

Audio and Apache HTTPD

Sander van Zoest <sander@vanzoest.com>

ApacheCon 2002, Las Vegas, NV

\$Revision: 1.5 \$

<http://sander.vanzoest.com/talks/2002/audio_and_apache/>

Table of Contents

1. Why serve Audio on the Net?
2. What makes delivering audio so different?
3. Different ways of delivery Audio on the Net
4. Configuring Mime Types
5. Common Audio File Formats
6. Linking to Audio via Apache HTTPD
7. Configuring Apache HTTPD specifcly to serve large Audio Files
8. Icecast/Shoutcast Protocol
9. References and Further Reading

1. Why serve Audio on the Net?

This is almost like asking, why are you reading this? it might be because of the excitement caused by the new media that has recently crazed upon the internet. People are looking to bring their lives onto the net, one of the things that brings that closer to a reality is the ability to hear live broadcasts of the worlds news, favorite sport; hear music and to teleconference with others. Sometimes it is simply to enhance the mood to a web site or to provide audio feedback of actions performed by the visitor of the web site.

2. What makes delivering audio so different?

The biggest reason to what makes audio different then traditional web media such as graphics, text and HTML is the fact that timing is very important. This caused by the significant increase in size of the media and the different quality levels that exist.

There really are two kinds of goals behind audio streams. In one case there is a need for immediate response the moment playback is requested and this can sacrifice quality. While in the other case quality and a non-interrupted stream are much more important.

This sort of timing is not really required of any other media, with the exception of video. In the case of HTML and images the files sizes are usually a lot smaller which causes the objects to load much quicker and usually are not very useful without having the entire file. In audio the middle of a stream can have useful information and still set a particular mood.

3. Different ways of delivery Audio on the Net.

Embedding audio in your Web Page

This used to be a lot more common in the past. Just like embedding an image in a web page, it is possible to add a sound clip or score to the web page.

The linked in audio files are usually short and of low quality to avoid a long delay for downloading the rest of the web page and the audio format needs to be supported by the browser natively or with a browser plug-in to avoid annoying the visitor.

This can be accomplished using the HTML 4.0 [HTML4] object element which works similar to how to specify an applet with the object element. In the past this could also be accomplished using the embed and bgsound browser specific additions to HTML.

Example

```
<object type="audio/x-midi" data="../media/sound.mid" width="200" height="26">
  <param name="src" value="../media/sound.mid">
  <param name="autostart" value="true">
  <param name="controls" value="ControlPanel">
</object>
```

Each param element is specific to each browser. Please check with each browser for specific information in regards to what param elements are available.

In this method of delivering audio the audio file is served up via the web server. When using an Apache HTTPD server make sure that the appropriate mime type is configured for the audio file and that the audio file is named and referenced by the appropriate extension.

Although the current HTML 4.01 [HTML4] says to use the object element many browsers out on the market today still look for the embed element. Below find a little snippet that will work in many browsers.

```
<object type="audio/x-midi" data="../media/sound.mid" width="200" height="26">
  <param name="src" value="../media/sound.mid">
  <param name="autostart" value="true">
  <param name="controls" value="ControlPanel">

  <embed type="audio/x-midi" src="../media/sound.mid"
    width="200" height="26" autoplay="true" controls="ControlPanel">
    <noembed>Your browser does not support embedded MIDI files.</noembed>
</object>
```

With the increasing installation base of the Flash browser plug-in by Macromedia most developers that are looking to provide this kind of functionality to a web page are creating flash elements that have their own way of adding audio that is discussed in Flash specific documents.

Downloading via HTTP

Using this method the visitor to the website will have to download the entire audio file and save it to the hard drive before it can be listened to. (1) This is very popular with people that want to listen to high quality streams of audio and have a 64 Kbps or below connection to the internet. In some cases where the demand for a stream is high or the internet is congested downloading the content even for high bandwidth users can be affective and useful.

