Subsections

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 44100 samples/s · 2 channels · 2 bytes/sample · 60 s/min ~ 10 MBytes of storage space on your harddisk.

If you wanted to download that over the internet, given an average 56k modem connected at 44k (which is a typical case), it would take you (at least) 10000000 bytes · 8 bits/byte / (44000 bits/s) · / (60 s/min) ~ 30 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 III) exploit the properties of the human ear (the perception of sound) to achieve a size reduction by a factor of 11 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*. *LAME* is such an encoder . The program that does the second part is called an audio *decoder*. One well-known MPEG Layer III decoder is `Xmms`, another `mpg123`. Both can be found on ww.mp3-tech.org .

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 $^{\text{kbits}}/_{\text{s}}$, or 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.

Table 1.1: Bitrate versus sound quality

Bitrate	Bandwidth	Quality comparable to or better than
16 kbps	4.5 kHz	shortwave radio
32 kbps	7.5 kHz	AM radio
96 kbps	11 kHz	FM radio
128 kbps	16 kHz	near CD
160-180 kbps (variable bitrate)	20 kHz	perceptual transparency
256 kbps	22 kHz	studio

Some command line examples

- Fixed bit rate jstereo 128kbs encoding:
  `lame sample.wav sample.mp3`
- Fixed bit rate jstereo 128kbs encoding, highest quality (recommended):
  `lame -h sample.wav sample.mp3`
- Fixed bit rate jstereo 112kbs encoding:
  `lame -b 112 sample.wav sample.mp3`
- To disable joint stereo encoding (slightly faster, but less quality at bitrates<=128kbs):
  `lame -m s sample.wav sample.mp3`
- Fast encode, low quality (no psycho-acoustics):
  `lame -f sample.wav sample.mp3`
- Variable bitrate (use -V n to adjust quality/filesize):
  `lame -h -V 6 sample.wav sample.mp3`
- Streaming mono 22.05 kHz raw pcm, 24 kbps output:
  `cat inputfile | lame -r -m m -b 24 -s 22.05 -- > output`
- Streaming mono 44.1 kHz raw pcm, with downsampling to 22.05 kHz:
  `cat inputfile | lame -r -m m -b 24 --resample 22.05 -- > output`

CBR/ABR/VBR: the 3 encoding modes

LAME is able to encode your music using one of its 3 encoding modes: constant bitrate (CBR), average bitrate (ABR) and variable bitrate (VBR).

Constant Bitrate (CBR)

This is the default encoding mode, and also the most basic. In this mode, the bitrate will be the same for the whole file. It means that each part of your mp3 file will be using the same number of bits. The musical passage beeing a difficult one to encode or an easy one, the encoder will use the same bitrate, so the quality of your mp3 is variable. Complex parts will be of a lower quality than the easiest ones. The main advantage is that the final files size won't change and can be accurately predicted.

Average Bitrate (ABR)

In this mode, you choose the encoder will maintain an average bitrate while using higher bitrates for the parts of your music that need more bits. The result will be of higher quality than CBR encoding but the average file size will remain predictible, so this mode is highly recommended over CBR. This encoding mode is similar to what is reffered as vbr in AAC or Liquid Audio (2 other compression technologies).

Variable bitrate (VBR)

In this mode, you choose the desired quality on a sqale from 9 (lowest quality/biggest distortion) to 0 (highest quality/lowest distortion). Then encoder tries to maintain the given quality in the whole file by choosing the optimal number of bits to spend for each part of your music. The main advantage is that you are able to specify the quality level that you want to reach, but the inconvenient is that the final file size is totally unpredictible.

Basic command line switch reference

Only the most usual switches are described here. However those should be sufficient for the vast majority of users.

switch	parameter
--abr	average bitrate encoding
-b	bitrate (8...320)
--decode	decoding only
-f	fast mode
-h	high quality
--help	help
-m	stereo mode (s, **j**, f, m)
-V	VBR quality setting (0...9)

* `--abr n` **average bitrate encoding**
Turns on encoding with a targeted average bitrate of n kbits, allowing to use frames of different sizes. The allowed range of n is 4-310, you can use any integer value within that range.

---

* `-b n` **bitrate**
For MPEG1 (sampling frequencies of 32, 44.1 and 48 kHz)
n = 32,40,48,56,64,80,96,112,128,160,192,224,256,320

For MPEG2 (sampling frequencies of 16, 22.05 and 24 kHz)
n = 8,16,24,32,40,48,56,64,80,96,112,128,144,160

Default is 128 kbs for MPEG1 and 80 kbs for MPEG2.

---

* `--decode` **decoding only**
Uses LAME for decoding to a wav file. The input file can be any input type supported by encoding, including layer I,II,III (MP3).

---

* `-f` **fast mode**

This switch forces the encoder to use a faster encoding mode, but with a lower quality.

---

* `-h` **high quality**

Use some quality improvements. Encoding will be slower, but the result will be of higher quality. This switch is always enabled when using VBR.

---

* `--help` **help**

Display a list of all available options.

---

* `-m s/j/m` **stereo mode**

Joint-stereo is the default mode for stereo files with VBR when -v is more than 4 or fixed bitrates of 160kbs or less. At higher fixed bitrates or higher VBR settings, the default is stereo.

stereo
Normal stereo mode.

joint stereo
In this mode, the encoder will make use of a correlation between both channels in order to achieve higher compression. This will enhance the quality of constant bitrate recordings, and reduce the size of variable bitrate recordings.

mono
The input will be encoded as a mono signal.

---

* `-V 0...9` **VBR quality setting**

Enable VBR (Variable BitRate) and specifies the value of VBR quality.
default=4
0=highest quality.

Full command line switch reference

note: Options which could exist without beeing documented here are considered as experimental ones. Such experimental options should usually not be used.

switch	parameter
-a	downmix stereo file to mono
--abr	average bitrate encoding
--allshort	use short blocks only
--athlower	lower the ATH
--athonly	ATH only
--athshort	ATH only for short blocks
--athtype	select ATH type
-b	bitrate (8...320)
-B	max VBR/ABR bitrate (8...320)
--bitwidth	input bit width
-c	copyright
--comp	choose compression ratio
--cwlimit	tonality limit
-d	block type control
--decode	decoding only
--disptime	time between display updates
-e	de-emphasis (**n**, 5, c)
-f	fast mode
-F	strictly enforce the -b option
--freeformat	free format bitstream
-h	high quality
--help	help
--highpass	highpass filtering frequency in kHz
--highpass-width	width of highpass filtering in kHz
-k	full bandwidth
--lowpass	lowpass filtering frequency in kHz
--lowpass-width	width of lowpass filtering in kHz
-m	stereo mode (s, **j**, f, m)

--mp1input	MPEG Layer I input file
--mp2input	MPEG Layer II input file
--mp3input	MPEG Layer III input file
--noath	disable ATH
--nohist	disable histogram display
--nores	disable bit reservoir
--noshort	disable short blocks frames
--notemp	disable temporal masking
-o	non-original
-p	error protection
--preset	use built-in preset
--alt-preset	use updated and much higher quality "alternate" presets
--priority	OS/2 process priority control
-q	algorithm quality selection
--quiet	silent operation
-r	input file is raw pcm
--resample	output sampling frequency in kHz (encoding only)
--r3mix	r3mix VBR preset
-s	sampling frequency in kHz
-S	silent operation
--scale	scale input
--scale-l	scale input channel 0 (left)
--scale-r	scale input channel 1 (right)
--short	use short blocks
--silent	silent operation
--strictly-enforce-ISO	strict ISO compliance
-t	disable INFO/WAV header
-V	VBR quality setting (0...9)
--vbr-new	new VBR mode
--vbr-old	older VBR mode
--verbose	verbosity
-x	swapbytes
-X	change quality measure

* `-a` **downmix**

Mix the stereo input file to mono and encode as mono.
The downmix is calculated as the sum of the left and right channel, attenuated by 6 dB.

This option is only needed in the case of raw PCM stereo input (because LAME cannot determine the number of channels in the input file).
To encode a stereo PCM input file as mono, use "lame -m s -a".

For WAV and AIFF input files, using "-m m" will always produce a mono .mp3 file from both mono and stereo input.

---

* `--abr n` **average bitrate encoding**

Turns on encoding with a targeted average bitrate of n kbits, allowing to use frames of different sizes. The allowed range of n is 8-310, you can use any integer value within that range.

It can be combined with the -b and -B switches like:
lame --abr 123 -b 64 -B 192 a.wav a.mp3
which would limit the allowed frame sizes between 64 and 192 kbits.

---

* `--allshort` **use short blocks only**

Use only short blocks, no long ones.

---

* `--athlower n` **lower the ATH**

Lower the ATH (absolute threshold of hearing) by n dB.
Normally, humans are unable to hear any sound below this threshold, but for music recorded at very low level this option might be usefull.

---

* `--athonly` **ATH only**

This option causes LAME to ignore the output of the psy-model and only use masking from the ATH (absolute threshold of hearing). Might be useful at very high bitrates or for testing the ATH.

---

* `--athshort` **ATH only for short blocks**

Ignore psychoacoustic model for short blocks, use ATH only.

---

* `--athtype 0/1/2` **select ATH type**

The Absolute Threshold of Hearing is the minimum threshold under which humans are unable to hear any sound. In the past, LAME was using ATH shape 0 which is the Painter & Spanias formula. Tests have shown that this formula is innacurate for the 13-22 kHz area, leading to audible artifacts in some cases. Shape 1 was thus implemented, which is over sensitive, leading to very high bitrates. Shape 2 formula was accurately modelized from real data in order to real optimal quality while not wasting bitrate. In CBR and ABR modes, LAME uses ATH shape 2 by default.

In VBR mode, LAME is adapting its shape according to the -V value, going gradually from the 0 shape at -V9 up to shape 2 at -V0.

---

* `-b n` **bitrate**

For MPEG1 (sampling frequencies of 32, 44.1 and 48 kHz)
n = 32,40,48,56,64,80,96,112,128,160,192,224,256,320

For MPEG2 (sampling frequencies of 16, 22.05 and 24 kHz)
n = 8,16,24,32,40,48,56,64,80,96,112,128,144,160

Default is 128 kbs for MPEG1 and 64 kbs for MPEG2.

