

MP3 文件格式解析

Peter Lee 2008-06-05

目录

- 一、概述
- 二、整个 MP3 文件结构
- 三、MP3 帧格式
 - 1. 帧头格式
 - 2. MAIN_DATA
- 四、ID3 标准
 - 1. ID3V1
 - 2. ID3V2
- 五、MP3 文件实例剖析
- 六、资料

一、概述

MP3 文件是由帧(frame)构成的,帧是 MP3 文件最小的组成单位。MP3 的全称应为 MPEG1 Layer-3 音频文件,MPEG(Moving Picture Experts Group)在汉语中译为活动图像专家组,特指活动影音压缩标准,MPEG 音频文件是 MPEG1 标准中的声音部分,也叫 MPEG 音频层,它根据压缩质量和编码复杂程度划分为三层,即 Layer-1、Layer2、Layer3,且分别对应 MP1、MP2、MP3 这三种声音文件,并根据不同的用途,使用不同层次的编码。MPEG 音频编码的层次越高,编码器越复杂,压缩率也越高,MP1 和 MP2 的压缩率分别为 4:1 和 6:1-8:1,而 MP3 的压缩率则高达 10:1-12:1,也就是说,一分钟 CD 音质的音乐,未经压缩需要 10MB 的存储空间,而经过 MP3 压缩编码后只有 1MB 左右。不过 MP3 对音频信号采用的是有损压缩方式,为了降低声音失真度,MP3 采取了“感官编码技术”,即编码时先对音频文件进行频谱分析,然后用过滤器滤掉噪音电平,接着通过量化的方式将剩下的每一位打散排列,最后形成具有较高压缩比的 MP3 文件,并使压缩后的文件在回放时能够达到比较接近原音源的声音效果。

二、整个 MP3 文件结构

MP3 文件大体分为三部分: TAG_V2(ID3V2), Frame, TAG_V1(ID3V1)

ID3V2

包含了作者，作曲，专辑等信息，长度不固定，扩展了 ID3V1 的信息量。

Frame

.
. .
.

Frame

一系列的帧，个数由文件大小和帧长决定

每个 FRAME 的长度可能不固定，也可能固定，由位率 **bitrate** 决定

每个 FRAME 又分为帧头和数据实体两部分

帧头记录了 mp3 的位率，采样率，版本等信息，每个帧之间相互独立

ID3V1

包含了作者，作曲，专辑等信息，长度为 128BYTE。

三、MP3 帧格式

1. 帧头格式

帧头长 4 字节，对于固定位率的 MP3 文件，所有帧的帧头格式一样其数据结构如下：

```
typedef FrameHeader {  
    unsigned int sync: 11;           //同步信息  
    unsigned int version: 2;         //版本  
    unsigned int layer: 2;           //层  
    unsigned int error_protection: 1; // CRC 校验  
    unsigned int bitrate_index: 4;   //位率  
    unsigned int sampling_frequency: 2; //采样频率  
    unsigned int padding: 1;         //帧长调节  
    unsigned int private: 1;         //保留字  
    unsigned int mode: 2;            //声道模式  
    unsigned int mode_extension: 2;  //扩充模式  
    unsigned int copyright: 1;       // 版权
```

```
unsigned int original: 1;           //原版标志
unsigned int emphasis: 2;          //强调模式
}HEADER, *LPHEADER;
```

帧头 4 字节使用说明见表 1。

MP3 帧长取决于位率和频率，计算公式为：

```
. mpeg1.0    layer1 :  帧长= (48000*bitrate)/sampling_freq + padding
              layer2&3: 帧长= (144000*bitrate)/sampling_freq + padding
. mpeg2.0    layer1 :  帧长= (24000*bitrate)/sampling_freq + padding
              layer2&3 : 帧长= (72000*bitrate)/sampling_freq + padding
```

