

This document outlines the two pieces of code required to integrate with the Adlucent Order Tracking system. Adlucent's tracking is based on two redundant methods of tracking clicks (to handle the multiple tracking methods employed by our Advertising Platforms,) in conjunction with a piece of code on your Order Confirmation page to report order details to Adlucent.

The first method of tracking clicks is a cookie-based method, where users are redirected through our servers and a cookie is placed on their machine with the necessary information for Adlucent to attribute the order. The code on the order confirmation page then makes a call to Adlucent's servers, allowing that tracking cookie to be read and the order information sent to be attributed based on the cookie. No code is required on the client side to support this method.

The second method of tracking clicks is based on a click ID assigned to each click event and handles tracking events in parallel using the browser's `sendBeacon()` method. This tracking method also allows us to record orders in the instance our tracking cookie from the above method is removed or blocked as third party. In this scenario, our tracking server is pinged with tracking information and a click ID, while the user is simultaneously sent to the landing page with the same ID appended.

Our second method of tracking requires a small piece of code be included on all pages of your site to watch for this incoming click ID value and to store it in a first party cookie when present. The code on the order confirmation page will then retrieve and report this cookie value when populated, allowing us to attribute the order based on the tracking information recorded against the associated click ID. The code included on the following page is required to support this method.

The long-term goal in supporting both methods of tracking is to ultimately migrate completely away from a cookie-based tracking method, and solely rely on tracking clicks via Click IDs in the future. This move is being made due to emerging industry trends (from major players such as Apple and Google) away from Cookie based tracking methods (i.e. Apple's Intelligent Tracking Prevention, Google Adwords' Parallel Tracking) in response to the growing amount of consumer concern and legislation regarding data collection and the privacy of User's information.

[space intentionally blank]

## SETTING THE ADLUCENT CLICK ID COOKIE

The first requirement of Adlucent’s tracking is the addition of code for setting a first party tracking cookie to all pages of your site. This code will watch for the presence of certain parameters and store them in a cookie under the name “\_adlclkid” when found. This code also watches for the presence of an “adlpxid” parameter and will report that value to Adlucent when present.

The parameters searched for are those used by the engines to provide click ID’s and allow attribution back to the clicked element. An Adlucent generated value will be used in the absence of engine provide ones. The “adlpxid” will be appended to product URLs under certain click scenarios from Google Local Inventory Ads and will report the required tracking information to Adlucent to properly attribute these clicks.

The specific code used to watch for these values, set the required cookie, and handle the request to our servers will be determined by your development team, though a small JavaScript snippet that accomplishes this can be found below and should be used if possible. If using a Tag Manager, this code would be installed as a Custom HTML tag and set to fire on all pages.

The only change required for this code is the value of “adlRetailer” and “adlDays”. “adlRetailer” will be set to a static value provided by Adlucent and will match the value used in the order confirmation pixel. “adlDays” controls the number of days before cookie expiration. This value is typically determined contractually and if a different value is required for your implementation, you will be notified by Adlucent.

```
<script type="text/javascript">
(function () {
  if (typeof window.URLSearchParams !== "function") {return;}
  var adlDays = 30;
  var adlRetailer = "<your retailer value>";
  var adlclkID, adlPxID, adlURL;
  var params = ["gclid", "gbraid", "wbraid", "adlclid"];
  var qs = new URLSearchParams(window.location.search);

  for (var i=0; i < params.length; i++ ) {
    if (qs.get(params[i])) {
      adlclkID = qs.get(params[i]); break;
    }
  }

  if (adlclkID) {
    document.cookie = "_adlclkID="+adlclkID+"; max-age="+ (60*60*24*adlDays).toString()+"; path="/;
  }
  if (qs.get("adlpxid")) {
    adlPxID = qs.get("adlpxid");
    adlURL =
document.location.protocol+"//tracking.deepsearch.adlucent.com/adlucent/Redirector/beacon?retailer="+adlRetailer+"&clickid="+adlclkID+"&pixelid="+adlPxID;
    var pix = new Image(1, 1);
    pix.src = adlURL;
  }
})();
</script>
```

## REPORTING ORDERS TO ADLUCENT

The second requirement for Adlucent’s tracking is the inclusion of code on your order confirmation page to report the order and details to Adlucent (the Adlucent Tracking Pixel.) This code snippet should be included on your confirmation page for **ALL** transactions that take place on your site. Examples of this code are show below along with instructions on what order data should be inserted into the snippet. In the examples, items indicated in bold and by angle brackets (e.g. **<your sku>**) will be replaced with the appropriate conversion data by your site, the remaining code should remain unchanged.

