



WHITE PAPER



# SITECATALYST JAVASCRIPT CODE

Upgrading from G Code to H Code

June 12, 2008

Version 2.0





# 1 SiteCatalyst JavaScript Code

Omniture has released a new version of the JavaScript code, also referred to as H code, which allows multiple versions of the SiteCatalyst JavaScript file to be included on the same page without interference. The main differences between G code and H code include object-orientation, flash tracking, and custom link tracking, as described in the following sections.

If you have many pages that may be referencing the G code version of the JavaScript file, Omniture recommends leaving the existing JavaScript file on your server. For example, many companies have third-party implementations that reference a JavaScript file hosted on the main server. If the G version of the JavaScript file is left in place, those pages will continue to be tracked. If you would like to know how many and which pages are still using the G version of your JavaScript file, consult your Omniture Implementation Engineer about creating a VISTA rule to identify the page views that were created by the G version of the JavaScript file.

## 1.1 Advantages to Upgrading to H Code

The advantages to an H Code upgrade are listed below and described in the following sections.

- Object-oriented code allows for multiple versions of JS on same page
- Simpler custom link tracking
- Simpler Flash tracking (outside of ActionSource tracking)
- Slightly smaller file size
- Larger custom plug-in library
- Will support future SiteCatalyst functionality
- Video tracking requires H15
- Some Genesis integrations require H code to use them.
- Upgrades to newer versions of H code will be very easy

### 1.1.1 Object-Oriented Code

In H code, all variables are declared as members of an object, which is instantiated in the JavaScript file, and then altered by assigning object-specific values on the HTML page. Since all variables and functions are members of a global object, all JavaScript syntax is different than in G code.



**NOTE:** H code uses 's' as the default object of reference. In cases where multiple instances of H code are used on the page, it is possible to change the object to customized object name (for example, 'mysite' as in 'mysite.prop1').

### 1.1.2 Custom Link Tracking

In addition to using object-oriented techniques, H code introduces a new function called `tl` (or track link). The `tl` function replaces the `s_inK`, `s_linkType` and `s_linkName` in G code, and replaces the call to `s_gs`. Overall, custom link tracking is much simpler with H code, but any custom link tracking with G code syntax will not work with an H code version of the JavaScript file.

### 1.1.3 Flash Tracking

As with G code, Flash tracking is built on top of the Custom Link Tracking syntax and functions. Consequently, all Flash objects that reference G code functions may need to be recompiled to include H code function calls. Instead of calling `sendAnalyticsEvent()` and `sendLinkEvent()`, Flash objects should call `t()` and `tl()` respectively.

### 1.1.4 Smaller File Size

The H Code is actually smaller than G Code. The benefit to this smaller file size is that the code actually takes a shorter time to load than G Code does. See *Code to Paste* in this document for an example of the two code types.

### 1.1.5 Custom Plug-in Library

The Omniture plug-ins offer an extension of the functionality of the core JS files. One benefit to upgrading to H Code is that the plug-ins are no longer developed for G Code. All existing plug-ins are developed for H Code, and all new plug-ins are engineered to work specifically with H Code or later.

### 1.1.6 Future SiteCatalyst Functionality

Though many of the SiteCatalyst features will work with G Code, one of the advantages to upgrading to H Code is that you will be assured that all current SiteCatalyst features can be used to H Code. For example, video tracking cannot be implemented with G Code, but you can implement it with H Code (specifically v15 of the H Code). Similarly, some Genesis integrations also require H Code. As Omniture continues to develop SiteCatalyst and other products, H Code will become even more necessary in order to implement the advanced functionalities offered by Omniture.



**NOTE:** Additionally, once you have implemented H Code, future upgrades to new versions of H Code will be much easier.

### 1.1.7 Future Code Upgrades

Omniture regularly release new versions of the code. Once you have successfully implemented H Code, future upgrades are simply a matter of using the SiteCatalyst Code Generator to download new versions of both the JavaScript code and the SiteCatalyst page code.

## 2 Code Changes

Several changes have been made to the Code to Paste, JavaScript Library File, and Plugins Code, all of which are used in the SiteCatalyst data collection process. Each of the changes is outlined in the following sections.

### 2.1 Code to Paste

The following table displays the differences between the G code (on left) and the H code (on right).