One of the advantages of downloading audio to the local computer hard drive is that it can be played back (once downloaded) any time as long as the audio file is accessible from the computer.

There are a lot of sites on the internet that provide this functionality for music and other audio files. It is also one of the easiest ways to delivery high quality audio to visitors.

(1) Microsoft Windows Media Player in conjunction with the Microsoft Internet Explorer Browser will automatically start playing the audio stream after a sufficient amount of the file has been downloaded. This can be accomplished because of the tight integration of the Browser and Media Player. With most audio players you can listen to a file being downloaded, but you will have to invoke the action manually.

On-Demand streaming via HTTP

The real difference between downloading and on-demand streaming is that in on-demand streaming the audio starts playing before the entire audio file has been downloaded. This is accomplished by a hand off from the browser to the audio player via an intermediate file format that has been configured by the browser to be handled by the audio player.

Look in a further section entitled "Linking to Audio via Apache HTTPD" below for more information about the different intermediate file formats.

This type of streaming is very popular among the open source crowd and is the most widely implemented using the MP3 file format. Apache, Shoutcast [SHOUTCAST] and Icecast [ICECAST] are the most common software components used to provide on-demand streaming via HTTP. Both Icecast and Shoutcast are not fully HTTP compliant, but Icecast is becoming closer. For more information about the Shoutcast and Icecast differences see the section below.

Sites like Live365.com and MP3.com are huge sites that rely on this method of delivery of audio.

On-Demand Streaming via RTSP/RTP

RTSP/RTP is a new set of streaming protocols that is getting more backing and becoming more popular by the second. The specification was developed by the Internet Engineering Task Force Working Groups AVT [IETF AVT] and MMUSIC [IETF MMUSIC]. RTP the Realtime Transfer Protocol has been around longer than RTSP and originally came out of the work towards a better teleconferencing, mbone, type system. RTSP is the Real-Time Streaming Protocol that is used as a control protocol and acts similarly to HTTP except that it maintains state and is bi-directional.

Currently the latest Real Networks Streaming Servers support RTSP and RTP and Real Networks own proprietary transfer protocol RDT. Apple's Darwin Streaming server is also RTSP/RTP compliant.

The RTSP/RTP protocol suite is very powerful and flexible in regards to most streaming needs. It has the ability to support "server-push" style stream redirects and has the ability to throttle streams to ensure the stream can sustain the limited bandwidth over the network.

For On-Demand streams the RTP protocol would usually stream over UDP and have a TCP connection open for RTSP. Because of the rich features provided by the protocol suite, it is not very well suited to allow people to download the stream and therefore the download via HTTP method might still be preferred by some.

Live Broadcast Streaming via RTSP/RTP

In the case of a live broadcast streaming RTSP/RTP shines. RTP allowing for UDP datagrams to be transmitted to clients allows for fast immediate delivery of content with the sacrifice of reliability. The RTP stream can be sent over IP Multicast to minimize bandwidth on the network.

Many Content Delivery Networks (CDNs) are starting to provide support for RTSP/RTP proxies that should provide a better quality streaming environment on the internet.

Much work is also being done in the RTP space to provide transfers over telecommunication networks such as cellular phones. Although not directly related, per se, it does provide a positive feeling knowing that all the audio related transfer groups seem to be working towards a common standard such as RTP.

On-Demand or Live Broadcast streaming via MMS.

This is the Microsoft Windows Media Technologies Streaming protocol. It is only supported by Microsoft Windows Media Player and currently only works on Microsoft Windows.

4. Configuring Mime Types

One of the most hardest things in serving audio has been the wide variety of audio codecs and mime types available. The battle of mime types on the audio player side of things isn't over, but it seems to be a little more controlled.

On the server side of things provide the appropriate mime type for the particular audio streams and/or files that are being served to the audio players. Although some clients and operating systems handle files fully based on the file extension. The mime type [\[RFC2045\]](#) is more specific and more defined.

