Audio Streaming

A New Professional Approach

Standards-Based to Maximize Your Audience

2020-02

Professional Features
Low Cost
High Performance
High Availability
High Reliability
High Quality

Greg Ogonowski
Nathan Niyomtham

www.indexcom.com

1249 S Diamond Bar Blvd
Suite #314
Diamond Bar CA 91765-4122
USA
+1 909 860-6760

Greg Ogonowski / Nathan Niyomtham
StreamS HiFi Radio/Modulation Index, LLC
2020-02

Why HLS/DASH.

Why StreamS HLSdirect™/DASHdirect™.

These are NOT questions.

HTTP **L**ive **S**treaming/**D**ynamic **A**daptive **S**treaming over **H**TTP
The WRITE tools for the WRITE job.
But it MUST be done RIGHT; in order to take advantage of ALL it has to offer.

Most of your streaming audience wants something that just works reliably; they don't care about the tech behind it. Many audio streams currently in service fall short. However, it is up to the content providers to make it work. If you stream audio, you need to know about this. This information is targeted to streaming professionals.

This paper/presentation describes a standards-based method of quality live streaming audio with dramatically decreased operating costs, increased reliability, professional features, and a better user experience. Streaming audio has now evolved past legacy protocols allowing reliable content delivery to mobile and connected car dashboards, where all the audience growth continues.

Currently, much live streaming audio is delivered using the popular SHOUTcast ICY protocol through a SHOUTcast or Icecast2 Streaming Server, or some proprietary derivation. Although these initial protocols served well (pun intended) initially, professional streaming requirements have changed in favor of a completely new approach using new streaming protocols that address new demands, provide performance increases, and lower deployment and operating cost. Standards-based HLS/DASH segmented streaming now provides all things necessary; however, it must be implemented correctly. Not all HLS/DASH found in the wild is equal or compatible.

Most video content providers have already upgraded their streaming technology to segmented streaming. These are just a few video content providers that have already discovered its benefits. It has made OTT video streaming commercially viable and the success that it has become, driving the cord-cutting trend.

If your audio stream provider and content distribution network do not support standards-based IETF Compliant HLS/DASH and do not closely follow the HLS/DASH specifications, make a change now before you are left behind in streaming technology. Be among the recognized video brands with streaming technology that is done the "write" way. Many developers and providers are unaware of these changes and are currently offering obsolete streaming technologies and services at very high costs.

To give listeners a good, reliable streaming experience and to keep them coming back, audio content providers need the same streaming technology changes that video has made. Great media should work, look, and sound the best it can, using the most up to date technology available on today's computers and mobile devices. It is your product. You owe this to yourself and your audience. Your revenue now depends upon it.

The Problem:

The main problem with audio streaming has been the lack of any formal standards that professionals require. Because of this, many existing streaming protocols have been modified and hacked to the point of becoming proprietary, leading to compatibility issues among streaming servers and player clients and devices.

This is NOT conducive to good business, frustrates users, and ultimately loses audience. AM/MW/FM transmitters all use standards-based protocols to transmit and receive, and have enjoyed huge success. As a result, any radio can receive any off-air signal. Compare this with the plethora of incompatible streaming players now in the hands of consumers. Simply put, streaming practices need to change if streaming audio is to continue to be a major driving force for audio content delivery. Standards-based IETF-HLS/MPEG-DASH offers this opportunity, addressing all the features and options that professionals need.

RTSP/RTP was basically the first streaming protocol for the Internet and was described under several Internet RFC (Request For Comments) open standards. All approved Internet standards have a corresponding RFC for compatibility and interoperability. This is extremely important for applications such as broadcasting and netcasting to maximize audience reach. If you stream using proprietary protocols, you limit your coverage and reduce your revenue opportunities. RTSP/RTP is very capable, being extensible in many ways. Unfortunately, that brought complexity to the protocol, and web browsers did not have direct support for RTSP/RTP. Furthermore, many developers didn't understand it and all its available options. So to simplify audio streaming, SHOUTcast ICY using MP3 was created.

SHOUTcast ICY (I Can Yell) (no joke) was originally a proprietary streaming audio protocol developed in 1998 by Nullsoft, Stephen 'Tag' Loomis, Tom Pepper and Justin Frankel for use with a proprietary non-open-source streaming server, Nullsoft DNAS, which has very limited features and extensibility. This specific protocol never became an Internet RFC, but was very similar to standard web HTTP with non-standard headers, yet different enough to cause browser incompatibilities. Traffic appeared as an infinite web page that would download continuously for the listening duration. Around the same time, Icecast appeared. Developed by Jack Moffet and Barath Raghavan, it originally used the Audiocast protocol. In 2004 they moved to Icecast2 using the easily reverse-engineered SHOUTcast ICY protocol for MP3 and open-source Ogg-Vorbis codecs. (Unlike MP3, Ogg-Vorbis never achieved wide commercial acceptance.) Icecast2 provided an advanced open-source ICY protocol server that was mostly, but not entirely, compatible with SHOUTcast. *For reliability both the SHOUTcast and Icecast2 ICY protocols require an encoder and decoder/player connection with a constant data stream, requiring perfect Internet service, which the Internet was never designed to deliver.* Furthermore, they were not designed with much consideration for the professional content provider. These protocols only provide basic streaming with limited metadata capabilities. ICY does not scale easily for large audiences, since it can't take advantage of standard cache configurations, and is therefore more costly to deliver. Unfortunately, despite all of its shortcomings, this is the most common live streaming protocol in use today, which has led to proprietary customization and caused further compatibility and reliability issues.

In 1993, Greg Ogonowski, who would become VP of Product Development at Orban, started listening to streaming audio on the Internet, which was then using RealAudio and Windows Media Audio codecs. Simply put, reliability and audio quality were horrible. Although novel, it was not nearly good enough to be commercially viable. In 1996, Greg started hearing streams that were reasonably reliable at low bitrates but suffered from execrable audio quality because the MP3 was bit-starved and sounded like very bad AM/MW radio. *"There had to be a better way,"* because this definitely had promise for worldwide audio media coverage. Recognizing a need, and realizing that the Internet was going to be the next way to deliver media, he spent the next six years researching better audio codecs in a quest to deliver better quality audio. In 2002, at an AES Convention in Los Angeles, Greg found a new audio codec, HE-AAC (aacPlus at that time) from Coding Technologies. Streaming audio history was about to be made. Greg was first to license AAC/HE-AAC under Orban for Internet streaming. In 2002 Greg, Ross Finlayson from live555.com (live.com at that time), and Nathan Niyomtham created the first live streaming encoder using MPEG standards-based AAC-LC/HE-AAC, Orban Opticodec-PC. It used RTSP/RTP originally; then popular ICY was added. Nullsoft/AOL Winamp soon followed. This changed everything, because now for the first time, commercial-grade, high quality audio could be streamed over the Internet using a dial-up connection with dramatically improved quality and reliability at lower cost. Greg and Nathan are now the StreamS HiFi Radio developers.

However, there was still another problem to be solved. There needed to be an AAC/HE-AAC player that was easy to use. So once again, working closely with Coding Technologies, Greg and Nathan created a Windows Media Player Plugin that supported AAC/HE-AAC over RTSP/RTP and HTTP/ICY. It enjoyed almost 10 million downloads before becoming obsolete once Microsoft and Apple included native AAC/HE-AAC support in their operating systems.

