MP3EncV3.0

Next Generation High-End MPEG Layer-3 Encoding

Fraunhofer Institute for Integrated Circuits
http://www.iis.fhg.de/audio/

25th March 1998

Contents

License agreement

USE OF THE SOFTWARE IS SUBJECT TO THE SOFTWARE LICENSE TERMS SET FORTH BELOW. USING THE SOFTWARE INDICATES YOUR ACCEPTANCE OF THESE TERMS. IF YOU DO NOT ACCEPT THESE TERMS, YOU MUST RETURN OR DELETE THE SOFTWARE IMMEDIATELY.
This software is distributed as demoware. You are entitled to use this software MPEG Layer-3 codec for 30 days for evaluation purposes. If you want to continue to use this software codec after the evaluation period, or if you want to use this software commercially, you are required to buy the software. In the sense of these license terms, USE shall mean running one of the programs and/or make use of bitstreams generated by this software. You may give copies of the demo version of this software to other people as long as no file is changed and no file is omitted. You may not sell, rent or lease the software to others.
If you have bought this product, you are entitled to use this product for your own use. You may not sell, rent or lease the software to others without written permission of Fraunhofer Institute for Integrated Circuits (IIS). You may use only one copy of the software at one time. You may not use this software on a network or on more than one computer at the same time without a licence for »concurrent use«.
You must not give away your personal registration code. Doing so will result in an infringement of copyright. Fraunhofer IIS and/or OPTICOM retain the right of claims for compensation in respect of damage which occured by your giving away of the registration code. This claim shall also extent to all costs which Fraunhofer IIS and/or OPTICOM incur in defending itself.
The license and right for the use of this Software does not include the right to create, generate, encode or otherwise modify data or bit streams:

- to be used, sold, published, distributed, disposed of or otherwise marketed via pre-recorded media, such as but not limited to CD-ROM, magnetic tapes, memory cards and the like;
- to be used, sold, reproduced, published, distributed, disposed of or otherwise marketed via any kind of network, if a user will have to pay a monetary or equivalent compensation for the access, copying etc. of such data or bit stream;
- for the purpose of broadcast and/or radio and/or multicast service transmission such as but not limited to »Internet Radio« and the like.

WARRANTY AND DISCLAIMER

There are no warranties associated with this software. While we believe that our software is reasonably bug free and well behaved, we are in no way responsible if our software does not work the way you would expect it to work. No matter if it locks up your computer, garbles your floppy disks or does any other harmful things to your computer — it is entirely your problem. Fraunhofer IIS and/or OPTICOM are not liable for any infringements or damages of third parties' rights in consequence of your use of this product. Fraunhofer IIS and/or OPTICOM are in no event liable for, respectively do not warrant the trustworthiness, quality, industrial exploitability, serviceability of this product for the supposed purpose or any other purposes. All orders are subject to the general terms and conditions »Allgemeine Verkaufs- und Lieferbedingungen« of OPTICOM.
This information may be subject to change.
All brand and product names are trademarks and/or registered trademarks of their respective owners. All rights reserved.

Why and how to buy this software

Why should you buy this software?

- Your license will be valid for all V3.xx versions of the encoder
- As a registered user, you can get free support by email, mail, fax or phone (see page 14)
- You support the development of better versions of MP3Enc and the development of even better compression algorithms.
- No more 30-second limit!
- The unlimited MPEG Layer-3 decoder l3dec V2.74 is included.

How to buy this software

Please see the OPTICOM Website, http://www.opticom.de/ for information on prices and ordering. You can also get information by sending a fax requesting prices to +49 (0) 9131 / 691–325 or by sending email to sales@opticom.de. If all else fails, OPTICOM can be reached by snail-mail at:

> OPTICOM
> Am Weichselgarten 7
> D-91058 Erlangen
> Germany

Please do not direct any questions about pricing and ordering information at Fraunhofer IIS or to the technical support facilities.

