# Tracking Complex Web Page Interactions Using dcsMultiTrack

The WebTrends dcsMultiTrack function lets you log virtual page views to WebTrends SmartSource Data Collector (SDC) servers for nearly all types of events, including events within a Flash application as well as interactions triggering Ajax activity. Although the SDC tag captures page views automatically, not all relevant data is captured by the tag. dcsMultiTrack is a way to program a Web site page so that data captured using the dcsMultiTrack functions and parameters is recorded. Data that dcsMultiTrack can capture includes:

- Flash events such as start, finish, progress, and clicks within a Flash application
- Ajax interactions
- Downloads of files that can not be tagged, such as pdf's
- Form completion
- Any event that launches JavaScript

This document describes how to implement dcsMultiTrack API calls to track data that is not available through the advanced JavaScript tags. Advanced JavaScript tags are the primary source for tracking events, while dcsMultiTrack tags are used to implement tracking for events that are not available using the advanced tags. For information on using Advanced JavaScript tags, or for information about tracking Flash events, see "Customizing JavaScript Tags" or "Tracking Macromedia Flash" in the *WebTrends System Administration Guide*.

## How dcsMultiTrack Works

When a visitor performs an action that activates dcsMultiTrack, the function relies on the information saved when it was initially collected when the page was loaded. The information sent by the tag becomes a single log file line in the SDC log. Unmodified, this would duplicate a page view that was already recorded when the HTML page was initially loaded. However, dcsMultiTrack can overwrite or supplement any data you choose from the hosting page, enabling you to specify exactly what information is recorded in the hit being reported by dcsMultiTrack.

Additionally, when Web 2.0 technologies are implemented on a site that does not result in new HTML pages being loaded, such as Flash, Ajax, and other interactive elements, these interactions are not reported with the standard (non-dcsMultiTrack) implementation. When you implement dcsMultiTrack on a target page, data is collected each time an interaction occurs. User interactions, such as selecting different tabs or other options to change the content displayed, are reported in the same way as loading a new HTML page. Using dcsMultiTrack on sites that do not use Web 2.0 technologies provides a detailed view of how customers interact with the site when analyzing traffic.

## Preparing for dcsMultiTrack

To implement dcsMultiTrack, you must include the dcsMultiTrack function in your WebTrends JavaScript tag. Because dcsMultiTrack relies on the standard WebTrends JavaScript tag to operate successfully, you must still deploy the standard WebTrends JavaScript tag to a page. The standard JavaScript tag on the HTML pages that include Flash, Ajax, or other technology you want to track using dcsMultiTrack.

If you use WebTrends Tag Builder to generate your tag, the JavaScript code includes the dcsMultiTrack definition. If you currently use an older version of WebTrends JavaScript code, you may need to modify the tag as described in the following sections. You can ensure that the version of the JavaScript includes the function definition for dcsMultiTrack by verifying that the line "`function dcsMultiTrack(){`" is present in the tag.

### Adding the dcsMultiTrack Function to an Existing JavaScript Tag

If you are using an older version of WebTrends the JavaScript tag, and you do not want to deploy a new tag created with WebTrends Tag Builder, you can update your existing tag to enable dcsMultiTrack.

**To add dcsMultiTrack to an existing tag:**

1. Locate the dcsMultiTrack function.
   a. If you use the SmartSource Data Collector (SDC) with WebTrends software, this function is located in the `multitrack.js` file on the SDC server in the following directory:

      `<`*`SDC Installation Directory`*`>\util\javascript\`

   b. If you use WebTrends On Demand, or if you cannot locate the `multitrack.js` file, contact WebTrends Support to obtain a copy of the dcsMultiTrack function definition.
2. Open the `multitrack.js` file and select all the code in the file. Copy this text to the clipboard.
3. Paste this code into your WebTrends Javascript tag immediately after the comment `Add customizations here` as shown in the following WebTrends JavaScript tag excerpt:

```
// Add customizations here

function dcsVar(){
var dCurrent=new Date();
WT.tz=(dCurrent.getTimezoneOffset()/60*-1)||"0";
WT.bh=dCurrent.getHours()||"0";
```

