Contents

WEBTRENDS

WebTrends SmartSource Data Management: Premier Client-Side Data Collection Technology

White Paper

January 24, 2003

Web analytics is a well-established practice used to understand how visitors interact with a web site. Traditionally this is accomplished by using a web analytics software tool, such as WebTrends Log Analyzer or WebTrends Reporting Center, with the log files produced by web servers. But web server log file analysis has some challenges, including data accuracy and administrative problems.

As a result, more and more organizations are implementing an alternative technique for capturing site traffic information called *Client Side Data Collection*[1], or Data Tagging for short. Popularized by analytics vendors that provide hosted service solutions[2], data tagging solves many problems intrinsic to web server log file analysis while introducing a few of its own. If you retain one thing from reading this document, it should be that neither approach is clearly superior to the other, despite what vendors that only offer a single approach will tell you. Each approach has distinct advantages and drawbacks.

This paper will explain the pros and cons of both analysis techniques, helping organizations to choose the approach that is better for their particular needs. We'll discuss WebTrends' own client-side data collection technology, SmartSource Data Management, and why it is superior to other data tagging technologies.

[1] There are many other names that refer to the same technique, including client-side tagging, page tagging, page beacons and page bugs.

[2] Also known as Application Server Provider (ASP) solutions, hosted services are an alternative to purchasing and implementing software, whereby the software is implemented by vendor on their own facilities, accessed via a browser and paid for on a lease basis.

The Challenges with Log File Analysis

Web server log files, such as those produced by Internet Information Server (IIS), iPlanet or Apache, are detailed accountings of *hits* made against a web site. A hit is an individual request made to a web server from a browser. Intuitively one would think a hit corresponds to a page on your site, but in fact most pages contain numerous objects, such as GIFs or JPGs, that each produce a hit in the log file. It is therefore common for a single page request, or *page view*, to result in numerous hits being entered into the log file.

So the first problem with web server log file analysis is that *the vast majority of the data captured in web server log files has limited use in understanding visitor behavior*. Many companies choose to filter out these superfluous hits from the analysis process, but the analysis engine must still read and parse each of these hits before determining if they can be disregarded. This of course adds a performance penalty to the analysis process.

A related but more serious issue with web server log files is the potential accuracy problems they create. *In many cases web server log files do not accurately represent the actual visitor interaction with a web site*. Proxy servers are one of many examples of how analysis results can be distorted. Proxy servers deflect page views against web servers by caching the most frequently requested pages. Local caches have a similar effect, handling browser requests through locally cached pages rather than making a request to the web server. In so doing, these page views are never recorded in the web server log files.

Thus the most popular pages of your site may have a relatively small number of hits appearing in the log files. Likewise, path analysis reports will have questionable accuracy, as an unknown number of page views may be missing from a visit. It's even possible, if not likely, for entire visits to be missing from the web server log files if every page the visitor requested was serviced by a cache.

Other accuracy problems are created by spiders, crawlers, robots and other automated mechanisms. These programs produce artificial hits within web server log files, i.e. hits that are not produced by people using a browser. As a result, the log file can contain hits that are not representative of actual visitor behavior. These accuracy problems are a serious matter for organizations that need a precise understanding of how visitors interact with their site.

The most commonly observed problem with log files is the daily administrative burden they create. *Collecting and analyzing log files from multiple web servers, especially if they are geographically dispersed, is often tedious and sometimes problematic*. The collection of superfluous hits discussed earlier aggravates this problem if large log files need to be transferred via FTP.

One final problem is peculiar to Macromedia Flash-based applications. *Flash applications do not produce log files*. Web server logs will contain hits to pages containing the Flash application along with hits to the application itself, but no information is captured on how visitors behave *within* the Flash application. This is a rather profound problem for organizations choosing to use Flash. On the one hand, Flash promises a far richer user experience than HTML-based pages; but on the other hand, it's not possible to actually analyze the user experience using web server log file analysis.

Data tagging solves all of these problems, while introducing some of its own. If none of the aforementioned issues has a substantial impact on your business, feel fortunate and continue to use your log file analysis tool as you do today. Otherwise read on. We'll continue by discussing how data tagging works.

