The Webalizer - A web server log file analysis tool

What is The Webalizer?

The Webalizer is a web server log file analysis program which produces usage statistics in HTML format for viewing with a browser. The results are presented in both columnar and graphical format, which facilitates interpretation. Yearly, monthly, daily and hourly usage statistics are presented, along with the ability to display usage by site, URL, referrer, user agent (browser), search string, entry/exit page, username and country (some information is only available if supported and present in the log files being processed). Processed data may also be exported into most database and spreadsheet programs that support tab delimited data formats.

The Webalizer supports CLF (common log format) log files, as well as Combined log formats as defined by NCSA and others, and variations of these which it attempts to handle intelligently. In addition, wu-ftpd xferlog formatted logs and squid proxy logs are supported.

Gzip compressed logs may now be used as input directly. Any log filename that ends with a '.gz' extension will be assumed to be in gzip format and uncompressed on the fly as it is being read. In addition, the Webalizer also supports DNS lookup capabilities if enabled at compile time. See the file DNS.README for additional information.

This documentation applies to The Webalizer Version 2.01

Running the Webalizer

The Webalizer was designed to be run from a Unix command line prompt or as a cron job. There are several command line options which will modify the results it produces, and configuration files can be used as well. The format of the command line is:

```
webalizer [options ...] [log-file]
```

Where 'options' can be one or more of the supported command line switches described below. 'log-file' is the name of the log file to process (see below for more detailed information). If a dash ("-") is specified for the log-file name, STDIN will be used.

Once executed, the general flow of the program follows:

- A default configuration file is scanned for. A file named 'webalizer.conf' is searched for in the current directory, and if found, it's configuration data is parsed. If the file is not present in the current directory, the file '/etc/webalizer.conf' is searched for and, if found, is used instead.

- Any command line arguments given to the program are parsed. This may include the specification of a configuration file, which is processed at the time it is encountered.

- If a log file was specified, it is opened and made ready for processing. If no log file was given, or the filename '-' is specified on the command line, STDIN is used for input.

- If an output directory was specified, the program does a 'chdir' to that directory in preparation for generating output. If no output directory was given, the current directory is used.

- If a non-zero number of DNS Children processes were specified, they will be started, and the specified log file will be processed, either creating or updateing the specified DNS cache file.

- If no hostname was given, the program attempts to get the hostname using a uname system call. If that fails, 'localhost' is used.

- A history file is searched for. This file keeps previous month totals used on the main index.html page. The default file is named 'webalizer.hist', kept in the specified output directory, however may be changed using the "HistoryName" configuration file keyword.

- If incremental processing was specified, a data file is searched for and loaded if found, containing the 'internal state' data of the program at the end of a previous run. The default file is named 'webalizer.current', kept in the specified output directory, however may be changed using the "IncrementalName" configuration file keyword.

- Main processing begins on the log file. If the log spans multiple months, a separate HTML document is created for each month.

- After main processing, the main 'index.html' page is created, which has totals by month and links to each months HTML document.

- A new history file is saved to disk, which includes totals generated by The Webalizer during the current run.

- If incremental processing was specified, a data file is written that contains the 'internal state' data at the end of this run.

Incremental Processing

Version 1.2x of The Webalizer adds incremental run capability. Simply put, this allows processing large log files by breaking them up into smaller pieces, and processing these pieces instead. What this means in real terms is that you can now rotate your log files as often as you want, and still be able to produce monthly usage statistics without the loss of any detail. This is accomplished by saving and restoring all relevant internal data to a disk file between runs. Doing so allows the program to 'start where it left off' so to speak, and allows the preservation of detail from one run to the next.

Some special precautions need to be taken when using the incremental run capability of The Webalizer. Configuration options should not be changed between runs, as that could cause corruption of the internal stored data. For example, changing the MangleAgents level will cause different representations of user agents to be stored, producing invalid results in the user agents section of the report. If you need to change configuration options, do it at the end of the month after normal processing of the previous month and before processing the current month. You may also want to delete the 'webalizer.current' file as well (or whatever name was specified using the "IncrementalName" configuration option).

The Webalizer also attempts to prevent data duplication by keeping track of the timestamp of the last record processed. This timestamp is then compared to current records being processed, and any records that were logged previous to that timestamp are ignored. This, in theory, should allow you to re-process logs that have already been processed, or process logs that contain a mix of processed/not yet processed records, and not produce duplication of statistics. The only time this may break is if you have duplicate timestamps in two separate log files... any records in the second log file that do have the same timestamp as the last record in the previous log file processed, will be discarded as if they had already been processed. There are lots of ways to prevent this however, for example, stopping the web server before rotating logs will prevent this situation. This setup also necessitates that you always process logs in chronological order, otherwise data loss will occur as a result of the timestamp compare.

Output Produced

The Webalizer produces several reports (html) and graphics for each month processed. In addition, a summary page is generated for the current and previous months (up to 12), a history file is created and if incremental mode is used, the current month's processed data. The exact location and names of these files can be changed using configuration files and command line options. The files produced, (default names) are:

```
index.html              - Main summary page (extension may be changed)
usage.png               - Yearly graph displayed on the main index page
usage_YYYYMM.html       - Monthly summary page (extension may be changed)
usage_YYYYMM.png        - Monthly usage graph for specified month/year
daily_usage_YYYYMM.png  - Daily usage graph for specified month/year
hourly_usage_YYYYMM.png - Hourly usage graph for specified month/year
site_YYYYMM.html        - All sites listing (if enabled)
url_YYYYMM.html         - All urls listing (if enabled)
ref_YYYYMM.html         - All referrers listing (if enabled)
agent_YYYYMM.html       - All user agents listing (if enabled)
search_YYYYMM.html      - All search strings listing (if enabled)
webalizer.hist          - Previous month history (may be changed)
webalizer.current       - Incremental Data (may be changed)
site_YYYYMM.tab         - tab delimited sites file
url_YYYYMM.tab          - tab delimited urls file
ref_YYYYMM.tab          - tab delimited referrers file
agent_YYYYMM.tab        - tab delimited user agents file
```

user_YYYYMM.tab   - tab delimited usernames file
search_YYYYMM.tab   - tab delimited search string file

The yearly (index) report shows statistics for a 12 month period, and links to each month. The monthly report has detailed statistics for that month with additional links to any URL's and referrers found. The various totals shown are explained below.

Hits

Any request made to the server which is logged, is considered a 'hit'. The requests can be for anything... html pages, graphic images, audio files, CGI scripts, etc... Each valid line in the server log is counted as a hit. This number represents the total number of requests that were made to the server during the specified report period.

Files

Some requests made to the server, require that the server then send something back to the requesting client, such as a html page or graphic image. When this happens, it is considered a 'file' and the files total is incremented. The relationship between 'hits' and 'files' can be thought of as 'incoming requests' and 'outgoing responses'.

Pages

Pages are, well, pages! Generally, any HTML document, or anything that generates an HTML document, would be considered a page. This does not include the other stuff that goes into a document, such as graphic images, audio clips, etc... This number represents the number of 'pages' requested only, and does not include the other 'stuff' that is in the page. What actually constitutes a 'page' can vary from server to server. The default action is to treat anything with the extension '.htm', '.html' or '.cgi' as a page. A lot of sites will probably define other extensions, such as '.phtml', '.php3' and '.pl' as pages as well. Some people consider this number as the number of 'pure' hits... I'm not sure if I totally agree with that viewpoint. Some other programs (and people :) refer to this as 'Pageviews'.