Chapter 1

For the impatient

If you are new to audio compression, you should read section 1.1 for an introduction about audio compression and MPEG Layer-3.
If, however, you want to jump right into the business of sound compression, then Section 1.2 will show you some prefabricated command lines that will give you compressed audio streams right away.
If you are an expert in audio coding already, the command line switch reference page (see page 7) might come in handy.

1.1 Introduction

There is a lot of confusion surrounding the terms *audio compression*, *audio encoding*, and *audio decoding*. This section will give you an overview what audio coding (another one of these terms...) is all about.

The purpose of audio compression

Up to the advent of audio compression, high-quality digital audio data took a lot of hard disk space to store. Let us go through a short example.
You want to, say, sample your favorite 1-minute song and store it on your harddisk. Because you want CD quality, you sample at 44.1 kHz, stereo, with 16 bits per sample.
44100 Hz means that you have 44100 values per second coming in from your sound card (or input file). Multiply that by two because you have two channels. Multiply by another factor of two because you have two bytes per value (that's what 16 bit means). The song will take up

$$44\,100\,\frac{\text{samples}}{\text{s}} \cdot 2\text{ channels} \cdot 2\frac{\text{bytes}}{\text{sample}} \cdot 60\frac{\text{s}}{\text{min}} \approx 10\text{ MByte}$$

of storage space on your harddisk.
If you wanted to download that over the internet, given an average 28.8 modem, it would take you (at least)

$$10\,000\,000\text{ bytes} \cdot 8\frac{\text{bits}}{\text{byte}}/(28\,800\frac{\text{bits}}{\text{s}} \cdot 60\frac{\text{s}}{\text{min}}) \approx 45\text{ minutes}$$

Just to download one minute of music!

Digital audio coding, which - in this context - is synonymously called digital audio compression as well, is the art of minimizing storage space (or channel bandwidth) requirements for audio data. Modern perceptual audio coding techniques (like MPEG Layer-3) exploit the properties of the human ear (the perception of sound) to achieve a size reduction by a factor of 12 with little or no perceptible loss of quality.
Therefore, such schemes are the key technology for high quality low bit-rate applications, like soundtracks for CD-ROM games, solid-state sound memories, Internet audio, digital audio broadcasting systems, and the like.

The two parts of audio compression

Audio compression really consists of two parts. The first part, called *encoding*, transforms the digital audio data that resides, say, in a WAVE file, into a highly compressed form called *bitstream*. To play the bitstream on your soundcard, you need the second part, called *decoding*. Decoding takes the bitstream and re-expands it to a WAVE file.
The program that effects the first part is called an audio *encoder*. MP3Enc is such an encoder; there are others, see http://www.fhg.iis.de/audio/.
The program that does the second part is called an audio *decoder*. One well-known MPEG Layer-3 decoder is WinPlay3, another `l3dec`. Both can be found on http://www.fhg.iis.de/audio/.

Compression ratios, bitrate and quality

It has not been explicitly mentioned up to now: What you end up with after encoding and decoding is not the same sound file anymore: All superflous information has been squeezed out, so to say. It is not the same *file*, but it will *sound* the same – more or less, depending on how much compression had been performed on it.
Generally speaking, the lower the compression ratio achieved, the better the sound quality will be in the end – and *vice versa*. Table 1.1 gives you an overview about quality achievable.
Because compression ratio is a somewhat unwieldy measure, experts use the term *bitrate* when speaking of the strength of compression. Bitrate denotes the average number of bits that one second of audio data will take up in your compressed bitstream. Usually the units used will be kbps, which is $\frac{\text{k bits}}{\text{s}}$, or $\frac{1000 \text{ bits}}{\text{s}}$. To calculate the number of bytes per second of audio data, simply divide the number of bits per second by eight.

