



WHITE PAPER



# SITECATALYST IMPLEMENTATION

Implementing without JavaScript

June 30, 2008

Version 2.0





# 1 Implementing without JavaScript

SiteCatalyst data collection is usually implemented using an HTML image tag that is created using JavaScript. The browser then requests the image. Data "piggybacks" this image request by placing variables into the query string of the image request. The JavaScript combines browser-level variables with page-level variables for a comprehensive data collection solution. In some cases, a fully server-created image tag may be appropriate. The standard elements of a JavaScript-based implementation are listed as follows.

**Table 1-A: Standard Elements of a JavaScript-based Implementation**

HTML Code	This portion consists of JavaScript code that is placed in HTML pages (or templates) which set the value of JavaScript variables, and then references a JavaScript library file which is then loaded and executed.
JavaScript Library	This file contains common code that (a) queries the browser about various properties such as JavaScript version, OS version, the size and resolution of monitor is being used, and other variables; (b) encodes and concatenates all the variables into an image request (<img>) that transports these variables to the SiteCatalyst data collection servers. This file is cached in the user's browser and only downloaded once per session.
<noscript> tag	A simplified version of the image request is placed within a <noscript> tag that executes if the user has disabled JavaScript, or does not have JavaScript capabilities. This part of the implementation is optional and generally only applies to approximately 2% of the Internet population.

JavaScript can detect browser settings that are not available to a server, such as browser window height/width, monitor resolution, and Netscape plug-ins. By using a server-side method to create an image tag, these variables cannot be captured. The JavaScript sets a random number in the image request in order to overcome browser and proxy server caching, and thus allows all pageviews to be accurately tracked. In certain situations, server-side code has advantages over the JavaScript-based code, including the following.

- JavaScript is very accurate (98-100%), but there are times when the utmost accuracy may be desired, even in situations where a user may quickly click to another page before the JavaScript has executed. Creating the image tag server-side increases the accuracy level by several percentage points.
- For https: (secure) connections, or for tracking conversion events, such as purchases, where accuracy is very important.
- This strategy may also be used to fully populate the image request within the <noscript> tag for tracking users without JavaScript, or with JavaScript disabled.

**⊗ WARNING!** The use of server-generated image tags requires additional time to implement, and may be more difficult to debug, deploy, and maintain. Omniture strongly encourages clients to use JavaScript-based data collection on every page where possible. Various reports and features, including ClickMap, Download Links, Exit Links and browser-based variables (browser width/height, etc.) cannot be collected or supported using this implementation method.

## 1.1 Variable Names

The names of SiteCatalyst variables differ between the image request and the JavaScript variable name. The implementation manual uses the JavaScript variable name exclusively. The following table maps the JavaScript variable to the query string parameter name in the SiteCatalyst image request.

### 1.1.1 G Code

Table 1-B: G-Code Versions of SiteCatalyst Variables

JavaScript Variable	Query String Parameter
var s_pageName	pageName
var s_server	server
var s_pageType	pageType
var s_channel	ch
var s_prop1 – s_prop50	c1 through c50
var s_campaign	v0
var s_state	state
var s_zip	zip
var s_events	events
var s_products	products
var s_purchaseID	purchaseID
var s_eVar1 – s_eVar50	v1 through v50
*Link Type	pe (lnk_d, lnk_e, lnk_o)
*Link Name	pev2
*Link URL	pev1
Referring URL	r
Current URL	g

## 1.1.2 H Code

**Table 1-C: H-Code Versions of JavaScript Variables**

JavaScript Variable	Query String Parameter
s.pageName	pageName
s.server	server
s.pageType	pageType
s.channel	ch
s.prop1 – s.prop50	c1 through c50
s.campaign	v0
s.state	state
s.zip	zip
s.events	events
s.products	products
s.purchaseID	purchaseID
s.eVar1 – s.eVar50	v1 through v50
*Link Type	pe (lnk_d, lnk_e, lnk_o)
*Link URL	pev2
Referring URL	r
Current URL	g





**NOTE:** When using the image request to track links, the type of link (download=lnk\_d, exit=lnk\_e, or custom link=lnk\_o) must be defined, as does the Link URL/Name (pev2). Note that links would require implementation by hand by inserting code within the <a href> tag.





