

# rotateloggs详解

更新：2023-05-22 15:25

## 一、概述

Rotateloggs是Apache log4j的一个工具。它是一个可执行文件，可用于将Web服务器的访问日志根据日期、大小、指定的时间间隔进行轮转。它主要用于长期记录某些内容以进行数据统计，也可以通过监控日志文件来错误诊断。

rotateloggs命令如果系统中未安装，在ubuntu中可以使用 `sudo apt install apache2-utils` 命令安装，而在centos7中则包含在httpd的安装包中，使用`yum install httpd`安装后可以获得rotateloggs命令行工具和Apache HTTPD服务器。

## 二、常用参数

Rotateloggs命令格式：`rotateloggs logfile rotationtime [options]`

其中logfile表示日志文件的路径，rotationtime则是轮转周期。常用参数如下：

- `-l`：以链接方式创建新的日志文件
- `-n`：设置保存日志文件的最大数量（默认是10）
- `-f`：强制覆盖已存在的日志文件
- `-v`：打印更详细的信息
- `-e`：为截断的日志消息添加一个error消息行

## 三、日志轮转方式

在rotateloggs中，日志轮转可以通过多种方式进行，如：

### 1、按大小轮转

```
rotatelog.exe access.log 5M
```

上述命令每5M生成一个新日志文件，并将旧的日志文件重命名，并添加日期时间后缀。

## 2、按日期轮转

```
rotatelog.exe access.log %Y%m%d
```

上述命令以日期为轮转周期（如20220101），生成新日志文件，并把旧的日志文件增加日期每天的后缀。这样可以保留七天、一个月或更长时间的日志。

## 3、按时间间隔轮转

```
rotatelog.exe access.log 86400
```

上述命令以每86400秒为周期生成新文件，同时archive时间戳的格式为%Y-%m-%d\_%H:%M:%S。

## 四、实践应用

以按大小轮转方式为例，下面是一个示例：

```
SetEnvIf Request_URI "^.*$" req_uri=$0  
CustomLog "|bin/rotatelog.exe C:\logs\access_%Y%m%d.log 5M" common env=req_uri
```

上述配置将访问日志按照大小为5M进行轮转，每个log文件的最大数量为10个。并且通过CustomLog设定后，根据URL请求的改变每次访问都会被记录为新日志。

## 五、注意事项

在使用rotatelog过程中，需要注意以下问题：

- 有时可能需要重启服务器才能应用更改后的设置

- 文件名必须以英文命名，否则会出现错误
- 必须为此执行文件指定完整路径
- 与其他日志记录工具如apache,nginx一起使用需要注意权限问题

## 总结

Rotatelog是一个非常实用的工具，主要用于Web服务器的日志记录和日志轮转。不仅可以按照日期、大小、指定时间轮转，还支持链接等轮转方式。使用rotatelog可以方便地记录和查看服务器访问日志，从而快速找到问题所在。

# Apache HTTP Server Version 2.4

## rotatelog - Piped logging program to rotate Apache logs

rotatelog is a simple program for use in conjunction with Apache's piped logfile feature. It supports rotation based on a time interval or maximum size of the log.

### Synopsis

---

```
rotatelog [ -l ] [ -L linkname ] [ -p program ] [ -f ] [ -D ] [ -t ] [ -v ] [ -e ] [ -c ] [ -n number-of-files ] logfile  
rotationtime|filesize(B|K|M|G) [ offset ]
```

### Options

---

**-l**

Causes the use of local time rather than GMT as the base for the interval or for `strftime(3)` formatting with size-based rotation.

**-L** *linkname*

Causes a hard link to be made from the current logfile to the specified link name. This can be used to watch the log continuously across rotations using a command like `tail -F linkname`.

If the linkname is not an absolute path, it is relative to rotatelog's working directory, which is the `ServerRoot` when rotatelog is run by the server.

**-p** *program*

If given, rotatelog will execute the specified program every time a new log file is opened. The filename of the newly opened file is passed as the first argument to the program. If executing after a rotation, the old log file is passed as the second argument. rotatelog does not wait for the specified program to terminate before continuing to operate, and will not log any error code returned on termination. The spawned program uses the same stdin, stdout, and stderr as rotatelog itself, and also inherits the environment.

**-f**

Causes the logfile to be opened immediately, as soon as rotatelog starts, instead of waiting for the first logfile entry to be read (for non-busy sites, there may be a substantial delay between when the server is started and when the first request is handled, meaning that the associated logfile does not "exist" until then, which causes problems from some automated logging tools)

**-D**

Creates the parent directories of the path that the log file will be placed in if they do not already exist. This allows `strftime(3)` formatting to be used in the path and not just the filename.

**-t**

Causes the logfile to be truncated instead of rotated. This is useful when a log is processed in real time by a command like tail, and there is no need for archived data. No suffix will be added to the filename, however format strings containing '%' characters will be respected.

**-T**

Causes all but the initial logfile to be truncated when opened. This is useful when the format string contains something that will loop around, such as the day of the month. Available in 2.4.56 and later.

**-v**

Produce verbose output on STDERR. The output contains the result of the configuration parsing, and all file open and close actions.

e

Echo logs through to stdout. Useful when logs need to be further processed in real time by a further tool in the chain.

-c

Create log file for each interval, even if empty.