Sites

Each request made to the server comes from a unique 'site', which can be referenced by a name or ultimately, an IP address. The 'sites' number shows how many unique IP addresses made requests to the server during the reporting time period. This DOES NOT mean the number of unique individual users (real people) that visited, which is impossible to determine using just logs and the HTTP protocol (however, this number might be about as close as you will get).

Visits

Whenever a request is made to the server from a given IP address (site), the amount of time since a previous request by the address is calculated (if any). If the time difference is greater than a pre-configured 'visit timeout' value (or has never made a request before), it is considered a 'new visit', and this total is incremented (both for the site, and the IP address). The default timeout value is 30

minutes (can be changed), so if a user visits your site at 1:00 in the afternoon, and then returns at 3:00, two visits would be registered. Note: in the 'Top Sites' table, the visits total should be discounted on 'Grouped' records, and thought of as the "Minimum number of visits" that came from that grouping instead. Note: Visits only occur on PageType requests, that is, for any request whose URL is one of the 'page' types defined with the PageType option. Due to the limitation of the HTTP protocol, log rotations and other factors, this number should not be taken as absolutely accurate, rather, it should be considered a pretty close "guess".

KBytes

The KBytes (kilobytes) value shows the amount of data, in KB, that was sent out by the server during the specified reporting period. This value is generated directly from the log file, so it is up to the web server to produce accurate numbers in the logs (some web servers do stupid things when it comes to reporting the number of bytes). In general, this should be a fairly accurate representation of the amount of outgoing traffic the server had, regardless of the web servers reporting quirks.

Note: A kilobyte is 1024 bytes, not 1000 :)

Top Entry and Exit Pages

The Top Entry and Exit tables give a rough estimate of what URL's are used to enter your site, and what the last pages viewed are. Because of limitations in the HTTP protocol, log rotations, etc... this number should be considered a good "rough guess" of the actual numbers, however will give a good indication of the overall trend in where users come into, and exit, your site.

Command Line Options

The Webalizer supports many different configuration options that will alter the way the program behaves and generates output. Most of these can be specified on the command line, while some can only be specified in a configuration file. The command line options are listed below, with references to the corresponding configuration file keywords.

General Options

-h Display all available command line options and exit program.

-v Display program version and exit program.

-d Display additional 'debugging' information for errors and warnings produced during processing. This normally would not be used except to determine why you are getting all those errors and wanted to see the actual data. Normally The Webalizer will just tell you it found an error, not the

actual data. This option will display the data as well.
Config file keyword: Debug

-F Specify that the log being used is a ftp log. Normally, the Webalizer expects to find a valid CLF or Combined format we server log file. This option allows you to process wu-ftpd xferlogs as well.
Config file keyword: LogType

-f Fold out of sequence log records back into analysis, by treating them as if they were the same date/time as the last good record. Normally, out of sequence log records are ignored. If you run apache, don't worry about this.
Config file keyword: FoldSeqErr

-i Ignore history file. USE WITH CAUTION. This causes The Webalizer to ignore any existing history file produced from previous runs and generate it's output from scratch. The effect will be as if The Webalizer is being run for the first time and any previous statistics will be lost (although the HTML documents, if any, will not be deleted) on the main index.html (yearly) web page.
Config file keyword: IgnoreHist

-p Preserve state (incremental processing). This allows the processing of partial logs in increments. At the end of the program, all relevant internal data is saved, so that it may be restored the next time the program is run. This allows sites that must rotate their logs more than once a month to still be able to use The Webalizer, and not worry about having to gather and feed an entire months logs to the program at the end of the month. See the section on "Incremental Processing" below for additional information. The default is to not perform incremental processing. Use this command line option to enable the feature.
Config file keyword: Incremental

-q Quiet mode. Normally, The Webalizer will produce various messages while it runs letting you know what it's doing. This option will suppress those messages. It should be noted that this WILL NOT suppress errors and warnings, which are output to STDERR.
Config file keyword: Quiet

-Q ReallyQuiet mode. This allows suppression of _all_ messages generated by The Webalizer, including warnings and errors. Useful when The Webalizer is run as a cron job.
Config file keyword: ReallyQuiet

-T Display timing information. The Webalizer keeps track of the time it begins and ends processing, and normally displays the total processing time at the end of each run. If quiet mode (-q or 'Quiet yes' in configuration file) is specified, this information is not displayed. This option forces the display of timing totals if quiet mode has been specified, otherwise it is redundant and will have no effect.
Config file keyword: TimeMe

-c file This option specifies a configuration file to use. Configuration files allow greater control over how The Webalizer behaves, and there are several ways to use them. As of version 0.98, The Webalizer searches for a default configuration file in the current directory named "webalizer.conf", and if not found, will search in the /etc/ directory for a file of the same name. In addition, you may specify a configuration file to use with this command line option.

-n name This option specifies the hostname for the reports generated. The hostname is used in the title of all reports, and is also prepended to URL's in the reports. This allows The Webalizer to be run on log files for 'virtual' web servers or web servers that are different than the machine the reports are located on, and still allows clicking on the URL's to go to the proper location. If a hostname is not specified, either on the command line or in a configuration file, The Webalizer attempts to determine the hostname using a 'uname' system call. If this fails, "localhost" will be used as the hostname.
Config file keyword: HostName

-o dir This options specifies the output directory for the reports. If not specified here or in a configuration file, the current default directory will be used for output.
Config file keyword: OutputDir

-x name This option allows the generated pages to have an extension other than '.html', which is the default. Do not include the leading period ('.') when you specify the extension.
Config file keyword: HTMLExtension

-P name Specify the file extensions for 'pages'. Pages (sometimes called 'PageViews') are normally html documents and CGI scripts that display the whole page, not just parts of it. Some system will need to define a few more, such as 'phtml', 'php3' or 'pl' in order to have them counted as well. The default is 'htm*' and 'cgi' for web logs and 'txt' for ftp.
Config file keyword: PageType

-t name This option specifies the title string for all reports. This string is used, in conjunction with the hostname (if not blank) to produce the actual title. If not specified, the default of "Usage Statistics for" will be used.
Config file keyword: ReportTitle

-Y Supress Country graph. Normally, The Webalizer produces country statistics in both Graph and Columnar forms. This option will suppress the Country Graph from being generated.
Config file keyword: CountryGraph

-G Supress hourly graph. Normally, The Webalizer produces hourly statistics in both Graph and Columnar forms. This option will suppress the Hourly Graph only from being generated.
Config file keyword: HourlyGraph

-H Supress Hourly statistics. Normally, The Webalizer produces

hourly statistics in both Graph and Columnar forms. This option will suppress the Hourly Statistics table only from being generated.
Config file keyword: HourlyStats

-L Disable Graph Legends. The color coded legends displayed on the in-line graphs can be disabled with this option. The default is to display the legends.
Config file keyword: GraphLegend

-l num Graph Lines. Specify the number of background reference lines displayed on the in-line graphics produced. The default is 2 lines, however can range anywhere from zero ('0') for no lines, up to 20 lines (looks funny!).
Config file keyword: GraphLines

-P name Page type. This is the extension of files you consider to be pages for Pages calculations (sometimes called 'pageviews'). The default is 'htm*' and 'cgi' (plus whatever HTMLExtension you specified if it is different). Don't use a period!

