



FLASH CREATIVE

**SPECIFICATION FOR DISPLAY AND TRACKING
COMPLIANCE IN MOJO ADSERVER**



SUPPORT FOR AS2 AND AS3

This document contains technical specs and code for both ActionScript 2 and ActionScript 3. Procedures will be initially discussed using ActionScript 2 examples and then followed up with ActionScript 3 examples. To make an easy distinction between code samples, ActionScript 2 samples will be written in a box with a white background and ActionScript 3 samples will be written in a box with a yellow background.

```
ActionScript 2 sample
```

```
ActionScript 3 sample
```

Please keep in mind that the code samples contained in this document are recommendations only and are not guaranteed to work in all Flash movies under all circumstances. They have been tested and found to work for the creative implementations that we commonly see. However, as you develop your Flash creative you will want to be aware if there are any potential conflicts with other code. Also note that there are other possible approaches in ActionScript to achieve the same results we have outlined. We recommend the variations outlined in this document because we have tested them and can offer support if necessary. You may choose to use your own coding methodology to implement your clickTAGs. However, if you encounter problems with their functionality, Mediaplex may not be able to support or troubleshoot for you.

HELP

For any additional questions or problems, contact Mediaplex Customer Support at:

customersupport@mediaplex.com

1-866-417-1271 coding guidelines



CREATIVE PRODUCTION

HANDLING SINGLE CLICK-THRU DESTINATIONS USING THE CLICKTAG VARIABLE

For MOJO Adserver to track the click-thru from a Flash movie to a destination page, a variable containing the MOJO Adserver click-thru URL must be passed from the HTML into the movie. This variable is the clickTAG. Below are instructions for configuring flash creative to comply with Adobe Flash specifications and to work with MOJO Adserver.

The `getURL()` method uses the value of the clickTAG variable defined in the HTML `<object>` and `<embed>` tags, and redirects the user's web browser to the specified location. MOJO Adserver will track the click through the clickTAG. The second argument ("`_blank`") in the `getURL` method directs the click-thru into a new browser window or tab.

ACTIONSCRIPT 2

Option 1

Apply the following ActionScript ***directly to the button object***:

```
on (release) {
    if (_root.clickTAG.substr(0,4) == "http") {
        getURL(_root.clickTAG, "_blank");
    }
}
```

Option 2

Add the following ActionScript on the ***first frame of the main timeline***:

```
clickButton.onRelease = function() {
    if (_root.clickTAG.substr(0, 4) == "http") {
        getURL(_root.clickTAG, "_blank");
    }
}
```

Option 2 requires that the button instance be named (in this example: `clickButton`). When referencing the button in the ActionScript, the name must match exactly.



Security Note

The ActionScript is verifying that the clickTAG URL begins with "http". This is an important security measure. If you do not take this precaution, a malicious HTML page could source your SWF and pass a `_root.clickTAG` URL that begins with "javascript:" or another scripting pseudo-protocol. If your ActionScript code were to call `getURL` with a maliciously crafted JavaScript URL, it would be possible for the site serving the malicious HTML page to obtain the contents of your HTTP cookies or perform other unwanted actions on your site's behalf.

ACTIONSCRIPT 3

For ActionScript 3, all code is placed on the first frame of the main timeline. Button instances must be named and referenced exactly in the ActionScript. The code on the first frame contains two parts.

The first part of the code defines the clickTAG variable and creates a function to activate the clickTAG:

```
//define clickTAG variable
var mojoURL:String = LoaderInfo(this.root.loaderInfo).parameters.clickTAG;
var brType:String = ExternalInterface.call("function getBrowser(){
    return navigator.userAgent}") as String;
function mojoClick(event : MouseEvent, clickName:String) : void {
    if (mojoURL.substr(0,4)=="http") {
        if (brType.indexOf("Firefox") != -1 || brType.indexOf("MSIE") != -1) {
            ExternalInterface.call("window.open", mojoURL + clickName, "_blank");
        }
    }
    else {
        navigateToURL(new URLRequest(mojoURL + clickName), "_blank");
    }
    return;
}}
```

The second part associates the clickTAG with the button object.

For single click-thru creative, use the following code:

```
//call standard clickTAG
clickButton.addEventListener(MouseEvent.CLICK,
function(event:MouseEvent):void {
    mojoClick(event, "") },false, 0, false);
```

In the sample above, the button instance name is `clickButton`. You will substitute this with the actual instance name of your button object or name your button instance 'clickButton.'