When used with variable bitrate encoding (VBR), -b specifies the minimum bitrate to be used. However, in order to avoid wasted space, the smallest frame size available will be used during silences.

---

* `-B n` **maximum VBR/ABR bitrate**

For MPEG1 (sampling frequencies of 32, 44.1 and 48 kHz)
n = 32,40,48,56,64,80,96,112,128,160,192,224,256,320

For MPEG2 (sampling frequencies of 16, 22.05 and 24 kHz)
n = 8,16,24,32,40,48,56,64,80,96,112,128,144,160

Specifies the maximum allowed bitrate when using VBR/ABR

The use of -B is NOT RECOMMENDED. A 128kbs CBR bitstream, because of the bit reservoir, can actually have frames which use as many bits as a 320kbs frame. VBR modes minimize the use of the bit reservoir, and thus need to allow 320kbs frames to get the same flexibility as CBR streams.

note: If you own an mp3 hardware player build upon a MAS 3503 chip, you must set maximum bitrate to no more than 224 kpbs.

* `--bitwidth 8/16/24/32` **input bit width**

Required only for raw PCM input files. Otherwise it will be determined from the header of the input file.

---

* `-c` **copyright**
Mark the encoded file as being copyrighted.

---

* `--comp` **choose compression ratio**
Instead of choosing bitrate, using this option, user can choose compression ratio to achieve.

---

* `--cwlimit n` **tonality limit**
Compute tonality up to freq (in kHz). Default setting is 8.8717.

---

* `-d` **block type control**
Allows the left and right channels to use different block size types.

---

* `--decode` **decoding only**
Uses LAME for decoding to a wav file. The input file can be any input type supported by encoding, including layer I,II,III (MP3) and OGG files. In case of MPEG files, LAME uses a bugfixed version of mpglib for decoding.

If -t is used (disable wav header), Lame will output raw pcm in native endian format. You can use -x to swap bytes order.

---

* `--disptime n` **time between display updates**
Set the delay in seconds between two display updates.

---

* `-e n/5/c` **de-emphasis**

n = (none, default)
5 = 0/15 microseconds
c = citt j.17

All this does is set a flag in the bitstream. If you have a PCM input file where one of the above types of (obsolete) emphasis has been applied, you can set this flag in LAME. Then the mp3 decoder should de-emphasize the output during playback, although most decoders ignore this flag.

A better solution would be to apply the de-emphasis with a standalone utility before encoding, and then encode without -e.

---

* `-f` **fast mode**

This switch forces the encoder to use a faster encoding mode, but with a lower quality. The behaviour is the same as the -q7 switch.

Noise shaping will be disabled, but psycho acoustics will still be computed for bit allocation and pre-echo detection.

---

* `-F` **strictly enforce the -b option**

This is mainly for use with hardware players that do not support low bitrate mp3.

Without this option, the minimum bitrate will be ignored for passages of analog silence, ie when the music level is below the absolute threshold of human hearing (ATH).

---

* `--freeformat` **free format bitstream**

Produces a free format bitstream. With this option, you can use -b with any bitrate higher than 8 kbps.

However, even if an mp3 decoder is required to support free bitrates at least up to 320 kbps, many players are unable to deal with it.

Tests have shown that the following decoders support free format:

FreeAmp up to 440 kbps
in_mpg123 up to 560 kbps
l3dec up to 310 kbps
LAME up to 560 kbps
MAD up to 640 kbps

---

* `-h` **high quality**

Use some quality improvements. Encoding will be slower, but the result will be of higher quality.
The behaviour is the same as the -q2 switch.
This switch is always enabled when using VBR.

---

* `--help` **help**

Display a list of all available options.

---

* `--highpass` **highpass filtering frequency in kHz**

Set an highpass filtering frequency. Frequencies below the specified one will be cutoff.

---

* `--highpass-width` **width of highpass filtering in kHz**

Set the width of the highpass filter. The default value is 15% of the highpass frequency.

---

* `-k` **full bandwidth**

Tells the encoder to use full bandwidth and to disable all filters. By default, the encoder uses some highpass filtering at low bitrates, in order to keep a good quality by giving more bits to more important frequencies.
Increasing the bandwidth from the default setting might produce ringing artefacts at low bitrates. Use with care!

---

* `--lowpass` **lowpass filtering frequency in kHz**

Set a lowpass filtering frequency. Frequencies above the specified one will be cutoff.

---

* `--lowpass-width` **width of lowpass filtering in kHz**

Set the width of the lowpass filter. The default value is 15% of the lowpass frequency.

---

* `-m s/j/f/d/m` **stereo mode**

Joint-stereo is the default mode for stereo files with VBR when -V is more than 4 or fixed bitrates of 160kbs or less. At higher fixed bitrates or higher VBR settings, the default is stereo.

stereo
In this mode, 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 or needs less bits because of a lower complexity.

joint stereo
In this mode, the encoder will make use of a correlation between both channels. The signal will be matrixed into a sum ("mid"), computed by L+R, and difference ("side") signal, computed by L-R, and more bits are allocated to the mid channel.
This will effectively increase the bandwidth if the signal does not have too much stereo separation, thus giving a significant gain in encoding quality.

Using mid/side stereo inappropriately can result in audible compression artifacts. To much switching between mid/side and regular stereo can also sound bad. To determine when to switch to mid/side stereo, LAME uses a much more sophisticated algorithm than that described in the ISO documentation, and thus is safe to use in joint stereo mode.

forced joint stereo
This mode will force MS joint stereo on all frames. It's slightly faster than joint stereo, but it should be used only if you are sure that every frame of the input file has very little stereo separation.

dual channels
In this mode, the 2 channels will be totally indenpendently encoded. Each channel will have exactly half of the bitrate. This mode is designed for applications like dual languages encoding (ex: English in one channel and French in the other). Using this encoding mode for regular stereo files will result in a lower quality encoding.

mono
The input will be encoded as a mono signal. If it was a stereo signal, it will be downsampled to mono. The downmix is calculated as the sum of the left and right channel, attenuated by 6 dB.

---

* `--mp1input` **MPEG Layer I input file**

Assume the input file is a MPEG Layer I file.
If the filename ends in ".mp1" or ".mpg" LAME will assume it is a MPEG Layer I file. For stdin or Layer I files which do not end in .mp1 or .mpg you need to use this switch.

---

* `--mp2input` **MPEG Layer II input file**
  Assume the input file is a MPEG Layer II (ie MP2) file.
  If the filename ends in ".mp2" LAME will assume it is a MPEG Layer II file. For stdin or Layer II files which do not end in .mp2 you need to use this switch.

---

* `--mp3input` **MPEG Layer III input file**
  Assume the input file is a MP3 file. Usefull for downsampling from one mp3 to another. As an example, it can be usefull for streaming through an IceCast server.
  If the filename ends in ".mp3" LAME will assume it is an MP3 file. For stdin or MP3 files which do not end in .mp3 you need to use this switch.

---

* `--noath` **disable ATH**
  Disable any use of the ATH (absolute threshold of hearing) for masking. Normally, humans are unable to hear any sound below this threshold.

---

* `--nohist` **disable histogram display**
  By default, LAME will display a bitrate histogram while producing VBR mp3 files. This will disable that feature.
  Histogram display might not be available on your release.

---

* `--nores` **disable bit reservoir**
  Disable the bit reservoir. Each frame will then become independent from previous ones, but the quality will be lower.

---

* `--noshort` **disable short blocks frames**
  Encode all frames using long blocks only. This could increase quality when encoding at very low bitrates, but can produce serious pre-echo artefacts.

---

* `--notemp` **disable temporal masking**
  Don't make use of the temporal masking effect.

---

* `-o` **non-original**
  Mark the encoded file as being a copy.

---

* `-p` **error protection**

Turn on CRC error protection.
It will add a cyclic redundancy check (CRC) code in each frame, allowing to detect transmission errors that could occur on the MP3 stream. However, it takes 16 bits per frame that would otherwise be used for encoding, and then will slightly reduce the sound quality.

---

* `--preset presetName` **use built-in preset**

Use one of the built-in presets (phone, phon+, lw, mw-eu, mw-us, sw, fm, voice, radio, tape, hifi, cd, studio).
"--preset help" gives more information about the used options in these presets.

---

* `--alt-preset presetName` **use updated and much higher quality "alternate" presets**

Use one of the built-in alternate presets (standard, fast standard, extreme, fast extreme, insane, or the abr/cbr modes).
"--alt-preset help" gives more information about the usage possibilities for these presets.

---

* `--priority 0...4` **OS/2 process priority control**

With this option, LAME will run with a different process priority under IBM OS/2.
This will greatly improve system responsiveness, since OS/2 will have more free time to properly update the screen and poll the keyboard/mouse. It should make quite a difference overall, especially on slower machines. LAME's performance impact should be minimal.

0 (Low priority)
Priority 0 assumes "IDLE" class, with delta 0.
LAME will have the lowest priority possible, and the encoding may be suspended very frequently by user interaction.

1 (Medium priority)
Priority 1 assumes "IDLE" class, with delta +31.
LAME won't interfere at all with what you're doing.
Recommended if you have a slower machine.

2 (Regular priority)
Priority 2 assumes "REGULAR" class, with delta -31.
LAME won't interfere with your activity. It'll run just like a regular process, but will spare just a bit of idle time for the system. Recommended for most users.

3 (High priority)
Priority 3 assumes "REGULAR" class, with delta 0.
LAME will run with a priority a bit higher than a normal process.
Good if you're just running LAME by itself or with moderate user interaction.

4 (Maximum priority)
Priority 4 assumes "REGULAR" class, with delta +31.
LAME will run with a very high priority, and may interfere with the machine response.
Recommended if you only intend to run LAME by itself, or if you have a fast processor.

Priority 1 or 2 is recommended for most users.

---

* `-q 0..9` **algorithm quality selection**

Bitrate is of course the main influence on quality. The higher the bitrate, the higher the quality. But for a given bitrate, we have a choice of algorithms to determine the best scalefactors and huffman encoding (noise shaping).

-q 0: use slowest & best possible version of all algorithms. -q 0 and -q 1 are slow and may not produce significantly higher quality.