How Data Tagging Works

Data tagging, as the name implies, works by embedding tags on your web pages that transmit relevant data to a centralized data collection server, facility or, in the case of WebTrends, the SmartSource Data Collector. The data collected is then analyzed directly from the Data Collector. The data tag itself is a small piece of scripting code, typically JavaScript, which transmits page-specific information via query string parameters. While the specific implementation varies between web analytics vendors, the general approach is the same across all commercial products and services that utilize data tagging.

The process begins when a page containing a SmartSource Tag is requested from the web server. The tag is a piece of scripting code (either JavaScript or VBScript in SmartSource) that is executed when the page is loaded into the browser. The primary purpose of the SmartSource Tag is to construct a GET request to the Data Collector for a 1x1 invisible GIF file. In addition to the name of the GIF file, the GET request contains query parameters loaded with page-specific data. The GIF file actually serves no purpose other than to provide a vehicle for transmitting the query string.

The SmartSource Tag constructs the query string by scanning the document source for HTML meta tags beginning with a specific identifier ("WT" in the case of SmartSource), such as the following:

```
<META name="WT.mc_n" content="Executive Mailer">
```

This meta tag identifies the page as the landing page for the "Executive Mailer" marketing campaign through the predefined WebTrends query parameter "WT.mc_n." There are many other predefined WebTrends query parameters, including content groups, ad views, ad clicks, scenarios, shopping cart contents, product names, product revenue and more[3]. And of course developers can add their own parameters. Note that SmartSource is one of a few data tagging implementations that segregate page-specific information from the main script, maximizing code modularity and reuse.

Once the document is completely scanned, a GET request is constructed by the SmartSource Tag as follows:

http://SDCHostName/dcs.gif?dcsuri=/Path/URL.HTML&WT.mc_n=Executive%20Mailer

Upon receiving the request, the Data Collector logs the hit into a SmartSource File, a standard W3C log file (the Data Collector is a web server application). But before doing so, it changes the base URL to the URL of the page containing the script, according to the 'dcsuri' parameter. This makes the resulting SmartSource File more conducive to analysis.

The irony of this is that the Data Collector produces a log file – precisely what the data tagging technique is supposed to avoid. But it's worth noting that *there is nothing inherently wrong with log files*. More to the point, there is nothing wrong with log files as a format for capturing large amounts of data. In fact they have proven to be very effective. They provide the most streamlined approach to capturing massive amounts of information, thus maximizing performance of mission-critical systems like web servers, while offloading the analysis work to non-customer-facing hardware. The issue is with the log files produced by web servers, as they cannot provide an accurate accounting of visitor traffic.

[3] SmartSource supports dozens of pre-built query parameters.

Beside data accuracy, data tagging provides other benefits. First of all, the SmartSource Tag is executed with each page view. This means the SmartSource File contains only one hit for each page request made to the web servers, regardless of how many objects are on the page. The result is that SmartSource Files are much smaller than the corresponding web server logs.

Another substantial benefit is that the Data Collector produces a single centralized log file rather than separate logs for each web server. This essentially eliminates the administrative headaches associated with gathering logs from multiple, geographically dispersed servers. The SmartSource File can even contain hits from multiple domains (the domain name can also be passed as a query parameter), allowing visitor behavior to be analyzed across departmental sites or even partner sites provided they permit your tags to be included on their pages.

Taken a step further, data tagging can provide information that is difficult or impossible to obtain otherwise. For example, data tags linked to your Data Collector can be included in your banner ads placed on other sites. SmartSource Tags can also be inserted into Flash applications, permitting a hit to be entered into the SmartSource File for each event fired in the program. This means visitor interaction with Flash applications can be analyzed just as HTML-based sites can.

