

---

# **py3createtorrent Documentation**

*Release 0.9.6-dev*

**Robert Nitsch**

**Jul 16, 2017**



---

# Contents

---

<b>1</b>	<b>py3createtorrent</b>	<b>3</b>
1.1	Motivation . . . . .	3
1.2	Requirements . . . . .	3
1.3	Installation . . . . .	4
1.4	Configuration . . . . .	4
1.5	Usage . . . . .	5
1.6	Examples . . . . .	8
<b>2</b>	<b>Changelog</b>	<b>13</b>
2.1	Version 0.9.5 . . . . .	13
2.2	Version 0.9.4 . . . . .	13
2.3	Version 0.9.3 . . . . .	13
2.4	Version 0.9.2 . . . . .	14
2.5	Version 0.9.1 . . . . .	14
2.6	Version 0.9 . . . . .	14
2.7	Version 0.8 . . . . .	14
<b>3</b>	<b>Developer documentation</b>	<b>15</b>
3.1	Todo . . . . .	15
3.2	Future . . . . .	15
<b>4</b>	<b>Indices and tables</b>	<b>17</b>



Contents:



*Create torrents via command line!*

py3createtorrent is a comprehensive shell/commandline utility for creating torrents (Linux & Windows). It's a GPL-licensed Python v3.1 script. I tested it with Ubuntu 8.04 / rTorrent and Windows 7 / µTorrent.

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 **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

## 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.

## Requirements

py3createtorrent requires at least Python 3.1 and the [py3bencode](#) module.

## Installation

Download the desired version from here: <https://bitbucket.org/rsnitsch/py3createtorrent/downloads>

The required py3bencode module is shipped alongside py3createtorrent. As long as you extract py3bencode into the same directory as the py3createtorrent script, you should be fine. This is the easy way and should be appropriate for the majority of users.

## Installing the py3bencode module globally

Advanced users might prefer to install the py3bencode module globally in their Python installation.

You can use `pip` to install the py3bencode module in your Python installation (in the site-packages, to be precise):

```
pip install hg+https://bitbucket.org/rsnitsch/py3bencode
```

Make sure to use the pip executable that belongs to the Python interpreter with which you will execute py3createtorrent.

If you don't have pip around (although I strongly recommend using it) you can also try to install py3bencode manually:

```
$ hg clone https://bitbucket.org/rsnitsch/py3bencode
$ cd py3bencode
$ python3 setup.py install
```

Note that any local version of py3bencode will take precedence over the global version installed in your site-packages. Thus, you will have to delete the py3bencode module that has been shipped alongside py3createtorrent, if you want the global version to be used.

## Configuration

There is a small configuration section in the script (at the top):

```
# #####
# CONFIGURATION

# configure your tracker abbreviations here
TRACKER_ABBR = {'openbt':      'udp://tracker.openbittorrent.com:80',
                'publicbt':   'udp://tracker.publicbt.com:80'}

# whether or not py3createtorrent is allowed to advertise itself
# through the torrents' comment fields
ADVERTISE = True

# /CONFIGURATION
# #####
```

## Tracker abbreviations

Tracker abbreviations allow you to specify one or more tracker URLs with a single word, like 'openbt' 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.py example openbt_
↪publicbt
Successfully created torrent:
  Name:                example
(...)
  Primary tracker:    udp://tracker.openbittorrent.com/announce
  Backup trackers:
    udp://tracker.publicbt.com:80/announce
```

In this case, py3createtorrent recognizes the tracker abbreviations ‘openbt’ and ‘publicbt’ 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_ABBR = {'mytrackergroup': ['udp://tracker.openbittorrent.com:80/announce',
                                   'udp://tracker.publicbt.com:80/announce'],
                'openbt':         'udp://tracker.openbittorrent.com:80/announce',
                'publicbt':       'udp://tracker.publicbt.com:80/announce' }
```

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

## 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.

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

```
$ py3createtorrent.py --comment "" ...
or
$ py3createtorrent.py -c "" ...
```

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

## Usage

Syntax:

```
Usage: py3createtorrent.py [options] <file-or-directory> <main-tracker-url> [<backup-
↪tracker-url> ...]
```

py3createtorrent **is** a comprehensive command line utility **for** creating torrents.

Options:

```
--version          show program's version number and exit
-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
-f, --force
    dont 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
```

## 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.

## 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.

## 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).

## Force (-f)

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

## 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.

## Quiet (-q)

py3createtorrent will try to stay completely silent on the commandline.

