



Clipstream

Playerless video and audio streaming

Clipstream™ Video 2.2 Technical Guide – Section 8 Applet Code

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™ Video Support at <http://clipstream.com/help> for clarification if you encounter any erroneous or inconsistent material in this document.

© 2000, 2001, 2002 Destiny Media Technologies, 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.

Section 8 - Applet Code

Applet Code Generator Page	8-2
Typical Applet Code	8-2
Where to Place the Applet Code	8-3
Body Tag Adjustment for Outlook Email	8-3
Applet Code Elements	8-4
Applet Tag	8-4
Typical Applet Tag	8-4
Changing the Applet Tag Parameters	8-4
Applet Tag Parameters	8-5
<applet values...>, </applet>	8-5
MAYSCRIPT	8-5
CODEBASE	8-5
archive	8-6
code	8-6
alt	8-7
name	8-7
width	8-7
height	8-8
General Applet Parameters	8-8
Pointers	8-8
Switches	8-8
Changing the General Applet Parameters	8-9
General Applet Parameter List	8-9
General Applet Parameter Syntax and Usage	8-10
AuthorizedDomainURL	8-10
AutoBitRate	8-11
AutoLoop	8-12
AutoPlay	8-12
BackgroundColor	8-12
CallBackOnPlay	8-13
Event[1...∞]	8-13
EventNumber	8-14
Key[1...∞]	8-14
KeyURL	8-14
MuteAudio	8-15
PanelEnable	8-15
PanelFlash	8-16
PanelImagesURL	8-16
PanelImageWidth	8-16
PanelLayout	8-17
PanelPopup	8-17
PanelPosition	8-18
PanelResizable	8-18
PreBuffer	8-19
SeekEnable	8-19
SeekEnableText	8-19
SeekMargin	8-20
TextColor	8-20
UseCaches	8-20
VideoBackground	8-21
VideoBackground[1...∞]	8-21
VideoBufferSize	8-22
VideoDelay	8-22

VideoEdgeColor	8-22
VideoEdgeWidth	8-23
VideoFrameBuffer.....	8-23
VideoHyperlinkInfo.....	8-24
VideoHyperlinkSize.....	8-24
VideoHyperlinkTarget	8-24
VideoHyperlink[1...∞].....	8-25
VideoMessageMode	8-26
VideoNumber	8-26
VideoPlaybackNumber	8-26
VideoSizeHeight	8-27
VideoSizeWidth.....	8-27
VideoTitleImageEndURL	8-27
VideoTitleImageURL.....	8-28
VideoTrackingURL.....	8-28
VideoURL[1...∞]	8-29
VideoURLEx[1...∞]	8-30
Alternate Messaging (for non-Java compliant viewers).....	8-31

Clipstream Video Applet Code

Every HTML document wanting to stream Clipstream Video must have the Clipstream Video applet code embedded into the body of the page HTML. The Clipstream Video applet code is simply a chunk of programming code that is cut and pasted into your HTML. The Applet code contains pointers to your files as well as custom parameters that control how your content is presented and controlled. Typically, this code is created using the Clipstream Video Applet Code Generation pages on the Clipstream.com website or by the handy code creation capability of the Clipstream Video Encoder. However more advanced and experienced users will be able to create and manipulate these parameters manually.

Applet Code Generator Page

In the help section of Clipstream.com you will find a Code Generator Utility (often referred to as the CodeGen page) to assist you with creating the necessary Clipstream Video applet code if you are unfamiliar or uncomfortable with manually changing the applet code within your HTML editor. The CodeGen page will ask you a number of simple questions and then create the code for you. We recommend that all new users use the CodeGen utility for the first few times they create the Clipstream Video applet code. In fact, even our developers use this utility as a starting point, prior to customizing the applet code. The CodeGen page can be found at:

http://www.clipstream.com/codegen/2.0/video_codegen_20_simple.shtml

An Advanced Code Generator Page is also available with more options is located at:

http://www.clipstream.com/codegen/2.0/video_codegen_20_advanced.shtml

Once your applet code is created, simply cut and paste it into your HTML.

