

ADITION HTML5 clicktag

An HTML5 creative can have one or more than one click areas and corresponding landing destinations. Since, an HTML5 creative is build often in third party tools, ADITION follows the standard guidelines from IAB/OVK to map, track and report the clicks. One has to setup the clicks (click-areas) as per the guidelines. The clicks in an HTML5 creatives should be build using IAB/OVK variables rather than absolute values (hard-coded URLs). The actual URLs/landings should be defined in ADITION. Please follow following instructions to build and use a ClickTag in ADITION:

1. Build an HTML5 creative using standard OVK keywords/variables
2. If an HTML5 creative is using one click area and only the variable - "clicktag" is used, then fill your destination URL correspond to the field named - "ClickURL" in the creative form in ADITION.
3. If an HTML5 creative is using two or more than two variables, then fill the series of destination URLs corresponding to the variables in ADITION such as clicktag1, clicktag2, and so on

ADITION expects at least one clickURL per creative. The variable "clicktag" is mapped to "ClickURL" in ADITION and it is a mandatory field, should be filled with a destination clickURL. So, one must use at least one click function and must use this variable in an HTML5 creative. Other variables can be used, when an HTML5 creative has multiple click destinations. ADITION expects one to use following OVK standard keywords/variables:

- clicktag
- clicktag1
- clicktag2
- clicktag3
- clicktag4
- clicktag5
- clicktag6
- clicktag7
- clicktag8
- clicktag9

Instruction/specification to build a clicktag

The aim of this section is purely to help ADITION customers with information about using OVK/IAB keywords/variables to define click areas, while building an HTML creative. One can always choose to use the keywords/variables in other way/form technically possible.

For a simple hyperlink tags

```
<html>
<head>
  <script>
    function mytag()
    {
      window.open(clicktag,"_blank")
    }
  </script>
</head>
<body>
  <a href="#" onclick="mytag();" >
    <div id="container1" style="width: 300px; height: 250px"> </div>
  </a>
</body>
</html>
```

OR

```
<html>
<head></head>
<body>
  <a href="#"clicktag" id="clicktag" target="_blank">
    <div id="container1" style="width: 300px; height: 250px"> </div>
  </a>

  <script>
    document.getElementById('clicktag').setAttribute('href', clicktag);
  </script>
</body>
</html>
```

For hyperlinks via script such as javascript function

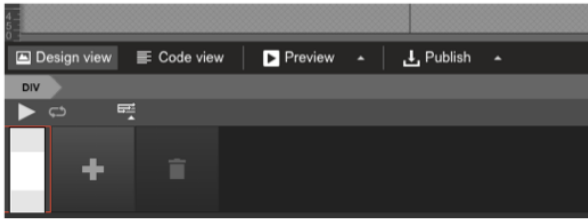
```
<script>
function xyz()
{
  window.open(clicktag,"_blank")
}
</script>
```

For multi-click tags

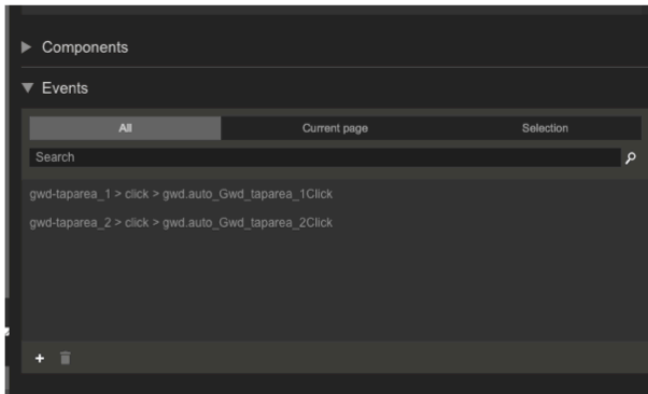
```
<html>
</head>
  <body>
    <a href="#clicktag" id="clicktag" target="_blank">
      <div id="container1" style="width: 300px; height: 250px"> </div>
    </a>
    <a href="#clicktag1" id="clicktag1" target="_blank">
      <div id="container2" style="width: 300px; height: 250px"> </div>
    </a>
    <script> document.getElementById('clicktag').setAttribute('href', clicktag);
    document.getElementById('clicktag1').setAttribute('href', clicktag1);
    </script>
  </body>
</html>
```