The registered mime types are maintained by IANA [\[IANA\]](#). On their site they have a list of all the registered mime types and their name space.

If you are planning on using a mime type that isn't registered by IANA then signal this in the name space by adding a "x-" before the subtype. Because this was not done very often in the audio space, there was a lot of confusion to what the real mime type should be.

For example the MPEG 1.0 Layer 3 Audio (MP3) [\[ORAMP3BOOK\]](#) mime type was not specified for the longest time. Because of this the mime type was `audio/x-mpeg`. Although none of the audio players understood `audio/x-mpeg`, but understood `audio/mpeg` it was not a technically correct mime type. Later audio players recognized this and started using the `audio/x-mpeg` mime type. Which in the end caused a lot of hassles with clients needing to be configured differently depending on the website and client that was used. Last november we thanked Martin Nilsson of the ID3 tagging project for registering `audio/mpeg` with IANA. [\[RFC3003\]](#)

Correct configuration of Mime Types is very important. Apache HTTPD ships with a fairly up to date copy of the mime.types file, so most of the default ones (including `audio/mpeg`) are there.

But in case you run into some that are not defined use the `mod_mime.c` directives such as `AddType` to fix this.

Examples

```
AddType audio/x-mpegurl .m3u
AddType audio/x-scpls .pls
AddType application/x-ogg .ogg
```

5. Common Audio File Formats

There are many audio formats and metadata formats that exist. Many of them do not have registered mime types and are hardly documented. This section is an attempt at providing the most accurate mime type information for each format with a rough description of what the files are used for.

Real Audio

Real Networks Proprietary audio format and meta formats. This is one of the more common streaming audio formats today. It comes in several sub flavors such as Real 5.0, Real G2 and Real 8.0 etc. The file size varies depending on the bitrates and what combination of bitrates are contained within the single file. The following mime types are used

```
audio/x-pn-realaudio .ra, .ram, .rm
audio/x-pn-realaudio-plugin .rpm
application/x-pn-realmedia
```

MPEG 1.0 Layer 3 Audio (MP3)

This is currently one of the most popular downloaded audio formats that was originally developed by the Motion Pictures Experts Group and has patents by the Fraunhofer IIS Institute and Thompson Multimedia. [ORAMP3BOOK] The file is a lossy compression that at a bitrate of 128kbps reduces the file size to roughly a MB/minute. The mime type is `audio/mpeg` with the extension of `.mp3` [RFC3003]

Windows Media Audio

Originally known as MS Audio was developed by Microsoft as the MP3 killer. Still relatively a new format but heavily marketed by and becoming more popular by the minute. It is a successor to the Microsoft Audio Streaming Format (ASF).

WAV

Windows Audio Format is a pretty semi-complicated encapsulating format that in the most common case is PCM with a WAV header up front. It has the mime type `audio/x-wav` with the extension `.wav`.

Vorbis

Ogg Vorbis [VORBIS] is still a relatively new format brought to life by CD Paranoia author Christopher Montgomery; known to the world as Monty. It is an open source audio format free of patents and gotchas. It is a codec/file format that is roughly as good as the MP3 format, if not much better. The mime type for Ogg Vorbis is `application/x-ogg` with the extension of `.ogg`.

MIDI

The MIDI standard and file format [MIDISPEC] have been used by Musicians for a long time. It is a great format to add music to a website without the long download times and needing special players or plug-ins. The mime type is `audio/x-midi` and the extension is `.mid`

Shockwave Flash (ADPCM/MP3)

Macromedia Flash [FLASHAUDIO] uses its own internal audio format that is often used on Flash websites. It is based on Adaptive Differential Pulse Code Modulation (ADPCM) and the MP3 file format. Because it is usually used from within Flash it usually isn't served up seperatedly but it's extension is `.swf`

There are many many many more audio codecs and file formats that exist. I have listed a few that won't be discussed but should be kept in mind. Formats such as PCM/Raw Audio (`audio/basic`), MOD, MIDI (`audio/x-midi`), QDesign (used by Quicktime), Beatnik, Sun's AU, Apple/SGI's AIFF, AAC by the MPEG Group, Liquid Audio and AT&T's a2b (AAC derivatives), Dolby AC-3, Yamaha's TwinVQ (originally by Nippon Telephone and Telegraph) and MPEG-4 audio.