Then came Adobe Flash, which supported AAC/HE-AAC. The Adobe Flash Player was important because it was installed on most user systems and therefore didn't require a cumbersome download and install. This posed another challenge, as it required an Adobe Flash Streaming Server using proprietary RTMP or FLV protocols and transports. Here we go again!

To circumvent the use of an RTMP server, FLV transport was incorporated in certain builds of Icecast2, namely Icecast2 KH Build, from Karl Heyes. Greg and Nathan supplied Karl with the necessary information for FLV metadata so it could be included with Icecast2 FLV. But using FLV required using an expensive custom Adobe Flash Player from content providers, which was good for wide distribution because there was a huge installed base of the Flash Player, but bad for a number of security reasons. Adobe did not have the expertise or resources to contend with the security and performance problems, so an alternative was desperately needed. Apple has completely disallowed Flash in its mobile operating system, iOS, for many of these reasons, and is being deprecated everywhere else. So Flash players and Flash protocols are undesirable and finished.

There was hope that the upcoming HTML5 standards-based protocol would include support for streaming media. *However the HTML delivery protocol, HTTP, is not designed for realtime media delivery, nor is the Internet.* There is a certain truth to "www" meaning the "world wide wait." HTTP and HTML are file-centric protocols never designed for realtime, continuous data delivery. Although continuous streams can be made to work over the Internet, they require good reliable continuous connectivity with low latency, high bandwidth, and large buffers, all of which are not available everywhere, especially on crowded mobile networks, which are extremely important for media delivery. Consumer Internet connectivity or bandwidth rarely lives up to its performance expectations, despite what providers promote, so to ensure a good audience experience, content providers must do everything possible to work around these limitations.

The Solution:

After legacy streaming protocols were modified and hacked, with developers doing whatever they wanted regardless of standards, a standards-based streaming protocol was desperately needed to leverage standard HTTP/HTML file delivery for both live and file-based (on-demand/podcast) streams, and to provide *all* the necessary features professional content providers demand and their audience expects. This is now achievable using standards-based segmented file-based streaming such as HLS (HTTP Live Streaming), now officially IETF RFC 8216, created by Roger Pantos and Bill May of Apple Inc., or MPEG-DASH (Dynamic Adaptive Streaming over HTTP), ISO MPEG Standard ISO/IEC 23009-1. Now web servers or inexpensive cloud storage can deliver live and file streams with dramatically improved reliability and performance at a lower cost. This is the kind of transport the Internet was designed to deliver. Finally, streaming can use *"the right tool for the right job,"* or put another way, *"the write tool for the write job,"* as segmented streaming encoders write files to a server with a file writer, rather than push a constant, potentially unreliable stream to and from a streaming server.

MPEG-DASH remained incomplete for audio-only applications, lacking a formal now-playing PAD/metadata specification for several years. When Apple added fMP4 CMAF MPEG-DASH segmenting to the IETF HLS specification for MPEG-DASH compatibility, StreamS pointed out this omission, and it was then decided to use the same in-band, synchronous ID3 metadata used in HLS ES (Elementary Stream) audio-only, inside an MP4 emsg box for fMP4. This is now a complete ISO standard allowing IETF HLS and MPEG-DASH fMP4 ISO BMFF segments to be 100% compatible, and the best choice for audio-only streaming.

HTML5-MSE (Media Source Extensions) players in standard web browsers with such support, can now be used without depending on Flash or MP3. HLS and MPEG-DASH provides everything needed. All major video content providers have now moved away from continuous streaming protocols in favor of segmented file-based streaming. Now, audio-only content providers need to follow the same path to achieve the same gains.

Finally, standards-based segmented streaming is a reality. To be compliant and compatible however, developers need to read and understand these protocols to provide the most benefit, maximum coverage and revenue potential. This technology is based on well-known HTTP protocols. Numerous developers are already familiar with this, which facilitates lowering development, deployment, and operating costs.

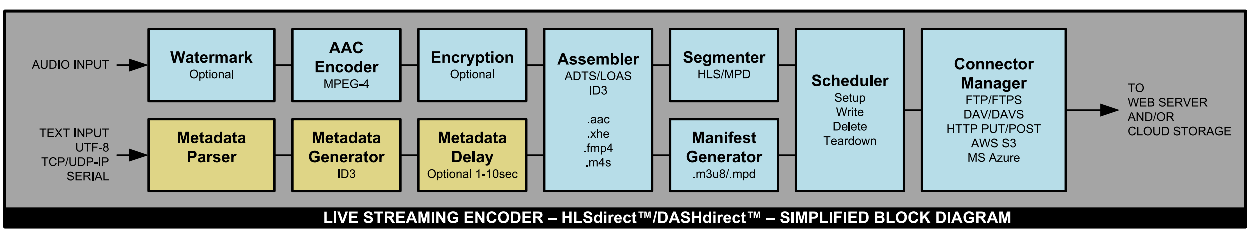
There are different ways to generate HLS/DASH streams. A streaming encoder that generates HLS/DASH directly, such as StreamS HLSdirect™/DASHdirect™, is responsible for assembling all timestamps, audio, and metadata. No special streaming server is required to use this full HLS/DASH implementation, further reducing cost while increasing performance and reliability. Other methods of generating HLS/DASH compromise the transport.

Performance, reliability, features, and compatibility that professional streamers demand are now available with StreamS HLSdirect™/DASHdirect™ Encoders that are fully IETF/ISO compliant. They are the first to support the new xHE-AAC codec, which allows reducing streaming bitrates even further. This new audio codec is available to existing AAC codec licensees at no additional cost. Operating system vendors can implement it simply.

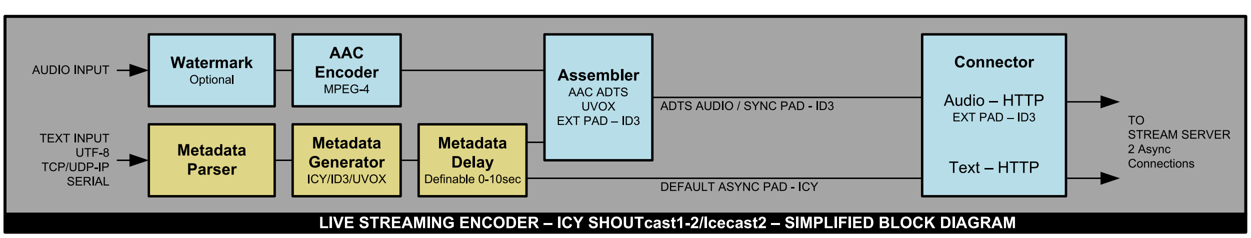
Many claim that Internet growth will intrinsically improve network reliability. While somewhat true, there will also be an increased network demand from device and increased network traffic. So to guarantee content reliability, segmented streaming is the only way.

Lift the limitations of legacy streaming audio protocols.

