Log file rotation - Howto

A "Howto" document for web server log file rotation with compression - Unix and Windows examples.

Why Rotate?

Urchin stores aggregated log file information in its own database enabling the end user to build 'real-time' visitor reports. With is own 'Log Tracker' keeping track of how far into your server log files it has processed, you could say Log file rotation can be ignored. However this isn't recommend for the following reasons:

- Disk Space - log files in use are uncompressed plain text and can consume large amounts of space. Typically compression will reduce file sizes by 20:1, so it make sense to do this. However, a web server log file can not be compressed while in use (locked), so it first must be rotated out of use.
- Periodically, you will need to check log files for warnings, error messages (scripts not working), Search Engine detection etc, and this is best done using manageable file sizes.
- Opening, closing and manipulating data for very large file sizes consumes system resources and will therefore slow down your server. It is much more efficient from both a system and application standpoint to manage several smaller logs than one very large log.
- Smaller files are much easier to back up and restore in the event of system failure.

Log file rotation is achieved quite simply on Unix machines using crontab and logrotate. We describe a separate method below for Windows.

System

This example was developed and tested on:

Unix: RedHat 6.x and 7.1/2/3 using Apache v1.3.9-29.
Windows: NT4/SP6, Windows 2000 SP1.

Schematic Unix Example

- Each night (or any set time period), Urchin runs on the selected log file*
- Urchin Log Tracker keeps an internal marker as to how far into the file it has analysed.

- Each month (or any set time period), rotate and compress log files
- Keep rotated log files for 12 months (or any number) - in case you need to re-analyse!

If you wish to rotate/compress files each and every time Urchin is scheduled to run, you can do this within Urchins' Log Manager. However, this can create a large number of log files, especially if you have multiple hosts/hosts.

Our log file rotation results in the following monthly files being created:

- httpd_log.1.gz, httpd_log.2.gz, httpd_log.3.gz etc...

In this example, apache is using the 'combined' log format described in httpd.conf e.g.

```
# The following directives define some format nicknames for use
# with a CustomLog directive (see below).
# LogFormat "%h %v %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Agent}i\" \"%{Cookie}i\"" combined
...
CustomLog /home/httpd/logs_dir/httpd_log combined
```

Note the **Logformat** directive maybe slightly different than what you see in some installations of apache. Ours is recommended as it allows you to simply use the default Urchin format: 'Log Format = auto'.

Unix Method

Each minute, the system crontab checks what jobs require scheduling. Scheduling is set in the /etc/crontab file.

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/
# column headings - thanks Toby
# mins, hr, date, month, day, command
# run-parts
# Min Hr Date Month Day Owner Command File
01 * * * * root run-parts /etc/cron.hourly
02 1 * * * root run-parts /etc/cron.daily
50 23 * * 0 root run-parts /etc/cron.weekly
01 00 1 * * root run-parts /etc/cron.monthly
```

and what jobs are to be run is described in for example /etc/cron.monthly. In the above example, the directory /etc/cron.monthly is checked at 00:01 on the first day of every month.

My /etc/cron.monthly directory contains a file logrotate, contents of which are:

```
#!/bin/sh
/usr/sbin/logrotate /etc/logrotate.conf -f
```

The first line is required and simply informs the operating system to use the system shell to run the next line (command). Line 2 does the rotation, using the program located at /usr/sbin/logrotate and the configuration file /etc/logrotate.conf . My logrotate.conf contains:

```
# system-specific logs may be configured here
############################################################
# #
# MONTHLY rotations #
# #
############################################################ # rotate
apache log files:
/home/httpd/logs/*_log {
ifempty
copytruncate
rotate 12
monthly
compress
}
```

Note the first part of this file (up to # system-specific logs may be configured here) are default parameters and are ignored here for clarity. Below this comment, parameters over-ride the defaults.

[One caveat of the default parameters is:

```
# send errors to root
errors your@emailaddress
```

This does not work as etc/crontab has: MAILTO=root which over-rides any set in logrotate.conf.]

The part that does the rotating/compressing (you can even e-mail the rotated file), follows the comment:

```
# system-specific logs may be configured here
/home/httpd/logs/*_log {
```

defines which files are to be rotated and must end in a sing closing brace '}'.

`ifempty`

defines that rotation will continue even if the file is empty.

`copytruncate`

defines a copy of the log file is created first and then its contents are removed (instead of simply creating a new one). This is required by apache as it can not be told to close a log file (release) without stopping the service. By this method the apache server does not have to be restarted.

`rotate 12`

Over-rides the default (4) by keeping 12 previous files.

`monthly`

Over-rides the default (weekly) by performing rotations monthly.

`compress`

Compress the log file usually by as much as 20:1. Auto-compression is probably logrotate's most powerful feature - something Window's struggles with! See below.

Read man logrotate for details concerning what other options may be useful to you.

[A caveat from the man logrotate page is that it appears to indicate the order of the commands is un-important. For instance, viewing the /var/log/news/* example nocompress appears after endscript. However changing this to compress will not work. It must come above postrotate. i.e. nocompress is actually ignored, but occurs as the default action.]

Windows Schematic Method

This is very similar to the Unix method above. Setup is much easier than for Unix (simply select a radio button in the IIS control panel). However there is little flexibility with this and compression is more complicated to achieve. The difference between Windows and Unix Schematics are:

- Windows log rotation is system wide - all virtual websites on the same server must therefore have the same log rotation settings (for Unix, this can be controlled on a per website basis).
- Windows rotation time periods are set to midnight only, on a daily, weekly or monthly basis (for Unix, any time period can be specified).
- Weekly rotation takes place on Sunday (not Monday), the first day of the week in the US.
- For compression, additional software is required e.g. WinZip, and a separate batch (script) file to run the compression with command line parameters.

Windows Method

The following discusses how you can add compression functionality and assumes you have already selected the required log file rotation frequency in the IIS control panel and have installed WinZip and the WinZip commandline add-on on your Windows machine in their respective default directories.

Create a file ziplogs.bat with your text editor containing the following:

```
ren "c:\Inetpub\weblog\w3svc1\monthly.zip" "monthly-old.zip"
"c:\program files\winzip\wzzip" -exomT
"c:\Inetpub\weblog\w3svc1\monthly.zip" "c:\weblog\w3svc1\*.log"
```

The first line renames the previous zip file. The second line calls the wzzip program (actually the WinZip commandline add-on) with options = exomT, and compresses all log files from the default IIS log file location into monthly.zip.

[Note, *.log is used here because Windows names its log files using a unique timestamp. However, following the initial run of the script, there should in fact only be one log file available - the last one just rotated.]

Options:

ex = Set the compression level to maximum
o = Change the zip file's date to the same as the newest file in the Zip file
m = Move files into the zip file
T = Include files older than the current date (if no date specified)

The most important option is 'T' (case sensitive). 'T' ensures only the last log file is included in the compression, not the newly created one. Use the Windows scheduler to run your ziplogs.bat batch file at 01:00 on the day in question i.e. just after the rotation.

As you will have noticed, the Windows method only gives you two backups of your log files (monthly.zip and monthly-old.zip). A more advance batch file is required if you wish to keep further (numbered) copies.