

---

# autoremove-torrents

发布 1.5.4

2022 年 06 月 25 日



---

目录:

---

<b>1</b>	<b>介绍</b>	<b>1</b>
1.1	支持的客户端 . . . . .	1
1.2	支持的属性 . . . . .	1
<b>2</b>	<b>安装与运行</b>	<b>3</b>
2.1	安装 . . . . .	3
2.2	运行 . . . . .	3
2.3	卸载 . . . . .	4
<b>3</b>	<b>配置</b>	<b>7</b>
3.1	第一部分: 任务名称 . . . . .	8
3.2	第二部分: 登录信息 . . . . .	8
3.3	第三部分: 策略块 . . . . .	9
3.4	第四部分: 删除数据 . . . . .	19
3.5	最后一步 . . . . .	19
<b>4</b>	<b>更新日志</b>	<b>21</b>
4.1	版本 1.5.4 . . . . .	21
4.2	版本 1.5.3 . . . . .	22
4.3	版本 1.5.2 . . . . .	22
4.4	版本 1.5.1 . . . . .	23
4.5	版本 1.5.0 . . . . .	23
4.6	文档迁移 . . . . .	24
4.7	版本 1.4.9 . . . . .	24
4.8	版本 1.4.8 . . . . .	24
4.9	版本 1.4.7 . . . . .	24
4.10	版本 1.4.6 . . . . .	25
4.11	版本 1.4.5 . . . . .	25

4.12	版本 1.4.4 . . . . .	25
4.13	版本 1.4.3 . . . . .	26
4.14	版本 1.4.2 . . . . .	26
4.15	版本 1.4.1 . . . . .	26
4.16	Wiki 更新 . . . . .	26
4.17	版本 1.4.0 . . . . .	26
4.18	版本 1.3.0 . . . . .	27
4.19	版本 1.2.5 . . . . .	27
4.20	版本 1.2.4 . . . . .	28
4.21	版本 1.2.3 . . . . .	28
4.22	版本 1.2.2 . . . . .	28
4.23	版本 1.2.1 . . . . .	29
4.24	版本 1.2.0 . . . . .	29
4.25	版本 1.1.0 . . . . .	29
4.26	文档勘误 . . . . .	29
4.27	第一个版本 . . . . .	29
<b>5</b>	<b>索引和表格</b>	<b>31</b>

这个程序是一个可以帮助您自动删除种子的工具。现在，你不再需要担心你的磁盘空间——根据你的策略，程序可以检查每个种子是否满足删除条件；如果是，那就自动删掉它。

让我们从这里开始。通过阅读文档，我们可以知道如何安装、配置和运行这个工具。

### 1.1 支持的客户端

截至目前，程序支持 qBittorrent/Transmission/ $\mu$ Torrent/Deluge。rTorrent 正在计划之中。

客户端	支持情况
qBittorrent	是
Transmission	是
$\mu$ Torrent	是
Deluge	是
rTorrent	计划中

### 1.2 支持的属性

✓ = 支持    ✗ = 不支持

属性/客户端	Deluge	qBittorrent	Transmission	$\mu$ Torrent
平均下载速度	✓ 2.0 或更高版本	✓	✓	
平均上传速度	✓	✓	✓	
类别	✓ 需要 Label 插件	✓	✓ 3.00+ 或更高版本	✓
已连接的下载者	✓	✓	✓	✓
已连接的做种者	✓	✓	✓	✓
创建时间	✓	✓	✓	
下载速度	✓	✓	✓	✓
下载量	✓	✓	✓	✓
空闲空间	✓	✓	✓	
最近活动	✓ 2.0 或更高版本	✓ v3.0 或更高版本	✓	
下载者	✓	✓	✓	✓
进度（百分比）	✓	✓	✓	✓
分享率	✓	✓	✓	✓
做种者	✓	✓	✓	✓
做种时间	✓	✓	✓	✓
种子大小	✓	✓	✓	✓
空闲状态（Stall）		✓	✓	
状态	✓	✓	✓	✓
Tracker	✓	✓	✓	✓
上传比率	✓	✓	✓	✓
上传速度	✓	✓	✓	✓
上传量	✓	✓	✓	✓