例如：位率为 64kbps，采样频率为 44.1kHz，padding（帧长调节）为 1 时，帧长为 210 字节。

帧头后面是可变长度的附加信息，对于标准的 MP3 文件来说，其长度是 32 字节，紧接其后的是压缩的声音数据，当解码器读到此处时就进行解码了。

表 1 MP3 帧头字节使用说明

名称	位长		说 明			
同步信息	11	第 1、2 字 节	所有位均为 1，第 1 字节恒为 FF。			
版本	2		00-MPEG 2.5	01-未定义	10-MPEG 2	11-MPEG 1
层	2		00-未定义	01-Layer 3	10-Layer 2	11-Layer 1
CRC 校 验	1		0-校验		1-不校验	

位率	4	第 3 字节	取样率，单位是 kbps，例如采用 MPEG-1 Layer 3，64kbps 是，值为 0101。																																																																																																																								
			<table border="1"> <thead> <tr> <th>bits</th><th>V1,L1</th><th>V1,L2</th><th>V1,L3</th><th>V2,L1</th><th>V2,L2</th><th>V2,L3</th></tr> </thead> <tbody> <tr><td>0000</td><td>free</td><td>free</td><td>free</td><td>free</td><td>free</td><td>free</td></tr> <tr><td>0001</td><td>32</td><td>32</td><td>32</td><td>32(32)</td><td>32(8)</td><td>8 (8)</td></tr> <tr><td>0010</td><td>64</td><td>48</td><td>40</td><td>64(48)</td><td>48(16)</td><td>16 (16)</td></tr> <tr><td>0011</td><td>96</td><td>56</td><td>48</td><td>96(56)</td><td>56(24)</td><td>24 (24)</td></tr> <tr><td>0100</td><td>128</td><td>64</td><td>56</td><td>128(64)</td><td>64(32)</td><td>32 (32)</td></tr> <tr><td>0101</td><td>160</td><td>80</td><td>64</td><td>160(80)</td><td>80(40)</td><td>64 (40)</td></tr> <tr><td>0110</td><td>192</td><td>96</td><td>80</td><td>192(96)</td><td>96(48)</td><td>80 (48)</td></tr> <tr><td>0111</td><td>224</td><td>112</td><td>96</td><td>224(112)</td><td>112(56)</td><td>56 (56)</td></tr> <tr><td>1000</td><td>256</td><td>128</td><td>112</td><td>256(128)</td><td>128(64)</td><td>64 (64)</td></tr> <tr><td>1001</td><td>288</td><td>160</td><td>128</td><td>288(144)</td><td>160(80)</td><td>128 (80)</td></tr> <tr><td>1010</td><td>320</td><td>192</td><td>160</td><td>320(160)</td><td>192(96)</td><td>160 (96)</td></tr> <tr><td>1011</td><td>352</td><td>224</td><td>192</td><td>352(176)</td><td>224(112)</td><td>112 (112)</td></tr> <tr><td>1100</td><td>384</td><td>256</td><td>224</td><td>384(192)</td><td>256(128)</td><td>128 (128)</td></tr> <tr><td>1101</td><td>416</td><td>320</td><td>256</td><td>416(224)</td><td>320(144)</td><td>256 (144)</td></tr> <tr><td>1110</td><td>448</td><td>384</td><td>320</td><td>448(256)</td><td>384(160)</td><td>320 (160)</td></tr> <tr><td>1111</td><td>bad</td><td>bad</td><td>bad</td><td>bad</td><td>bad</td><td>bad</td></tr> </tbody> </table> <p> V1 - MPEG 1 V2 - MPEG 2 and MPEG 2.5 L1 - Layer 1 L2 - Layer 2 L3 - Layer 3 "free" 表示位率可变 "bad" 表示不允许值 </p>				bits	V1,L1	V1,L2	V1,L3	V2,L1	V2,L2	V2,L3	0000	free	free	free	free	free	free	0001	32	32	32	32(32)	32(8)	8 (8)	0010	64	48	40	64(48)	48(16)	16 (16)	0011	96	56	48	96(56)	56(24)	24 (24)	0100	128	64	56	128(64)	64(32)	32 (32)	0101	160	80	64	160(80)	80(40)	64 (40)	0110	192	96	80	192(96)	96(48)	80 (48)	0111	224	112	96	224(112)	112(56)	56 (56)	1000	256	128	112	256(128)	128(64)	64 (64)	1001	288	160	128	288(144)	160(80)	128 (80)	1010	320	192	160	320(160)	192(96)	160 (96)	1011	352	224	192	352(176)	224(112)	112 (112)	1100	384	256	224	384(192)	256(128)	128 (128)	1101	416	320	256	416(224)	320(144)	256 (144)	1110	448	384	320	448(256)	384(160)	320 (160)	1111	bad	bad	bad	bad
bits	V1,L1	V1,L2	V1,L3	V2,L1	V2,L2	V2,L3																																																																																																																					
0000	free	free	free	free	free	free																																																																																																																					
0001	32	32	32	32(32)	32(8)	8 (8)																																																																																																																					
0010	64	48	40	64(48)	48(16)	16 (16)																																																																																																																					
0011	96	56	48	96(56)	56(24)	24 (24)																																																																																																																					
0100	128	64	56	128(64)	64(32)	32 (32)																																																																																																																					
0101	160	80	64	160(80)	80(40)	64 (40)																																																																																																																					
0110	192	96	80	192(96)	96(48)	80 (48)																																																																																																																					
0111	224	112	96	224(112)	112(56)	56 (56)																																																																																																																					
1000	256	128	112	256(128)	128(64)	64 (64)																																																																																																																					
1001	288	160	128	288(144)	160(80)	128 (80)																																																																																																																					
1010	320	192	160	320(160)	192(96)	160 (96)																																																																																																																					
1011	352	224	192	352(176)	224(112)	112 (112)																																																																																																																					
1100	384	256	224	384(192)	256(128)	128 (128)																																																																																																																					
1101	416	320	256	416(224)	320(144)	256 (144)																																																																																																																					
1110	448	384	320	448(256)	384(160)	320 (160)																																																																																																																					
1111	bad	bad	bad	bad	bad	bad																																																																																																																					
采样频率	2		采样频率，对于 MPEG-1: 00-44.1kHz 01-48kHz 10-32kHz 11-未定义 对于 MPEG-2: 00-22.05kHz 01-24kHz 10-16kHz 11-未定义 对于 MPEG-2.5: 00-11.025kHz 01-12kHz 10-8kHz 11-未定义																																																																																																																								

