Paper AD04

CREATING MEANINGFUL DATA FROM WEB LOGS USING BASE SAS®

Jenine Eason, Autotrader.com, Peachtree City, GA
Jerry Johannesen, Autotrader.com, Marietta, GA

ABSTRACT

Analyzing web logs can bring very valuable information to your company. Not only can data collected from web logs give insight to management about what is happening on a web site, it also provides exact data for trending that sales can use to increase profitability. Without using a pre-packaged tool, base SAS can be utilized to read and evaluate web logs. The greatest challenge is dissecting a web log and parsing its values. This paper will bring understanding to web logs by identifying its different pieces and how best to collect them. We'll explore some of the basic reporting metrics utilizing transformed web logs.

INTRODUCTION

This paper will impart many years of SAS experience gained in the processing of internet web log data and turning enormous volumes of data into information and extending it into corporate business intelligence. The internet has developed its own jargon of web site traffic metrics, which are generally agreed upon, and left the "how to" application up to the user community. We seek to overcome the lack of available information regarding the measurement of internet web site metrics, specify the key values, and elaborate on how to compute them using base SAS.

The basis for accurate web site accounting is the log file. It captures extremely fine details of every user interaction. Personal information is not collected or inferred in the analysis of web log files, as reflected in each web site's privacy statement. The need to improve the commercial web site, increase purchasing behavior, and to improve the usability for all consumer interactions has led Internet sites to report activity per visit and user based on the cookies metric. This is not as simple as it sounds. There is more than one way to get to individual user activity.

Simply counting rows of web logs does not produce accurate measurement as there are many vagaries to the counting process that need to be applied. These include checking status codes, removing internal company traffic, spiders, web robot activity, and non-pageview log rows. Determining what needs to be read, filtered out, and how to do these activities remains more of an art than a science. There is usually more then one right answer. Cookies also have their own set of special rules that apply to accurately measure visits and users.

WEB LOG FORMATS

In order to effectively manage and report on a web site, it is necessary to get feedback about activity on the servers. All web servers have the ability to log client interactions into one or more log files or to pipe the log information to other applications. The two most popular formats are the Common Log Format (CLG) or the advanced versions Extended Common Log Format (ECLF) and Internet Information Services (IIS), a component of Microsoft Windows. The Apache HTTP Server provides very comprehensive and flexible logging capabilities. This document describes how to configure its logging capabilities, and how to understand what the logs contain.

The log file entries produced in *IIS* will look something like this:

> 192.168.114.201, -, 03/20/01, 7:55:20, W3SVC2, SALES1, 172.21.13.45, 4502, 163, 3223, 200, 0, GET, /DeptLogo.gif, -, 172.16.255.255, anonymous, 03/20/01, 23:58:11, MSFTPSVC, SALES1, 172.16.255.255, 60, 275, 0, 0,

The log file entries produced in *CLF* will look something like this:

> 10.102.8.152 - - [05/Nov/2003:00:19:54 -0500] "GET /inventory/index.jsp HTTP/1.1" 200 4028 "http://www.mycompany.com/index.jsp" "Mozilla/4.08 [en] (Win98; I ;Nav)"

The various components of the CLF log entry are described as follows:

10.102.8.152

This is the IP address of the client (remote host) which made the request to the server. This identifier is frequently referred to as a Cookie.

[05/Nov/2003:00:19:54 -0500]
The date/time the server completed responding to the request. The format is:

```
[day/month/year:hour:minute:second zone]
day = 2*digit
month = 3*letter
year = 4*digit
hour = 2*digit
minute = 2*digit
second = 2*digit
zone = (`+' | `-') 4*digit
```

"GET /inventory/index.jsp HTTP/1.1"
The request line from the client is given in double quotes. The request line contains a great deal of useful information. First, the method used by the client is GET. Second, the client requested the URL /inventory/index.jsp, and third, the client used the protocol HTTP/1.1.

200
The status is a 3 digit code which the server returns to the browser such as 200(complete) or 404 (not found). See appendix A for a complete list.

4028
This entry indicates the size of the object returned to the client, not including the response headers. If no content was returned to the client, this value will be "-". This is also referred to as Bytes and refers to the size of the page content. The larger the content, the slower the URL will load.

"http://www.mycompany.com/index.jsp"
The "Referrer" HTTP request header or referring URL. In this case, the referring URL, mycompany.com/index.jsp, links to /inventory/index/jsp.

"Mozilla/4.08 [en] (Win98; I ;Nav)"
The User Agent (Browser Operating System) is the name and version of the client software making the request and the operating system under which the client is operating. The web server uses this information to determine the feature set supported by the client browser. This ensures the response contains only items that can properly be interpreted and displayed by the browser.

READING WEB LOGS WITH SAS

There is ample software and products that can tackle reading web logs. Not surprisingly, SAS is one of them. If you don't have one of the packages or have needs that require additional capabilities, you can get the job done using Base SAS. Once there is a basic understanding of what a web log is, what it contains, and which segments to extract, you can begin creating a read routine. Below are tips and tactics to create a meaningful SAS dataset from web logs that can be used for a multitude of purposes.