## 2.1 安装

有两个方法可以安装 `autoremove-torrents`。强烈推荐使用 `pip` 安装。

### 2.1.1 从 `pip` 安装

```
pip install autoremove-torrents
```

### 2.1.2 从 GitHub 安装

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

## 2.2 运行

只需在你的终端中输入以下命令：

```
autoremove-torrents
```

`autoremove-torrents` 会在当前工作目录中寻找 `config.yml` 文件。有关更多命令行参数，请查看下面的表格。

## 2.2.1 参数列表

---

**注解：** 如果你使用参数的全名，你需要用等号来引出参数的值；不过，如果你使用参数的缩写，你只需要用一个空格把参数的值引出来。

---

参数名	参数缩写	描述
<code>-view</code>	<code>-v</code>	运行并查看有哪些种子可以删除，但不要真正地删除它们。
<code>-conf</code>	<code>-c</code>	指定配置文件的路径。
<code>-task</code>	<code>-t</code>	运行指定的任务。参数值就是要执行的任务名。
<code>-log</code>	<code>-l</code>	指定日志文件的路径。
<code>-debug</code>	<code>-d</code>	Enable debug mode and output more logs.

例如：

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

它等价于：

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

## 2.3 卸载

### 2.3.1 通过 pip 卸载

如果你的 `autoremove-torrents` 是通过 `pip` 安装的，你就可以用 `pip` 简单地卸载掉它：

```
pip uninstall autoremove-torrents
```

### 2.3.2 手动卸载

然而，如果你的 `autoremove-torrents` 是通过 `setup.py` 安装的，你需要手动地删除所有的文件。



### 第一步

```
cd autoremove-torrents
```

### 第二步

重新安装这个程序并记录复制了哪些文件：

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

### 第三步

用 `xargs` 去删除文件：

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

或者，如果你在用 Windows，可以使用 Powershell：

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

参考：<https://stackoverflow.com/questions/1550226/python-setup-py-uninstall>



## CHAPTER 3

---

### 配置

---

在我们运行 `autoremove-torrents` 之前，我们需要创建一个 `config.yml` 文件，它可以用来保存配置。

**警告：** 为了防止误删种子，我们非常建议您在修改了配置文件之后，运行一次 `autoremove-torrents --view` 以预览运行结果。

本脚本使用 YAML 语言作为配置文件的语言。YAML 语言结构清晰，我认为它比 JSON 更友好，也更易于学习。

请看下面的例子，这个任务块可以被分成 3 部分。

```
# 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:
```

(下页继续)

(续上页)

```
- 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)
delete_data: true

# Add more tasks here...
```

当然，一个配置文件可以包含多个任务块，并且每个任务块还可以包含多个策略块。每个任务块代表一个 BT 客户端，每个策略块代表一种种子。

## 3.1 第一部分：任务名称

给任务取一个名字。

---

**注解：** 任务名前面不能有空格。

---

## 3.2 第二部分：登录信息

这部分是您的登录信息。

### 3.2.1 对于 qBittorrent、Transmission、 $\mu$ Torrent

对于 qBittorrent、Transmission 或  $\mu$ Torrent，本程序借助客户端的 WebUI 工作。

- **client**: 客户端类型，不区分大小写。
- **host**: WebUI 的访问地址。该地址必须指明一个协议 (<http://> 或 <https://>)。
- **username**: WebUI 的用户名。
- **password**: WebUI 的密码。

### 3.2.2 对于 Deluge

本程序通过 RPC 协议访问 Deluge。

- `client`: 客户端类型, 这里填 Deluge 即可。
- `host`: Deluge Daemon 的地址 (或域名) 以及端口, 例如 127.0.0.1:58846。
- `username`: Deluge Daemon 的用户名。
- `password`: Deluge Daemon 的密码。