-q 2: recommended. Same as -h.

-q 5: default value. Good speed, reasonable quality.

-q 7: same as -f. Very fast, ok quality. (psycho acoustics are used for pre-echo & M/S, but no noise shaping is done.

-q 9: disables almost all algorithms including psy-model. poor quality.

---

* `-r` **input file is raw pcm**

Assume the input file is raw pcm. Sampling rate and mono/stereo/jstereo must be specified on the command line. Without -r, LAME will perform several fseek()'s on the input file looking for WAV and AIFF headers.
Might not be available on your release.

---

* `--resample 8/11.025/12/16/22.05/24/32/44.1/48` **output sampling frequency in kHz**

Select ouptut sampling frequency (for encoding only).
If not specified, LAME will automatically resample the input when using high compression ratios.

---

* `--r3mix` **r3mix VBR preset**

Uses r3mix VBR preset.
See www.r3mix.net for more details.

---

* `-s 8/11.025/12/16/22.05/24/32/44.1/48` **sampling frequency**

Required only for raw PCM input files. Otherwise it will be determined from the header of the input file.

LAME will automatically resample the input file to one of the supported MP3 samplerates if necessary.

---

* `-S / --silent / --quiet` **silent operation**

Don't print progress report.

---

* `--scale n` **scales input by n**
* `--scale-l n` **scales input channel 0 (left) by n**
* `--scale-r n` **scales input channel 1 (right) by n**

Scales input by n. This just multiplies the PCM data (after it has been converted to floating point) by n.

n > 1: increase volume
n = 1: no effect
n < 1: reduce volume

Use with care, since most MP3 decoders will truncate data which decodes to values greater than 32768.

---

* `--short` **use short blocks**

Let LAME use short blocks when appropriate. It is the default setting.

---

* `--strictly-enforce-ISO` **strict ISO compliance**

With this option, LAME will enforce the 7680 bit limitation on total frame size.
This results in many wasted bits for high bitrate encodings but will ensure strict ISO compatibility.
This compatibility might be important for hardware players.

---

* `-t` **disable INFO/WAV header**

Disable writing of the INFO Tag on encoding.
This tag in embedded in frame 0 of the MP3 file. It includes some information about the encoding options of the file, and in VBR it lets VBR aware players correctly seek and compute playing times of VBR files.

When '--decode' is specified (decode to WAV), this flag will disable writing of the WAV header. The output will be raw pcm, native endian format. Use -x to swap bytes.

---

* `-V 0...9` **VBR quality setting**

Enable VBR (Variable BitRate) and specifies the value of VBR quality.
default=4
0=highest quality.

---

* `--vbr-new` **new VBR mode**

Invokes the newest VBR algorithm. During the development of version 3.90, considerable tuning was done on this algorithm, and it is now considered to be on par with the original --vbr-old.
It has the added advantage of being very fast (over twice as fast as --vbr-old).

---

* `--vbr-old` **older VBR mode**

Invokes the oldest, most tested VBR algorithm. It produces very good quality files, though is not very fast. This has, up through v3.89, been considered the "workhorse" VBR algorithm.

---

* `--verbose` **verbosity**

Print a lot of information on screen.

---

* `-x` **swapbytes**

Swap bytes in the input file or ouptut file when using --decode.
For sorting out little endian/big endian type problems. If your encodings sounds like static, try this first.

---

* `-X 0...7` **change quality measure**

When LAME searches for a "good" quantization, it has to compare the actual one with the best one found so far. The comparison says which one is better, the best so far or the actual. The -X parameter selects between different approaches to make this decision, -X0 beeing the default mode:

-X0
The criterions are (in order of importance):
* less distorted scalefactor bands
* the sum of noise over the thresholds is lower
* the total noise is lower

-X1
The actual is better if the maximum noise over all scalefactor bands is less than the best so far .

-X2
The actual is better if the total sum of noise is lower than the best so far.

-X3
The actual is better if the total sum of noise is lower than the best so far and the maximum noise over all scalefactor bands is less than the best so far plus 2db.

-X4
Not yet documented.

-X5
The criterions are (in order of importance):
* the sum of noise over the thresholds is lower
* the total sum of noise is lower

-X6
The criterions are (in order of importance):
* the sum of noise over the thresholds is lower
* the maximum noise over all scalefactor bands is lower
* the total sum of noise is lower

-X7
The criterions are:
* less distorted scalefactor bands
or
* the sum of noise over the thresholds is lower

ID3 tags

LAME is able to embed ID3 v1, v1.1 or v2 tags inside the encoded MP3 file. This allows to have some usefull information about the music track included inside the file. Those data can be read by most MP3 players.

Lame will smartly choose wich tags to use. It will add ID3 v2 tags only if the input comments won't fint in v1 or v1.1 tags, ie if they are more than 30 characters. In this case, both v1 and v2 tags will be added, to ensure reading of tags by MP3 players wich are unable to read ID3 v2 tags.

ID3 comments switches	**parameters**
`--tt "title"`	title of song
`--ta "artist"`	artist who did the song
`--tl "album"`	album where it came from
`--ty "year"`	year in which the song/album was made
`--tc "comment"`	additional info
`--tn "track"`	track number of the song on the CD (1 to 255, creates an ID3 v 1.1 tag)
`--tg "genre"`	genre of song (name or number)

ID3 behaviour switches	
`--add-id3v2`	force addition of version 2 tag
`--id3v1-only`	add only a version 1 tag
`--id3v2-only`	add only a version 2 tag
`--space-id3v1`	pad version 1 tags with spaces instead of nulls
`--pad-id3v2`	pad version 2 tags with extra 128 bytes
`--genre-list`	print alphabetically sorted ID3 genre list and exit

The following genres are supported:

00 - Blues
01 - Classic Rock
02 - Country
03 - Dance
04 - Disco
05 - Funk
06 - Grunge
07 - Hip-Hop
08 - Jazz
09 - Metal
10 - New Age
11 - Oldies
12 - Other
13 - Pop
14 - R&B
15 - Rap
16 - Reggae
17 - Rock
18 - Techno
19 - Industrial
20 - Alternative
21 - Ska
22 - Death Metal
23 - Pranks
24 - Soundtrack
25 - Euro-Techno
26 - Ambient
27 - Trip-Hop
28 - Vocal
29 - Jazz+Funk
30 - Fusion
31 - Trance
32 - Classical
33 - Instrumental
34 - Acid
35 - House
36 - Game
37 - Sound Clip
38 - Gospel
39 - Noise
40 - Alternative Rock
41 - Bass
43 - Punk
44 - Space
45 - Meditative
46 - Instrumental Pop
47 - Instrumental Rock
48 - Ethnic
49 - Gothic

100 - Humour
101 - Speech
102 - Chanson
103 - Opera
104 - Chamber Music
105 - Sonata
106 - Symphony
107 - Booty Bass
108 - Primus
109 - Porn Groove
110 - Satire
111 - Slow Jam
112 - Club
113 - Tango
114 - Samba
115 - Folklore
116 - Ballad
117 - Power Ballad
118 - Rhytmic Soul
119 - Freestyle
120 - Duet
121 - Punk Rock
122 - Drum Solo
123 - Acapella
124 - Euro-House
125 - Dance Hall
126 - Goa
127 - Drum & Bass
128 - Club-House
129 - Hardcore
130 - Terror
131 - Indie
132 - BritPop
133 - Negerpunk
134 - Polsk Punk
135 - Beat
136 - Christian Gangsta
137 - Heavy Metal
138 - Black Metal
139 - Crossover
140 - Contemporary C
141 - Christian Rock
142 - Merengue
143 - Salsa
144 - Thrash Metal
145 - Anime
146 - JPop
147 - SynthPop

50 - Darkwave
51 - Techno-Industrial
52 - Electronic
53 - Pop-Folk
54 - Eurodance
55 - Dream
56 - Southern Rock
57 - Comedy
58 - Cult
59 - Gangsta
60 - Top 40
61 - Christian Rap
62 - Pop/Funk
63 - Jungle
64 - Native US
65 - Cabaret
66 - New Wave
67 - Psychadelic
68 - Rave
69 - Showtunes
70 - Trailer
71 - Lo-Fi
72 - Tribal
73 - Acid Punk
74 - Acid Jazz
75 - Polka
76 - Retro
77 - Musical
78 - Rock & Roll
79 - Hard Rock
80 - Folk
81 - Folk-Rock
82 - National Folk
83 - Swing
84 - Fast Fusion
85 - Bebob
86 - Latin
87 - Revival
88 - Celtic
89 - Bluegrass
90 - Avantgarde
91 - Gothic Rock
92 - Progressive Rock
93 - Psychedelic Rock
94 - Symphonic Rock
95 - Slow Rock
96 - Big Band
97 - Chorus
98 - Easy Listening
99 - Acoustic

History

Starting with LAME 3.0:
red = features and bug fixes which effect quality
blue = features and bug fixes which effect speed
black = usability, portability, other

LAME 3.93alpha (CVS)

LAME 3.92 April 14 2002

- Alexander Leidinger: add non linear psymodel (compile time option, disabled by default), workaround a bug in gcc 3.0.3 (compiler options, based upon suggestions from various people, see archives and changelog for more)
- Steve Lhomme: ACM wrapper (MS-Windows codec)
- Steve Lhomme: less memory copying on stereo (interleaved) input
- Takehiro Tominaga: Inter-channel masking, enables with --interch x option
- For buggy versions of gcc compiler (2.96*), back off on some of the advanced compiler options

LAME 3.91   December 29 2001

- Darin Morrison: Bugfix for --alt-preset (for content with low volume, clean vocals), only important for the "fast standard" preset
- Alexander Leidinger:
  - add some missing files to the distribution
  - add --alt-preset to the man page