TRANSLATE URL ENCODING

URL encoding can wreak havoc when reading a web log. If they can't be avoided by having the web developers' work around using them, then they need to be dealt with while reading the logs. A URL encode comes across as something like %2F when it really should translate to /. This occurs when web developers write their code, and use special characters like =, &, #, etc, so the encoding is a work around. The following code will clean up most instances.

```
        /* Remove URL encoding */
refurl = tranwrd(refurl,'%2F','/');
refurl = tranwrd(refurl,'%26','&');
refurl = tranwrd(refurl,'%3D','=');
refurl = tranwrd(refurl,'%3F','?');
refurl = tranwrd(refurl,'%3F%26','?');
refurl = tranwrd(refurl,'?&','?');
```

PARSING A URL

The parsing of logs is probably the single biggest task to convert web logs into meaningful SAS datasets. This can be done by working past the previously identified "parts" of the log and getting into the meat of the URL. Much valuable information can be passed through a URL. The URL below has 7 values that can be obtained (make, address, search_type, ac_affilt, borschtid, x, y). Note that they are after the page name and are preceded with an ampersand.

http://www.mycompany.com/findasweater/findasweater2.jtmpl?make=wool&address=30269&search_type=blue&ac_affilt=aol&borschtid=13717085352744529900&x=38&y=13

Know what values need to be captured for tracking purposes. Keeping up with the list of values is an issue for maintaining your log reading routine. The output dataset below called "new variables" will help keep tabs on them. Use the following code to parse out these values. The first step is identifying how many values are in each string to be parsed. The next step assigns values. Note several of the 7 parsing values are not listed in the code below. That is because they are not values we want to capture.

```
        /* Parse URL values and assign values*/

            * count number of values in URL (number of =);

  equalsigns = length(compress(lowcase(URLString),'abcdefghijklmnopqrstuvwxyz0123456789@#$%^&*()_.+"-
                    \|/?><]{};:'))+1;
  if equalsigns = 0 then equalsigns = 1;

            * Parse URL values;

  do i = 2 to equalsigns;
     tempvalue = substr(scan(URLString,i,'='),index(scan(URLString,i,'='),'=')+1,index(scan(URLString,i,'='),'&')-
                        index(scan(URLString,i,'='),'=')-1);
     if index(tempvalue,'?') then tempvalue=scan(tempvalue,1,'?');
     else if index(tempvalue,'&')>1 then tempvalue=scan(tempvalue,1,'&');
     else if index(tempvalue,'&')=1 then tempvalue='';
     tempname=scan(scan(URLString,i-1,'='),2,'?&');
     If tempname='' then tempname=scan(scan(URLString,i-1,'='),1,'?&');
     select(tempname);
          when ('make')          make     = tempvalue ;
          when ('address')       address  = tempvalue ;
          when ('search_type')   search_type = tempvalue ;
          when ('ac_affilt')     ac_affilt = tempvalue ;
          otherwise              newvar = tempname;
   end;
   if newvar in ('borschtid' 'x' 'y'') then newvar = '';
   if newvar = tempvalue then newvar = '';
   if index(newvar,'no_cache') or index(newvar,'HTTP') then newvar = '';
   if newvar ne '' then output newvariables;
  end;
```

There are other values you will want to locate and capture. You will need to understand how your log is parsed and distinguish the patterns that identify specific items. Then you can use a combination of SAS functions to extract the data. Such items might include the referring URL, browser, status, datetime stamp, and don't forget the cookie.

PAGEVIEWS AND REDIRECTORS

Pageviews account for a small portion of rows in web logs for an internet site. Filters must be applied to remove requests or rows of images, redirectors, asis tags, and other anomalies. The redirector is a useful internet tracking method, especially for capturing links that leave your web site and refer to a partner or content provider. If the link, at its simplest, was incorporated in a web page to another site, you would never observe any activity on the link. The web log entry would appear on the target company's web server logs, not yours. In order to track this, or any other link, the redirect method skips through an additional, unnoticed, page on your site. It redirects the user to the target destination after making a log entry. This traffic can be identified in the web logs as status 302 rows (Appendix A).

OUTPUT DATASETS

The luxury of processing your own web logs is that you can create SAS datasets for very specific needs. Perhaps you want one dataset with all pages viewed on the web site. There is a multitude of trending, tracking, and pathing reporting that can be utilized with such a dataset. For an internet company that sells items in a "shopping cart", a Selection dataset would be very useful. Most internet sites allow for some sort of emailing. A dataset that constitutes the collection of all these emails could be yet another valuable SAS dataset. A Sessions dataset would collect all activity surrounding and individual visitor session and provide much trending opportunities. Outputting to as many unique scenarios as you need allows future use of this data in smaller, more efficient sizes. Reporting from the additional datasets allows for unlimited SAS reporting opportunities.

FILTERS – SPIDERS, SCRAPERS, AND WEB ROBOTS

Multiple filters need to be applied to processed web log datasets prior to counting pages, visits, and traffic patterns. These include: removing internal traffic, spiders, scrapers, and non-pageview rows. Otherwise you will be counting rows of log files, and not pages and visitors.

INTERNAL TRAFFIC

Internal traffic is generally removed from web log traffic measurements because you want to count your customer's traffic, not your own company's web site usage. If you are measuring an intranet site that is only used inside the company, then you will want to leave internal traffic and instead identify the server and operational groups that view the intranet for rollups. In both cases, the TCP/IP address is used.