## Using dcsMultiTrack

Any number of parameters can be passed to dcsMultiTrack providing they are passed to the function in pairs, and the parameters used are JavaScript strings enclosed in single or double quotes. The first element of each pair is the name of the WebTrends query parameter you want to set, while the second element of each pair is the value you want the WebTrends query parameter to contain. To be recognized by WebTrends query parameters, the query parameter names must begin with `WT.`, `DCS.`, or `DCSext.`. The values that you specify replace the values that would normally be gathered by the WebTrends JavaScript tag. To avoid duplication of page views, you can report the dcsMultiTrack event with a different URL and page title than the initial page view for the HTML page. This requires using the `DCS.dcsuri` and `WT.ti` parameters with the dcsMultiTrack function. The following example illustrates this implementation:

```
dcsMultiTrack('DCS.dcsuri','/folder/movie.swf','WT.ti',
   'Name%20of%20the%20Movie');
```

In the above example, `%20` was used to replace spaces in the page title. Because spaces can have a special meaning in many scripting languages, the best practice is to use URL encoding to represent any non-ASCII character, especially spaces, quotes, and parentheses.

Other parameters may be passed to dcsMultiTrack as well. Any WebTrends query parameter that you can set with a META tag can also be set with tdcsMultiTrack. If the page contains META tags or parameters that specify WebTrends query parameters that data is sent to the SmartSource Data Collector along with values for the initial HTML page view. For a complete list of WebTrends query parameters, see the *WebTrends Administrators User's Guide*. For a list of JavaScript event handlers and their corresponding attributes, refer to a JavaScript Web site such as http://www.w3schools.com/jsref/jsref_events.asp.

In the following example, values from the initial HTML page view are sent to the WebTrends data collectors. If the page contains META tags or parameters that specify WebTrends query parameters, that data is also sent.

To invoke the dcsMultiTrack function to report a virtual page view to the WebTrends data collector, use the following syntax:

```
dcsMultiTrack('parameter1', 'value1', 'parameter2', 'value2');
```

## Persistent and Inherited Query Parameters

If you do not specify values for each parameter-value pair used with the dcsMultiTrack call, WebTrends uses values from the most recent dcsMultitrack call. These inherited parameters are specific to dcsMultiTrack while JavaScript query parameters are persistent.

If you want to exclude data from specific query parameter tags, you can do so by specifying blank values in the dcsMultiTrack statement. For example, to remove the `WT.mc_id` campaign tracking parameter, modify the dcsMultiTrack tag syntax as follows:

```
dcsMultiTrack('DCS.dcsuri','/folder/movie.swf','WT.ti',
   'Name%20of%20the%20Movie', 'WT.mc_id', '');
```

## Tracking Off-Site Links and Downloads

If you want to track a specific HTML link, whether to another site, a downloadable file, or any other linked target, you can implement the dcsMultiTrack function to collect this data whenever the link is clicked, using the JavaScript "onclick" event. The following example illustrates how to track clicks on a link that leads to a different site by activating the dcsMultiTrack function:

```
<a href="http://www.somedomain.com/folder/page_of_interest.html"
onclick="dcsMultiTrack('DCS.dcssip', 'www.somedomain.com',
'DCS.dcsuri', '/folder/page_of_interest.html',
'WT.ti', 'Page%20of%20Interest');">
Page of Interest</a>
```

In this example, the URL `http://www.somedomain.com/folder/page_of_interest.html` is included in the Pages report, as well as other reports. Even though the target page is not tagged with the WebTrends JavaScript code, the dcsMultiTrack function call simulates that page view by setting the domain, URL stem, and page title.

The same implementation can be used to track click frequency for link to download a file, such as a PDF file, Word document, sound file, or any other resource links included on the HTML page. The following example illustrates how the dcsMultiTrack function is used to report clicks to a PDF document:

```
<a href="/downloads/sales_brochure.pdf"
onclick="dcsMultiTrack('DCS.dcsuri', '/downloads/sales_brochure.pdf',
'WT.ti', 'Sales%20Brochure');">
Download a Sales Brochure</a>
```

Using this code would return reports showing a URL at the same domain as the HTML page, with the file path `/downloads/sales_brochure.pdf`. If you use the software version of WebTrends Analytics with the default configuration, this appears in the Downloaded Files report, but not the Pages report. WebTrends Analytics On Demand includes this information in both reports.

It is possible to use general event handlers to track all links or all downloads for a page or site. However, this level of implementation is beyond the scope of this document. Before implementing dcsMultiTrack to track a specific link or download, check with your WebTrends administrator to verify if such an event handler is implemented on your site.

