

# AudioClipstream™ Tracking

**Disclaimer:** All content presented herein is subject to change without notice and is deemed as accurate as possible at time of publication. Please consult with Clipstream Support at [help@clipstream.com](mailto:help@clipstream.com) for clarification if you encounter any erroneous or inconsistent material in this document.

© 2000, 2001, 2002 Destiny Software Productions, Inc. All Rights Reserved. Clipstream, AudioClipstream, Clipstream Email, Bannerstream, Clipstream AudioMail and VideoClipstream are trademarks of Destiny Software Productions, Inc. All other trademarks are the property of their respective owners.

**Tracking**

**Impression Tracking**..... 3

**Click Through Tracking**..... 3

**Log Analysis**..... 3

**Marketing/Tracking Elements** ..... 4

    AudioClipstream.zip..... 4

    AudioHyperlink..... 4

    AudioHyperlinkInfo ..... 4

    AudioHyperlinkTarget ..... 4

    AudioStreamURL..... 5

    AudioTrackingURL..... 5

        Using the AudioTrackingURL Parameter with a CGI ..... 5

        Sample Tracking CGI Script..... 6

        Sample Log File ..... 7

        Interpreting the CGI Log File..... 7

    AuthorizedDomainURL ..... 8

    Button Images..... 8

    Event..... 8

    KeyURL ..... 8

    Logolcon ..... 8

    LogoLink ..... 8

## Impression Tracking

With AudioClipstream impressions can be tracked in a number of different ways.

1. The file audioclipstream.zip is called from the server every time the ad is displayed. This request can be tracked within the log of the server. Depending on the caching setup on the users system, however, this is not entirely accurate.
2. If images or other content elements are requested as part of the initial ad display, these transactions are recorded in the server log.
3. Requests for null images and elements can be implemented as part of the overall Clipstream ad package. These transactions are recorded in the server log or by some other tracking mechanism.
4. If the audio stream is set to auto play, streaming content files are requested from the server. These transactions are recorded in the server log.
5. A specific URL can be specified in the applet code that is called whenever the audio applet is loaded. The following information is passed and can be logged with a CGI or other script:
  - A. Action: Applet Load
  - B. Definable Parameter: User definable parameter, such as tracking or account number.

## Click Through Tracking

Click through events can tracked as follows:

1. User clicks to start, stop or pause the streaming content playback. A specific URL can be specified in the applet code that is called whenever the audio applet is loaded, started or stopped. The following information is passed and can be logged with a CGI or other script:
  - C. Action: Applet Loaded/Started/Stopped
  - D. Connection: auto-sensed kbps
  - E. Position: Start time and stop time (relative to content timeline)
  - F. Clip name (as defined in the applet code)
  - G. Definable Parameter: User definable parameter, such as an ID or account number.
2. Time based events can trigger calls to a web page, server or CGI using parameters within the applet.
3. User clicks within the audio applet or active link frame in the audio player to execute action such as visiting a web page or form, or opening a window. Serving of subsequent web page or event is recorded by it's web server. In addition, multiple links or events can be activated during specific time periods during the streaming playback. Tracking information can be passed with the URL or event call such as URL requested, time (relative to playback) clicked, and audio coordinates (X,Y, relative to applet) clicked. This information can be recorded via script or CGI on any server.

## Log Analysis

AudioClipstream facilitates fairly detailed statistical tracking through the web server log analysis tool you are likely using already. Since AudioClipstream uses typical HTTP for the serving of the streams and the necessary elements used by the applet, it is possible to accurately track a number of statistics.

### **Marketing/Tracking Elements**

A number of elements in the AudioClipstream applet code can be used in combination with you log analysis tool or with custom scripts and logs of your own design. The following is a brief description of the elements available. Detailed usage information can be found in the sections HTML Applet Parameters section.

#### **AudioClipstream.zip**

This file is requested from the server every time the applet is displayed. From this number you can see how many times the audio was available to hear.