The local network administrator should be able to provide the range of TCP/IP addresses that are used in your company. Most corporations go through a firewall to get to the Internet, which results in everyone in the office appearing to have the same IP address in the web logs. Ask your network administrator to keep you informed of any infrastructure changes they make, such as adding in new hardware, or switching internet carriers. Otherwise, your web log reading routine will get out of date as the infrastructure changes, and you won't remove internal traffic over time.

IP ADDRESSES

IP addresses are composed of four octets, in the form 1.2.3.4; the third value is generally useful for separating servers within a company for counting internal traffic. If your network administrator cannot identify which servers connect to which organizations, you can plug a laptop into the company network at various locations and discover this yourself. To find out the TCP/IP address assigned to your PC after logging onto the network, go to the Start/Run/Command prompt and enter "IPCONFIG". The system will respond with several values. The one named IP ADDRESS is the TCP/IP number. Discovering the organizational servers in place at your company will allow you to assign a Group Name or Floor Number to each range of IP addresses. That way, you can see which areas of the company are using which pages of the internet, and how many times.

If you network administrator is unable to tell you what your company's internet firewall IP address is, or if you want to verify it yourself and monitor it over time there are web sites available to do a trace route lookup back to you machines IP. One such web site is http://www.slac.stanford.edu/cgi-bin/nph-traceroute.pl . If there are dial-in facilities or VPN servers you should try each of these methods for connecting to your company's network, as they usually have different internet connections for security purposes. Note: your security department may visit you after running the trace route, as these can appear as hacker port scans to them.

Once the IP address ranges are identified, the SAS programming required to remove or name the Floors is simple. For example:

```
if substr(usrhost,1,6) eq '10.102' or
   substr(usrhost,1,11) eq '206.251.31.' or
   substr(usrhost,1,11) eq '208.166.96.' then delete;
```

This filters out traffic when Usrhost is the TCP/IP address read from the web logs. You usually need to examine three of the four octets of the IP address in order to capture the server activity.

```
if substr(usrhost,1,11) eq '206.251.31.' then floor='1st'; else
if substr(usrhost,1,11) eq '208.166.96.' then floor='2nd' ;
```

This assigns a value to Floor based on the server that the internal user is coming through to get to the intranet.

SPIDERS

Spiders, also referred to as robots, webots, travelers, and index agents are machine-generated search engines that catalog the internet. They provide a necessary service to Google and the other search engine sites by providing fast access to the internet resources by creating a world wide index of available information. It is important to remove or separate this web traffic from that typical user traffic visiting your site. It is a generally accepted practice to identify and remove spider traffic when reporting the web site metrics to outside interests, through a variety of methods including user agent string exclusion and IP address.

Spiders have a distinct signature in their behavior, as most try to find and execute each and every link on your website sequentially. This creates a single page view of every area of you site, which can result in hundreds or thousands of pages viewed in a single session by a single visitor. This is one of the reasons it is important to remove the spidering activity from reporting web metrics, as it will tend to skew and overstate traffic.

There is a "spidering standard", which most well written spiders follow. The standard includes the use of an exclusion list for pages your web site wishes to exclude from spider activity. This text file is named robots.txt and exists in the root directory of a web site, e.g. www.autotrader.com/robots.txt. It is used to prevent spiders from getting lost in dynamically created webpages, such as those created by Google. In a dynamic web page the content depends on the search parameters entered and exists for display to the single individual requesting it. The results are an infinite possibility of URL strings with different search parameters.

Spider exclusion methods include lists of IP addresses known to be "spidering" sites and user agent string signatures. Due to the constantly changing nature of internet servers and IP addresses, an IP exclusion list is insufficient to remove spiders, and must be done in conjunction with a user agent string exclusion lists. If your company uses an audit service, such as ABCi or I/Pro, their proprietary lists seem to be the best quality and kept up to date. Several free sources of these lists are shown in appendix B.

USING SAS TO EXCLUDE SPIDERS

The filtering of spiders, either by IP address or user agent string is a simple matter, once the source lists are identified. IP exclusions may be for a range of IP addresses, instead of a single IP address. SAS is very good at separating an IP address into its components, called octets, since the period is the delimiter between the octets. One example is:

```
one=length(scan(usrhost,1));
two=length(scan(usrhost,2));
tre=length(scan(usrhost,3));
four=length(scan(usrhost,4));
onetwo=substr(usrhost,1,(1+one+two));
three=substr(usrhost,1,(2+one+two+tre));
```

Once the IP address is separated into its components the filtering of ranges of IP addresses is simple. A class B filter is done against the first two octets, e.g. 168.126.xx.xx. This is variable Onetwo in the code example above. Class C ranges are more commonly used as they target entire servers and use three of the four octets, e.g. 168.126.56.xx. In the code sample above, this the field Three given that Usrhost is the web log's TCP/IP address value.

To compare a list of values against individual IP addresses, SAS offers a variety of alternatives, including match/merging, creating SAS FORMATS, lists, do over arrays, MACRO assignments, index pattern matches, and of course hard coded values. The right method for your application depends on the length of the exclusion list and the frequency of updates.