-m num Specify a 'visit timeout'. Visits are calculated by looking at the time difference between the current and last request made by a specific host. If the difference is greater that the visit timeout value, the request is considered a new visit. This value must be formatted as HHMMSS, and you can suppress leading zeros. The default is 30 minutes (3000).
Config file keyword: VisitTimeout

-M num Mangle user agent names. Normally, The Webalizer will keep track of the user agent field verbatim. Unfortunately, there are a ton of different names that user agents go by, and the field also reports other items such as machine type and OS used. For Example, Netscape 4.03 running on Windows 95 will report a different string than Netscape 4.03 running on Windows NT, so even though they are the same browser type, they will be considered as two totally different browsers by The Webalizer. For that matter, Netscape 4.0 running on Windows NT will report different names if one is run on an Alpha and the other on an Intel processor! Internet Exploder is even worse, as it reports itself as if it were Netscape and you have to search the given string a little deeper to discover that it is really MSIE! In order to consolidate generic browser types, this option will cause The Webalizer to 'mangle' the user agent field, attempting to consolidate generic browser types. There are 6 levels that can be specified, each producing different levels of detail. Level 5 displays only the browser name (MSIE or Mozilla) and the major version number. Level 4 will also display the minor version number (single decimal place). Level 3 will display the minor version number to two decimal places. Level 2 will add any sub-level designation (such as Mozilla/3.01Gold or MSIE 3.0b). Level 1 will also attempt to add the system type. The default Level 0 will disable name mangling and leave the user agent field unmodified, producing the greatest amount of detail.
Configuration file keyword: MangleAgents

-g num This option allows you to specify the level of domains name

grouping to be performed. The numeric value represents the level of grouping, and can be thought of as the 'number of dots' to be displayed. The default value of 0 disables any domain name grouping.
Configuration file keyword: GroupDomains

-D name This allows the specification of a DNS Cache file name. This filename MUST be specified if you have dns lookups enabled (using the -N command line switch or DNSChildren configuration keyword). The filename is relative to the default output directory if an absolute path is not specified (ie: starts with a leading '/'). This option is only available if DNS support was enabled at compile time, otherwise an 'Invalid Keyword' error will be generated. See the DNS.README file for additional information regarding DNS lookups.

-N num Number of DNS child processes to use for reverse DNS lookups. If specified, a DNSCache name MUST be specified also. If you do not wish a DNS cache file to be generated, specify a value of zero ('0') to disable it. This does not prevent using an existing cache file, only the generation of one at run time. See the DNS.README file for additional information regarding DNS lookups.

Hide Options

The following options take a string argument to use as a comparison for matching. Except for the IndexAlias option, the string argument can be plain text, or plain text that either starts or ends with the wildcard character '*'.

For Example:

Given the string "yourmama/was/here", the arguments "was", "*here" and "your*" will all produce a match.

-a name This option allows hiding of user agents (browsers) from the "Top User Agents" table in the report. This option really isn't too useful as there are a zillion different names that current browsers go by, depending where they were obtained, however you might have some particular user agents that hit your site a lot that you would like to exclude from the list. You must have a web server that includes user agents in it's log files for this option to be of any use. In addition, it is also useless if you disable the user agent table in the report (see the -A command line option or "TopAgents" configuration file keyword). You can specify as many of these as you want on the command line. The wildcard character '*' can be used either in front of or at the end of the string. (ie: Mozilla/4.0* would match anything that starts with the string "Mozilla/4.0").
Config file keyword: HideAgent

-r name This option allows hiding of referrers from the "Top Referrer"

table in the report. Referrers are URL's, either on your own local site or a remote site, that referred the user to a URL on your web server. This option is normally used to hide your own server from the table, as your own pages are usually the top referrers to your own pages (well, you get the idea). You must have a web server that includes referrer information in the log files for this option to be of any use. In addition, it is also useless if you disable the referrers table in the report (see the -R command line option or "TopReferrers" configuration file keyword). You can specify as many of these as you like on the command line.
Config file keyword: HideReferrer

-s name This option allows hiding of sites from the "Top Sites" table in the report. Normally, you will only want to hide your own domain name from the report, as it usually is one of the top sites to visit your web server. This option is of no use if you disable the top sites table in the report (see the -S command line option or "TopSites" configuration file option).
Config file keyword: HideSite

-X This causes all individual sites to be hidden, which results in only grouped sites to be displayed on the report.
Config file keyword: HideAllSites

-u name This option allows hiding of URL's from the "Top URL's" table in the report. Normally, this option is used to hide images, audio files and other objects your web server dishes out that would otherwise clutter up the table. This option is of no use if you disable the top URL's table in the report (see the -U command line option or "TopURLs" configuration file keyword).
Config file keyword: HideURL

-I name This option allows you to specify additional index.html aliases. The Webalizer usually strips the string 'index.' from URL's before processing, which has the effect of turning a URL such as /somedir/index.html into just /somedir/ which is really the same URL and should be treated as such. This option allows you to specify _additional_ strings that are to be treated the same way. Use with care, improper use could cause unexpected results. For example, if you specify the alias string of 'home', a URL such as /somedir/homepages/brad/home.html would be converted into just /somedir/ which probably isn't what was intended. This option is useful if your web server uses a different default index page other than the standard 'index.html' or 'index.htm', such as 'home.html' or 'homepage.html'. The string specified is searched for _anywhere_ in the URL, so "home.htm" would turn both "/somedir/home.htm" and "/somedir/home.html" into just "/somedir/". Go easy on this one, each string specified will be scanned for in EVERY log record, so if you specify a bunch of these, you will notice degraded performance. Wildcards are not allowed on this one.
Config file keyword: IndexAlias

Table Size Options

-e num This option specifies the number of entries to display in the "Top Entry Pages" table. To disable the table, use a value of zero (0).
Config file keyword: TopEntry

-E num This option specifies the number of entries to display in the "Top Exit Pages" table. To disable the table, use a value of zero (0).
Config file keyword: TopExit

-A num This option specifies the number of entries to display in the "Top User Agents" table. To disable the table, use a value of zero (0).
Config file keyword: TopAgents

-C num This option specifies the number of entries to display in the "Top Countries" table. To disable the table, use a value of zero (0).
Config file keyword: TopCountries

-R num This option specifies the number of entries to display in the "Top Referrers" table. To disable the table, use a value of zero (0).
Config file keyword: TopReferrers

-S num This option specifies the number of entries to display in the "Top Sites" table. To disable the table, use a value of zero (0).
Config file keyword: TopSites

-U num This option specifies the number of entries to display in the "Top URL's" table. To disable the table, use a value of zero (0).
Config file keyword: TopURLs

CONFIGURATION FILES

The Webalizer allows configuration files to be used in order to simplify life for all. There are several ways that configuration files are accessed by the Webalizer. When The Webalizer first executes, it looks for a default configuration file named "webalizer.conf" in the current directory, and if not found there, will look for "/etc/webalizer.conf". In addition, configuration files may be specified on the command line with the '-c' option. There are lots of different ways you can combine the use of configuration files and command line options to produce various results. The Webalizer always looks for and reads configuration options from a default configuration file before doing anything else. Because of this, you can override options found in the default file by use of additional configuration files specified on the command line or command line options themselves. If you specify a configuration file on the command line, you can override options in it by additional command line options which follow. For example, most users will most likely want to create the default file /etc/webalizer.conf and place options in it to specify the hostname, log

file, table options, etc... At the end of the month when a different log file is to be used (the end of month log), you can run The Webalizer as usual, but put the different filename on the end of the command line, which will override the log file specified in the configuration file. It should be noted that you cannot override some configuration file options by the use of command line arguments. For example, if you specify "Quiet yes" in a configuration file, you cannot override this with a command line argument, as the command line option only _enables_ the feature (-q option).