STREAMING AUDIO ENCODER BLOCK DIAGRAMS

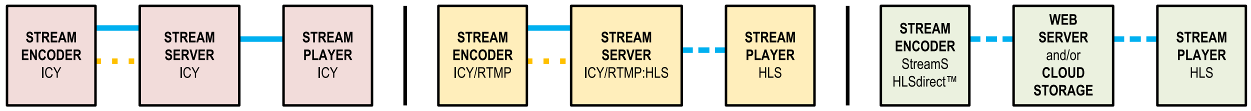


Live Streaming Audio Encoder – StreamS HLSdirect™/DASHdirect™



Live Streaming Audio Encoder – StreamS ICY Legacy

STREAMING SEGMENT/BITSTREAM DIAGRAMS - Simplified



ICY : ICY / FLV
SHOUTcast/Icecast

Requires a specialized streaming server and infrastructure.

Requires a constant data stream to and from the streaming server, which the Internet was not built for.

Subject to reliability issues.

Metadata is out-of-band, and not synchronous. It is sent from the server at a definable interval, not realtime, which can never be synchronous or on time.

Metadata is not extensible without legacy player compatibility issues.

ICY / FLV / RTMP : HLS/DASH
Hybrid HLS/DASH – Incomplete

50% HLS/DASH hybrid implementation.

Not all HLS/DASH advantages.

Still requires a specialized streaming server and infrastructure to generate HLSDASH segments from ICY/RTMP ingest.

Still requires a constant data stream to the streaming server, which the Internet was not built for.

Still subject to reliability issues.

Segmented data streams are only for players.

HLS : HLS/DASH
StreamS HLSdirect™/DASHdirect™

100% HLS/DASH implementation.

ALL HLS/DASH has to offer.

Encoder generates all HLS/DASH segments.

Leverages inexpensive web servers and/or cloud storage.

Uses segmented data streams to and from the server appearing exactly like web traffic, which the Internet was built for.

Metadata is extensible, in-band and completely synchronous.

Easily scaled for high availability.

This implementation provides a true professional streaming audio solution.

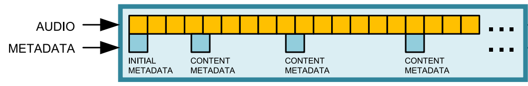
HLS/DASH is not just a media delivery protocol to reach audience players.
HLS/DASH is also a media delivery protocol to reach servers.
Using these protocols on both sides of media delivery is more reliable and robust than legacy protocols.

To take full advantage of what HLS/DASH has to offer, it should be used on both media stream ingest to servers and client players.

Content Distribution Networks (CDNs), that are not offering HLS/DASH ingest and delivery protocols are simply doing a huge disservice to the streaming industry, offering outdated technology. Many content distribution networks deprive their clients of new improved technologies, because of their inability or lack of commitment to develop the software to get them there. Choose your content distribution network very carefully.

STREAMING SEGMENT/BITSTREAM DIAGRAMS - Detailed

ENCODER GENERATED BITSTREAM AND METADATA



ICY encoders produce asynchronous out-of-band metadata.
This is the most common streaming audio encoder available, complete with all the unavoidable ICY limitations. Subject to encoder-to-server stream reliability issues. Given minimum network latency, it will work *most* of the time. Recovery is not graceful. Audio and metadata do not remain in sync, and makes this a poor choice for professional streaming, especially requiring precision metadata and/or cueing.

ICY BITSTREAM AND METADATA – AAC ADTS / xHE-AAC LOAS
TCP/IP HTTP – Single continuous HTTP file transfer for audio, and a separate metadata update connection.

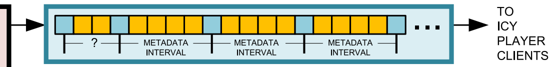
SERVER
Streaming

Special

HTTP ICY
Streaming Server

Most Popular
Protocol

SERVER GENERATED BITSTREAM

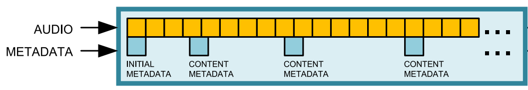


Server inserts asynchronous metadata into the audio bitstream as best effort from an out-of-band metadata connection, and is output at server configurable intervals. Metadata is asynchronous to the audio, as the encoder is also asynchronous, time sync is only approximate, and always varies, and cannot be used for precise metadata and content insertion. Subject to server-to-client stream reliability network issues. Given enough buffering and minimum network latency, this will work *most* of the time. Recovery is not graceful.
This is a poor, unreliable, low feature streaming protocol.

ICY BITSTREAM WITH METADATA – AAC ADTS / xHE-AAC LOAS
TCP/IP HTTP Data Stream – Single infinite HTTP file transfer.
Metadata is NOT the same as Encoder Input.

Bitstream Diagram – Encoder-Server-Client – ICY SHOUTcast/Icecast : ICY/FLV

ENCODER GENERATED BITSTREAM AND METADATA



ICY encoders produce asynchronous out-of-band metadata.
This is the most common streaming audio encoder available, complete with all the unavoidable ICY limitations. Subject to encoder-to-server stream reliability issues. Given minimum network latency, it will work *most* of the time. Recovery is not graceful. Audio and metadata do not remain in sync, and makes this a poor choice for professional streaming, especially requiring precision metadata and/or cueing.

ICY BITSTREAM AND METADATA – AAC ADTS / xHE-AAC LOAS
TCP/IP HTTP – Single continuous HTTP file transfer for audio, and a separate metadata update connection.

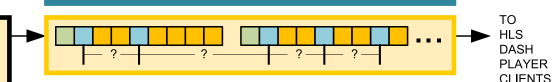
SERVER
Streaming

Special

HTTP ICY
Ingest

HTTP HLS/DASH
Streaming Server

SERVER GENERATED FILE SEGMENT

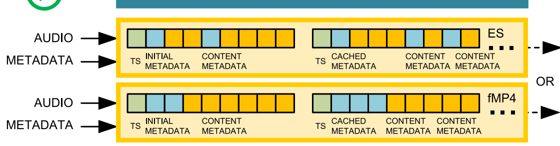


Server inserts asynchronous metadata into the segments as best effort from an ICY connection. Metadata is asynchronous to the audio, as the encoder is also asynchronous, and time sync is only approximate and always varies. Alternate metadata in .m3u8 is non-compliant and also asynchronous. Cannot be used for precise metadata and content insertion. Subject to encoder-to-server stream reliability network issues.
This is a poor, incomplete, compromise HLS/DASH implementation.

HLS/DASH FILE SEGMENT – ES Audio-Only - AAC ADTS / xHE-AAC LOAS or Fragmented fMP4 ISO BMFF CMAF – HLS/DASH Compatible
HTTP File Transfer – Segments are separate HTTP file transfers.
Metadata is NOT the same as encoder generated segments.

Bitstream Diagram – Encoder-Server-Client – ICY SHOUTcast/Icecast / FLV / RTMP : HLS / DASH

ENCODER GENERATED FILE SEGMENT



A compliant HLS/DASH Encoder is responsible for audio encoding, and the assembly of timestamps, synchronous metadata, and audio frames into a single segment bitstream, forming complete file segments, and then uploaded to the server sequentially. All audio and metadata "heavy-lifting" is done at the encoder level. Robust encoder-to-server connection is maintained by separate TCP connections for segments.

