

Apache Module `mod_log_config`

Description:	Logging of the requests made to the server
Status:	Base
Module Identifier:	<code>log_config_module</code>
Source File:	<code>mod_log_config.c</code>

Summary

This module provides for flexible logging of client requests. Logs are written in a customizable format, and may be written directly to a file, or to an external program. Conditional logging is provided so that individual requests may be included or excluded from the logs based on characteristics of the request.

Three directives are provided by this module: `TransferLog` to create a log file, `LogFormat` to set a custom format, and `CustomLog` to define a log file and format in one step. The `TransferLog` and `CustomLog` directives can be used multiple times in each server to cause each request to be logged to multiple files.

Topics

Custom Log Formats	1
Security Considerations.....	3
URI References	5

Directives

<code>CookieLog</code>	3
<code>CustomLog</code>	3
<code>LogFormat</code>	4
<code>TransferLog</code>	5

See also

- [Apache Log Files](#)¹

Custom Log Formats

The format argument to the `LogFormat` and `CustomLog` directives is a string. This string is logged to the log file for each request. It can contain literal characters copied into the log files and the c-type control characters `"\n"` and `"\t"` to represent new-lines and tabs. Literal quotes and back-slashes should be escaped with back-slashes.

The characteristics of the request itself are logged by placing `"%"` directives in the format string, which are replaced in the log file by the values as follows:

Format String	Description
<code>%%</code>	The percent sign
<code>%. . . a</code>	Remote IP-address
<code>%. . . A</code>	Local IP-address
<code>%. . . B</code>	Bytes sent, excluding HTTP headers.
<code>%. . . b</code>	Bytes sent, excluding HTTP headers. In CLF format, <i>i.e.</i> a '-' rather than a 0 when no bytes are sent.
<code>%. . . {Foobar} C</code>	The contents of cookie <i>Foobar</i> in the request sent to the server.
<code>%. . . D</code>	The time taken to serve the request, in microseconds.

Apache Module mod_log_config

Format String	Description
%...{FOOBAR}e	The contents of the environment variable <i>FOOBAR</i>
%...f	Filename
%...h	Remote host
%...H	The request protocol
%...{Foobar}i	The contents of <i>Foobar</i> : header line(s) in the request sent to the server.
%...l	Remote logname (from identd, if supplied)
%...m	The request method
%...{Foobar}n	The contents of note <i>Foobar</i> from another module.
%...{Foobar}o	The contents of <i>Foobar</i> : header line(s) in the reply.
%...p	The canonical port of the server serving the request
%...P	The process ID of the child that serviced the request.
%...q	The query string (prepended with a ? if a query string exists, otherwise an empty string)
%...r	First line of request
%...s	Status. For requests that got internally redirected, this is the status of the *original* request --- %...>s for the last.
%...t	Time, in common log format time format (standard english format)
%...{format}t	The time, in the form given by format, which should be in <code>strftime(3)</code> format. (potentially localized)
%...T	The time taken to serve the request, in seconds.
%...u	Remote user (from auth; may be bogus if return status (%s) is 401)
%...U	The URL path requested, not including any query string.
%...v	The canonical <code>ServerName</code> of the server serving the request.
%...V	The server name according to the <code>UseCanonicalName</code> setting.
%...X	<p>Connection status when response is completed:</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>X = connection aborted before the response completed. + = connection may be kept alive after the response is sent. - = connection will be closed after the response is sent.</p> </div> <p>(This directive was %...c in late versions of Apache 1.3, but this conflicted with the historical <code>ssl %...{var}c</code> syntax.)</p>
%...I	Bytes received, including request and headers, cannot be zero. You need to enable <code>mod_logio</code> to use this.
%...O	Bytes sent, including headers, cannot be zero. You need to enable <code>mod_logio</code> to use this.

The "... " can be nothing at all (e.g., "%h %u %r %s %b"), or it can indicate conditions for inclusion of the item (which will cause it to be replaced with "-" if the condition is not met). The forms of condition are a list of HTTP status codes, which may or may not be preceded by "!". Thus, "%400,501{User-agent}i" logs `User-agent:` on 400 errors and 501 errors (Bad Request, Not Implemented) only; "%!200,304,302{Referer}i" logs `Referer:` on all requests which did *not* return some sort of normal status.

Note that there is no escaping performed on the strings from %...r, %...i and %...o. This is mainly to comply with the requirements of the Common Log Format. This implies that clients can insert

 Apache Module mod_log_config

control characters into the log, so care should be taken when dealing with raw log files.