LAME 3.90   December 21 2001

- Many small improvements and bug fixes not added to history
- John Dahlstrom: more fine tuning on the auto adjustment of the ATH
- Robert Hegemann: small speed and quality improvements for the old VBR code (--vbr-old).
- Robert Hegemann: some short block bug fixes
- Robert Hegemann: Big improvements to --vbr-mtrh, now encodes much more frequencies over 16khz
- Robert Hegemann: --vbr-new code disabled (outdated and lower quality) and replaced with --vbr-mtrh (Both --vbr-new and --vbr-mtrh now default to mtrh)
- Robert Hegemann: reordering of --longhelp to give more information, --extrahelp dropped
- Darin Morrison: Totally revamped and extremely high quality unified preset system and other general quality improvements now available with --alt-presets:
    - some improvements to psychoacoustics (vast improvements over default L.A.M.E. modes) when --alt-preset is used including:
        - Improved tuning of short block usage.
        - Improved quantization selection usage (the -X modes), now adapts between appropriate modes on the fly. Also helps on "dropout" problems and with pre-echo cases.
        - Improved joint stereo usage. Thresholds are better tuned now and fix some "dropout" problems L.A.M.E. suffers from on clips like serioustrouble.
        - Improved noise shaping usage. Now switches between noise shaping modes on the fly (toggles -Z on and off when appropriate) which allows lower bitrates but without the quality compromise.
        - Clips vastly improved over default L.A.M.E. modes (vbr/cbr/abr, including --r3mix): castanets, florida_seq, death2, fatboy, spahm, gbtinc, ravebase, short, florida_seq, hihat, bassdrum, 2nd_vent_clip, serioustrouble, bloodline, and others. No degraded clips known.
        - VBR bitrates are now more "stable" with less fluctuation -- not dipping too low on some music and not increasing too high unnecessarily on other music. "--alt-preset standard" provides bitrates roughly within the range of 180-220kbps, often averaging close to 192kbps.
    - --alt-presets replace the --dm-presets and "metal" preset is removed and replaced with generic abr and cbr presets.
    - --alt-preset extreme (note the 'e') replaces xtreme to help eliminate some confusion
    - --alt-preset vbr modes now have a fast option which offers almost no compromise in speed.
    - --alt-preset standard (and "fast standard") are now much lower in bitrate, matching --r3mix with an overall average, though offering higher quality especially on difficult test samples.
    - --alt-presets are no longer just "presets" as in a collection of switches, instead they are now quality "modes" because of special code level tunings (those mentioned above).
    - Use --alt-preset help for more information.
- Roel VdB: more tuning on the --r3mix preset
- Jon Dee, Roel VdB: INFO tag
- Alexander Leidinger, mp3gain@hotmail.com: added --scale-l and --scale-r to scale stereo channels independantly
- Takehiro Tominaga: new noise shaping mode, offering more "cutting edge" shaping according to masking, enabled via -q0
- Mark Taylor: More work on --nogap
- Gabriel Bouvigne: Small changes to abr code for more accurate final bitrate
- Gabriel Bouvigne, mp3gain@hotmail.com: Preliminary ReplayGain analysis code added (not functional yet)
- Gabriel Bouvigne, Alexander Leidinger: Documentation updates
- John Dahlstrom, DSPguru@math.com: floating point interface function in the Windows DLL

LAME 3.89beta   July 5 2001

- John Stewart: long filename support for Win9x/NT.
- Takehiro Tominaga: LAME can calculate the CRC of VBR header, so now "lame -pv" works fine.
- Robert Hegemann: Improvements of the new VBR code (--vbr-mtrh).
- Robert Hegemann: New VBR code (--vbr-mtrh) is now defaulted to get more feedback. The VBR speed is now on par with CBR. We will use the old VBR code in the release.
- Gabriel Bouvigne: Change of the maximum frame size limit. LAME should now be more friendly with hardware players.
- Gabriel Bouvigne: Size of VBR is now more balanced according to the -V value.
- Alexander Leidinger: Finished the implementation of the set/get functions.
- John Dahlstrom: LAME now handles 24bits input
- Mark Taylor: bugs in lame --decode causing truncation of mp3 file fixed
- Mark Taylor: preliminary --nogap support
- "Final" API completed: shared library safe! This API is frozen and should be backwords compatiable with future versions of libmp3lame.so, but we will continue to add new functionality.

LAME 3.88beta   March 25 2001

- A lot of work that was never added to the History!
- Frank Klemm and Gabriel Bouvigne: New ATH formula. Big improvement for high bitrate encodings.
- Takehiro Tominaga: Temporal masking
- Gabriel Bouvigne/Mark Taylor: auto adjustment of ATH
- Robert Hegemann: Better outer_loop stopping criterion. Enabled with -q2 or better.
- Robert Hegemann/Naoki Shibata: slow/carefull noise shaping. -q3..9: amplify all distorted bands. -q2: amplify distorted bands within 50%. -q1-0: amplify only most distorted band at each iteration.
- Takehiro Tominaga: Interframe, shortblock temporal masking.
- Takehiro Tominaga: LAME restructured into a shared library and front end application. Slight changes to the API. More changes are coming to turn LAME into a true shared library (right now you have to recompile if you upgrade the library :-(
- Naoki Shibata:
    - improvements to psychoacoustics (--nspsytune)
    - BUG in long block pre echo control fixed (some out of range array access in M/S psychoacoustics)
- Ralf Kempkens: Visual Basic Script for lame, suggested to put it on your Windows Desktop and you can drag'n'drop Waves to encode on it.
- Alexander Stumpf: improved lame.bat for 4Dos users
- Mark Taylor: Several bugs fixed in the resampling code.
- Frank Klemm, Robert Hegemann: added assembler code for CPU feature detection on runtime (MMX, 3DNow, SIMD)
- Takehiro Tominaga: 3DNow FFT code.
- Florian Bome, Alexander Leidinger: more work on configure stuff
- Alexander Leidinger: automake/libtool generated Makefiles and TONS of other work.
- Alexander Leidinger: Much work towards shared library style API.
- Anonymous: New more efficient RTP code.
- Mark Taylor: psycho-acoustic data now computed for all scalefactor bands (up to 24 kHz)
- Mark Taylor, Takehiro Tominaga: All ISO table data replaced by formulas - should improve MPEG2.5 results for which we never had correct table data.

LAME 3.87alpha September 25 2000

- Mark Taylor: Bug fixed in LAME/mpglib error recovery when encountering a corrupt MP3 frame during *decoding*.
- Albert Faber: added LayerI+II decoding support
- Frank Klemm: added improved CRC calculation
- Frank Klemm: substantial code cleanup/improvements
- Robert Hegemann: Bug fixes
    - in huffman_init, could lead to segmentation faults (only in rare cases, most likely at lower sample rates)
    - M/S switching at lower sample rates (the fact there is no 2nd granule was ignored)
- Robert Hegemann: speed up in VBR
- Jarmo Laakkonen: Amiga/GCC settings for Makefile.unix.
- Magnus Holmgren: README and Makefile for (free) Borland C++ compiler. Will also compile lame_enc.dll, but this is untested.
- Florian Bome: LAME finally has a ./configure script!!

LAME 3.86beta August 6 2000

- Christopher Wise: A makefile for DJGPP, the DOS version of gcc. Now most windows users should be able to compile LAME with minimal effort.
- Robert Hegemann: old VBR: fixed some bugs and Takehiro's scalefac_scale feature (not yet on by default.) older LAME versions did not allow to spent more than 2500 bits of 4095 possible bits to a granule per channel, now fixed.
- Robert Hegemann: new VBR: analog silence treatment like in old VBR
- William Welch: Improved options for Linux/Alpha gcc and ccc compilers in Makefile.
- Mathew Hendry: setting appropriate CRC bit for additional Xing-VBR tagging frame
- Don Melton: added ID3 version 2 TAG support
- John Dahlstrom: fixed bug allowing timing information (for status in command line encoder) to overflow.
- Tamito KAJIYAMA, Fixed several bugs in the LAME/Vorbis interface.
- Mark Taylor: lame --decode will recognize Album ID tags
- Naoki Shibata: Additive masking and other improvements to psycho acoustics. (not yet on by default)

LAME 3.85beta   July 3 2000

- Takehiro Tominaga: mid/side stereo demasking thresholds updated.
- Takehiro Tominaga: New short block MDCT coefficient data structure. Should allow for future speed improvements.
- Robert Hegemann: fixed bug in old VBR routine, the --noath mode messed up the VBR routine resulting in very large files
- Robert Hegemann: found bugs in some sections when using 32 bit floating point. Default is now back to 64bit floating point.
- Takehiro Tominaga: Modified PE formula to use ATH.
- S.T.L.: README.DJGPP - instructions for compiling LAME with DJGPP, the dos version of gcc.

LAME 3.84beta June 30 2000

- Mark Weinstein: .wav file output (with --decode option) was writing the wrong filesize in the .wav file. Now fixed.
- Mark Taylor: (optional) Vorbis support, both encoding and decoding. LAME can now produce .ogg files, or even re-encode your entire .ogg collection into mp3.   (Just kidding: it is always a bad idea to convert from one lossy format to another)
- ?: Bug fixed causing VBR to crash under windows.   (pretab[] array overflow)
- Sergey Sapelin: Another bug found in the mpg123 MPEG2 tables. Now fixed for the mpg123 based decoder in LAME.
- Marco Remondini: VBR histogram works in win32. compile with -DBRHIST -DNOTERMCAP
- Takehiro Tominaga: LAME CBR will now use scalefac_scale to expand the dynamic range of the scalefactors.
- Iwasa Kazmi: Library improvements: exit()'s, printf, fprintf's are being replaced by interceptable macros.

LAME 3.83beta May 19 2000

- Mark Taylor: Bug in buffering routines: in some cases, could cause MDCT to read past end of buffer. Rare in MPEG2, even more rare for MPEG1, but potentially serious!
- Mark Taylor: MDCT/polyphase filterbank was not being "primed" properly. Does not effect output unless you set the encoder delay lower than the default of 576 samples.
- Mark Taylor: "vdbj" and "Caster" found several VBR bugs (now fixed):   1. Analog silence detection only checked frequencies up to 16 kHz. 2. VBR mode could still somehow avoid -F mode. 3. VBR mode would ignore noise above 16 kHz (scalefactor band 22), Now calc_noise1 will compute the noise in this band when in VBR mode. Not calculated in CBR mode since CBR algorithm has no way of using this information.
- Mark Taylor: scalefactor band 22 info (masking(=ATH), noise and energy) now displayed in frame analyzer.
- VBR code ATH tuning was disabled by accident in 3.81, now fixed.
- Mark Taylor: lame --decode will produce .wav files. (oops - size is off by a factor of 4)

LAME 3.82beta   May 11 2000

- Robert Hegemann: Fixed bug in high bitrate joint stereo encodings.
- Naoki Shibata: new long block MDCT routine

LAME 3.81beta May 8 2000

- all ISO code removed!
- Takehiro Tominaga and Naoki Shibata: new window subband routines.
- Naoki Shibata: Bug fix in mpglib (decoding) lib: in some cases, MDCT coefficients from previous granule was incorrectly used for the next granule.
- ISO 7680 bit buffer limitation removed. It can be reactivated with "--strictly-enforce-ISO" Please report any trouble with high bitrates.

