

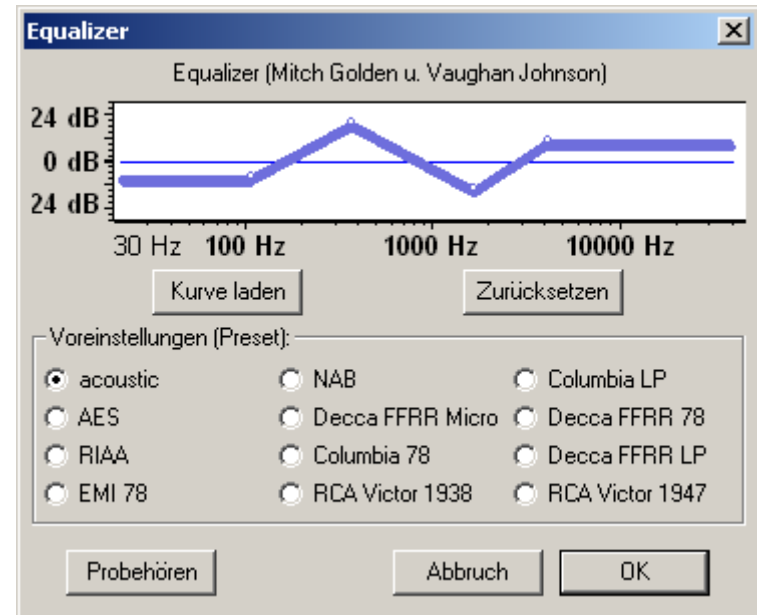
# Digitale Medien

Übung

Vielzahl von Filtern/Effekten vorhanden

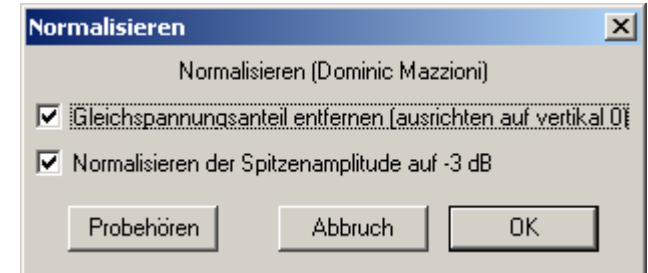
## Equalizer

Erlaubt einzelne Bereiche des Frequenzspektrums gezielt lauter oder leiser zu machen.



## Normalisieren

Bringt ein Tonsignal auf eine einheitliche Lautstärke.



## Kompressor

„dient der Einschränkung der Dynamik eines Signals“ (Wikipedia)

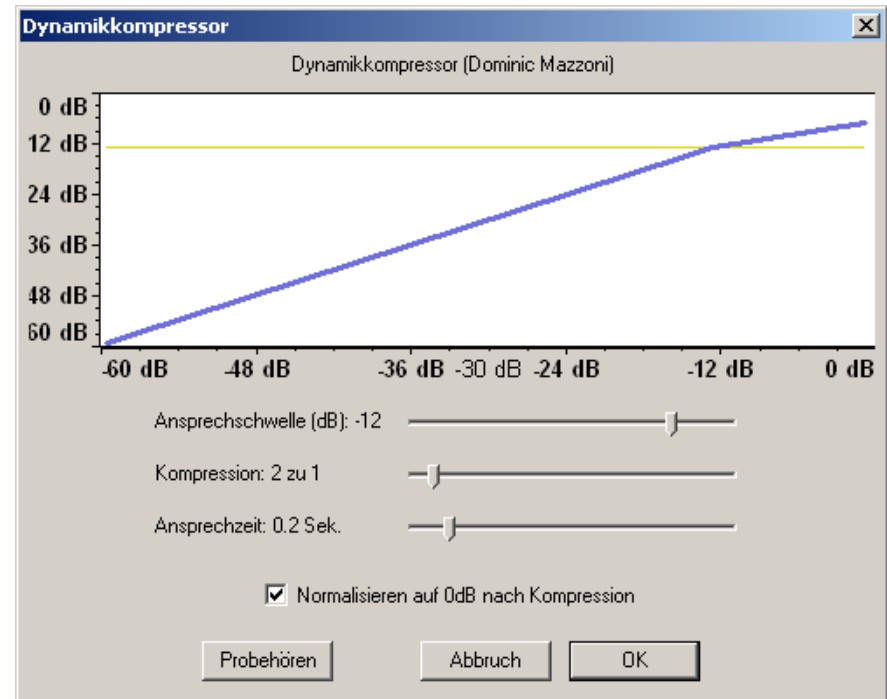
Macht z.B. zu laute Passagen eines Stücks leiser und "glättet" es damit