Typical Applet Code

```
<applet MAYSCRIPT CODEBASE="http://www.yoursite/medialocation/"
archive="videoclipstream.zip" code="videoclipstream.class" alt="Clipstream
Video Seek" name="VCSPlayer" width=320 height=255>
  <param name="VideoURL1" value="500 kbps,510,500.vcs">
  <param name="VideoURL2" value="300 kbps,310,300.vcs">
  <param name="VideoURL3" value="150 kbps,160,150.vcs">
  <param name="VideoURL4" value="100 kbps,110,100.vcs">
  <param name="VideoURL5" value="44 kbps,40,44.vcs">
  <param name="VideoURL6" value="30 kbps,28,30.vcs">
  <param name="VideoURL7" value="20 kbps,0,22.vcs">
  <param name="VideoBufferSize" value="5">
  <param name="VideoTitleImageURL" value="320x240.jpg">
  <param name="PanelImagesURL" value="buttons.gif">
  <param name="PanelImageWidth" value="15,15,15,7,6,1,6,15,52">
  <param name="PanelLayout" value="PRSMC">
  <param name="PanelEnable" value="true">
  <param name="PanelPosition" value="bottom">
  <param name="BackgroundColor" value="#000000">
  <param name="TextColor" value="#FFFFFF">
  <param name="AutoPlay" value="FALSE">
  <param name="Key1" value="your code key">
  <param name="SeekMargin" value="5,3">
  <param name="SeekEnable" value="true">
```

A text message, image, and/or click-able link can be inserted here for the viewers that do not have Java enabled or cannot view Java in their email client or browser.

```
</applet>
```

Where to Place the Applet Code

The Clipstream Video applet code is intended to be placed within the body of your HTML page, between the body tags as follows:

```
<BODY>
```

Insert applet code between these tags.

```
</BODY>
```

Body Tag Adjustment for Outlook Email

Outlook has a bug that could cause it to crash when the email window containing Clipstream Video is closed. To reduce this risk, the author of any Clipstream Video content intended to be viewed from within the Outlook email viewer should insert the following parameter into the <body> tag of their HTML document:

```
onUnload="this.focus()"
```

This parameter would look as follows within the body tag:

```
<BODY onUnload="this.focus()" >
```

This tag keeps the focus of the email client on the email window instead of the Java applet and should prevent the crash. Clipstream Video's developers have created a work-around in version 2 that keeps this from happening but that line of code is still recommended.

Applet Code Elements

The Clipstream Video applet code contains several key elements:

- **Applet Tag** – which defines the basic dimensions of the applet and where to find the applet Java classes
- **Parameters** – which specify all other information the applet needs to operate properly. There are two types of parameters:
 - **Pointers** - to the locations (URLs) of the essential files needed for proper remote playback
 - **Switches** - that allow you to customize how the movie is played and how it is presented.
- **Alternate Messaging** – which is displayed when Java is not available.

The following sections will run through these elements in detail.

Applet Tag

The applet tag defines and constrains where the applet code begins and ends and contains a large amount of user configurable parameters. There are some portions of this code that should not be changed, and those will be noted. Each instance of the Clipstream Video applet must have an applet opening and closing tag.

Typical Applet Tag

A typical applet opening tag looks as follows:

```
<applet MAYSCRIPT  
CODEBASE="http://www.yoursite/medialocation/"  
archive="videoclipstream.zip" code="videoclipstream.class"  
alt="Clipstream Video Seek" name="VCSPlayer" width=320  
height=256>
```

Applet opening tag parameters will be discussed in the section **Applet Tag Parameters**.

The applet is terminated using the following necessary closing tag:

```
</applet>
```

Changing the Applet Tag Parameters

Even minor changes or omissions to the applet tag code can prevent Clipstream Video from operating as expected, so care should be taken when adjusting any parameter. Each time you change a value, you must save the page, re-upload it and reload your browser's cache before you can see the results. The codebase statement is useful for testing your html locally while playing a file that is on your server. See more on codebase in the next page.

Applet Tag Parameters

<applet values...>, </applet>

Type

Tag

Values

Various

Default

None

Description

Defines the beginning and end of the applet code. There are no user configurable parameters for this particular part of the Applet tag. This beginning and end tag are necessary.

Example

```
<applet>  
    applet code  
</applet>
```

MAYSCRIPT

Type

Switch

Values

None

Default

None

Description

Toggles the ability of the applet code to use defined Java methods. This switch is optional, but must be used if Java methods are to be called.

Example

MAYSCRIPT

CODEBASE

Type

URL

Values

URL

Default

None

Description

Defines the location of the videoclipstream.zip file (Java classes). This parameter is necessary when remotely delivering Clipstream Video content. For example, it is sometimes necessary for the applet and content to be located on a different server than the HTML page on which it appears. The CODEBASE statement allows this. General applet URL parameters can be referred to relative to the CODEBASE URL. Note that all elements used by the applet must be served from the same server.

Example

CODEBASE="http://www.clipstream.com"

archive

Type

string

Values

File name

Default

None

Description

Defines the name of the Java class archive referenced by the applet code. In most cases of Clipstream Video, the archive is always videoclipstream.zip. The end user should not change this value. There are a few exceptions to this, such as the dsLoader applet for small banners, but that is very specialized, and will be discussed elsewhere.