6. Linking to Audio via Apache HTTPD

There are many different ways to link to audio from the Apache HTTPD web server. It seems as if every codec has their own metafile format. The metafile format is provided to allow the browser to hand off the job of requesting the audio file to the audio player, because it is more familiar with the file format and how to handle streaming or how to actually connect to the audio server then the web browser is.

This section will discuss the more common methods to provide streaming links to provide that gateway from the web to the audio world.

Probably the one that is the most recognized file is the RAM file.

RAM

Real Audio Metafile. It is a pretty straight forward way that Real Networks allowed their Real Player to take more control over their proprietary audio streams. The file format is simply a URL on each line that will be streamed in order by the client. The mime type is the same as other RealAudio files `audio/x-pn-realaudio` where the pn stands for Progressive Networks the old name of the company.

Example

```
http://www.example.com/audio/file1.ra
http://www.example.com/audio/file2.ra
C:\audio\file1.ra
```

PLS

This is the playlist files used by Nullsoft's Winamp MP3 Player. Later on it got more widely used by Nullsoft's Shoutcast and has the mime type of `audio/x-scp1s` with the extension `.pls`. Before shoutcast the mimetype was simply `audio/x-pls`. As you can see in the example below it looks very much like a standard windows INI file format. The `length<number>` value can be either `-1` for a continues live broadcast or the length of the track in seconds.

Example

```
[playlist]
numberofentries=2
File1=http://www.example.com/audio/talk.mp3
Title1=ApacheCon 2002: Audio and Apache
Length1=-1
File2=http://www.example.com/audio/break.mp3
Title2=Break Between Sessions
Length2=900
```

M3U

This next one is the MPEG Layer 3 URL Metafile that has been around for a very long time as a playlist format for MP3 players. It supported URLs pretty early on by some players and got the mime type `audio/x-mpegurl` and is now used by Icecast and many destination sites such as MP3.com. The format is exactly the same as that of the RAM file, just a list of urls that are seperated by line feeds.

Example

```
http://www.example.com/audio/file1.mp3
http://www.example.com/audio/file2.mp3
C:\audio\file1.mp3
```

Certain MP3 players found the simple M3U format limiting and have added additional capabilities. Most of these capabilities, were already present in the PLS format and therefore have limited support over all. The ability to add comments has been added. Comments in M3U start with a `#` symbol and will be ignored by most players. Within the comments special tags such as `EXTM3U`, `EXTINF`.

`EXTM3U` signifies to the MP3 player that this M3U file has additional information. This extended data is provided before each URI with the `EXTINF:` tag.

Example

```
#EXTM3U
#EXTINF:-1,ApacheCon 2002: Audio and Apache
http://www.example.com/audio/talk.mp3
```

```
#EXTINF:900,Break Between Sessions
http://www.example.com/audio/break.mp3
```

SDP

This is the Session Description Protocol [RFC2327] which is heavily used within RTSP and is a standard way of describing how to subscribe to a particular RTP stream. The mime type is `application/sdp` with the extension `.sdp`.

Example

```
v=0
a=control:rtsp://stream.example.com/
c=IN IP4 192.168.12.44
a=lang:en
a=recvonly
m=audio 1554 RTP/AVP 96
e=sander-sdp@vanzoest.com (Sander van Zoest)
u=http://sander.vanzoest.com/talks/2002/audio_and_apache/
s=ApacheCon 2002: Audio and Apache
```

Sometimes you might see RTSL (Real-Time Streaming Language) floating around. This was an old Real Networks format that has been superseded by SDP. It's mimetype was `application/x-rtsl` with the extension of `.rtsl`

ASX

Is a Windows Media Metafile format [MSASX] that is based on early XML standards. It can be found with many extensions such as `.wvx`, `.wax` and `.asx`. I am not aware of a mime type for this format.