To integrate the Adlucent pixel, the JavaScript snippet will need to be inserted into your order confirmation page in between the <body> tags. The specific location it is placed depends upon the availability of the order information on your page. The snippet should be placed in a location where this data is accessible and complete, likely just above the closing </body> tag. An example snippet is shown below including all variables supported by the Adlucent pixel. The specific variables required for your implementation will be provided by Adlucent during the integration process.

```

<script type="text/javascript">
function getCookieValue(a) {
    var b = document.cookie.match('(^\s*|)\s*' + a + '\s*=\s*(\s*([^;]+)\s*)');
    return b ? b.pop() : '';
}
var orderData = [];
var retailer = "<your retailer value>";
var clickId = getCookieValue("_adlClkID");
var orderKey = "<your order key>";
var customerType = "<customerType>";
var customerID = "<customerID>";

orderData.push(['<your sku>', '<item qty>', '<unit price>']);

(function() {
var adl = document.createElement("script"); adl.type = "text/javascript"; adl.async = true;
adl.src = document.location.protocol + "//tracking.deepsearch.adlucent.com/adl.js";
var s = document.getElementsByTagName("script")[0]; s.parentNode.insertBefore(adl, s);
})();
</script>

```

The variable names provided CANNOT be modified and will need to be input exactly as provided to be collected. The cookie reading function provided and used to populate clickId can be excluded if another method of populating cookie values is available, such as a sitewide function included in a library or a dataLayer variable in a Tag Manager set based on a cookie value.

The anonymous function at the end handles reporting the order to Adlucent and should not be modified.

Additional detail about each piece of the pixel is provided on the following pages.

As mentioned, various fields (denoted in bold and in brackets) will need to be replaced by your system with the appropriate order data. The most basic of these replacements are for the “retailer” and “orderKey” variables, and for the SKU/Quantity/Price information in the “orderData” array. These can be seen highlighted below:

```

<script type="text/javascript">
function getCookieValue(a) {
    var b = document.cookie.match('(^\s*\s*' + a + '\s*\s*=\s*\s*([\^;]+)');
    return b ? b.pop() : '';
}
var orderData = [];
var retailer = "<your retailer value>";
var clickId = getCookieValue("_adlClkID");
var orderKey = "<your order key>";
var customerType = "<customerType>";
var customerID = "<customerID>";
orderData.push(['<your sku>', '<item qty>', '<unit price>']);
(function() {
var adl = document.createElement("script"); adl.type = "text/javascript"; adl.async = true;
adl.src = document.location.protocol + "//tracking.deepsearch.adlucent.com/adl.js";
var s = document.getElementsByTagName("script")[0]; s.parentNode.insertBefore(adl, s);
})();
</script>

```

- The **<retailer>** variable represents an ID that will be provided to you by Adlucent. It will be static and should be input exactly as provided. If this retailer value is incorrect, Adlucent will be unable to process your orders properly.
- The **<orderKey>** variable should be filled in with the ID used to identify the order in your system. This will be recorded as the Order ID in Adlucent’s system and represents the ID that will be shown in Order reporting.
  - This variable could be filled in with a string value (e.g. orderKey = "FF345678123"; ) or could be derived dynamically depending on the specifics of your confirmation page. (e.g. orderKey = document.getElementById("orderKey").innerHTML; )
  - This value is used to deduplicate received pixels to prevent confirmation page reloads and similar events from reporting multiple orders. As such it is expected that each generated orderID will be unique.
- A triplet of **<SKU>**, **<Qty>**, and **<Price>** values should be inserted into the **<orderData>** array for each line item purchased in the order. That is, for multiple SKU orders, an orderData.push(); will be included for each SKU in the order.
  - The SKU Value reported in the Adlucent pixel should match the ID value provided in the Adlucent product feed, such that product information can be pulled into order reports by aligning these values.
  - The Qty value should be formatted as an integer and the Price as a decimal with 2 places.
  - Be sure to use an opening bracket “[” and a closing bracket “]” surrounding the each set of values being inserted into the array, as you are inserted an array of strings.
  - For example, a two item order would contain the following lines:

```

orderData.push([123, '3', '24.99']);
orderData.push([456, '1', '39.99']);