Einstellungen:

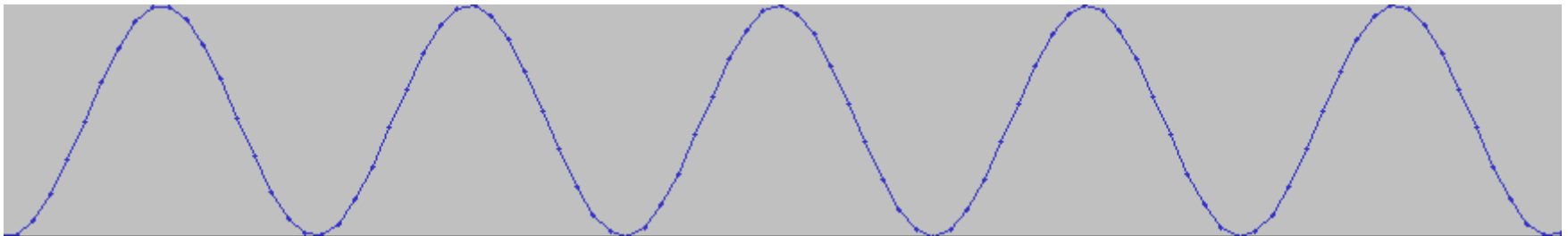
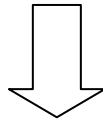
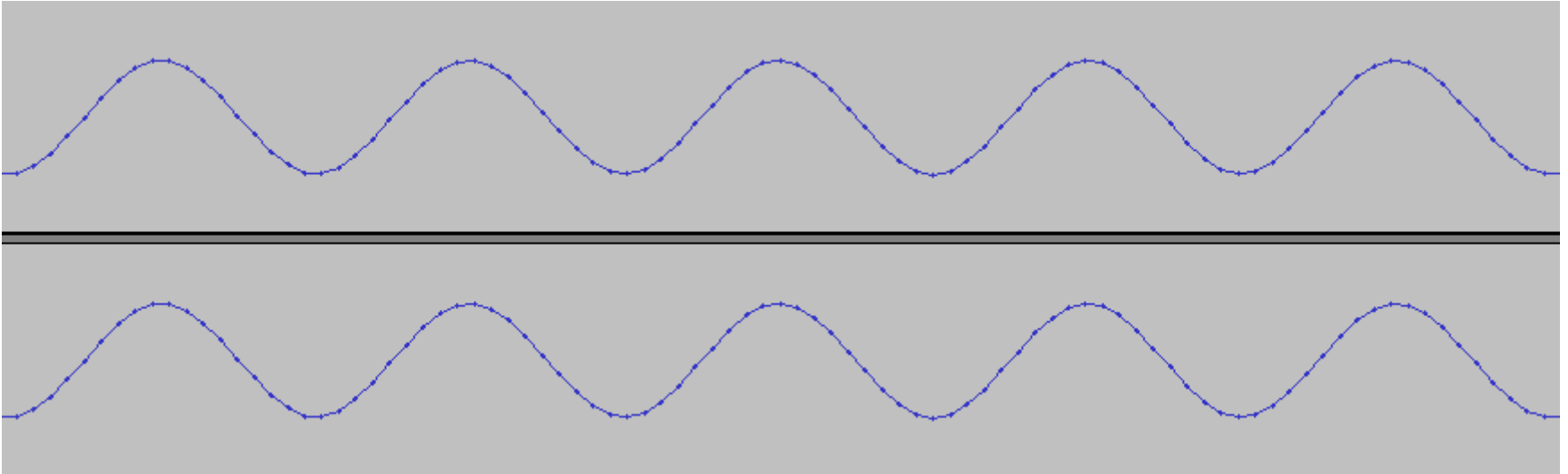
Ansprechschwelle (Treshold): Ab welcher Lautstärke wird Kompressor ausgelöst?