## 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.

## 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.

## 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.).

## 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](#).

## 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.

---

## 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.*

## 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.

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

### Command:

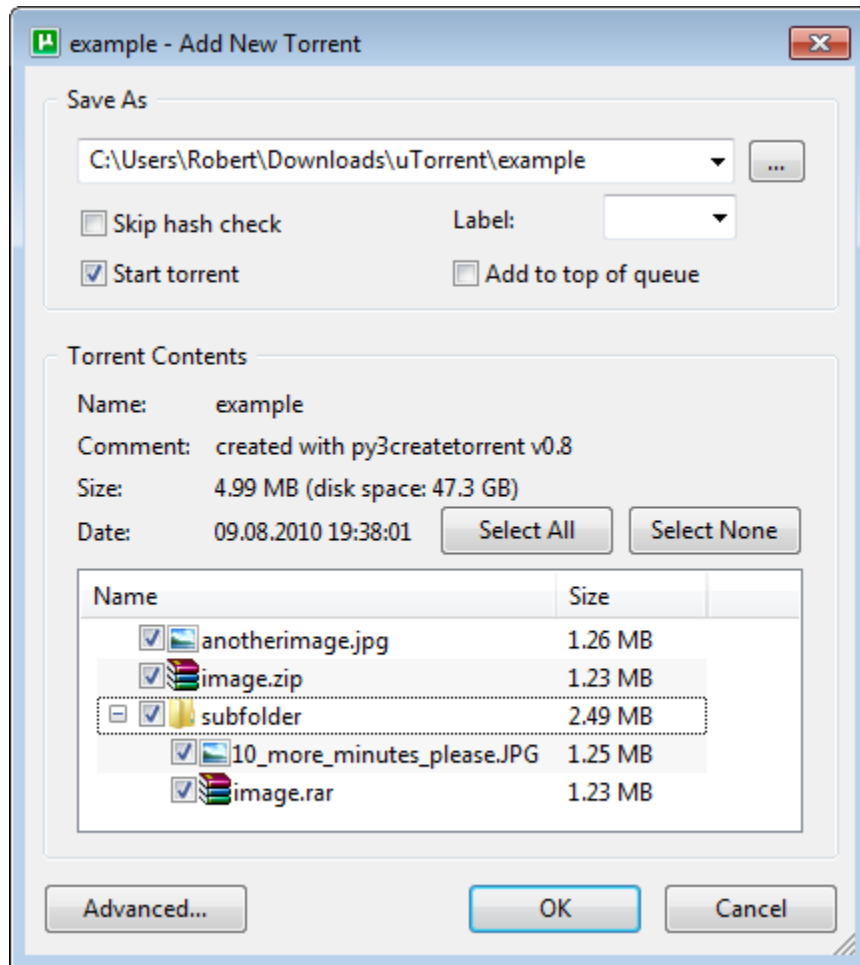
```
C:\Users\Robert\Desktop\Python\createtorrent>py3createtorrent.py example udp://
↳tracker.openbittorrent.com/announce
```

Alternative, equivalent command using a tracker abbreviation for convenience:

```
C:\Users\Robert\Desktop\Python\createtorrent>py3createtorrent.py example openbt
```

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

In µTorrent 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.

## Example 2 - from directory, excluding subfolders

### Command:

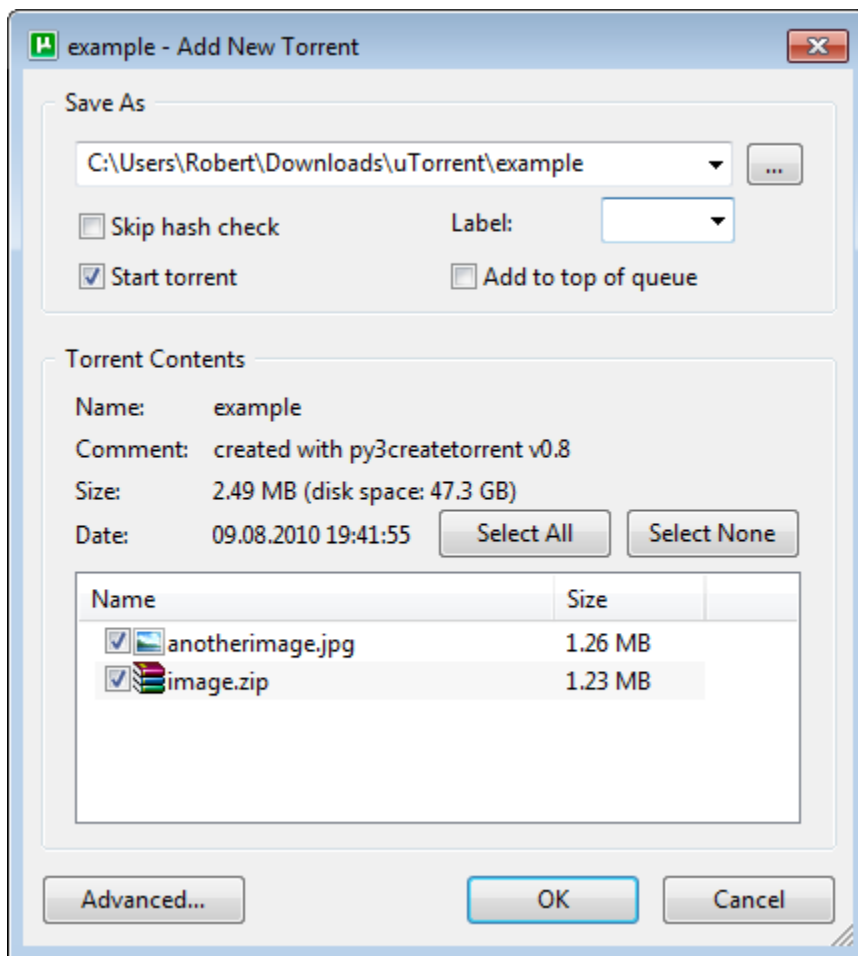
```
C:\Users\Robert\Desktop\Python\createtorrent>py3createtorrent.py -e example\subfolder_
↪example udp://tracker.openbittorrent.com/announce
```