```

- Discounts affecting the order total or item price can be handled via multiple methods:
  - Discounts directly affecting the cost of an item should be reflected in the unit price reported. For example, an item selling for \$24.99 normally but with a \$10 off coupon code should be reported with a price of 14.99.
  - Discounts affecting the cost of the order can be reported as an additional line item in the orderData array. This is done by inserting a SKU of 'DISCOUNT', a quantity of 1, and price of the NEGATIVE discount amount into the orderData array, e.g.

```
orderData.push(['DISCOUNT', '1', '-10.00']);
```

- If multiple order level discounts are applied, multiple instances of the 'DISCOUNT' SKU can be inserted, as we will aggregate them appropriately on this end. For example, the below discounts would be recorded in our system as 2 discounts for 25.99 off the value of the order.

```
orderData.push(['DISCOUNT', '1', '-5.99']);  
orderData.push(['DISCOUNT', '1', '-20.00']);
```

[space intentionally blank]

In addition to these basic order attributes, the Adlucent pixel can capture and report metrics around Customer Acquisition, cross device tracking, and the segmentation of New/Returning/Guest transactions. This is enabled by providing the two variables highlighted below, “customerType” and “customerID”, with the appropriate information regarding the customer who has converted.

*These variables are considered optional in the Adlucent pixel and if this level of information is not available at checkout these variables can be set to an empty string value or left out of the pixel entirely. Additional detail on these variables can be found below:*

```

<script type="text/javascript">
function getCookieValue(a) {
  var b = document.cookie.match('(^\|;)\s*' + a + '\s*=\s*([^\|;]+)');
  return b ? b.pop() : '';
}
var orderData = [];
var retailer = "<your retailer value>";
var clickId = getCookieValue("_adlClkID");
var orderKey = "<your order key>";
var customerType = "<customerType>";
var customerID = "<customerID>";
orderData.push(['<your sku>', '<item qty>', '<unit price>']);
(function() {
var adl = document.createElement("script"); adl.type = "text/javascript"; adl.async = true;
adl.src = document.location.protocol + "//tracking.deepsearch.adlucent.com/adl.js";
var s = document.getElementsByTagName("script")[0]; s.parentNode.insertBefore(adl, s);
})();
</script>

```

- The **<customerType>** variable accepts any string value, though best practice is to provide either “New”, “Repeat”, or “Guest.” The definition of a “New” vs. “Repeat” customer will be different for each retailer and these definitions will need to be determined internally prior to reporting these values to Adlucent, as we allow for rolling up and segmenting of metrics on the values you provide. If there are additional customer types, or a classification scheme that is more suited for your business, those values may be included instead of the traditional types.
- The **<customerID>** variable accepts any string value, though some sort of internal customer ID is typically provided. This value will need to be accessible and constant during future checkouts, so that it is the same for subsequent conversions by that customer, and so that they may be looked up in your internal system if any reconciliation is needed. This variable should **NOT** contain any personally identifying information.

[space intentionally blank]

## Testing, Verification & Assistance

When implementing the Adlucent pixel or going through this document, please do not hesitate to reach out to your Adlucent contacts for any assistance or clarification. They will help you when possible or place you in touch with a member of Adlucent's technical staff for assistance if necessary.

Once implemented, we're also available to help QA the pixel if access can be given to a development or staging site. If this is desired, just let your Adlucent contacts know and they'll place you in touch with a member of Adlucent's technical staff for assistance.

If you or your developer has fired pixel strings in testing and you'd like to verify the data received by Adlucent against the data expected, just reach out to your team with a listing of the Order ID's and order information expected to be passed for each, and we can verify them against the data in our system and provide any feedback needed.

Regardless of Adlucent's involvement in the QA of the pixel, test orders will need to be placed following the implementation to verify reporting is coming through as expected from live clicks and conversions. If there is a Dummy Credit Card or Test Account that is used by your site to place orders that will not be charged or fulfilled, Adlucent will need access to this to place test orders without the need to contact customer service and cancel the order. Otherwise, the team will work with you to determine the proper method of placing test orders and having them cancelled, typically using Company or Personal Cards to place the orders.

Example orders demonstrating the fully formed Adlucent snippet can be found on the following pages that may be helpful in setting up your pixel. These examples exclude the cookie setting function for brevity and assume it is defined outside of the pixel.

