autoremove-torrents

Release 1.5.3

Jan 21, 2021

Contents:

CHAPTER 1

Introduction

This program is a tool that can help you remove torrents automatically. Now, you don't need to worry about your disk space anymore - according to your strategies, the program will check each torrent if it satisfies the remove condition; If so, delete it automatically.

Let's start here. By reading this documentation, we can learn how to install, config and run this tool.

1.1 Supported Clients

Until now, this program supports qBittorrent/Transmission/μTorrent. Deluge and rTorrent are both on my planning list.

Client	Support
qBittorrent	Yes
Transmission	Yes
μTorrent	Yes
Deluge	On Planning
rTorrent	On Planning

1.2 Supported Properties

✓ = Supported = Unsupported

Properties/Clients	Deluge	qBittorrent	Transmission	μTorrent
Average Download Speed	✓ 2.0 or later	✓	✓	
Average Upload Speed	✓	✓	✓	
Category	✓ Requires Label plugin	✓	✓ 3.00+ or later	✓
Connected Leecher	✓	✓	✓	✓
Connected Seeder	✓	✓	✓	✓
Create Time	✓	✓	✓	
Download Speed	✓	✓	✓	✓
Downloaded Size	✓	✓	✓	✓
Free Space	✓	✓	✓	
Last Activity	✓ 2.0 or later	✓ v3.0 or later	✓	
Leecher	✓	✓	✓	✓
Progress	✓	✓	✓	✓
Ratio	✓	✓	✓	✓
Seeder	✓	✓	✓	✓
Seeding Time	✓	✓	✓	✓
Size	✓	✓	✓	✓
Stall Status		✓	✓	
Status	✓	✓	✓	✓
Tracker	✓	✓	✓	✓
Upload Ratio	✓	✓	✓	✓
Upload Speed	✓	✓	✓	✓
Uploaded Size	✓	✓	✓	✓

CHAPTER 2

Install and Run

2.1 Install

There are two ways to install `autoremove-torrents`, but I highly recommend installing from pip.

2.1.1 Install from pip

```
pip install autoremove-torrents
```

2.1.2 Install from GitHub

```
git clone https://github.com/jerrymakesjelly/autoremove-torrents.git
cd autoremove-torrents
python3 setup.py install
```

2.2 Run

Just type the following command line in your terminal:

```
autoremove-torrents
```

`autoremove-torrents` will look for the `config.yml` in the current working directory. For more command line arguments, please see the table below.

2.2.1 Arguments List

Note: When you are using the full name of the arguments, you need to lead the values of the arguments with a equal sign. But if you are using the abbreviation, you only need a space to lead the argument values.

Arug-ments	Argument Abbrevia-tions	Description
–view	*-v*	Run and see which torrents will be removed, but don't really remove them.
–conf	*-c*	Specify the path of the configuration file.
–task	*-t*	Run a specific task only. The argument value is the task name.
–log	*-l*	Sepcify the path of the log file.
–debug	*-d*	Enable debug mode and output more logs.

For example:

```
autoremove-torrents --view --conf=/home/myserver/autoremove-torrents/config.yml
```

Also, it equals to:

```
autoremove-torrents -v -c /home/myserver/autoremove-torrents/config.yml
```

2.3 Uninstall

2.3.1 Uninstall from pip

If your autoremove-torrents was installed via pip, you can simply uninstall it by using pip:

```
pip uninstall autoremove-torrents
```

2.3.2 Uninstall manually

However, if it was installed by `setup.py`, you need to remove all the files manually.

Step1

```
cd autoremove-torrents
```

Step2

Reinstall the program and record which files were copied:

```
python3 setup.py install --record files.txt
```

Step3

Use `xargs` to remove each file:

```
cat files.txt | xargs rm -rf
```

Or if you're running Windows, use Powershell:

```
Get-Content files.txt | ForEach-Object {Remove-Item $_ -Recurse -Force}
```

Reference: https://stackoverflow.com/questions/1550226/python-setup-py-uninstall

CHAPTER 3

Configuration

Before we run `autoremove-torrents`, we need to create a `config.yml` to save our configurations.

