Lame-enc DLL

version 1.16

(Lame engine version 3.87)

Programmers Manual

Homepage: http://www.cdex.n3.net
E-mail: mailto: cdex@softwarecenter.net

Distribution

People and companies who wants to distribute lame_enc.dll with their commercial products are free to do so as far as I'm concerned, but should be aware that lame_enc.dll might infringe certain MP3 related software patents held by Fraunhofer IIS in certain countries.

Disclaimer

lame_enc.dll and this manual is distributed 'as is' with no warranty of any kind. The Author is not to be held responsible for the result of any use or misuse of this product.

Current Bugs and Limitations

Although the interface is designed to be able to handle multiple parallel streams it can't be done yet due to limitations in the engine, only one stream is allowed.

Future Compatibility

This interface should be compatible with all future versions of lame_enc.DLL without any need to recompile your programs. You should therefore **not** check the version number upon start and prevent users from running your program with a later version of lame_enc.DLL.

How to use the DLL

1. Fill in a BE_CONFIG structure and send it to beInitStream(). Make sure that BE_ERR_SUCCESSFUL is returned.

2. Reserve at least the amount of memory returned in dwBufferSize as your output buffer.

3. Call beEncodeChunk() until you've encoded everything you want.

4. Call beDeinitStream() to make sure that all encoded data is flushed out before closing the stream.

5. Close the stream using beCloseStream()

6. And optionally call the beWriteVBRHeader() functions, to insert a Xing MP3 Header

A handy feature is the available Lame_enc.dll debug option, which will dump the important lame internal settings to a text file.

Return Values

See the header-file for a complete list of function return values. All functions should return BE_ERR_SUCCESSFUL unless something went wrong.

Type definitions

The DLL is by default compiled with the MS Visual C/C++ compiler, which has the following type definitions:

Type	Description
CHAR	signed char (8 bits)
BYTE	unsigned char (8 bits)
SHORT	signed short (16 bits)
WORD	unsigned short (16 bits)
INT	signed long (32 bits)
LONG	signed long (32 bits)

BOOL	signed long (32 bits) (YES, 32 bits for a one bit value) TRUE = 0 FALSE=-1
DWORD	unsigned long (32 bits)
FLOAT	floating point (32 bits)
DOUBLE	float point (64 bits)
LPCSTR	const char* (32 bits pointer to zero terminated character string)

Within the lame_enc.dll All the structure elements are one byte alligned (due to backwards compatibility wiht BladEnc.DLL!

The BE_CONFIG Structure

Currently there the BE_CONFIG structure has to varians, the old MP3 config structure that is truly compatible with the old BladeEnc interface, and the new defined LHV1 structure, which can set far more options in the lame encoder

The MP3 BE_CONFIG - structure (OBSOLETE)

This is the old structure as it was originally defined by the BladeEnc.DLL interface. However, I do highly recommend to use the new Lame specific config structure, since it gives you more control over the Lame encoder settings.

These are the members of the BE_CONFIG structure you need to fill in before you call beInitStream():

dwConfig	Specifies what kind of output you want. Since only MP3 currently is supported you must set this to **BE_CONFIG_MP3**
format.mp3.dwSampleRate	Samplerate in Hz for MP3 file. This can be set to either **32000**, **44100** or **48000**.
format.mp3.byMode	Stereomode for MP3 file. This can be either **BE_MP3_MODE_STEREO**, **BE_MP3_MODE_DUALCHANNEL** or **BE_MP3_MODE_MONO.**
format.mp3.bitrate	Bitrate (i.e. size) of MP3 file in kBit/s. Allowed bitrates are: **32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256** and **320**.
format.mp3.bCopyright	If this is set to TRUE the Copyright bit in the MP3 stream will be set.
format.mp3.bCRC	Set this to TRUE in order to enable CRC-checksum in the bitstream.
format.mp3.bOriginal	If this is set to TRUE the Original bit in the MP3 stream will be set.
format.mp3.bPrivate	If this is set to TRUE the Private bit in the MP3 stream will be set.

The LHV1 BE_CONFIG - structure (recommended)

These are the members of the LHV1 BE_CONFIG structure, you need to fill in before you call beInitStream():