The configuration files are standard ASCII text files that may be created or edited using any standard editor. Blank lines and lines that begin with a pound sign ('#') are ignored. Any other lines are considered to be configuration lines, and have the form "Keyword Value", where the 'Keyword' is one of the currently available configuration keywords defined below, and 'Value' is the value to assign to that particular option. Any text found after the keyword up to the end of the line is considered the keyword's value, so you should not include anything after the actual value on the line that is not actually part of the value being assigned. The file "sample.conf" provided with the distribution contains lots of useful documentation and examples as well. It should be noted that you do not have to use any configuration files at all, in which case, default values will be used (which should be sufficient for most sites).

General Configuration Keywords

LogFile This defines the log file to use. It should be a fully qualified
name (ie: contain the path), but relative names will work as well. If not specified, the logfile defaults to STDIN.

LogType This specified the log file type being used. Normally, The Webalizer processes web logs in either CLF or Combined format. You may also process wu-ftpd xferlog formatted logs, or squid proxy logs by setting the appropriate type using this keyword. Values may be either 'clf', 'ftp' or 'squid'. Ensure that you specify the proper file type, otherwise you will be presented with a long stream of 'invalid record' messages ;)
Command line argument: -F

OutputDir This defines the output directory to use for the reports. If it is not specified, the current directory is used.
Command line argument: -o

HistoryName Allows specification of a history path/filename if desired. The default is to use the file named 'webalizer.hist', kept in the normal output directory (OutputDir above). Any name specified is relative to the normal output directory unless an absolute path name is given (ie: starts with a '/').

ReportTitle This specifies the title to use for the generated reports. It is used in conjunction with the hostname (unless blank) to produce the final report titles. If not defined, the default of "Usage Statistics for" is used.
Command line argument: -t

HostName This defines the hostname. The hostname is used in the report title as well as being prepended to URL's in the "Top URL's" table. This allows The Webalizer to be run on "virtual" web servers, or servers that do not reside on the local machine, and allows clicking on the URL to go to the right place. If not specified, The Webalizer attempts to get the hostname via a 'uname' system call, and if that fails, will default to "localhost".
Command line argument: -n

UseHTTPS Causes the links in the 'Top URL's' table to use 'https://' instead of the default 'http://' prefix. Not much use if you run a mix of secure/insecure servers on your machine. Only useful if you run the analysis on a secure servers logs, and want the links in the table to work properly.

Quiet This allows you to enable or disable informational messages while it is running. The values for this keyword can be either 'yes' or 'no'. Using "Quiet yes" will suppress these messages, while "Quiet no" will enable them. The default is 'no' if not specified, which will allow The Webalizer to display informational messages. It should be noted that this option has no effect on Warning or Error messages that may be generated, as they go to STDERR.
Command line argument: -q

TimeMe This allows you to display timing information regardless of any "quiet mode" specified. Useful only if you did in fact tell the webalizer to be quiet either by using the -q command line option or the "Quiet" keyword, otherwise timing stats are normally displayed anyway. Values may be either 'yes' or 'no', with the default being 'no'.
Command line argument: -T

GMTTime This keyword allows timestamps to be displayed in GMT (UTC) time instead of local time. Normally The Webalizer will display timestamps in the time-zone of the local machine (ie: PST or EDT). This keyword allows you to specify the display of timestamps in GMT (UTC) time instead. Values may be either 'yes' or 'no'. Default is 'no'.

Debug This tells The Webalizer to display additional information when it encounters Warnings or Errors. Normally, The Webalizer will just tell you it found a bad record or field. This option will enable the display of the actual data that produced the Warning or Error as well. Useful only if you start getting lots of Warnings or Errors and want to determine the cause. Values may be either 'yes' or 'no', with the default being 'no'.
Command line argument: -d

IgnoreHist This suppresses the reading of a history file. USE WITH EXTREME CAUTION as the history file is how The Webalizer keeps track of previous months. The effect of this option is as if The Webalizer was being run for the very first time, and any previous data is discarded. Values may be

either 'yes' or 'no', with the default being 'no'.
Command line argument: -i

FoldSeqErr — Allows log records that are out of sequence to be folded back into the analysis, by treating them as if they had the same date/time as the last good record. Normally, out of sequence log records are simply ignored. If you run apache, don't worry about this.

VisitTimeout — Set the 'visit timeout' value. Visits are determined by looking at the time difference between the current and last request made by a specific site. If the difference in time is greater than the visit timeout value, the request is considered a new visit. The value must be in the form of HHMMSS, leading zeros suppressed. The default value of 30 minutes (3000) should be fine for most.
Command line argument: -m

PageType — Allows you to define the 'page' type extension. Normally, people consider HTML and CGI scripts as 'pages'. This option allows you to specify what extensions you consider a page. Default is 'htm*' and 'cgi' for web logs, and 'txt' for ftp logs.
Command line argument: -P

GraphLegend — Enable/disable the display of color coded legends on the produced graphs. Default is 'yes', to display them.
Command line argument: -L

GraphLines — Specify the number of background reference lines to display on produced graphs. The default is 2. To disable the use of background lines, use zero ('0').
Command line argument: -l

CountryGraph — This keyword is used to either enable or disable the creation and display of the Country Usage graph. Values may be either 'yes' or 'no', with the default being 'yes'.
Command line argument: -Y

DailyGraph — This keyword is used to either enable or disable the creation and display of the Daily Usage graph. Values may be either 'yes' or 'no', with the default being 'yes'.

DailyStats — This keyword is used to either enable or disable the creation and display of the Daily Usage statistics table. Values may be either 'yes' or 'no', with the default being 'yes'.

HourlyGraph — This keyword is used to either enable or disable the creation and display of the Hourly Usage graph. Values may be either 'yes' or 'no', with the default being 'yes'.
Command line argument: -G

HourlyStats — This keyword is used to either enable or disable the creation and display of the Hourly Usage statistics table. Values may be either 'yes' or 'no', with the default being 'yes'.
Command line argument: -H

IndexAlias This allows additional 'index.html' aliases to be defined. Normally, The Webalizer scans for and strips the string "index." from URL's before processing them. This turns a URL such as /somedir/index.html into just /somedir/ which is really the same URL. This keyword allows _additional_ names to be treated in the same fashion for sites that use different default names, such as "home.html". The string is scanned for anywhere in the URL, so care should be used if and when you define additional aliases. For example, if you were to use an alias such as 'home', the URL /somedir/homepages/brad/home.html would be turned into just /somedir/ which probably isn't the intended result. Instead, you should have specified 'home.htm' which would correctly turn the URL into /somedir/homepages/brad/ like intended. It should also be noted that specified aliases are scanned for in EVERY log record... A bunch of aliases will noticeably degrade performance as each record has to be scanned for every alias defined. You don't have to specify 'index.' as it is always the default.
Command line argument: -I

MangleAgents The MangleAgents keyword specifies the level of user agent name mangling, if any. There are 6 levels that may be specified,
each producing a different level of detail displayed. Level 5 displays only the browser name (MSIE or Mozilla) and the major version number. Level 4 adds the minor version (single decimal place). Level 3 adds the minor version to two decimal places. Level 2 will also add any sub-level designation (such as Mozilla/3.01Gold or MSIE 3.0b). Level 1 will also attempt to add the system type. The default level 0 will leave the user agent field unmodified and produces the greatest amount of detail.
Command line argument: -M

SearchEngine This keyword allows specification of search engines and their query strings. Search strings are obtained from the referrer field in the record, and in order to work properly, the Webalizer needs to know what query strings different search engines use. The SearchEngine allows you to specify the search engine and it's query string to parse the search string from. The line is formatted as: "SearchEngine engine-string query-string" where 'engine-string' is a substring for matching the search engine with, such as "yahoo.com" or "altavista". The 'query-string' is the unique query string that is added to the URL for the search engine, such as "search=" or "MT=" with the actual search strings appended to the end. There is no command line option for this keyword.