> **Warning:** In order to avoid the torrents being mistakenly deleted, we highly recommend you to run `autoremove-torrents --view` once to preview the results after modifying the configuration file.

The script uses the YAML language as the language of the configuration file. The YAML language has a clear structure, so I think it's more friendly than the JSON and easy to learn.

Look at the example please, the task block can be divided into 3 parts.

```yaml
# A task block
my_task:           # Part 1: Task Name
  # Part 2: Login Information
  client: qbittorrent
  host: http://127.0.0.1:9091
  username: admin
  password: adminadmin
  # Part 3: Strategies Block (Remove Conditions)
  strategies:
    strategy1:     # Part I: Strategy Name
      # Part II: Filters
      categories:
        - IPT
      # Part III: Remove Condition
      ratio: 1
      seeding_time: 1209600
    strategy2:
      all_categories: true
      excluded_categories:
        - IPT
      seeding_time: 259200
    # Add more strategies here...
  # Part 4: Decide whether to remove and delete data (optional)
```

(continues on next page)

(continued from previous page)

```
  delete_data: true

# Add more tasks here...
```

Centainly, the configuration file can contain more than one task blocks, and a task block can contain more than one strategy blocks. Each task block represents a BT client, and each strategy block represents a kind of torrents.

3.1 Part 1: Task Name

Just name your task.

Note: No spaces are allowed before the task name.

3.2 Part 2: Login Information

This part is your login inforamtion.

3.2.1 For qBittorrent, Transmission or μTorrent

For qBittorrent/Transmission/μTorrent, this program works with your client's WebUI.

- `client`: Your client name. It's case-insensitive.
- `host`: The URL of your client's WebUI, and the URL must have a scheme (http:// or https://).
- `username`: The username of the WebUI.
- `password`: The password of the WebUI.

3.2.2 For Deluge

This program accesses Deluge via its RPC protocol.

- `client`: Your client name. Here is Deluge.
- `host`: The IP address (or domain name) and the port number of your Deluge Daemon, for example, `127.0.0.1:58846`.
- `username`: The username of the Deluge Daemon.
- `password`: The password of the Deluge Daemon.

Example:

```
my_task:
  client: deluge
  host: 127.0.0.1:58846
  username: localclient
  password: 357a0d23f09b9f303f58846e41986b36fef2ac88
```

Note:

1. Don't write any schemes in `host` field. The program uses neither HTTP protocol nor HTTPS protocol to access Deluge.
2. The port number is the port number of the Deluge Daemon, not the WebUI. You can find it in the Connection Manager of your WebUI.
3. When you are running the autoremove-torrents and the Deluge on different computers, please make sure that your Deluge accepts remote connections. You can modify this setting at **Preferences -> Daemon -> Allow Remote Connections**.

Note: Generally, you can find the username and password in `~/.config/deluge/auth`. Also, you can create a new user by adding a new line to the end of the file.

For more information of the authentication, please visit https://dev.deluge-torrent.org/wiki/UserGuide/Authentication.

3.3 Part 3: Strategy Block

This part contains strategy blocks. Each strategy block can be divided into 3 parts, too.

3.3.1 Part I: Strategy Name

Just name your strategy like the task name.

3.3.2 Part II: Filters

The removing condtions are only available for the torrents you chosen. There are 9 filters available.

- `all_trackers`/`all_categories`/`all_status`: Choose all the trackers/categories/status.
- `categories`: Choose torrents in these categories.
- `excluded_categories`: Don't choose torrents in these categories.
- `trackers`: Choose torrents in these trackers.
- `excluded_trackers`: Don't choose torrents in these trackers.
- `status`: Choose torrents in these status. Available status is as follows:

Status	Remarks
Downloading	/
Uploading	/
Checking	/
Queued	/
Paused	Transmission doesn't have this status.
Stopped	qBittorrent doesn't have this status.
Error	/
StalledUpload	μTorrent doesn't have this status.
StalledDownload	μTorrent doesn't have this status.