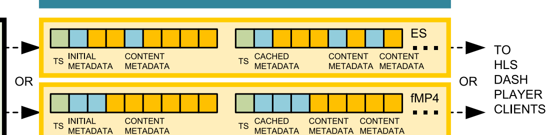
HLS/DASH FILE SEGMENT – ES Audio-Only - AAC ADTS / xHE-AAC LOAS or Fragmented fMP4 ISO BMFF CMAF – HLS/DASH Compatible
HTTP File Transfer – Segments are separate HTTP file transfers.
Metadata remains synchronous in encoder generated segments.

SERVER
File-Based

Standard

HTTP
Web Server
or
Cloud Storage

SERVER TRANSFERRED FILE SEGMENT

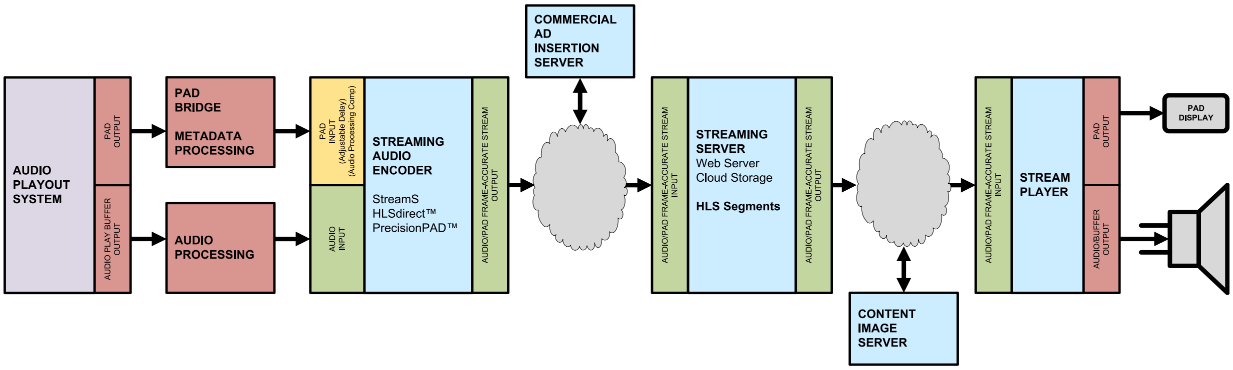


Server is completely transparent and passes HLS/DASH file segments through without modification. Metadata remains synchronous to the audio, as assembled by a compliant encoder, regardless of server perfor-mance, and allows precise metadata and content insertion. Robust server-to-client connection is maintained by multiple TCP connections and inherent segment buffer.
This is the best HLS implementation, offering the most flexibility, options, and eliminating all ICY, FLV, and RTMP limitations.

HLS/DASH FILE SEGMENT – ES Audio-Only - AAC ADTS / xHE-AAC LOAS or Fragmented fMP4 ISO BMFF CMAF – HLS/DASH Compatible
HTTP File Transfer – Segments are separate HTTP file transfers.
Metadata EXACTLY the same as encoder generated segments.

Bitstream Diagram – Encoder-Server-Client – HLS/DASH : HLS/DASH

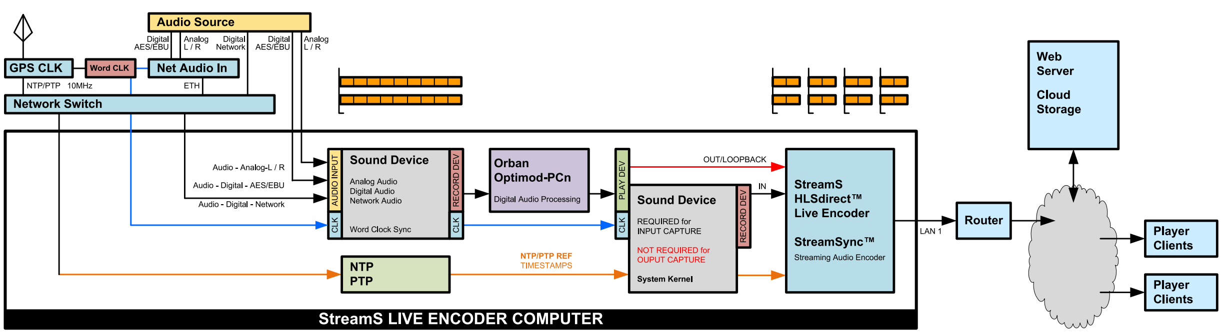
STREAMING AUDIO PLAYOUT-to-PLAYER SYSTEM BLOCK DIAGRAM



Typical Audio Streaming Playout System

- Audio Playout System
- Audio Processing
- PAD Processing
- Audio Streaming Encoder
- Servers – Streaming/Content Ad Insertion/Content Images
- Client Players

STREAMING ENCODER/SERVER/PLAYER BLOCK DIAGRAM



Typical Complete Audio Streaming System Block Diagram

Audio Input – Audio Processing – Encoder – Server – Player Clients

Audio Input Options:

- Analog Audio
- Digital AES/EBU Audio
- Digital Network Audio

StreamS HLSdirect™/DASHdirect™ Live Streaming Audio Encoders

Optimized for professional audio streaming applications including all time-synchronous PAD/metadata.
Video encoders are excessively expensive for audio-only use, and lack the necessary specialized features.

Licensed for Commercial Use
Unlike open-source implementations that require additional codec licensing for commercial use, StreamS HLSdirect™/DASHdirect™ Live Encoder software is ready to go without any additional licensing, and is DMCA Compliant.

Standards-Based Compliant
HLS is IETF RFC 8216 and ISO 23000-19.
DASH is ISO/IEC 23009-1 and ISO 23000-19.
Finally, modern streaming media protocols engineered for the *real* Internet.

No Dedicated Streaming Server Software Required
Leverages standard file-based web server or cloud infrastructure to stream live, and to serve files for on-demand/podcast content. Does not require any special web server module, allowing any web server on any platform, including simple cloud storage, to be used, available from many competitive providers at lower cost than dedicated streaming servers. Cloud server applications are not required, keeping performance high, and cost low. Uses StreamS HLSdirect™/DASHdirect™ for direct ingest to content servers with HLS encoder-to-server connections and no packet or segment conversion. HTTP/2 ready. Does not rely on legacy ICY (SHOUTcast/Icecast) or Adobe Flash RTMP protocols for audio ingest, with their associated limitations, to deliver converted HLS/DASH.

Servers

Cloud Infrastructures

Decreases Deployment Costs
Does not require separate streaming server infrastructures and specialized deployment staff.

Decreases Operating Costs
Does not require separate streaming server hardware, software, and licensing, and no extra specialty support staff.