USER AGENT STRINGS

User agent strings have proven to be more effective at keeping up with the continuing change of internet web service providers (ISP), proxy servers, and search engines that pop up each day. The user agent string is a signature of the browser being used to contact the web site, for example:

```
"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
"Mozilla/4.0 (compatible; MSIE 5.17; Mac_PowerPC)"
"Mozilla/4.0 (compatible; MSIE 6.0;
MSN 2.5; AOL 9.0; Windows 98; AT&T WNS5.0)"
"Mozilla/4.0 (compatible; MSIE 6.0; AOL 9.0; Windows NT 5.1; .NET CLR 1.0.3705; .NET CLR 1.1.4322)"
"Mediapartners-Google/2.1 (+http://www.googlebot.com/bot.html)"
```

This information is captured in the web log files on each page request and provides an effective mechanism to remove spiders. You will need an up-to-date list of spider user agent strings. One such free list can be found on http://www.psychedelix.com/agents.html. SAS provides multiple methods for matching this list against web logs.

Our preferred method for user agent string identification utilizes the index pattern matching function. For example:

```
if index(lowcase(agentstr), 'keynote') or
index(lowcase(agentstr), 'sureseeker') or
index(lowcase(agentstr), 'wget') or
index(lowcase(agentstr), 'libwww') or
index(lowcase(agentstr), 'spider') or
index(lowcase(agentstr), 'slurp') or
index(lowcase(agentstr), 'msiecrawler') or
index(lowcase(agentstr), 'scooter') then delete;
```

Our implementation uses a hard coded list of user agent string spiders that have been modified to remove the version of the spider number from the exact string. This provides the operational system the ability to immediately catch new point releases of all existing spiders engines. There is a manual update component to this method to add in new spider engines on a routine basis, and is generally done on a monthly basis. Note: the actual production list is much longer then shown in this example.

Routine maintenance of the IP/user agent string exclusion lists needs to be performed in some manner each month or quarter in order to keep up with the constantly changing internet complexion. New spider engines can be identified by checking the new version of IP and user agent string exclusion lists to the old list on your system each month. Additionally if you capture activity on the /robots.txt page and compare that daily against your existing list of IP and user agent string exclusions new spiders can be identified. A third mechanism is to examine your single user/cookie web activity to identify IP/user agent strings that requested an inordinate number of pages. There is a point at which it is unlikely that human behavior can generate thousands of pages consecutively, and it has to be a machine based page request. That point depends on your web sites content and audience. Some proxy servers (like AOL) provide access for thousands of people to web sites through a single proxy server and cache the information. Consequently not all high volume outlier sessions are spiders.

SCRAPERS

Scrapers are machine based web activity and differ from spiders in their purpose and behavior. Spiders generally exist in order to index the internet and serve up the results into search engines. Scrapers usually exist to gather and store web content for redisplay on another web site, caching, and for financial gain.

Have you ever seen a text dialogue box, such as the one on ticketmaster.com? To complete an order, it asks you to enter the word shown above. This is a method widely used to defeat scrapers, as the word shown is modified to no longer be machine-readable. The ticketmaster.com situation is a perfect example why some people engineer scrapers, also called webots, or web robots. These applications exist to perform web request activity, hundreds and thousands of times when launched against a web site. They generally have to be individually tailored to the web site of interest in order to perform the desired behavior. In the case of ticketmaster.com, the desired result is the best available ticket. This automated behavior can overwhelm an internet site. This can result in “denial of service” or “website unavailable” messages. As a result of this undesirable behavior and to protect individuals from ticket scalpers, ticketmaster.com and others have successfully implemented the “enter the word above” solution.

Scraping is also an issue for web sites that have material that other sites wish to distribute for which they don’t want to pay. An example would be articles that a web site has to pay authors a fee to publish. Web log activity can be analyzed to identify what, when, how many times, and who is performing this content scraping. It generally occurs in large volumes from a single source. Piracy and security concerns also direct web site analysts to be concerned about web site scraping. An example would be repeated password or credit card entries that appear to be sequential and repeated. The IP address and user agent string are quite useful in such a situation to identify the source of traffic. Reverse IP address lookup sites are common on the internet, and can pinpoint the requestor.

DATA MANAGEMENT CONSIDERATIONS

Internet web traffic analysis tends to work on large volumes of data – on the scale of the telecommunications call logs. Multi-million row inputs and gigabytes of resulting SAS datasets are commonplace. Consequently it is important to employ every available method of efficient programming techniques and tricks to keep things as containable as possible. Following are several recommendations from our web log analysis experiences.

PICTURES

Stop logging JPEG, GIF and all other images. This is a configuration option on the web server settings that needs to be requested from your web hosting administrator. We have seen no benefit to having the images logged in the session. Such records can easily double the size of the web log files being generated by the internet sessions.

COMPRESSION

Use the SAS compression option religiously. This can be set in the SAS autoexec and provides for the system to automatically remove blanks and whitespace as datasets are written to disk. There is some CPU tradeoff, but the reduction in input/output data movement more than overcomes the overhead of the compression algorithms. File size reduction depends on your data and disk storage media. There can be a 40% savings in disk space routinely, sometimes as much as 70%, which tends to make runs 40% faster.