Extension Usage

<code>.wax</code>	All digital media is audio-only, with <code>.wma</code> file name extensions.
<code>.wvx</code>	Media contains video, with a <code>.wmv</code> file name extension.
<code>.asx</code>	Media created old versions of Windows Media Technologies and has an <code>.asf</code> file name extension.

Example

```
<ASX version="3.0">
  <ABSTRACT>Audio and Apache by Sander van Zoest explains on how
  to tie an Apache Web Server to Multimedia Content.</ABSTRACT>
  <TITLE>ApacheCon 2002: Audio and Apache</TITLE>
  <AUTHOR>Sander van Zoest</AUTHOR>
  <COPYRIGHT>2002 Alexander van Zoest</COPYRIGHT>
  <MoreInfo href="http://sander.vanzoest.com/talks/2002/audio_and_apache" />
  <Entry>
    <Ref href="MMS://netshow.example.com/wtoc.asf" />
    <Banner href="http://www.example.com/banner1.gif">
      <MoreInfo href="http://sander.vanzoest.com/talks/2002/audio_and_apache" />
      <Abstract>This is the description for this clip.</Abstract>
    </Banner>
  </Entry>
</ASX>
```

SMIL

Is the Synchronized Multimedia Integration Language [SMIL20] that is now a W3C Recommendation [W3SYMM]. It was originally developed by Real Networks to provide an HTML-like language to their Real Player that was more focused on multimedia. The mime type is `application/smil` with the extensions of either `.smil` or `.smi`

Example


```

<smil>
  <head>
    <layout>
      <root-layout id="video" width="159" height="20"/>
      <region id="comment" left="10" top="9" width="34" height="29" z-index="1"/>
      <region id="stats" left="105" top="14" width="43" height="75" z-index="1"/>
      <region id="title" left="12" top="99" width="113" height="15" z-index="1"/>
      <region id="caption" left="29" top="90" width="102" height="20" z-index="2">
    </layout>
  </head>
  <body>
    <seq>
      
      
    </seq>
  </body>
</smil>

```

MHEG

Is a hypermedia language developed by the ISO group. [MHEG1] [MHEG5] and [MHEG5COR]. It has been adopted by the Digital Audio Visual Council [DAVIC]. It is more used for teleconferencing, broadcasting and television, but close enough related that it receives a mention here. The mime type is `application/x-mheg` with the extension of `.mheg`

Example

```

{:Application
  ( '/startup' 0 )           // Application content reference
  :Items (
    {:Link
      1
      :EventSource 0          // Check this application...
      :EventType IsRunning    // ... for the IsRunning event
      :LinkEffect (
        // Load the scene
        :TransitionTo (( '~/hello.mhg' 0 ) )
      )
    }
  )
  :BackgroundColour '=FF=FF=FF=00' // White
  :TextColour '=00=00=00=00'       // Black
  :Font "rec://font/us1"           // Font to use for text rendering
  :FontAttributes "plain.26.32.0"  // Default font attributes
  :BitmapCHook 4                  // Default bitmap content hook
}

{:Scene
  ( "~/hello.mhg" 0 )
  :Items
  (
    // Declare a background Rectangle that covers the screen.
    {:Rectangle
      1
      :OrigBoxSize 640 480          // Size of rectangle
      :OrigPosition 0 0             // Position at top left
      :OrigRefLineColour '=ff=ff=ff=00' // White
      :OrigRefFillColour '=ff=ff=ff=00' // White
    }
  )
  :SceneCS 640 480
}

```

7. Configuring Apache HTTPD specifcly to serve large Audio Files

Some of the most common things that you will need to adjust to be able to serve many large audio files via the Apache HTTPD Server. Because of the difference in size between HTML files and Audio files, the `MaxClients` will need to be adjusted appropriately depending on the amount of time

listeners end up tying up a process. If you are serving high quality MP3 files at 128kbps for example you should expect more then 5 minute download times for most people.