Some commonly used log format strings are:

Common Log Format (CLF)

```
"%h %l %u %t \"%r\" %>s %b"
```

Common Log Format with Virtual Host

```
"%v %h %l %u %t \"%r\" %>s %b"
```

NCSA extended/combined log format

```
"%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\""
```

Referer log format

```
"%{Referer}i -> %U"
```

Agent (Browser) log format

```
"%{User-agent}i"
```

Note that the canonical `ServerName` and `Listen` of the server serving the request are used for `%v` and `%p` respectively. This happens regardless of the `UseCanonicalName` setting because otherwise log analysis programs would have to duplicate the entire vhost matching algorithm in order to decide what host really served the request.

Security Considerations

See the security tips² document for details on why your security could be compromised if the directory where logfiles are stored is writable by anyone other than the user that starts the server.

CookieLog Directive

Description:	Sets filename for the logging of cookies
Syntax:	<code>CookieLog filename</code>
Context:	server config, virtual host
Status:	Base
Module:	mod_log_config
Compatibility:	This directive is deprecated.

The `CookieLog` directive sets the filename for logging of cookies. The filename is relative to the `ServerRoot`. This directive is included only for compatibility with `mod_cookies`, and is deprecated.

CustomLog Directive

Description:	Sets filename and format of log file
Syntax:	<code>CustomLog file pipe format nickname</code> <code>[env=[!]environment-variable]</code>
Context:	server config, virtual host
Status:	Base
Module:	mod_log_config

The `CustomLog` directive is used to log requests to the server. A log format is specified, and the logging can optionally be made conditional on request characteristics using environment variables.

 Apache Module mod_log_config

The first argument, which specifies the location to which the logs will be written, can take on one of the following two types of values:

file

A filename, relative to the `ServerRoot`.

pipe

The pipe character "|", followed by the path to a program to receive the log information on its standard input.

Security:

If a program is used, then it will be run under the user who started httpd. This will be root if the server was started by root; be sure that the program is secure.

The second argument specifies what will be written to the log file. It can specify either a *nickname* defined by a previous `LogFormat` directive, or it can be an explicit *format* string as described in the log formats section.

For example, the following two sets of directives have exactly the same effect:

```
# CustomLog with format nickname
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log common

# CustomLog with explicit format string
CustomLog logs/access_log "%h %l %u %t \"%r\" %>s %b"
```

The third argument is optional and allows the decision on whether or not to log a particular request to be based on the presence or absence of a particular variable in the server environment. If the specified environment variable³ is set for the request (or is not set, in the case of a 'env=!name' clause), then the request will be logged.

Environment variables can be set on a per-request basis using the `mod_setenvif` and/or `mod_rewrite` modules. For example, if you want to record requests for all GIF images on your server in a separate logfile but not in your main log, you can use:

```
SetEnvIf Request_URI \.gif$ gif-image
CustomLog gif-requests.log common env=gif-image
CustomLog nongif-requests.log common env=!gif-image
```

LogFormat Directive

Description:	Describes a format for use in a log file
Syntax:	<code>LogFormat format nickname [nickname]</code>
Default:	<code>LogFormat "%h %l %u %t \"%r\" %>s %b"</code>
Context:	server config, virtual host
Status:	Base
Module:	mod_log_config

This directive specifies the format of the access log file.

The `LogFormat` directive can take one of two forms. In the first form, where only one argument is

Apache Module mod_log_config

specified, this directive sets the log format which will be used by logs specified in subsequent `TransferLog` directives. The single argument can specify an explicit *format* as discussed in custom log formats section above. Alternatively, it can use a *nickname* to refer to a log format defined in a previous `LogFormat` directive as described below.

The second form of the `LogFormat` directive associates an explicit *format* with a *nickname*. This *nickname* can then be used in subsequent `LogFormat` or `CustomLog` directives rather than repeating the entire format string. A `LogFormat` directive that defines a nickname **does nothing else** -- that is, it *only* defines the nickname, it doesn't actually apply the format and make it the default. Therefore, it will not affect subsequent `TransferLog` directives. In addition, `LogFormat` cannot use one nickname to define another nickname. Note, that the nickname should not contain percent signs (%).

Example

```
LogFormat "%v %h %l %u %t \"%r\" %>s %b" vhost_common
```

TransferLog Directive

Description:	Specify location of a log file
Syntax:	<code>TransferLog file pipe</code>
Context:	server config, virtual host
Status:	Base
Module:	mod_log_config

This directive has exactly the same arguments and effect as the `CustomLog` directive, with the exception that it does not allow the log format to be specified explicitly or for conditional logging of requests. Instead, the log format is determined by the most recently specified `LogFormat` directive (which does not define a nickname). Common Log Format is used if no other format has been specified.

Example

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\"  
\"%{User-agent}i\""  
TransferLog logs/access_log
```

URI References

- [1] <http://httpd.apache.org/docs-2.1/logs.html>
- [2] http://httpd.apache.org/docs-2.1/misc/security_tips.html#serverroot
- [3] <http://httpd.apache.org/docs-2.1/env.html>