OMNITURE™

WHITE PAPER

LINK TRACKING

File Downloads, Exit Links, and Custom Links

April 23, 2008 Version 3.0

1 Link Tracking

SiteCatalyst will automatically track file downloads and exit links based on parameters set in the JavaScript file. If needed, these types of links can be manually tracked using custom link code as explained below. In addition, custom link code can be used to track generic custom links which can be used for a variety of tracking and reporting needs. The following link types and reports are reported in SiteCatalyst:

Table 1-A: Link Types and Reports

Type	Report Location
File Downloads	Traffic > Site Traffic > File Downloads (SiteCatalyst v.13.5) Site Content>Links>File Downloads (SiteCatalyst v.14)
Exit Links	Paths > Entries & Exits > Exit Links (SiteCatalyst v.13.5) Site Content>Links>Exit Links (SiteCatalyst v. 14)
Custom Links	Traffic > Custom Insight > Custom Links (SiteCatalyst v.13.5) Site Content>Links>Custom Links (SiteCatalyst v.14)

Because clicking a link often takes a visitor off the current page, a 500 ms delay is used to ensure that an image request is sent to Omniture before the user leaves the page. This delay is only necessary when leaving the page, but is typically present when the tl() function is called. If you want to disable the delay, pass the keyword 'true' as the first parameter when calling the tl() function. The first parameter is typically the object being clicked on, but if true is used, there will be no delay. Here are two examples.

- Use a 500 ms delay to insure data is collected before leaving the page.

  s.tl(this,'o','link name')
- Disable the 500 ms delay when the user is not going to leave the page.

  s.tl(true,'o','link name')

The 500ms delay is a maximum delay. If the image requested returns in less than 500 ms, the delay will stop immediately, allowing the visitor to move onto the next page or next action within the page.

1.1 Automatic Tracking of Exit Links and File Downloads

The JavaScript file can be configured to automatically track file downloads and exit links based on parameters that define file download file types and exit links. The parameters that control automatic tracking are as follows.

```
s.trackDownloadLinks=true
s.trackExternalLinks=true
s.linkDownloadFileTypes="exe,zip,wav,mp3,mov,mpg,avi,doc,pdf,xls"
s.linkInternalFilters="javascript:,mysite.com,[more filters here]"
s.linkLeaveQueryString=false
```

The parameters trackDownloadLinks and trackExternalLinks determine if automatic file download and exit link tracking are enabled. When enabled, any link with a file type matching one of the values in linkDownloadFileTypes will be automatically tracked as a file download. Any link with a URL that does not contain one of the values in linkInternalFilters will be automatically tracked as an exit link.

1.1.1 Example 1

The file types "jpg" and "aspx" are not included in linkDownloadFileTypes above; therefore, no clicks on them will be automatically tracked and reported as file downloads.

The parameter linkLeaveQueryString modifies the logic used to determine exit links. When linkLeaveQueryString=false, exit links are determined using only the domain, path and file portion of the link URL. When linkLeaveQueryString=true, the query string portion of the link URL is also used to determine an exit link.

1.1.2 Example 2

With the following settings, the example below will be counted as an exit link.

JS file
```
s.linkInternalFilters="javascript:,mysite.com"
s.linkLeaveQueryString=false
```

HTML
```
<a href='http://othersite.com/index.html?r=mysite.com
```

1.1.3 Example 3

With the following settings, the link below will NOT be counted as an exit link.

JS file
```
s.linkInternalFilters="javascript:,mysite.com"
s.linkLeaveQueryString=true
```

HTML
```
<a href='http://othersite.com/index.html?r=mysite.com
```

NOTE: A single link can only be tracked as a file download or exit link, with file download taking priority. If a link is both an exit link and file download based on the parameters linkDownloadFileTypes and linkInternalFilters, it will be tracked and reported as a file download and not an exit link. The following table summarizes the automatic tracking of file downloads and exit links.

Table 1-B: Automatic Tracking of File Downloads and Exit Links

Type	Description and Report Location
File Downloads	Any URL with a file type that is listed in linkDownloadTypes is considered a download link. Report: Traffic > Site Traffic > File Downloads (v. 13.5) Site Content>Links>File Downloads (v.14)
Exit Links	Any URL that does not contain one of the values in linkInternalFilters is considered an exit link. Determination of an exit link is modified by the setting of linkLeaveQueryString. Report: Paths > Entries & Exits > Exit Links (v.13.5) Site Content>Links>Exit Links (v.14)

1.2 Manual Link Tracking Using Custom Link Code

Custom link code allows file downloads, exit links and custom links to be tracked in situations where automatic tracking is not sufficient or applicable. Custom link code is typically implemented by adding an onClick event handler to a link, or adding code to an existing routine; however it can be implemented from essentially any JavaScript event handler or function.

NOTE: Uses of custom link code are often very specific to your Web site and reporting needs. You may wish to contact your Implementation Consultant before implementing custom link code.