LAME 3.80beta May 6 2000

- Takehiro Tominaga: more efficient and faster huffman encoding!
- Takehiro Tominaga and Mark Taylor: much improved short block compression!
- Tomasz Motylewski and Mark Taylor: MPEG2.5 now supported!
- Mark Taylor: incorporated Takehiro's bitstream.c! bitstream.c used by default, but old ISO bitstream code can also be used.
- Scott Manley and Mark Taylor: good resampling routine finaly in LAME. uses a 19 point FIR filter with Blackman window. Very slow for non integer resampling ratios.
- Iwasa Kazmi: fixed SIGBUS error: VBR and id3 tags were using data after it was free()'d.
- Robert Hegemann: Improved VBR tuning. #define RH_QUALITY_CONTROL and #RH_SIDE_VBR now the defaults.
- Robert Hegemann: LAME version string now added to ancillary data.
- Kimmo Mustonen: VBR histogram support for Amiga.
- Casper Gripenberg: VBR stats (but not histogram) for DOS verson.
- Robert Hegemann: rare VBR overflow bug fixed.
- Zack: -F option strictly enforces the VBR min bitrate. Without -F, LAME will ignore the minimum bitrate when encoding analog silence.
- Shawn Riley: User can now specify a compression ratio (--comp <arg>) instead of a bit rate. Default settings based on a compression ratio of 11.0
- Mark Taylor: free format bitstreams can be created with --freeformat, and specify any integer bitrate from 8 to 320kbs with -b.
- Mark Taylor: lame be used as a decoder (output raw pcm only): lame --decode input.mp3 output.pcm

LAME 3.70   April 6 2000

- "LAME 3.69beta becomes LAME 3.70 "stable"

LAME 3.69beta   April 6 2000

- "spahm": default mode selection bug fixed. In some cases, lame was defaulting to regular stereo instead of jstereo when the user did not specify a mode.

LAME 3.68beta April 4 2000

- Mark Taylor: mono encoding bug in DLL fixed.
- Ingo Saitz: bug in --cwlimit argument parsing fixed.
- Scott Manly: bug in 4-point resample code fixed.

LAME 3.67beta March 27 2000

- Robert Hegemann: jstereo now enabled for MPEG2 encodings
- Mark Taylor: old M/S stereo mode which used L/R maskings has been removed.
- Mark Taylor: Xing MPEG2 VBR headers now working.
- Mark Taylor: When quantized coefficients are all 0 in a band, set scalefactors to 0 also to save a few bits.
- Ingo Saitz: Problems with framesize calculation when using -f fast-math option fixed.

LAME 3.66beta March 21 2000

- Bug fixes in BladeEnc DLL, possible click in last mp3 frame, VBR historgram display, byteswapping option, ASM quantize routines work for both float and double.

LAME 3.65beta   March 17 2000

- Enabled ASM version of quantize_xrpow() - accidently disabled in lame3.64.

LAME 3.64beta March 16 2000

- Don Melton: id3v1.1 tags & id3 bugfixes
- Gabriel Bouvigne: L/R matching block type fix
- Bug fixed which was allowing quantized values to exceed the maximum when not using -h
- Mark Taylor: Fitlers based on polyphase filterbank. should be slightly better since the responce is independent of the blocktype, and they are slightly faster.
- Mark Taylor: API: the API changed slightly - and this should be the final version. There is a new routine: lame_encode_buffer() which takes an arbritray sized input buffer, resamples & filters if necessary, encodes, and returns the mp3buffer. There are also several new #defines, so it is possible to compile a simple encoding library with no decoding or file I/O or command line parsing. see the file API for details.
- Mark Taylor: MSVC stuff: lame.exe (with and without the frame analyzer) and the CDex lame_enc.dll should compile under MSVC. The MSVC5 project files may need some tweaking. In particular, you need to make sure LAMEPARSE, LAMESNDFILE and HAVEMPGLIB are defined. (and HAVEGTK for the GTK stuff).

LAME 3.63beta February 20 2000

- Robert Hegemann: FPE with -h fixed?
- Mathey Hendry: FPE error catching for Cygwin, FPE fix for vbr mode and output to /dev/null
- Jeremy Hall: Fixed problems with input files where the number of samples is not known.
- Mathew Hendry: ASM quantize_xrpow() for GNU i386
- Wilfried Behne quantize_xrpow ()for PowerPC and non-ASM
- Takehiro Tominaga: GOGO FFTs (not yet used?)

LAME 3.62beta   February 9 2000

- Iwasa Kazmi: frame analyzer short block display of single subblocks (press 1,2 or 3)
- Ingo Saitz: --help option added, with output to stdout
- Alfred Weyers: short block AAC spreading function bug fixed
- Takehiro Tominaga: new scalefac data structure - improves performance!
- Lionel Bonnet: Bug fixed in MPEG2 scalefactor routine: scalefactors were being severly limited.
- Takehiro Tominaga: faster FFT routines from. These routines are also compatible with the GOGO routines, in case someone is interested in porting them back to LAME.
- Sigbjørn Skjæret, Takehiro Tominaga: faster pow() code.
- Joachim Kuebart: Found some unitialized variables that were effecting quality for encodings which did not use the -h option (now fixed).
- Mark Taylor: More modularization work.   It is now possible to use LAME as a library where you can set the encoding parameters directly and do your own file i/o.   The calling program is now it's own mp3 output. For an example of the LAME API, see main.c, or mp3rtp.c or mp3x.c. These can all be compiled as stand alone programs which link with libmp3lame.a.
- Felix vos Leitner: mp3rtp fixes. mp3rtp is a standalone program which will encode and stream with RTP.
- Robert Hegemann: Information written to stderr displaying exactly which type of lowpass filter (if any) is being used.
- Iwasa Kazmi: mpglib (the mpg123 decoder) scsfi decoding fixes.
- Takehiro Tominaga: More mpglib scsfi decoding fixes.

LAME 3.61beta January 14 2000

- Mark Taylor: Fixed bug with lowpass filters when using VBR with a 64kbs or lower min bitrate setting.
- Takehiro Tominaga: more efficient huffman encoding splitting.

LAME 3.60beta   January 9 2000

- Mark Taylor: Distribution now comes with self test. Needs work to be automated, see 'make test' in Makefile.
- Mark Taylor: AAC spreading function now the default
- Gabriel Bouvigne: updated HTML docs
- Felix von Leitner: compute correct file length from Xing header (if present) when input file is a mp3 file
- Felix von Leitner: mp3rtp (standalone) program now included. Not yet tested. mp3rtp ip:port:ttl <infile>   /dev/null will stream directly to ip:port using RTP.

LAME 3.59beta January 4 2000

- Takehiro Tominaga: --noath option. Disables ATH maskings.
- Gabriel Bouvigne: updated HTML docs.
- Iwasa Kazmi: makefile fixes
- Mark Taylor: Fixed bug where first frame of data was always overwritten with 0's. Thanks to 'gol'
- Mark Taylor: bug fixes in mid/side masking ratios (thanks to Menno Bakker)
- Mark Taylor: replaced norm_l, norm_s table data with formulas.

LAME 3.58beta December 13 1999

- Segher Boessenkool: More accurate quantization procedure! Enabled with -h.
- Mathew Hendry, Acy Stapp and Takehiro Tominaga: ASM optimizations for quantize_xrpow and quantize_xrpow_ISO.
- Chuck Zenkus: "encoder inside" logo on web page
- Mark Taylor: a couple people have asked for this.   Allow LAME to overide VBR_min_bitrate if analog_silence detected.   Analog_silence defined a la Robert: energy < ATH.
- An Van Lam: Valid bitrates were being printed for layer 2, not layer 3!
- Ethan Yeo: Makefile.MSVC updated
- Mark Stephens: updated all MSVC project files
- Robert Hegemann: lowpass and highpass filters can be enabled with --lowpass, --highpass
- Mark Taylor: MS switching is now smoother: ms_ratio average over 4 granules
- Takehiro Tominaga: Scalefactor pre-emphasis fixed (and now turned back on)
- Takehiro Tominaga: Bug in M/S maskings: switch to turn on stereo demasking code was buggy.

LAME 3.57beta November 22 1999

- Sigbjørn Skjæret, patch to allow encoding from 8bit input files when using LIBSNDFILE
- Mark Taylor: Automatic downsampling to nearest valid samplerate.
- Mark Taylor: Scalefactor bands demarked on MDCT plot in frameanalyzer
- Mark Taylor: Scalefactor preemphasis disabled for now.   The algorithm was often doing more harm than good.

LAME 3.56beta November 19 1999

- Kimmo Mustonen: portabilty code cleanup.
- Vladimir Marek: id3 genre patch.
- Conrad Sanderson: new applypatch script.
- Mark Taylor: Initial window type now "STOP_TYPE" to reduce initial attenuation. This is needed because the new encoder delay is so short. With a NORM_TYPE, the first 240 samples would be attenuated.
- Mark Taylor: Padding at end of file now adjusted (hopefully!) to produce as little padding as possible while still guarantee all input samples are encoded.
- Takehiro Tominaga: Reduced shortblock extra bit allocation formulas by 10% since new huffman coding is at least 10% more efficient.

LAME 3.55beta November 11 1999

- Albert Faber: updated BladeEnc.dll
- Mark Taylor: Simple lowpass filter added to linear downsampling routine.
- Nils Faerber: updated man page.
- Mark Taylor: All floating point variables are delcared FLOAT or FLOAT8. Change the definition of FLOAT8 in machine.h to run at 32bit preceision.
- Mark Taylor: Bug (introduced in 3.54beta) in stereo->mono downsampling fixed.