<pre> &lt;!-- SiteCatalyst code version: G.9. Copyright 1997-2003 Omniture, Inc. More info available at http://www.omniture.com --&gt;&lt;script language="JavaScript"&gt;&lt;!-- /* You may give each page an identifying name, server, and channel on the next lines. */ var s_pageName="" var s_server="" var s_channel="" var s_pageType="" var s_prop1="" var s_prop2="" var s_prop3="" var s_prop4="" var s_prop5="" /* E-conversion Variables */ var s_campaign="" var s_state="" var s_zip="" var s_events="" var s_products="" var s_purchaseID="" var s_eVar1="" var s_eVar2="" /***** INSERT THE DOMAIN AND PATH TO YOUR CODE BELOW *****/ //--&gt;&lt;/script&gt;&lt;script language="JavaScript" src="//INSERT-DOMAIN- AND-PATH-TO-CODE-HERE/s_code_remote.js"&gt;&lt;/script&gt; &lt;!-- End SiteCatalyst code version: G.9. --&gt; </pre>	<pre> &lt;!-- SiteCatalyst code version: H.0. Copyright 1997-2005 Omniture, Inc. More info available at http://www.omniture.com --&gt; &lt;script language="JavaScript" src="//INSERT-DOMAIN-AND-PATH-TO- CODE-HERE/s_code.js"&gt;&lt;/script&gt; &lt;script language="JavaScript"&gt;&lt;!-- /* You may give each page an identifying name, server, and channel on the next lines. */ s_pageName="" s_server="" s_channel="" s_pageType="" s_prop1="" s_prop2="" s_prop3="" s_prop4="" s_prop5="" /* E-conversion Variables */ s_campaign="" s_state="" s_zip="" s_events="" s_products="" s_purchaseID="" s_eVar1="" s_eVar2="" /***** DO NOT ALTER ANYTHING BELOW THIS LINE ! *****/ var s_code=s.t();if(s_code)document.write(s_code)//--&gt;&lt;/script&gt; &lt;!-- End SiteCatalyst code version: H.0. --&gt; </pre>
--	--

The table above shows the Code to Paste for both G code and H code. The differences between the two code versions are highlighted in yellow. First, in G code, the location in which you reference the JavaScript file is located at the bottom of the code, while in H code, the reference is located at the top of the Code to Paste. Second, var is removed from all variable names in H code with the exception of the s\_account variable. Similarly, s\_ has been changed to s. in all variables with the exception of s\_account and s\_objectID. Next, the name of the .JS file has changed from s\_code\_remote.js in G code to s\_code.js in H code. Finally, the s.t function has been added to the Code to Paste so that the image request can be automatically sent after the variables are loaded, which enables the user to determine the point at which the image request is sent.

## 2.2 JavaScript Library File

The name of the JavaScript Library File has changed from s\_code\_remote.js in G code to s\_code.js in H code. Additionally, in the Code to Paste in the previous section, the G code displays the JavaScript Library File at the top of the code, while the H code displays the JavaScript Library File at the bottom of the code. The H code version in the Code to Paste shown above highlights the new name of JavaScript Library file.



**NOTE:** Omniture offers an infrequently used version of G code with s\_code.js as a file name.

## 2.3 Plugins


The following table outlines the differences between the plugin code used in G code and H code. G code is displayed on the left, and H code is displayed on the right.

<pre>/* Plugin Config */ var s_usePlugins=true function s_doPlugins() {   /* Add calls to plugins here */   // External Campaigns   s_vp_getCGI('s_campaign','source'); }</pre>	<pre>/* Plugin Config */ s.usePlugins=true function s_doPlugins(s) {   /* Add calls to plugins here */   // External Campaigns   s.campaign=s.getQueryParam('source') } s.doPlugins=s_doPlugins;</pre>
---	--

In the code above, the s\_vp\_getCGI plugin is replaced by s.getQueryParam. H code version of the other plugins used in your JavaScript file will be provided by Omniture. For more information, contact your Omniture Implementation Engineer.

## 3 Implementing Code Changes

When upgrading to H code, change the code on all HTML pages to the format outlined on page 2 of this document. Additionally, you should consider any custom links or Flash code you have implemented. The following sections describe the changes required to custom links and Flash tracking.

 **WARNING!** If you upgrade to H code in the JavaScript file, ensure that you also upgrade to H code in Flash and Link Tracking; otherwise, you will receive JavaScript errors when Flash executes and visitors click custom links.