-n *number-of-files*

Use a circular list of filenames without timestamps. This option overwrites log files at startup and during rotation. With -n 3, the series of log files opened would be "logfile", "logfile.1", "logfile.2", then overwriting "logfile".

When this program first opens "logfile", the file will only be truncated if -t is also provided. Every subsequent rotation will always begin with truncation of the target file. For size based rotation without -t and existing log files in place, this option may result in unintuitive behavior such as initial log entries being sent to "logfile.1", and entries in "logfile.1" not being preserved even if later "logfile.n" have not yet been used.

Available in 2.4.5 and later.

*logfile*

The path plus basename of the logfile. If *logfile* includes any '%' characters, it is treated as a format string for `strftime(3)`. Otherwise, the suffix *.nnnnnnnnnn* is automatically added and is the time in seconds (unless the -t option is used). Both formats compute the start time from the beginning of the current period. For example, if a rotation time of 86400 is specified, the hour, minute, and second fields created from the `strftime(3)` format will all be zero, referring to the beginning of the current 24-hour period (midnight).

When using `strftime(3)` filename formatting, be sure the log file format has enough granularity to produce a different file name each time the logs are rotated. Otherwise rotation will overwrite the same file instead of starting a new one. For example, if *logfile* was `/var/log/errorlog.%Y-%m-%d` with log rotation at 5 megabytes, but 5 megabytes was reached twice in the same day, the same log file name would be produced and log rotation would keep writing to the same file.

If the logfile is not an absolute path, it is relative to `rotatelogs'` working directory, which is the `ServerRoot` when `rotatelogs` is run by the server.

*rotationtime*

The time between log file rotations in seconds. The rotation occurs at the beginning of this interval. For example, if the rotation time is 3600, the log file will be rotated at the beginning of every hour; if the rotation time is 86400, the log file will be rotated every night at midnight. (If no data is logged during an interval, no file will be created.)

*filesize(B|K|M|G)*

The maximum file size in followed by exactly one of the letters B (Bytes), K (KBytes), M (MBytes) or G (GBytes).

When time and size are specified, the size must be given after the time. Rotation will occur whenever either time or size limits are reached.

*offset*

The number of minutes offset from UTC. If omitted, zero is assumed and UTC is used. For example, to use local time in the zone UTC -5 hours, specify a value of -300 for this argument. In most cases, -1 should be used instead of specifying an offset.

## Examples

---

```
CustomLog "|bin/rotatelogs /var/log/logfile 86400" common
```

This creates the files `/var/log/logfile.nnnn` where `nnnn` is the system time at which the log nominally starts (this time will always be a multiple of the rotation time, so you can synchronize cron scripts with it). At the end of each rotation time (here after 24 hours) a new log is started.

```
CustomLog "|bin/rotatelogs -l /var/log/logfile.%Y.%m.%d 86400" common
```

This creates the files `/var/log/logfile.yyyy.mm.dd` where `yyyy` is the year, `mm` is the month, and `dd` is the day of the month. Logging will switch to a new file every day at midnight, local time.

```
CustomLog "|bin/rotatelogs /var/log/logfile 5M" common
```

This configuration will rotate the logfile whenever it reaches a size of 5 megabytes.

```
ErrorLog "|bin/rotatelogs /var/log/errorlog.%Y-%m-%d-%H_%M_%S 5M"
```

This configuration will rotate the error logfile whenever it reaches a size of 5 megabytes, and the suffix to the logfile name will be created of the form `errorlog.YYYY-mm-dd-HH_MM_SS`.

```
CustomLog "|bin/rotatelogs -t /var/log/logfile 86400" common
```

This creates the file `/var/log/logfile`, truncating the file at startup and then truncating the file once per day. It is expected in this scenario that a separate process (such as `tail`) would process the file in real time.

```
CustomLog "|bin/rotatelogs -T /var/log/logfile.%d 86400" common
```

If the server is started (or restarted) on the first of the month, this appends to `/var/log/logfile.01`. When a log entry is written on the second of the month, `/var/log/logfile.02` is truncated and new entries will be added to the top. This example keeps approximately 1 months worth of logs without external maintenance.

## Portability

The following logfile format string substitutions should be supported by all `strftime(3)` implementations, see the `strftime(3)` man page for library-specific extensions.

%A	full weekday name (localized)
%a	3-character weekday name (localized)
%B	full month name (localized)
%b	3-character month name (localized)
%c	date and time (localized)
%d	2-digit day of month
%H	2-digit hour (24 hour clock)
%I	2-digit hour (12 hour clock)
%j	3-digit day of year
%M	2-digit minute
%m	2-digit month
%p	am/pm of 12 hour clock (localized)
%S	2-digit second
%U	2-digit week of year (Sunday first day of week)
%W	2-digit week of year (Monday first day of week)
%w	1-digit weekday (Sunday first day of week)
%X	time (localized)

%x	date (localized)
%Y	4-digit year
%y	2-digit year
%Z	time zone name
%%	literal `%'

## Comments

---

**Notice:**

This is not a Q&A section. Comments placed here should be pointed towards suggestions on improving the documentation or server, and may be removed by our moderators if they are either implemented or considered invalid/off-topic. Questions on how to manage the Apache HTTP Server should be directed at either our IRC channel, #httpd, on Libera.chat, or sent to our mailing lists.

---

**Copyright 2023 The Apache Software Foundation.  
Licensed under the Apache License, Version 2.0.**