Stale, older data that does not need to be processed on a regular basis will benefit from a further operating system compression such as zip, pkzip, compress, gnu gzip commands. Packing SAS data in this manner achieves another 80% compression in most cases. The tradeoff here is that datasets must be uncompressed before SAS can use them. This is an attractive alternative to offline tape storage when dealing with large SAS files. Batch files, shell scripts, SAS macros, or wildcard command line execution of compression commands are helpful in managing packing and unpacking entire libraries, or folders. Further information on how to divide up the log files into multiple runs can be found in the SUGI 25 paper, "*AutoTrader.com: From Chevette to Corvette*".

TAKE ADVANTAGE OF MULTIPLE DIRECTORIES

Unix implementations of SAS offer several alternative data management opportunities. SAS libnames point to directories on the file system. As files grow, it is not uncommon to grow out of the disk space, and have alternative disk packs added over time. Unix provides the ability to create a virtual file system by linking one directory to another, using the "ln –s" command. No matter how large your disk arrays are, there is a physical limit to the amount of disks that can be presented as a single filesystem. Applying the virtual file system over the top of a physical provides the capability to seamlessly combine multiple file systems into one logical. The autoexec and SAS libname statements can use the virtual filesystem in order to avoid having to change libname definitions as the filesystem grows and data is moved across multiple physical file systems.

For example:

> if the /data/a/meta library needs to move from the /data/a file system to the /data/b filesystem, the meta folder can be physically moved. At this point move the link to it and replace the meta folder, that was originally on /data/a.

Another Unix option that can help better manage web log data is the pipe operator. The pipe takes the output of one Unix command as input to another command. It is an option on the SAS filename statement. This allows you to keep the web logs compressed (with the gnu zip function gzip) until run time and uncompress logs directly into the SAS system through the pipe. They never have to be unzipped on the file system in order for SAS to read them in. One example of the SAS filename statement that accomplishes this is:

```
Filename LOGS_IN pipe '/usr/local/bin/gunzip -c /mis/logs/servername/access/*040101*gz';
```

Keep the file system organized into multiple directories that make sense for your business. In a large web site the servers come and go randomly. It is best to avoid hard coding folder names or server names that will need to change over time. You can use the web site server name as the generated naming convention for folders. This provides the ability to create a data driven read routine that is customized for each run and add parallelism to reading web log files for maximum throughput. Examination of the file system for server names and hourly log files by having SAS run operating system DIR or LS commands works well. For example, here the lS command is issued to determine the servers with logs, and stored as a dataset:

```
data howmany;
   length machin $ 50 machine $24;
   filename fred pipe 'ls -trld /mis/logs/*/access';
   infile fred ;
   input @1 @"/mis/logs/" machin $ ;
   machine=scan(machin,1,'/');
   nummach+1;
   call symput('machines',nummach);
run;
```

INDUSTRY STANDARDS

There are basic metrics generally accepted as standard by the industry. These metrics are tracked by third-party auditors of Web site traffic using industry-compliant standards that offer unbiased validation of a publisher's Web site. These metrics are typically presented by 3rd party companies with reports such as those found at http://www.ipro.com/pdf/custom_itm_report_sample.pdf or http://abcas2.accessabc.com/ABCiWeb/900114_1003_IA.PDF

Why do Web sites have standards of measurements and enlist 3rd party auditors?

1. 3rd party validation of data to marketers who are making online media buyers' decisions.
2. Secure instant credibility with advertisers and their agencies.
3. Establish their property as a market Leader.
4. Promote industry-compliant methodologies and standards for web measurement.

STANDARD METRICS

Let's explore some of the standard industry metrics:

Visits - A series of requests by a visitor without 30 consecutive minutes of inactivity.

Visitors - An individual who interacts with a Web site using Unique IP Addresses from one of the approved heuristic methodologies.

Unique Visitor - The number of different individuals who visit a site within a specific time period. To identify unique visitors, Web sites need a unique identifier, which may be obtained through some form of user registration or identification system.

Page Views - The combination of one or more files presented to a viewer as a single document as a result of a single request received by the server.

Page Request - The act of a user directing the Web Browser to "get" a page from a site, and the transmittal of that page to the user.

Referring Source – The source from which point visitors enter the Web site.

Ad Serving - The successful display of an advertisement on a browser screen (exclusive of non-qualifying activity and internal users).

Ad Request - The initial request of an advertisement from the browser as measured by the server that "redirects" a browser to the specific location of the advertisement (exclusive of non-qualifying activity and internal users).

Click-through - The result of "clicking on" an advertisement that links to the advertiser's Web site or another page within the Web site (exclusive of non-qualifying activity and internal users).

Session – The entire web activity on a site by a single user (cookie) without a 30 minute interruption.

Visits, or sessions, are determined from the valid pageviews remaining after all the filters to remove internal and spider activity have been applied. The information is sorted by user, datetime stamp, and summarized by user with the requirement that pages are viewed by the user within 30 minutes of the last pageview. If the same user views more pages after the 30 minute time period, then another new session or visit begins. The order of pageviews within each visit is useful for determining the common paths through the web site, namely what sequence of pages are typically viewed in specific order.