Incremental This allows incremental processing to be enabled or disabled. Incremental processing allows processing partial logs without the loss of detail data from previous runs in the same month. This feature saves the 'internal state' of the program so that it may be restored in following runs. See the section above titled "Incremental Processing" for additional information. The value may be 'yes' or 'no', with the default being 'no'.

Command line argument: -p

IncrementalName
Allows specification of the incremental data filename if desired. Normally, the file named "webalizer.current' is used, kept in the standard output directory. If specified, filenames are relative to the standard output directory, unless an absolute name is given (ie: starts with '/').

DNSCache Specifies the DNS cache filename. This name is relative to the default output directory unless an absolute name is given (ie: starts with '/'). See the DNS.README file for additional information.

DNSChildren The number of DNS children processes to run in order to create/update the DNS cache file. If specified, the DNS cache filename must also be specified (see above). Use a value of zero ('0') to disable. See the DNS.README file for additional information.

Top Table Keywords

TopAgents This allows you to specify how many "Top" user agents are displayed in the "Top User Agents" table. The default is 15. If you do not want to display user agent statistics, specify a value of zero (0). The display of user agents will only work if your web server includes this information in its log file (ie: a combined log format file).
Command line argument: -A

AllAgents Will cause a separate HTML page to be generated for all normally visable User Agents. A link will be added to the bottom of the "Top User Agents" table if enabled. Value can be either 'yes' or 'no', with 'no' being the default.

TopCountries This allows you to specify how many "Top" countries are displayed in the "Top Countries" table. The default is 30. If you want to disable the countries table, specify a value of zero (0).
Command line argument: -C

TopReferrers This allows you to specify how many "Top" referrers are displayed in the "Top Referrers" table. The default is 30. If you want to disable the referrers table, specify a value of zero (0). The display of referrer information will only work if your web server includes this information in its log file (ie: a combined log format file).
Command line argument: -R

AllReferrers Will cause a separate HTML page to be generated for all normally visable Referrers. A link will be added to the "Top Referrers" table if enabled. Value can be either 'yes' or 'no', with 'no' being the default.

TopSites This allows you to specify how many "Top" sites are displayed in the "Top Sites" table. The default is 30. If you want to disable the sites table, specify a value of zero (0).
Command line argument: -S

TopKSites Identical to TopSites, except for the 'by KByte' table. Default is 10. No command line switch for this one.

AllSites Will cause a separate HTML page to be generated for all normally visable Sites. A link will be added to the bottom of the "Top Sites" table if enabled. Value can be either 'yes' or 'no', with 'no' being the default.

TopURLs This allows you to specify how many "Top" URL's are displayed in the "Top URL's" table. The default is 30. If you want to disable the URL's table, specify a value of zero (0).
Command line argument: -U

TopKURLs Identical to TopURLs, except for the 'by KByte' table. Default is 10. No command line switch for this one.

AllURLs Will cause a separate HTML page to be generated for all normally visable URLs. A link will be added to the bottom of the "Top URLs" table if enabled. Value can be either 'yes' or 'no', with 'no' being the default.

TopEntry Allows you to specify how many "Top Entry Pages" are displayed in the table. The default is 10. If you want to disable the table, specify a value of zero (0).
Command line argument: -e

TopExit Allows you to specify how many "Top Exit Pages" are displayed in the table. The default is 10. If you want to disable the table, specify a value of zero (0).
Command line argument: -E

TopSearch Allows you to specify how many "Top Search Strings" are displayed in the table. The default is 20. If you want to disable the table, specify a value of zero (0). Only works if using combined log format (ie: contains referrer information).

TopUsers This allows you to specify how many "Top" usernames are displayed in the "Top Usernames" table. Usernames are only available if you use http authentication on your web server, or when processing wu-ftpd xferlogs. The default value is 20. If you want to disable the Username table, specify a value of zero (0).

AllUsers Will cause a separate HTML page to be generated for all normally visable usernames. A link will be added to the bottom of the "Top Usernames" table if enabled. Value can be either 'yes' or 'no', with 'no' being the default.

AllSearchStr Will create a separate HTML page to be generated for all

normally visable Search Strings. A link will be added to the bottom of the "Top Search Strings" table if enabled. Value can be either 'yes' or 'no', with 'no' being the default.

Hide Object Keywords

These keywords allow you to hide user agents, referrers, sites, URL's and usernames from the various "Top" tables. The value for these keywords are the same as those used in their command line counterparts. You can specify as many of these as you want without limit. Refer to the section above on "Command Line Options" for a description of the string formatting used as the value. Values cannot exceed 80 characters in length.

HideAgent — This allows specified user agents to be hidden from the "Top User Agents" table. Not very useful, since there a zillion different names by which browsers go by today, but could be useful if there is a particular user agent (ie: robots, spiders, real-audio, etc..) that hits your site frequently enough to make it into the top user agent listing. This keyword is useless if 1) your log file does not provide user agent information or 2) you disable the user agent table.
Command line argument: -a

HideReferrer — This allows you to hide specified referrers from the "Top Referrers" table. Normally, you would only specify your own web server to be hidden, as it is usually the top generator of references to your own pages. Of course, this keyword is useless if 1) your log file does not include referrer information or 2) you disable the top referrers table.
Command line argument: -r

HideSite — This allows you to hide specified sites from the "Top Sites" table. Normally, you would only specify your own web server or other local machines to be hidden, as they are usually the highest hitters of your web site, especially if you have their browsers home page pointing to it.
Command line argument: -s

HideAllSites — This allows hiding all indvidual sites from the display, which can be useful when a lot of groupings are being used (since grouped records cannot be hidden). It is particularly useful in conjunction with the GroupDomain feature, however can be useful in other situations as well. Value can be either 'yes' or 'no', with 'no' the default.
Command line argument: -X

HideURL — This allows you to hide URL's from the "Top URL's" table. Normally, this is used to hide items such as graphic files, audio files or other 'non-html' files that are transferred to the visiting user.
Command line argument: -u

HideUser This allows you to hide Usernames from the "Top Usernames" table. Usernames are only available if you use http based authentication on your web server.

Group Object Keywords

The Group* keywords allow object grouping based on Site, URL, Referrer, User Agent and Usernames. Combined with the Hide* keywords, you can customize exactly what will be displayed in the 'Top' tables. For example, to only display totals for a particular directory, use a GroupURL and HideURL with the same value (ie: '/help/*'). Group processing is only done after the individual record has been fully processed, so name mangling and site total updates have already been performed. Because of this, groups are not counted in the main site total (as that would cause duplication). Groups can be displayed in bold and shaded as well. Grouped records are not, by default, hidden from the report. This allows you to display a grouped total, while still being able to see the individual records, even if they are part of the group. If you want to hide the detail records, follow the Group* directive with a Hide* one using the same value. There are no command line switches for these keywords. The Group* keywords also accept an optional label to be displayed instead of the actual value used. This label should be separated from the value by at least one whitespace character, such as a space or tab character. See the sample.conf file for examples.

GroupReferrer Allows grouping Referrers. Can be handy for some of the major search engines that have multiple host names a referral could come from.

GroupURL This keyword allows grouping URL's. Useful for grouping complete directory trees.

GroupSite This keywords allows grouping Sites. Most used for grouping top level domains and unresolved IP address for local dial-ups, etc...

GroupAgent Groups User Agents. A handy example of how you could use this one is to use "Mozilla" and "MSIE" as the values for GroupAgent and HideAgent keywords. Make sure you put the "MSIE" one first.

