NAME

wwwstat – summarize WWW server (httpd) access statistics

SYNOPSIS

wwwstat [**–F** *system_config*] [**–f** *user_config*] [*options*...] [**--**] [*summary* | *logfile* | **+** | **–**]...

DESCRIPTION

wwwstat reads a sequence of **httpd** common logfile format (CLF) access_log files and/or prior **wwwstat** output summary files and/or the standard input and outputs a summary of the access statistics in HTML.

Since **wwwstat** does not make any changes to the input files or write any files in the server directories, it can be run by any user with read access to the input logfile(s) and summary file(s). This allows people other than the webmaster to run specialized analyses of just the things they are interested in summarizing.

wwwstat provides World Wide Web (WWW) access statistics, which does not necessarily correspond to statistics on individual users. It counts the number of **HTTP** requests received by the server and the amount of bytes transmitted in response to those requests, according to what is in the logfile(s), and outputs those counts as tables broken down by category of request.

wwwstat output summaries can be read by **gwstat** to produce fancy graphs of the summarized statistics. The **splitlog** program can be used to split a large logfile into separate files by entry prefix or URL path.

wwwstat 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.

Output Sections

wwwstat's output consists of a set of cross-reference links, the sum totals and averages for the processed data, and a sequence of amount-by-category tables partitioned into sections. The section categories are based on the characteristics evident from the access request, as provided by the common logfile format (see **NOTES**). These include:

Request Date	e.g., "Feb 2 1996"
Request Hour	e.g., "00" through "23"
Client Domain	The Fully-Qualified Domain Name (FQDN) suffix that corresponds to an organization type or country name.
Reversed Subdomain	The FQDN, usually minus the first (machine name) component, and reversed so that it is easier to read when sorted.
URL/Archive	Grouping based on Request-URI or non-success status code.
Identity	The user identity based on IdentityCheck token or Authorization field.

Each section can be enabled/disabled using the configuration files or command-line options (see **Section Display Options**).

Output Table Format

Inside each section, the statistics are presented as a preformatted table.

```
%Reqs %Byte  Bytes Sent  Requests   category-type
----- ----- ------------ -------- |---------------
NN.NN NN.NN NNNNNNNNNNNN NNNNNNNN | category-value
100.0 100.0 NNNNNNNNNNNN NNNNNNNN | category-value
```

Requests	Requests received for this category-value.
Bytes Sent	Bytes transmitted for this category-value.
%Reqs	(<Requests>/<Total Requests>)∗100.
%Byte	(<Bytes Sent>/<Total Bytes>)∗100.

The table can be sorted by category-value (**–sort key**), number of requests received (**–sort req**), or number of bytes received (**–sort byte**). It can also be limited to the **–top** *N* entries.

OPTIONS

Configuration Options

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

–F *filename*
: Get system 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 **wwwstat.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 **wwwstat**. 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** "wwwstat.rc"]

–f *filename*
: Get user configuration defaults from the given file. If used, this **must** be the first argument on the command-line after **–F** (if any). The file is the same format as for the **–F** option (see **wwwstat.rc**). 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** ".wwwstatrc"]

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

Diagnostic Options

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

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

–v Verbose display to STDERR of each log entry processed.

–x Display to STDERR all requests resulting in HTTP error responses.

–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. For the latter reason, the webmaster should run **wwwstat** with this option on a regular basis.

Display Options

These options modify the output format.

–H *string*
: Use the given string as the HTML title and heading for output.

–X *string*
: Use the given string as the cross-reference URL to the last summary output. Any occurrence of the characters "%M" or "%Y" are replaced by the month and year, respectively, of the month prior to the first log entry date. The empty string will exclude any cross-reference.

–R Display the daily stats table sorted in reverse. This option is primarily for use with the **gwstat** program for producing graphs of the output.

–l
–L Do (**–l**) or don’t (**–L**) display the full DNS hostname of clients in your local domain (which is determined by the configured value of $AppendToLocalhost) in the section on subdomain statistics. The default [**–L**] is to strip the machine name from local addresses.