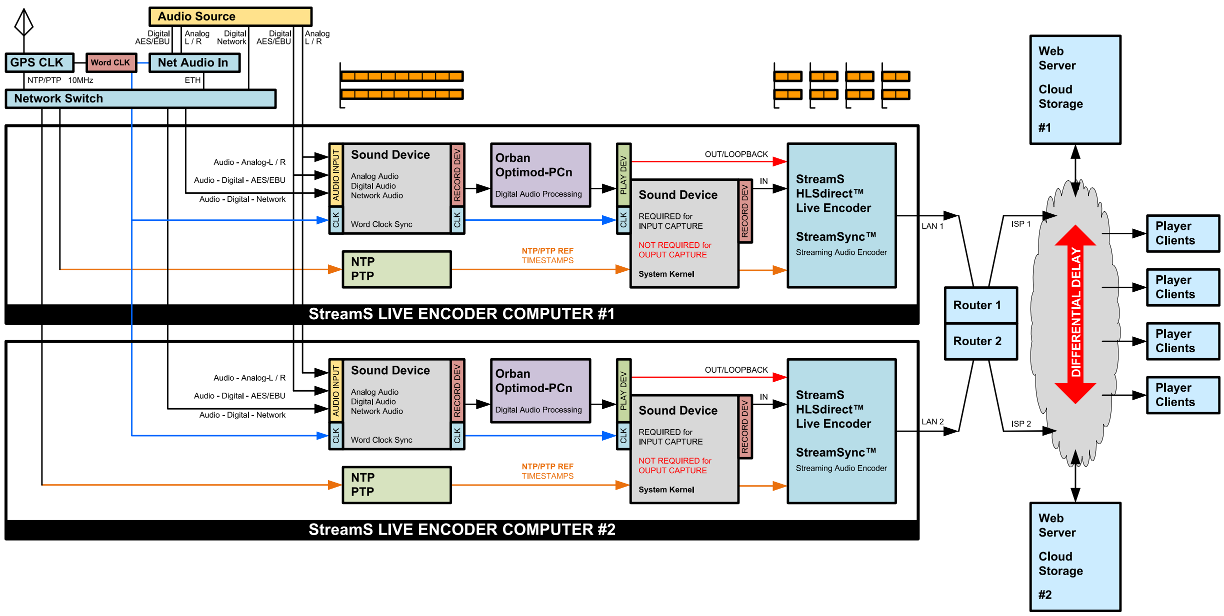
Increases Reliability
Less hardware and software to fail. HLS/DASH protocols are greatly more robust than ICY or RTMP/FLV.

Network Scalability
Less-complex server infrastructure to scale compared to dedicated streaming servers.
Allows reliable high-availability network scalability.

Improved Server Performance
Lower server load, since there are no continuous connections to a dedicated streaming server, and metadata does not require special processing or caching.

Redundant Synchronous Encoder Configuration
StreamS StreamSync™ allows multiple encoders to be configured to run concurrently, in parallel, time-synchronized, for complete unattended stream redundancy, from any audio source and/or location. Streams are locked and synchronized using AES Word Clock and/or optional PTP, optionally referenced to GPS. Should an encoder become unavailable for whatever reason, the player client seamlessly gets data from one of the other stream ingest points specified in the Playlist Manifest without gaps, buffering, or other interruptions. Metadata also remains time-synchronous.
Specialized RTP sources are not required.
Fully compliant HLS/DASH player required.

REDUNDANT SYNCHRONOUS SERVER BLOCK DIAGRAM



Full Redundant Synchronous Streaming Audio Encoder/Server Configuration

- Provides FULL Encoder/Server Redundancy
- Completely Seamless Switching
- One Encoder/Server shutdown/failure can occur without any Player Clients aware or glitching
- Eliminates Long-Term Audio Drift
- Prevents Differential Offset
- Uses any Audio Source Format
- Optional further improvements can be made by locking Player Client Audio Clock to GPS

IETF HLS/MPEG-DASH
Uses fully compliant Apple IETF HLS, and ISO MPEG-DASH, the HTTP segmented streaming protocols. Overcomes obstacles and limitations of ICY, FLV and RTMP streaming. More robust than ICY, FLV or RTMP delivery, especially on crowded mobile networks. TTL and Keep-Alive problems associated with legacy streaming protocols are eliminated. Network traffic appears to player clients as HTTP-delivered downloadable files on standard and/or custom ports. This makes it easy to penetrate most firewalls, relieving streaming support issues and creating a much better user experience. Although any port can be used for either encoder or player HLS/DASH connections, typical common HTTP/HTTPS port 80/443 configurations are used to eliminate any firewall issues for maximum player coverage.

Optimized for Audio Streaming Delivery
HLS/DASH Audio-Only Elementary Stream - ES ADTS and CMAF fMP4 supports complete in-band, standards-based ID3v2.4, audio frame-accurate, realtime synchronous metadata.
This is much more efficient than Transport Stream – TS for audio-only delivery.
TS does not include a PAD/metadata standard, making it a non-starter for professional audio content providers.
ES ADTS and CMAF ISO BMFF fMP4 are specified for Audio-Only HLS/DASH. It is also much easier for content and/or ad insertion implementations. TS is NOT recommended for Audio-Only because of high overhead.

AAC/HE-AAC/xHE-AAC Audio Codecs – Stereo/Surround
Fraunhofer IIS
Uses Genuine Fraunhofer AAC/HE-AAC(formerly aacPlus) MPEG-4/xHE-AAC (the latest MPEG-D audio standard codec) commercial floating-point implementation for the ultimate wideband audio quality. This is one of the best-sounding AAC encoders available. No transcoding from other coded formats are used, which compromises audio quality.

AAC/HE-AAC/xHE-AAC replaces outdated, primitive MP3. It is the same core audio codec technology used in the number one on-line music provider, Apple iTunes Music Store. It is less expensive to stream and receive, is more efficient and reliable, and sounds better than MP3 based on double-blind listening tests by credible professional and academic organizations. AAC higher coding efficiencies reduce data usage and listening fatigue. HE-AAC slashes streaming costs by up to 75% while providing higher reliability and quality, especially on today's crowded, bandwidth-limited mobile networks. AAC provides higher quality at lower bitrates than MP3 at higher bitrates.

Surround streams are Multi-Channel AAC/HE-AAC. Surround decoders are included in Windows and macOS. No Dolby decoders are necessary, eliminating additional licensing, development, and/or client decoder installation.

Eliminates developer confusion over ADTS and RAW AAC bitstreams.

AAC family codecs maximize your streaming investment by ensuring the widest software and device coverage to reach the widest audience. Player clients can leverage operating system AAC codecs without any additional licensing. Mobile devices leverage AAC DSP-based codecs to improve battery life. AAC codecs replace legacy MP3 in mobile devices and operating systems. It is a simple matter of economics: MP3 is only useful to reach older generation players that do not include AAC, so how much is it worth to cover the bottom end of your audience with MP3?

Open-source codecs have serious disadvantages. They suffer from lower efficiency and very little native commercial software and device support, where all the numbers matter. Furthermore, they are not part of the HLS specifications. AAC family codecs are the clear winner here.

Adaptive Multiple Bitrates
Allows players to automatically adjust to varying network conditions.
Synchronous adaptive multiple bitrate streaming keeps audio in sync when switching.
No transcoding providing the best possible audio quality at all bitrates.
A fully compliant HLS/DASH player is required.

Segmented Encoder Output
StreamS HLSdirect™/DASHdirect™ provides server-independent, encoder-generated segments with StreamS

PrecisionPAD™ and a single-segmented bitstream with synchronous audio and metadata.
Elementary Stream - ES ADTS and MPEG-DASH Compatible Fragmented MP4 – CMAF fMP4 ISO BMFF are supported.
Segmented encoder-to-server connections allow simple web servers and cloud storage to stream content, and provide robust content ingest while surviving poor network conditions. There is no need for specialized, dedicated and expensive streaming servers.
Server-generated HLS/DASH is less desirable. It requires a continuous encoder-to-server connection and suffers from expense, reliability and metadata latency issues.
Eliminates the need for unreliable, uncompressed, high bitrate encoder-to-server or cloud connections.