- `excluded_status`: Don't choose these torrents in these status. Available status is shown in the table above.

The result of each filter is a set of torrents.

Note: When two or three of `categories`, `trackers` and `status` filter are specificed, the program will take the intersection of these sets, and subtracts set `excluded_categories`, `excluded_trackers` and `excluded_status`.

Note:

1. Don't write sockets in `trackers`. The `trackers` field only needs hostname, for example, just fill `tracker.site1.com` for `https://tracker.site1.com`.
2. In 1.4.4 and later version, if there's only one item in `categories`, `trackers` or `status`, it's not necessary to use list structure. A single-line text is enough, for example:

```
categories: cata1
```

```
status: uploading
```

3. The `StalledUp` and `StalledDown` is the new status in version 1.4.5. In this program, `Uploading` inlcudes the torrents in `StalledUpload` status, and `Downloading` includes the torrents in `StalledDownload` status.

Let's see some examples. Select those torrents whose categories are Movies or Games:

```
my_task:
  client: xxx
  host: xxx
  username: xxx
  password: xxx
  strategies:
    my_strategy:
      categories:
        - Movies
        - Games
      # Removing conditions are here
      # ...
```

Select those torrents whose hostnames of tracker are tracker.aaa.com or x.bbb.com:

```
my_task:
  client: xxx
  host: xxx
  username: xxx
  password: xxx
  strategies:
    my_strategy:
      trackers:
        - tracker.aaa.com
        - x.bbb.com
      # Removing conditons are here
      # ...
```

Select torrents whose categories are Movies or Games, but exclude those torrents whose tracker is tracker.yyy.com:

```yaml
my_task:
  client: xxx
  host: xxx
  username: xxx
  password: xxx
  strategies:
    my_strategy:
      categories:
        - Movies
        - Games
      excluded_trackers:
        - tracker.yyy.com
      # Removing conditions are here
      # ...
```

Select those torrents whose categories is Movies and status is uploading:

```yaml
my_task:
  client: xxx
  host: xxx
  username: xxx
  password: xxx
  strategies:
    my_strategy:
      categories:
        - Movies
      status:
        - Uploading
      # Removing conditions are here
      # ...
```

3.3.3 Part III: Remove Condition

There are 2 ways to set removing condition.

1. Use Removing Condition Keywords Directly (Recommended)

Use the removing condition keywords directly. There are 18 remove conditions.

Note: As long as a chosen torrent satisfies one of these conditions, it will be removed.

The first 15 conditions are here. In order to avoid torrents being mistakenly deleted, some conditions are only available for certain torrent status.

Condition	Unit	Available Status	Description
`ratio`		All	Maximum ratio
`create_time`	Second	All	The maximum time elapsed since the torrent was added to the client. When a torrent reaches the limit, it will be removed (no matter what state it is).
`seeding_time`	Second	All	Maximum seeding time of a torrent.
`max_download`	GiB	All	Maximum downloaded size of a torrent. Torrents whose downloaded size exceed this limitation will be removed.
`max_downloadspeed`	KiB/s	Downloading	Maximum download speed of a torrent. Torrents that exceed the limitation will be removed.
`min_uploadspeed`	KiB/s	Downloading or Uploading	Minimum upload speed of a torrent. Torrents below this speed will be removed.
`max_average_downloadspeed`	KiB/s	All	Maximum average download speed. Just like `max_downloadspeed`.
`min_average_uploadspeed`	KiB/s	All	Minimum average upload speed. Just like `min_uploadspeed`.
`max_size`	GiB	All	Torrent size limitation. Remove those torrents whose size exceeds the limit.
`max_seeder`		All	Maximum number of seeders. When the seeders exceeds the limitation, the torrent will be removed.
`max_upload`	GiB	All	Maximum uploaded size of a torrent. Torrents whose uploaded size exceed this limitation will be removed.
`min_leecher`		All	Minimum number of leechers. When the number of leechers is less than the settings, the torrent will be removed.
`max_connected_seeder`		Downloading or Uploading	Maximum number of connected seeders. Just like `max_seeder`.
`min_connected_leecher`		Downloading or Uploading	Minimum number of connected leechers. Just like `min_leecher`.
`last_activity`	Second	All	The maximum time allowed since a torrent has stopped being active, that is, the maximum time without uploading or downloading. When the torrent reaches the limit, it will be removed.
`max_progress`	Percent (%)	All	The maximum download progress. The maximum value is 100.
`upload_ratio`		All	The maximum upload ratio. Note that the upload ratio here is different from the ratio. For each torrent, the upload ratio is `uploaded size` divided by its `size`.