例:

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

注解:

1. 不要在 `host` 字段中填写任何协议, 本程序既不使用 HTTP 协议访问 Deluge, 也不用 HTTPS 协议。
2. 端口号是 Deluge Daemon 的端口号, 不是 WebUI 的端口号。可以在 WebUI 中的 Connection Manager 里看到端口号。
3. 如果你的 autoremove-torrents 和 Deluge 运行在不同的计算机上, 请确保您的 Deluge 允许接受远程计算机的连接。您可以在 **Preferences -> Daemon -> Allow Remote Connections** 中修改这个设置。

**注解:** 一般来说, 您可以在 `~/.config/deluge/auth` 中找到用户名和密码。您也可以在文件末尾加一行以添加一个用户。

有关身份验证方面的更多信息, 请访问 <https://dev.deluge-torrent.org/wiki/UserGuide/Authentication>。

## 3.3 第三部分: 策略块

这个部分是策略块。每个策略块也可以分为 3 个部分。

### 3.3.1 第 1 部分: 策略名称

就像给任务取名一样, 给你的策略也取个名字。

### 3.3.2 第 2 部分：过滤器

当前策略只会对你所选择的种子有效。有 9 个过滤器可以用。

- `all_trackers/all_categories/all_status`: 选择所有的 Tracker/分类/种子状态。
- `categories`: 选择这些分类的种子。
- `excluded_categories`: 排除这些分类的种子。
- `trackers`: 选择这些 Tracker 的种子。
- `excluded_trackers`: 排除这些 Tracker 的种子。
- `status`: 选择这些状态的种子，可以选择的状态如下表：

状态	备注
Downloading	/
Uploading	/
Checking	/
Queued	/
Paused	Transmission 无此状态。
Stopped	qBittorrent 无此状态。
Error	/
StalledUpload	$\mu$ Torrent 无此状态。
StalledDownload	$\mu$ Torrent 无此状态。

- `excluded_status`: 排除这些状态的种子，可选的状态如上表。

每个过滤器的结果都是一个种子的集合。

---

**注解：** 如果 `categories`、`trackers`、`status` 三个过滤器的其中两个或者三个同时存在，则程序会取这些集合的交集，并减去 `excluded_categories` 集合、`excluded_trackers` 集合和 `excluded_status` 集合。

---

**注解：**

1. 不要在 `trackers` 中写套接字，`trackers` 字段只需要填主机名。例如 `https://tracker.site1.com` 只填写 `tracker.site1.com`。
2. 在 1.4.4 以及以后的版本中，如果 `categories`、`trackers` 或 `status` 中的内容只有一项，可以不使用列表，写一个单行文本就行，例如：

```
categories: catal
```

```
status: uploading
```

3. StalledUp 与 StalledDown 为 1.4.5 版本新增的状态。在本程序中, Uploading 包含 StalledUpload 状态的种子, Downloading 包含 StallDownload 状态的种子。

让我们先看一些例子。例如, 选择分类是 Movies 或 Games 的种子:

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

选择 Tracker 主机名是 tracker.aaa.com 或 x.bbb.com 的种子:

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

选择分类是 Movies 或 Games, 但 Tracker 不是 tracker.yyy.com 的种子:

```
my_task:
  client: xxx
  host: xxx
  username: xxx
```

(下页继续)

(续上页)

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

选择分类是 Movies 而且状态是正在上传的种子:

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

### 3.3.3 第三部分：删除条件

有两种设置删除条件的方法。

#### 1. 直接使用删除条件关键词（推荐）

直接使用删除条件关键词，有 18 个删除条件。

---

**注解：** 只要选择的种子满足这些条件之一，它就会被删除。

---

前 18 个条件如下。为了防止误删种子，某些条件只对特定的状态有效。



条件	单位	可用状态	说明
ratio		全部	分享率上限
create_time	秒	全部	种子从添加到客户端到现在所经过的时间，以秒为单位。在这里设置一个添加时间的上限，当种子在客户端中存活时间超过此上限后，种子会被直接删除（不管种子现在正处于哪种状态）。
downloading_time	秒	全部	最长下载时间。
seeding_time	秒	全部	最长做种时间。
max_download	GiB	全部	种子最大下载量，超过此限制的种子会被删除。
max_downloadspeed	KiB/s	Down-loading	种子的最大下载速度，超过此限制的种子会被删除。
min_uploadspeed	KiB/s	Down-loading 或 Up-loading	种子的最小上传速度，低于此速度的种子会被删除。
max_average_download_speed	KiB/s	全部	最大平均下载速度，如同 max_downloadspeed。
min_average_upload_speed	KiB/s	全部	最小平均上传速度，如同 min_uploadspeed。
max_size	GiB	全部	种子大小限制，超过此限制的种子会被删除。
max_seeder		全部	最大做种者数，超过此限制的种子会被删除。
max_upload	GiB	全部	种子的最大上传体积，超过此限制的种子会被删除。
min_leecher		全部	最小下载者数，低于此设置的种子会被删除。
max_connected_seeder		Down-loading 或 Up-loading	已连接做种者的最大值，如同 max_seeder。
min_connected_leecher		Down-loading 或 Up-loading	已连接下载者数的最小值，如同 min_leecher。
last_activity	秒	全部	种子从停止活动到现在所允许经过的最长时间，即允许的没有上传或下载活动的最长时间。当种子到达此限制时，它就会被删除。
max_progress	百分比 (%)	全部	允许的最大下载进度，最大值是 100。
upload_ratio		全部	最大上传比率。注意，此处的上传比率与分享率不同，对于每个种子来说，上传比率是指上传量除以种子大小的比值。

**注解：** 对于 1.5.4 或以上版本，last\_activity 的默认行为已经改变。在默认情况下，last\_activity 只考

考虑曾经活动过的种子；而对于从未活动过的种子，`last_activity` 不会删除它们。

此外，若需要删除从未活动过的种子，请使用 `last_activity: Never` 或者 `last_activity: None`。

---

除了上面的删种条件，这里还有 3 个删种条件。当剩下的种子触发这些删种条件时，它们就会被删除。

- **seed\_size**: 计算上述选择的种子的总大小。如果总大小超过限制，一部分种子会被删除。需要设置以下两个属性：
  - **limit**: 总大小限制，以 GiB 为单位。
  - **action**: 确定哪部分种子将被删除。可以是以下值：

值	说明
remove-old-seeds	尽量删除旧的种子。
remove-new-seeds	尽量删除新的种子。
remove-big-seeds	尽量删除体积大的种子。
remove-small-seeds	尽量删除体积小的种子。
remove-active-seeds	尽量删除活跃的种子。
remove-inactive-seeds	尽量删除不活跃的种子。
remove-fast-upload-seeds	尽量删除上传速度快的种子。
remove-slow-upload-seeds	尽量删除上传速度慢的种子。

---

**注解:** 与 `last_activity` 类似，`remove-active-seeds` 与 `remove-inactive-seeds` 动作将优先考虑曾经活动过的种子。只有在这些种子都被删除但限制条件仍然没有被满足的情况下，从未活动过的种子才会被删除（但不保证删除顺序）。

---

- **maximum\_number**: 判断上述选择的种子的总个数。如果个数超出设置的最大值，一部分种子会被删除，就像 `seed_size` 条件一样。需要设置以下两个属性：
  - **limit**: 总个数限制
  - **action**: 确定哪些种子会被删除，取值与含义同上表。
- **free\_space**: 检查磁盘的剩余空间是否符合设定。如果空间不足，一部分种子会被删除，就像 `seed_size` 条件一样。需要设置以下三个属性：
  - **min**: 剩余空间最小值，以 ‘GiB’ 为单位；当目录的剩余空间小于这个值时，删除策略会被触发。
  - **path**: 需要监控剩余空间的目录
  - **action**: 删除策略，用于确定哪些种子会被删除。取值与含义同上表
- **remote\_free\_space**: 同样也是根据磁盘剩余空间来决定哪些种子会被删除，但这里使用 BT 客户端汇报的磁盘剩余空间大小。它的行为与 `free_space` 一致。
  - **min**: 允许的剩余空间最小值，以 *GiB* 为单位。

