

# ADTECH

Technical Guide

# Flash Banner Programming

ADTECH Helios IQ

2009-11-26

ADTECH AG

## Table of Contents

---

Introduction.....	3
Flash Functions for Tracking .....	4
Flash Functions for Layer Banners .....	5
Flash Functions for Expandable Banners.....	6
Flash Functions for Loading Additional Files.....	8
Security Settings.....	9

---

## Introduction

---

**Introduction**

This document describes how to prepare Flash files for tracking and other functions.

**Note:** This document does not describe how to program Flash files. See your Flash software documentation (Adobe Flash for example).

---

**Copyright and confidentiality**

All information from the ADTECH user guides is proprietary and to be treated as strictly confidential. Data is exclusively destined for the exclusive and internal use of the ADTECH customer. Any use, transmission, provision of access to third parties, circulation or any other utilization of the data or of information provided, other than contractual, is strictly prohibited.

---

**Address and contact**

ADTECH AG  
Robert-Bosch-Str. 32  
63303 Dreieich  
GERMANY

Phone: +49 (0) 61 03 - 57 15-0

Fax: +49 (0) 61 03 - 57 15-111

E-Mail: [info@adtech.de](mailto:info@adtech.de)

URL: <http://www.adtech.de/>

CEO: Dirk Freytag

---

## Flash Functions for Tracking

### Link URL with ClickTAGs

The link URL in the Flash file will be passed in the variable `clickTAG`. The variable will be connected to a button action in the most cases.

- The exact upper and lower case of the variable is mandatory! Although Helios IQ detects different variants for upper and lower case the given one is highly recommended: "clickTAG"! We do not recommend the following variants: `clickTag`, `clicktag`, `ClickTag`, `CLICKTAG`.
- The link URL will be entered in the Helios IQ front end during banner booking (see [Booking Flash Banners](#) and [The Flash Banner Options Window](#)).

The link URL can be programmed with the following ActionScript commands:

Type	ActionScript Command
getURL (Flash version 6 and higher)	<pre>on (release) {   getURL(_root.clickTAG, "_blank"); }</pre>
getURL (Flash version 5 and lower)	<pre>on(release) {   getURL(clickTAG, "_blank"); }</pre>
getUrlfunction (ActionScript 3)	<pre>if (root.loaderInfo.parameters["clickTAG"]) {   var clickTAG:String =   root.loaderInfo.parameters["clickTAG"]; } click_btn.addEventListener(MouseEvent.CLICK, getUrlfunction); function getUrlfunction(ev:Event = null): void {   ExternalInterface.call("window.open", clickTAG, "_blank"); }</pre> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>• <code>click_btn</code> is the name of the link button. You can change it into every name / label you want.</li> <li>• It is possible to embed the clickTAG via external files.</li> <li>• The command has been tested with Windows XP and various browsers: Internet Explorer 6 and 7: It works (it overrides the pop-up blocker). Firefox: It works (it overrides the pop-up blocker). Opera: The browser asks the user to open the pop-up window manually. Safari: It does not work.</li> </ul>

### Multiple ClickTAGs

Multiple clickTAGs will be programmed just like single clickTAGs.

- If a Flash file needs to contain more than one link URL they will be passed via variables with postpositioned numbers (`clickTAG1`, `clickTAG2` etc.) to differentiate them from each other.
- Helios IQ supports up to 10 clickTAGs in a Flash file.

## Flash Functions for Layer Banners

---

### Open layer

The variable `openTAG` contains the URL to show the layer. By default a Flash layer will be delivered visibly. So this function will be needed only to show the layer again, after it was hidden.

The layer opening can be programmed with the following ActionScript commands:

Type	ActionScript Command
<code>getURL</code>	<code>getURL(_root.openTAG, "_self");</code>
<code>fscommand</code>	<pre>on(release) {     fscommand("show"); }</pre>

---

### Close layer

The variable `closeTAG` contains the URL to close the layer.

**Note:** Layer banners should be closed in either case! They should be closed too if the Flash banner is completely transparent at the end of the animation because links that are covered by the transparent Flash banner cannot be clicked in some browsers.

The layer closing can be programmed with the following ActionScript commands:

Type	ActionScript Command
<code>getURL</code>	<pre>on(release) {     getURL(_root.closeTAG, "_self"); }</pre>
<code>fscommand</code>	<pre>on(release) {     fscommand("close"); }</pre>

---

## Flash Functions for Expandable Banners

### Expand banner

The variable `expandTAG` contains the URL to expand the expandable Flash banner to its full size. The expanded and collapsed sizes will be defined in the Helios IQ front end during banner booking (see [How to Book Expandable Banners \(with 1 File\)](#) and [The Flash Banner Options Window](#)).