Kompression (Ratio): Wie stark wird das Signal komprimiert?

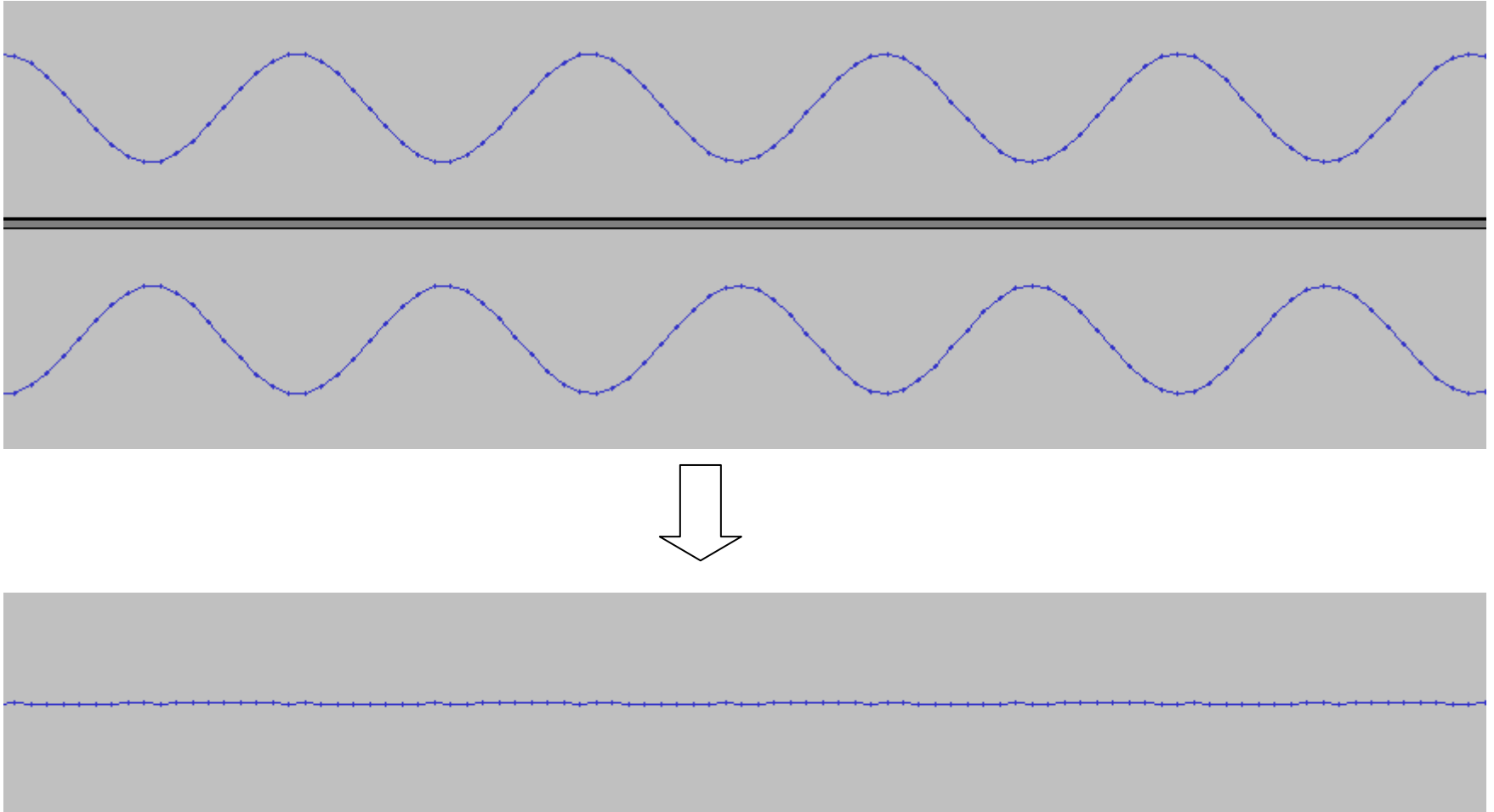
Ansprechzeit (Attack): Wie lange dauert es, bis der Kompressor aktiv wird?