- path: BT 客户端需要检查剩余空间大小的目录路径
- action: 删除策略

**注解:** 如果你的 autoremove-torrents 和你的 BT 客户端运行在不同的机器上, 你就需要使用 `remote_free_space` 去查询剩余空间了。除此之外, `free_space` 和 `remote_free_space` 是一样的。

请注意, 并不是所有的客户端都支持检测特定目录的剩余空间。截至目前, 只有 Deluge 和 Transmission 支持查询特定目录; 若在 qBittorrent 中使用 `remote_free_space` 的 `path` 参数, 它将会被忽略。

这是一个例子。对于分类是 xxx 或 yyy 的种子, 分享率大于 1 或者做种时间大于 1209600 秒的种子会被删除:

```
my_task:
  client: xxx
  host: xxx
  username: xxx
  password: xxx
  strategies:
    my_strategy:
      categories:
        - xxx
        - yyy
      ratio: 1
      seeding_time: 1209600
```

这也是一个例子。对于全体种子, 做种时间大于 259200 秒的种子会被删除:

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

这还有一个例子。对于全体种子, 当 `/home/myserver/downloads` 的剩余空间小于 10GiB 时, 程序会尝试删除体积大的种子:

```
my_task:
  client: xxx
```

(下页继续)

(续上页)

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

这是最后一个例子。对于全体种子，首先删掉那些分享率大于 3 的种子，然后如果剩下的种子的总大小还大于 500GiB 就尽量删除一些活跃的种子，直到总大小小于 500GiB 为止：

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

## 2. 使用 remove 关键词（高级）

使用 `remove` 关键词。`remove` 关键词是在 1.4.0 版本中新增的关键词，用于支持复杂的删除条件的设置。`remove` 关键词后接一个表达式，表达式由以下语法构成：

1. `< 参数 > < 比较运算符 > < 数值 >`

参数：可选列表如下，不区分大小写。

---

**注解：** 某些属性只能在特定的状态中使用，不在可用状态的种子不会被删除。

---

参数	单位	可用状态	说明
average_downloadspeed	KiB/s	全部	平均下载速度。
average_uploadspeed	KiB/s	全部	平均上传速度。
connected_leecher	/	Downloading 或 Uploading	已连接的下载者数。
connected_seeder	/	Downloading 或 Uploading	已连接的做种者数。
create_time	秒	全部	种子从添加到客户端到现在所经过的时间。
download	GiB	全部	下载量
download_speed	KiB/s	Downloading	下载速度
downloading_time	秒	全部	下载时间。
last_activity	秒	全部	种子从停止活动（无上传或下载）到现在所经过的时间。
leecher	/	全部	下载者数。
progress	%	全部	下载进度。
ratio	/	全部	分享率
seeder	/	全部	做种者数。
seeding_time	秒	全部	做种时间
size	GiB	全部	种子大小。
upload	GiB	全部	上传量
upload_ratio	/	全部	上传量 / 种子大小
upload_speed	KiB/s	Downloading 或 Uploading	上传速度

比较运算符：可用的运算符如下。

比较运算符	说明
<	小于
>	大于
=	等于

数值：指定一个数值，支持整数和小数。

该语法直接选出符合条件的种子，并直接删除它们或者与后面的复合表达式配合使用。这是一个例子，它直接删除做种时间大于 259200 秒的种子：

```
my_task:
  client: xxx
  host: xxx
  username: xxx
```

(下页继续)

(续上页)

```
password: xxx
strategies:
  my_strategy:
    remove: seeding_time > 259200
```

