

# **FAAD - ISO/MPEG 2/4 AAC Decoder Library V1.0**

Freeware Advanced Audio Coding  
(<http://www.audiocoding.com/>)

© 2001 AudioCoding.com. All rights reserved.

# 1 Contents

FAAD - ISO/MPEG 2/4 AAC Decoder Library V1.0 .....	1
1     Contents.....	2
2     Scope .....	3
3     Interface description.....	4
4     Usage .....	5
4.1   Calling sequence .....	5
5     Function reference .....	6
5.1   Initialization / De-initialization .....	6
5.1.1   faacDecOpen() .....	6
5.1.2   faacDecInit().....	6
5.1.3   faacDecGetProgConfig() .....	6
5.1.4   faacDecClose() .....	7
5.2   Decoder configuration.....	7
5.2.1   faacDecGetCurrentConfiguration() .....	7
5.2.2   faacDecSetConfiguration() .....	8
5.3   Decoding functions .....	8
5.3.1   faacDecDecode() .....	8
6     Data structures reference .....	10
6.1   faacDecConfiguration.....	10

## **2 Scope**

This document describes the interface and usage of the

### **FAAD - ISO/MPEG 2/4 AAC Decoder Library**

Developed for the Freeware Advanced Audio Coding project.

### 3 Interface description

The ISO/MPEG 2/4 AAC Decoder Library provides a high-level interface for decoding MPEG2 and MPEG4 ISO AAC files. The following header file is provided for usage in C/C++ programs:

**faad.h**: function prototypes

The decoder core resides in a statically linkable library called libfaad.lib (*Microsoft Windows*) or libfaad.a (*UNIX*). There are various example programs that show how to use the library.

## 4 Usage

### 4.1 Calling sequence

For decoding AAC bitstreams the following calling sequence is mandatory:

- Call **faacDecOpen()** for every decoder instance you need.
- To set decoder options, call **faacDecGetCurrentConfiguration()**, change the parameters in the structure accessible by the returned pointer and then call **faacDecSetConfiguration()**.
- Call **faacDecInit()** to get the number of channels and sampling rate from the AAC file.
- As long as there is still data left to decode, call **faacDecDecode()** to decode the data. The decoder returns the audio data in a client-supplied buffer.
- If **faacDecDecode()** returns **FAAC\_OK\_CHUPDATE** a call to **faacDecGetProgConfig()** should be made to get the new channel configuration. If needed some data should be reallocated.
- After all the data has been decoded by **faacDecDecode()**, call **faacDecClose()** to close the decoder instance.

## 5 Function reference

### 5.1 Initialization / De-initialization

#### 5.1.1 faacDecOpen()

*Prototype*

```
faacDecHandle FAADAPI faacDecOpen();
```

*Description*

Open and initialize one instance of the decoder.

*Return value*

An initialized encoder handle. If anything goes wrong NULL is returned.

#### 5.1.2 faacDecInit()

*Prototype*

```
int FAADAPI faacDecInit  
(  
    faacDecHandle hDecoder,  
    unsigned char *inputBuffer,  
    unsigned long *sampleRate,  
    unsigned long *numChannels  
);
```

*Description*

Initialize the file that is going to be decoded.

*Parameters*

- hDecoder  
A decoder handle returned by **faacDecOpen()**
- inputBuffer  
Contains data from the file that's going to be decoded. This is meant to get the first header from the file to determine the file's sampling rate and the number of channels.
- sampleRate  
Will be filled with the sampling rate of the AAC file.
- numChannels  
This will be filled with the number of channels in the AAC file.

*Return value*

The number of bytes that the calling application has to shift forward in the input buffer.

#### 5.1.3 faacDecGetProgConfig()

*Prototype*

```
int FAADAPI faacDecGetProgConfig  
(  
    faacDecHandle hDecoder,
```

```

    faacProgConfig *progConfig
);

```

*Description*

Gives the updated channel configuration after decoding a frame. This function only has to be called when **faacDecDecode()** returns **FAAD\_OK\_CHUPDATE**, giving the calling program the opportunity to re-allocate **sampleBuffer**, if needed, so that there is enough memory available in the next call to **faacDecDecode()**.

*Parameters*

- **hDecoder**  
A decoder handle.
- **progConfig**  
The program configuration of the last frame. Not used in the current version of the library.

*Return value*

Number of channels in the previous frame.

#### 5.1.4 faacDecClose()

*Prototype*

```

void FAADAPI faacDecClose
(
    faacDecHandle hDecoder
);

```

*Description*

Closes a decoder instance.

*Parameters*

- **hDecoder**  
A decoder handle returned by **faacDecOpen()**.

## 5.2 Decoder configuration

#### 5.2.1 faacDecGetCurrentConfiguration()

*Prototype*

```

faacDecConfigurationPtr FAADAPI
faacDecGetCurrentConfiguration
(
    faacDecHandle hDecoder
);

```

*Description*

Get a pointer to a structure describing the current decoder configuration. You may change this structure and feed it into **faacDecSetConfiguration()**.

## 5.2.2 faacDecSetConfiguration()

### *Prototype*

```
int FAADAPI faacDecSetConfiguration
(
    faacDecHandle hDecoder,
    faacDecConfigurationPtr config
);
```

### *Description*

Set a new decoder configuration. See `faacDecGetCurrentConfiguration()`.

## 5.3 Decoding functions

### 5.3.1 faacDecDecode()

#### *Prototype*

```
int FAADAPI faacDecDecode
(
    faacDecHandle hDecoder,
    unsigned char *inputBuffer,
    unsigned long *bytesConsumed,
    short *sampleBuffer
);
```

#### *Description*

Decode one audio frame from an AAC file.

#### *Parameters*

- `hDecoder`  
A decoder handle.
- `inputBuffer`  
Contains data from the AAC file to be decoded. This should always be at least 768 bytes times the number of channels in the AAC file.
- `bytesConsumed`  
The number of bytes consumed from the input buffer by the decoder. This value should be used by the calling program to shift forward in the input file.
- `sampleBuffer`  
Contains the decoded audio samples in standard PCM order. Supplied by the program calling this function.

#### *Return value*

If this function returns `FAAD_OK` the frame is decoded correctly. If it returns `FAAD_OK_CHUPDATE`, no error occurred but the number of channels in the frame is different than the number of channels before (either difference from initialization or from a previous



frame). This gives the calling program the opportunity to re-allocate data (sampleBuffer), if necessary.

If it returns FAAD\_ERROR, an error occurred in this frame but decoding can be continued, otherwise if it returns FAAD\_FATAL\_ERROR, an error occurred and decoding has to be stopped.

## 6 Data structures reference

### 6.1 *faacDecConfiguration*

#### *Definition*

```
typedef struct faacDecConfiguration
{
}
faacDecConfiguration, *faacDecConfigurationPtr;
```

#### *Description*

Through this structure you can change the decoder configuration.

#### *Fields*

None, yet. Currently there are no decoder options to set. This structure and accompanying functions are only implemented so that the library will stay backwards compatible in function definitions.