Additionally, with server-generated image tags, the image request is executed by a web browser. If you need to send data directly from your servers to Omniture's servers, contact Omniture Live Support to determine the best transfer method.

## 2 Other Requirements

The following table outlines additional requirements/configurations for implementing SiteCatalyst without JavaScript. Additionally, you can view sample code to further understand the implementation.

**Table 2-A: Additional Requirements**

Requirement	Description						
Case-Sensitive	The parameter names (pageName, purchaseID, etc.) are case-sensitive and will not properly record data unless they appear as designated in the table displayed in <i>Variable Names</i> in this document.						
Encode Parameters Query	 <p><b>NOTE:</b> The values for each of the query string parameters must be URL encoded. URL encoding converts characters that are normally “illegal” when appearing in a query string, such as a space character, into an encoded character beginning with “%.” For example, a space character is converted into “%20.”</p> <p>The JavaScript version of this function is called “escape” (and to decode, “unescape”). Microsoft IIS Version 5.0 also includes an “Escape” and “Unescape” function for encoding query strings. Other web server scripting languages also provide encoding/decoding utilities.</p>						
Maximum Variable Length	The maximum length of any single parameter is 100 characters. The exception to this list is Current URL and Referring URL, which may be up to 255 characters. Additionally, the “products” variable may be longer, but each individual “sub-field” (such as “Category” or “Product”) must not be longer than 100 characters.						
Invalid Characters	Characters with character codes above decimal 128 are invalid, as are not-printing character codes under 128. HTML formatting (“<h1>”) is also invalid, as are trademark, registered trademark, and copyright symbols.						
Secure (https:> vs. Non-Secure (http:.) Image Requests	<p>ON pages that are accessed via https (secure protocol), the URL portion of the image request changes to accommodate a different set of data collection servers. The following table illustrates the different URLs used for secure and non-secure image requests.</p> <table border="1" data-bbox="527 1409 1430 1690"> <thead> <tr> <th>Protocol</th> <th>URL</th> </tr> </thead> <tbody> <tr> <td>https:</td> <td>https://namespace.&lt;data center-specific...&gt;.2o7.net/b/ss/reportsuite/1/G.5--NS/...</td> </tr> <tr> <td>http:</td> <td>http://namespace.&lt;data center-specific URL*&gt;.2O7.net/b/ss/reportsuite/1/G.5--NS/...</td> </tr> </tbody> </table> <p> <b>NOTE:</b> The * in the URL above denotes a data-center specific URL that will be provided to you by your Omniture Implementation Consultant. Omniture uses several data centers, and it is necessary to implement the correct URL to which your organization has been assigned.</p> <p>For clients who use multiple report suites, the multiple report suites should be listed</p>	Protocol	URL	https:	https://namespace.<data center-specific...>.2o7.net/b/ss/reportsuite/1/G.5--NS/...	http:	http://namespace.<data center-specific URL*>.2O7.net/b/ss/reportsuite/1/G.5--NS/...
Protocol	URL						
https:	https://namespace.<data center-specific...>.2o7.net/b/ss/reportsuite/1/G.5--NS/...						
http:	http://namespace.<data center-specific URL*>.2O7.net/b/ss/reportsuite/1/G.5--NS/...						