1.2 Some examples

- Encode a WAVE-file `myfile.wav` to a bitrate of 128000 bits/s, writing to a plain bitstream `myfile.mp3`

Bitrate	Bandwidth	Quality comparable to or better than
8 kBps	2.5 kHz	POTS (telephone sound)
16 kBps	4.5 kHz	shortwave radio
32 kBps	7.5 kHz	AM radio
64 kBps	11 kHz	FM radio
128 kBps	15 kHz	CD

Table 1.1: Bitrate versus sound quality

```
mp3enc -br 128000 -if myfile.wav -of myfile.mp3
```

- Encode a plain PCM file (2-channel, 44.1 kHz) to a plain 56 kBit/s Layer-3 stream, using the encoder as filter
  `readFromSoundCard | mp3enc -sti -sto -iff "nc=2 sr=44100 bps=16" -br 56000 -qual 3 | streamToWeb`

1.3 Command line switch reference

switch	parameter	see section
`-br`	bitrate	2.1.2
`-if`	input file name	
`-of`	output file name	
`-iff`	input file format	
`-l3wav`	write Microsoft RIFF/WAVE layer-3 file	2.1.6
`-sti`	take input from pipe (stdin)	
`-sto`	write output into a pipe (stdout)	
`-qual`	quality	2.2.2
`-esr`	effective sampling rate	2.2.1
`-crc`	CRC checksum	2.2.2
`-dm`	downmix stereo file to mono	2.2.2
`-v`	be verbose	
`-no-is`	do not use intensity stereo	2.1.3

Chapter 2

MP3Enc Features

2.1 Basics

2.1.1 Samplerate

Sample rate is the rate at which the samples are read from your sound card when you sample. Sample rate is directly linked to audio bandwidth achievable: A sound file with a sample rate of 8 kHz does not contain frequencies beyond 4 kHz. This means that you should always use the highest sample rate that your sound card supports when you sample a signal.
The encoder changes the sample rate of your audio data to match it to the audio quality of the bitstream produced by the encoder. This process is called *downsampling*.

2.1.2 Bitrate

The main parameter controlling the sound quality is the *bitrate* that the encoder runs at. In a nutshell, the higher the bitrate, the better the quality.
The bitrate of the encoder is linked to the samplerate that the encoded file will have. Usually, the encoder will choose a samplerate that is suited best for encoding at that bitrate. You can override this samplerate using the `-esr` switch (see section 2.2.1).
The bitrate of the bitstream output is selected via the `-br` switch. The bitrate is specified in bits/second. The bitrate is the total bitrate for all encoded channels, i.e. if you select `-br 112000` and encode in stereo, both channels will be stuffed into one bitstream of 112000 bits/second.
The encoder supports bitrates of 8, 16, 18, 20, 24, 32, 40, 48, 56, 64, 96, 112, 128, 160, 192 and 256 kBit/s. While all of these can be used with mono signals, stereo works from 20 kBit/s on upwards.

2.1.3 Stereo mode

If encoding stereo, the bitrate of the encoder is linked to a stereo mode. MPEG Layer-3 knows four modes for stereo encoding.

Bitrates	stereo mode
8000–18000	mono only
18000–96000	MS/IS stereo
96000–192000	MS stereo
192000–256000	stereo

Table 2.1: different stereo modes

dual channel (also known as *dual mono*) In this mode, the encoder treats the two input channels as separate entities, assuming there is no similarity between the channels. This would be appropriate if you e.g. have a bilingual signal where one channel contains a german speaker and one contains an english speaker.

stereo In this mode, like in dual channel above, the encoder makes no use of potentially existing correlations between the two input channels. It can, however, negotiate the bit demand between both channel, i.e. give one channel more bits if the other contains silence.

MS stereo In this mode, the encoder will make use of a correlation between both channels. The signal will be matrixed into a sum (»mid«) and difference (»side«) signal. For quasi-mono signals, this will give a significant gain in encoding quality.

This mode does not destroy phase information like IS stereo (see below) and thus can be used to encode DOLBY ProLogic^{tm} surround signals.

MS/IS stereo In this mode, high-frequency parts of the signal will be downmixed to mono and transmitted with a direction information (which is basically a pan). This mode (called »intensity stereo« will loose phase information and should not be used for high-quality encoding.