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

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

```
py3createtorrent.py -e exclusion1 -e exclusion2 yourfolder tracker-url
```

In µTorrent it will look like this:



## Example 3 - from directory, excluding files

### Command:

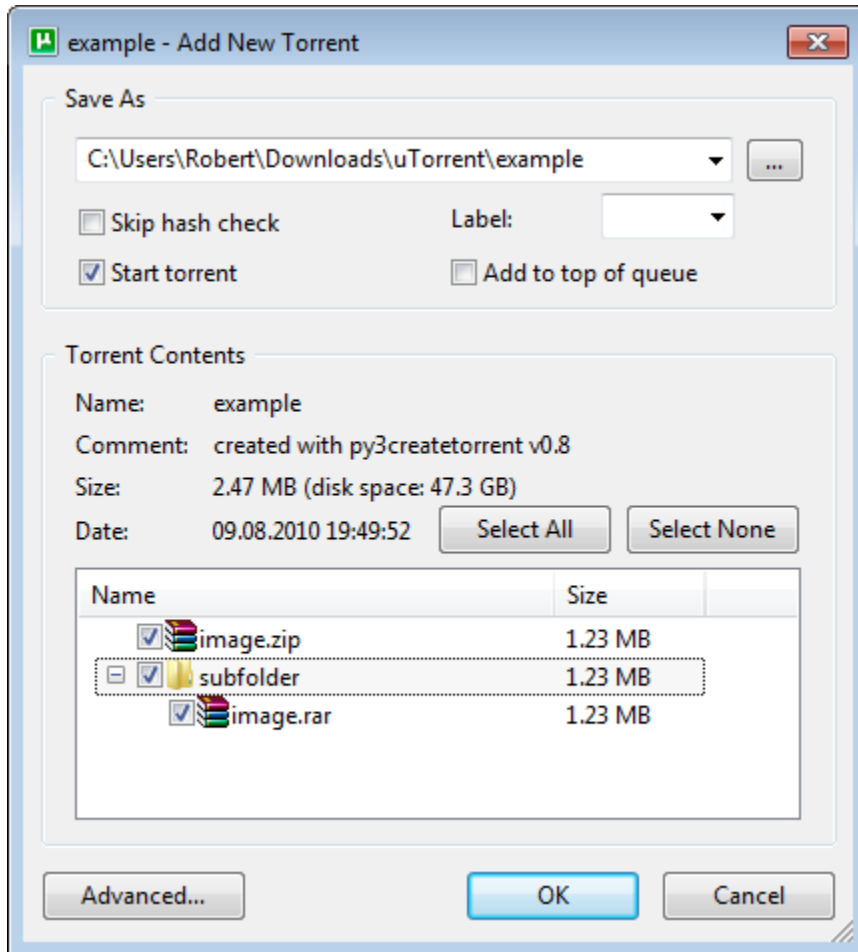
```
C:\Users\Robert\Desktop\Python\createtorrent>py3createtorrent.py -e_
↪example\anotherimage.jpg -e example\subfolder\10_more_minutes_please.JPG example_
↪udp://tracker.openbittorrent.com/announce
```

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.py --exclude-pattern
↪ "(jpg|JPG) $" example openbt
```

**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:



## 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.





#### 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)

#### 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)

#### 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`”.

## 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`)

## Version 0.9.1

*Release date: 2010/10/17*

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

## 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)

## Version 0.8

*Release date: 2010/08/10.*

Initial release.

### Todo

- use configuration files instead of requiring users to edit the script
- 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

### 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`