LAME 3.54beta November 8 1999

- Mark Taylor: Encoder delay is now 48 samples. Can be adjusted to 1160 to sync with FhG (see ENCDELAY in encoder.h) This is kind of amazing, since if Takehiro put his MDCT/filterbank routine in a decoder, we could have a total delay of only 96 samples.
- Mark Taylor: More inconstancies found and fixed in MPEG2 tables.
- Mark Taylor: Resampling from an MP3 input file now works. But we still dont have a lowpass filter so dont expect good results.

LAME 3.53beta November 8 1999

- Takehiro Tominaga: Fixed MPEG2 problem in new MDCT routines. Takehiro's combined filterbank/MDCT routine is now the default. Removes all buffering from psymodel.c and the filterbanks/MDCT routines.

LAME 3.52beta November 8 1999

- By permission of copyright holders of all GPL code in LAME, all GPL code is now released under a modified version of the LGPL (see the README file)
- By popular demand, all C++ comments changed to C style comments
- Mark Taylor: Linear resampling now works. Use --resample to set an output samplerate different from the input samplerate. (doesn't seem to work with mp3 input files, and there is no lowpass filter, so dont expect good results just yet)
- Takehiro Tominaga: Faster Huffman encoding routines

The following changes are disabled because of MPEG2 problems. But to try them, set MDCTDELAY=48 in encoder.h, instead of MDCTDELAY=528.:

- Takehiro Tominaga: New MDCT routines with shorter delay (48 samples instead of 528) and even faster than the old routines.
- Takehiro Tominaga: Removed extra buffering in psymodel.c

LAME 3.51 November 7 1999

- Takehiro Tominaga: Bug in quantize.c absolute threshold of hearing calculation for non-44.1 kHz input files.

LAME 3.50 November 1 1999

- LAME 3.37beta becomes official LAME 3.50 release

LAME 3.37beta November 1 1999

- Lionel Bonnet: Found severe bug in MPEG2 Short block SNR.
- Sergey Sapelin: VBR Toc improvement.
- Sergey Dubov: fskip() routine
- Conrad Sanderson: replacement for filterbank.c.   Not much faster but amazingly simpler.

LAME 3.36beta October 25 1999

- Albert Faber: more MSVC and BladeDLL updates
- Kimmo Mustonen: Much code cleanup and Amiga updates
- Anton Oleynikov: Borland C updates
- Mark Taylor: More stdin fixes: For some reason, forward fseek()'s would fail when used on pipes even though it is okay with redirection from "<". So I changed all the forward fseek()'s to use fread(). This should improve stdin support for wav/aiff files. If you know the input file is raw pcm, you can still use the '-r' option to avoid *all* seeking of any kind.

LAME 3.35beta October 21 1999

- Leonid Kulakov: Serious bug in MPEG2 scalefactor band tables fixed.
- Portability patches from: Anton Oleynikov, Sigbjørn Skjæret, Mathew Hendry, Richard Gorton
- Alfred Weyers: compiler options, updated timestatus.
- Albert Faber: BladeDll and other updates (new machine.h).
- Monty: updated Makefile to fix gcc inline math bug.

LAME 3.34beta October 12 1999

- Mark Taylor: Bug fixed: minimum bitrate in VBR mode could be ignored for a few frames.
- Mark Taylor: New (minor) VBR tunings.
- Tim Ruddick: New wav/aiff header parsing routines. Better parsing and fewer fseek()'s.
- Anton Oleynikov: patches to work with Borland C
- Gabriel Bouvigne: Experimental voice option enabled with --voice

LAME 3.33beta October 11 1999

- Robert Hegemann: RH VBR mode now the default and only VBR mode. The new code will always quantize to 0 distortion and the quality is increased by reducing the masking from the psy-model. -X0 is still the default for now.
- Robert Hegemann: new -X5 mode
- Mathew Hendry: New timing code, removes the need for HAVETIMES
- Mathew Hendry: assembler quantize_xrpow for Windows
- Iwasa Kazmi: stdin/stdout patch for Windows
- Mark Taylor: New option: "--athonly" will ignore the psy-model output and use only the absolute threshold of hearing for the masking.

LAME 3.32beta October 8 1999

- Takehiro Tominaga: faster long block spreading function convolution for non 44.1 kHz sampling frequencies, and faster short block spreading function convolution for all sampling frequencies.
- Takehiro Tominaga: Completly rewritten huffman table selection and count_bits(). More efficient table selection results in many more bits per frame.
- Takehiro Tominaga: More efficient scalefac compress setting.
- Mike Cheng: new calc_noise2()
- Alfred Weyers: patch for timestatus() seconds rollover

LAME 3.31beta September 28 1999

- Albert Faber: updated his BladeDLL code. This allows LAME to be compiled into a BladeEnc compatiable .dll.
- Mike Cheng: faster l3psycho_ener() routine.
- Sigbjørn Skjæret: more code cleanup.

LAME 3.30beta September 27 1999

- Conrad Sanderson: ID3 tag code added (type 'lame' for instructions)
- new mdct.c from Mike Cheng (no faster, but much cleaner code)
- Mathew Hendry: Microsoft nmake makefile and a couple other changes for MSVC
- More modulization work: One input sound file interface handles mp3's, uncompressed audio, with or without LIBSNDFILE. Fixes (hopefully) a bunch of file I/O bugs introduced in 3.29 (Mark Taylor)
- LAME will now print valid samplerate/bitrate combinations (Mark Taylor)
- stdin/stdout fix for OS/2 (Paul Hartman)
- For mp3 input files, totalframes estimated based on filesize and first frame bitrate. (Mark Taylor)
- Updated all functions with new style prototypes. (Sigbjørn Skjæret)

LAME 3.29beta September 21 1999

- Bug in bigv_bitcount fixed. Loop.c was overestimating the number of bits needed, resulting in wasted bits every frame. (Leonid A. Kulakov)
- Bug in *_choose_table() fixed   These routines would not sellect the optimal Huffman table in some cases.   (Leonid A. Kulakov)
- Tuning of ATH normalization (macik)
- Removed unused variables and fixed function prototypes (Sigbjørn Skjæret)
- Sami Farin sent a .wav file that LAME built in support choked on.   I added a slightly more sophisticated wav header parsing to handle this, but if you have trouble, use libsndfile.
- Resampling hooks and options added. Buffering and resampling routines need to be written.
- LAME will now take an mp3 file as input. When resampling code is working, LAME will be able to (for example) convert a high bitrate stereo mp3 to a low bitrate mono mp3 for streaming.

LAME 3.28beta September 15 1999

- Serious bug fixed in high frequency MDCT coefficients. Huffman coding was reversing the order of the count1 block quadruples.   (Leonid A. Kulakov)
- nint() problems under Tru64 unix fixed and preprocessor variable HAVE_NINT removed. (Bob Bell)
- Compiler warning fixes and code cleanup   (Sigbjørn Skjæret, Lionel Bonnet)
- USAGE file now includes suggestions for downsampling. For low bitrate encodings, proper downsampling can give dramatically better results. (John Hayward-Warburton)

LAME 3.27beta September 12 1999

- Several bugs in encode.c and l3bitstream.c fixed by Lionel Bonnet.
- Bugs in new VBR (#define RH) formula for mono input file and mid/side encoding fixed.

LAME 3.26beta September 10 1999

- The "-m m" option (mono .mp3 file) will automatically mix left and right channels if the input file is stereo. (Alfred Weyers)
- New quant_compare algorithm (method for deciding which of two quantizations is better) enabled with -X4 (Greg Maxwell)
- New mid/side VBR bit allocation formula. Mid channel bits are set by the quality requirements, and then the side channel uses a reduced number of bits (in a proportion coming from the fixed bitrate code). This might not be optimal, but it should be pretty good and no one knows what the optimal solution should be. (Greg Maxwell)
- New VBR (#define RH) tunings based on detailed listening tests by Macik and Greg Maxwell.
- Sigbjørn Skjæret fixed several compiler warnings (which turned out to be potential bugs)
- Takehiro Tominaga fixed a low bitrate bug in reduce_side()
- Alfred Weyers fixed some buffer overflows.
- New ATH (absolute threshold of hearing) formula replaces buggy ISO code, and adds analog silence treatment (removal of coefficients below below ATH).   These are turned on by default but have not been fully tested. (Robert Hegemann)
- Bug in short block spreading function fixed. (Robert Hegemann)

LAME 3.25beta August 22 1999

- Sigbjørn Skjæret fixed a zero byte malloc call. This bug was introduced in 3.24 and causes problems on non Linux systems.
- Bit allocation routines would sometimes allocate more than 4095 bits to one channel of one granule. A couple of people reported problems that might be caused by this, especially at higher bitrates.
- Nils Faerber updated the man page and fixed many of the compiler warnings.

LAME 3.24beta   August 15 1999

- This release contains the following new code (for developers) which is disabled by default:
- Robert Hegemann: Completely overhauled VBR code. Now computes exact number of bits required for the given qualty and then quantized with the appropriate bitrate.
- Several new quantization quality measures.

LAME 3.23beta August 8 1999

- Very nice continuously updated VBR histogram display from Iwasa Kazmi. (disabled with --nohist).
- More modulerization work. The encoding engine can now be compiled into libmp3lame, but the interface is awkward.
- Bug fixed in FFT Hann window formula (Leonid A. Kulakov).
- New LAME logo on the download page. Created by Chris Michalisles.
- Several VBR algorithm improvements from Robert Hegemann. New quantization noise metrics and VBR quality measure takes into account mid/side encoding. Should produce smaller files with the same quality, especially when using jstereo.

LAME 3.22beta July 27 1999

- Downsampling (stereo to mono) bug with MPEG2 fixed. (Mike Oliphant)
- Downsampling now merges L & R channels - before it only took the L channel.
- More modularization and code cleanup from Albert Faber and myself.
- Input filesize limit removed for raw pcm input files. For other file types, LAME will still only read the first 2^32 samples, (27 hours of playing time at 44.1 kHz).