HLS/DASH was engineered for the real Int**e**rnet, not an Intr**a**net. The Internet was never designed to stream data continuously. ICY, FLV and RTP streams require a continuous encoder-to-server connection, as well as reconnect algorithms, which are never seamless. If an ICY, FLV or RTP encoder must reconnect, all player clients will usually disconnect from the server, then requiring another reconnect, which again, is not seamless. This can also inflict metadata latency. By passing encoder-generated segments to the server using TPC/IP, StreamS HLSdirect™/DASHdirect™ encoder-to-server connections overcome these limitations.

HLS/DASH segments can be archived on the server if enough data storage is allocated. Because the segments contain metadata, they can be used for playback that is identical to the original live stream and can use the same client players. This is also perfect for metadata-rich podcasts. No more need for two different file and stream formats for live and file on-demand content. These segments also qualify for legal affidavit stream content verification, and can be used to generate accurate stream reports, ensuring correct performance liabilities.

Encoder Reliability
HLS/DASH segments provide intrinsic Keep-Alive. Each segment represents a new request and connection to the server; segmented data is thus transferred to the server reliably, with TCP/IP ensuring seamless, gapless retries caused by poor network connectivity. Server switching and restarts can occur without any content interruption.

Timestamps
HLS/DASH segments contain the required compliant precision audio frame timestamp, allowing HLS player clients to stay on time without buffer overrun or underrun, and synchronize multiple redundant streams/encoders. Using this feature requires fully complaint HLS/DASH Players that supports timestamps.

Player Compatibility
Reach more players for more coverage and opportunity without expensive poor proprietary players.
No longer constrained to custom player solutions with limited performance and compatibility.

Solves the cross-browser, cross-platform AAC Player problem using one single codebase HTML5-MSE browser player, leveraging the operating system native AAC codecs for both stereo and surround. Plays in all modern browsers with no special players or software to download and install. HTML5-MSE players can be customized by content providers with graphics and metadata options, and modified or updated at any time.

All current version Apple macOS and iOS Devices, Microsoft Windows 10 Edge Browser, and select Android Devices, have native support for HLS and AAC/HE-AAC. HTML5-MSE browsers such as Chrome and Firefox can use JavaScript HLS/DASH players, of which there are now several available, and are also Flash-free. Optionally, legacy operating systems and browsers can use custom Adobe Flash Players. This results in 100% browser coverage without expensive, obsolete MP3 fallbacks, further reducing costs. iOS and Android Player Apps cover mobile devices. This gives your audience the best possible user experience everywhere.

Following the IETF HLS and MPEG DASH standards eliminates the need for complete custom player protocol development and custom proprietary metadata, which would otherwise result in not only added expense, but many player compatibility issues. To reach the huge number of player devices and smart speakers now available, standards *must* be followed. You are now streaming to players and devices that are costing consumers well over US$1,000.00, including automotive systems. They expect your content to work and to deliver quality commensurate with other high-quality sources available to them.

Player Reliability

HLS/DASH segments provide intrinsic Keep-Alive. Each segment represents a new request and connection to the server from the player clients; segmented data is thus transferred reliably, with TCP/IP ensuring retries caused by poor network connectivity.

HLS/DASH segments also provide for an intrinsic large buffer without requiring any server fast start algorithms. HLS was engineered for the real Int**e**rnet, not an Intr**a**net. The Internet was never designed to stream data continuously, so ICY/FLV streams require large buffers on both the server and player, as well as reconnect algorithms, which are never seamless. HLS/DASH overcomes this limitation.

Player Starts Immediately

Since HLS/DASH segments are cached on the server, player clients can download as fast as the network connection allows and will start playing immediately. There is no need to support special streaming server fast-start protocols or be constrained by the ability of the streaming server to fill a buffer before play starts. Metadata is also displayed immediately without multiple network connections, which cause additional metadata latency.

HLS/DASH Players

HLS/DASH Players with standards-based frame/sample-accurate PAD/metadata support.
There is no shortage of player software and hardware.
The *new* normal is now *the* normal.

Audio Software Players

Internet Browsers

HTML5 Players

Mobile Players

Audio Hardware Players

TV Players

FTP / DAV / Cloud RFC Compliant Ingest

StreamS HLSdirect™/DASHdirect™ uses standards-based, RFC Compliant FTP, FTPS, HTTP DAV, HTTP DAVS, HTTP POST/PUT, Amazon AWS S3, Google Cloud, Microsoft Azure BLOB, Limelight, and Akamai MSL 4 for streaming encoder-to-server connection and segment uploads. It provides complete file management for segments and automatically deletes expired segments and files. It eliminates high-overhead restrictive server management scripts and does not require the obsolete, proprietary Adobe Flash RTMP for the streaming encoder-to-server connection, eliminating unnecessary complexity and Flash security risks: it's 100% Flash-free.
Encoder output is segmented HLS/DASH.

Secure Separate Stream Ingest

Ingest is isolated between stream input and output. The ingest URL can be completely different than the stream URL and also allows separate ingest servers to be used.
The encoder supports SSL/TLS connections for all ingest protocols.

HTTPS/SSL/TLS

You can deliver streams to player clients over HTTPS using SSL/TLS for secure connections. All stream network traffic can be concealed using SSL-compliant player clients with public or private keys.

Optional Stream Authentication
Protect your subscription-based streams by using stream players that support secure authentication.

Optional Audio Encryption
You can protect your content.
Encrypt content for subscription-based services by using encryption compliant player clients.
Support for all popular browsers with Apple FairPlay, Google Widevine, and Microsoft PlayReady.

Optional Audio Watermark
An optional watermark generator supports immediate content accountability and traceability while playing when using watermark-compliant player clients.

Encoder Software
Encoder runs as a Windows Service, not a stand-alone user-mode application.
There are no login or scripts required for the encoder to run and hence no associated system security issues.
A Session Builder Application is used to configure the encoder audio and metadata, with an easy to use GUI, and is not necessary to remain running.
Survives Microsoft Windows Remote Desktop audio disconnect flaws common to stand-alone audio applications, which Microsoft refuses to address.
Encoder software is capable of running on local or cloud or centralized virtual systems.

VM Compatible
Virtual machine compatible with:
VMware
VirtualBox
XenServer
Parallels Desktop for Mac
Apple Boot Camp

Standard Audio Device Support

The audio Input uses any Microsoft Windows WDM-compliant audio device, including stereo sound cards, multichannel sound cards, virtual sound drivers for inter-app connections without sound cards, and network audio drivers for audio input without sound cards. This allows any Windows Audio compliant sound devices and applications, such as audio playout and processing, to be direct-connected without proprietary audio protocols or pipe APIs and without physical audio cables for those applications not requiring live audio capture. This includes Axia Livewire, RAVENNA, Lavo R3LAY, Audinate Dante, and Wheatstone WheatNet. Qualified multichannel sound cards and network audio drivers can be used for multiple stereo 2.0 and/or multichannel 5.1/7.1 input sources.