## 2. < 表达式 1> and < 表达式 2> 以及 < 表达式 1> or < 表达式 2>

该语法是复合表达式。

- and: 选择同时满足 表达式 1 和 表达式 2 的种子 (交集)。
- or: 选择满足 表达式 1 或者 表达式 2 或者同时满足两个表达式的种子 (并集)。

这是一个例子。对于全体种子, 分享率超过 2 而且做种时间超过 60000 秒的种子会被删除:

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

这是另外一个例子。对于全体种子, 分享率小于 1 或者做种时间超过 60000 秒的种子会被删除:

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

## 3. (< 表达式 >)

一个表达式加上括号以后, 它仍然是表达式。使用括号可以改变优先级。可以使用多重括号嵌套。

这是一个例子。对于全体种子, 做种时间超过 60000 秒的种子, 或者分享率大于 3 而且添加时间超过 1400000 秒的种子会被删除:

```
my_task:
  client: xxx
  host: xxx
```

(下页继续)

(续上页)

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

## 3.4 第四部分：删除数据

决定是否在删除种子的同时也删除数据。如果此字段未指定，则默认值为 `false`。

## 3.5 最后一步……

记得检查你的配置文件，确保它按照你的想法工作（不然可能会误删种子了）。使用以下命令行以查看将要删除的种子，但不会真正地删除它们：

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





## 4.1 版本 1.5.4

发布日期：2022 年 6 月 19 日星期日

### 4.1.1 变动

- 移除端口开放状态 (Outgoing Port Status) 的信息。 (#101) (#135)
  - 我们确认了一个 bug，即，当我们使用 Transmission 并在 IPv6 网络下检查端口开放状态时，Transmission 的端口检查器会报错并提示 “portTested: http error 400: Bad Request”。
  - 由于没有删种条件需要依赖这个端口开放状态，所以我们删了它。
- 修改了 last\_activity 的行为。 (#93) (#98) (#109)
  - 默认情况下，last\_activity 不再删除那些从未活跃过的种子。
  - 在需要的情况下，这些从未活跃过的种子可以用以下方式去删除：
    - \* 对于 last\_activity 条件，使用 last\_activity:never 或者 last\_activity: none。
    - \* 对于 remove 表达式，使用 last\_activity = never 或者 last\_activity = none。

### 4.1.2 新功能

- 在 `action` 关键字中添加了 `remove-slow-upload-seeds` 和 `remove-fast-upload-seeds` 两个动作。  
(#127) 感谢 @vincent906!
- `remove` 表达式中支持等号 (=)。
- 添加 `downloading_time` 条件。( #88) 感谢 @dantebalba!

### 4.1.3 修复

- 修复了上传/下载量以及 `free_space`、`remote_free_space` 不能正确处理小数的问题。( #133) 感谢 @sfwn!
- 修复了 `last_activity` 条件在 Deluge 2.0.3 及以上版本无效的问题。( #119)

## 4.2 版本 1.5.3

发布日期: 2020 年 8 月 27 日星期四

### 4.2.1 新功能

- 可以通过 `--debug` 或 `-d` 命令行启用调试模式。( #76)
- 添加上传量与下载量两个条件。( #79)

### 4.2.2 修复

- 修复 `psutil` 在群晖的兼容问题 (用于检查磁盘剩余空间)。( #61)
- 修复由主机 URL 末尾的 / 导致的 API 不兼容的问题。( #81)

## 4.3 版本 1.5.2

发布日期: 2020 年 3 月 27 日星期五

### 4.3.1 变动

- 客户端名称不再对大小写敏感。

### 4.3.2 新功能

- 支持 Deluge。 (#8)
- 使用批量删除提升删除效率。

### 4.3.3 修复

- 修复配置文件中的多语言支持问题。 (#69)

## 4.4 版本 1.5.1

发布日期: 2020 年 2 月 29 日星期六

### 4.4.1 修复

- 修复了 1.5.0 版本中丢失的状态 StalledUpload 和 StalledDownload。 (#66)