Example

archive="videoclipstream.zip"

code

Type

string

Values

File name

Default

None

Description

Defines the name of the Java classes referenced by the applet code. In all cases of Clipstream Video, the code value is always videoclipstream.class. This value should not be changes by the end user.

Example

code="videoclipstream.class"

alt

Type

string

Values

File name

Default

None

Description

Defines text to be displayed in the event the HTML page displaying Clipstream Video is viewed by a non-Java compatible viewer. In this instance, the text entered here will be displayed.

Example

alt="Clipstream Video Seek"

name

Type

string

Values

name

Default

None

Description

This attribute identifies the Clipstream Video applet to other applets and applications on the page. Scripting languages, such as JavaScript, will use this name to identify this instance of the Clipstream Video Applet when taking advantage of available Java methods.

Example

name="VCSPlayer"

width

Type

integer

Values

0 - ∞

Default

None

Description

Defines the width of the applet display in the HTML document. This attribute encompasses the entire width of the applet including the video frame and the video control panel.

Example

width="320"

height

Type

integer

Values

0 - ∞

Default

None

Description

Defines the height of the applet display in the HTML document. This attribute encompasses the entire height of the applet including the video frame and the video control panel.

Example

height="240"

General Applet Parameters

The general applet parameters specify all information needed by the applet for the playback and display of the video stream. Parameters can be broken into two types:

Pointers

An example of a pointer is the line:

```
<param name="VideoTitleImageURL" value="title.gif">
```

This parameter is labeled as "VideoTitleImageURL" (<param name="VideoTitleImageURL"). The value "*title.gif*" tells the applet player *where* to look for the title image to be displayed when the video is loaded and waiting for playback to begin.

Switches

An example of a switch is the line:

```
<param name="AutoPlay" value="false">
```

This parameter tells the player not ("false") to begin playing a video automatically once the page is loaded. (A "True" value would begin playing it automatically)

Other switches have values that are based upon numbered values.

The line below tells the player how much of a buffer (preloaded data) to load before it begins playing. Changing the number up or down decides the buffer size.

```
<param name="VideoBufferSize" value="5">
```

Changing the General Applet Parameters

Even minor changes or omissions to the applet code can have drastic effects on the quality of playback, so care should be taken when adjusting any parameter. Each time you change a value, you must save the page, re-upload it and reload your browser's cache before you can see the results.

General Applet Parameter List

The Clipstream Video applet has been designed to be as customizable and flexible as possible using parameters available within the applet code itself. The following list outlines all of the parameters available to the streaming author. Detailed descriptions follow.

PARAM	Description
AuthorizedDomainURL	Points to Text file listing sites permitted to codebase secure content
AutoBitRate	enable "Auto" in clip selection
AutoLoop	loop the clip
AutoPlay	start playing upon start/Rollover, FALSE/TRUE/ROLLOVER
BackgroundColor	background color in 0xRRGGBB format
CallBackOnPlay	set the callback on play JavaScript method
Event	time, target, and URL for each event Format: [LOAD STOP time in ms],target,URL
EventNumber	number of events
Key	Code Key
KeyURL	Code Key URL, point to a text file on the server with code keys on each line
MuteAudio	Mutes audio when playback is initiated
PanelEnable	enable panel buttons, same as PanelLayout=""
PanelFlash	Play/PauseStop button flash Mode: 0 - no flash 1 - flash on roll over 2 - flash play on waiting (first time only) 3 - flash play on waiting
PanelImagesURL	url for the panel image
PanelImageWidth	widths for each buttons within the panel image in the following order:PLAY, PAUSE, STOP, SEEK_BEGIN, SEEK_KNOB,SEEK_MID,SEEK_END, MUTE, CONN, SPEED