Beside these condition, the other 3 remove conditions are here. The rest of the torrents will be removed if they trigger these conditions.

- `seed_size`: Calculate the total size of the torrents you chosen. If the total size exceeds the limit, some of the torrents will be removed. The following two properties must be specificed.
  - `limit`: Limit of the total size, in GiB.
  - `action`: Determine which torrents will be removed. Can be the following values:

Value	Description
remove-old-seeds	Try to remove old seeds.
remove-new-seeds	Try to remove new seeds.
remove-big-seeds	Try to remove large seeds.
remove-small-seeds	Try to remove small seeds.
remove-active-seeds	Try to remove active seeds.
remove-inactive-seeds	Try to remove inactive seeds.

- `maximum_number`: Set the maximum number of torrents. When the number of chosen torrents is exceed the maximum number, some of the torrents will be deleted, just like the condition *seed_size*. The following two properties must be specified:
  - `limit`: Maximum number limitation
  - `action`: Determine which torrents will be removed. The values and its meanings are in the table above.
- `free_space`: Check the free space on disk is enough or not. When the free space is not enough, some of the chosen torrents will be deleted, just like the condition *seed_size*. The following three properties should be specified:
  - `min`: Minimum free space, in *GiB*. When the free space of the specified directory is less than this value, the removing strategy will be trigger.
  - `path`: Directory that needs to be monitored
  - `action`: Removing strategy, which determines which torrents will be removed. The values and its meanings are in the table above.
- `remote_free_space`: Decide which torrents to be removed based on the free space too, but use the free space data reported by the bittorrent client. Its behavior is the same as the `free_space`.
  - `min`: Minimum free space, in *GiB*.
  - `path`: Directory that needs to be checked by the bittorrent client.
  - `action`: Removing strategy.

Note: If your autoremove-torrents and your bittorrent client are running on different machines, you need to use `remote_free_space` to check the free spaces. Besides, `free_space` and `remote_free_space` are the same.

Please note that not all of the clients support checking the specified path. Currently, only Deluge and Transmission support, and the parameter `path` in `remote_free_space` will be ignored in qBittorrent.

Here is an example. For torrents whose categories are xxx or yyy, it removes the torrents which ratio is greater than 1 or seeding time is more than 1209600 seconds:

```
my_task:
  client: xxx
  host: xxx
  username: xxx
  password: xxx
  strategies:
    my_strategy:
      categories:
        - xxx
        - yyy
```

(continues on next page)

(continued from previous page)

```
      ratio: 1
      seeding_time: 1209600
```

Here is another example. For all torrents, it removes the torrents which seeding time is greater than 259200 seconds:

```
my_task:
  client: xxx
  host: xxx
  username: xxx
  password: xxx
  strategies:
    my_strategy:
      seeding_time: 259200
```

Here is another another example. For all torrents, when the free space in directory */home/myserver/downloads* is less than 10GiB, the program will try to remove the big torrents:

```
my_task:
  client: xxx
  host: xxx
  username: xxx
  password: xxx
  strategies:
    my_strategy:
      free_space:
        min: 10
        path: /home/myserver/downloads
        action: remove-big-seeds
```

Here is the last example. For all torrents, remove those torrents whose ratio is greater than 3 first, and then if the total size of the rest of torrents is larger than 500 GiB, it will remove active torrents until the total size is less than 500 GiB:

```
my_task:
  client: xxx
  host: xxx
  username: xxx
  password: xxx
  strategies:
    my_strategy:
      ratio: 3
      seed_size:
        limit: 500
        action: remove-active-seeds
```

2. Use `remove` Keyword (Advanced)

Use the `remove` keyword. The `remove` keyword is a new keyword in version 1.4.0, which supports the complex removing condition. The `remove` keyword is followed by an expression, which consists of the following syntax:

1. `<Parameter> <Comparison Operator> <Value>`

   `Parameter`: Available parameters are as follows, and they are case-insensitive.