GroupDomains Allows automatic grouping of domains. The numeric value represents the level of grouping, and can be thought of as 'the number of dots' to display. A 1 will display second level domains only (xxx.xxx), a 2 will display third level domains (xxx.xxx.xxx) etc... The default value of 0 disables any domain grouping.
Command line argument: -g

GroupUser Allows grouping of usernames. Combined with a group name, this can be handy for displaying statistics on a particular group of users without displaying their real usernames.

GroupShading Allows shading of table rows for groups. Value can be
              'yes' or 'no', with the default being 'yes'.

GroupHighlight Allows bolding of table rows for groups. Value can be
               'yes' or 'no', with the default being 'yes'.

Ignore/Include Object Keywords

These keywords allow you to completely ignore log records when generating statistics, or to force their inclusion regardless of ignore criteria. Records can be ignored or included based on site, URL, user agent, referrer and username. Be aware that by choosing to ignore records, the accuracy of the generated statistics become skewed, making it impossible to produce an accurate representation of load on the web server. These keywords behave identical to the Hide* keywords above, where the value can have a leading or trailing wildcard '*'. These keywords, like the Hide* ones, have an absolute limit of 80 characters for their values. These keywords do not have any command line switch counterparts, so they may only be specified in a configuration file. It should also be pointed out that using the Ignore/Include combination to selectively exclude an entire site while including a particular 'chunk' is _extremely_ inefficient, and should be avoided. Try grep'ing the records into a separate file and process it instead.

IgnoreSite   This allows specified sites to be completely ignored from
              the generated statistics.

IgnoreURL   This allows specified URL's to be completely ignored from
              the generated statistics. One use for this keyword would
              be to ignore all hits to a 'temporary' directory where
              development work is being done, but is not accessible to
              the outside world.

IgnoreReferrer This allows records to be ignored based on the referrer
               field.

IgnoreAgent   This allows specified User Agent records to be completely
              ignored from the statistics. Maybe useful if you really
              don't want to see all those hits from MSIE :)

IgnoreUser   This allows specified username records to be completely
              ignored from the statistics. Usernames can only be used
              if you use http authentication on your server.

IncludeSite   Force the record to be processed based on hostname. This
              takes precedence over the Ignore* keywords.

IncludeURL   Force the record to be processed based on URL. This takes
              precedence over the Ignore* keywords.

IncludeReferrer Force the record to be processed based on referrer.
                This takes precedence over the Ignore* keywords.

IncludeAgent Force the record to be processed based on user agent.
              This takes precedence over the Ignore* keywords.

IncludeUser Force the record to be processed based on username. Usernames are only available if you use http based authentication on your server. This takes precedence over the Ignore* keywords.

Dump Object Keywords

The Dump* Keywords allow text files to be generated that can then be used for import into most database, spreadsheet and other external programs. The file is a standard tab delimited text file, meaning that each column is separated by a tab (0x09) character. A header record may be included if required, using the 'DumpHeader' keyword. Since these files contain all records that have been processed, including normally hidden records, an alternate location for the files can be specified using the 'DumpPath' keyword, otherwise they will be located in the default output directory.

DumpPath Specifies an alternate location for the dump files. The default output location will be used otherwise. The value is the path portion to use, and normally should be an absolute path (ie: has a leading '/' character), however relative path names can be used as well, and will be relative to the output directory location.

DumpExtension Allows the dump filename extensions to be specified. The default extension is "tab", however may be changed with this option.

DumpHeader Allows a header record to be written as the first record of the file. Value can be either 'yes' or 'no', with the default being 'no'.

DumpSites Dump tab delimited sites file. Value can be either 'yes' or 'no', with the default being 'no'. The filename used is site_YYYYMM.tab (YYYY=year, MM=month).

DumpURLs Dump tab delimited url file. Value can be either 'yes' or 'no', with the default being 'no'. The filename used is url_YYYYMM.tab (YYYY=year, MM=month).

DumpReferrers Dump tab delimited referrer file. Value can be either 'yes' or 'no', with the default being 'no'. Filename used is ref_YYYYMM.tab (YYYY=year, MM=month). Referrer information is only available if present in the log file (ie: combined web server log).

DumpAgents Dump tab delmited user agent file. Value can be either 'yes' or 'no', with the default being 'no'. Filename used is agent_YYYYMM.tab (YYYY=year, MM=month). User agent information is only available if present in the log file (ie: combined web server log).

DumpUsers Dump tab delimited username file. Value can be either 'yes' or 'no', with the default being 'no'. FIlename used is user_YYYYMM.tab (YYYY=year, MM=month). The

username data is only avilable if processing a wu-ftpd xferlog or http authentication is used on the web server and that information is present in the log.

DumpSearchStr Dump tab delimited search string file. Value can be either 'yes' or 'no', with the default being 'no'. Filename used is search_YYYYMM.tab (YYYY=year, MM=month). the search string data is only available if referrer information is present in the log being processed and recognized search engines were found and processed.

HTML Generation Keywords

These keywords allow you to customize the HTML code that The Webalizer produces, such as adding a corporate logo or links to other web pages. You can specify as many of these keywords as you like, and they will be used in the order that they are found in the file. Values cannot exceed 80 characters in length, so you may have to break long lines up into two or more lines. There are no command line counterparts to these keywords.

HTMLExtension Allows generated pages to use something other than the default 'html' extension for the filenames. Do not include the leading period ('.') when you specify the extension.
Command line argument: -x

HTMLPre Allows code to be inserted at the very beginning of the HTML files. Defaults to the standard HTML 3.2 DOCTYPE record. Be careful not to include any HTML here, as it is inserted _before_ the <HTML> tag in the file. Use it for server-side scripting capabilities, such as php3, to insert scripting files and other directives.

HTMLHead Allows you to insert HTML code between the <HEAD></HEAD> block. There is no default. Useful for adding scripts to the HTML page, such as Javascript or php3, or even just for adding a few META tags to the document.

HTMLBody This keyword defines HTML code to be placed immediately after the <HEAD> section of the report, just before the title and "summary period/generated on" lines. If used, the first HTMLHead line MUST include a <BODY> tag. Put whatever else you want in subsequent lines, but keep in mind the placement of this code in relation to the title and other aspects of the web page. Some typical uses are to change the page colors and possibly add a corporate logo (graphic) in the top right. If not specified, a default <BODY> tag is used that defines page color, text color and link colors (see "sample.conf" file for example).

HTMLPost This keyword defines HTML code that is placed after the title and "summary period/generated on" lines, just before the initial horizontal rule <HR> tag. Normally this keyword isn't needed, but is provided in case you included a large

graphic or some other weird formatting tag in the HTMLHead section that needs to be cleaned up or terminated before the main report section.

HTMLTail This keyword defines HTML code that is placed at the bottom right side of the report. It is inserted in a <TABLE> section between table data <TD>..</TD> tags, and is top and right aligned within the table. Normally this keyword is used to provide a link back to your home page or insert a small graphic at the bottom right of the page.

HTMLEnd This allows insertion of closing code, at the very end of the page. The default is to put the closing </BODY> and </HTML> tags. If specified, you _must_ specify these tags yourself.

Notes on Web Log Files

The Webalizer supports CLF log formats, which should work for just about everyone. If you want User Agent or Referrer information, you need to make sure your web server supplies this information in it's log file, and in a format that the Webalizer can understand. While The Webalizer will try to handle many of the subtle variations in log formats, some will not work at all. Most web servers output CLF format logs by default. For Apache, in order to produce the proper log format, add the following to the httpd.conf file:

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

This instructs the Apache web server to produce a 'combined' log that includes the referrer and user agent information on the end of each record, enclosed in quotes (This is the standard recommended by both Apache and NCSA). Netscape and other web servers have similar capabilities to alter their log formats. (note: the above works for apache servers up to V1.2. V1.3 and higher now have additional ways to specify log formats... refer to included documentation).