The basic code to track a link using custom link code is shown in the following example.

```
<a href="index.html" onClick="
 var s=s_gi('rsid'); **see note below on the rsid**
 s.tl(this,'o','Link Name');
 ">My Page</a>
```

NOTE: The s_gi function must contain your report suite ID as a function parameter. Be sure to swap out rsid for your unique report suite ID.

The s_gi function must contain the report suite ID for reporting. The tl function accepts 3 parameters: the 'this' object (a required parameter), the link type ('e' for an exit link, 'd' for a file download, or 'o' for a generic custom link), and the link name.

NOTE: If the link name parameter is not defined, the URL of the link (determined from the "this" object) will be used as the link name.

In addition, SiteCatalyst variables can be defined as part of custom link code.

1.3 Setting Additional Variables for File Downloads, Exit Links, and Custom Links

Two SiteCatalyst parameters (linkTrackVars and linkTrackEvents) control which SiteCatalyst variables are set for file downloads, exit links and custom links, and are, by default, set within the JS file as follows.

```
s.linkTrackVars="None"
s.linkTrackEvents="None"
```

The linkTrackVars parameter should include each variable that you wish to pass to SiteCatalyst with every file download, exit link and custom link. The linkTrackEvents parameter should include each event you want to pass to SiteCatalyst with every file download, exit link and custom link. When one of these link types occur, the current value of each variable identified will be passed to SiteCatalyst.

For example to track "prop1", "eVar1", and "event1" with EVERY file download, exit link, and custom link, use the following settings within the global JS file:

```
s.linkTrackVars="prop1,eVar1,events"
s.linkTrackEvents="event1"
```

NOTE: The variable pageName cannot be set for a file download, exit link or custom link, which is because each of the link types is not a page view and hence does not have an associated page name.

NOTE: If linkTrackVars (or linkTrackEvents) is null (or an empty string) then ALL SiteCatalyst variables (or events) that are defined for the current page will be passed to SiteCatalyst, which will most likely inflate instances of each variable inadvertently and should be avoided.

1.3.1 Best Practices

The settings for linkTrackVars and linkTrackEvents within the JS file affect every file download, exit link and custom link. Instances of each variable and event can be inflated in situations where the variable (or event) applies to the current page but not the specific file download, exit link or custom link.

To ensure the proper variables are set with custom link code, Omniture recomends setting linkTrackVars and linkTrackEvents within the custom link code, as follows.

```
<a href="index.html" onClick="
  var s=s_gi('rsid');
  s.linkTrackVars='prop1,prop2,events';
  s.linkTrackEvents='event1';
  s.prop1='Custom Property of Link';
  s.events='event1';
  s.tl(this,'o','Link Name');
  ">My Page</a>
```

The values of linkTrackVars and linkTrackEvents override the settings in the JS file and ensure that only the variables and events specified in the custom link code are set for the specific link.

NOTE: In the above example, the value for prop1 is set within the custom link code itself, but the value of prop2 comes from the current value of the variable as set on the page.

1.4 Using Function Calls with Custom Link Code

Due to the complex nature of custom link code, you can consolidate the code into a self contained JavaScript function (defined on the page or in a linked JavaScript file) and make calls to the function within the onClick handler, as shown in the following example.

```
<script language=JavaScript>
function linkCode(obj) {
 var s=s_gi('rsid');
 s.linkTrackVars='None';
 s.linkTrackEvents='None';
 s.tl(obj,'d','PDF Document');
}
</script>
…
<a href="document.pdf" onClick="linkCode(this)">My Page</a>
```

NOTE: The approach above does have implications for ClickMap accuracy of custom links. Refer to the *ClickMap* white paper.

With the method shown above, it is possible for the link to be double counted in situations where the link would normally be captured by automatic file download or exit link tracking. In the example above, the file "document.pdf" would be reported as a file download by automatic means, AND the link name "PDF Document" would be reported as a file download by the custom link code. To ensure link double counting does not occur, use the following modified JavaScript function.

```
<script language=JavaScript>
function linkCode(obj) {
 var s=s_gi('rsid');
 s.linkTrackVars='None';
 s.linkTrackEvents='None';
 var lt=obj.href!=null?s.lt(obj.href):"";
 if (lt=="") { s.tl(obj,'d','PDF Document'); }
}
</script>
```

The last two lines of the code above modify the behavior of custom link code so that only the automatic tracking behavior occurs; eliminating any possible double counting.

NOTE: You can pass the link type and link name as additional parameters for the JavaScript function.

1.5 Validating File Downloads, Exit Links, and Custom Links

Since file downloads, exit links and custom links are not page views, the Debugger tool cannot be used to verify parameters and variable settings. To fully validate download, exit, and custom links, Omniture recommends using a packet sniffer utility to examine the links in real-time. Ethereal (http://www.ethereal.com) is a Java-based IP packet sniffer used by the Engineering and Implementation Teams at Omniture.

OMNITURE™

CALL 1.877.722.7088
1.801.722.0139

www.omniture.com
info@omniture.com

550 East Timpanogos Circle
Orem, Utah 84097

OMNITURE™