PanelLayout	order of the buttons, Play/Stop/Random/Conn/Mute
PanelPopup	show the popup menu at start
PanelPosition	top/bottom
PanelResizable	resize the panel for different Video dimension?
PreBuffer	preload the whole clip before playing
SeekEnable	enable seeking (the seek bar is still up, but cannot be dragged)
SeekEnableText	not used, enable/disable text on seek bar?
SeekMargin	left and right margin, in pixels
TextColor	text color in 0xRRGGBB format
UseCaches	use browser's cache or not
VideoBackground	background image used inside applet at atartup
VideoBackground#	background image used inside applet for each stream
VideoBufferSize	buffer size in seconds
VideoDelay	video delay relative to audio, used to sync audio
VideoEdgeColor	edge color in 0xRRGGBB format
VideoEdgeWidth	edge width in pixels, build 1114 includes the panel buttons
VideoFrameBuffer	decoded video bitmap buffer number, was VideoBitmapSize
VideoHyperlink	start_sec, end_sec, url START, title hyperlink url END, title at the end hyperlink url
VideoHyperlinkInfo	send clip, timestamp, pos with the Hyperlink? Useful for hotspot stuff
VideoHyperlinkSize	max number of video hyperlink URLs
VideoHyperlinkTarget	target frame to show the URL
VideoMessageMode	1:on screen, 2:in status bar, 3:all
VideoNumber	number of video URLs
VideoPlaybackNumber	set limit of plays, default unlimited
VideoSizeHeight	scale video to height in pixels, 0 to use stream's width
VideoSizeWidth	scale video to width in pixels, 0 to use stream's height
VideoTitleImageEndURL	title image at the end
VideoTitleImageURL	title image
VideoTrackingURL	URL to call upon load, start, stop, you can prefix which events to track, default to all. The cgi will be called with the following appended to the URL: action=[load start stopped]{&kbps=sniffed_kbps& pos=start_sec,stop_sec}&clientip=IP_ADDR extra parameter allowed
VideoURL	video url 1,2,3,... format "Name,MinBitRate,URL"
VideoURLEx#	location of title image in applet and stream on server

General Applet Parameter Syntax and Usage

AuthorizedDomainURL

Type
String

Values
URL

Default
None

Description

URL points to a text file on the server outlining sites allowed to codebase applets using content encoded with a secure ID. If this parameter is not used, content encoded with the secure ID will only be servable by the host server. Codebase protection, requiring the use of the AuthorizedDomainURL parameter to enable codebasing, is automatic when streams are encoded using a Secure ID (SID) or when the applet and its contents (vcs files) are located in a subfolder behind a folder called `/_scs_`.

Example

```
<param name="AuthorizedDomainURL" value="authorizedsites.txt">
```

Format of the text file is as follows:

```
URL #Comment1 ##Comment2 #EXP:YYMMDD
```

- URL is the URL of the site allowed to CodeBase. Sub directories can be specified. Note that when the URL is specified, all directories below are permitted.
- #Comment1 is any user definable comment
- #Comment2 is any user definable comment
- #EXP:YYMMDD is the date that the CodeBase permissions expire for that URL (Time based permission)

Text File Format Example:

```
#This is the list of web sites using codebase to remote streaming
http://www.clipstream.com## Clipstream
http://clipstream.com
http://www.dsnyc.com          ## DSNY
http://dsnyc.com
http://www.videoclipstream.com/demo
http://www.videoclipstream.com/akamai/testclips          #EXP:011201
```

AutoBitRate

Type

Boolean

Values

True, False

Default

True

Description

Automatically selects appropriate clip for playback based on the users' bandwidth available, compared to the settings in the VideoURL parameters.

Example

```
<param name="AutoBitRate" value="true">
```

AutoLoop

Type
Boolean

Values
True, False

Default
False

Description
Automatically loops the video when video end of stream is reached.

Example
<param name="AutoLoop" value="true">

AutoPlay

Type
Boolean

Values
True, False, Rollover, LoadOnly

Default
False

Description
Starts video playback as soon as video applet is loaded. Alternatively, the applet can be set to respond to a user rollover in the complete applet window for instigating initial playback. Using LoadOnly will start the clip in pause mode, allowing the full clip to load while waiting for user interaction for playback.

Example
<param name="AutoPlay" value="true">

BackgroundColor

Type
string

Values
#RRGGBB (RGB Hexadecimal)

Default
#FFFFFF

Description

Specifies the background color of the applet and textual displays, such as the bandwidth detect indicator and the buffering indicator as expressed in standard RGB hexadecimal. Background color of the seek knob rollover time indicator is not affected by this parameter. Control panel image components are not affected by this parameter.

Example

```
<param name="BackgroundColor" value="#FFFF00">
```

CallBackOnPlay

Type

string

Values

JavaScript Function

Default

None

Description

Specifies a JavaScript function that the applet can communicate with.

Example

```
<param name="CallBackOnPlay" value="JSStart();">
```

Event[1...∞]

Type

String

Values

URL / CGI call

Default

None

Description

Specifies a time based triggered event. Events are defined as time in milliseconds, a target for opening or executing, and a URL or CGI to execute.

Example

```
<param name="Event1"  
value="5000,BLANK,www.clipstream.com">
```

```
<param name="Event2"  
value="10000,Frame1,www.yahoo.com">
```

```
<param name="Event3"  
value="15495,Frame2,www.google.com">
```

EventNumber

Type
Integer

Values
Number of events.