Notes on FTP Log Files

The Webalizer now supports ftp logs produced by wu-ftpd and others, as a standard 'xferlog'. To process an ftp log, you must either use the -Ff command line option or have "LogType ftp" in your configuration file. Support for additional formats may be forthcoming, however a future version of the Webalizer is in the works that will allow user defined log formats, so this will become a non-issue. It is recommended that you create a separate configuration file for ftp analysis, since the values used for your web server will most likely not be suited for ftp log analysis (ie: page types, hostname, etc.. should be different).

Because of the difference in web and ftp logs, there are a few limitations:

- Because there is no concept of a 'response code' in ftp world, response codes are restricted to either 200 (OK) or 206 (Partial Content), based on the completion status found in xferlog (for wu-ftpd, 'i'=incomplete and will generate a 206, 'c'=complete and will generate a 200). If your ftp server doesn't supply the completion status, all requests will be assigned a response code of 200. This allows the usage graph to display all transfer requests (hits), and how many of those completed in success (files - ie: 200 response codes).

- Page totals won't accurately reflect reality, since there isn't really the concept of a 'page' in regards to ftp services. I have found that setting the PageType value to "README", "FIRST", etc... seems to work fairly well however, and will give a pretty good indication of how many 'non-binary' files were requested. Of course, the content of your ftp site will be different, so your results may vary.

- Visit totals also won't accurately reflect reality, since visits are triggered on PageType requests (see above). What you usually wind up with is visits=sites in most cases.

- Entry/Exit pages will not be calculated for ftp logs.

- For obvious reasons, referrers and user agents are not supported.

- You _cannot_ analyze both web and ftp logs at the same time.. they must be done separately in different runs.

Notes on Referrers

Referrers are weird critters... They take many shapes and forms, which makes it much harder to analyze than a typical URL, which at least has some standardization. What is contained in the referrer field of your log files varies depending on many factors, such as what site did the referral, what type of system it comes from and how the actual referral was generated. Why is this? Well, because a user can get to your site in many ways... They may have your site bookmarked in their browser, they may simply type your sites URL field in their browser, they could have clicked on a link on some remote web page or they may have found your site from one of the many search engines and site indexes found on the web. The Webalizer attempts to deal with all this variation in an intelligent way by doing certain things to the referrer string which makes it easier to analyze. Of course, if your web server doesn't provide referrer information, you probably don't really care and are asking yourself why you are reading this section...

Most referrer's will take the form of "http://somesite.com/somepage.html", which is what you will get if the user clicks on a link somewhere on the web in order to get to your site. Some will be a variation of this, and look something like "file:/some/such/sillyname", which is a reference from a HTML document on the users local machine. Several variations of this can be used, depending on what type of system the user has, if he/she is on a local network, the type of network, etc... To complicate things even more, dynamic HTML documents and HTML documents that are generated by CGI scripts or external programs produce lots of extra information which is tacked on to the end of the referrer string in an almost infinite number of ways. If the user just typed your URL into their browser or clicked on

a bookmark, there won't be any information in the referrer field and will take the form "-".

In order to handle all these variations, The Webalizer parses the referrer field in a certain way. First, if the referrer string begins with "http", it assumes it is a normal referral and converts the "http://" and following hostname to lowercase in order to simplify hiding if desired. For example, the referrer "HTTP://WWW.MyHost.Com/This/Is/A/HTML/Document.html" will become
"http://www.myhost.com/This/Is/A/HTML/Document.html". Notice that only the "http://" and hostname are converted to lower case... The rest of the referrer field is left alone. This follows standard convention, as the actual method (HTTP) and hostname are always case insensitive, while the document name portion is case sensitive.

Referrers that came from search engines, dynamic HTML documents, CGI scripts and other external programs usually tack on additional information that it used to create the page. A common example of this can be found in referrals that come from search engines and site indexes common on the web. Sometimes, these referrers URL's can be several hundred characters long and include all the information that the user typed in to search for your site. The Webalizer deals with this type of referrer by stripping off all the query information, which starts with a question mark '?'. The Referrer "http://search.yahoo.com/search?p=usa%26global%26link" will be converted to just "http://search.yahoo.com/search".

When a user comes to your site by using one of their bookmarks or by typing in your URL directly into their browser, the referrer field is blank, and looks like "-". Most sites will get more of these referrals than any other type. The Webalizer converts this type of referral into the string "- (Direct Request)". This is done in order to make it easier to hide via a command line option or configuration file option. This is because the character "-" is a valid character elsewhere in a referrer field, and if not turned into something unique, could not be hidden without possibly hiding other referrers that shouldn't be.

Notes on Character Escaping

The HTTP protocol defines certain ways that URL's can look and behave. To some extent, referrer fields follow most of the same conventions. Character escaping is a technique by which non-printable or other non-ASCII (and even some ASCII) characters can be used in a URL. This is done by placing the Hexadecimal value of the character in the URL, preceeded by a percent sign '%'.
Since Hex values are made up of ASCII characters, any character can be escaped to ensure only printable ASCII characters are present in the URL. Some systems take this concept to the extreme and escape all sorts of stuff, even characters that don't need to be escaped. To deal with this, The Webalizer will un-escape URL's and referrers before being processed. For Example, the URL "/www.mrunix.net/%7Ebrad/resume.html" is the same URL as "/www.mrunix.net/~brad/resume.html", a very common form of a URL to access users web pages. If the URL's were not un-escaped, they would be treated as two separate documents, even though they are really one and the same.

Search String Analysis

The Webalizer will do a minimal analysis on referrer strings that it finds, looking for well known search string patterns. Most of the major search engines are supported, such as Yahoo!, Altavista, Lycos, etc... Unfortunately, search engines are always changing their internal/CGI query formats, new search engines are coming on line every day, and the ability to detect _all_ search strings is nearly impossible. However, it should be accurate enough to give a good indication of what users were searching for when they stumbled across your site. Note: as of version 1.31, search engines can now be specified within a configuration file. See the sample.conf file for examples of how to specify additional search engines.

Notes on Visits/Entry/Exit Figures

The majority of data analyzed and reported on by The Webalizer is as accurate and correct as possible based on the input log file. However, due to the limitation of the HTTP protocol, the use of firewalls, proxy servers, multi-user systems, the rotation of your log files, and a myriad of other conditions, some of these numbers cannot, without absolute accuracy, be calculated. In particular, Visits, Entry Pages and Exit Pages are suspect to random errors due to the above and other conditions. The reason for this is twofold, 1) Log files are finite in size and time interval, and 2) There is no way to distinguish multiple individual users apart given only an IP address. Because log files are finite, they have a beginning and ending, which can be represented as a fixed time period. There is no way of knowing what happened previous to this time period, nor is it possible to predict future events based on it. Also, because it is impossible to distinguish individual users apart, multiple users that have the same IP address all appear to be a single user, and are treated as such. This is most common where corporate users sit behind a proxy/firewall to the outside world, and all requests appear to come from the same location (the address of the proxy/firewall itself). Dynamic IP assignment (used with dial-up internet accounts) also present a problem, since the same user will appear as to come from multiple places.

For example, suppose two users visit your server from XYZ company, which has their network connected to the Internet by a proxy server 'fw.xyz.com'. All requests from the network look as though they originated from 'fw.xyz.com', even though they were really initiated from two separate users on different PC's. The Webalizer would see these requests as from the same location, and would record only 1 visit, when in reality, there were two. Because entry and exit pages are calculated in conjunction with visits, this situation would also only record 1 entry and 1 exit page, when in reality, there should be 2.