帧长调节	1	第 4 字节	用来调整文件头长度，0-无需调整，1-调整，具体调整计算方法见下文。																	
保留字	1		没有使用。																	
声道模式	2		表示声道， 00-立体声 Stereo 01-Joint Stereo 10-双声道 11-单声道																	
扩充模式	2		当声道模式为 01 是才使用。																	
			<table><tr><th>Value</th><th>强度立体声</th><th>MS 立体声</th></tr><tr><td>00</td><td>off</td><td>off</td></tr><tr><td>01</td><td>on</td><td>off</td></tr><tr><td>10</td><td>off</td><td>on</td></tr><tr><td>11</td><td>on</td><td>on</td></tr></table>			Value	强度立体声	MS 立体声	00	off	off	01	on	off	10	off	on	11	on	on
			Value	强度立体声	MS 立体声															
			00	off	off															
			01	on	off															
		10	off	on																
11	on	on																		
版权	1	文件是否合法，0-不合法 1-合法																		
原版标志	1	是否原版， 0-非原版 1-原版																		
强调方式	2	用于声音经降噪压缩后再补偿的分类，很少用到，今后也可能不会用。 00-未定义 01-50/15ms 10-保留 11-CCITT J.17																		

2. MAIN_DATA

MAIN_DATA 部分长度是否变化决定于 FRAMEHEADER 的 bitrate 是否变化,一首 MP3 歌曲,它有三个版本:96Kbps(96 千比特位每秒)、128Kbps 和 192Kbps。Kbps(比特位速率),表明了音乐每秒的数据量,Kbps 值越高,音质越好,文件也越大,MP3 标准规定,不变的 bitrate 的 MP3 文件称作 CBR,大多数 MP3 文件都是 CBR 的,而变化的 bitrate 的 MP3 文件称作 VBR,每个 FRAME 的长度都可能是变化的。下面是 CBR 和 VBR 的不同点:

1)CBR:固定位率的 FRAME 的大小是固定的(公式如上所述),只要知道文件总长度,和帧长即可由播放每帧需 26ms 计算得出 mp3 播放的总时间,也可通过计数帧的个数控制快进、快退慢放等操作。注:有些时候,并不是所有的帧都是等长的,有的帧可能多一个或几个字节。