Table 2.1 gives you an overview which mode will be used for which bitrate.

2.1.4 Encoding speed

Several factors influence the speed of the encoder. They include:

- Number of channels in the output signal. If your output signal has only one channel, the encoder will run at twice the speed compared to stereo encoding.
- Output sample rate. If the encoder produces a file at 22.050 kHz (that is, a file that contains 22050 samples per second), it runs at twice the speed compared to one that produces twice the number of samples per second (i.e. produces a 44.1 kHz output).
- Mismatch between input and output sample rate. If your input and output sample rates differ, the encoder will have to run a resampling filter and thus will be slower. (Integer ratios between input and output sample rate perform slightly better than non-integer ratios, though).

- Time-domain bandlimiting. The encoder needs to band-limit the signal to compress it. By default, the encoder will use a high-quality time domain filter to do this band-limiting. You can tell it to use a faster filter, possibly sacrificing some quality (see 2.2.2).
- Full huffman search and careful iteration. You can tell the encoder to try hard to do the best encoding possible, at the expense of a factor of up to three in running time (see 2.2.2).

Version V3.0 of the encoder reaches realtime speed on a Pentium 166 when encoding at 64 kBit/s, 22,050 kHz, stereo. On a SUN Sparc Ultra-1 (143 MHz) the performance is similar.

2.1.5 Input file specification

The encoder can read AIFF, AIFF-C, WAV/RIFF and raw PCM data files. While the first three only work from a file, plain PCM data can be fed into the encoder via a pipe. This is useful for live encoding (also known as *streaming*).

Input from file: filename

`-if filename` will tell the encoder the filename it reads it input from. If the file is a RIFF/WAVE file or an AIFF/AIFC file, the encoder will automatically adapt to the sound file format. For other formats or plain PCM data, see below.

Piping data into the encoder

`-sti` tells the encoder to get its input from stdin rather than from a file. This only works when the input is plain pcm data (see below).

plain PCM data input

If the encoder gets its input as plain pcm data (or if it does not recognize the sound format by itself), you need to tell it all about the structure of the PCM stream, i.e. the number of bits per sample, the number of channel and the samplerate.

`-iff fileformat` This is a string containing name=value pairs, separated by blanks. Table 2.2 gives a reference which names and values are possible here.

For stereo files, the encoder assumes that the PCM data is interleaved and that the sample for the right channel follows that for the left channel.

As an example, `-iff "nc=2 sr=44100 bps=16"` would be used to read a 44.1 kHz stereo file with 16 bits per sample while `-iff "nc=1 sr=8000 bps=8"` would tell the encoder that the data is mono, sampled at 8 kHz with 8 bits per sample.
Remember that this feature is only needed for input from files other than RIFF/WAV, AIFF and AIFC.

Name	Value(s)	Explanation
`sr`	any	The rate the PCM signal is sampled at [Hz]
`nc`	1, 2	The number of channels in the signal
`bps`	8, 16, 24, 32	The number of bits per sample
`little-endian`		The signal is little-endian (Intel format)
`big-endian`		The signal is big-endian (Motorola format)

Table 2.2: input file format specification

2.1.6 Output file specification

On output, the encoder can be instructed to write a plain Layer-3 bitstream or a wave file containing the Layer-3 stream. These wave files can be played by the media control on a machine running under Microsoft Windows that has the MPEG Layer-3 ACM codec installed (you can get one by installing Microsoft Netshow^{tm}, http://www.microsoft.com/netshow/).
If the output is a plain Layer-3 stream, it can be piped into other applications. This is useful for live streaming.

`-of filename` tells the encoder the filename of the file that the encoder will write the bitstream to. If the file does not exist, it is created; if it does exist, it will be overwritten.

`-l3wav` tells the encoder to wrap the MPEG Layer-3 file into a Microsoft RIFF/WAVE file.

Streaming data out of the encoder

`-sto` tells the encoder to write its output into stdout rather than in a file. This only works when the output is a raw Layer-3 bitstream (i.e. it does not work in conjunction with `-l3wav`).