As another example, say a single user at XYZ company is surfing around your website.. They arrive at 11:52pm the last day of the month, and continue surfing until 12:30am, which is now a

new day (in a new month). Since a common practice is to rotate (save then clear) the server logs at the end of the month, you now have the users visit logged in two different files (current and previous months). Because of this (and the fact that the Webalizer clears history between months), the first page the user requests after midnight will be counted as an entry page. This is unavoidable, since it is the first request seen by that particular IP address in the new month.

For the most part, the numbers shown for visits, entry and exit pages are pretty good 'guesses', even though they may not be 100% accurate. They do provide a good indication of overall trends, and shouldn't be that far off from the real numbers to count much. You should probably consider them as the 'minimum' amount possible, since the actual (real) values should always be equal or greater in all cases.

Exporting Webalizer Data

The Webalizer now has the ability to dump all object tables to tab delimited ascii text files, which can then be imported into most popular database and spreadsheet programs. The files are not normally produced, as on some sites they could become quite large, and are only enabled by the use of the Dump* configuration keywords. The filename extensions default to '.tab' however may be changed using the 'DumpExtension' keyword. Since this data contains all items, even those normally hidden, it may not be desirable to have them located in the output directory where they may be visable to normal web users.. For this reason, the 'DumpPath' configuration keyword is available, and allows the placement of these files somewhere outside the normal web server document tree. An optional 'header' record may be written to these files as well, and is useful when the data is to be imported into a spreadsheet.. databases will not normally need the header. If enabled, the header is simply the column names as the first record of the file, tab separated.

Log files and The Webalizer

Most sites will choose to have The Webalizer run from cron at specified intervals. Care should be taken to ensure that data is not lost as a result of log file rotations. A suggested practice is to rotate your web server logs at the end of each month as close to midnight as possible, then have The Webalizer process the 'end of month' log file before running statistics on the new, current log. On our systems, a shell script called 'rotate_logs' is run at midnight, the end of each month. This script file looks like:

```
------------------------- file: rotate_logs ------------------------------
#!/bin/sh

# halt the server
kill `cat /var/lib/httpd/logs/httpd.pid`
```

```
# define backup names
OLD_ACCESS_LOG=/var/lib/httpd/logs/old/access_log.`date +%y%m%d-%H%M%S`
OLD_ERROR_LOG=/var/lib/httpd/logs/old/error_log.`date +%y%m%d-%H%M%S`

# make end of month copy for analyzer
cp /var/lib/httpd/logs/access_log /var/lib/httpd/logs/access_log.backup

# move files to archive directory
mv /var/lib/httpd/logs/access_log `echo $OLD_ACCESS_LOG`
mv /var/lib/httpd/logs/error_log  `echo $OLD_ERROR_LOG`

# restart web server
/usr/sbin/httpd

# compress the archived files
/bin/gzip $OLD_ACCESS_LOG
/bin/gzip $OLD_ERROR_LOG
------------------------ end of file ----------------------------------
```

This script first stops the web server using a 'kill' command. Apache keeps the PID of the server in the file httpd.pid, so we use it as the argument for the kill. Next, it defines some names for the backup files, which are basically the name of the files with the date and time appended to the end of them. It then makes a copy of the log file, appended with '.backup' in the log directory, moves the current log files to an archive directory (/var/lib/httpd/logs/old) and restarts the server. This setup allows the web server to be down for the minimum amount of time needed, which is important for busy sites. If you don't want to stop the server, you can remove the initial 'kill' command, and replace the '/usr/sbin/httpd' line with "kill -1 `cat /var/lib/httpd/logs/httpd.pid`" command instead, On most web servers, this will cause a restart of the server and create the new log files in the process...

At this point, we have made copies of the previous months logs, the web server is going about it's business as usual, and we have all the time in the world to do any other additional processing we want. The last two lines of the script compress the archived logs using the GNU zip program (gzip). Remember, we still have a copy of the log which we can now run The Webalizer on without having to do any further processing.

Next, we define two crontab entries. The first runs the above 'rotate_logs' script at midnight at the end of the month. The second runs The Webalizer on the '.backup' log file created above at 5 minutes after midnight. This gives other end of month processing jobs a chance to run so we don't bog the system down too much. If you have lots of end of month stuff going on, you can change the timing to suit your needs. The crontab entries look something like:

```
------------------------ crontab entries --------------------------------
# Rotate web server logs and run monthly analysis
0 0 1 * *       /usr/local/adm/rotate_logs
5 0 1 * *       /usr/bin/webalizer -Q /var/lib/httpd/logs/access_log.backup
------------------------ end of crontab ---------------------------------
```

As you can see, the log rotations occur at midnight, and the analysis is done at 5 minutes after. Once you verify that The Webalizer ran successfully, the access_log.backup file can be deleted as it isn't

needed any more. If you need to re-run the analysis, you still have the compressed archive copy that the shell script created. In order for the above analysis to work properly, you should have already created an /etc/webalizer.conf configuration file suitable for your site, or otherwise specify configuration options or a configuration file on the crontab command line above.

If you want The Webalizer to be run more often than once a month, you can specify additional crontab entries to do this as well. Care should be taken however to ensure that The Webalizer is not running when the end of month processing above occurs, or unpredictable results may happen (such as an inability to rotate the logs due to a file lock). The easiest way is to run it on the half hour with a crontab entry like:

```
30 * * * *      /usr/bin/webalizer
```

Language Support

Version 1.0x of The Webalizer added language support. This support is only provided at compile time in the form of an include file containing all the strings used by The Webalizer. The source distribution contains all language files that were available at the time, with English being the default as that is the only human language I speak fluently, and me Espanol es muy malo. Several people have already indicated the desire to do translations into various languages, and as I receive the language files, will make them available via ftp at ftp://ftp.mrunix.net/pub/webalizer/lang. Unless there happens to be a binary distribution in the language you need, you will need to grab the source distribution and compile the program yourself. See the file INSTALL that comes in the source distribution for information on how to use a language other than English.

It should also be noted that the GD graphics library, used to produce the in-line graphics in the output HTML, doesn't support extended character sets, so if you are translating the language file, you will no doubt encounter this problem.

New: You can now specify the language to use when you are building program from source, using the configure script. Just add --with-language=language_name , where 'language_name' is the name of a valid language file in the /lang/ directory. For example, --with-language=french will build using French as the default language. You should consult the INSTALL file for additional information on building the program from source.

Known Issues

- Memory Usage. The Webalizer makes liberal use of memory for internal data structures during analysis. Lack of real physical memory will noticeably degrade performance by doing lots of swapping between memory and disk. One user who had a rather large log file noticed that The

Webalizer took over 7 hours to run with only 16 Meg of memory. Once memory was increased, the time was reduced to a few minutes.

- Performance. The Hide*, Group*, Ignore*, Include* and IndexAlias configuration options can cause a performance decrease if lots of them are used. The reason for this is that every log record must be scanned for each item in each list. For example, if you are Hiding 20 objects, Grouping 20 more, and Ignoring 5, each record is scanned, at most, 46 times (20+20+5 + an IndexAlias scan). On really large log files, this can have a profound impact. It is recommended that you use the least amount of these configuration options that you can, as it will greatly improve performance.

Final Notes

A lot of time and effort went into making The Webalizer, and to ensure that the results are as accurate as possible. If you find any abnormalities or inconsistent results, bugs, errors, ommisions or anything else that doesn't look right, please let me know so I can investigate the problem or correct the error. This goes for the minimal documentation as well. Suggestions for future versions are also welcome and appreciated.