–o
–O Do (**–o**) or don’t (**–O**) display the full DNS hostname of clients outside your local domain in the section on subdomain statistics. The default [**–O**] is to strip the machine name from outside addresses.

–u
–U Do (**–u**) or don’t (**–U**) display the IP address of clients with unresolved domain names in the section on subdomain statistics. The **–dns** option can be used to resolve some names, but not all IP

hosts have a DNS name (SLIP/PPP connections) and sometimes a host's DNS service is inaccessible. The default [**–U**] is to group all such addresses under the category "Unresolved".

–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. [**–nodns**]

–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"]

–trunc *N*
Truncate the URLs listed in the archive section after the *Nth* hierarchy level. This option is commonly used to reduce the output size and memory requirements of **wwwstat** by grouping the requests by directory tree instead of listing every URL. The default [**–trunc 0**] is to display every requested URL.

–files
–nofiles Do (**–files**) or don't (**–nofiles**) include the last component of a URL (usually the filename) in the archive section. This option is commonly used to reduce the output size and memory requirements of **wwwstat** by grouping the requests by directory instead of listing every URL. The default [**–files**] is to display the entire requested URL.

–link
–nolink Do (**–link**) or don't (**–nolink**) add a hypertext link around each archive URL. This option is useful for local maintenance, but it is not recommended for publication of the HTML results (it often results in links to temporary or nonexistant resources, and leads people/robots to resources that might not be publically available). [**–nolink**]

–cgi
–nocgi Do (**–cgi**) or don't (**–nocgi**) prefix the summary output with CGI header fields appropriate for use with the HTTP common gateway interface. Using **wwwstat** as a CGI script is not recommended — it is usually better to simply run the wwwstat program periodically and serve the static output file. [**–nocgi**]

Section Display Options

These options change the display of entire sections (as opposed to the entries within those sections). They allow the user to enable or disable an entire section, set the sorting method for that section, and limit the number of displayed entries for that section. These options are context-sensitive and processed in the order given.

–all
–noall Include (**–all**) or exclude (**–noall**) all of the display sections. The **–noall** option is commonly used just prior to one or more of the other section options, such that only the listed sections are displayed.

–daily
–nodaily
Include (**–daily**) or exclude (**–nodaily**) the section of statistics by request date and set the scope for later **–sort** and **–top** options to this section.

–hourly
–nohourly
Include (**–hourly**) or exclude (**–nohourly**) the section of statistics by request hour and set the

scope for later **–sort** and **–top** options to this section.

–domain
–nodomain
Include (**–domain**) or exclude (**–nodomain**) the section of statistics by the client's Internet domain and set the scope for later **–sort** and **–top** options to this section.

–subdomain
–nosubdomain
Include (**–subdomain**) or exclude (**–nosubdomain**) the section of statistics by the client's Internet subdomain (reversed for display) and set the scope for later **–sort** and **–top** options to this section.

–archive
–noarchive
Include (**–archive**) or exclude (**–noarchive**) the section of statistics by requested URL/archive and set the scope for later **–sort** and **–top** options to this section.

–r
–ident
–noident
Include (**–r** or **–ident**) or exclude (**–noident**) the section of statistics by the identity of the user (if IdentityCheck is ON) or the authentication userid (if supplied) and set the scope for later **–sort** and **–top** options to this section. **DO NOT PUBLISH** this information, as that would reveal security-related identities and be a violation of privacy. This option is provided for administrative purposes only.

–sort (**key**|**byte**|**req**)
Sort this section by its primary key, the number of bytes transmitted, or the number of requests received. [**–sort key**]

–top *N* Display only the top N entries for this section. This option assumes that the **–sort** option has been set to either bytes or requests.

–both Display both the top N entries for this section [10, sorted by requests], and then the full section (all entries) sorted by key.

Search Options

These options are used to limit the analysis to requests matching a 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. Search options are only applied to logfile entries; any summary files input must have been created with the same search options.

