



WHITE PAPER



# FORM ANALYSIS PLUG-IN (H CODE)

Setup and Configuration

February 21, 2006

Version 1.0





# 1 Implementation

The Form Analysis Plug-in is an additional function that is added to the JavaScript library. The primary function of the Form Analysis Plug-in is to track the use of forms on your site. Possible outcomes of form usage include abandonment, success, or error. Examples of form elements include text boxes, radio buttons, drop-down boxes, et cetera.

If a visitor accesses a web page that contains a form being tracked by SiteCatalyst, the Form Analysis plug-in immediately recognizes the form. SiteCatalyst reports abandonment of the form, successful use of the form, or errors in using the form.

Form analysis is useful when you want to understand why some visitors to your site are leaving pages that contain forms. For example, if your Shopping Cart Page contains a field in which the visitor is to enter their credit card number, but the visitor departs the page or site at that point, you will better be able to manage your site and forms by knowing where and why your visitors exit the page.

To implement the Form Analysis Plug-in contact Omniture Live Support. Your Implementation Consultant will gather the appropriate information to configure the Form Analysis plugin before creating a new JavaScript file. For example, the Implementation Engineer will need to know whether Form Analysis data should be displayed in a Traffic Custom Insight Report or in a Commerce Custom Evar Report (or both). The *List of Variables* section this document outlines all of the configurations that the Implementation Engineer will assist you with.

Once your JavaScript file has been updated, you will receive a new JavaScript that should replace the SiteCatalyst JavaScript file currently being used on your site. Additionally, the modifications made to the .js file by the Omniture Implementation Consultant will include functions that allow you to track errors and successes for Forms on your site. For more information on adding functions to the .js file, refer to *Form Errors and Form Successes* in this document.

## 1.1 Initial Setup Summary

To use the Form Analysis plugin, follow the steps below.

1. Request that your Implementation Consultant update your JavaScript file to contain the Form Analysis plugin.
2. Make sure the doPlugins function contains a call to the setupFormAnalysis function.
3. Configure the Form Analysis variables to track the forms of your choice.

The following is an example of how the code may look in your JavaScript file. For more information on the role of each configuration variable, refer to *Configuring Reports for Traffic or Commerce* and *Configuration Variables* in this document.

```
/* Form Analysis Config */
s.formList="ContactForm,registration"
s.trackFormList=true
s.trackPageName=false
s.useCommerce=false
s.varUsed="prop6"
s.eventList="" //Abandon, Success, Error

/* Plugin Config */
s.usePlugins=true
function s_doPlugins(s) {
  /* Add calls to plugins here */
  s.setupFormAnalysis();
}
s.doPlugins=s_doPlugins
```

## 1.2 Configuring Reports for Traffic or Commerce

You can decide whether form analysis data will be displayed in the Traffic or Commerce section of the SiteCatalyst reporting module. For example reports, refer to the *Form Analysis Plugin Overview* whitepaper. Based on your decision of viewing the reports in Traffic or Commerce reports, your Omniture Implementation Consultant will make the required code changes and variable configurations. When the code changes are complete, your Implementation Consultant will send you a new JavaScript file, which should replace the existing JavaScript file on your site.

### 1.2.1 Variable Configuration

The following table illustrates how form analysis variables can be configured to send data into Traffic Custom Insight 6 or Commerce Custom Evar 6 reports.

Traffic Configuration	Commerce Configuration
<pre>/* Form Analysis Config */ s.formList="" s.trackFormList=false s.trackPageName=true s.useCommerce=false s.varUsed="prop6" // Abandon, Success, Error s.eventList=""</pre>	<pre>/* Form Analysis Config */ s.formList="" s.trackFormList=false s.trackPageName=true s.useCommerce=true s.varUsed="eVar6" // Abandon, Success, Error s.eventList="event3,event4,event5"</pre>

As shown in the table above, the variables that determine whether to send data into a Traffic or Commerce report are `useCommerce`, `varUsed`, and `eventList`. If `useCommerce` is set to "true," the plugin expects `varUsed` to contain an eVar (e.g. "eVar6") and `eventList` to contain a list of events. Likewise, if `useCommerce` is set to "false," the plugin expects `varUsed` to be a prop (e.g. "prop6") and ignores `eventList`.

