NAME

splitlog – split WWW server (httpd) access logfiles

SYNOPSIS

splitlog [**–f** *configfile*] [*options*...] [**--**] [*logfile* | **+** | **–**]...

DESCRIPTION

splitlog reads a sequence of **httpd** common logfile format (CLF) access_log files and/or the standard input and splits the logfile entries into separate files according to the entry's requested URL or virtual host prefix.

splitlog is intended to be run periodically by the webmaster as a means for providing individual logfiles for each of the customers of a server, since it is less efficient for the server itself to generate multiple logfiles. **splitlog** does not make any changes to the input file and can be configured to write the split files in any directory. By default, a cached DNS lookup is performed on any IP addresses which are unresolved in the input file. The log entries can also be anonymized if there are concerns about the requesting clients' privacy.

splitlog is a **perl** script, which means you need to have a **perl** interpreter to run the program. It has been tested with **perl** versions 4.036 and 5.002.

OPTIONS

Configuration Options

These options define how **splitlog** should establish defaults and interpret the command-line.

–f *filename*
: Get the configuration defaults from the given file. If used, this **must** be the first argument on the command-line, since it needs to be interpreted before the other command options. The file **splitlog.rc** is included with the distribution as an example of this file; it contains **perl** source code which directly sets the control and display options provided by **splitlog** and contains a function for altering the split logfile name-selection algorithm. If *filename* is not a pathname, the include path (see **FILES**) is searched for *filename*. An empty string as *filename* will disable this feature. [**–f** "splitlog.rc"]

--
: Last option (the remaining arguments are treated as input files).

Diagnostic Options

These options provide information about **splitlog** usage or about some unusual aspects of the logfile(s) being processed.

–h
: Help — display usage information to STDERR and then exit.

–e
: Display to STDERR all invalid log entries. Invalid log entries can occur if the server is miswriting or overwriting its own log, if the request is made by a broken client or proxy, or if a malicious attacker is trying to gain privileged access to your system.

Process Options

These options modify how and where logfile entries are written.

–x
: Discard any logfile entries without a filename key instead of placing them in a special OTHERS.log.

–v
: Use a prefix of the input file entries (ended by the first ":" or space) for selecting the output filename instead of, or in addition to, the URL path. The most likely use for such a prefix is for the requested virtual host.

–dir *directory*
: Place the output logfiles in the given directory instead of the current working directory.

–anon *imu*
: Anonymize the logfile entries before writing them to split logs. The value is some combination of the letters "i" (ident field is removed), "m" (machine name is replaced with ANON or 0), and "u" (authentication userid field is removed).

–dns

–nodns Do (**–dns**) or don’t (**–nodns**) use the system’s hostname lookup facilities to find the DNS hostname associated with any unresolved IP addresses. Looking up a DNS name may be **very** slow, particularly when the results are negative (no DNS name), which is why a caching capability is included as well. [**–dns**]

–cache *filename*

Use the given DBM database as the read/write persistent DNS cache (the .dir and .pag extensions are appended automatically). Cached entries (including negative results) are removed after the time configured for $DNSexpires [two months]. No caching is performed if *filename* is the empty string, which may be needed if your system does not support DBM or NDBM functionality. Running **–dns** without a persistent cache is not recommended. [**–cache** "dnscache"]

Search Options

These options are used to include or exclude logfile entries from being output according to whether or not they match a given pattern. The pattern is supplied in the form of a **perl regular expression**, except that the characters "+" and "." are escaped automatically unless the **–noescape** option is given. Enclose the pattern in single-quotes to prevent the command shell from interpreting some special characters. Multiple occurrences of the same option results in an OR-ing of the regular expressions.

–a *regexp*

–A *regexp*

Include (**–a**) or exclude (**–A**) all requests containing a hostname/IP address matching the given perl regular expression.

–c *regexp*

–C *regexp*

Include (**–c**) or exclude (**–C**) all requests resulting in an **HTTP** status code matching the given perl regular expression.

–d *regexp*

–D *regexp*

Include (**–d**) or exclude (**–D**) all requests occurring on a date (e.g., "Feb 02 1994") matching the given perl regular expression.

–t *regexp*

–T *regexp*

Include (**–t**) or exclude (**–T**) all requests occurring during the hour (e.g., "23" is 11pm – 12pm) matching the given perl regular expression.