2)VBR:VBR 是 XING 公司推出的算法,所以在 MP3 的 FRAME 里会有“XING”这个关键字(现在很多流行的小软件也可以进行 VBR 压缩,它们是否遵守这个约定,那就不得而知了),它存放在 MP3 文件中的第一个有效 FRAME 里,它标识了这个 MP3 文件是 VBR 的。同时第一个 FRAME 里存放了 MP3 文件的 FRAME 的总个数,这就很容易获得了播放总时间,同时还有 100 个字节存放了播放总时间的 100 个时间分段的 FRAME 的 INDEX,假设 4 分钟的 MP3 歌曲,240S,分成 100 段,每两个相邻 INDEX 的时间差就是 2.4S,所以通过这个 INDEX,只要前后处理少数的 FRAME,就能快速找出我们需要快进的 FRAME 头。

表 2 VBR 文件第一帧结构

字节	说 明	
1—4	与 CBR 相同的标准声音帧头	
5—40	存放 VBR 文件标识“Xing”（58 69 6E 67），此标识具体位置视采用的 MPEG 标准和声道模式而定。标识的前后字节没有使用。	
	36— 39	MPEG-1 和非单声道(常见)
	21— 24	MPEG-1 和单声道
	21— 24	MPEG-2 和非单声道
	13— 16	MPEG-2 和单声道
41—44	标志，说明是否存储了帧数、文件长度、目录表和 VBR 规模信息，如果存储了，则 01 02 04 08。	
45—48	帧数（包括第一帧）	
49—52	文件长度	
53— 152	目录表，用来按时间进行字节定位。	
153— 156	VBR 规模，用于位率变动	

另可参考下文：

This system was created to minimize file lengths and to preserve sound quality.

Higher frequencies generally needs more space for encoding (thats why many codecs cut all frequencies above cca 16kHz) and lower tones requires less. So if some part of song doesnt consist of higher tones then using eg. 192kbps is wasting of space. It should be enough to use only eg.96kbps.

And it is the principle of VBR. Codec looks over frame and then choose bitrate suitable for its sound quality.

It sounds perfect but it brings some problems:

If you want to jump over 2 minutes in song, it is not a problem with CBR because you are able simply count amount of Bytes which is necessary to skip. But it is impossible with VBR. Frame lengths should be arbitrary so you have to either go frame by frame and counts (time consuming

and very unpractical) or use another mechanism for approximate count.

If you want to cut 5 minutes from the middle of VBR file (all we know CDs where last song takes 10 minutes but 5 minutes is a pure silence, HELL!) problems are the same.

Result? VBR files are more difficult for controlling and adjusting. And I don't like feeling that sound quality changes in every moment. And AFAIK many codecs have problems with creation VBR in good quality.

Personally I can't see any reason why to use VBR - I don't give a fuck if size of one CD in MP3 is 55 MB with CBR or 51 MB with VBR. But everybody has a different taste... some people prefer VBR.

VBR File Structure is the same as for CBR. But the first frame doesn't contain audio data and it is used for special

information about VBR file.

Structure of the first frame:

Byte	Content
------	---------

0-3	Standard audio frame header (as described above). Mostly it contains values FF FB 30 4C, from which you can count $\text{FrameLen} = 156$ Bytes. And that's exactly enough space for storing VBR info.
-----	--

This header contains some important information valid for the whole file:

- MPEG (MPEG1 or MPEG2)
- SAMPLING rate frequency index
- CHANNEL (JointStereo etc.)

4-x	Not used till string "Xing" (58 69 6E 67). This string is used as a main VBR file identifier. If it is not found, file is supposed to be CBR. This string can be placed at different locations according to values of MPEG and CHANNEL (ya, these from a few lines upwards):
-----	--

36-39	"Xing" for MPEG1 and CHANNEL != mono (mostly used)
-------	--

21-24	"Xing" for MPEG1 and CHANNEL == mono
-------	--------------------------------------

21-24	"Xing" for MPEG2 and CHANNEL != mono
-------	--------------------------------------

13-16	"Xing" for MPEG2 and CHANNEL == mono
-------	--------------------------------------

After "Xing" string there are placed flags, number of frames in file and a size of file in Bytes. Each of these items has 4 Bytes and it is stored as 'int' number

in memory. The first is the most significant Byte and the last is the least.