Default
None

Description
Specifies the number of events the applet will execute.

Example
<param name="EventNumber" value="3">

Key[1...∞]

Type
string

Values
Key as issued by Clipstream

Default
none

Description
Specifies the code key for your server as issued to you from Clipstream. Any code key can be used for testing on a local hard drive, but only the code key for your specific IP or URL will enable Clipstream Video to be streamed from your server. Single or multiple code keys can be specified, depending on your server environment. Code keys can also be created to span a range of IP addresses within the same Class C.

NOTE: if the key is not specified, mistyped, expired, invalid or otherwise incorrect, the video stream will only play the first 20 seconds of the stream, and then produce an invalid code key message.

Example
<param name="Key" value="VALSKDNCSD894123">

or

```
<param name="Key1" value="VALSKDNCSD894123">  
<param name="Key2" value="VAEKDISCSD896269">  
<param name="Key3" value="VAMOESLCSD898346">
```

KeyURL

Type

string

Values

URL

Default

none

Description

Specifies the location and file name of a text file which contains a list of Clipstream Video code keys. This is useful when creating clips which must be served on several, non-sequential servers. The text file allows you to simply manage your code keys in the event your server environment changes, without having to change the key code in all instances of the applet code.

Example

```
<param name="KeyURL" value="./codekeys.txt">
```

Format of .txt file:

```
VALSKDNCSD894123  
VAEKDISCSD896269  
VAMOESLCSD898346  
Etc . . . . .
```

MuteAudio

Type

Boolean

Values

True, False

Default

False

Description

Disables audio playback when playback is initiated.

Example

```
<param name="MuteAudio" value="true">
```

PanelEnable

Type

Boolean

Values

True, False

Default

True

Description

Displays/enables panel buttons.
Not necessary when **PanelLayout** used.

Example

```
<param name="PanelEnable" value="true">
```

PanelFlash

Type

integer

Values

0, 1, 2, 3

- 0 – no flash
- 1 – flash on rollover
- 2 – Flash play on waiting (only when page loads)
- 3 – Flash play (always)

Default

2

Description

Specifies how play/pause & stop buttons flash

Example

```
<param name="PanelFlash" value="1">
```

PanelImagesURL

Type

string

Values

URL, .gif

Default

none

Description

Specifies the location and name of the panel button image. Image must be located on the same server as the videoclipstream.zip file.

Example

```
<param name="PanelImagesURL" value="./images/buttons.gif">
```

PanelImageWidth

Type

String

Values

0 - ∞

Numeric
Pixels

Default

none

Description

Specifies widths in pixels for each element in the control panel using the PanelImage in the following order:

PLAY, PAUSE, STOP, SEEK_BEGIN, SEEK_KNOB,
SEEK_MID, SEEK_END, MUTE, CONNECTION

Dimensions specified must absolutely correspond with the PanelImagesURL file, even if all controls are not used or specified by PanelLayout.

Example

```
<param name="PanelImageWidth" value="15,15,15,7,6,1,6,15,52">
```

PanelLayout

Type

string

Values

P,S,R,M,C

P – Play
S – Stop
R – Random (seek bar)
M – Mute
C – Connection Speed, Vide Choice or Chapter

Default

PSRMC

Description

Specifies the order of display for panel button images. Not all buttons need to be specified if you wish them omitted from the display.

Example

```
<param name="PanelLayout" value="PSRMC">
```

PanelPopup

Type

boolean

Values

True, false

Default

false

Description

Show the connection speed popup at video startup. The popup panel can also be used to display and select different videos and chapters while using a single bit rate.

Example

```
<param name="PanelPopup" value="false">
```

PanelPosition

Type

top/bottom

Values

top, bottom

Default

bottom

Description

Displays panel buttons in specified position relative to the video frame.

Example

```
<param name="PanelPosition" value="bottom">
```

PanelResizable

Type

boolean

Values

True, false

Default

True

Description

Toggles the applet to resize the control panel for videos of different dimensions (i.e. videos at different bit rates/connection speeds might be different sizes). The panel can be of a larger size than the video, however the overall dimensions of the applet must accommodate this.

Example

```
<param name="PanelResizable" value="true">
```

PreBuffer

Type
Boolean

Values
True, False

Default
False

Description
Buffers the video completely prior to playback. This has memory issues for longer videos. This is useful for short clips. If minimum connection speed specified by VideoURL is not met, the video will prebuffer prior to playback.

