6.2 Example 2 - from directory, excluding subfolders

Command:

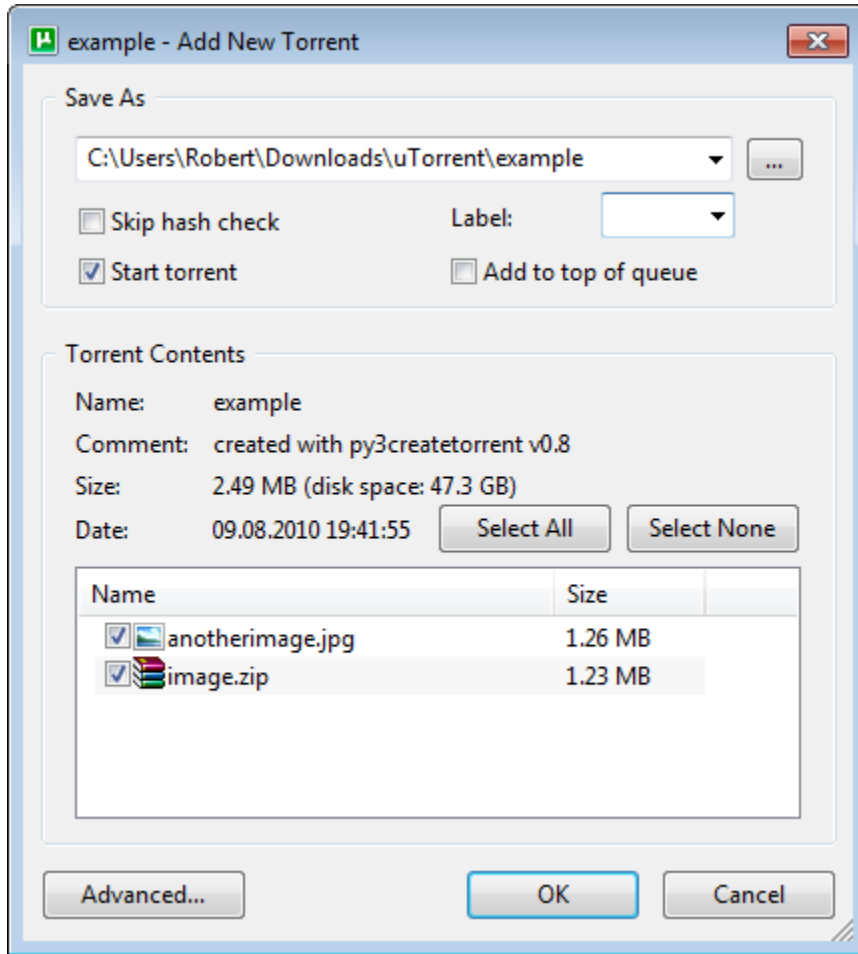
```
C:\Users\Robert\Desktop\Python\createtorrent>py3createtorrent -e example\subfolder_
↪example -t udp://tracker.opentrackr.org:1337/announce
```

Effect: Creates example.torrent inside the current directory. examplesubfolder has been excluded.

Tip: Of course you can exclude multiple subfolders, e.g.:

```
py3createtorrent -e exclusion1 -e exclusion2 yourfolder -t tracker-url
```

In µTorrent it will look like this:



1.6.3 Example 3 - from directory, excluding files

Command:

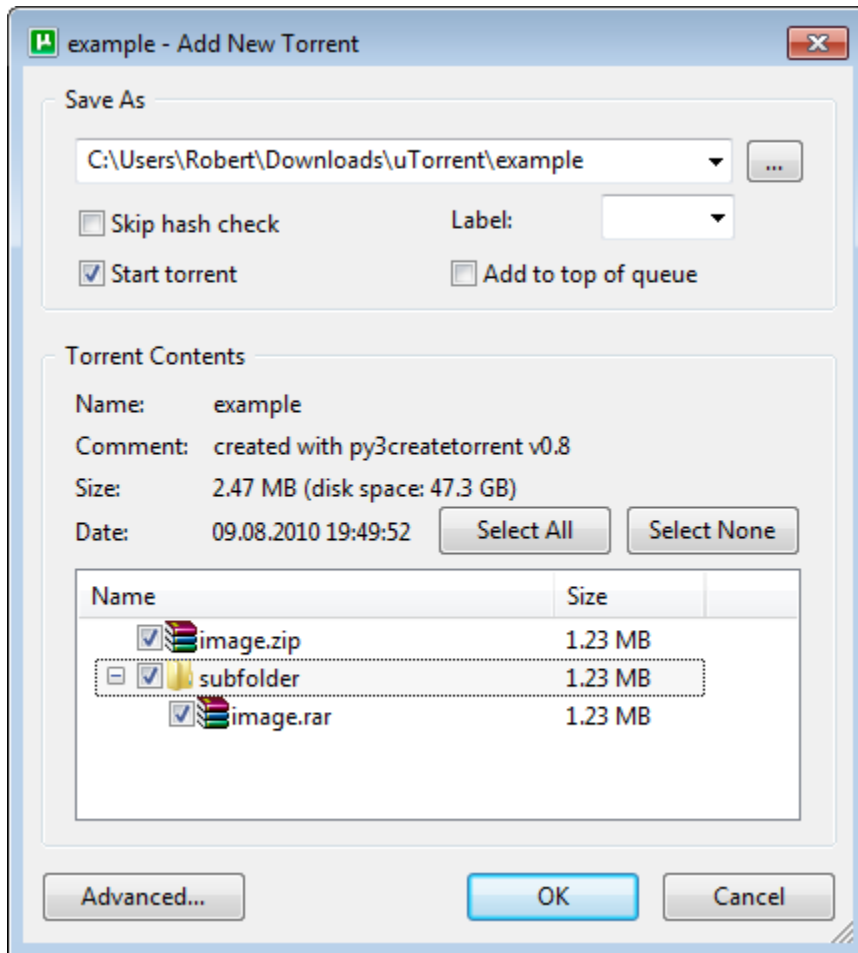
```
C:\Users\Robert\Desktop\Python\createtorrent>py3createtorrent -e example\anotherimage.
↪ jpg -e example\subfolder\10_more_minutes_please.JPG example -t udp://tracker.
↪ opentrackr.org:1337/announce
```

Alternative, equivalent command using **regular expressions** instead of specifying each jpg separately (also using a tracker abbreviation to make it even shorter):

```
C:\Users\Robert\Desktop\Python\createtorrent>py3createtorrent --exclude-pattern
↪ "(jpg|JPG)$" example -t opentrackr
```

Effect: Creates example.torrent inside the current directory. exampleanotherimage.jpg and examplesubfolder10_more_minutes_please.JPG have been excluded.

In µTorrent it will look like this:



1.6.4 Creating torrents of single files

It's almost the same as for creating directories, except, of course, you can't use the exclude-option anymore.

1.7 Configuration

If present, the configuration file `py3createtorrent.cfg` will be loaded from the user's home directory. The configuration file uses JSON format. Use `--config` to load the config from another location. Use `--verbose` for troubleshooting this, if it does not work as expected.

Warning: Before version 1.0, the configuration had to be changed by manually editing the `py3createtorrent.py` script file. If you're still using version 0.x, please upgrade or switch to the old documentation of the 0.x branch.

1.7.1 Default

If the configuration file is not present, the following default values will be used:


```
{
  "best_trackers_url": "https://raw.githubusercontent.com/ngosang/trackerslist/master/
↪trackers_best.txt",
  "tracker_abbreviations": {
    "opentrackr": "udp://tracker.opentrackr.org:1337/announce",
    "coppersurfer": "udp://tracker.coppersurfer.tk:6969/announce",
    "cyberia": "udp://tracker.cyberia.is:6969/announce"
  },
  "advertise": true
}
```

For details on the individual configuration parameters, please refer to the following sub-sections.

1.7.2 Best trackers URL

You can change the URL from which the best tracker URLs are loaded when using the *bestN shortcut*. The default URL is:

```
https://raw.githubusercontent.com/ngosang/trackerslist/master/trackers_best.txt
```

To change it, you can use a config file like this:

```
{
  "best_trackers_url": "https://ngosang.github.io/trackerslist/trackers_best_ip.txt"
}
```

1.7.3 Tracker abbreviations

Tracker abbreviations allow you to specify one or more tracker URLs with a single word, like ‘opentrackr’ in the default configuration. They add a lot of convenience, e.g. look at this neat & clear command:

```
C:\Users\Robert\Desktop\Python\createtorrent>py3createtorrent example -t opentrackr -
↪t coppersurfer
Successfully created torrent:
  Name:          example
  (...)
  Primary tracker:  udp://tracker.opentrackr.org:1337/announce
  Backup trackers:
    udp://tracker.coppersurfer.tk:6969/announce
```

In this case, py3createtorrent recognizes the tracker abbreviations ‘opentrackr’ and ‘coppersurfer’ and automatically inserts the according tracker announce URLs.