## 4.5 版本 1.5.0

发布日期: 2020 年 2 月 28 日星期五

### 4.5.1 新功能

- status 中添加 Error 状态。
- 添加对 Transmission 标签的支持 (#24)。
- 添加删除条件 “最大下载速度” max\_downloadspeed、“最小上传速度” min\_uploadspeed。
- 添加删除条件 “最小平均上传速度” min\_average\_uploadspeed、“最大平均下载速度” max\_average\_downloadspeed (#49)。
- 添加删除条件 “最大种子大小” max\_size (#21)。
- 添加删除条件 “最大做种数” max\_seeder、“最小下载数” min\_leecher (#62)。
- 添加删除条件 “最大已连接做种者” max\_connected\_seeder、“最小已连接下载者” min\_connected\_leecher。
- 添加删除条件 “最后活动时间” last\_activity, 以删除一段时间内没有上传或下载速度的种子 (#1) (#9)。
- 添加删除条件 “最大下载百分比” max\_progress。

- `free_space`、`maximum_number`、`seed_size` 的 `action` 中添加 `remove-active-seeds`、`remove-inactive-seeds` 动作，用于根据最后活动时间去尽量删除活跃的种子或者不活跃的种子 (#9)。
- 添加了新的删除条件“上传比率” `upload_ratio`，可以根据上传量占种子大小的比例删种 (#55)。

### 4.5.2 修复

- 修复了无法登录使用纯数字用户名或密码的客户端的问题。 (#64)
- 修复了在没有标签属性的 Transmission 中任务无法执行的问题。
- 修复了删种条件可能对未打标签以及没有 Tracker 的种子无效的问题。
- 修复了 status 中遗漏的  $\mu$ Torrent 状态“排队中” (Queued)。

## 4.6 文档迁移

日期：2020 年 2 月 3 日星期一

- 迁移文档到 Read the Docs。

## 4.7 版本 1.4.9

发布日期：2020 年 1 月 26 日星期日

### 4.7.1 新功能

- 添加了 `free_space` 条件（最小剩余空间）。感谢 @drawwon!

## 4.8 版本 1.4.8

发布日期：2020 年 1 月 7 日星期二

### 4.8.1 修复

- 修复了在 qBittorrent v4.2.1 中不能删除种子的问题。对造成的不便深感抱歉。 (#53)

## 4.9 版本 1.4.7

发布日期：2020 年 1 月 6 日星期一

### 4.9.1 新功能

- 添加了 qBittorrent v4.2.1 中新 API 的支持。(#46)

---

**注解：** 注意：这个版本有 bug，请升级到 1.4.8 或者更高的版本。

---

## 4.10 版本 1.4.6

发布日期：2019 年 9 月 17 日星期二

### 4.10.1 修复

- 修复了当 Tracker 的 URL 包含端口时，*tracker* 过滤器需要指定端口的问题。

## 4.11 版本 1.4.5

发布日期：2019 年 6 月 6 日星期四

### 4.11.1 新功能

- 添加了 *StalledUpload* 与 *StalledDownload* 状态。

## 4.12 版本 1.4.4

发布日期：2019 年 5 月 22 日星期三

### 4.12.1 新功能

- 新特性：如果过滤器的内容只有一行，现在可以直接写出而不需要使用列表。

### 4.12.2 修复

- 修复了当 *seed\_size* / *maximum\_number* 条件与 *ratio* / *create\_time* / *seeding\_time* 条件同时使用时任务会失败的问题。(#33)

## 4.13 版本 1.4.3

发布日期: 2019 年 5 月 19 日星期日

### 4.13.1 变动

- 添加对 Python 2.7 的支持。(#29)
- 停止支持 Python 3.4。([kennethreitz/requests#5092](#))

## 4.14 版本 1.4.2

发布日期: 2019 年 5 月 13 日星期一

### 4.14.1 修复

- 修复了丢失的语法分析器文件。(#32)
- 修复了运算符的结合性。现在, 运算符 *and* 和 *or* 保证是左结合的。(#32)