MULTIPLE CLICK-THRU DESTINATIONS IN ONE FLASH MOVIE

Append a parameter=value pair to the end of the clickTAG. This parameter and its value are checked using rules (targeting directives) in MOJO Adserver and set to resolve to a specific destination URL. In the following example the parameter "ck" is given a value of "1". "ck" is commonly used as the parameter name but you can use any name you like.

ACTIONSCRIPT 2 EXAMPLE:

```
_root.clickTAG + "&ck=1"
```

Each additional click-thru destination thereafter should follow the same format:

```
_root.clickTAG + "&ck=2"  
_root.clickTAG + "&ck=3"
```

Sample ActionScript for multiple click-thrus

```
on (release) {  
    if (_root.clickTAG.substr(0,4) == "http") {  
        getURL(_root.clickTAG + "&ck=1", "_blank");  
    }  
}
```

MULTIPLE CREATIVE VARIATIONS WITH MULTIPLE CLICK-THRU DESTINATIONS

If you are using multiple creative variations in the same campaign and using multiple click-thru destinations within each creative variation, you will want to use more specific parameter names to segregate the different creative and destinations.

```
_root.clickTAG + "&banner_a_ck=1"  
_root.clickTAG + "&banner_a_ck=2"  
_root.clickTAG + "&banner_b_ck=1"  
_root.clickTAG + "&banner_b_ck=2"  
  
on (release) {  
    if (_root.clickTAG.substr(0,4) == "http") {  
        getURL(_root.clickTAG + "&banner_a_ck=1", "_blank");  
    }  
}
```



ACTIONSCRIPT 3

As with single click-thru destination creative, all code will be added to the first frame of the main timeline. The clickTAG variable will need to be defined using the first block of code from the single click-thru ActionScript 3 section above.

For creative with multiple click-thru destinations, add parameters using the following code:

```
//call clickTAGs with parameters
clickButton1.addEventListener(MouseEvent.CLICK,
function(event:MouseEvent):void {
    mojoClick(event, "&ck=1") },false, 0, false);

clickButton2.addEventListener(MouseEvent.CLICK,
function(event:MouseEvent):void {
    mojoClick(event, "&ck=2") },false, 0, false);
```

In the sample above, the button names are `clickButton1` and `clickButton2`. You will substitute these names with the actual names of your buttons. Also, the parameter=value pairs in the sample are `"&ck=1"` and `"&ck=2"`. Be sure to update these with your own values if you want (although it is recommended to keep these). Just as with ActionScript 2 you can make more specific parameter names (i.e. `-banner_a_ck=1`).

REFERENCING THE CLICKTAG FROM WITHIN AN EMBEDDED SWF

If one SWF is displayed inside of another, the two SWFs have a parent/child relationship – the child being the SWF that is embedded within the host (parent) SWF

If you wish to reference the clickTAG from a button that exists within the child SWF, you must preface the clickTAG with `_level0` (level zero) rather than `_root`.

Example:

```
on (release) {
    if (_level0.clickTAG.substr(0,4) == "http") {
        getURL(_level0.clickTAG, "_blank");
    }
}
```



PACKAGING AND DELIVERY

Upload Flash creative into MOJO Adserver from the Creatives tab at either the client or campaign level. Instructions for this can be found in the Adserver Trafficking & Reports training manual available here: <https://app.mediaplex.com/help/Resources/index.html>.

FLASH BANNER DELIVERY SPECIFICATIONS

Please include the following contained in a Zip file:

- Published SWF for each creative.
- One designated default GIF or JPG banner per SWF to be served as a default (no Flash Player).
- For Flash movies with multiple click-thrus, include a text document listing each name value pair and corresponding destination URL.

Example:

Name	Value	Destination
ck	1	http://www.mediaplex.com
ck	2	http://www.mastertms.com
ck	3	http://www.mojorichmedia.com

UPLOADING FLASH CREATIVES INTO MOJO ADSERVER

1. From the Creatives tab, click the [Flash \(.swf\)](#) link.
2. In the Creative Details page, complete the New Flash Creative form:

Upload File	Enter or Browse for the SWF file.
Upload Alt Image	Enter or browse for the alternative GIF, JPG, or PNG image to be shown in case the Flash creative cannot be displayed.
Default Alt Image	Check only if a Default Alternate Image has been configured at the Client level for this creative size.
Size	Enter or select the dimensions of the Flash movie.
Click Thru URL	Enter default click-thru URL for this creative. This is typically only used if there is a unique landing page associated with this creative.
Org Code	Apply an organizational code to this creative, if desired. Org codes can be used later to create filters in reports.
Player Version	Choose the minimum Flash Player version necessary to play the SWF. If a user's Flash Player is lower than the designated version, the Alt Image will be displayed instead of the SWF.