	<p>only in the directory section, and not the domain section of the URL, as shown below.</p> <table border="1" data-bbox="535 239 1430 459"> <thead> <tr> <th>Protocol</th> <th>URL</th> </tr> </thead> <tbody> <tr> <td>https:</td> <td>https://102.112.207.net/b/ss/<u>suite1,suite2</u>/1/G.5--NS/...</td> </tr> <tr> <td>http:</td> <td>http://<u>suite1</u>.112.207.net/b/ss/<u>suite1,suite2</u>/1/G.5--NS/...</td> </tr> </tbody> </table> <p> <b>NOTE:</b> The * in the URL above denotes a data-center specific URL that will be provided to you by your Omniture Implementation Engineer. Omniture uses several data centers, and it is necessary to implement the correct URL to which your organization has been assigned.</p>	Protocol	URL	https:	https://102.112.207.net/b/ss/ <u>suite1,suite2</u> /1/G.5--NS/...	http:	http:// <u>suite1</u> .112.207.net/b/ss/ <u>suite1,suite2</u> /1/G.5--NS/...
Protocol	URL						
https:	https://102.112.207.net/b/ss/ <u>suite1,suite2</u> /1/G.5--NS/...						
http:	http:// <u>suite1</u> .112.207.net/b/ss/ <u>suite1,suite2</u> /1/G.5--NS/...						
<p>URL and Referring URL</p>	<p>The URL and Referring URL may be populated from the server in the “g=” and “r=” variables. Use the “Request ServerVariables (“HTTP_REFERER”) or “Request ServerVariables (“URL”)” (IIS/ASP), or the appropriate variable for your server/scripting technology. The referring URL (r=) is extremely important for tracking referring URLs, domains, search engines, and search terms.</p> <p> <b>NOTE:</b> If “pageName” is not being used, it is imperative that the Current URL field is uniquely populated. If neither pageName nor Current URL (g=) are populated, the record is invalid and is not processed. At a minimum, the URL is a required field in order to process the record.</p>						
<p>Effects of Caching</p>	<p>HTML and other web pages can be cached by browsers or servers that are between the visitor and the web site that is serving the content. Caching prevents an accurate count of page views and other events unless a “cache-busting” technique is employed.</p> <p> <b>NOTE:</b> Omniture’s standard JavaScript includes a dynamic method of changing the image request to avoid page and image caching, allowing an accurate count of page views. However, in creating a server-side image request, this randomization does not occur. Page reloads and cached pages (either in the browser’s cache or in a proxy server) will not be counted in certain cases when using server-side image requests. SSL (https:) pages are not, by definition, ever cached so this warning applies only to non-secure (http:) pages. Additionally, pages with parameters (http://www.samplesite.com/page.asp?parameter=1) or certain file extensions (.asp, .jsp, etc.) are also not cached.</p> <p>The examples below also illustrate a “minimal JavaScript” solution that primarily assembles the image request server-side, and then tacks on a random number in the browser. This method overcomes the caching that would otherwise be encountered on static HTML pages accessed via the http: protocol.</p>						
<p>nameSpace Variable</p>	<p>The nameSpace query string parameter is required for non-JavaScript implementations.</p> <p>Example: ns=nameSpace</p> <p> <b>NOTE:</b> Contact your Omniture Implementation Consultant to obtain your organization’s nameSpace value.</p>						

## 2.1 Sample Code

The following sample illustrates the use of a server-generated image tag within a HTML sample page. The table below displays the values used in the sample.

**Table 2-B: Values Used in the Sample Code**

Variable	Value
pageName	Order Confirmation
Current URL	https://www.somesite.com/cart/confirmation.asp
events	purchase,event1
c1	Registered
purchaseID	0123456
products	Books;Book Name;1;19.95
state	CA
zip	90210
a random #	123456

### 2.1.1 Example 1

The example below displays a server-side image tag. The random number (in yellow) prevents caching of the image.

```
<html>
<head>
</head>
<body>
Order Confirmation<br>
Thanks for your order #0123456.

</body>
</html>
```

### 2.1.2 Example 2

The example below shows a minimal JavaScript image tag.

```
<html>
<head>
```



```
</head>
<body>
Order Confirmation<br>
Thanks for your order #0123456.
<script language="javascript"><!--
s.s_date = new Date();
s.s_rdm = s.s_date.getTime();
s.s_desturl="";
document.write(s.s_desturl);
//--></script>
</body>
</html>
```

## Appendix A: Supported SiteCatalyst Reports

Most of the out-of-the-box SiteCatalyst reports are available with a non-JavaScript implementation. However, some of the reports do require JavaScript in order to show data; i.e. the Browser Height report takes JavaScript information from the browser. Therefore, if you decide you do not want to implement with JavaScript, then you will not have every SiteCatalyst report available for use, but you can work with Omniture ClientCare to use other variables to populate those reports. For more information, contact Omniture ClientCare.



CALL 1.877.722.7088  
1.801.722.0139

[www.omniture.com](http://www.omniture.com)  
[info@omniture.com](mailto:info@omniture.com)

550 East Timpanogos Circle  
Orem, Utah 84097