Example
<param name="PreBuffer" value="true">

SeekEnable

Type
boolean

Values
True, False

Default
True

Description
Enables seek / random access functionality. If set to false, seek bar is still displayed, but not functional. VCS files created with previous version of Clipstream Video will have all functionality except Seek. To enable seek on older VCS files, they will need to be re-encoded into 2.0 format. When using non-seek enabled files, the seek bar can be removed from the UI using the PanelLayout parameter by simply omitting the "R" value.

Example
<param name="SeekEnable" value="true">

SeekEnableText

(always enabled, not switchable)

Type
boolean

Values
True

Default
True

Description

Enable/Disable text on seek bar. This parameter is currently always enabled and not necessary in the applet code. This parameter is not user switchable.

Example

```
<param name="SeekEnableText" value="true">
```

SeekMargin

Type

String

Values

0 - ∞

Numeric

Pixels

Default

10,10

Description

Specifies the left and right margin of the seek bar, relative to the width settings as specified in the parameter PanellImageWidth, in pixels.

Example

```
<param name="SeekMargin" value="10,10">
```

TextColor

Type

string

Values

#RRGGBB (Color Hexadecimal)

Default

#000000

Description

Specifies the text color of textual displays, such as the bandwidth detect indicator, the buffering indicator, and the connection selection popup as expressed in standard RGB hexadecimal. Text color of the seek knob rollover time indicator is not affected by this parameter.

Example

```
<param name="TextColor" value="#FFFF00">
```

UseCaches

Type

boolean

Values

true, false

Default

true

Description

Specifies whether or not the browser should cache the current stream and applet.

Example

```
<param name="UseCaches" value="False ">
```

VideoBackground

Type

string

Values

X location, Y location, image

Default

0,0,null

Description

Specifies a location within the applet frame to place a background image. Use of X and Y locations are optional and default to 0,0.

Example

```
<param name="VideoBackground" value="10,10,background.gif">
```

VideoBackground[1...∞]

Type

string

Values

X location, Y location, image

Default

0,0,null

Description

Specifies a location within the applet frame to place a background image for each specific stream. VideoBackground number correlates to VideoURL number Use of X and Y locations are optional and default to 0,0.

Example

```
<param name="VideoBackground1" value="10,10,sunrise.gif">
```

```
<param name="VideoBackground2" value="10,10,sunset.gif">
```

VideoBufferSize

Type
integer

Values
1 - ∞ (seconds)

Default
10

Description
Specifies the number of seconds of compressed video and uncompressed audio to load into the input buffer. The VideoBufferSize is used to compensate for playback quality over various bandwidths to ensure enough video has been downloaded to playback smoothly. The input buffer also requires system RAM and should be calculated in conjunction with the VideoFrameBuffer.

Example
<param name="VideoBufferSize" value="10">

VideoDelay

Type
integer

Values
+ or - integer

Default
0

Description
Specifies video playback delay relative to audio in milliseconds to assist with audio syncing. This parameter is for unique circumstances and is rarely used.

Example
<param name="VideoDelay" value="1000">

VideoEdgeColor

Type
string

Values
#RRGGBB (Color Hexadecimal)

Default

#000000

Description

Specifies the color of the edge frame as expressed in standard RGB hexadecimal.

Example

```
<param name="VideoEdgeColor" value="#FFFF00">
```

VideoEdgeWidth

Type

integer

Values

0 - ∞ (pixels)

Default

0

Description

Specifies the width of the edge frame surrounding the video frame and control panel. Note that this frame size must also be accounted for in the height and width specifications in the applet header.

Example

```
<param name="VideoEdgeWidth" value="10">
```

VideoFrameBuffer

Type

Integer

Values

1 - ∞ (frames)

Default

4

Description

Specifies number of uncompressed video frames to load in the output buffer for playback. This number is a balance between CPU speed and memory (RAM) on the playback system. For example, those systems with low CPU speed should have a higher frame buffer to smooth playback. The trade off here is more RAM will be required. This is a particular issue for IMac systems. For those systems with high CPU speed, less frames need to be buffered. This parameter works in combination with the VideoBufferSize parameter.

Example

```
<param name="VideoFrameBuffer" value="4">
```

VideoHyperlinkInfo

Type
Boolean

Values
True, false

Default
False

Description
Enables applet to send data with the URL such as VideoURL, elapsed time in milliseconds, and position on the frame which was clicked along with the VideoHyperlink parameter. This can be particularly useful for tracking systems, and “hotspotting”.

