

---

# autotorrent Documentation

*Release latest*

May 14, 2015



<b>1</b>	<b>Requirements</b>	<b>3</b>
<b>2</b>	<b>Install</b>	<b>5</b>
<b>3</b>	<b>Upgrade from previous version</b>	<b>7</b>
<b>4</b>	<b>Configuration</b>	<b>9</b>
4.1	general . . . . .	9
4.2	client . . . . .	9
4.3	disks . . . . .	10
<b>5</b>	<b>Scan modes</b>	<b>11</b>
5.1	Mode: normal . . . . .	11
5.2	Mode: exact . . . . .	11
5.3	Mode: unsplitable . . . . .	11
5.4	Mode: hash_name . . . . .	11
5.5	Mode: hash_size . . . . .	11
5.6	Mode: hash_slow . . . . .	12
<b>6</b>	<b>Instructions</b>	<b>13</b>
<b>7</b>	<b>FAQ</b>	<b>15</b>
<b>8</b>	<b>License</b>	<b>17</b>



Given an input torrent, it will scan your collection for the files in the torrent. If all (or most) the files are found, a folder with links to all the files will be created and the torrent added to the torrent client.

All you need to do is download the torrents and AutoTorrent plays mix and match to make it possible to seed as much as possible across trackers.



---

## Requirements

---

- Linux, BSD, OSX - Something not windows
- rTorrent, Deluge and Transmission
- Python 2.6, 2.7, 3.3, 3.4
- Shell / SSH / Putty



---

### Install

---

From PyPi (stable):

```
virtualenv autotorrent-env
autotorrent-env/bin/pip install autotorrent
```

From GitHub (develop):

```
virtualenv autotorrent-env
autotorrent-env/bin/pip install git+https://github.com/JohnDoe/autotorrent.git#develop
```

Get the configuration file

```
wget -Oautotorrent.conf https://github.com/JohnDoe/autotorrent/raw/develop/autotorrent.conf.example
```



---

## Upgrade from previous version

---

### Upgrading from PyPi (stable)

```
autotorrent-env/bin/pip install --upgrade autotorrent
```

### Upgrading from Github (develop)

```
autotorrent-env/bin/pip install git+https://github.com/JohnDoe/autotorrent.git#develop --upgrade --i
```



---

## Configuration

---

All settings can be found and changed in `autotorrent.conf`, this file must reside in the same folder as `autotorrent` is executed from.

### 4.1 general

- `db` - Path to the database file
- `store_path` - Folder where the virtual folders seeded, resides
- `ignore_files` - A comma separated list of files that should be ignored (supports wildcards)
- `add_limit_size` - Max size, in bytes, the total torrent size is allowed to vary
- `add_limit_percent` - Max percent the total torrent size is allowed to vary
- `link_type` - What kind of link should AutoTorrent make? the options are hard and soft.
- `scan_mode` - options are unsplitable, normal and exact. These can be used in combination. See the `scan_mode` section for more information.

the `add_limit_*` variables allow for downloading of e.g. different NFOs and other small files that makes a difference in the torrents.

### 4.2 client

- `client` - torrent client to use, choices are: rtorrent, deluge and transmission

#### 4.2.1 rtorrent settings

- `url` - URL to rtorrent, must be to the XMLRPC server or SCGI server.
- `label` - Label added to torrents when added to rtorrent (used in rutorrent only)

the `url` supports both SCGI directly and XMLRPC via HTTP.

To use `scgi`, prefix the `url` with `scgi` instead of `http`, e.g. `scgi://127.0.0.1:10000/`

To use unix socket for `scgi`, make an `url` with no `ip:port` and instead a path, e.g. `scgi:///tmp/rtorrent.socket`

### 4.2.2 deluge settings

- host - an ip:port pair, e.g. 127.0.0.1:12345
- username - deluge rpc username
- password - deluge rpc password

### 4.2.3 transmission settings

- url - an url where transmission can be reached, e.g. <http://username:password@127.0.0.1:9091>

## 4.3 disks

A list of disks where to build the search database from.