Note: Single abbreviations may be replaced by multiple tracker announce URLs. This way you can also create sort of “tracker groups” for different kinds of torrents.

Example configuration:

```
{
  "tracker_abbreviations": {
    "mytrackergroup": [
      "udp://tracker.opentrackr.org:1337/announce",
      "udp://tracker.coppersurfer.tk:6969/announce"
    ]
  }
}
```

(continues on next page)

(continued from previous page)

```
    ],
    "opentrackr": "udp://tracker.opentrackr.org:1337/announce",
    "coppersurfer": "udp://tracker.coppersurfer.tk:6969/announce"
  }
}
```

Just specify lists of announce URLs instead of a single announce URL to define such groups.

1.7.4 Advertise setting

The `advertise` setting defines whether py3createtorrent is allowed to advertise itself through the comment field, if the user hasn't specified a comment. Possible values are `true` (the default) or `false` - without any quotes.

To disable advertising, you can use the following in your config file:

```
{
  "advertise": false
}
```

If you want to disable advertising for a single torrent only, you can use the `--comment` option to specify an empty comment:

```
$ py3createtorrent --comment "" ...

or

$ py3createtorrent -c "" ...
```

py3createtorrent will not advertise itself in this case, because you explicitly specified the empty comment.

2.1 Version 1.0.1

Release date: 2022/01/05

- fixed: Faulty torrent name and ‘name’ field when using relative path ‘.’, see #11

2.2 Version 1.0.0

Release date: 2021/01/11

- changed: **requires Python 3.5+ now**
- changed: specifying trackers is now optional with the new `-t` switch, thus **trackerless torrents are now possible**
- added: **bestN shortcut!** It is now possible to add the best N trackers from [ngosang/trackerslist](https://ngosang.github.io/trackerslist/) by using the new `bestN shortcut`
- changed: use **external JSON files for configuration**, by default try to load `.py3createtorrent.cfg` from user’s home directory
- added: DHT bootstrap nodes can now be specified with the new `--node` switch (doing so is recommended for trackerless torrents)
- added: **webseed support** with the new `--webseed` switch (GetRight style, i.e. http://bittorrent.org/beps/bep_0019.html)
- changed: increased max piece size to 16 MiB
- changed: show warning if piece size is not a multiple of 16 KiB
- changed: updated the default trackers (openbt is now opentracker, dropped publicbt, added cyberia and copper-surfer)
- added: Pipfile and Pipfile.lock for pipenv support

- added: README.md
- changed: improved performance of single file torrent creation
- fixed: parentheses are now allowed in names specified by using the `--name` switch
- refactored: switched to `bencode.py` module for encoding the torrent data
- refactored: switched from `optparse` to `argparse`
- refactored: reformatted code with `yapf`, using a new column limit of 120
- refactored: added type hints to enable analysis with `mypy` (we use Python's typing module which was added in Python 3.5, thus Python 3.5 is the new minimum version that is required)

2.2.1 Version 1.0.0b2 (beta version)

Release date: 2021/01/08

- fixed: parentheses are now allowed in names specified by using the `--name` switch

2.2.2 Version 1.0.0b1 (beta version)

Release date: 2021/01/04

- changed: **requires Python 3.5+ now**
- changed: specifying trackers is now optional with the new `-t` switch, thus **trackerless torrents are now possible**
- added: **bestN shortcut!** It is now possible to add the best N trackers from [ngosang/trackerslist](https://github.com/ngosang/trackerslist) by using the new `bestN shortcut`
- changed: use **external JSON files for configuration**, by default try to load `.py3createtorrent.cfg` from user's home directory
- added: DHT bootstrap nodes can now be specified with the new `--node` switch (doing so is recommended for trackerless torrents)
- added: **webseed support** with the new `--webseed` switch (GetRight style, i.e. http://bittorrent.org/beps/bep_0019.html)
- changed: increased max piece size to 16 MiB
- changed: show warning if piece size is not a multiple of 16 KiB
- changed: updated the default trackers (openbt is now `opentracker`, dropped `publicbt`, added `cyberia` and `coppersurfer`)
- added: `Pipfile` and `Pipfile.lock` for `pipenv` support
- added: README.md
- changed: improved performance of single file torrent creation
- refactored: switched to `bencode.py` module for encoding the torrent data
- refactored: switched from `optparse` to `argparse`
- refactored: reformatted code with `yapf`, using a new column limit of 120
- refactored: added type hints to enable analysis with `mypy` (we use Python's typing module which was added in Python 3.5, thus Python 3.5 is the new minimum version that is required)