#### **AudioHyperlink**

Using this parameter, the stream designer can specify URLs for particular time periods during the audio playback. When the user clicks the screen, they will be taken to the specified URL, or the URL will execute the specified action. Used in combination with the AudioHyperlinkInfo parameter makes "hot spotting" possible.

#### **AudioHyperlinkInfo**

Appends data to clickable URLs within the audio stream such as AudioURL, elapsed time in milliseconds, and position on the frame that was clicked along with the AudioHyperlink parameter. This can be particularly useful for tracking systems, and "hotspotting". Information can be tracked with your web log or stored in a custom log file of your design. From this information it can be ascertained exactly how long the user viewed the audio, allowing Clipstream designers to gage the effectiveness of their content.

Examples of values appended to URL:

URL: <http://www.clipstream.com/audio.22>

Time: 43200 ms

Coordinate: (158,118,160,120)

158,118 represent the position of the cursor when clicked (relative to top left corner of audio applet), and 160,120 represent the overall dimensions of audio.

These numbers automatically scale with the applet size.

#### **AudioHyperlinkTarget**

Particularly useful for marketing purposes for defining where and how URLs are executed in the audio stream. Links can be opened in the current window, a new window, or execute scripts, etc. Any typical HTML target can be implemented here.

### AudioStreamURL

Requests a particular stream from the server. From this it can be ascertained which speed was most requested to get a better idea of the capabilities of the listening audience.

### AudioTrackingURL

Specifies a URL or CGI to call when the user instigates the start or stop of the audio stream and returns the following information:

Action: was the audio Loaded, Started or Stopped?  
Connection: what speed was the user using?  
Time: when did the user start and stop the audio?  
Who: what was the IP address that viewed the audio? (via CGI)  
Extra Data: customized data can be passed.

Tracking data is appended to the URL as follows:

```
action={loaded|started|stopped}&kbps=sniffed_kbps&  
pos=start_millisec,stop_millisec
```

An extra parameter in the URL is allowed as demonstrated below in "userid=123"

Examples of values appended to URL:

When stream is started:

```
Track.cgi?userid=123&action=started&kbps=56&pos=0,0
```

When stream is stopped:

```
Track.cgi?userid=123&action=stopped&kbps=56&pos=0,120
```

### Using the AudioTrackingURL Parameter with a CGI

Query string submitted by applet:  
**extra1=value1&extra2=value2&action=loaded**

Data provided:

Date	<b>Tue Apr 23 09:04:52 2002</b>
Clip	
Action	<b>loaded</b>
kbps	
Start position	
Stop position	
Extra 1 (user defined)	<b>value1</b>
Extra 2 (user defined)	<b>value2</b>
Client IP (from cgi)	<b>192.168.0.1</b>
User Agent (from cgi)	<b>Java 1.1</b>

### Sample Tracking CGI Script

```
#!/usr/bin/perl -w

## Use the Perl CGI module
use CGI;

## Get the current time
my $now = scalar localtime;

## output in plain text
print "Content-type: text/plain\n\n";

## Return something to the applet
print "OK\n";

## Create new CGI object
my $req = new CGI;

## Get all the values passed to the script
my $clip = $req->param("clip");
my $action = $req->param("action");
my $kbps = $req->param("kbps");
my $pos = $req->param("pos");

## Get the extra values assigned to the tracking URL (any number of
user provided parameters, these are just examples)
## These are optional and do not need to be provided
## The parameter in the applet will look like:
## <param name="AudioTrackingURL" value="/cgi-
bin/tracking.cgi?extra1=value1&extra2=value2">
my $extra1 = $req->param("extra1");
my $extra2 = $req->param("extra2");

## Log results to file (you can also input the above values into a
database)
my $logFile = "./tracking.log";

## Open the log file (append)
open (LOGFILE,">>$logFile") || die "[$now] Can't open/create
$logFile : $!\n";

## Write info to the log file
print LOGFILE "[$now] $clip $action <$kbps> $pos $extra1 $extra2
$ENV{'REMOTE_ADDR'} [$ENV{'HTTP_USER_AGENT'}]\n";

## close log file
close LOGFILE;

1;
```

