NAME
> rotatelogs - Piped logging program to rotate Apache logs

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

SUMMARY
> **rotatelogs** 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.

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*
>
> **-p** *program*
>   If given, **rotatelogs** 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. **rotatelogs** 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 rotatelogs
>   itself, and also inherits the environment.
>
> **-f**   Causes the logfile to be opened immediately, as soon as **rotatelogs** 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

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). .PP 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, **-l** 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 '%'