Following schema is for MPEG1 and CHANNEL != mono:

40-43 Flags

Value	Name	Description
00 00 00 01	Frames Flag	set if value for number of frames in file is stored
00 00 00 02	Bytes Flag	set if value for filesize in Bytes is stored
00 00 00 04	TOC Flag	set if values for TOC (see below) are stored
00 00 00 08	VBR Scale Flag	set if values for VBR scale are stored

All these values can be stored simultaneously.

44-47 Frames

Number of frames in file (including the first info one)

48-51 Bytes

File length in Bytes

52-151 TOC (Table of Contents)

Contains of 100 indexes (one Byte length) for easier lookup in file. Approximately solves problem with moving inside file.

Each Byte has a value according this formula:

$$(\text{TOC}[i] / 256) * \text{fileLenInBytes}$$

So if song lasts eg. 240 sec. and you want to jump to 60. sec. (and file is 5 000 000 Bytes length) you can use:

$$\text{TOC}[(60/240)*100] = \text{TOC}[25]$$

and corresponding Byte in file is then approximately at:

$$(\text{TOC}[25]/256) * 5000000$$

If you want to trim VBR file you should also reconstruct Frames, Bytes and TOC properly.

152-155 VBR Scale

I dont know exactly system of storing of this values but this item probably doesnt have deeper meaning.

四、ID3 标准

MP3 帧头中除了存储一些象 private、copyright、original 的简单音乐说明信息以外，没有考虑存放歌名、作者、专辑名、年份等复杂信息，而这些信息在 MP3 应用中非常必要。1996 年，FrickKemp 在“Studio 3”项目中提出了在 MP3 文件尾增加一块用于存放歌曲的说明信息，形成了 ID3 标准，至今已制定出 ID3 V1.0, V1.1, V2.0, V2.3 和 V2.4 标准。版本越高，记录的相关信息就越丰富详尽。

1. ID3V1

ID3 V1.0 标准并不周全，存放的信息少，无法存放歌词，无法录入专辑封面、图片等。V2.0 是一个相当完备的标准，但给编写软件带来困难，虽然赞成此格式的人很多，在软件中真正实现的却极少。绝大多数 MP3 仍使用 ID3 V1.0 标准。此标准是将 MP3 文件尾的最后 128 个字节用来存放 ID3 信息，这 128 个字节使用说明见表 3。

表 3 ID3 V1.0 文件尾说明

字节	长度 (字节)	说 明
1-3	3	存放“TAG”字符，表示 ID3 V1.0 标准，紧接其后的是歌曲信息。
4-33	30	歌名
34-63	30	作者
64-93	30	专辑名
94-97	4	年份
98-127	30	附注
128	1	MP3 音乐类别，共 147 种。

表 4 MP3 音乐类别：

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'

80	Folk	81	Folk/Rock	82	National Folk	83	Swing
84	Fast-Fusion	85	Bebob	86	Latin	87	Revival
88	Celtic	89	Bluegrass	90	Advantgarde	91	Gothic Rock
92	Progressive Rock	93	Psychadelic Rock	94	Symphonic Rock	95	Slow Rock
96	Big Band	97	Chorus	98	Easy Listening	99	Acoustic
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

Any other value should be considered as 'Unknown'

2. ID3V2

ID3V2 到现在一共有 4 个版本,但流行的播放软件一般只支持第 3 版,既 ID3v2.3。由于 ID3V1 记录在 MP3 文件的末尾,ID3V2 就只好记录在 MP3 文件的首部了(如果有一天发布 ID3V3,真不知道该记录在哪里)。也正是由于这个原因,对 ID3V2 的操作比 ID3V1 要慢。而且 ID3V2 结构比 ID3V1 的结构要复杂得多,