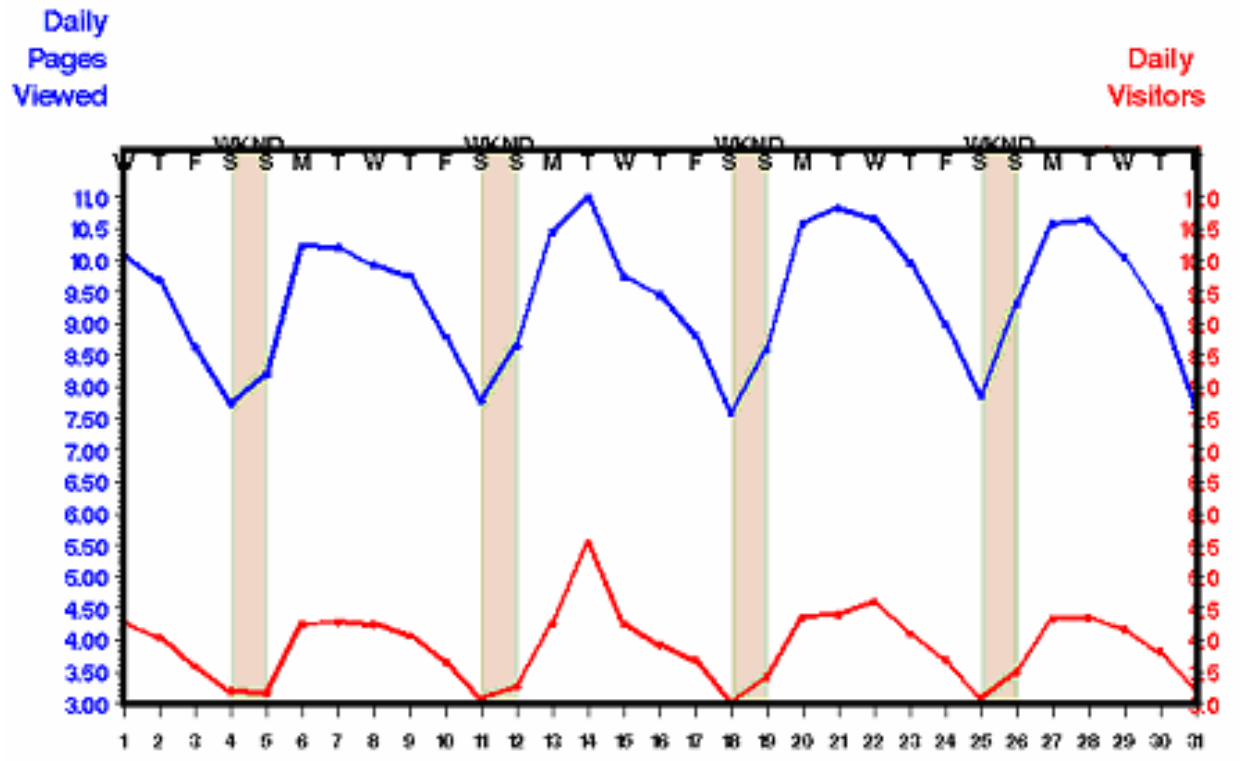
Visitors, or unique visitors, are user counts that are unique over time - typically measured by both day, and month. Since this is a unique measurement across a time period, you cannot sum the daily uniques to get monthly uniques. This is best done with a NODUPKEYS option on the sort procedure or DISTINCT on the SQL procedure. The user value being measured will depend on the website. Secure registered user logins are ideal for counting visitors, but most websites do not require users to login or register making this approach typically impossible. Cookies are used for counting users, and should be the persistent type that does not expire for many years, not the session type of cookie that is only assigned during the user visits. Persistent cookies remain on the user's computer. This means you should recognize that these visitor counts are really computers accessing the web site. One person may use more then one computer. The opposite is also true. For example: in a library setting where one computer may represent more then one user.

Counting visitors using persistent cookies also needs to take into account users that have their browsers set to disable cookies or other clean sweep software that actively remove cookies. The preferred method for identifying and correcting this type of user counting problem is to enforce the requirement that every cookie must have at least two log entries, where one is a valid page view. You should then apply an approved heuristic methodology, such as concatenating the IP address with the user agent string values into unique combinations. Simply counting unique cookies from the web log files and failing to account for the fact that some cookies are disabled by the browser (or ISP Proxy servers) can overstate the unique visitor count metric by twenty five percent or more depending on your web site audience.

INTERNAL REPORTS USING SAS BASED ON INDUSTRY STANDARDS

In addition to reporting done by 3rd party vendors, internal reporting is also necessary. SAS is an excellent tool to provide endless amounts of valuable information about Web site activity.

This graph created with PROC GCHART tracks daily page views and visitors. Such information is often the first thing viewed by all levels of a company's employees to track the pulse of the Web site. Activity trends, media marketing campaign impact, and other such information can be quickly apparent from such a graph.