### Sample Log File

```
192.168.0.1 - - [24/Jun/2002:14:31:44 -0700] "GET /cgi-  
bin/TRACK.cgi?action=loaded&clientip=0.0.0.0&timestamp=1024954982 101  
HTTP/1.1" 302 322 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Win32)"  
192.168.0.1 - - [24/Jun/2002:14:32:19 -0700] "GET /cgi-  
bin/TRACK.cgi?action=loaded&clientip=0.0.0.0&timestamp=1024955016 370  
HTTP/1.1" 302 322 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Win32)"  
192.168.0.1 - - [24/Jun/2002:14:32:23 -0700] "GET /cgi-  
bin/TRACK.cgi?action=loaded&clientip=0.0.0.0&timestamp=1024955020 396  
HTTP/1.1" 302 322 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Win32)"  
192.168.0.1 - - [24/Jun/2002:14:32:39 -0700] "GET /cgi-  
bin/TRACK.cgi?action=loaded&clientip=0.0.0.0&timestamp=1024955036 779  
HTTP/1.1" 302 322 "-" "Mozilla/4.0 (Windows 2000 5.0)"  
192.168.0.1 - - [24/Jun/2002:14:32:45 -0700] "GET /cgi-  
bin/TRACK.cgi?action=started&clip=http%3A%2F%2Fclipstream.com%2Fsamp  
le.20%3F1024955041786&kbps=483&pos=0,0&clientip=0.0.0.0&timestamp=102  
4955042497 HTTP/1.1" 302 322 "-" "Mozilla/4.0 (Windows 2000 5.0)"  
192.168.0.1 - - [24/Jun/2002:14:32:50 -0700] "GET /cgi-  
bin/TRACK.cgi?action=stopped&clip=http%3A%2F%2Fclipstream.com%2Fsamp  
le.20%3F1024955041786&kbps=483&pos=0,4500&clientip=0.0.0.0&timestamp=  
1024955047615 HTTP/1.1" 302 322 "-" "Mozilla/4.0 (Windows 2000 5.0)"
```

**NOTE:** EPOCH timestamp is generally not used for statistical analysis – it is used to suppress caching.

### Interpreting the CGI Log File

**Loaded:** TRACK.cgi?action=loaded&clientip=0.0.0.0&timestamp=1024955036

action = loaded  
clip = null  
kbps = null  
pos = null

**Started:** TRACK.cgi?action=started&clip=http%3A%2F%2Fclipstream.com%2Fs  
sample.20%3F1024955041786&kbps=483&pos=0,0&clientip=0.0.0.0  
&timestamp=1024955042497

action = started  
clip = <http://clipstream.com/sample.20> (encoded format with appended  
timestamp)  
kbps = 483  
pos = 0,0

**Stopped:** TRACK.cgi?action=stopped&clip=http%3A%2F%2Fclipstream.com%  
2Fs  
sample.20%3F1024955041786&kbps=483&pos=0,4500&clientip=0.  
0.0.0&timestamp=1024955047615

action = stopped  
clip = <http://clipstream.com/sample.20> (encoded format with appended  
timestamp)  
kbps = 483  
pos = 0,4500

### **AuthorizedDomainURL**

The number of times this file was requested will tell you the number of times the audio was requested to play.

### **Button Images**

The number of times any of these images was requested will tell you the number of times the applet was successfully loaded.

ButtonPauseDown  
ButtonPauseOver  
ButtonPauseUp  
ButtonPlayDown  
ButtonPlayOver  
ButtonPlayUp  
ButtonStopDown  
ButtonStopOver  
ButtonStopUp

### **Event**

Your log analysis should be able to tell how many times specific elements used during the execution of an event were called.

### **KeyURL**

Number of times this file was requested will indicate number of times play button or auto play has been initiated.

### **Logolcon**

The number of times this file is requested will tell you how many times the applet was successfully displayed.

### **LogoLink**

Click throughs from the applet can be tracked with the statistics of this link.