AudioClipstream™ Serving

Serving

Serving Introduction

Contrary to other streaming solutions, the basics of serving AudioClipstream streaming audio are just that – Basic. To serve AudioClipstream you need nothing more than a typical web server. Whether it is an NT server, a Netscape Server, Unix, Linux or Mac – it does not matter. We have even successfully implemented AudioClipstream on a Lotus Domino Server. No special configuration, hardware or software is usually needed. There is really only one thing that must be remembered when serving AudioClipstream:

> **The audioclipstream.zip file, and any files used by the applet (such as the .20 and .32 files) must be located on the same server. This is a Java security feature that we have no control over.**

Uploading The Necessary AudioClipstream™ Files

For AudioClipstream™ to work from a web page you need to upload the following files.

1. Your web page with the AudioClipstream™ code embedded in the desired location.
2. The AudioClipstream™ player applet: audioclipstream.zip ***DO NOT UNZIP THIS FILE***.
3. Your encoded .20 and .32 AudioClipstream™ video files. If you intend to enable seek in your player, you must also upload the .20i and .32i files (created by the encoder – see section *Encoder*).
4. Your interface graphics if you are changing the look of the player. This includes all button images for the player control panel graphics. If you do not include these graphics, the stock player should show.

NOTE: It is not necessary to upload everything to the same directory as long as the locations of these files are specified using and absolute or relative URL in the HTML Applet Code. If these files are moved at a later time, the audio will likely not play properly or not at all.

File Size Estimations / Storage Considerations

When designing a serving solution for AudioClipstream, it is important to take into account the amount of storage space that will be needed for your AudioClipstream files.

The following equation can be used for estimating file sizes:

Streaming bit rate (kbps) x file length (seconds) / 8 = file size (KB)

For example:

A 2.5 minute clip at high speed (32kbps) would be no larger than 600KB

150 x 32 / 8 = 600

Bandwidth Usage Estimation

Calculating bandwidth usage and maximum concurrent viewers possible is fairly simple, and is mainly dictated by the power of your server hardware, and your available bandwidth. Assuming that your hardware can serve 100% of the bandwidth available, the equation is simply the available bandwidth divided by

the mixture of bit rates being served. It is difficult to estimate exactly what mixture of bit rates will be accessed at any given point in time, so it is better to take an average, or most popular bit rate and divide that into the available bandwidth.

For example a T1 connection at 1,500 kbps of data transfer could support 46.9 56k AudioClipstream listeners.

1,500 KB / 32kbps = 46.9

Registering Mime types

In some cases, a server will have mime types restricted, or not even registered. This is usually to prevent certain file types from being accessed from the server for various maintenance, procedure and security issues.

AudioClipstream requires that its mime types be registered with the server, or that the server allow any mime type. In short, the server must have the mime types for the extensions .20, .32, .20i, .32i, and .zip registered. In most cases the server will support these mime types automatically. The best thing to do prior to implementing any of these changes is to simply post an AudioClipstream clip to a page and try it out. If you know the file is on the server, yet the clip will not play (a *cannot find file* error dialog should be displayed in the browser in most instances), then try registering the mime types as follows:

Most Servers

On your server, add three lines to the .htaccess file:

AddType video/x-javaclip zip

AddType video/x-javaclip vcs

Once completed, make sure you can type in the URL of each file (.20, .32, .20i, .32i, .zip) in the browser and get something to download.

Apache Servers

AddType
Syntax: AddType mime-type extension extension...
Context: server config, virtual host, directory, .htaccess
Override: FileInfo
Status: Base
Module: mod_mime

The AddType directive adds to the list of filename extensions that filenames may end in for the specified content type. Mime-enc is the mime type to use for documents ending in extension. After content encoding and language extensions have been removed.

Example: AddType image/gif GIF

It is recommended that new mime types be added using the AddType directive rather than changing the TypesConfig file. Note that, unlike the NCSA httpd, this directive cannot be used to set the type of particular files.

Firewalls / Proxy / IP Issues

Firewalls and proxy servers frequently cause problems for many Internet applications. AudioClipstream is no exception.

If a firewall/proxy prevents AudioClipstream content from being served or viewed it is best to first check with your network administrator to ensure that .zip files and/or java is not being blocked. If so, then have the network administrator assign permissions for these specific items.

When encountering a Proxy Server using Microsoft Internet Explorer, the AudioClipstream applet will load and show the buttons but will not play because it fails the IP check (proxy servers frequently cannot resolve an IP address form a named URL). With Netscape, it won't even load the applet via the proxy. Using an absolute URL in CODEBASE and VideoURL parameter solves all the problems. For example:

<APPLET CODEBASE=" http://255.255.255.255" ARCHIVE="AudioClipstream.zip" CODE="AudioClipstream.class" ALT="The AudioClipstream player" NAME="AudioClipstream" WIDTH="87" HEIGHT="45">

Web sites with non-static IP addresses cannot use this method. Since the IP address changes all the time, they cannot use the IP address in the URL. In this instance, people behind a proxy without external IP resolution will not be able to listen. A solution is currently being investigated.

Code Keys – Development/Staging/Production Environments

Medium to large sites most often adopt a multi-tiered environment for web site development consisting of a Development Server, a Staging Server, and a Production Server. When AudioClipstream is being implemented into such an environment, a typical code key may not work properly, since each server will likely employ a different IP address. This becomes a problem when files are transferred from one server to another, the code key will be invalid for the new server, requiring each instance of the applet code to be updated on the new server with a new code key. This can be very labor intensive on large sites in particular.

To ease development in these situations, a code key can be generated to span multiple IP addressed within a specific Class C IP Block. When this new key is implemented, the applet code and key will be valid on a server within the block, and not require changes as files are moved from development, to staging, to production.

If your implementation requires a spanned code key, please contact your Clipstream Account Manager and provide them with the IP range you require.

Remote Delivery

To this point, for simplicity's sake, all sample implementations of AudioClipstream were assuming the audioclipstream.zip, and all other necessary files were located on the same server, and in the same directory/folder as the web page that wanted to display AudioClipstream. This sort of implementation is good for small-scale deployments of AudioClipstream, but does not really suit larger sites, and sites using dynamic page creation and database stores of web objects.

AudioClipstream can also be implemented to take advantage of clips stored remotely on another server, or simply in another location on a web server. Frequently, similar file types, such as the .20 and .32 files are centrally located in a file structure and are referred to from other pages. This will allow for easy maintenance of web pages for upgrading, and reduce the number of resources that may be duplicated across a typical web site.

A simple modification to the AudioClipstream applet code will allow remote delivery of AudioClipstream though the addition of a CODEBASE statement to the first line of the applet code. This modification is as follows:

<APPLET CODEBASE="http://www.your site.com/audio" ARCHIVE="audioclipstream.zip" CODE="audioclipstream.class" ALT="The AudioClipstream Movie Player" NAME="Audioclipstream" WIDTH="35" HEIGHT="53">

In this instance, the applet code is directed to look for its resources (in particular the audioclipstream.zip file) in web location of http://www.your site.com/video. This location can be edited to suit your particular implementation. You can put the applet code on any web page, as long as the codebase statement points to the server with your content and applet.

When implementing a remote delivery of AudioClipstream there are two essential things to note:

1. The audioclipstream.zip applet and the ..20, .32, .20i and .32i files must be delivered from the same Internet server. However, they do not necessarily need to be in the same directory on that server.
2. You will need to have a code key specifically for that server.