The expanding can be programmed with the following ActionScript commands:

Type	ActionScript Command
getURL	<pre>on(release) {   getURL(_root.expandTAG, "_self"); }</pre>
fscommand	<pre>on(release) {   fscommand("expand"); }</pre>

### Collapse banner

The variable `collapseTAG` contains the URL to collapse the expandable Flash banner to its collapsed size. The expanded and collapsed sizes will be defined in the Helios IQ front end during banner booking (see [How to Book Expandable Banners \(with 1 File\)](#) and [The Flash Banner Options Window](#)).

The collapsing can be programmed with the following ActionScript commands:

Type	ActionScript Command
getURL	<pre>on(rollOut) {   getURL(_root.collapseTAG, "_self"); }</pre>
fscommand	<pre>on(rollOut) {   fscommand("collapse"); }</pre>

### Change height

The command `expandheight` changes the height of a layer. The desired height will be given as a parameter.

The changing of height can be programmed with the following ActionScript command (example with a height of 500 pixels):

Type	ActionScript Command
fscommand	<pre>on(rollOver) {   fscommand("expandheight", 500); }</pre>

**Change width**

The command `expandwidth` changes the width of a layer. The desired width will be given as a parameter.

The changing of width can be programmed with the following ActionScript command (example with a width of 500 pixels):

Type	ActionScript Command
fscommand	<pre>on(rollover) {     fscommand("expandwidth", 500); }</pre>

## Flash Functions for Loading Additional Files

### Loading additional files

To load additional files from a Flash file there are 2 ways, depending on where the file will be loaded from:

- Both Flash files were uploaded to Helios IQ: See *Additional file from Helios IQ* below in this topic.
- The second file will be loaded from an external source: See *Additional file from an external source* below in this topic.

To load additional files from the Flash file you can use any file format like .flv, .swf, .mp3, .avi etc.

### Additional file from Helios IQ

Additional files can be loaded from Helios IQ with the variable pathTAG. The variable contains the path to the banner files that the user uploaded. The additional files have to be uploaded into Helios IQ.

The loading can be programmed with the following ActionScript commands (example with a file flash2.swf):

Type	ActionScript Command
loadMovie (without pathTAG)	<code>loadMovie("flash2.swf")</code>
loadMovie (with pathTAG)	<pre>if(_root.pathTAG == undefined) {     _root.pathTAG = ""; } _root.loadMovie(_root.pathTAG+"flash2.swf");</pre>

### Additional file from an external source

Additional files can be loaded from external sources with the loadMovie command.

**Note:** You need to use security settings (see [Security Settings](#) on page 9).

The loading can be programmed with the following ActionScript command (example with a URL `http://www.adtech.de/movies/flash2.swf`):

Type	ActionScript Command
loadMovie	<code>loadMovie("http://www.adtech.de/movies/flash2.swf")</code>

## Security Settings

### Introduction

Since Adobe Flash version 7 the security settings for Flash files have changed.

- If the video in the Flash file is controlled with JavaScript function, or
- if external files or videos should be loaded,

then the respective domain has to be “unblocked” out of the Flash file with the command `system.security.allowDomain`.

**Attention:** It needs to be assured that these ActionScript security setting commands are executed!

### CURRENTDOMAIN and System.security.allowDomain

In the variable `CURRENTDOMAIN` the name of the domain where the banner will be shown is stored (Example: `www.adtech.de`). The variable will be passed to the function `system.security.allowDomain` to give JavaScript access to the Flash video. For example this is necessary if the Flash video will be started or stopped with JavaScript functions.

The domain access can be programmed with the following ActionScript command:

Type	ActionScript Command
<code>system.security.allowDomain</code> (Flash version 7)	<code>system.security.allowDomain(_root.CURRENTDOMAIN);</code>

### Security settings for loading external files

**Scenario:** A Flash video (a) booked in Helios IQ loads a Flash video (b) which is stored on an external server. The reloaded Flash video (b) accesses the link URL of the first Flash video.

This is only possible if `system.security.allowDomain` is implemented correctly.

- Flash 6 and 7: In order for the banner to function correctly even if delivered via different (ad) server, the URL of the parent element is identified with the variable `parent._url`. This way, you do not need to manually enter all necessary domains.
- Flash 8: It is possible to permit all domains with the placeholder “\*”.

The access can be programmed with the following ActionScript commands:

Type	ActionScript Command
<code>system.security.allowDomain</code> (Flash version 6 and 7)	<code>system.security.allowDomain(_parent._url);</code>
<code>system.security.allowDomain</code> (Flash version 8)	<code>system.security.allowDomain("*");</code>

**JavaScript access  
on ActionScript  
functions**

**Scenario:** A Flash video is to be restarted via JavaScript. To allow this in Flash, the current domain has to be allowed. The domain is passed in the variable CURRENTDOMAIN.

The access can be programmed with the following ActionScript commands:

Type	ActionScript Command
system.security.allowDomain (Flash version 6 and 7)	system.security.allowDomain(_root.CURRENTDOMAIN);
system.security.allowDomain (Flash version 8)	system.security.allowDomain("*");