A few tools specific examples Google Web Designer

Google web designer allows to edit the code directly via **code view**. In the code, for each click button/area you have setup with GWD, you need to remove the destination url and replace it with a javascript variables. If you are using simple hyperlink tags, then please follow example mentioned earlier.



```
<script type="text/javascript" gwd-events="handlers">
  gwd.auto_Gwd_taparea_1Click = function(event) {
    // GWD Predefined Function
    window.open(clicktag, "_blank");
  };
  gwd.auto_Gwd_taparea_2Click = function(event) {
    // GWD Predefined Function
    window.open(clicktag1, "_blank");
  };
  gwd.auto_Gwd_taparea_3Click = function(event) {
    // GWD Predefined Function
    window.open(clicktag2, "_blank");
  };
</script>
```



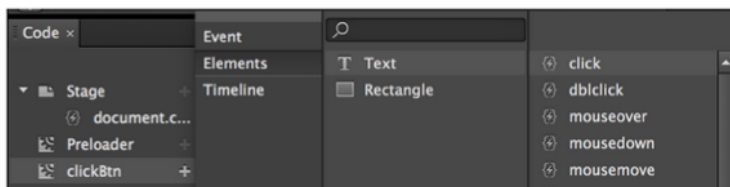
Example

Please download an example of ADITION clicktags directly from (<https://wiki.adition.com/index.php?title=Datei:HTML5-ClickTag.pdf>). A few information about the example:

It contains multiple elements and contains multiple click areas. There are also a few un-clickable areas. The banner uses 5 clicktags (clicktag, clicktag1, clicktag2, clicktag3 and clicktag4). clicktag covers the main upper part of the banner with the green image. clicktag1, clicktag2, clicktag3 and clicktag4 are the buttons respectively, in the lower part of the banner, from left to right.

Adobe Edge Animation CC

Once you've built your click button/area, you need to add a click event handler on the button. In this example, we have a symbol on the scene which contains a rectangle graphic and a text. The text field is the same size of the rectangle, and over the rectangle, so we'll use that as the trigger.



Here's a sample code you can add in this handler so you can retrieve the click url and open a new window to that url:



```
function(sym,e)
{
    window.open(clicktag,"_blank");
}
```

Alternatively, you can open a javascript file named - XXXXX_edgeActions.js and edit it directly as

```
Symbol.bindElementAction(compId,symbolName,"${_Stage}","click",function(sym,e){window.open(clicktag,"_blank");});
```

```
1
2 (function($,Edge,compId){var Composition=Edge.Composition,Symbol=Edge.Symbol;
3 //Edge symbol: 'stage'
4 (function(symbolName){Symbol.bindTriggerAction(compId,symbolName,"Default Timeline",9750,function(sym,e){sym.play("loop");});
5 //Edge binding end
6 Symbol.bindTriggerAction(compId,symbolName,"Default Timeline",9500,function(sym,e){});
7 //Edge binding end
8 Symbol.bindElementAction(compId,symbolName,"${_Stage}","click",function(sym,e){window.open(clicktag,"_blank");});
9 //Edge binding end
10 Symbol.bindElementAction(compId,symbolName,"${_HoverArea}","mouseover",function(sym,e){sym.$("IhoverArea").css('cursor','pointer');});
11 //Edge binding end
12 })(stage);
13 //Edge symbol end:'stage'
14
15 //=====
16
17 //Edge symbol: 'hoverArea'
18 (function(symbolName){})(hoverArea);
19 //Edge symbol end:'hoverArea'
20
21 //=====
22
23 //Edge symbol: 'IhoverArea'
24 (function(symbolName){Symbol.bindElementAction(compId,symbolName,"${_Rectangle3}","mouseover",function(sym,e){});
25 //Edge binding end
26 })(IhoverArea);
27 //Edge symbol end:'IhoverArea'
28 })(jQuery,AdobeEdge,"EDGE-171073236");
```