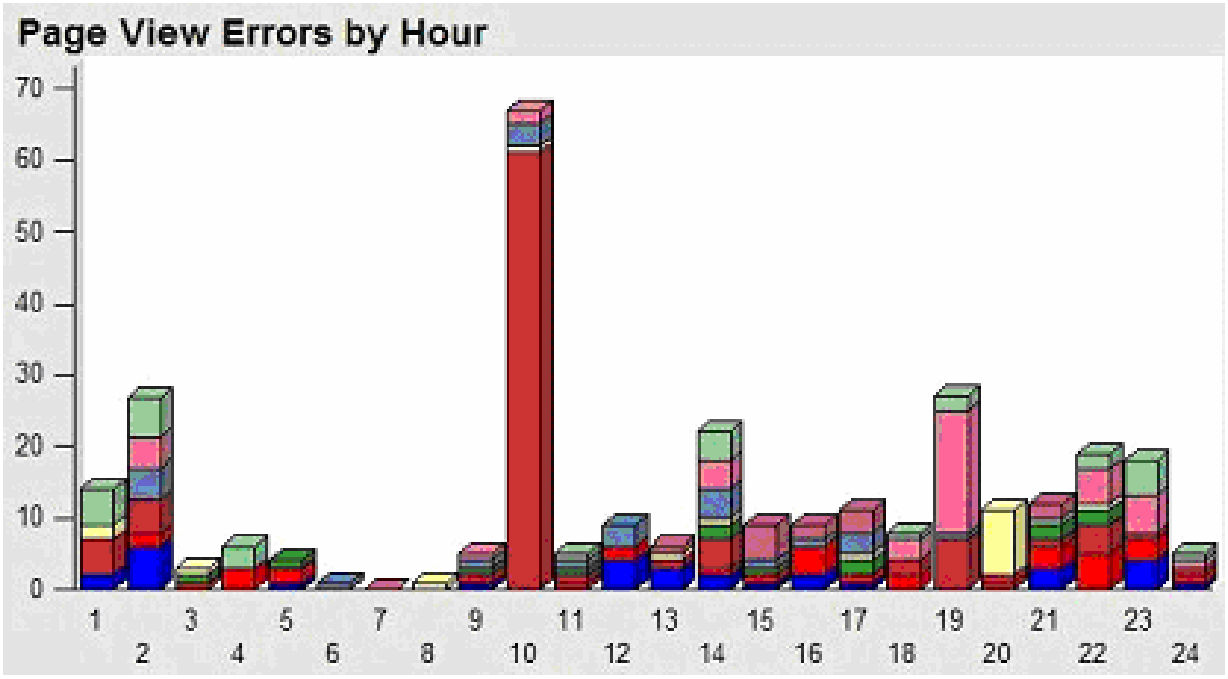


This table using PROC REPORT provides a list of browsers used to access the Web site. Such information is valuable to site developers to know which browsers their site needs to support.

VISITORS PER BROWSER

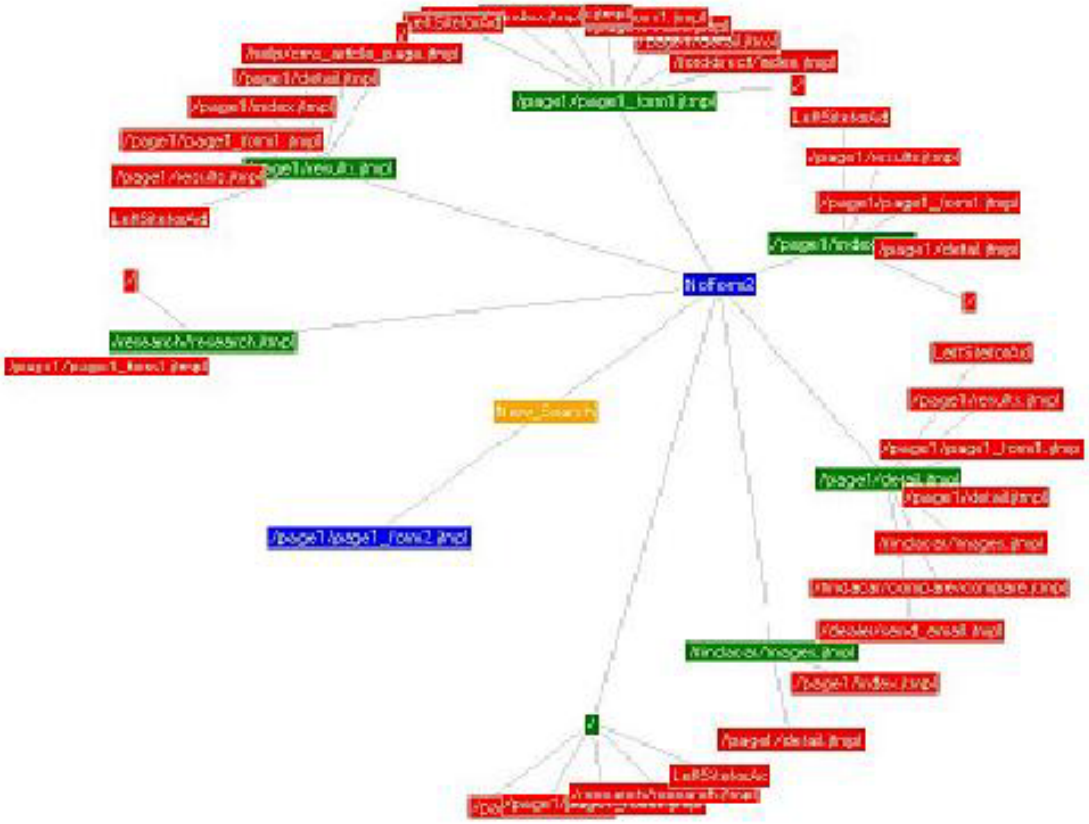
Broswer	Yymmdd	Yymmdd	Yymmdd
MSIE2.0	0	1	0
MSIE3.0	123	123	456
MSIE4.0	123	0	1
MSIE4.5	123	123	456
MSIE5.0	123	123	456
MSIE5.1	123	123	456
MSIE5.2	123	123	456
MSIE5.5	123	123	456
MSIE6.0	123	123	456
Netscape/7.0	123	123	266
Netscape/7.01	123	123	426
Netscape/7.02	1	0	0
Netscape/7.0	0	0	0
Netscape/7.1	3	9	5
Netscape6/6.0	0	1	2
Netscape6/6.1	123	187	195
Netscape6/6.2	717	9,227	15,649