[space intentionally blank]

Order Example				
<i>RetailerID:</i> RetailerX	<i>Customer Type:</i> New	<i>SKU:</i> z12	<i>SKU:</i> 5559	<i>SKU:</i> 8975
<i>OrderID:</i> g78989	<i>Customer ID:</i> 101789	<i>Qty:</i> 2	<i>Quantity:</i> 4	<i>Quantity:</i> 1
		<i>Unit Price:</i> 2.99	<i>Unit Price:</i> 25.99	<i>Unit Price:</i> 109.99
<pre> &lt;script type="text/javascript"&gt; function getCookieValue(a) {     var b = document.cookie.match('(^\s*)\s*' + a + '\s*=\s*([^\s;]+)'); return b ? b.pop() : ''; } var orderData = []; var retailer = "RetailerX"; var clickId = getCookieValue("_adlClkID"); var orderKey = "g78989"; var customerType = "New"; var customerID = "101789"; orderData.push(['z12', '2', '2.99']); orderData.push(['5559', '4', '25.99']); orderData.push(['8975', '1', '109.99']); (function() { var adl = document.createElement("script"); adl.type = "text/javascript"; adl.async = true; adl.src = document.location.protocol + "//tracking.deepsearch.adlucent.com/adl.js"; var s = document.getElementsByTagName("script")[0]; s.parentNode.insertBefore(adl, s); })(); &lt;/script&gt; </pre>				

Order Example – Order with Discount			
<i>RetailerID:</i> RetailerX	<i>Customer Type/ID:</i> not available.	<i>SKU:</i> 17559751	<i>SKU:</i> 8985218
<i>OrderID:</i> P000013939		<i>Qty:</i> 2	<i>Quantity:</i> 10
<i>Discount:</i> \$10 Off.		<i>Unit Price:</i> 14.75	<i>Unit Price:</i> 1.99
<pre> &lt;script type="text/javascript"&gt; function getCookieValue(a) {     var b = document.cookie.match('(^\s*)\s*' + a + '\s*=\s*([^\s;]+)'); return b ? b.pop() : ''; } var orderData = []; var retailer = "RetailerX"; var clickId = getCookieValue("_adlClkID"); var orderKey = " P000013939"; orderData.push(['17559751', '2', '14.75']); orderData.push(['8985218', '10', '1.99']); orderData.push(['DISCOUNT', '1', '-10.00']); (function() { var adl = document.createElement("script"); adl.type = "text/javascript"; adl.async = true; adl.src = document.location.protocol + "//tracking.deepsearch.adlucent.com/adl.js"; var s = document.getElementsByTagName("script")[0]; s.parentNode.insertBefore(adl, s); })(); &lt;/script&gt; </pre>			

**Order Example – Order with Multiple Discounts**

<i>RetailerID:</i> RetailerX	<i>CustomerID:</i> 7549	<i>Sku:</i> 685952	<i>Sku:</i> 2215	<i>Sku:</i> 793521
<i>OrderID:</i> P000013939	<i>CustomerType:</i> New	<i>Qty:</i> 4	<i>Quantity:</i> 1	<i>Quantity:</i> 1
<i>Discount:</i> \$15 Off \$50 orders, \$5 off from coupon code.		<i>Unit Price:</i> 4.99	<i>Unit Price:</i> 49.99	<i>Unit Price:</i> 9.99

```

<script type="text/javascript">
function getCookieValue(a) {
    var b = document.cookie.match('(^\|;)\s*' + a + '\s*=\s*([^\;]+)'); return b ? b.pop() : '';
}
var orderData = [];
var retailer = "RetailerX";
var clickId = getCookieValue("_adlClkID");
var orderKey = " P000013939";
var customerType = "New";
var customerID = "7549";
orderData.push(['685952', '4', '4.99']);
orderData.push(['2215', '1', '49.99']);
orderData.push(['793521', '1', '9.99']);
orderData.push(['DISCOUNT', '1', '-15.00']);
orderData.push(['DISCOUNT', '1', '-5.00']);
(function() {
var adl = document.createElement("script"); adl.type = "text/javascript"; adl.async = true;
adl.src = document.location.protocol + "//tracking.deepsearch.adlucent.com/adl.js";
var s = document.getElementsByTagName("script")[0]; s.parentNode.insertBefore(adl, s);
})();
</script>

```