## Tracking Form Elements

The dcsMultiTrack function can be activated by any JavaScript event. For example, if you want to track how a visitor progresses through a particular form so that you can configure a Scenario Analysis report to track where visitors drop out of the process, you can use the dcsMultiTrack function to track each form element separately, sending a virtual page view whenever a visitor fills out a form element.

The following example illustrates how to set up dcsMultiTrack to track which form elements the visitor completes. Note that each form element to be tracked is tagged with `DCS.dcsuri` and specifies the URL page where the tagged element `.htm` file resides.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Sign Up Here!</title>
<meta name="WT.si_n" content="RegistrationForm">
<meta name="WT.si_x" content="1">
</head>
<body>
<form action="http://www.tipjar.com/cgi-bin/test method=post>
<p>
<input name="First Name" type="text" value="" size="40" maxlength="100"
onchange="dcsMultiTrack('DCS.dcsuri', '/FirstName.htm',
'WT.ti', 'First%20Name%20Field', 'WT.si_n', 'RegistrationForm',
'WT.si_x', '2');">First Name</p>
<p>
<input name="Last Name" type="text" value="" size="40" maxlength="100"
onchange="dcsMultiTrack('DCS.dcsuri', '/LastName.htm',
'WT.ti', 'Last%20Name%20Field', 'WT.si_n', 'RegistrationForm',
'WT.si_x', '3');">Last Name</p>
<p>
<input name="Email Address" type="text" value="" size="40" maxlength="100"
onchange="dcsMultiTrack('DCS.dcsuri', '/Email.htm',
'WT.ti', 'Email%20Field', 'WT.si_n', 'RegistrationForm',
'WT.si_x', '4');">Email Address</p>
<p>
<input type=submit value="submit"></p>
</form>
</body>
</html>
```

When a visitor fills in each field of the form, the Pages report shows the URL for the HTML page itself, as well as URLs for `/FirstName.htm`, `/LastName.htm`, and `/Email.htm`. The Registration Form Scenario Analysis report is also be populated with all four steps reflected in this example. If the "Thank you" page has the appropriate META tags for the fifth step of the scenario, you can obtain a granular view of how visitors are interacting with your form.

## Using Flash with getURL

To log a virtual page view whenever a visitor plays a particular Flash movie embedded in your page, you can activate the dcsMultiTrack function from within the ActionScript for the Flash object using `getURL`. For example, you can insert the following code in the first frame of the Flash movie:

```
getURL("JavaScript:dcsMultiTrack('DCS.dcsuri', '/Support/movie.html/',
   'WT.ti', 'Support%20and%20Services%20Movie')");
```

Note that in ActionScript, this code must appear on a single line with no line break. This code does not need to be on its own layer.

You can track other Flash events in addition to plays. To capture data when a button is clicked within a Flash application, use the dcsMultiTrack function from within the event handler for that button. For example, if you have a button that plays an additional piece of video, your event handler syntax may look similar to the following example:

```
on (release) {
   getURL("JavaScript:dcsMultiTrack('DCS.dcsuri', '/Support/ws.html/'
      'WT.ti', 'Training%20Info%20Button', 'DCSext.WT.type1', 'Flash')");
gotoAndPlay(2315);
}
```

Using this implementation, any visitor interaction that triggers a Flash event can be reported to WebTrends as a virtual page view.

## Using Flash with ExternalInterface

Since the `getURL` function was removed from the ActionScript language in version 3, calling the dcsMultiTrack JavaScript function is slightly more complex in the newer version of Flash. Adobe recommends using the `ExternalInterface` object to call external JavaScript functions. In this case, your ActionScript syntax may look similar to the following example:

```
if (ExternalInterface.available) {
   try {
      ExternalInterface.call("dcsMultiTrack", "DCS.dcsuri",
           "/flash/support_tab.html", "WT.ti", "Support%20Tab");
   } catch (error:Error) {
      // error handling here
   } catch (error:SecurityError) {
      // error handling here
   }
}
```

As in the `getURL` examples in the previous section, this code can be embedded either in the first frame of a movie, or in an event handler.

This usage of the `ExternalInterface` object assumes that the HTML page (with the WebTrends JavaScript code) is the container for your Flash application. Implementations that do not use HTML as the container for Flash applications are more complex and may require the engagement of a WebTrends Professional Services engineer to assist you with your implementation.

## Tracking Ajax Interactions

The term Ajax is used to refer to a variety of specific combinations of technologies. Most of these technologies focus on the use of JavaScript to directly modify the Web page using a Document Object Model (DOM), based on customer interaction and data fetched from the server. Ajax can be used to provide rich applications within the Web browser ranging from calendar applications to mapping tools, and can also be applied in a simpler form for minor changes like updating a drop-down menu based on visitor input.

Because Ajax typically uses JavaScript events, implementing dcsMultiTrack within an Ajax application is straightforward. If an Ajax event is triggered by a customer interaction that you want to report on, the same Ajax event that updates the page in response to that interaction can also call the dcsMultiTrack function.

The following example illustrates a simple Ajax control implementation where the visitor's selection in a drop-down menu changes the options available in a second drop-down menu. The dcsMultiTrack function tracks each time a user chooses a different option in the first drop-down box.

```
<select name="drop-down1" onChange="UpdateDropDown(this.value);">
<-- options -->
</select>
<select name="drop-down2">
</select>
...
<script type="text/javascript">
function UpdateDrownDown(OptChoice) {
   var drop2 = parelmts["drop-down2"];
// fetch data for the second dropdown
parelmts.length = xmlreply.length;
   for (o=1; o < xmlreply.length; o++) {
      drop2[o].text = xmlreply[o];
   }
   dcsMultiTrack('DCS.dcsuri', '/dropdown.html', 'WT.ti', 'Drop%20Down',
      'DCSext.Choice', OptChoice);
}
```

In this example, each use of the first drop-down menu is reported as a virtual page view at the URL `http://www.yourdomain.com/dropdown.html`. The user-defined Choice parameter is also populated with the selection the visitor made in the drop-down menu.

Due to the wide variety of Ajax implementations, attempting to provide examples for every possible scenario is beyond the scope of this document. However, the examples provided are applicable to nearly any Ajax application. In most cases, the modification to the DOM is encapsulated within a function. Therefore, activating the dcsMultiTrack function within your own function is a discreet method to implement WebTrends tracking within your Ajax application.

## Tracking Using Document Object Model (DOM)

For complex Web sites that enforce rigorous separation of logic and presentation, you can use the Document Object Model (DOM) method to implement the dcsMultiTrack function. Implement dcsMultiTrack on a DOM element ID or other criteria such as anchor tags that lead to PDF documents. To illustrate basic implementation using DOM with dcsMultiTrack, two examples are included in the following sections. The first example illustrates activating dcsMultiTrack for a particular DOM element ID. The second example illustrates activating dcsMultiTrack for a specific DOM tag name.

---

**Note:**
Using DOM methods with dcsMultiTrack is a complex implementation that requires an understanding of DOM methodology and programming skills that are beyond the scope of this document. The following examples are for illustration purposes only–**do not copy and paste the code examples to your Web page** as they may not work as expected.

For information on DOM methods, code samples, and validating syntax, see http://w3schools.com/htmldom/default.asp or other applicable Document Object Model Web sites.

---

### Example: Specifying DOM Element ID with dcsMultiTrack

The following example shows how to activate dcsMultiTrack for a particular DOM element ID:

```
<script type="text/javascript">
var x=getElementByID("page_of_interest");
x.onclick="dcsMultiTrack('DCS.dcssip', 'www.domain_name',
 'DCS.dcsuri', '/folder/page_of_interest.html',
 'WT.ti', 'Page%20of%20Interest');"
</script>
...
<a id="page_of_interest" href="http://www.somedomain.com/folder/
page_of_interest.html">Page of Interest</a>
```

### Example: Specifying DOM Tag Name with dcsMultiTrack

The following example shows how to activate dcsMultiTrack for a particular DOM tag name:

```
<a id="page1" href="http://www.somedomain.com/folder/page_of_interest.pdf">Page of
Interest 1</a>

<script type="text/javascript">
var x=document.body.getElementsByTagName("a");
for (i=0;i<x.length;i++) {
   if (x[i].href.toUpperCase().indexOf("PDF") >= 0) {
      var url_split=x[i].href.match(/(\w+):\/\/([\w.]+)(\/\S*)/);
      var wt_domain=url_split[2];
      var wt_uri=url_split[3];
      var wt_title=x[i].innerHTML
      x[i].onclick=function(){
           dcsMultiTrack('dcssip', wt_domain, 'dcsuri', wt_uri, 'WT.ti', wt_title);
      };
   }
}
</script>
```