   Note: Some properties can only be used in specific status. The torrents not in available status will not be

removed.

Parameter	Unit	Available Status	Description
`average_downloadspeed`	KiB/s	All	Average download speed.
`average_uploadspeed`	KiB/s	All	Average upload speed.
`connected_leecher`	/	Downloading or Uploading	The number of connected leecher.
`connected_seeder`	/	Downloading or Uploading	The number of connected seeder.
`create_time`	Second	All	The elapsed time since the torrent was added to the client.
`download`	GiB	All	Downloaded Size
`download_speed`	KiB/s	Downloading	Download speed.
`last_activity`	Second	All	The elapsed time since the torrent has stopped being active (without uploading or downloading).
`leecher`	/	All	The number of leechers.
`progress`	%	All	The download progress.
`ratio`	/	All	Ratio
`seeder`	/	All	The number of seeders.
`seeding_time`	Second	All	Seeding time.
`size`	GiB	All	The torrent size.
`upload`	GiB	All	Uploaded Size
`upload_ratio`	/	All	uploaded size / size
`upload_speed`	KiB/s	Downloading or Uploading	Upload Speed

`Comparison Operator`: Available parameters are as follows. This program doesn't provide the `equal` sign, because the status data of the torrents change quickly, and usually it's meaningless to set a specific value.

Comparison Operator	Description
<	Less Than
>	Greater Than

`Value`: Specify a numeric value. Supports integers and floats.

This syntax selects the eligible torrents directly, and removes them directly or works with the following compound expressions. Here is an example, it removes the torrents which seeding time is greater than 259200 seconds:

```
my_task:
  client: xxx
  host: xxx
  username: xxx
  password: xxx
  strategies:
    my_strategy:
      remove: seeding_time > 259200
```

2. `<Expression 1> and <Expression 2>` and `<Expression 1> or <Expression 2>`

   This syntax is a compound expression.

- `and`: Select torrents that meet both the `Expression 1` and `Expression 2` (intersection).
- `or`: Select torrents that meet one or both of the `Expression 1` and `Expression 2` (Union).

Here is an example. For all torrents, it removes those torrents which ratio is greater than 2 **and** seeding time is more than 60000 seconds:

```
my_task:
  client: xxx
  host: xxx
  username: xxx
  password: xxx
  strategies:
    my_strategy:
      remove: ratio > 2 and seeding_time > 60000
```

Here is another example. For all torrents, it removes those torrents which ratio is less than 1 **or** seeding time is more than 60000:

```
my_task:
  client: xxx
  host: xxx
  username: xxx
  password: xxx
  strategies:
    my_strategy:
      remove: ratio < 1 or seeding_time > 60000
```

3. `(<Expression>)`

   When an expression is enclosed in parentheses, it is still an expression. Using parentheses can change the priority. And you can use multiple parentheses for nesting.

   Here is an example. For all torrents, it removes those torrents which seeding time is more than 60000 seconds, **or** those torrents which ratio is greater than 3 **and** added time is more than 1400000 seconds:

```
my_task:
  client: xxx
  host: xxx
  username: xxx
  password: xxx
  strategies:
    my_strategy:
      remove: seeding_time > 60000 or (ratio > 3 and create_time > 1400000)
```

3.4 Part 4: Delete data

Determine whether to delete data at the same time. If this field isn't specificed, the default value is `false`.

3.5 The Last Step...

Remember to check your configuration file and make sure it works as you think. Use the following command line to see the torrents that will be removed (but not really remove them).

```
autoremove-torrents --view
```

CHAPTER 4

Indices and tables

- genindex
- modindex
- search