LIBXAAC Encoder

Getting Started Guide

Document Number	LIBXAAC-Enc-GSG
Version	1.0
Date	June 28, 2023

Ittiam Systems Confidential

Ittiam Systems (P) Ltd,
The Consulate, 1 Richmond Road,
Bangalore 560 025, India

Notice

Ittiam Systems reserves the right to make changes to its products or discontinue any of its products or offerings without notice.

Ittiam warrants the performance of its products to the specifications applicable at the time of sale in accordance with Ittiam's standard warranty.

Revision History

Version	Date	Changes
1.0	June 28, 2023	Original

Contents

Tables

Figures

1. Introduction

1.1 Motivation

Extended HE-AAC, the latest innovation member of the MPEG AAC codec family, is ideally suited for adaptive bit rate streaming and digital radio applications. Extended HE-AAC bridges the gap between speech and audio coding and ensures consistent high-quality audio for all signal types, including speech, music, and mixed material. It is the required audio codec for DRM (Digital Radio Mondiale). When it comes to coding, the codec is incredibly effective, generating high-quality audio for music and speech at bitrates as low as 6 kbit/s for mono and 12 kbit/s for stereo services. By switching to extremely low bitrate streams, Extended HE-AAC streaming apps and streaming radio players can provide uninterrupted playback even during very congested network conditions.

As the Extended High Efficiency AAC Profile is a logical evolution of the MPEG Audio's popular AAC Family profiles, the codec supports AAC-LC, HE-AACv1 (AAC+) and HE-AACv2 (eAAC+) audio object type encoding. The bitrate that was saved with AAC family tools can be used to enhance video quality. Extended HE-AAC is a well-liked option for a number of applications since it is a strong and effective audio codec that provides high-quality audio at low bitrates.

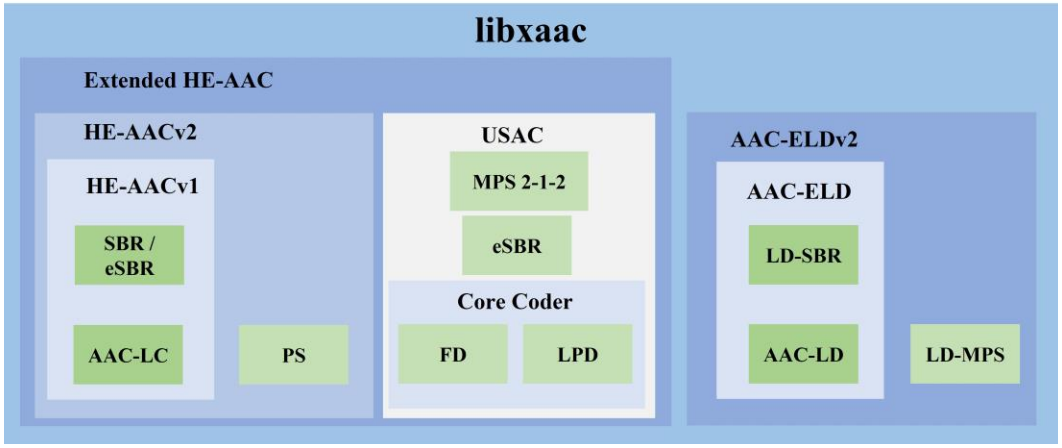

Figure 1-1: Block Diagram of libxaac

One of the key features of libxaac encoder (refer to above image) is that it has support for AAC-LD (Low Delay), AAC-ELD (Enhanced Low Delay), and AAC-ELDv2 (Enhanced Low Delay version 2) modes. AAC-LD mode provides low latency encoding, making it suitable for applications such as interactive communication and live audio streaming. It helps to reduce the delay in the encoding process to improve the real-time performance of the system. AAC-ELD mode improves the low-delay performance of HE-AAC by reducing the coding delay while maintaining high audio quality. It was observed that minimum delay it can achieve is 15ms. In order to achieve low delay coding scheme and low bitrate, it uses the Low Delay SBR tool. AAC-ELDv2 is the most advanced version of AAC-based low delay coding. It provides an enhanced version of AAC-ELD, which provides even lower coding delay and higher audio quality.

Overall, libxaac encoder, with support for AAC-LD, AAC-ELD, and AAC-ELDv2 modes, is a versatile audio coding technology that can be used for a wide range of applications, such as broadcasting, streaming, and teleconferencing which requires high-quality audio compression with minimal delay.

MPEG-D USAC (Unified Speech and Audio Coding) support for libxaac encoder will be coming soon, which will help in improved audio quality at reduced bitrates. USAC technology will provide efficient and high quality compression of speech and audio signals, making it a valuable addition to our libxaac capabilities and features.

1.2 Scope

This document will provide information to developers in terms of the following:

- Running the sample Application (**Chapter 2**)
  - This chapter gives a complete overview of the procedure to run the sample application provided.
- Testing the encoder (**Chapter 3.1**)
  - This chapter describes the tools for validation.

1.3 Glossary

Term	Explanation
AAC	Advanced Audio Coding
ADTS	Audio Data Transport Stream
MPEG	Motion Pictures Expert Group
HE-AAC	High Efficiency Advanced Audio Coding
LC	Low Complexity
LD	Low Delay
ELD	Enhanced Low Delay
MPS	MPEG Surround

2. Running the Sample Application

2.1 Parameter file

The application reads testvector names from the parameter file `paramfilesimple.txt` in `test\encoder`.