This graph identifies server error activity. It was created using PROC GCHART with ODS and java drill-down capabilities. Such information is vital to web site maintenance groups. They can track negative activity, isolate its causes and make corrections quickly with such valuable information.



Tracking site navigation using SAS TREEVIEW, as seen below, allows a company to identify site activity patterns. Understanding how visitors navigate around a site can reveal strengths and weaknesses otherwise undetectable.

PAGEVIEW NAVIGATION



SAS MDDB provides an interactive tool used to identify advertising performance. This tool can report nearly any related metric in regards (in this case, for Ad Serving).

Marketing MDDB

Down

Ad Type, Child (hier)
Child, Flight (hier)
Child, Order (hier)
Order, Flight (hier)
Parent, Child, Order, Flight (hier)

View Report

Month		JAN04	
		Page Views	Percent
Ad Type		Sum	Average
Brand		100	0.19%
CPO		200	0.27%
Finance		501	0.05%
Insurance		267	0.04%
Marketing		259	3.14%
TOTAL		xxx,xxx	0.15%

WEB REPORTING

At SUGI 25 in 1999, AutoTrader.com received the SAS Institute Enterprise Computing award for their success in managing and reporting voluminous web log data. Based on this award, the authors of this paper previously presented at SUGI 25 the lessons learned from that labor: "*AutoTrader.com: From Chevette to Corvette*". We shared what we had learned about the dot.com world. We delved into the pitfalls the fast growth of the company put before us. Not only how that growth presented us with challenges in managing the growing amounts of data, but the growing request for meaningful data. As the company grew, so did the need for industry standards and accurate metric reporting. More eyes than ever were on the results.

CONCLUSION

Web traffic measurement standards continue to evolve over time, but the basics remain the same. We hope that this paper proves to be useful in defining what to capture and how to measure internet traffic using SAS. The surprising lack of documentation and standardization surrounding internet measurement is an indication of how misunderstood the finer points of accurate metrics really are to this day. Standard log file reporting software fail to report the same counts as the audit bureaus I/Pro and ABCi Interactive. Clearly understanding your web logs and their content will allow reporting with confidence and validity.

REFERENCES

http://www.abcinteractiveaudits.com
http://www.ipro.com
http://www.slac.stanford.edu/cgi-bin/nph-traceroute.pl
http://www.psychedelix.com/agents.html
Ralph Kimball and Richard Merz,The Data Webhouse Toolkit, (Wiley 2000)
Jenine Eason and Jerry Johannesen, SUGI 25 "AutoTrader.com: From Chevette to Corvette".

ACKNOWLEDGMENTS

Thank you, Carla Mast, for your considerable editing skills and valuable input.

APPENDIX A

HTTP RETURN STATUS CODES

100	Continue
101	Switching Protocols
200	OK
201	Created
202	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content
300	Multiple Choices
301	Moved Permanently
302	Moved Temporarily
303	See Other
304	Not Modified
305	Use Proxy
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Conflict
409	Request Time-Out
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request-URI Too Large

425	Unsupported Media Type
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Time-Out
505	HTTP Version not Supported

APPENDIX B

SPIDER RESOURCES ON THE WWW

http://www.psychedelix.com/agents.html
http://www.robotstxt.org/wc/active/html/index.html
http://www.netsys.com/cgi-bin/display_article.cgi?1193
http://www.usetracker.com/faq/blocked_IP_addresses.asp
http://www.iplists.com/
http://www.searchenginedictionary.com/spider-names.shtml
http://www.icehousedesigns.com/engines/spiderlist.php3
http://www.jafsoft.com/searchengines/webbots.html#search_engine_robots_and_others
http://www.siteware.ch/webresources/useragents/spiders/
http://www.simplythebest.net/info/useragents/spiders.html

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Jenine Eason
AutoTrader.com
5775 Peachtree Dunwoody Road
Atlanta, GA 30354
Work Phone: (404) 843-7199
Email: Jenine.Eason@AutoTrader.com

Jerry Johannesen
AutoTrader.com
5775 Peachtree Dunwoody Road
Atlanta, GA 30354
Work Phone: (404) 269-6986
Email: Jerry.Johannesen@AutoTrader.com