Precision Audio Capture

32-bit Audio Capture Engine to feed the high quality commercial floating-point AAC Encoders.
Any codec produced overshoot will be recovered by a floating-point AAC Decoder for distortion-free audio.
Multi-client audio device drivers allow multiple encoders to be run from single audio devices.
Multi-bitrate encoders are synchronous from one single audio capture.
Audio clock sync achieves no drift with StreamSync™ and compatible Input or Output Loopback Capture Devices.

True Realtime PAD/Metadata

StreamS PrecisionPAD™ provides compliant, extensible, in-band, on-time, synchronous, frame-accurate ES ADTS ID3 and time-accurate CMAF fMP4 emsg ID3 PAD/metadata. Standards-based HLS specified, in-line ID3v2.4 UTF-8 supports extended and international characters. This is unachievable with ICY, contrary to popular belief, as ICY metadata is asynchronous. PAD/metadata in .m3u8 manifest/playlist files is not used, as it is not HLS complaint, causes huge latency issues, and does not display in compliant players. HLS/DASH compliant Players receive and parse this metadata in realtime without expensive archaic asynchronous polling commonly used for ICY players.

The encoder accepts several extensible TXT and XML standards-based metadata formats over TCP or UDP without complicated scripting for the encoder or server, which can cause additional PAD/metadata latency.

StreamS PrecisionPAD™ provides an accurate listener experience with on-time PAD/metadata display and allows precise content/ad insertion timing and control using hidden extensible control fields. With accurate PAD/metadata, programming continuity can be maintained without embarrassing gaps and cuts that are unacceptable for professional, fast paced, high-energy program presentations. Other methods such as archaic silence sense and audible tone methods are obsolete and simply not needed, nor desired, since they affect program content in adverse ways.

Used in conjunction with properly designed playout software, encoder-generated HLS/DASH precisely synchronizes PAD/metadata to program element boundaries or timing, as opposed to ICY SHOUTcast/Icecast2 protocols, which can only refresh metadata at fixed intervals that may be unsynchronized with these boundaries or timings.

Secure PAD/metadata updates are assembled within the encoder and not the server, enhancing security. ICY SHOUTcast/Icecast2 protocols are less secure because metadata is assembled at the server instead of the encoder, making encoder/server communication vulnerable to attacks. Extensible PAD/metadata and user-definable PAD/metadata fields for complete custom applications, including content/ad insertion and other signaling, are supported without legacy limitations. Extensible PAD/metadata allows a given player to display only the fields desired by its provider without embarrassing garbage control characters. PAD/metadata is not cached on the server, reducing server load. The encoder also accepts externally generated ID3v2.4 frames.

PAD/metadata is DMCA Compliant and can be used to generate accurate stream reports, ensuring accurate performance liabilities.

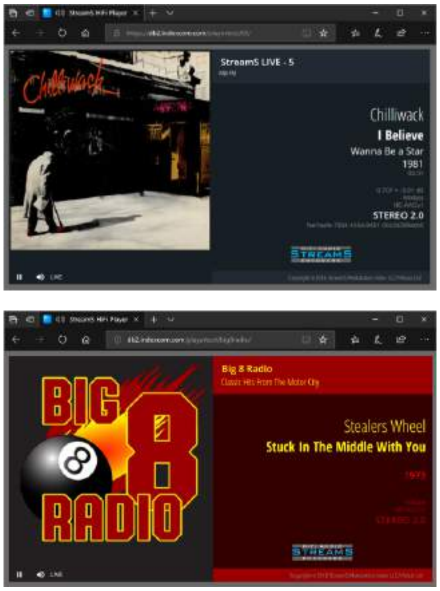
Apple iTunes Radio Compliant

Apple HLS and PAD/metadata are 100% Apple iTunes Player and iTunes Radio compliant with artist, title, album, and images. Other extensible PAD/metadata fields may be included without limiting player compatibility or displaying in the wrong fields.

HLS/DASH Players

HTML5-MSE Players

Metadata is no longer limited to simple Artist – Title. Using true extensible PAD/metadata, many metadata fields can be included without player compatibility issues, including non-visible control.



HTML5-MSE Standard Player
Microsoft Windows 10 Edge Browser

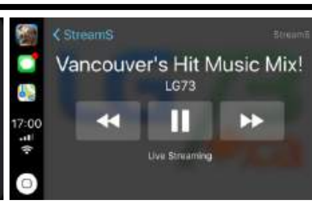
HTML5-MSE Diagnostic Player
Microsoft Windows 10 Edge Browser

Apple CarPlay Player – Dashboard Display

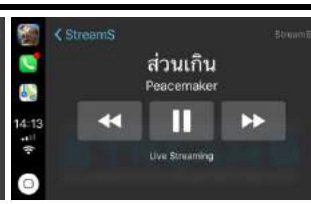
Since the typical factory automobile dashboard display is large, for minimum distraction and safety reasons the program-associated images and album art are constrained to a blurred background on the dashboard. However, the text is very prominently displayed, so it is vital to make sure this looks good.

The connected car and digital dashboard are here now. Reach it reliably and professional with StreamS HLSdirect™.

Title / Artist – English
Content Image Background

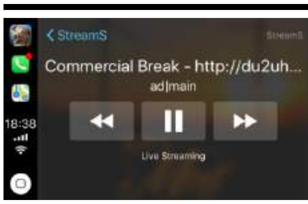


Title / Artist – English
Content Image Background



Title / Artist – Thai
Any Language – Any Character Set

Title / Artist – Chinese
Any Language – Any Character Set

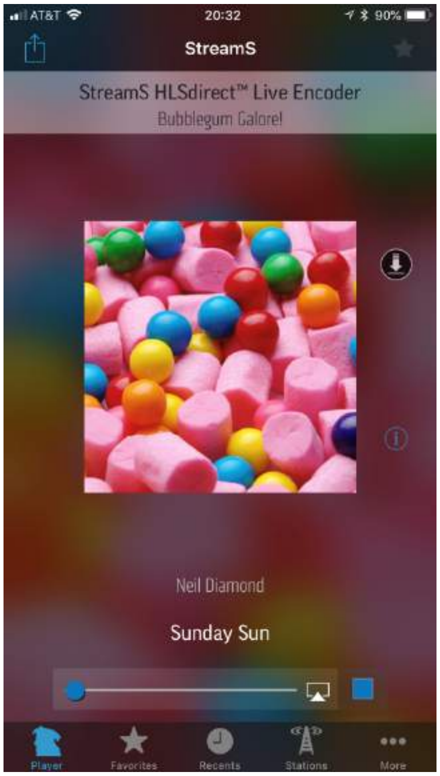


What NOT to do!
Don't send control characters in visible fields.
Your listeners have no idea what this is, and when it shows up on the big display, it looks unprofessional and embarrassing.

Works with Apple CarPlay

The connected car and digital dashboard of today feature interfaces such as Apple CarPlay.
With Apps such as StreamS HiFi Radio, stunning streaming audio quality can be enjoyed on the go.