LAME 3.21beta July 26 1999

- Correct Mid/Side masking thresholds for JSTEREO mode! This is enabled with -h. It makes LAME about 20% slower since it computes psycho-acoustics for L,R Mid and Side channels.
- "Analog silence" threshold added. Keeps VBR from upping the bitrate during very quite passages. (Robert.Hegemann)
- New VBR quality setting from Robert Hegemann. It is based on the idea that distortion at lower bit rates sounds worse than at higher bitrates, and so the allowed distortion (VBR quality setting) is proportional to the bitrate. Because of this, default minimum bitrate is now 32kbs.
- Expermental subblock gain code enabled with -Z.
- New "-r" option for raw pcm input files. With -r, LAME will not do any fseek()'s or look for wav and aiff headers on the input file.
- Bug fixes in mp3x (frame analyzer) for viewing frames near end of the file.
- Bug fixed to allow setting the sampling rate of raw pcm input files.

LAME 3.20beta July 19 1999

- Bug in get_audio.c fixed. Libsndfile wrappers would not compile (Miguel Revilla Rodriguez)
- Nils Faerber found some unitialized variables and some wierd extranous computations in filter_subband, now fixed. This was causing seg faults on some machines.

LAME 3.19beta July 18 1999

- Oops! Robert Hegemann immediatly found a bug in the new (old -Z option) quantization code. calc_noise1 was not returning tot_noise, so non ms-stereo frames were buggy.

LAME 3.18beta July 17 1999

- Many psycho-acoustic bug fixes. Dan Nelson discovered a bug in MPEG2: For short blocks, the code assumes 42 partition bands. MPEG1 sometimes has less, MPEG2 can have more. In MPEG1, this bug would not have effected the output if your compiler initializes static variables to 0 on creation. In MPEG2 it leads to array out-of-bounds access errors. Finally, there was a related bug in MPEG1/MPEG2, short & long blocks where the energy above 16 kHz was all added to partition band 0. (the lowest frequeny partition band!)
- The -Z option (Gabriel Bouvigne's idea of using total quantization noise to choose between two quantizations with the same value of "over") is now the default. I believe this helps remove the trilling sound in Jan's testsignal4.wav. The quality of testsignal2.wav and testsignal4.wav are now better than Xing and getting closer to FhG.
- Bug fixes in frame & sample count for downsampling mode. (ben "jacobs")
- Patches to improve modulization. (ben "jacobs")

LAME 3.17beta July 11 1999

- substantial code cleanup towards goal of making LAME more modular.

LAME 3.16beta July 11 1999

- New tunings of window switching, and better bit allocation based on pe. (Jan Rafaj. improves both testsignal2.wav and testsignal4.wav).
- Bug in mid/side quantization when side channel was zero fixed. (Albert Faber)
- Removed some extranous computations in l3psy.c (Robert Hegemann)
- More detailed timing status info, including hours display. (Sakari Ailus) and percentage indicator (Conrad Sanderson).
- Window_subband and calc_noise1,calc_noise2 speedups. Quantize_xrpow speedup should be significant on non GNU/intel systems. (Mike Cheng)
- Better initial guess for VBR bitrate. Should speed up VBR encoding. (Gabriel Bouvigne)
- More advanced .wav header parsing. fixes bugs involving click in first frame. (Robert.Hegemann)
- Correct filesize and total frame computation when using LIBSNDFILE (ben "jacobs")
- Click in last frame (buffering problem) when using libsndfile fixed.
- Audio I/O code overhauled. There is now a uniform audio i/o interface to libsndfile or the LAME built in wav/aiff routines. All audio i/o code localized to get_audio.c.

LAME 3.15beta

- times()/clock() problem fixed for non-unix OS. (Ben "Jacobs")
- Fixed uninitialized pe[] when using fast mode. (Ben "Jacobs")

LAME 3.13 June 24 1999

- Patches for BeOS from Gertjan van Ratingen.
- Makefile info for OS/2 Warp 4.0 (from dink.org).
- Status display now based on wall clock time, not cpu time.
- mem_alloc no longer allocates twice as much memory as needed (Jan Peman).

3.12pre9

- Updated BLADEDLL code to handle recent changes (Albert Faber).
- Bug fixed in parsing options when not using GTK (Albert Faber).
- MPEG2 Layer III psycho acoustics now working.
- Improved huffman encoding Chris Matrakidis. (10% faster). I dont know how he finds these improvements! LAME with full quality now encodes faster than real time on my PII 266.
- Fixed time display when encoding takes more than 60 minutes.

3.12pre8

- New mid/side stereo criterion. LAME will use mid/side stereo only when the difference between L & R masking thresholds (averaged over all scalefactors) is less then 5db. In several test samples it does a very good job mimicking the FhG encoder.
- Bug in mid/side stereo fixed: independent variation of mid & side channel scalefactors disabled. Because of the way outer_loop is currently coded, when encoding mid/side coefficietns using left/right thresholds, you have to vary the scalefactors simultaneously.
- Bug in side/mid energy ratio calculation fixed. (Thanks to Robert Hegemann)
- Default mode is stereo (not jstereo) if bitrate is chosen as 192kbs or higher. Tero Auvinen first pointed out that FhG seems to think at 160kbs, their encoder is so good it doesn't need jstereo tricks. Since LAME is not as good as FhG, I am going to claim that 192kbs LAME is so good it doens't need jstereo tricks, and thus it is disabled by default.
- WAV header parsing for big-endian machines, and automatic detection of big-endian machines. (Thanks to Sigbjørn Skjæret).
- added 56 sample delay to sync LAME with FhG.
- MP3x (frame analyzer) can now handle MPEG2 streams.

3.12pre7

- MPEG2 layer III now working! lower bit rates (down to 8kbs) and 3 more sampling frequencies: 16000, 22050, 24000Hz. Quality is poor - the psy-model does not yet work with these sampling frequencies.
- Fixed "ERROR: outer_loop(): huff_bits < 0." bug when using VBR.
- bash and sh scripts to run LAME on multiple files now included. (from Robert Hegemann and Gerhard Wesp respectively)
- bug fix in encoding times for longer files from (Alvaro Martinez Echevarria)
- yet another segfault in the frame analyzer fixed.
- ISO psy-model/bit allocation routines removed. This allowed makeframe() to be made much simpler, and most of the complicated buffering is now gone. Eventually I would like the encoding engine to be a stand alone library.

3.12pre6

- Better VBR tuning. Find minimum bitrate with distortion less than the allows maximum. A minimum bit rate is imposed on frames with short blocks (where the measured distortion can not be trusted). A minimum frame bitrate can be specified with -b, default=64kbs.
- LIBSNDFILE support. With libsndfile, LAME can encode almost all sound formats. Albert Faber did the work for this, including getting libsndfile running under win32.
- CRC checksum now working! (Thanks to Johannes Overmann)
- frame analyzer will now work with mono .mp3 files
- more code tweeks from Jan Peman.
- Compaq-Alpha(Linux) fixes and speedups from Nils Faerber.
- Faster bin_search_StepSize from Juha Laukala.
- Faster quantize() from Mike Cheng
- Faster quantize_xrpow() from Chris Matrakidis. xrpow_flag removed since this option is now on by default.
- Fixed .wav header parsing from Nils Faerber.
- Xing VBR frame info header code from Albert Faber. "Xing" and "LAME 3.12" embedded in first frame.
- Bug in VBR bit allocation based on "over" value fixed.

LAME 3.11 June 3 1999

- Almost all warnings (-Wall) now fixed! (Thanks to Jan Peman)
- More coding improvements from Gabriel Bouvigne and Warren Toomey.
- VBR (variable bit rate).   Increases bit rate for short blocks and for frames where the number of bands containing audible distortion is greater than a given value. Much tuning needs to be done.
- Patch to remove all atan() calls from James Droppo.

LAME 3.10 May 30 1999

- Fast mode (-f) disables psycho-acoustic model for real time encoding on older machines. Thanks to Lauri Ahonen who first sent a patch for this.
- New bit reservoir usage scheme to accommodate the new pre-echo detection formulas.
- Tuning of AWS and ENER_AWS pre-echo formulas by Gabriel Bouvigne and myself. They work great! now on by default.
- In jstereo, force blocktypes for left & right channels to be identical. FhG seems to do this. It can be disabled with "-d".
- Patches to compile MP3x under win32 (Thanks to Albert Faber).
- bin_serach_stepsize limited to a quantizationStepSize of -210 through 45.
- outer_loop() will now vary Mid & Side scalefactors independently. Can lead to better quantizations, but it is slower (twice as many quantizations to look at). Running with "-m f" does not need this and will run at the old speed
- Bug in inner_loop would allow quantizations larger than allowed. (introduced in lame3.04, now fixed.)
- Updated HTML documentation from Gabriel Bouvigne.
- Unix man page from William Schelter.
- numlines[] bug fixed. (Thanks to Rafael Luebbert, MPecker author).
- Quantization speed improvements from Chirs Matrakidis.
- When comparing quantizations with the same number of bands with audible distortion, use the one with the largest scalefactors, not the first one outer_loop happened to find.
- Improved defination of best quantization when using -f (fast mode).
- subblock code now working. But no algorithm to choose subblock gains yet.
- Linux now segfaults on floating point exceptions. Should prevent me from releasing binaries that crash on other operating systems.

May 22 1999

- Version 3.04 released.
- Preliminary documentation from Gabriel Bouvigne.
- I wouldn't have thought it was possible, but now there are even more speed improvements from Chris Matrakidis! Removed one FFT when using joint stereo, and many improvements in loop.c.
- "Fake" ms_stereo mode renamed "Force" ms_stereo since it forces mid/side stereo on all frames. For some music this is said to be a problem, but for most music mode is probably better than the default jstereo because it uses specialized mid/side channel masking thresholds.
- Small bugs in Force ms_stereo mode fixed.
- Compaq Alpha fixes from Nathan Slingerland.
- Some new experimental pre-echo detection formulas in l3psy.c (#ifdef AWS and #ifdef ENER_AWS, both off by default. Thanks to Gabriel Bouvigne and Andre Osterhues)
- Several bugs in the syncing of data displayed by mp3x (the frame analyzer) were fixed.
- highq (-h) option added. This turns on things (just one so far) that should sound better but slow down LAME.