dwConfig	Specifies what kind of output you want. Since only MP3 currently is supported you must set this to **BE_CONFIG_LAME**
format.LHV1.dwStructVersion	Indicates the version number of the structure, current version number is 1
format.LHV1.dwStructSize	Specifies the size of the BE_CONFIG structure (currently 331 bytes)
format.LHV1.dwSampleRate	Samplerate in Hz for MP3 file. This can be set to either: **32000**, **44100** or **48000** for MPEG-I **16000**, **22050** or **24000** for MPEG-I **8000**, **11025** or **12000** for MPEG-II.5
format.LHV1.dwReSampleRate	Specifies to which sample rate the input stream has to be resampled, if set to 0, the encoder will decide which ReSample rate to use
format.LHV1.nMode	Stereomode for MP3 file. This can be either **BE_MP3_MODE_STEREO**, **BE_MP3_MODE_JSTEREO, BE_MP3_MODE_DUALCHANNEL** or **BE_MP3_MODE_MONO.**
format.LHV1.dwBitrate	For CBR, this specifies the actual bitrate, for VBR, it specifies the minimum bitrate Allowed bitrates are: **32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256** and **320**.for MPEG-I Allowed bitrates are: **8, 16, 24, 32, 40, 48, 56, 64, 80, 96, 112, 128, 144** and **160**.for MPEG-II
format.LHV1.dwMaxBitrate	For CBR this setting is ignored, when using VBR, it specifies the maximum bitrate
format.LHV1.nPreset	The quality option can be set to one of the following presets: NOPRESET,PHONE,SW,AM,FM,VOICE,RADIO,TAPE,HIFI,CD,STUDIO as defined in the LAME_QUALTIY_PRESET. Keep in mind that the presets can overwrite some of the other settings, since it is called right before the encoder is initialized
format.LHV1.bCopyright	If this is set to TRUE the Copyright bit in the MP3 stream will be set.
format.LHV1.bCRC	Set this to TRUE in order to enable CRC-checksum in the bitstream.
format.LHV1.bOriginal	If this is set to TRUE the Original bit in the MP3 stream will be set.
format.LHV1.bPrivate	If this is set to TRUE the Private bit in the MP3 stream will be set.
format.LHV1.bWriteVBRHeader	Sepecifes if the a XING VBR header should be written or not. When this option is enabled, you have to call the beWriteVBRHeader function when encoding has been completed
format.LHV1.bEnableVBR	Specifies if VBR encoding option shall be used or not, possible values are TRUE/FALSE
format.LHV1.nVBRQuality	Quality option if VBR is enabled (0=highest quality, 9 is lowest quality)

format.LHV1.dwVbrAbr_bps	If the Average Bit Rate is specified, the lame encoder ignores the nVBRQuality settings
format.LHV1.bNoBitRes	Disables the bit-resorvoir and disables the insertion of padded frames
format.mp3.btReserved	For future use, set all elements to zero

---

beInitStream()

Synopsis: BE_ERR beInitStream(PBE_CONFIG *pbeConfig*, PDWORD *dwSamples*, PDWORD *dwBufferSize*, PHBE_STREAM *phbeStream*)

Parameters:

pbeConfig	Pointer at the struct containing encoder settings.
dwSamples	Pointer at double word where number of samples to send to each *beEncodeChunk()* is returned.
dwBufferSize	Pointer at double word where minimum size in bytes of output buffer is returned.
phbeStream	Pointer at integer where Stream handle is returned.

Description: This function is the first to call before starting an encoding stream.

---

beEncodeChunk()

Synopsis: BE_ERR beEncodeChunk(HBE_STREAM *hbeStream*, DWORD *nSamples*, PSHORT *pSamples*, PBYTE *pOutput*, PDWORD *pdwOutput*)

Parameters:

hbeStream	Handle of the stream.
nSamples	Number of samples to be encoded for this call. This should be identical to what is returned by *beInitStream()*, unless you are encoding the last chunk, which might be smaller.
pSamples	Pointer at the 16-bit signed samples to be encoded. These should be in stereo when encoding a stereo MP3 and mono when encoding a mono MP3.

pOutput	Where to write the encoded data. This buffer should be at least of the minimum size returned by *beInitStream()*.
pdwOutput	Where to return number of bytes of encoded data written. The amount of data written might vary from chunk to chunk.

Description: Encodes a chunk of samples. *Please note that if you have set the output to generate mono MP3 files you must feed beEncodeChunk() with mono samples!*

---

beDeinitStream()

Synopsis: BE_ERR beDeinitStream(HBE_STREAM *hbeStream*, PBYTE *pOutput*, PDWORD *pdwOutput*)

Parameters:

hbeStream	Handle of the stream.
pOutput	Where to write the encoded data. This buffer should be at least of the minimum size returned by *beInitStream()*.
pdwOutput	Where to return number of bytes of encoded data written.

Description: This function should be called after encoding the last chunk in order to flush the encoder. It writes any encoded data that still might be left inside the encoder to the output buffer. This function should NOT be called unless you have encoded all of the chunks in your stream.

---

beCloseStream()

Synopsis: BE_ERR beCloseStream(HBE_STREAM *hbeStream*)

Parameters:

hbeStream	Handle of the stream.

Description: Last function to be called when finished encoding a stream. Should unlike *beDeinitStream()* also be called if the encoding is canceled.

---

beVersion()

Synopsis: VOID beVersion(PBE_VERSION *pbeVersion*)

Parameters:

pbeVersion	Pointer at struct where version number, release date and URL for homepage is returned.

Description: Returns information like version numbers (both of the DLL and encoding engine), release date and URL for lame_enc's homepage. All this information should be made available to the user of your product through a dialog box or something similar.

---

beWriteVBRHeader()

Synopsis: VOID beWriteVBRHeader(LPCSTR *pszMP3FileName*)

Parameters:

pszMP3FileName	Const Pointer zero terminated string, that contains the MP3 file name.

Description: Writes a Xing Header in front of the MP3 file. Make sure that the MP3 file is closed, and the the beConfig.format.LHV1.bWriteVBRHeader has been set to TRUE. In addition, it is always save to call beWriteVBRHeader after the encoding has been finished, even when the beConfig.format.LHV1.bWriteVBRHeader is not set to TRUE

Lame_enc.dll debug option

The lame_enc.dll has a built-in debug options, that dumps all the important internal settings to a text file. To enable this feature, create a text file in the Windwos directory which is named lame_enc.ini, and should contain the following two lines

[debug]
WriteLogFile=1

Save this text file, and each time you encode a file, the settings are added to a file name lame_enc.txt, that is located in the same directory as the lame_enc.dll