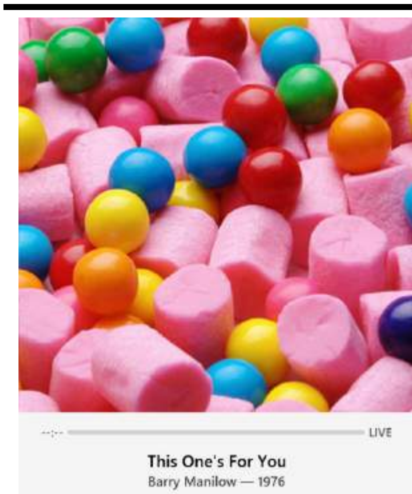
Mobile Players

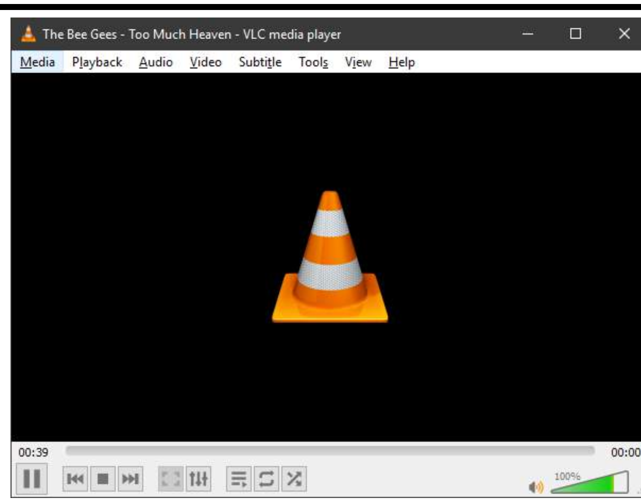


Apple iOS – StreamS HiFi Radio App
App REQUIED – iOS does not support HTLM5-MSE

Android Chrome – StreamS HTML5-MSE Player
App NOT REQUIED to Play and Display Metadata

Application Players





Apple iTunes Mini Player
Metadata and Content Images displayed

VLC Player
Metadata displayed in Application Title Bar

Frame-Accurate Audio Frames
HLS segments have frame-accurate lengths to prevent fragmented frame audio decoder errors.

Audio Player Gain Metadata
In conjunction with compliant player clients, this metadata controls the player client audio level to achieve consistent audio loudness between program elements, which is especially important for server-side content/ad insertion applications. This feature facilitates EBU R128/BS.1770 compliance by using the appropriate analysis software and sending the resulting Target Loudness Level PAD/metadata through the streaming encoder or in the PAD/metadata of the server-side content/ad insertions to compliant player clients. This metadata is carried in the stream metadata ID3v2.4 frame to accommodate the majority of AAC-LC/HE-AAC decoders that do not support codec metadata. DRC cannot be easily supported with this method.

Audio Codec Metadata
In conjunction with compliant player clients, this metadata controls the player client audio level and/or DRC to achieve consistent audio loudness between program elements, which is especially important for server-side content/ad insertion applications. This feature facilitates EBU R128/BS.1770 compliance by using the appropriate analysis software in the encoder and sending the resulting Target Loudness Level and/or DRC codec metadata in the streaming encoder or in the codec metadata of the server-side content/ad insertions to compliant xHE-AAC decoder player clients. This metadata is carried in the audio codec metadata by the audio encoder to accommodate compliant xHE-AAC decoders with Normalization and DRC support.

SNMP Monitoring – (StreamS Encoder Systems Only)
You can monitor encoder parameters and status using network SNMP. Status alarms are provided through the TCP/IP control interface.

Logging
All encoder events are displayed in the StreamS Encoder Session Builder and also logged as Windows events viewable through the standard Windows Event Viewer.

HLS Analytics
StreamS MetriX™ analytics package and services offer *accurate* HLS/DASH listener statistics.

And Last but certainly NOT Least…
Orban Optimod™ Native Audio Processing

Professional audio processing should be a part of any audio stream expected to attract and hold audiences, and to compete with other streams. With smart mobile devices and speakers, this is now more important than ever. Many of these devices provide mono sound only, and Orban Optimod™ 1600PCn audio processing software provides exclusive, unique processing that optimizes the sound through single-speaker devices.

Legendary, award winning Orban Optimod™ processing is used worldwide on more broadcasts and netcasts than all others combined. It prevents the biggest listener irritants: audio inconsistency and volume level variations, and does it all in a musical and natural way without introducing obvious artifacts that can ruin the integrity of your content. There is nothing worse than cheap telephonic audio processing pumping away, insulting the original artist's intent.

There are major differences between audio processors; processing is not a commodity! Processing is both a scientific and artistic endeavor. One without the other does not get you there—this is reason Orban processing gets results and is the success that it is.

Audio processing completes your stream.

Quick Summary

Reduced Cost
Increased Reliability
Increased Quality
Increased Coverage
Easier to Develop, Deploy, and Maintain
StreamS HLSdirect™/DASHdirect™ is the only FULLY Compliant Streaming Audio Encoder available
Not using HLS/DASH segmented streaming is a disservice to both content providers and audiences.

https://www.indexcom.com/products/encoder/

Summary

Video and audio streaming technology have evolved. Most audio streamers and content distribution networks have not updated and are still streaming using archaic equipment, protocols, and codecs.

Until recently, little development has been done for audio streaming while video has moved forward. Many CDNs refuse to invest in new tech that decreases their revenue, even though it may benefit their clients. They have done very little to improve performance using new protocols. Moving from MP3 to AAC 20 years ago was similar. Many CDNs even use incomplete or proprietary protocols in an attempt to lock their customers and make it difficult to move to a competitive provider. Many provide free, low-performance commodity and toy encoders to attract and motivate clients. This stifles streaming technological advancement. Ultimately your audience suffers and you lose audience share. They now have expensive mobile devices and connected car digital dashboards that they expect to produce high quality, reliable results. You have one chance to make a first impression, and you want your audience to return.

Adapt or die.

The Need

Streaming audio, especially terrestrial radio streaming, has needed guidance. Using limited-feature legacy streaming protocols no longer works reliably for today's advanced feature-rich players and devices.

Many terrestrial radio content providers have viewed streaming as an afterthought and/or a secondary service. However, traditional terrestrial broadcast is now waning. Media delivery is evolving and streaming media delivery requires the same attention to detail as terrestrial. Streaming is where your new audience is.

Streaming is now a mainstream medium, present in connected car digital dashboards and replacing portable radios with mobile devices and smart speakers. To implement new IT broadcast technologies, many terrestrial broadcasters have offloaded streaming administration to companies that have little to no broadcast experience, or to companies they have acquired that use amateur or obsolete streaming technologies with non-compliant, incomplete implementations. Even some of the larger content providers and content delivery networks are guilty of this. These proprietary streams impede your coverage and make it difficult for content providers to reach as many streaming devices as possible. The variations create a moving target for player developers, who find it difficult to keep up. Proprietary, non-standard streaming technology ultimately costs you audience share because so much works incorrectly. This situation is pretty much "the blind leading the blind." It creates a "deep digital-divide," where media delivery suffers and your audience is deprived.

Standards-based streaming solves this problem. It is very important that content providers understand the technologies and implementations they deploy. Most potential listeners are not motivated to research what player to use to consume non-standard streams. Like AM, FM, and TV, they want it to "just work."