Addition von ähnlichen (kohärenten) Signalen führt zu einer Verstärkung der Amplitude.



Addition von zeitlich verschobenen Tonsignalen führt zu einer Reduzierung der Amplitude.



# Kombination von Tonquellen

Audio

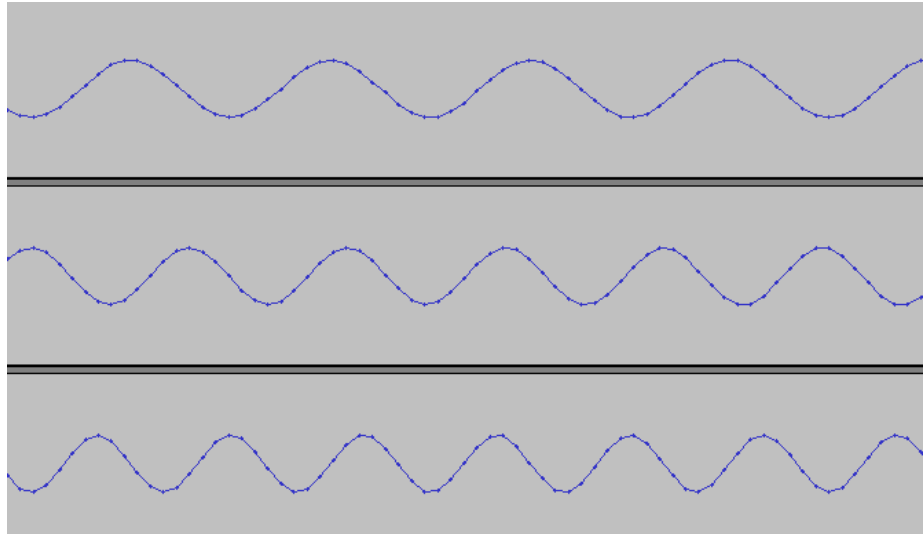
Sinusschwingungen mit den Frequenzen

523,25 (C)

659,26 (E)

783,99 (G)