One final benefit of data tagging is that it facilitates rapid availability of analysis information. Since the data collection is centralized, analysis of the SmartSource File can occur as often as is necessary or practical, unlike web server log file analysis where log files commonly need to be transferred on a nightly basis to the analysis machines. This “real-time” analysis benefit, along with data tagging itself, is often erroneously equated to hosted service solutions only—but in fact it’s applicable to WebTrends software solutions. All WebTrends software and hosted service solutions support data tagging, thus allowing organizations to choose the solution that is best without regard to the data collection methodology. And to date only WebTrends provides both software and hosted service products and supports data tagging in its software products.

Drawbacks and Other Issues with Data Tagging

While data tagging provides more accurate analysis results, eliminates most of the administrative overhead of web server log files and captures information that was otherwise impossible to obtain, there are some drawbacks to the data tagging methodology. The first and most obvious is that data tagging requires development effort. In order for data tagging to work, a piece of script must be placed onto each page of your site (or at least on each page in which you wish to analyze visitor behavior). In many cases this can be accomplished quickly by putting the script in a template, but there are also cases in which the script needs to differ from page to page.

Another problem is that data tagging doesn't capture some information that can be derived from web server log files. Most of this information is related to the hit-level diagnostics that web servers capture but data tagging does not, including the amount of data transferred and web server load balance information. Since information like this is typically not vital on a daily basis, organizations can use traditional web server log file analysis on occasion to gather statistical samples while still utilizing data tagging for all other analysis needs. Note that only WebTrends provides both data tagging and web server log file analysis methodologies.

Some organizations are concerned over the increased bandwidth requirements of the data tags. But in most cases the added data transmissions are small. In SmartSource the script is less than 2KB, and the GIF file is only 43 bytes.

Perhaps the most problematic issue, at least for some organizations, is that data tagging utilizes scripting and cookies (note that WebTrends SmartSource can be implemented without cookies). For most companies this is not an issue, but in some cases, such as government organizations, script is not permitted. Organizations with a strict anti-script policy cannot utilize data tagging.

One function of the SmartSource Data Collector not described earlier is the use of a persistent cookie sent by the Data Collector to the browser (in the default case). In SmartSource this cookie contains a single ID field used to uniquely identify a visitor. This way subsequent visits from the same visitor can be identified and used to determine the lifetime value of the visitor, the recency and frequency of visits and much more. Of all the vendors that provide client-side data collection, only WebTrends does not require the use of a cookie—one of the unique benefits of SmartSource Data Management that we'll discuss next.

Unique Advantages of WebTrends SmartSource Data Management

The most common problems organizations face in implementing data tagging are the use of cookies, with the implied privacy concerns, and the development effort needed to insert, test and maintain the tags. With WebTrends SmartSource Data Management both of these issues are largely neutralized.

First of all and as mentioned earlier, cookies are not required for WebTrends SmartSource Data Collection. But while it may be tempting to simply disable cookies, it's important to know that information on returning visitors cannot be obtained without cookies unless you implement your own visitor identification method, such as authenticated logins or by passing your own cookie. Besides identifying returning visitors, many web analytics vendors use cookies to store information about past visits, such as the date/time of the last visit, past purchase behavior and more. Thus disabling the cookie, if it were even possible with other vendor solutions, would render many marketing reports worthless. But in WebTrends historical visitor information is stored in a server-side table (called the Visitor History Table). So even if cookies are disabled in the visitor's browser, historical information will still be captured. The only thing stored in the WebTrends SmartSource cookie is an ID field used to identify the returning visitor, which can be supplanted by an alternative visitor identification technique.

If cookies are used, WebTrends SmartSource supports an unprecedented set of cookie configuration options. First of all, the software implementation of SmartSource permits the cookie to be 1st-party, as opposed to 3rd-party. A 1st-party cookie is one that is served by your own domain, thus eliminating some security concerns. Conversely, 3rd-party cookies are served by an outside domain, such as a hosted service solution, increasing privacy concerns with cookies that contain more than a basic ID field. When evaluating a web analytics hosted service, it's critical to understand what information is maintained in the cookie[4].

The other primary concern with data tagging is the development effort required to insert, test and maintain the tags on your web site. Most web analytics hosted service vendors merely provide a text file containing the script, requiring developers to manually insert the script into their pages. In order to simplify this process for its customers, WebTrends provides a development kit specifically designed to expedite the insertion of data tags. We'll discuss this, along with the general implementation process next.