This will significantly impact your webserver since this means that that process is occupied for the entire time. Because of this you will also want to increase the `Timeout` Directive to a higher number. This is to ensure that connections do not get disconnected half way through a transfer and having that person hit "reload" and connect again.

Because of the amount of time the downloads tie up the processes of the server, the smallest footprint of the server in memory would be recommended because that would mean you could run more processes on the machine.

After that normal performance tweaks such as max file descriptor changes and longer tcp listen queues apply.

8. Icecast/Shoutcast Protocol

Both protocols are very tightly based on HTTP/1.0. The main difference is a group of new headers such as the `icy` headers by Shoutcast and the new `x-audiocast` headers provided by Icecast.

A typical shoutcast request from the client.

```
GET / HTTP/1.0

ICY 200 OK
icy-noticel:<BR>This stream requires <a href="http://www.winamp.com/">
    Winamp</a><BR>
icy-notice2:SHOUTcast Distributed Network Audio Server/posix v1.0b<BR>
icy-name: Great Songs
icy-genre: Jazz
icy-url: http://shout.serv.dom/
icy-pub: 1
icy-br: 24

<data><songtitle><data>
```

The `icy` headers display the song title and other formation including if this stream is public and what the bitrate is.

A typical icecast request from the client.

```
GET / HTTP/1.0
Host: icecast.serv.dom
x-audiocast-udpport: 6000
Icy-MetaData: 0
Accept: */*

HTTP/1.0 200 OK
Server: Icecast/VERSION
Content-Type: audio/mpeg
x-audiocast-name: Great Songs
x-audiocast-genre: Jazz
x-audiocast-url: http://icecast.serv.dom/
x-audiocast-streamid:
x-audiocast-public: 0
x-audiocast-bitrate: 24
x-audiocast-description: served by Icecast

<data>
```

NOTE: I am mixing the headers of the controlling client with those from a listening client.

The CPAN Perl Package `Apache::MP3` by Lincoln Stein implements a little of each which works because MP3 players tend to support both.

One of the big differences in implementations between the listening clients is that Icecast uses an out of band UDP channel to update metadata while the Shoutcast server gets its meta data from the client embedded within the MP3 stream. The general meta data for the stream is set up via the `icy` and `x-audiocast` HTTP headers.

Although the MP3 standard documents were written for interrupted communication it is not very specific on that. So although it doesn't state that there is anything wrong with embedding garbage between MPEG frames the players that do not understand it might make a noisy bleep and chirps because of it.

9. References and Further Reading

[DAVIC]

Digital Audio Visual Council <<http://www.davic.org/>>

[FLASHAUDIO]

Tony DeYoung, "Flash Audio Tutorials", ARTLab, 2002.
<http://www.sonify.org/tutorials/links/pages/flash_audio/>

[HTML4]

D. Ragget, A. Le Hors, I. Jacobs, "HTML 4.01 Specification", W3C Recommendation, December, 1999. <<http://www.w3.org/TR/html401/>>

[IANA]

Internet Assigned Numbers Authority. <<http://www.iana.org/>>

[ICECAST]

Icecast Open Source Streaming Audio System. <<http://www.icecast.org/>>

[IETFAVT]

Audio/Video Transport WG, Internet Engineering Task Force.
<<http://www.ietf.org/html.charters/avt-charter.html>>

[IETFMMUSIC]

Multiparty Multimedia Session Control WG, Internet Engineering Task Force.
<<http://www.ietf.org/html.charters/mmusic-charter.html>>

[IETF SIP]