For more information on the variables used in the Form Analysis Plug-in, refer to *Configuration Variables* in this document.

## 1.3 Form Errors and Form Successes

Errors and Successes are metrics that can be displayed in SiteCatalyst. The first step to displaying form analysis data in SiteCatalyst is to make the required modifications to the .js file. The function relating to Form Errors and Form Successes is added to the .js file by the Omniture Implementation Consultant. This function is called `sendFormEvent` and should be called when a form submission contains an error or is successfully submitted. An example of a JavaScript function that is used to send error and success information to SiteCatalyst is displayed below.

```
// Example error checking code
function checkError(){
  // check that the credit card number contains 16 digits
  if( window.document.forms[0].creditcard.value.length!=16 ) {
    s.sendFormEvent('e', 'Contact Us', 'ContactForm',
      'Credit Card: Invalid Length');
    return false;
  }
  // more error checking

  s.sendFormEvent('s', 'Contact Us', 'ContactForm');
```

```
    return true;
}
// call checkError when the form is submitted
window.document.forms[0].onsubmit=checkError;
```



**NOTE:** When using server-side error checking, simply insert a call to `sendFormEvent` in your HTML page *below* the SiteCatalyst JavaScript file.

In the example above, the `sendFormEvent` function uses three or four parameters, as shown here.

```
sendFormEvent( error_type, page_name, form_name, element_error)
```

The following table outlines how each parameter is used.

**Table 1-A: Parameter Use**

Name	Use
error_type	'e' for error or 's' for success
page_name	If trackPageName is set to "true," this parameter should contain the name of the page that contains the form. If false, this parameter should be "".
form_name	The name of the form in which the error occurred.
element_error	A description of the error, including the name of the erring element (if desired).

## 1.4 Reported Fields

The Form Analysis plugin tracks the most recent form element "touched." This means that if a user views a form, but does not take an action that may alter it (though he or she may click in a field), the plugin will report "No Data Entered." If the user interacts with a field by typing in a text box or clicking in a drop-down menu, for example, the last of those form fields touched will be the one that is reported in SiteCatalyst.

## 1.5 Supported Browsers

The Form Analysis plugin uses the `onUnload` JavaScript event to track when visitors leave a page. Most major browsers support the `onUnload` event, including Internet Explorer, Mozilla, Firefox, Netscape 6+, AOL 9.0, Konqueror and others. The following browsers do not support the `onUnload` event and therefore will not be reported in Form Analysis reports.

- Netscape 4x and below
- Opera
- Safari.

## 2 Configuration Variables

The following sections define each of the variables used by Form Analysis and outlines possible values for each.

### 2.1.1 formList

The formList variable is a comma-separated list of form names. These forms will be tracked or excluded from tracking based on the value of trackFormList. In other words, the formList variable works together with the trackFormList variable. If trackFormList is true, then form analysis will track all forms listed in the formList variable. If trackFormList is false, then the form analysis data from all forms except those listed in the formList variable will populate the SiteCatalyst reports, as shown below.

```
s.formList="contactForm,Registration"
```

```
s.formList=""
```



**NOTE:** Form names must match the exact value of the name parameter used in the HTML form element.



**WARNING:** Make sure none of your forms contain any elements called “name.” If a form has a form element called “name,” that form will not be tracked with this plugin.

### 2.1.2 trackFormList

trackFormList should be set to “true” or “false.” If true, all forms in formList will be tracked with the Form Analysis plugin, as shown below. If false, all forms in the formList variable will be excluded from Form Analysis.

```
s.trackFormList=true
```

```
s.trackFormList=false
```

For example, if you populate the formList variable as formList=“form1,form2,form3,” SiteCatalyst only displays form analysis data from form1, form2, and form3. However, if you give a value of “false” to the trackFormList variable, SiteCatalyst will report form analysis data for all forms in your site except for form1, form2, and form3.