### 3.1 Custom Link Code Changes

As noted above, Custom Link Tracking has a new syntax in H code. The most noticeable difference is the introduction of the tl() function (track link). The following table outlines the difference between custom link tracking in G and H code. G code is on the left, and H is on the right.

<pre>onClick="s_prop1='my value'; s_linkType='o'; s_linkName='my link'; s_link=s_co(this);" s_gs('reportsuite1');"</pre>	<pre>onClick="s=s_gi('reportsuite1'); s.prop1='my value'; s.tl(this,'o','my link');"</pre>
--	--

Notice the call to the global s\_gi() function in the example above. This is the 'get instance' function, which returns a pointer to the object identified by 'reportsuite1.' While not required, this ensures that you are referencing the correct object in case there are multiple objects on the page and ensures that s is pointing to an actual object, in case a global variable called 's' was used elsewhere in the HTML page.



**NOTE:** In the above example, replace 'reportsuite1' with the report suite ID. In addition, s.prop1 is optional and is present to illustrate the use of property variables in a custom link with H code. Please remember to set the s.linkTrackVars and s.linkTrackEvents as appropriate in H code (similar to G code) when using custom variables (props or eVars) in a custom link.

### 3.2 Flash Code Changes

The sendAnalyticsEvent and sendLinkEvent functions are no longer used in H code because of the new architecture and functions. The sendAnalyticsEvent and sendLinkEvent functions are replaced with t() and tl() respectively. As of the time of this writing, the Macromedia component available in G code has not yet been configured to support H code. The following examples show the difference between G and H code when sending a link event or a page event.

#### 3.2.1 Sending a Page View Event in Flash

<pre>getURL("s_prop1='my value'; s_pageName='flash page 1'; sendAnalyticsEvent('reportsuite1')");</pre>	<pre>getURL("s=s_gi('reportsuite1'); s.prop1='my value'; s.pageName='flash page 1'; s.t()");</pre>
---	--

Notice that in H code (right) s\_gi is called to insure that the variable 's' points to the correct instance of the object. The t() function is what sends the data to SiteCatalyst.

### 3.2.2 Sending a Link Event in Flash

```
getURL("s_prop1='my value';  
sendLinkEvent('reportsuite1')");
```

```
getURL("s=s_gi('reportsuite1');  
s.prop1='my value';  
s.tl(true,'o','flash link')");
```

Notice that the tl() function is used in the same way that it is used in Custom Link Tracking. Rather than sending 'this' as the first parameter, in Flash Tracking you should use the boolean value 'true.'

### 3.3 Sample Upgrade from G to H-Code

The upgrade to H-code from G-code presents a challenge given the inherent object-oriented behavior of H-code. H-code pages that were specifically tagged with s\_pageName, s\_prop1, etc. will need to be changed to s.pageName, s.prop1, etc. in order for H-code to work correctly.

In order to simplify this upgrade process, it may be easier to include a section of Javascript code that copies all of the G-code variable values into their corresponding H-code equivalents. This code is placed in the Javascript include file for H-code.

This code uses up to five props and five eVars so if you are using additional custom variables, please ensure these are added to the list. Also, this code is intended to simplify the upgrade process and will still require page level changes as described in *Code to Paste* in this document. Specifically, the call to the t() function is required to be collected.



**NOTE:** This code requires the existence of the corresponding G-code variables. If you are not using these variables in your implementation, please remove the appropriate lines from the code below.

```
// Code to copy G-code variable values to H-code equivalents  
s.pageName = s_pageName;  
s.channel = s_channel;  
s.events = s_events;  
s.products = s_products;  
s.purchaseID = s_purchaseID;  
s.prop1 = s_prop1;  
s.prop2 = s_prop2;  
s.prop3 = s_prop3;  
s.prop4 = s_prop4;  
s.prop5 = s_prop5;  
// Insert addition s_props  
s.eVar1 = s_eVar1;s.eVar2 = s_eVar2;  
s.eVar3 = s_eVar3;  
s.eVar4 = s_eVar4;  
s.eVar5 = s_eVar5;  
// Insert addition s_eVars  
s.hier1 = s_hier1;
```

Though the example given above is useful, the recommended approach to upgrading from G code to H code is to change the JavaScript on every page being tracked to reference the H code object. For example, change all instances of s\_pageName to s.pageName.



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