但比前者全面且可以伸缩和扩展。

下面就介绍一下 ID3V2.3。

每个 ID3V2.3 的标签都是一个标签头和若干个标签帧或一个扩展标签头组成。关于曲目的信息如标题、作者等都存放在不同的标签帧中,扩展标签头和标签帧并不是必要的,但每个标签至少要有一个标签帧。标签头和标签帧一起顺序存放在 MP3 文件的首部。

1、标签头

在文件的首部顺序记录 10 个字节的 ID3V2.3 的头部。数据结构如下:

```
char Header[3];    /*必须为"ID3"否则认为标签不存在*/  
  
char Ver;          /*版本号 ID3V2.3 就记录 3*/  
  
char Revision;     /*副版本号此版本记录为 0*/  
  
char Flag;         /*存放标志的字节,这个版本只定义了三位,稍后详细解说*/  
  
char Size[4];      /*标签大小,包括标签头的 10 个字节和所有的标签帧的大小*/
```

1).标志字节

标志字节一般为 0,定义如下:

abc00000

a -- 表示是否使用 Unsynchronisation(这个单词不知道是什么意思,字典里也没有找到,一般不设置)

b -- 表示是否有扩展头部,一般没有(至少 Winamp 没有记录),所以一般也不设置

c -- 表示是否为测试标签(99.99%的标签都不是测试用的啦,所以一般也不设置)

2).标签大小

一共四个字节,但每个字节只用 7 位,最高位不使用恒为 0。所以格式如下

0xxxxxxx 0xxxxxxx 0xxxxxxx 0xxxxxxx

计算大小时要将 0 去掉,得到一个 28 位的二进制数,就是标签大小(不懂为什么要这样做),计算公式如下:

```
int total_size;  
  
total_size = (Size[0]&0x7F)*0x200000  
            +(Size[1]&0x7F)*0x400 0x4000  
            +(Size[2]&0x7F)*0x80  
            +(Size[3]&0x7F)
```

2、标签帧

每个标签帧都有一个 10 个字节的帧头和至少一个字节的固定长度的内容组成。

它们也是顺序存放在文件

中,和标签头和其他的标签帧也没有特殊的字符分隔。得到一个完整的帧的内容只有从帧头中的到内容大小后才能读出,读取时要注意大小,不要将其他帧的内容或帧头读入。

帧头的定义如下:

```
char FrameID[4]; /*用四个字符标识一个帧,说明其内容,稍后有常用的标识对照表*/
```

```
char Size[4]; /*帧内容的大小,不包括帧头,不得小于 1*/
```

```
char Flags[2]; /*存放标志,只定义了 6 位,稍后详细解说*/
```

1).帧标识

用四个字符标识一个帧,说明一个帧的内容含义,常用的对照如下:

TIT2=标题 表示内容为这首歌的标题,下同

TPE1=作者

TALB=专集

TRCK=音轨 格式:N/M 其中 N 为专集中的第 N 首,M 为专集中共 M 首,N 和 M 为 ASCII 码表示的数字

TYER=年代 是用 ASCII 码表示的数字

TCOM=类型 直接用字符串表示

COMM=备注 格式:"eng\0 备注内容",其中 eng 表示备注所使用的自然语言

2).大小

这个可没有标签头的算法那么麻烦,每个字节的 8 位全用,格式如下

```
XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
```

算法如下:

```
int FSize;
```

```
FSize = Size[0]*0x100000000
```

```
+Size[1]*0x10000
```

```
+Size[2]*0x100
```

```
+Size[3];
```

3).标志

只定义了 6 位,另外的 10 位为 0,但大部分的情况下 16 位都为 0 就可以了。格式如下:

```
abc00000 ijk00000
```

a -- 标签保护标志,设置时认为此帧作废

b -- 文件保护标志,设置时认为此帧作废

c -- 只读标志,设置时认为此帧不能修改(但我没有找到一个软件理会这个标志)

i -- 压缩标志,设置时一个字节存放两个 BCD 码表示数字

j -- 加密标志(没有见过哪个 MP3 文件的标签用了加密)

k -- 组标志,设置时说明此帧和其他的某帧是一组

值得一提的是 winamp 在保存和读取帧内容的时候会在内容前面加个'\0',并把这个字节计算在帧内容的大小中。

附:帧标识的含义

4). Declared ID3v2 frames

The following frames are declared in this draft.