The keys must be sequential, i.e. disk1, disk2, disk3 etc.

---

## Scan modes

---

There are currently three scan modes supported by AutoTorrent. These modes can be used in combination and should all improve the end result.

The modes are named normal, exact and unsplitable. They can be combined by adding a comma between them, e.g. `scan_mode=normal,exact,unsplitable`

### 5.1 Mode: normal

It takes the filename and size and tries to find files with same name and size.

This mode cannot handle duplicate filename/size pairs.

### 5.2 Mode: exact

The perfect way to move torrent client as it tries to set the download path to the old path.

This mode does not allow for missing files and is intended to re-add non-renamed back to a torrent client.

### 5.3 Mode: unsplitable

This mode takes scene releases and extracted dvd/bluray isos into consideration and relies on the folder it thinks is the main / head folder. Perfect for cross-seeding scene releases.

### 5.4 Mode: hash\_name

This mode tries to hashcheck files with the exact name as wanted, but the size might be different (up to 10% different). If pieces match, then it is resized to fit original size and written to the destination directory.

Make sure there is enough space in the target directory.

### 5.5 Mode: hash\_size

This mode tries to hashcheck files with the exact size as wanted, but the name might be different.

## 5.6 Mode: hash\_slow

This mode tries to hashcheck files with a size within 10% of the original. If pieces match, then it is resized to fit original size and written to the destination directory.

Make sure there is enough space in the target directory.

This mode is very slow as it will try a lot of files.

---

## Instructions

---

Start by installing and configuring.

Step 1, build the database with `autotorrent -r`, this may take some time.

Step 2, have some torrents ready and run `autotorrent -a folder/with/torrents/*.torrents`, this command will spit out how it went with adding the torrents.

And you're good to go.



**Q: How are files with relative path in the configuration file, found?**

The paths should be relative to the configuration file, e.g. `/home/user/autotorrent-env/autotorrent.conf`, then `store_path=store_paths/X/` resolves to `/home/user/autotorrent-env/store_path/`

**Q: I have three sites I cross-seed between, how do you suggest I structure it?**

Say, you have site X, Y and Z. You want to seed across the sites as they share lots of content. You download all your data into `/home/user/downloads/`. For this you will need three configuration file, one for each site.

AutoTorrent is installed into `/home/user/autotorrent-env/`.

Only `store_path` is recommended to vary between the configuration files (the others are optional).

- `store_path` for site X - `/home/user/autotorrent-env/store_paths/X/`
- `store_path` for site Y - `/home/user/autotorrent-env/store_paths/Y/`
- `store_path` for site Z - `/home/user/autotorrent-env/store_paths/Z/`

disks paths can be:

- `disk1=/home/user/downloads/`
- `disk2=/home/user/autotorrent-env/store_paths/X/`
- `disk3=/home/user/autotorrent-env/store_paths/Y/`
- `disk4=/home/user/autotorrent-env/store_paths/Z/`

**Q: Can I use the same Database file for several configuration files?**

Yes, if they have the same disks. Don't worry about adding the `store_path` to the disks, AutoTorrent will figure it out.

**Q: What problems can occur?**

One big problem is that the files are not checked for their actual content, just if their filename matches and size matches. If AutoTorrent tries to use a file that is not complete, then you can end up sending loads of garbage to innocent peers, although they should blackball you quite fast.

**Q: I want to cross-seed RARed scene releases, what do you think about that?**

The actual `.rar` files must be completely downloaded and the same size. Things that can vary are: `nfos`, `sfvs`, `samples` and `subs`.

The releases must also have an `sfv` in the same folder as the `rar` files files.

**Q: What are hardlinks and what are the risks or problems associated with using them?**

See: <http://www.cyberciti.biz/tips/understanding-unixlinux-symbolic-soft-and-hard-links.html>



---

**License**

---

MIT, see LICENSE