This is a brief and informal document targeted to those who want to deal with the MPEG format. If you are one of them, you probably already know what is MPEG audio. If not, jump to http://www.mp3.com/ or http://www.layer3.org/ where you will find more details and also more links. This document does not cover compression and decompression algorithm.

NOTE: You cannot just search the Internet and find the MPEG audio specs. It is copyrighted and you will have to pay quite a bit to get the Paper. That's why I made this. Information I got is gathered from the Internet, and mostly originate from program sources I found available for free. Despite my intention to always specify the information sources, I am not able to do it this time. Sorry, I did not maintain the list. :-(

These are not a decoding specs, it just informs you how to read the MPEG headers and the MPEG TAG. MPEG Version 1, 2 and 2.5 and Layer I, II and III are supported, the MP3 TAG (ID3v1 and ID3v1.1) also.. Those of you who use Delphi may find MPGTools Delphi unit (freeware source) useful, it is where I implemented this stuff.

I do not claim information presented in this document is accurate. At first I just gathered it from different sources. It was not an easy task but I needed it. Later, I received lots of comments as feedback when I published this document. I think this last release is highly accurate due to comments and corrections I received.

This document is last updated on December 22, 1999.

MPEG Audio Compression Basics

This is one of many methods to compress audio in digital form trying to consume as little space as possible but keep audio quality as good as possible. MPEG compression showed up as one of the best achievements in this area.

This is a lossy compression, which means, you will certainly loose some audio information when you use this compression methods. But, this lost can hardly be noticed because the compression method tries to control it. By using several quite complicate and demanding mathematical algorithms it will only loose those parts of sound that are hard to be heard even in the original form. This leaves more space for information that is important. This way you can compress audio up to 12 times (you may choose compression ratio) which is really significant. Due to its quality MPEG audio became very popular.

MPEG standards MPEG-1, MPEG-2 and MPEG-4 are known but this document covers first two of them. There is an unofficial MPEG-2.5 which is rarely used. It is also covered.

MPEG-1 audio (described in ISO/IEC 11172-3) describes three Layers of audio coding with the following properties:

- one or two audio channels
- sample rate 32kHz, 44.1kHz or 48kHz.
- bit rates from 32kbps up to 448kbps

Each layer has its merits.

MPEG-2 audio (described in ISO/IEC 13818-3) has two extensions to MPEG-1, usually referred as MPEG-2/LSF and MPEG-2/Multichannel.

MPEG-2/LSF has the following properties:

- one or two audio channels
- sample rates half those of MPEG-1
- bit rates from 8 kbps up to 256kbps.

MPEG-2/Multichannel has the following properties:

- up to 5 full range audio channels and an LFE-channel (Low Frequency Enhancement <> subwoofer!)
- sample rates the same as those of MPEG-1
- highest possible bitrate goes up to about 1Mbps for 5.1

MPEG Audio Frame Header

An MPEG audio file is built up from smaller parts called frames. Generally, frames are independent items. Each frame has its own header and audio informations. There is no file header. Therefore, you can cut any part of MPEG file and play it correctly (this should be done on frame boundaries but most applications will handle incorrect headers). For Layer III, this is not 100% correct. Due to internal data organization in MPEG version 1 Layer III files, frames are often dependent of each other and they cannot be cut off just like that.

When you want to read info about an MPEG file, it is usually enough to find the first frame, read its header and assume that the other frames are the same This may not be always the case. Variable bitrate MPEG files may use so called bitrate switching, which means that bitrate changes according to the content of each frame. This way lower bitrates may be used in frames where it will not reduce sound quality. This allows making better compression while keeping high quality of sound.