2.3 Version 0.9.7

Release date: 2020/07/23

- new: switch `--source` to include a metainfo field 'source', which is required by some private trackers (contributed by cপুরুলেস)
- changed: slightly improved docs on `--date` switch (now mentions the special value `-2` for disabling the date field altogether)
- changed: slightly improve handling of negative timestamp values for `--date` switch

2.4 Version 0.9.6

Release date: 2019/08/29

- new: exit gracefully if `py3bencode` module could not be imported and show instructions on how to fix this
- docs: Updated docs after migration from Mercurial/Bitbucket.org to Git/Github.com

2.5 Version 0.9.5

Release date: 2013/06/04

- new: switch `--md5` to request MD5 hashes; they are now turned off by default, resulting in a **significant performance improvement** (af745c8581de)
- new: switch `--exclude-pattern-ci` for case-insensitive regular expressions (1c68ad21c72f)
- fixed: On Windows, the `--exclude-pattern` switch has not been case-sensitive (f5c00b9eccbc)
- fixed docs: installation instructions for `py3bencode` using `setup.py` (276a82e1cbc3)

2.6 Version 0.9.4

Release date: 2013/02/26

- new: documentation is now part of the repository, based on Sphinx (dd3d74f5cc26 and following)
- fixed: UDP tracker announce urls (c639e2f8408a + 69afadea92e4)
- fixed: piece count calculation (8450c6470d7f)
- fixed: inconsistent number of blank lines after summary (30f870d55c56)

2.7 Version 0.9.3

Release date: 2010/12/13

- ! fixed: tracker abbreviations for openbittorrent fixed. **new default abbreviations: openbt and publicbt** for openbittorrent and publicbittorrent. Note that both of them do no more offer a http announce URL (they are pure UDP trackers now).
- fixed: did not prompt the user when overwriting an existing torrent using the `-o <output directory>` switch

- fixed: version number was still 0.9 (now 0.9.3, of course), so it did not identify itself correctly, e.g. when issuing “-version”.

2.8 Version 0.9.2

Release date: 2010/11/09

- fixed: the private switch (-private / -P) did not have any effect (reported by steven)
- fixed: wrong email address (now ...+dev@gmail.com instead of dev+...@gmail.com)

2.9 Version 0.9.1

Release date: 2010/10/17

- !!! fixed: torrents for single files could not be created (reported by JWA)

2.10 Version 0.9

Release date: 2010/08/19

- !!! fixed: creating torrents with multiple trackers did not work. the announce-list has been created in a wrong way by version 0.8.
- added: possibility to create tracker abbreviations. by default there is ‘obt’ for OBT((Open BitTorrent - an open tracker project))’s announce urls
- added: print summary after writing the torrent file
- added: skipping symlinks that point to files or directories that have already been processed (or are still being processed)
- added: -v / -verbose option (reports skipped & processed files)
- added: -q / -quiet option (at the moment this option only removes the summary in the end)
- added: you may now use -exclude-pattern to exclude files/folders based on regular expressions
- changed: applied [Python Style Guide \(PEP 8\)](#)
- changed: using `#!/usr/bin/env python3` instead of `#!/usr/bin/python3`
- changed: removed the huge `get_size` function, there was a better way to go
- fixed: removing duplicate trackers now
- fixed: empty comment now disables comment field (didn’t work before)

2.11 Version 0.8

Release date: 2010/08/10.

Initial release.

3.1 Todo

- implement multi-threading
- add the ability to define how the announce-list should be constructed in detail (tracker tiers etc., see [Multitracker Metadata Extension](#))
- add webseed support ([Hoffman style](#) & [GetRight style](#))
- improve behaviour when detecting the need for overwriting an existing torrent (if the user aborts, the torrent has to be calculated again)
- isn't the comment field supposed to be multi-lined?
- create test cases, unit tests, ...
- provide more examples in the documentation

3.2 Future

Some ideas regarding the long-term future:

- create GUI
- make it possible to edit existing torrents

CHAPTER 4

Indices and tables

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