OMNITURE™

WHITE PAPER

EVENT SERIALIZATION

Counting Event Instances

August 7, 2008 Version 2.0

1 Event Serialization

Event serialization is the process of removing duplicate events on each page view of the site with SiteCatalyst tags. Event serialization is useful in the following instances.

- A page may be reloaded or refreshed and repeatedly send an event to SiteCatalyst. Event serialization prevents events from being recounted by using a "serial number" for each event.
- The user saves the page to his/her machine for later review—a scenario that is quite common on purchase confirmation pages to review purchase receipts. Event serialization prevents the subsequent page reloads from re-counting the events.

This document describes the process to implement event serialization for conversion and custom events.

NOTE: In order to use event serialization, you must first contact Omniture Live Support in order to have it enabled.

1.1 Default Behavior

The default behavior of SiteCatalyst code is to count each instance of an event. That is, each pageview for which an event is set will be counted, even on page reloads or page refreshes. The "s.purchaseID" variable is used in order to uniquely identify each order (purchase). This allows a user to reload the order page without recounting the order, revenue, or products. A similar feature is available for all events within SiteCatalyst, including pre-defined events such as "prodView" and "scCheckout", as well as all custom events.

1.2 Methods of Event Serialization

There are two methods of keeping events from counting more than once. The first way is to use a unique identifier to allow an event to fired only once per unique ID. The second way is to use SiteCatalyst's database to allow an event to be fired only once per visit. Each method will be discussed here. To implement event serialization, just provide a unique ID for the event, for example event1:1234ABCD.

1.2.1 Event Serialization – Once Per Unique ID

Once event serialization is implemented, if SiteCatalyst receives a duplicate number, it will ignore the event (an event is counted only once per unique value). If the number is unique, another event instance will be counted, as shown in the following example.

User Name	Description	Event Syntax	SiteCatatalyst Total Event1 Count
John	User views page for the first time	event1:1000	1
John	User reloads the page (a form submit may fail, and cause the page to reload)	event1:1000	1
Stacy	User views page for the first time	event1:1001	2
Stacy	User reloads the page (a form submit may fail, and cause the page to reload)	event1:1001	2
Jill	User views page for the first time, enters information correctly, and moves on to next page.	event1:1002	3
Jamie	User views page for the first time	event1	4
Jamie	User forgets to fill in the last name field on the form. The form is displayed again with the missing information highlighted.	event1	5

s.Events Syntax

Use the following syntax for serializing events. Note that in the first example, event1 is serialized, while event2 is not serialized (an instance will be counted for each pageview or page reload).

NOTE: For G Code, replace s. with s_.

```
s.events="[event]:[serial number]"
```

Samples:

```
s.events="event1:12341234"
s.events="event1:12341234,event2"
s.events="purchase,event1:12341234"
```

Serialization may be applied to both custom events (event1-event20) as well as pre-defined conversion events (prodView, scView, scAdd, scRemove, scOpen, scCheckout). Use the "s.purchaseID" variable in order to serialize the purchase event.

1.2.2 Event Serialization – Once Per Visit

SiteCatalyst offers a feature to only allow an event to be fired once per visit. Give the name of the report suite and event name to Omniture Live Support to enable the event for you.

Serializing Evars

The same functionality used to keep an event from being fired more than once per visit, can be applied to evar instances. When enabled for an evar, click-throughs and instances will only be counted once per visit for a specific evar. Notice that this is not counting once per value, but once per evar. This means that if an evar or campaign variable is set to record once per visit, then only the first value seen in a visit will show a click-through or instance. However, if the evar is set to allocate credit to the most recent value, then the most recent value will still receive credit, even if it does not have an instance. The following example ample will help to illustrate this point.

If s.campaign is set to "Record once per visit", and within a single visit 20 pages are viewed. First, there are 10 pages where s.campaign is set to "abc," and then 10 pages are viewed where s_campaign is set to "xyz." On all 20 pages, event1 is fired. The following screen shot illustrates the results. Notice that there are no instances or Click-throughs associated with xyz, but it does receive credit for all events fired.

Figure 1-A: Tracking Code Report

Details

	Tracking Code	Click-throughs		Custom 1	
1.	abc	1	100.0%	10	50.0%
2.	xyz	0	0.0%	10	50.0%
	Total	1		20	

In most cases Omniture recommends using the setValOnce and setOncePer JavaScript plugins. Those plugins allow a value to be set only once, whereas the SiteCatalyst database will allow the eVar to be set only once.

1.3 Purchase Events

For the purchase event, SiteCatalyst variables are used to capture specific purchase information and s.purchaseID is used to serialize (de-duplicate) the event. Note that "purchaseID" is limited to 20 characters.

CAUTION!: The "purchaseID" variable serializes all events passed in the variable "s.events", and will override any serialization value for events.

NOTE: In the case where "purchaseID" is left blank, each instance of the "purchase" event, even page reloads, would be counted.

Specific server-side code can be used to generate the unique number (alphanumeric value) embedded in the HTML source. Usually the Order ID, or similar alphanumeric value, is used for this purpose. This value should not change if the user was to refresh the page. The following is a short example from the Rapid Deployment Guide for SiteCatalyst (note the "purchaseID" code in bold):

1.3.1 Syntax

```
s.products="Category;ProductName;Qty;total_price
         [,Category2;ProductName2;Qty;total_price]"
s.state="XX"
s.zip="00000"
s.purchaseID="<%=getPurchaseID()%>"
s.events="purchase"
```

1.3.2 Examples

```
s.products="Footwear;Hiking Boots (1234);1;170.00"
s.state="UT"
s.zip="84097"
s.purchaseID="12341234"
s.events="purchase"
```

1.4 Additional Notes

- Be sure to use server-side code to generate the unique identifier for the event. Do not use a JavaScript randomization function to generate a number, which will generate unique numbers each time the page is loaded (each instance/pageview would count).
- The unique identifiers are applicable to all users across all sessions. Therefore, ensure the identifier is unique across users and sessions. For instance, if the Order ID repeats after 30 days, append the date of the order in order to make the Order ID sufficiently unique.
- The serialization value may be alphanumeric values up to 20 characters in length. This is identical to the limitations of "s.purchaseID" (replace s. with s_ for G Code).
- The "s.purchaseID" variable serializes all events passed in the variable "s.events," and will override any serialization value for events. Do not use event serialization for any events if the "s.purchaseID" variable is used on the current page (replace s. with s_ for G Code).

OMNITURE™

CALL 1.877.722.7088
1.801.722.0139

www.omniture.com
info@omniture.com

550 East Timpanogos Circle
Orem, Utah 84097

OMNITURE™