[4] Some web analytics vendors introduce severe privacy issues with their cookies. Omniture, for instance, stores historical visitor information for **all** of its customers in a single cookie. Thus one of your competitors or any other Omniture customer can easily obtain visitor information for your site.

Implementing Data Tagging

Incorporating data tags into a site is not much different from inserting any other code. As explained earlier, there are two parts to the tags. The first is the script that constructs the GET request to the SmartSource Data Collector. The second is a series of meta tags that identify the analysis settings that are specific to the page. In general the process of implementing data tags on a site should look something like this:

1. Identify the pages that require data tagging and determine the methodology for inserting the script on each page[5]
2. Determine the analysis settings for each page
3. Insert the scripts and meta tags onto your pages
4. Test and debug the tags before checking code into the staging or live site

The process detailed above largely assumes you are retrofitting an existing site with data tags. But that does not necessarily need to be the case. In fact, the greatest potential benefit to data tagging is that it encourages site designers and developers to think about web analytics in the early stages of design. Organizations that incorporate web analytics into their design work are much more likely to create better user experiences and more effective sites.

To assist in the process, WebTrends offers a free resource from its web site called the WebTrends Developer Kit. The Developer Kit is currently available for Macromedia Dreamweaver MX developers, supporting additional products in the future, and contains a suite of tools designed to streamline the process of tagging and testing a web site with SmartSource Tags.

The first of these tools is a Dreamweaver Extension that minimizes the hand typing of the meta tags. The Extension is a Dreamweaver panel containing a tree listing of the pre-defined WebTrends query extension names and each of their elements, as defined by the developer. The meta tags are inserted in the code window through a simple point-and-click operation. The scripting code itself is added to the pages in a similar way.

Once the pages have been tagged, the developer creates a SmartSource File by previewing the pages in a browser, as normal, with the SmartSource Data Collector that's included in the Developer Kit. The last component of the Developer Kit, WebTrends Reporting Center, is then used to analyze the SmartSource File to test if the tags were implemented properly. Each of these steps is performed on the developer's local machine to tag and test the SmartSource Tags before implementing them on a live site.

Once development is completed, the SmartSource Tags can be used with either the WebTrends Reporting Service or any of the WebTrends software products (except Log Analyzer) by simply changing the domain name of the SmartSource Data Collector from the local test copy to the production server or service. One of the most compelling advantages of SmartSource is that you can easily migrate your web analytics solution from hosted service to software, or vice versa, with virtually no change to the SmartSource Tags.

[5] There are several approaches to inserting the script into the pages of your site. The most obvious is to insert the entire script into each individual page. But a better approach is to embed the script into a template, allowing changes to the script to be made globally. Similarly, the script can be incorporated into an #include file via an <include> tag or via a server-side include.

Summary

Client-side data collection is quickly growing in popularity as a superior approach to collecting web visitor behavior information in many situations. It provides greater reporting accuracy and lower administrative overhead, but at the same time increases development time of a web site and introduces security and privacy concerns for some organizations. As a result, organizations need to carefully analyze the costs and benefits of data tagging versus web server log file analysis and determine which is better, or if both should be used.

Organizations also need to thoroughly evaluate the data tagging technologies of each web analytics vendor they are considering, as some have serious security and privacy issues and others do not help with the implementation of the data tags. WebTrends SmartSource Data Management is clearly superior to other data tagging technologies in the industry:

- SmartSource is the only client-side data collection technology available in both software products and a hosted service, permitting organizations to freely migrate between software and service, or to incorporate a combination of the two.
- SmartSource offers the greatest level of security and privacy protection with the most flexible cookie configuration options in the market.
- Only SmartSource has a complete Developer Kit to assist in the implementation and testing of data tags.

SmartSource is one of many ways WebTrends provides its customers with the flexibility and options they need to maximize the effectiveness of web analytics towards improving their web sites. All organizations recognize that their analytical needs will change over time. It is thus critical to choose a web analytics vendor and solution that will adapt to these changes.

WP10577SSDM 0103