## 4.15 版本 1.4.1

发布日期: 2019 年 5 月 6 日星期一

### 4.15.1 修复

- 修复了丢失的依赖项 *ply*。
- 修复了在 *remove* 条件中重复定义的警告。

## 4.16 Wiki 更新

日期: 2019 年 5 月 6 日星期一

- 增加了关于 *remove* 条件的描述。

## 4.17 版本 1.4.0

发布日期: 2019 年 5 月 1 日星期三

### 4.17.1 变动

- 删除了 `seeding_time` 和 `ratio` 条件中的限制 (#19)。
  - 在之前的版本, `seeding_time` 和 `ratio` 条件只会删除那些正在做种的种子。设置这个限制是为了给用户提供一个通过修改种子的状态 (例如暂停做种) 来避免种子被删除的方法。
  - 不过现在我们有状态过滤器 (`status`), 所以这个限制就显得多余了, 而且可能会使它的行为跟用户预想的不一樣。

### 4.17.2 新功能

- 支持自定义删除表达式 (#15)。
  - 现在我们可以直接而明确地写出我们想要的表达式了, 例如 `remove: ratio > 1`。
  - 复合的条件表达式也支持, 例如 `remove: (seeding_time < 86400 and ratio > 1) or (seeding_time > 86400 and ratio > 3)`。
  - 旧的写法仍然可用。

## 4.18 版本 1.3.0

发布日期: 2019 年 4 月 17 日星期三

### 4.18.1 变动

- 日志系统已更新:
  - 日志路径可以被指定 (使用 `--log` 参数, 例如 `--log=/home/jerrymakesjelly/logs`) (#23)。
  - 日志文件将按天存储在不同的文件中 (格式: `autoremove.%Y-%m-%d.log`)。
- 全部单词 `seed` 修改为 `torrent` (#25)。
- 删去了所有不必要的调试日志。

### 4.18.2 修复

- 修复了在 qBittorrent 拥有大量的种子时程序会卡住的问题 (#22)。
- 修复了状态过滤器在工作时会写入重复的日志的问题。

## 4.19 版本 1.2.5

发布日期: 2019 年 1 月 10 日星期一

### 4.19.1 修复

- 修复了在设置多个策略时种子数量不正确的问题 (#10)。感谢 @momokoo!
- 修复了在 qBittorrent 中种子数不正确的问题 (#13)。

## 4.20 版本 1.2.4

发布日期: 2018 年 5 月 31 日星期四

### 4.20.1 修复

- 修复了启动失败的问题。

## 4.21 版本 1.2.3

发布日期: 2018 年 5 月 30 日星期三

### 4.21.1 变动

- 现在程序在一个任务失败时不会直接退出。

### 4.21.2 新功能

- 允许使用环境变量去指定 host、username 和 password。
- 允许 username 和 password 留空 (或者其中之一留空), 使得不用用户名或密码也可以登录 WebUI。

## 4.22 版本 1.2.2

发布日期: 2018 年 5 月 27 日星期日

### 4.22.1 新功能

- 增加了新过滤器: 种子状态。
- 增加了新条件: 最大种子数量。



## 4.23 版本 1.2.1

发布日期: 2018 年 5 月 26 日星期六

### 4.23.1 修复

- 修复了 `setup.py` 的问题。

## 4.24 版本 1.2.0

发布日期: 2018 年 5 月 26 日星期六

- 程序已发布至 PyPI!
- 重构已完成。
  - 新特性很快会被加入。
  - 现在你可以通过 `pip` 安装程序。

## 4.25 版本 1.1.0

发布日期: 2018 年 5 月 14 日星期一

- 创建了 `setup.py`。
  - 现在你可以直接使用 `autoremove-torrents` 命令而不是 `python3 main.py`。

## 4.26 文档勘误

日期: 2018 年 3 月 28 日星期三

- `delete_data` 字段不应该被缩进。

## 4.27 第一个版本

发布日期: 2018 年 3 月 22 日星期四

- 第一个版本:bowtie:



## CHAPTER 5

---

### 索引和表格

---

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