2.2 Advanced features

2.2.1 Overriding default settings

Many of the following features override the encoder's idea of best-quality settings. You should be aware that overriding the encoder default settings is something for experts. You might wreck the encoding quality in a number of ways without first noticing it. Also, the encoder is not guaranteed to run at all parameter combinations. **Proceed at your own risk!**

`-esr` Output (*effective*) sample rate. Usually, the encoder will choose an output sample rate from 8, 16, 32, or 48 kHz. With some soundcards, it is not possible to play files with sample rates of 48 kHz, others cannot do 32 kHz. With this switch, you can tell the encoder to use another output sample rate.[1]

[1] You can also use this switch to match your output sample rate to an integer fraction of the input sample rate to get slightly faster performance

`-dual` Use dual channel stereo instead of the default mode (see table 2.1). At bitrates of 128 kBit/s and below, this switch will almost certainly decrease the sound quality.

`-bw` Tell the encoder to use another bandwidth. Increasing the bandwidth from the default setting will work for some signals, but might produces ringing artefacts for others. Use with care!

It is not possible to choose bandwidths above half the output sample rate.

`-no-is` Tell the encoder not to use intensity stereo (see 2.1.3). Some special signals experience susceptible loss of quality if phase information is destroyed; in these cases, you may gain some sound quality using this switch.

2.2.2 Tids & bits

`-crc` For transmission over serial lines with bit errors, parts of the bitstream can be protected by calculating a CRC checksum. If you are just producing for harddisk storage, there is no need to set this switch.

`-dm` To encode at bitrates ranging from 8 to 18 kBit/s, you need a mono input signal. This switch tells the encoder to downmix a stereo input signal into one channel, producing mono output. The downmix is calculated as the sum of the left and right channel, attenuated by 6 dB.

`-qual` This switch controls the tradeoff between fast encoder operation and best sound quality. Table 2.3 gives you an overview which features of the encoder are switched on/off by the `-qual` switch.

In future versions of the encoder, more features might be controlled by this switch. The only facts you should count on:

- fastest operation is guaranteed with `-qual 0`
- highest encoding quality is reached with `-qual 9`.[2]

[2] For some figures on encoding speed see section 2.1.4

Feature	Explanation
Soft time-domain filtering	Use a high-quality time domain filter instead of fast MDCT
Best match sampling rate	Use the best sample rate without regard to filter running time. Adapting to this sample rate might use CPU-intensive filtering.
Full huffman search	Find the best huffman code book possible to encode the spectrum of each frame. A few percent bits can be saved in each frame, available for higher quality in following frames.
Many outer loops	Shape the quantization noise very carefully.

Table 2.3: Features controlled by the `-qual` switch

Chapter 3

Troubleshooting

No software is free of errors. If you believe you have found an error in the operation of MP3Enc, and you have checked the list below, please report the error to our bugtracking address.

3.1 Is it really a bug?

Before you report a bug to our engineers, please verify that the bug is really in the software and not in your configuration. Table 3.1 helps you track down the bug yourself and see if it can be fixed.

3.2 Reporting the bug

To assist our engineers in the processing of your bug report, we ask you to include in your mail

- The version of the encoder you are using.
- Your user name and serial number as reported by the encoder.
- The operating system (name and version) you are running the software with. If you are using a sort of UNIX, please cite the output of `uname -a`. If you are using Windows, please right-click on the »My Computer« icon that usually resides in the top left-hand corner of your screen and report the lines following »System« and »Computer«.
- The exact command line that you entered before you encountered the error.
- The output of the encoder when appending the `-v` switch to the command line.

If you have gathered this information, please fax it to OPTICOM (fax: +49 (0) 9131 / 691–325) or write an email to l3bugs@iis.fhg.de.
If you have bought this product, you will get an immediate acknowledgement by email once your bug report has reached us. You will receive a second email as soon as the bug report has been processed. Expect some delay between the first and second email.