May 18 1999

- Version 3.03 released.
- Faster (20%) & cleaner FFT (Thanks to Chris Matrakidis http://www.geocities.com/ResearchTriangle/8869/fft_summary.html)
- mods so it works with VC++ (Thanks to Gabriel Bouvigne, www.mp3tech.org)
- MP3s marked "original" by default (Thanks to Gabriel Bouvigne, www.mp3tech.org)
- Can now be compiled into a BladeEnc compatible .DLL (Thanks to Albert Faber, CDex author)
- Patches for "silent mode" and stdin/stdout (Thanks to Lars Magne Ingebrigtsen)
- Fixed rare bug: if a long_block is sandwiched between two short_blocks, it must be changed to a short_block, but the short_block ratios have not been computed in l3psy.c. Now always compute short_block ratios just in case.
- Fixed bug with initial quantize step size when many coefficients are zero. (Thanks to Martin Weghofer).
- Bug fixed in MP3x display of audible distortion.
- improved status display (Thanks to Lauri Ahonen).

May 12 1999

- Version 3.02 released.
- encoder could use ms_stereo even if channel 0 and 1 block types were different. (Thanks to Jan Rafaj)
- added -k option to disable the 16 kHz cutoff at 128kbs. This cutoff is never used at higher bitrates. (Thanks to Jan Rafaj)
- modified pe bit allocation formula to make sense at bit rates other than 128kbs.
- fixed l3_xmin initialization problem which showed up under FreeBSD. (Thanks to Warren Toomey)

May 11 1999

- Version 3.01 released
- max_name_size increased to 300 (Thanks to Mike Oliphant)
- patch to allow seeks on input file (Thanks to Scott Manley)
- fixes for mono modes (Thanks to everyone who pointed this out)
- overflow in calc_noise2 fixed
- bit reservoir overflow when encoding lots of frames with all zeros (Thanks to Jani Frilander)

May 10 1999

- Version 3.0 released
- added GPSYCHO (developed by Mark Taylor)
- added MP3x (developed by Mark Taylor)
- LAME now maintained by Mark Taylor

November 8 1998

- Version 2.1f released
- 50% faster filter_subband() routine in encode.c contributed by James Droppo

November 2 1998

- Version 2.1e released.
- New command line switch **-a** auto-resamples a stereo input file to mono.
- New command line switch **-r** resamples from 44.1 kHz to 32 kHz [this switch doesn't work really well. Very tinny sounding output files. Has to do with the way I do the resampling probably]
- Both of these were put into the ISO code in the encode.c file, and are simply different ways of filling the input buffers from a file.

October 31 1998

- Version 2.1d released
- Fixed memory alloc in musicin.c (for l3_sb_sample)
- Added new command line switch (-x) to force swapping of byte order
- Cleaned up memory routines in l3psy.c. All the mem_alloc() and free() routines where changed so that it was only done *once* and not every single time the routine was called.
- Added a compile time switch -DTIMER that includes all timing info. It's a switch for the time being until some other people have tested on their system. Timing code has a tendency to do different things on different platforms.

October 18 1998

- Version 2.1b released.
- Fixed up bug: all PCM files were being read as WAV.
- Played with the mem_alloc routine to fix crash under amigaos (just allocating twice as much memory as needed). Might see if we can totally do without this routine. Individual malloc()s where they are needed instead
- Put Jan Peman's quality switch back in. This reduces quality via the '-q ' switch. Fun speedup which is mostly harmless if you're not concerned with quality.
- Compiling with amiga-gcc works fine

October 16 1998

- Version 2.1a released. User input/output has been cleaned up a bit. WAV file reading is there in a very rudimentary sense ie the program will recognize the header and skip it, but not read it. The WAV file is assumed to be 16bit stereo 44.1 kHz.

October 6 1998

- Version 2.1 released with all tables now incorporated into the exe. Thanks to **Lars Magne Ingebrigtseni**(larsi@ifi.uio.no)

October 4 1998 In response to some concerns about the quality of the encoder, I have rebuilt the encoder from scratch and carefully compared output at all stages with the output of the unmodified ISO encoder. Version2.0 of LAME is built from the ISO source code (dist10), and incorporates modifications from myself and the 8hz effort. The output file from LAME v2.0 is *identical* to the output of the ISO encoder for my test file.Since I do not have heaps of time, I left the ISO AIFF file reader in the code, and did not incorporate a WAV file reader.Added section on quality **October 1 1998**

- Updated web page and released LAME v1.0

Up to September 1998

Working on the 8hz source code...

- Patched the 8hz source code
- 45% faster than original source (on my freebsd p166).
    - m1 - sped up the mdct.c and quantize() functions [MDCTD, MDCTD2, LOOPD]
    - m2 - sped up the filter_subband routine using **Stephane Tavenard** 's work from musicin [FILTST]
    - m2 - minor cleanup of window_subband [WINDST2]
    - m2 - Cleaned up a few bits in l3psy.c. Replaced a sparse matrix multiply with a hand configured unrolling [PSYD]
    - m3 - (amiga only) Added in the asm FFT for m68k (based on sources from **Henryk Richter** and **Stephane Tavenard**)
    - m4 - raw pcm support back in
    - m5 - put in a byte-ordering switch for raw PCM reading (just in case)
    - m6 - reworked the whole fft.c file. fft now 10-15% faster.
    - m7 - totally new fft routine. exploits fact that this is a real->complex fft. About twice as fast as previous fastest fft (in m6). (C fft routine is faster than the asm one on an m68k!)
    - m8
        - - Now encodes from stdin. Use '-' as the input filename. Thanks to **Brad Threatt**
        - - Worked out that the 1024point FFT only ever uses the first 6 phi values, and the first 465 energy values. Saves a bunch of calculations.
        - - Added a speed-up/quality switch. Speed is increased but quality is decreased *slightly*. My ears are bad enough not to be able to notice the difference in quality at low settings :). Setting '-q 1' improves speed by about 10%. '-q 100' improves speed by about 26%. Enoding of my test track goes from 111s (at default '-q 0') to 82s (at -q 100). Thanks to **Jan Peman** for this tip.
    - m9 - fixed an error in l3psy.c. numlines[] is overwritten with incorrect data. Added a new variable numlines_s[] to fix this. Thanks again to **Jan Peman**.
    - m10 - Down to 106 seconds by selecting a few more compiler options. Also added a pow20() function in l3loop.c to speed up (ever so slightly) calls to pow(2.0, x)
    - m11
        - No speedups. Just cleaned up some bits of the code.
        - Changed K&R prototyping to 'normal' format. Thanks to **Steffan Haeuser** for his help here.
        - Changed some C++ style comments to normal C comments in huffman.c
        - Removed the #warning from psy_data.h (it was getting annoying!)
        - Removed reference in bitstream.c to malloc.h. Is there a system left where malloc.h hasn't been superceded by stdlib.h?
    - In Progess:
        - my PSYD hack for the spreading functions is only valid for 44.1 kHz - Should really put in a "if freq = 44.1 kHz" switch for it. Someone might want to extend the speedup for 48 and 32 kHz.
        - Putting in Jan Peman's quantanf_init speedup.

Contributors

The following people contributed to the LAME development:

Lauri Ahonen
Sakari Ailus
Tero Auvinen
Alex Ballegooy
Wilfried Behne
Bob Bell
Segher Boessenkool
Florian Bome
Lionel Bonnet
Gabriel Bouvigne
Erik de Castro Lopo
Michael Cheng
John Dahlstrom
Goran Dokic
James Droppo
Sergey Dubov
Alvaro Martinez Echevarria
Joakim Enerstam
Albert Faber
Nils Faerber
Sami Farin
Mikhail M. Fedotov
Joseph Fourier
Jani Frilander
Richard Gorton
Casper Gripenberg
Steffan Haeuser
Jeremy Hall
Paul Hartman
John Hayward-Warburton
Robert Hegemann
Mathew Hendry
Magnus Holmgren
Ben "Jacobs"
Tamito Kajiyama
Mo Katz
Iwasa Kazmi
Ralf Kempkens
Frank Klemm
Joachim Kuebart
Leonid Kulakov
Dmitry Kutsanov
Jarmo Laakkonen
An van Lam

Juha Laukala
Felix von Leitner
Rafael Luebbert
Macik
Lars Magne Ingebrigtsen
Scott Manley
Vladimir Marek
Sergey A. Maslyakov
Chris Matrakidis
Greg Maxwell
Scott Miller
Darin Morrison
Tomasz Motylewski
Kimmo Mustonen
Dan Nelson
Anton Oleynikov
Mike Oliphant
André Osterhues
Johannes Overmann
Jan Peman
Jan Rafaj
Gertjan van Ratingen
Miguel Revilla Rodriguez
Shawn Riley
Tim Ruddick
Ingo Saitz
Conrad Sanderson
Sergey Sapelin
William Schelter
Naoki Shibata
Sigbjørn Skjæret
Nathan Slingerland
Patrick De Smet
Acy Stapp
Mark Stephens
Alexander Stumpf
Stephane Tavenard
Mark Taylor
Brad Threatt
Takehiro Tominaga
Warren Toomey
Roel Van Den Berghe
Christopher Wise
Martin Weghofer
William Welch
Gerhard Wesp
Alfred Weyers
Ethan Yeo
Chuck Zenkus

Original ISO contributors:

Bill Aspromonte
Shaun Astarabadi
R. Bittner
Karlheinz Brandenburg
W. Joseph Carter
Jack Chang
Mike Coleman
Johnathan Devine
Ernst Eberlein
Dan Ellis
Peter Farrett
Jean-Georges Fritsch
Vlad Fruchter
Hendrik Fuchs
Bernhard Grill
Amit Gulati
Munsi Haque
Chuck Hsiao
Toshiyuki Ishino
Masahiro Iwadare
Earl Jennings
James Johnston
Leon v.d. Kerkhof
Don Lee
Mike Li
Yu-Tang Lin
Soren Neilsen
Simao F. Campos Neto
Mark Paley
Davis Pan
Tan Ah Peng
Kevin Peterson
Juan Pineda
Ernst F. Schroeder
Peter Siebert
Jens Spille
John Stewart
Sam Stewart
Al Tabayoyon
Kathy Wang
Franz-Otto Witte
Douglas Wong