The frame header is constituted by the very first four bytes (32bits) in a frame. The first eleven bits (or first twelve bits, see below about frame sync) of a frame header are always set and they are called "frame sync". Therefore, you can search through the file for the first occurence of frame sync (meaning that you have to find a byte with a value of 255, and followed by a byte with its three (or four) most significant bits set). Then you read the whole header and check if the values are correct. You will see in the following table the exact meaning of each bit in the header, and which values may be checked for validity. Each value that is specified as reserved, invalid, bad, or not allowed should indicate an invalid header. Remember, this is not enough, frame sync can be easily (and very frequently) found in any binary file. Also it is likely that MPEG file contains garbage on it's beginning which also may contain false sync. Thus, you have to check two or more frames in a row to assure you are really dealing with MPEG audio file.

Frames may have a CRC check. The CRC is 16 bits long and, if it exists, it follows the frame header. After the CRC comes the audio data. You may calculate the length of the frame and use it if you need to read other headers too or just want to calculate the CRC of the frame, to compare it with the one you read from the file. This is actually a very good method to check the MPEG header validity.

Here is "graphical" presentation of the header content. Since at the time I created this document there were no such information publicly available, I've creted notation to explain content and meaning of data within mpeg file. Characters from A to M are used to indicate different fields. In the table, you can see details about the content of each field.

`AAAAAAAA AAABBCCD EEEEFFGH IIJJKLMM`

Sign	Length (bits)	Position (bits)	Description
A	11	(31-21)	Frame sync (all bits set)
B	2	(20,19)	MPEG Audio version ID 00 - MPEG Version 2.5 01 - reserved 10 - MPEG Version 2 (ISO/IEC 13818-3) 11 - MPEG Version 1 (ISO/IEC 11172-3) Note: MPEG Version 2.5 is not official standard. Bit No 20 in frame header is used to indicate version 2.5. Applications that do not support this MPEG version expect this bit always to be set, meaning that frame sync (A) is twelve bits long, not eleve as stated here. Accordingly, B is one bit long (represents only bit No 19). I recommend using methodology presented here, since this allows you to distinguish all three versions and keep full compatibility.
C	2	(18,17)	Layer description 00 - reserved 01 - Layer III 10 - Layer II 11 - Layer I
D	1	(16)	Protection bit 0 - Protected by CRC (16bit crc follows header) 1 - Not protected
E	4	(15,12)	Bitrate index

bits	V1,L1	V1,L2	V1,L3	V2,L1	V2, L2 & L3
0000	free	free	free	free	free
0001	32	32	32	32	8
0010	64	48	40	48	16

0011	96	56	48	56	24
0100	128	64	56	64	32
0101	160	80	64	80	40
0110	192	96	80	96	48
0111	224	112	96	112	56
1000	256	128	112	128	64
1001	288	160	128	144	80
1010	320	192	160	160	96
1011	352	224	192	176	112
1100	384	256	224	192	128
1101	416	320	256	224	144
1110	448	384	320	256	160
1111	bad	bad	bad	bad	bad

NOTES: All values are in kbps
V1 - MPEG Version 1
V2 - MPEG Version 2 and Version 2.5
L1 - Layer I
L2 - Layer II
L3 - Layer III
"free" means free format. If the correct fixed bitrate (such files cannot use variable bitrate) is different than those presented in upper table it must be determined by the application. This may be implemented only for internal purposes since third party applications have no means to find out correct bitrate. However, this is not impossible to do but demands lot's of efforts.
"bad" means that this is not an allowed value

MPEG files may have variable bitrate (VBR). This means that bitrate in the file may change. I have learned about two used methods:

- bitrate switching. Each frame may be created with different bitrate. It may be used in all layers. Layer III decoders must support this method. Layer I & II decoders may support it.
- bit reservoir. Bitrate may be borrowed (within limits) from previous frames in order to provide more bits to demanding parts of the input signal. This causes, however, that the frames are no longer independent, which means you should not cut this files. This is supported only in Layer III.