Nacheinander zuschalten



# Hexeditor

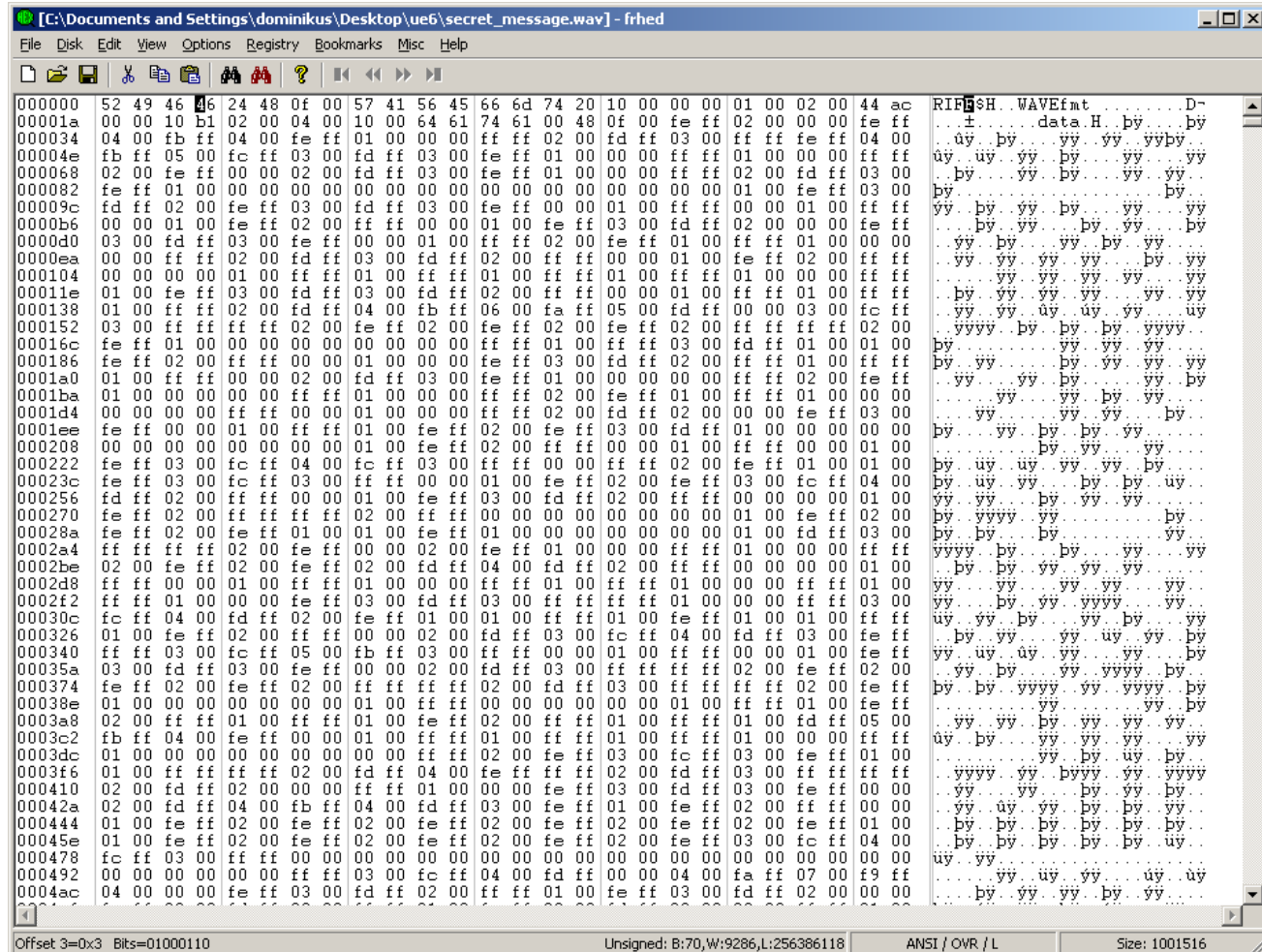
Zeigt einzelne Bytes einer Datei an

z.B. Linux:

khxedit

Windows:

fhred ([www.kibria.de/frhed.html](http://www.kibria.de/frhed.html))





Hexadezimal nach Dezimal

**A4C**<sub>16</sub>                    (0xA4C)

$$\mathbf{C} * 16^0 + \mathbf{4} * 16^1 + \mathbf{A} * 16^2 =$$

$$12 \quad + 64 \quad + 2560 \quad =$$

$$\mathbf{2636}_{10}$$

0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Google

"0xFFFF to decimal"

"XXXXXX to hex"

Hexadezimal nach Dezimal

0xFF

0x7F

0x1C0

0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Google

"0xFFFF to decimal"

"XXXXXX to hex"

## Big-Endian versus Little-Endian

**Big-Endian:** Höchster Wert zuerst234 =>  $2 * 100 + 3 * 10 + 4 * 1$   $\longrightarrow$ **Little-Endian:** Niedrigster Wert zuerst234 =>  $2 * 1 + 3 * 10 + 4 * 100$   $\longleftarrow$ 

Sprache:

24 = „twenty-four“ (Englisch – Big-Endian)

24 = „vierundzwanzig“ (Deutsch – Little-Endian)

Im Speicher:

0A      11      34      FF

Big-Endian:      0A 11 34 FF =&gt; 168 899 839

Little-Endian:      FF 34 11 0A =&gt; 4 281 602 314



33      FA      CC      00

Big-Endian:      0x33FACC00 => 872 074 240

Little-Endian:      0x00CCFA33 => 13 433 395

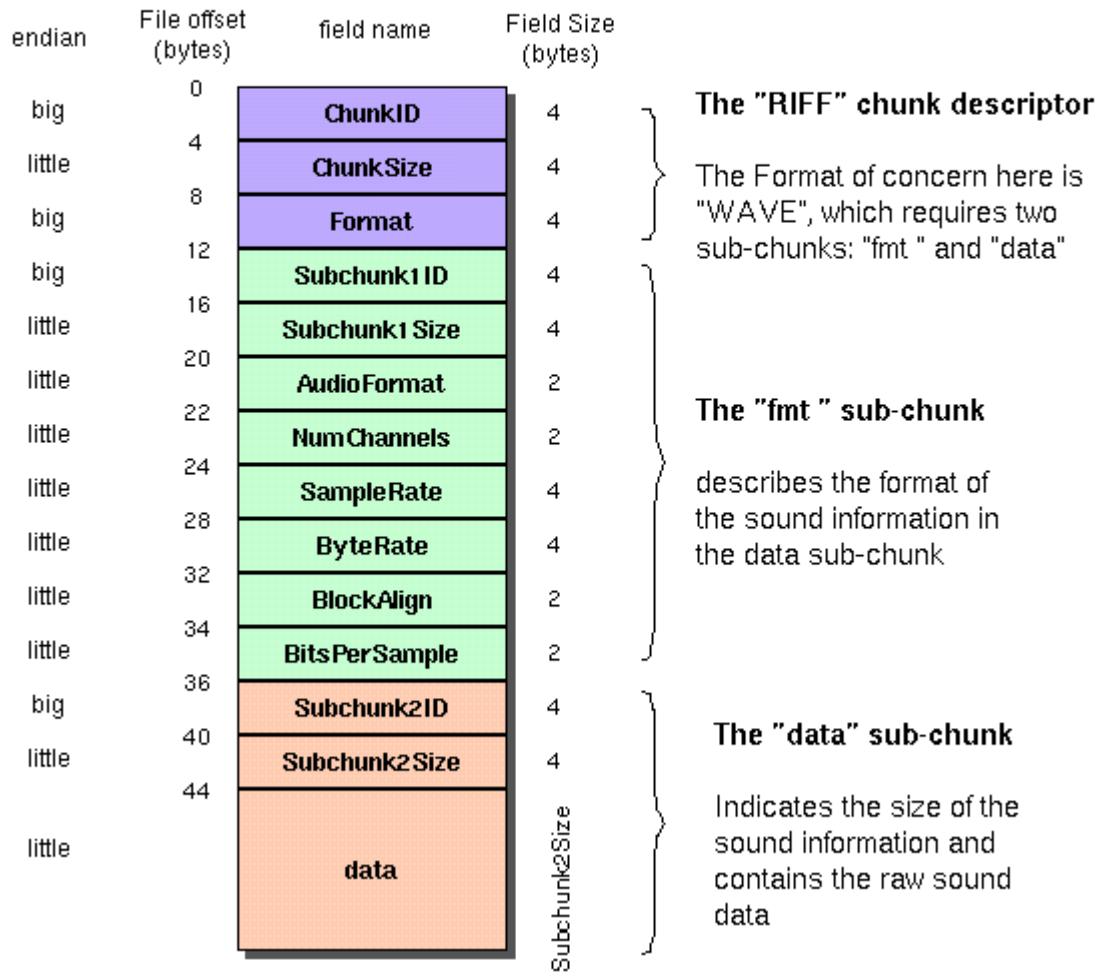
00      00      0F      11

Big-Endian:      0x00000F11 => 3 857

Little-Endian:      0x00CCFA33 => 286 195 712

# The Canonical WAVE file format

WAVE



Beispiel:

`/home/proj/mi_dm/audio/8bit_sinus.wav`

WAVE-Format:

<http://www.sonicspot.com/guide/wavefiles.html>

<http://ccrma.stanford.edu/CCRMA/Courses/422/projects/WaveFormat/>

# The Canonical WAVE file format

endian	File offset (bytes)	field name	Field Size (bytes)
big	0	ChunkID	4
little	4	ChunkSize	4
big	8	Format	4
big	12	Subchunk1 ID	4
little	16	Subchunk1 Size	4
little	20	AudioFormat	2
little	22	NumChannels	2
little	24	SampleRate	4
little	28	ByteRate	4
little	32	BlockAlign	2
little	34	BitsPerSample	2
big	36	Subchunk2ID	4
little	40	Subchunk2 Size	4
little	44	data	Subchunk2Size