Example
<param name="VideoHyperlinkInfo" value="true">

Examples of values appended to URL:

URL: <http://www.videoclipstream.com/demo.vcs>
Time: 43200 ms
Coordinate: (158,118,160,120)
158,118 represent the position of the cursor when clicked (relative to top left corner of video frame), and 160,120 represent the overall dimensions of video. These numbers automatically scale with the video size.

VideoHyperlinkSize

Type
integer

Values
0 - ∞ (hyperlinks)

Default
8

Description
Specifies the number of hyperlinks used in the video stream. This value is necessary when specifying hyperlinks as it allows Clipstream Video to create the hyperlink array without increasing the size of the applet.

Example
<param name="VideoHyperlinkSize" value="20 ">

VideoHyperlinkTarget

Type
string

Values

Target

Default

VCSHTMLFrame

Description

Specifies which frame or new window in which to open link specified in VideoHyperLink parameter. Any target can be specified.

Example

```
<param name="VideoHyperlinkTarget" value="_blank">
```

VideoHyperlink[1...∞]

Type

integer

Values

Integer (start seconds), Integer (end seconds), URL, start, end

Default

none

Description

Enables the video frame within the specified period to link to the specified URL when clicked upon.

- “start” value will make the specified hyperlink active in the video frame before the video begins playback and during buffering.
- “end” value will make the hyperlink active after playback has been stopped.
- Integer values indicate the active time period in seconds (begin and end) of a specified video frame hyperlink.
- If a hyperlink has not been specified for any point during playback, clicking on the frame will instigate play and pause functionality.

Example

These first two examples demonstrate a hyperlink in the frame prior to playback and in the frame after playback was stopped. In this instance during playback clicking in the frame will start and pause playback.

```
<param name="VideoHyperlink1" value="start, http://www.dsnny.com">  
<param name="VideoHyperlink2" value="end,http://www.dsnny.com">
```

In the next example, it has been specified that during playback between the 35 and 40 second marks, that clicking the video frame will go to the link www.sun.com.

```
<param name="VideoHyperlink3" value="35,40,http://www.sun.com">
```

VideoMessageMode

Type
integer

Values
1,2,3

- 1 – on screen
- 2 – in browser status bar
- 3 – both

Default
3

Description
Specifies the display location of textual video information (such as elapsed time, or buffering).

Example
<param name="VideoMessageMode" value="2">

VideoNumber

Type
integer

Values
1 - 15

Default
none

Description
Specifies the number of video files used by the applet. Used by the applet to help define total number of videos referenced with VideoURL number parameters not sequential from zero.

Example
<param name="VideoNumber" value="5">

VideoPlaybackNumber

Type
integer

Values
0 - ∞

Default
∞

Description

Specifies the number of times the video can be played back on the current web page.

Example

```
<param name="VideoPlaybackNumber" value="2">
```

Video will only be able to be played twice, unless the web page is refreshed.

VideoSizeHeight

Type

integer

Values

0 - ∞ (pixels)

Default

0

Description

Scales the height of the video to the specified pixels. Default setting of 0 (zero) uses original height of encoded content.

Example

```
<param name="VideoSizeHeight" value="240">
```

VideoSizeWidth

Type

integer

Values

0 - ∞ (pixels)

Default

0

Description

Scales the width of the video to the specified pixels. Default setting of 0 (zero) uses original width of encoded content.

Example

```
<param name="VideoSizeWidth" value="320">
```

VideoTitleImageEndURL

Type

string

Values

URL, .GIF, .JPG

Default

Last frame of video stream

Description

Specifies the location and name of static image file to be displayed when video ends. This image will resize to match the current frame size.

Example

```
<param name="VideoTitleImageEndURL" value="/image/endtitle.gif">
```

VideoTitleImageURL

Type

string

Values

URL, .GIF, .JPG, NULL, #RRGGBB

Default

Standard "Powered by Clipstream Video" message

Description

Specifies the location and name of static image file to be displayed when video loads, prior to playback. This image will not scale to match different sized video frames and should be created to match the largest size of video frame presented. In addition, using a value of NULL will display a black frame for the title image, or alternatively enter in an HEX RGB value to specify a frame color for the title image.