More about VBR you may find on Xing Tech site

For Layer II there are some combinations of bitrate and mode which are not allowed. Here is a list of allowed combinations.

bitrate	allowed modes
free	all
32	single channel
48	single channel
56	single channel
64	all
80	single channel
96	all
112	all
128	all
160	all
192	all
224	stereo, intensity stereo, dual channel
256	stereo, intensity stereo, dual channel
320	stereo, intensity stereo, dual channel
384	stereo, intensity stereo, dual channel

F 2 (11,10) Sampling rate frequency index (values are in Hz)

bits	MPEG1	MPEG2	MPEG2.5
00	44100	22050	11025
01	48000	24000	12000
10	32000	16000	8000
11	reserv.	reserv.	reserv.

G 1 (9) Padding bit
0 - frame is not padded
1 - frame is padded with one extra slot
Padding is used to fit the bit rates exactly. For an example: 128k 44.1kHz layer II uses a lot of 418 bytes and some of 417 bytes long frames to get the exact 128k bitrate. For Layer I slot is 32 bits long, for Layer II and Layer III slot is 8 bits long.

How to calculate frame length

First, let's distinguish two terms frame size and frame length. Frame size is the number of samples contained in a frame. It is constant and always 384 samples for Layer I and 1152 samples for Layer II and Layer III. Frame length is length of a frame when compressed. It is calculated in slots. One slot is 4 bytes long for

Layer I, and one byte long for Layer II and Layer III. When you are reading MPEG file you must calculate this to be able to find each consecutive frame. Remember, frame length may change from frame to frame due to padding or bitrate switching.

Read the BitRate, SampleRate and Padding of the frame header.

For Layer I files us this formula:

FrameLengthInBytes = (12 * BitRate / SampleRate + Padding) * 4

For Layer II & III files use this formula:

FrameLengthInBytes = 144 * BitRate / SampleRate + Padding

Example:
Layer III, BitRate=128000, SampleRate=441000, Padding=0
==> FrameSize=417 bytes

H 1 (8) Private bit. It may be freely used for specific needs of an application, i.e. if it has to trigger some application specific events.

I 2 (7,6) Channel Mode
00 - Stereo
01 - Joint stereo (Stereo)
10 - Dual channel (Stereo)
11 - Single channel (Mono)

J 2 (5,4) Mode extension (Only if Joint stereo)

Mode extension is used to join informations that are of no use for stereo effect, thus reducing needed resources. These bits are dynamically determined by an encoder in Joint stereo mode.

Complete frequency range of MPEG file is divided in subbands There are 32 subbands. For Layer I & II these two bits determine frequency range (bands) where intensity stereo is applied. For Layer III these two bits determine which type of joint stereo is used (intensity stereo or m/s stereo). Frequency range is determined within decompression algorythm.

Layer I and II		Layer III	
value	Layer I & II	Intensity stereo	MS stereo
00	bands 4 to 31	off	off
01	bands 8 to 31	on	off
10	bands 12 to 31	off	on
11	bands 16 to 31	on	on

K	1	(3)	Copyright 0 - Audio is not copyrighted 1 - Audio is copyrighted
L	1	(2)	Original 0 - Copy of original media 1 - Original media
M	2	(1,0)	Emphasis 00 - none 01 - 50/15 ms 10 - reserved 11 - CCIT J.17

MPEG Audio Tag ID3v1

The TAG is used to describe the MPEG Audio file. It contains information about artist, title, album, publishing year and genre. There is some extra space for comments. It is exactly 128 bytes long and is located at very end of the audio data. You can get it by reading the last 128 bytes of the MPEG audio file.

AAABBBBB BBBBBBBB BBBBBBBB BBBBBBBB
BCCCCCCC CCCCCCCC CCCCCCCC CCCCCCCD
DDDDDDDD DDDDDDDD DDDDDDDD DDDDDEEE
EFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFG

Sign	Length (bytes)	Position (bytes)	Description
A	3	(0-2)	Tag identification. Must contain 'TAG' if tag exists and is correct.
B	30	(3-32)	Title
C	30	(33-62)	Artist
D	30	(63-92)	Album
E	4	(93-96)	Year
F	30	(97-126)	Comment
G	1	(127)	Genre

The specification asks for all fields to be padded with null character (ASCII 0). However, not all applications respect this (an example is WinAmp which pads fields with <space>, ASCII 32).