–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 2 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]

filename...
Read the given file and determine from the first line whether it is a previous output summary or a CLF logfile. If the *filename*'s extension indicates that is is compressed (gz|z|Z), then pipe it through the configured decompression program [**gunzip –c**] first. Summary files must have been created with the same (or similar) configuration and command-line options as the currently running program; if not, weird things will happen.

USAGE

wwwstat is used for many purposes:

- as a diagnostic utility for measuring server activity, finding incorrect URL references, and detecting attempted misuse of the server;
- as a public relations tool for measuring technology or information transfer (i.e., Is the message getting out? To the right people?);
- as an archival tool for tracking web usage over time without storing the entire logfile; and,
- most often, as an easy mechanism for justifying all the hard work that went into creating the web content that people out there are requesting.

In most cases, **wwwstat** is run on a periodic basis (nightly, weekly, and/or monthly) by a wrapper program as a **crontab** entry shortly after midnight, typically in conjunction with rotating the current logfile. The output is usually directed to a temporary file which can later be moved to a published location. The temporary file is necessary to avoid erasing your published file during wwwstat's processing (which would look very odd if someone tried to GET it from your web).

wwwstat can be run as a CGI script (**–cgi**), but that is not recommended unless the input logfile is very small.

All of the command-line options, and a few options that are not available from the command-line, can be changed within the user and system configuration files (see **wwwstat.rc**). These files are actually **perl** library modules which are 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.

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.

EXAMPLES

Summarize requests from commercial domains.
wwwstat –a ’.com$’

Summarize requests from the host kiwi.ics.uci.edu
wwwstat –a ’ˆkiwi.ics.uci.edu$’

Summarize requests not from kiwi.ics.uci.edu
wwwstat –A ’ˆkiwi.ics.uci.edu$’

Summarize requests resulting in temporary redirects
wwwstat –c ’302’

Summarize requests resulting in server errors
wwwstat –c ’ˆ5’

Summarize unsuccessful requests
wwwstat –C ’ˆ2’ –C ’304’

Summarize requests in first week of the month
wwwstat –d ’ [1-7] ’

Summarize requests in second week of the month
wwwstat –d ’ ([89]|1[0-4]) ’

Summarize requests in third week of the month
wwwstat –d ’ (1[5-9]|2[01]) ’

Summarize requests in fourth week of the month
wwwstat –d ’ 2[2-8] ’

Summarize requests in leftover days of the month
wwwstat –d ’ (29|30|31) ’

Summarize requests in February
wwwstat –d ’Feb’

Summarize requests in year 1994
wwwstat –d ’1994’

Summarize requests not in April
wwwstat –D ’Apr’

Summarize requests between midnight and 1am

wwwstat –t ’00’

Summarize requests not received between noon and 1pm
wwwstat –T ’12’

Summarize requests with a gif extension
wwwstat –n ’.gif$’

Summarize requests under user’s URL
wwwstat –n ’ˆ/˜user/’

Summarize requests not under "hidden" paths
wwwstat –N ’/hidden/’

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 include and configuration files.

FILES

Unless a pathname is supplied, the configuration files are 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).

.wwwstatrc User configuration file.

wwwstat.rc System configuration file.

domains.pl Mapping of Internet domain to country or organization.

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

SEE ALSO

crontab(1), **gwstat**(1), **httpd**(1m), **perl**(1), **splitlog**(1)

More info and the latest version of wwwstat 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 HTTP

HTTP/1.1 Proposed Standard
R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk, and T. Berners-Lee. "Hypertext Transfer Protocol -- HTTP/1.1", U.C. Irvine, DEC, MIT/LCS, August 1996.
http://www.ics.uci.edu/pub/ietf/http/

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

DIAGNOSTICS

See also the **Diagnostic Options** above.

"[none] to [none]" dates
wwwstat did not find any matching data to summarize. If you get such an empty summary, it means that either: 1) there was no valid data (the input files are all invalid or empty), or 2) none of

the data matched the search options given. Try using the **–e** option to show invalid data.