Example

```
<param name="VideoTitleImageURL" value="/image/title.gif">
```

- Displays the title.gif image on load.

```
<param name="VideoTitleImageURL" value="NULL ">
```

- Displays a black frame on load.

```
<param name="VideoTitleImageURL" value="#FF34BC">
```

- Displays a color frame on load.
-

VideoTrackingURL

Type

String

Values

URL / CGI call

Default

none

Description

Specifies the URL or CGI to call when the user instigates the start or stop of the video for tracking purposes. In addition, the URL or CGI can be called upon loading of the applet, thus making accurate impression tracking possible, regardless of whether or not the applet has been cached. Tracking data is appended to the URL as follows:

```
action={load|start|stop}&kbps=sniffed_kbps&
pos=start_sec,stop_sec&clientip=IP_ADDR
```

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

Example

```
<param name="VideoTrackingURL" value="track.cgi?userid=123">
```

Examples of values appended to URL:

When applet is loaded:

```
Track.cgi?userid=123&action=load&kbps=56&pos=0,0
&clientip=mpypc/192.168.0.1
```

When stream is started:

```
Track.cgi?userid=123&action=start&kbps=56&pos=0,0
&clientip=mpypc/192.168.0.1
```

When stream is stopped:

```
Track.cgi?userid=123&action=stop&kbps=56&pos=0,1
20&clientip=mpypc/192.168.0.1
```

VideoURL[1...∞]

Type
String

Values
String (name), Integer (Bitrate), URL

Default
none

Description
Specifies the various attributes of the video file to be played in the following order:

- Text to be displayed in Connection Popup (String) (please note space restriction in panel design)
- Minimum Bitrate (kbps) for Auto Bitrate Detection and file selection (integer)
- URL for location and name of video file or still image (for low bandwidth connections).

Example

```
<param name="VideoURL1" value="Modem,40,Video_Modem.vcs">
```

```
<param name="VideoURL2" value="Cable,120,Video_Cable.vcs">
<param name="VideoURL3" value="LAN,300,Video_LAN.vcs">
```

NOTE: If the minimum bit rate is not met, the lowest bit rate will be pre-loaded (completely buffered) prior to playback).

NOTE: List order is important to ensure the reliability of auto bitrate detection. List your bit rates in order from lowest bitrate to highest bitrate as shown above.

DO NOT MIX VideoURL AND VideoURLEx IN THE SAME APPLLET

VideoURLEx[1...∞]

Type
String

Values
String (name), Integer (Bitrate), Integer (x coordinate), Integer (y coordinate), URL

Default
none

Description

This parameter is similar in function to the VideoURL parameter, however it have the added functionality of being able to specify the XY location of the video frame within the overall applet for each bitrate when playback is initiated.

Specifies the various attributes of the video file to be played in the following order:

- Text to be displayed in Connection Popup (String) (please note space restriction in panel design)
- Minimum Bitrate (kbps) for Auto Bitrate Detection and file selection (integer)
- X location of the upper left corner of the video frame relative to the upper left corner of the applet
- Y location of the upper left corner of the video frame relative to the upper left corner of the applet
- URL for location and name of video file or still image (for low bandwidth connections).

Format
Name,MinBitRate,Video_X,Video_Y,URL

Example
<param name="VideoURL1" value="Modem,40,10,10,Video_Modem.vcs">
<param name="VideoURL2" value="Cable,120,100,40,Video_Cable.vcs">
<param name="VideoURL3" value="LAN,300,0,0,Video_LAN.vcs">

NOTE: If the minimum bit rate is not met, the lowest bit rate will be pre-loaded (completely buffered) prior to playback).

NOTE: List order is important to ensure the reliability of auto bitrate detection. List your bit rates in order from lowest bitrate to highest bitrate as shown above.

DO NOT MIX VideoURL AND VideoURLEx IN THE SAME APPLET

Alternate Messaging (for non-Java compliant viewers)

In some cases, the device or application that is viewing Clipstream Video content may not support java applets. In these cases, it is important to provide alternate messaging or imagery to the end user in normal HTML format. This can be accomplished within the Clipstream Video applet code very easily.

The Clipstream Video applet code created by the CodeGen pages comes with alternate messaging to assist the viewer with enabling Java on their device. In the applet code as demonstrated in the section **Typical Applet Code** the alternate messaging appears as such:

```
<hr>  
A text message, image, and/or clickable link can be inserted here  
for the viewers that do not have Java enabled or cannot view  
Java in their email or browser clients.  
<hr>
```

This block of code can be anything you wish. As a point of practice, try to keep the alternate messaging as the last item in the applet prior to the </applet> close.

In most cases we recommend keeping the default statement as is, however, as your situation may demand, altering this statement might be prudent. For example, if you are presenting a banner ad, you may wish this default messaging to be an image of your banner with a link, so that although the audio may not be heard, the viewer will still get the value of an image only banner.

In addition, there are a few free email viewers that prevent applets from running. This is another opportunity to change the alternate messaging to ensure your audio is heard by providing a link to a web page where your audio can be heard.

In particular, the Mac and AOL email viewers display alternate text messaging poorly. We recommend using a click-able linked image the same size as your applet dimensions as your alternate messaging.