### 2.1.3 useCommerce

If you want to display your form analysis data in a Traffic Custom Insight Report, useCommerce should be set to “false.” If you want Form Analysis to be shown in a commerce report, useCommerce should be set to “true,” as shown below. For sample Traffic and Commerce reports, refer to the *Form Analysis Plugin Overview* whitepaper.

```
s.useCommerce=true
```

```
s.useCommerce=false
```

If useCommerce is true, varUsed should contain the name of an eVar (e.g. eVar6) and eventList should contain one or more events (e.g. event4,event5). On the other hand, if useCommerce is false, varUsed should be a prop variable (e.g. prop6) and eventList will be ignored.



**NOTE:** If you use the useCommerce variable, eventList should contain at least one event and varUsed should contain an eVar.

### 2.1.4 eventList

This variable should only be used if useCommerce is true (you want to display data in a Commerce Report). eventList is a comma separated list of up to three events. The first in the list will be used to count abandons, the second

successful submissions, and the third errors. You may leave out either or both of the last two events, as shown in the example below.

```
s.eventList="event3,event4,event5" // abandon, success, error
s.eventList="event3,,event5"       // abandon and error
s.eventList="event3,event4"        // abandon and success
s.eventList="event3"               // abandon only
```



**NOTE:** If you use the eventList variable, useCommerce should be set to “true” and varUsed should be an eVar. Additionally, you are only given a certain number of events in your Omniture contract. However, you can purchase more, if necessary.

### 2.1.5 varUsed

The varUsed variable is to be used to store form analysis data. For traffic reports, use a prop. For commerce, use an eVar.



**NOTE:** If useCommerce is set to “true,” varUsed should be an eVar, as shown below.

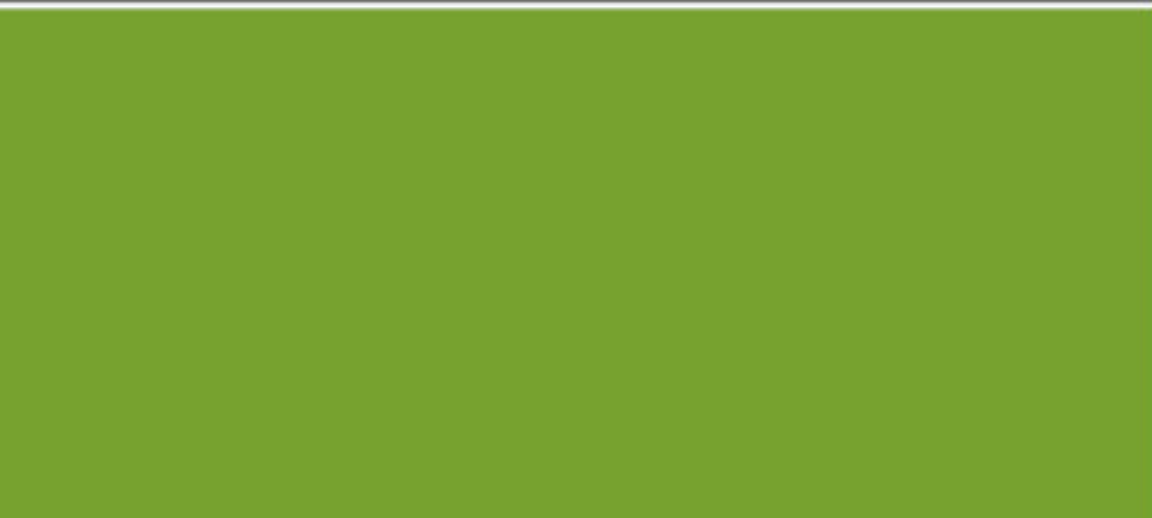
```
s.varUsed="eVar6" // useCommerce is true
s.varUsed="prop6" // useCommerce is false
```

### 2.1.6 trackPageName

If trackPageName is set to “true,” the page name that contains the form being tracked is added to form analysis data in SiteCatalyst reports. A value of “false” removes the page name from the form analysis data that is displayed in SiteCatalyst reports, as shown below.

```
s.trackPageName=true
s.trackPageName=false
```

The Traffic Custom Insight Reports display form analysis data as Page Name:Form Name:Form Outcome (Abandon, Success, or Failure):Form Field. The Commerce Custom Insight Reports display form analysis data as Page Name:Form Name: Field Name.



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

OMNITURE™  
— — —