



WHITE PAPER



SITECATALYST CODE

Size and Caching

March 20, 2006

Version 1.0



1 Overview

SiteCatalyst implementation is a simple process of adding two components (the SiteCatalyst HTML code and a JavaScript library file) to the pages of a web site.

Deploying SiteCatalyst involves pasting HTML code onto each page (or page template) of a web site. This HTML code contains variables and other identifiers that facilitate the data collection process. These variables may be dynamically populated with server or application variables. The code snippet also calls the JavaScript library file, which contains SiteCatalyst-specific JavaScript functions used during metrics collection. Client browsers cache this file after the first request, resulting in virtually no incremental server load for the instrumented site.

Omniture provides a number of methods and technologies to implement the SiteCatalyst data collection. An implementation can consist of the following implementation strategies.

Table 1-A: Implementation Strategies

Standard Code	This strategy minimizes the amount of code that is placed within the HTML page and accurately measures all metrics for modern browsers that support JavaScript. It also includes support for ClickMap, Exit Link and Download Link Tracking, and Form Abandonment. Most functions are contained in the JavaScript library file, which is cached by the browser after the first page.
Non-JavaScript Code	This strategy includes supports all of the functionality of Standard Code, but requires some additional code in the HTML page (an image tag). This option may also require some customization of the code on the HTML page. Users who don't support JavaScript represent about 1.6% of Internet users.

1.1 Caching

The JavaScript file is cached in the user's browser after initially loaded. The file is only downloaded once per session and is not downloaded on each page, although it is used by every page on the site. Therefore, on a site where users average more than a couple of pages per session – which includes most web sites – transferring most of the JavaScript into this file results in less overall downloaded data.

1.2 Code Size

The following table outlines the average size of Omniture's code under various situations. When using the non-JavaScript code option, the size of the HTML image tag varies based on the amount of data gathered from the page. For more information, refer to the *Implementing without JavaScript* whitepaper.

Table 1-B: Average Size of Omniture's Code

	Minimal Code on Page	Standard Code	Non-JavaScript Code
Code on Page	458 bytes	666 bytes	200-2,000 bytes*
Code in JS file	17,392 bytes	17,392 bytes	0 bytes



NOTE: The JavaScript file may be configured to include additional functions called plugins. These plugins, when included, will increase the size of the JavaScript file. For example, the `getQueryParam` plugin consists of 679 bytes.

In a typical session consisting of ten pages, a user would download the following amount of data.

Table 1-C: Sample Code Size

	1-page Visit	5-page Visit	10-page Visit
All variables present (666 bytes/page)	18,058 bytes	20,722 bytes	24,052 bytes
Minimal variables present, no plug-ins (458 bytes/page)	17,850 bytes	19,682 bytes	21,972 bytes

1.3 Compression Options

To decrease download time of the core JavaScript file, while maintaining all functionality of the JavaScript file and associated plugins, you can install a compress utility in your web server. Most browsers support compression (they will uncompress files upon receipt) and some server-side compression utilities support negotiation with the browser to determine whether compression is supported.

Examples of server-side compression utilities include `mod_gzip` or `mod_deflate` for Apache and Apache-based web servers, and the HTTP Compression option in IIS.



CALL 1.877.722.7088
1.801.722.0139

www.omniture.com
info@omniture.com

550 East Timpanogos Circle
Orem, Utah 84097