There is a small change proposed in **ID3v1.1** structure. The last byte of the Comment field may be used to specify the track number of a song in an album. It should contain a null character (ASCII 0) if the information is unknown.

Genre is a numeric field which may have one of the following values:

0 'Blues'	20 'Alternative'	40 'AlternRock'	60 'Top 40'
1 'Classic Rock'	21 'Ska'	41 'Bass'	61 'Christian Rap'
2 'Country'	22 'Death Metal'	42 'Soul'	62 'Pop/Funk'
3 'Dance'	23 'Pranks'	43 'Punk'	63 'Jungle'
4 'Disco'	24 'Soundtrack'	44 'Space'	64 'Native American'
5 'Funk'	25 'Euro-Techno'	45 'Meditative'	65 'Cabaret'
6 'Grunge'	26 'Ambient'	46 'Instrumental Pop'	66 'New Wave'
7 'Hip-Hop'	27 'Trip-Hop'	47 'Instrumental Rock'	67 'Psychadelic'
8 'Jazz'	28 'Vocal'	48 'Ethnic'	68 'Rave'
9 'Metal'	29 'Jazz+Funk'	49 'Gothic'	69 'Showtunes'
10 'New Age'	30 'Fusion'	50 'Darkwave'	70 'Trailer'
11 'Oldies'	31 'Trance'	51 'Techno-Industrial'	71 'Lo-Fi'
12 'Other'	32 'Classical'	52 'Electronic'	72 'Tribal'
13 'Pop'	33 'Instrumental'	53 'Pop-Folk'	73 'Acid Punk'
14 'R&B'	34 'Acid'	54 'Eurodance'	74 'Acid Jazz'
15 'Rap'	35 'House'	55 'Dream'	75 'Polka'
16 'Reggae'	36 'Game'	56 'Southern Rock'	76 'Retro'
17 'Rock'	37 'Sound Clip'	57 'Comedy'	77 'Musical'
18 'Techno'	38 'Gospel'	58 'Cult'	78 'Rock & Roll'
19 'Industrial'	39 'Noise'	59 'Gangsta'	79 'Hard Rock'

WinAmp expanded this table with next codes:

80 'Folk'	92 'Progressive Rock'	104 'Chamber Music'	116 'Ballad'
81 'Folk-Rock'	93 'Psychedelic Rock'	105 'Sonata'	117 'Poweer Ballad'
82 'National Folk'	94 'Symphonic Rock'	106 'Symphony'	118 'Rhytmic Soul'
83 'Swing'	95 'Slow Rock'	107 'Booty Brass'	119 'Freestyle'
84 'Fast Fusion'	96 'Big Band'	108 'Primus'	120 'Duet'
85 'Bebob'	97 'Chorus'	109 'Porn Groove'	121 'Punk Rock'
86 'Latin'	98 'Easy Listening'	110 'Satire'	122 'Drum Solo'
87 'Revival'	99 'Acoustic'	111 'Slow Jam'	123 'A Capela'

88 'Celtic'	100 'Humour'	112 'Club'	124 'Euro-House'
89 'Bluegrass'	101 'Speech'	113 'Tango'	125 'Dance Hall'
90 'Avantgarde'	102 'Chanson'	114 'Samba'	
91 'Gothic Rock'	103 'Opera'	115 'Folklore'	

Any other value should be considered as 'Unknown'

MPEG Audio Tag ID3v2

This is new proposed TAG format which is different than ID3v1 and ID3v1.1. Complete tech specs for it may be found at http://www.id3.org/.

Created on September 1998. by Predrag Supurovic.
Thanks to Jean for debugging and polishing of this document, Peter Luijer, Guwani, Rob Leslie and Franc Zijderveld
for valuable comments and corrections.