

Implementing a Web Analytics Solution: Perils, Pitfalls and Practical Advice, Part 1

By Gary Angel

Common these days is the search engine optimization of Document Titles. Unfortunately, Google cares more about Document Titles than your users probably do -- so many sites have implemented heavily SE-optimized titling. The results are invariably useless for measurement. Keep this in mind if your current page names are fine but you have SEO on your list of to dos.

Uniting Fault-Tolerance with Redundancy: Discover how Dataside was able to deliver a secure and reliable secondary load-balanced data center site to meet the needs of a major financial institution.

Implementing a tag-based Web analytics solution is rarely the slam-dunk that vendors imply. Depending on how your site is structured and how dynamic your content is, tagging can range from fairly straightforward to fiendishly complex. How can you tell which experience you're likely to be in for?

This article will consider some of the most common stumbling blocks to getting a good tagging implementation, including how to recognize when and where you are likely to have difficulties, and what to do about them. This will include page-naming conventions; creating a site hierarchy; handling dynamic content; setting up conversion processes; custom variable construction; setting up population groups and planning for ASI, or Active Segments; and setting up campaigns and commerce variables.

Examples will be drawn from both SiteCatalyst and HBX, but the basic lessons will be applicable to most enterprise-class Web analytics solutions.

A Brief Introduction to Tagging

To understand where and why you might have tagging issues, it's essential to understand just what a "tag" is.

When you implement a "tag" based Web measurement [solution](#), you typically add a small chunk of Javascript code to every page on the site. For small sites, that sometimes means -- quite literally -- touching every page on the site. However, larger sites tags are almost always implemented by adding the "tag" -- the Javascript code -- to a common page template or to a chunk of HTML that is already included at the top or, more commonly, the bottom of every page of your site.

This area, often called a "Global Footer," has things like the copyright or global navigation options like site map in it. By adding the Javascript tag to this code, you can place the tag on thousands or hundreds of thousands of pages at once.

That tag is designed to execute at some point during the page load -- in most cases at the very end of the page load. When it executes, it sends information about the page load to the tagging vendor. There are a couple of pieces of information that it will always pass -- these include the page URL, the visitor's browser type, the visitor's IP address, and similar information about the visitor and the page.

While all this information is important, however, it isn't the stuff that usually causes any problems. Most tagging problems come into play because the tag needs to pass other information -- specific to your business and Web site -- to the tagging vendor.

Page Naming

The most common additional piece of information to pass is a page name. You don't always have to pass a page name -- some sites use the page URL as the name. But URL's are often long, filled with a mixture of useful and useless parameters, and difficult to understand. If your URLs are already readable and usable, then you've crossed one likely hurdle to tagging difficulties. If your URL's aren't, then you'll have to find a different way to get a page name.

By far, the most common solution is to use the Document Title -- the text that appears up in the Browser Bar. Depending on your page-naming strategy, document titles may make a perfect page name for your measurement solution. If so, congratulations -- but there are many reasons why they probably won't.

Most sites are less than rigorous about giving each page a unique name. They use the same page title -- the same as far as Web Measurement is concerned -- throughout the site. So you can end up conflating multiple pages into a single "page" -- an error that can be difficult to find and diagnose, and can cause havoc with your measurement.

Even more common these days is the search engine optimization (SEO) of Document Titles. Unfortunately, Google (Nasdaq: GOOG) cares more about Document Titles than your users probably do -- so many sites have implemented heavily SE-optimized titling. The results are invariably useless for measurement. Keep this in mind if your current page names are fine but you have SEO on your list of to dos.

So, if you can't use URL or document title, what's left? Nothing really -- except some hard work. If you use a content management system that publishes a

document ID, you can often use this to fix up page names in conjunction with your tagging vendor.

If not, then you will probably need to write some Javascript that parses the URL and makes it into something maximally readable before passing it on. This isn't technically demanding work, but if your staff are new to implementing Web measurement and aren't Javascript experts, you're in for heavy sledding.

Site Hierarchy

Once you've solved your page-naming issue, then next tagging bugaboo is probably going to be establishing a content hierarchy. A content hierarchy serves to group pages of your site together for purposes of analysis and reporting. If you want to know how many unique visitors came to a particular section of the site, you're going to need a hierarchy.

Unfortunately, with most tagging solutions you can't create a hierarchy dynamically on the back-end. (Vendors use the codename "tagless" for functionality that you can achieve without having to embed stuff in your pages. "Tagless" is good, and the more the better!). Usually, the site hierarchy comes from your built-in directory structure. Alas, your built-in directory structure is usually nothing like your actual site structure and is only rarely useful for analysis.

So, once again, you are going to have to put some real work into building this -- or punt on a hierarchy. Unlike page names, hierarchy isn't essential -- just very nice -- so it's no surprise that we see lots of implementations where the choice was to punt and move onto the next issue.

Handling Dynamic Content

Your site is built from simple HTML pages right? Uh no -- probably not. You are using ASP or JSP or asp.net or Flash or one of about a hundred other tools and choices for popping out Web pages. Many of these work almost seamlessly with tagging solutions, but some don't.

Here are the biggest culprits: Flash and ASP.

Flash is not hard to tag -- all of the major tagging vendors provide fairly simple calls that a Flash designer can use to signal various events in the Flash. However, that means opening up your existing Flash objects -- something that most organizations seem loath to do. In addition, it means getting your Flash vendor comfortable with the Web measurement.

For some this will be old hat. Others will make you feel like a child going to the dentist. Minimally, you're going to want to tag for Flash Start, but what about

other Flash Events? Should you treat them as page views? If so, which ones? If your site uses Flash in any significant fashion, be prepared for some tagging pain.

ASP Forms are another common tagging villain. These are typically used for unimportant stuff like your Conversion process! You can't typically punt on handling these -- and if the actual amount of work is fairly small, chances are your IT organization is going to send you back to that childhood dentist for even contemplating changing these pages.

Gary Angel co-founded SEMphonic, a search engine marketing tool provider and Web analytics consultancy, and is president and chief technology officer. He's responsible for leading SEMphonic's development of Web analytics and SEM decision making tools for Web marketing professionals. In addition, he helps companies like WebMD, Intuit, American Express and Charles Schwab maximize their Web channel marketing through intelligent use of Enterprise Web Analytics.