The syntax for writing `paramfilesimple.txt` file is:

```
@Input_path <input folder path>
@Output_path <output folder path>
@Start
<command line 1>
<command line 2>
....
@Stop
```

Command	Explanation
@Start	All commands following this command will be executed
@Stop	All commands until this command will be executed
@Input_path	Path specified in this command will be appended to the path and the input file name specified by all -ifile: commands until the next occurrence of this command.
@Output_path	Path specified in this command will be appended to the path and the output file name specified by all -ofile: commands until the next occurrence of this command.
<command line 1>	For the complete list of configurable parameters, please refer to the table below.

Table 2-1 Parameter file commands

A sample syntax for the command line is as follows:

```
-ifile:<input_wav_file> -ofile:<output_aac_file> -br:<bit_rate> -esbr:< 0/1 (eSBR enable/disable) > -aot:<2 (AAC-LC)/23 (AAC-LD)/39 (AAC-ELD) /5 (HE-AACv1)/ 29 (HE-AACv2) > -adts:<0/1 (ADTS header enable/disable) >
```

Note that the switch `-adts` along with `-aot` determine the format of the encoded bitstream (as shown in Table 2-2) and are not mandatory to be specified.

-adts	-aot: 2 / 5 / 29	-aot: 23 / 39
0	Raw (.es) + Meta (.txt)	Raw (.es) + Meta (.txt)
1	ADTS (.aac)	Raw (.es) + Meta (.txt)

Table 2-2 Bitstream header formats supported

A few examples are shown below:

`-ifile:HarryPotter.wav -ofile:HarryPotter_128.es -br:128000 –aot:23 -adts:0` → This generates an AAC-LD stream in **Raw(.es) + Meta(.txt)** format

`-ifile:HarryPotter.wav -ofile:HarryPotter_64.aac -br:64000 –aot:5 -adts:1` → This generates an HE-AACv1 stream in **ADTS** format

2.2 Configurable Parameters

The complete list of parameters specified through the command line that can be used with the encoder is listed below. For the valid range of each parameter, please refer to the API Document.

Parameter	Explanation
SPECIFY INPUT/OUTPUT FILES	
-ifile:	Provide the name and path for the input file to the encoder
-ofile:	Provide the name and path for the output file of the encoder
ENCODING OPTIONS	
-br:	Provide bit rate to be used for encoding in bps (bits per second). Default is 48000.
-aot:	Flag to set encoding profile. The valid values for this option are 2, 5, 23, 29 and 39. Set to 2 (AAC-LC) by default. 1. -aot:2 – AAC-LC encoding profile (Default value) 2. –aot:5 – HE-AACv1 (Legacy SBR) encoding profile 3. –aot:23 – AAC-LD encoding profile

	4. -aot:29 – HE-AACv2 encoding profile 5. -aot:39 – AAC-ELD encoding profile
-esbr:	Flag to enable eSBR for HE-AACv1 encoding profile. The valid values for this option are 0 or 1: 1. -esbr:1 – Use eSBR 2. -esbr:0 – Use legacy SBR (Default value)
-mps:	Flag to enable MPS encoding when AOT is set to AAC-ELDv2. The valid values for this option are 0 or 1: 1. -mps:1 – Use MPS encoding 2. -mps:0 – Skip MPS encoding (Default value)
-tns:	Flag to enable temporal noise shaping. The valid values for this option are 0 or 1. Default value is 1.
-adts:	Flag to enable ADTS header. The valid values for this option are 0 and 1 when -aot option is set to 2 or 5 or 29. Default is 0.
-full_bandwidth:	Enable use of full bandwidth of input. The valid values for this option are 0 and 1. Default is 0.
-max_out_buffer_per_ch:	Bitreservoir size. The valid values for this option are between -1 and 6144. -1 to omit use of bit reservoir. Default value is 384.
-tree_cfg:	Flag to denote tree configuration for MPS. The valid values for this option are 1. -tree_cfg:0 for 212 configuration 2. -tree_cfg:1 for 5151 configuration 3. -tree_cfg:2 for 5152 configuration 4. -tree_cfg:3 for 525 configuration. Default value is 0 for stereo input and 1 for 6-channel input
-framesize:	Flag to denote frame-size (in samples) used by the core coder for AAC-LC / HE-AACv1 / HE-AACv2 and AAC-LD / AAC-ELD / AAC-ELDv2 profiles. The valid values for this option are 960 or 1024 for AAC-LC / HE-AACv1 / HE-AACv2 and 480 or 512 for AAC-LD / AAC- ELD / AAC-ELDv2. Default value is 1024 for AAC- LC / HE-AACv1 / HE-AACv2 and 512 for AAC-LD / AAC-ELD / AAC-ELDv2.
OUTPUT FILE FORMATS	
	Note: 1. By default, .es format is used

Table 2-3 List of Configurable Parameters

3. Test Procedure

This chapter describes the procedure for testing quality of libxaac encoder.

3.1 Testing the Encoder

libxaac decoder is used for validating the encoded bitstream and decoding. The input and output filenames need to be changed appropriately.

The usage of the decoder is described below:

1. Decode the encoder generated file(s) encoded in ADTS format using the libxaac decoder using the following command line:

   ```
   xaacdec.exe -ifile:<adts_input_file>.aac -
   ofile:<adts_output_file>.wav
   ```

2. Decode the encoder generated file(s) encoded in raw format using the libxaac decoder using the following command line:

   ```
   xaacdec.exe -ifile:<raw_input>.es  -ofile:<output>.wav -
   imeta:<meta_input>.txt -mp4:1
   ```

4. Reference

[1] Build Procedure