Session Initiation Protocol WG, Internet Engineering Task Force.
<<http://www.ietf.org/html.charters/sip-charter.html>>

[IPMULTICAST]

Transmit information to a group of recipients via a single transmission by the source, in contrast to unicast. IP Multicast Initiative <<http://www.ipmulticast.com/>>

[MIDISPEC]

The International MIDI Association, "MIDI File Format Spec 1.1",
<http://sander.vanzoest.com/talks/2002/audio_and_apache/midispec.html>

[MHEG1]

ISO/IEC, "Information Technology - Coding of Multimedia and Hypermedia Information - Part 1: MHEG Object Representation, Base Notation (ASN.1)"; Draft International Standard ISO 13522-1;1997; <<http://www.mheg.org/>> <<http://www.iso.org/>> <<http://www.ansi.org/>>

[MHEG5]

ISO/IEC, "Information Technology - Coding of Multimedia and Hypermedia Information - Part 5: Support for Base-Level Interactive Applications"; Draft International Standard ISO 13522-5:1997; <<http://www.mheg.org/>> <<http://www.iso.org/>> <<http://www.ansi.org/>>

[MHEG5COR]

Information Technology - Coding of Multimedia and Hypermedia Information - Part 5: Support for base-level interactive applications - - Technical Corrigendum 1; ISO/IEC 13522-5:1997/Cor.1:1999(E) <<http://www.mheg.org/>> <<http://www.iso.org/>>
<<http://www.ansi.org/>>

[MSASX]

Microsoft Corp. "All About Windows Media Metafiles". October 2000.

[<http://msdn.microsoft.com/library/en-us/dnwmmt/html/asx.asp>](http://msdn.microsoft.com/library/en-us/dnwmmt/html/asx.asp)

[ORAMP3BOOK]

S. Hacker; MP3: The Definitive Guide; O'Reilly and Associates, Inc. March, 2000.

[<http://www.oreilly.com/catalog/mp3/>](http://www.oreilly.com/catalog/mp3/)

[RFC2045]

N. Freed and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996. [<http://www.ietf.org/rfc/rfc2045.txt>](http://www.ietf.org/rfc/rfc2045.txt)

[RFC2327]

M. Handley and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[<http://www.ietf.org/rfc/rfc2327.txt>](http://www.ietf.org/rfc/rfc2327.txt)

[RFC3003]

M. Nilsson, "The audio/mpeg Media Type", RFC 3003, November 2000.

[<http://www.ietf.org/rfc/rfc3003.txt>](http://www.ietf.org/rfc/rfc3003.txt)

[SHOUTCAST]

Nullsoft Shoutcast MP3 Streaming Technology. [<http://www.shoutcast.com/>](http://www.shoutcast.com/)

[SMIL20]

L. Rutledge, J. van Ossenbruggen, L. Hardman, D. Bulterman, "Anticipating SMIL 2.0: The Developing Cooperative Infrastructure for Multimedia on the Web"; 8th International WWW Conference, Proc. May, 1999.

[<http://www8.org/w8-papers/3c-hypermedia-video/anticipating/anticipating.html>](http://www8.org/w8-papers/3c-hypermedia-video/anticipating/anticipating.html)

[W39CIR]

V. Krishnan and S. G. Chang, "Customized Internet Radio"; 9th International WWW Conference Proc. May 2000. [<http://www9.org/w9cdrom/353/353.html>](http://www9.org/w9cdrom/353/353.html)

[VORBIS]

Ogg Vorbis - Open Source Audio Codec [<http://www.xiph.org/ogg/vorbis/>](http://www.xiph.org/ogg/vorbis/)

[W3SYMM]

W3C Synchronized Multimedia Activity (SYMM Working Group);

[<http://www.w3.org/AudioVideo/>](http://www.w3.org/AudioVideo/)

\$Date: 2002/10/11 05:44:29 \$

Copyright © 2002 Alexander van Zoest. All Rights Reserved.