## The "RIFF" chunk descriptor

The Format of concern here is "WAVE", which requires two sub-chunks: "fmt " and "data"

## The "fmt " sub-chunk

describes the format of the sound information in the data sub-chunk

## The "data" sub-chunk

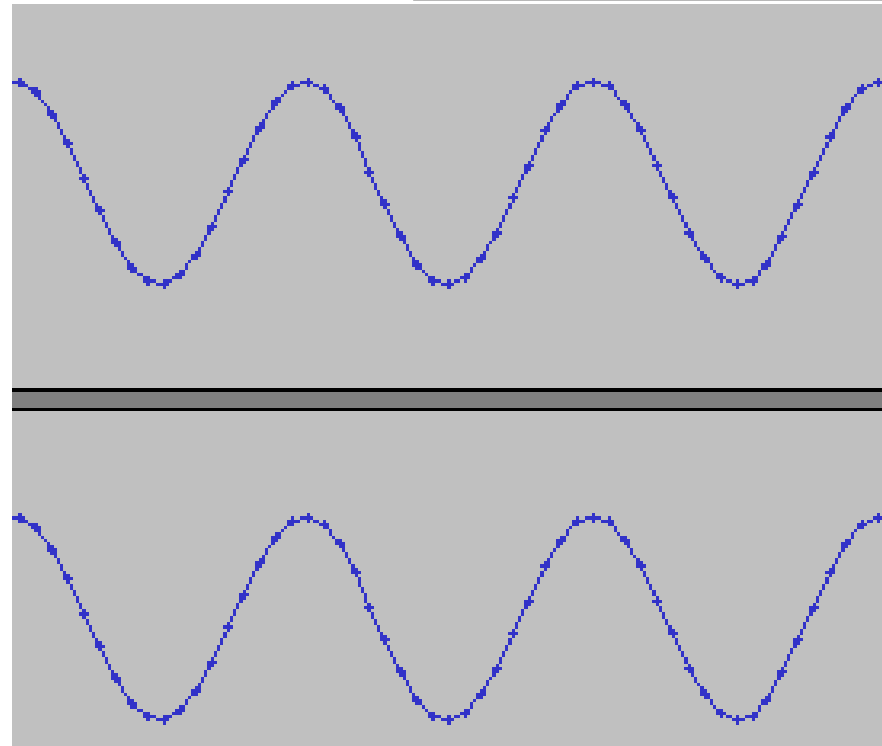
Indicates the size of the sound information and contains the raw sound data

0000	52 49 46 46	RIFF
0004	1c 30 14 00	.0..
0008	57 41 56 45	WAVE
000c	66 6d 74 20	fmt
0010	10 00 00 00	....
0014	01 00 01 00	....
0018	44 ac 00 00	D~..
001c	44 ac 00 00	D~..
0020	01 00 08 00	....
0024	64 61 74 61	data
0028	f8 2f 14 00	ø/..
002c	80 88 90 97	....
0030	9f a7 af b6	.S~¶
0034	bd c4 cb d1	KAEN
0038	d7 dd e2 e7	xYâç

```

00024 64 61 da
00026 74 61 ta
00028 80 a9 .@
0002a 03 00 ..
0002c 80 7f ..
0002e 95 95 ...
00030 a8 a8 ...
00032 b7 b7 ..
00034 be be %%%
00036 bf bf %%
00038 b8 b8 %
0003a aa aa %
0003c 97 97 ..
0003e 82 82 ..
00040 6c 6c ll
00042 58 58 XX
00044 49 49 II
00046 41 41 AA
00048 40 40 @@
0004a 46 46 FF
0004c 54 54 TT
0004e 66 66 ff
00050 7b 7b {{
00052 91 91 ''
00054 a5 a5 %%
00056 b4 b4 ''
00058 bd bd %%
0005a bf bf %%
0005c b9 b9 11
0005e ad ad --
00060 9b 9b ..
00062 86 86 ..
00064 70 70 pp
00066 5c 5c \

```



8-Bit Stereo

## Interleaving

Die beiden Stereospuren werden abwechselnd in der Datei abgelegt