Symptom	Check this:
`AL error : AL_detect : Unable to open file!`	Have you given an input file to the encoder (see section 2.1.5)? Does the input file exist? Is it readable?
`could not open output file`	Have you given an output file to the encoder (see section 2.1.6)? Does the output directory exist and is it writeable? Does a file of the same name exist and is it deleteable?
`No parameters for this bitrate/samplerate`	Did you override any of the encoders parameters (stereo mode, samplerate)? If so, try another samplerate.
`bitrate too low/high`	MPEG Layer-3 only allows bitrates ranging from 8 kBit/s to 320 kBit/s.
The Layer-3 file sounds muffled	Try using a higher bitrate. Try using a higher bandwidth (see section 2.2.1). Try using a higher effective sample rate (see section 2.2.1)
The stereo image is destroyed.	Try using the `-no-is` switch

Table 3.1: Bug symptoms and possible causes

3.3 Sample bug report

This is a sample bug report that you may use as a template for your own.

```
To: l3bugs@iis.fhg.de
Subject: Mp3enc bug

Hello,

I am using mp3enc Version V3.0 on a PC (according to
the System Properties dialog, it is running Microsoft Windows NT
4.00.1381; the computer contains a x86 Family 5 Model 2 Stepping 12
AT/AT compatible and 64,951 KB RAM).

My serial number and user name (as reported by mp3enc) are
 Hantan Blaumilch, 123456.

When I run the program as

mp3enc -br 127957 -if myfile.wav -of foobar.mp3 -v

I get the following error message:

*********** MPEG Layer-3 Encoder V3.00 (build Mar  4 1998) ***************
                     (C) 1998 by Fraunhofer IIS-A

This program is protected by copyright law and international treaties.
```

```
Any reproduction or distribution of this program, or any portion
of it, may result in severe civil and criminal penalties, and will be
prosecuted to the maximum extent possible under law.

in:  44100 Hz, 2 channel(s), 16 bit/sample
out: 44100 Hz, 2 channel(s), 128000 bit/s
MS Stereo ON
6144 /  830902 ( 1%)
** mp3enc error: Illegal codebook encountered.

Regards,
  Hantan
```

Bibliography

[1] M. Bosi, K. Brandenburg, et al. ISO/IEC MPEG-2 advanced audio coding. In *101st AES conference*, Los Angeles, November 1996. preprint 4382.

[2] K. Brandenburg and M. Bosi. Overview of MPEG-audio: Current and future standards for low bit-rate audio coding. In *99th AES conference*, New York, October 1995. preprint 4130.

[3] K. Brandenburg, G. Stoll, et al. The ISO/MPEG-audio codec: A generic standard for coding of high quality digital audio. In Neil Gilchrist and Christer Grewin, editors, *Collected Papers On Digital Audio Bit-Rate Reduction*, pages 31–42. AES, 1996.

[4] R. Buchta, S. Meltzer, et al. The Worldstar sound format. In *101st AES conference*, Los Angeles, November 1996. preprint 4385.

[5] S. Church, B. Grill, H. Popp, et al. ISDN and ISO/MPEG layer-3 audio coding: Powerful new tools for broadcast and audio production. In *95th AES conference*, Amsterdam, October 1993. preprint 3743.

[6] M. Dietz, H. Popp, et al. Audio compression for network transmission. In *99th AES conference*, New York, October 1995. preprint 4129.

[7] E. Eberlein, H. Popp, et al. Layer-3, a flexible coding standard. In *94th AES conference*, Berlin, March 1993. preprint 3493.

[8] B. Grill, J. Herre, et al. Improved MPEG-2 audio multi-channel encoding. In *96th AES conference*, Amsterdam, February 1994. preprint 3865.

[9] J. Herre, K. Brandenburg, et al. Second generation ISO/MPEG audio layer-3 coding. In *98th AES conference*, Paris, February 1995.

[10] Witte, M. Dietz, et al. Single chip implementation of an ISO/MPEG layer-3 decoder. In *96th AES conference*, Amsterdam, February 1994. preprint 3805.

Index