100% unresolved

If the subdomain section indicates that all of the client requests come from unresolved hostnames (IP addresses), this probably means that your server is running without DNS resolution (common for very busy sites). You can use the **–dns** option to have **wwwstat** perform the hostname lookups. If 100% of the hosts are still unresolved with the **–dns** option in effect, then it may be that all of the clients accessing your server are doing so from temporary SLIP/PPP addresses without DNS names, or it may be a problem with wwwstat's DNS cache (delete the cache files), with your system's DNS software (contact your system administrator), or with your network connection.

NOTES

Hits vs Requests vs Visitors

wwwstat counts HTTP requests received by the server. When a request is successful, it is often referred to as a "hit". Retrieving a single image is one GET request. Retrieving an HTML page is also one GET request, but that does not include the separate requests made for in-line images or related objects. Checking to see if a cached image is still valid (a HEAD or conditional GET) is also one request.

In all sections except the archive section, **wwwstat** shows the statistics for all requests (successful or not). In the archive section, it normally shows all non-successful requests under a special category for the status code and only successful requests (hits) under the URL or archive tree associated with the request. However, this grouping of non-successful requests is disabled when **wwwstat** is used with the search options **–n**, **–c**, and **–C**, since those options are normally used for finding error conditions.

wwwstat does not count "visitors" -- individual people or programs making the requests. HTTP does not, by default, provide any information that can be accurately correlated to an individual person, though it is possible (in an unreliable manner) to use HTTP extensions and request profiles as a means of tracking individual client programs. Such tracking requires extensive resources (memory and diskspace) and is often considered a violation of privacy.

With the exception of the ident section, **wwwstat** does not reveal information about the individual people making requests. Unless the output is limited to a specific URL or a specific hostname, **wwwstat**'s output does not connect the requester to the URL being requested.

Common Logfile Format

The httpd common logfile format (CLF) was defined in early 1994 as the result of discussions among server and access_log analyzer developers (Roy Fielding, John Franks, Kevin Hughes, Ari Luotonen, Rob McCool, and Tony Sanders) on how to make it easier for analysis tools to be used across multiple servers. The format is:

remote_host ident authuser [date-time zone] "Request-Line" Status-Code bytes

where	means
remote_host	Client DNS hostname or IP address
ident	Identity check token or "–"
authuser	Authorization user-id or "–"
date-time	dd/Mmm/yyyy:hh:mm:ss
zone	+dddd or –dddd
Request-Line	The first line of the HTTP request, which normally includes the method, URL, and HTTP-version.
Status-Code	Response status from server or "–"
bytes	Size of Entity-Body transmitted or "–"

with each field separated by a single space (it turns out that problems occur if the ident token contains a space, which was not anticipated by the original designers).

LIMITATIONS

wwwstat cannot be more accurate than its input.

The common logfile format does not include the amount of bytes transferred in HTTP header fields and in error responses. **wwwstat** attempts to estimate those bytes based on the response code. Although the built-in estimates will suffice for most applications, your results will be more accurate if the estimates are customized for the particular server software that generated the logfile.

Modern httpd servers have extended the CLF to include additional fields (Referer and User-Agent) or to make the entire format configurable. Although **wwwstat** is able to read logfiles which append information to the CLF, it will not make use of that additional information. However, **wwwstat** is written in **perl**, so if you want to parse a different format all you have to do is change the parsing code.

wwwstat does not do anything with Referer [sic] or User-Agent information that may be present in extended logfiles. In order to do anything interesting with Referer, the program would have to build a Request-URI x Referer x Count table, which would require huge gobs of memory and is better done using a separate program with a persistent database. Naturally, this is easy to do once you learn **perl**.

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).

wwwstat was originally based on a multi-server statistics program called **fwgstat-0.035** by Jonathan Magid (jem@sunsite.unc.edu) which, in turn, was heavily based on **xferstats** (packaged with the version 17 of the Wuarchive FTP daemon) by Chris Myers (chris@wugate.wustl.edu).

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.