–m *regexp*

–M *regexp*

Include (**–m**) or exclude (**–M**) all requests using an HTTP method (e.g., "HEAD") matching the given perl regular expression.

–n *regexp*

–N *regexp*

Include (**–n**) or exclude (**–N**) all requests on a URL (archive name) matching the given perl regular expression.

–noescape

Do not escape the special characters ("+" and ".") in the remaining search options.

INPUT

After parsing the options, the remaining arguments on the command-line are treated as input arguments and are read in the order given. If no input arguments are given, the configured default logfile is read [**+**].

– Read from standard input (STDIN).

+ Read the default logfile. [as configured]

logfile ...
Read the given logfile. If the *logfile*'s extension indicates that is is compressed (gz|z|Z), then pipe it through the configured decompression program [**gunzip –c**] first.

USAGE

In most cases, **splitlog** is run on a periodic basis by a wrapper program as a **crontab** entry shortly after midnight, typically in conjunction with rotating the current logfile. The **–D today** option can be used to split the main logfile on a daily basis without rotation.

All of the command-line options, and a few options that are not available from the command-line, can be changed within the user configuration file (see **splitlog.rc**). This file is actually a **perl** library module which is executed as part of the program's initialization. The example provided with the distribution includes complete documentation on what variables can be set and their range of values. If the default algorithm for selecting the split logfile name isn't desired, or if some set of names should be combined into a single file, then uncomment the user_path_map() function and define your own name-selection algorithm.

The **wwwstat** program can be used to analyze the resulting logfiles. See **wwwstat** for a description of the common logfile format.

Perl Regular Expressions

The Search Options and many of the configuration file settings allow for full use of perl regular expressions (with the exception that the –a, –A, –n and –N options treat '+' and '.' characters as normal alphabetic characters unless they are preceded by the **–noescape** option). Most people only need to know the following special characters:

ˆ at start of pattern, means "starts with pattern".
$ at end of pattern, means "ends with pattern".
(...) groups pattern elements as a single element.
? matches preceding element zero or one times.
***** matches preceding element zero or more times.
+ matches preceding element one or more times.
. matches any single character.
[...] denotes a class of characters to match. [ˆ...] negates the class. Inside a class, '-' indicates a range of characters.
(A|B|C) matches if A or B or C matches.

Depending on your command shell, some special characters may need to be escaped on the command line or enclosed in single-quotes to avoid shell interpretation.

ENVIRONMENT

HOME Location of user's home directory, placed on INC path.

LOGDIR Used instead of HOME if latter is undefined.

PERLLIB A colon-separated list of directories in which to look for the user configuration file.

FILES

Unless a pathname is supplied, the configuration file is obtained from the current directory, the user's home directory (**HOME** or **LOGDIR**), the standard library path (**PERLLIB**), and the directory indicated by the command pathname (in that order).

splitlog.rc User configuration file.

dnscache.dir
dnscache.pag DBM files for persistent DNS cache.

SEE ALSO

crontab(1), **httpd**(1m), **perl**(1), **wwwstat**(1)

More info and the latest version of splitlog can be obtained from

http://www.ics.uci.edu/pub/websoft/wwwstat/
ftp://www.ics.uci.edu/pub/websoft/wwwstat/

If you have any suggestions, bug reports, fixes, or enhancements, please join the <wwwstat-users@ics.uci.edu> mailing list by sending e-mail with "subscribe" in the subject of the message to the request address <wwwstat-users-request@ics.uci.edu>. The list is archived at the above address.

More About Perl

The Perl Language Home Page
http://www.perl.com/perl/index.html

Johan Vromans' Perl Reference Guide
http://www.xs4all.nl/~jvromans/perlref.html

AUTHOR

Roy Fielding (fielding@ics.uci.edu), University of California, Irvine. Please do not send questions or requests to the author, since the number of requests has long since overwhelmed his ability to reply, and all future support will be through the mailing list (see above).

This work has been sponsored in part by the Defense Advanced Research Projects Agency under Grant Numbers MDA972-91-J-1010 and F30602-94-C-0218. This software does not necessarily reflect the position or policy of the U.S. Government and no official endorsement should be inferred.