3. Click **Submit**. The Creative Details page is displayed along with the automatically generated HTML and JavaScript files necessary to serve the SWF.
4. Click **Submit** again. The Campaign page, Creatives tab is displayed with a list of all creatives. Flash creatives are ready to be assigned to placements.
5. If there are any supporting flash files (i.e. – embedded SWFs) to upload, do so by uploading them in the Files tab immediately to the right of the Creatives tab.

CONFIGURING PLACEMENTS FOR CREATIVE WITH MULTIPLE CLICK-THRUS

For placements where Flash creative with multiple click-thrus will be running, it will be necessary to create **Data Match Rules** to associate a unique click-thru with each click interaction. These are set up under the Rules tab of the placement running the creative.

Example: Data match XML for configured ck=1 rule

```
<datamatch name="ck">  
<![CDATA[1]]>  
</datamatch>
```

For additional examples and instructions on configuring rules refer to the Help section in the Mojo Adserver or the Adserver Targeting Basics user guide available here:

<https://app.mediaplex.com/help/Resources/index.html>.



TESTING

TESTING CLICKTAG FUNCTIONALITY IN MOJO ADSERVER

All Flash movies and their click-thru URLs should be tested before trafficking the ad tags to publishers. Banners should then be tested with all current web browsers, including:

- Firefox
- Chrome
- Internet Explorer
- Opera

After creative has been uploaded into Adserver and associated with a placement, click [Submit](#) to submit placement changes. Be sure that the correct click-thru URL has been entered for the creative, placement, or campaign. Click [Active](#) to generate an insertion order fragment test. Because we are working with SWF creative, this test must be executed using JavaScript so click on the [JavaScript](#) link when prompted. When the creative is displayed, click in the appropriate area of the creative to test the clickTAG. If everything is configured properly you will be redirected to the correct destination. If you have more than one creative in rotation in the placement, you can refresh the test page until another creative is displayed and repeat the click test. You will also want to be sure that the destination page is opening up in a new tab or window if you included the "_blank" argument.

TESTING MULTIPLE CLICKTAG FUNCTIONALITY

After creative have been uploaded, assigned to placements, and the necessary data match rules have been established, perform the same steps as above. For each click check to see that the expected click-thru destination has been reached.



TROUBLESHOOTING

COMMON PROBLEMS

Click-thru doesn't work

Check to make sure that the clickTAG variable is properly encoded. Please refer to the Creative Production section of this document to help resolve these problems.

Be sure the button used to trigger the click is visible and positioned in front of all other objects on the stage.

One Part of the creative is clickable and another part is not

Check that the part that is not clickable has the clickTAG variable implemented correctly and that the button is configured correctly and on top of all other objects on the stage. If you want parts of a creative to click-thru to different locations, refer to the *Multiple click-thru destinations in one Flash movie* section in this document.

Click-thru works but reporting is showing zero clicks

Typically this is because the creative has a hardcoded URL. The solution is to properly code the clickTAG variable in the SWF. See the *Hardcoded URL pitfall* below.

PITFALLS

Literal string pitfall

Placing quotation marks (single or double) around the clickTAG makes it a literal string:

```
on (release) {
    getURL("_root.clickTAG", "_blank");
}
```

This results in the click-thru URL being exactly “_root.clickTAG” as opposed to the actual URL defined as the value for the clickTAG variable. The clickTAG must be an *expression* as opposed to a *'literal string.'*

Hardcoded URL pitfall

```
on (release) {
    getURL("http://www.mediaplex.com", "_blank");
}
```



This results in the inability to track clicks or modify the click-thru destination from within MOJO Adserver because the user is sent directly to the hard-coded page without being redirected through the Mediaplex servers.

Case-sensitivity pitfall

```
on (release) {  
    getURL(_root.ClickTag, "_blank");  
}
```

The clickTAG variable is case sensitive. There are a number of different case versions used by various ad servers (i.e. – clickTAG vs. clickTag). Mediaplex can support additional versions but we recommend using exactly `clickTAG` whenever possible. A case mismatch will typically result in the click not being redirected correctly and not being tracked.