AENC Audio encryption

APIC Attached picture

COMM Comments

COMR Commercial frame

ENCR Encryption method registration

EQUA Equalization

ETCO Event timing codes

GEOB General encapsulated object

GRID Group identification registration

IPLS Involved people list

LINK Linked information

MCDI Music CD identifier

MLLT MPEG location lookup table

OWNE Ownership frame

PRIV Private frame

PCNT Play counter

POPM Popularimeter

POSS Position synchronisation frame

RBUF Recommended buffer size

RVAD Relative volume adjustment

RVRB Reverb

SYLT Synchronized lyric/text

SYTC Synchronized tempo codes

TALB Album/Movie/Show title

TBPM BPM (beats per minute)

TCOM Composer

TCON Content type

TCOP Copyright message

TDAT Date

TDLY Playlist delay

TENC Encoded by

TEXT Lyricist/Text writer

TFLT File type

TIME Time

TIT1 Content group description

TIT2 Title/songname/content description

TIT3 Subtitle/Description refinement

TKEY Initial key

TLAN Language(s)

TLEN Length

TMED Media type

TOAL Original album/movie/show title

TOFN Original filename

TOLY Original lyricist(s)/text writer(s)

TOPE Original artist(s)/performer(s)

TORY Original release year

TOWN File owner/licensee

TPE1 Lead performer(s)/Soloist(s)

TPE2 Band/orchestra/accompaniment

TPE3 Conductor/performer refinement

TPE4 Interpreted, remixed, or otherwise modified by

TPOS Part of a set

TPUB Publisher

TRCK Track number/Position in set

TRDA Recording dates

TRSN Internet radio station name

TRSO Internet radio station owner

TSIZ Size

TSRC ISRC (international standard recording code)

TSSE Software/Hardware and settings used for encoding

TYER Year

TXXX User defined text information frame

UFID Unique file identifier

USER Terms of use

USLT Unsynchronized lyric/text transcription

WCOM Commercial information

WCOP Copyright/Legal information

WOAF Official audio file webpage

WOAR Official artist/performer webpage

WOAS Official audio source webpage

WORS Official internet radio station homepage

WPAY Payment

WPUB Publishers official webpage

WXXX User defined URL link frame

五、MP3 文件实例剖析

在 VC++ 中打开一个名为 **test.mp3** 文件，其内容如下：

```
000000 FF FB 52 8C 00 00 01 49 09 C5 05 24 60 00 2A C1
000010 19 40 A6 00 00 05 96 41 34 18 20 80 08 26 48 29
000020 83 04 00 01 61 41 40 50 10 04 00 C1 21 41 50 64
.....
0000D0 FE FF FB 52 8C 11 80 01 EE 90 65 6E 08 20 02 30
```

0000E0 32 0C CD C0 04 00 46 16 41 89 B8 01 00 08 36 48

0000F033 B7 00 00 01 02 FF FF FF F4 E1 2F FF FF FF FF

.....

0001A0 DF FF FF FB 52 8C 12 00 01 FE 90 58 6E 09 A0 02

0001B0 33 B0 CA 85 E1 50 01 45 F6 19 61 BC 26 80 28 7C

0001C0 05 AC B4 20 28 94 FF FF FF FF FF FF FF FF FF

.....

001390 7F FF FF FF FD 4E 00 54 41 47 54 45 53 54 00 00

0013A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

.....

0013F000 00 00 00 00 04 19 14 03 00 00 00 00 00 00 00

001400 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

001410 00 00 00 00 00 00 4E

该文件长度 1416H（5.142K），帧头为：FF FB 52 8C，转换成二进制为：

11111111 11111011

01010010 10001100

对照表 1 可知，test.mp3 帧头信息见表 5。

表 5 test.mp3 文件帧头信息

名称	位值	说 明
同步信息	111111111111	第 1 字节恒为 FF，11 位均为 1。
版本	11	MPEG 1
层	01	Layer 3
CRC 校验	1	不校验
位率	0101	64kbps
频率	00	44.1kHz
帧长调节	1	调整，帧长是 210 字节。
保留字	0	没有使用。
声道模式	10	双声道
扩充模式	00	未使用。

版权	1	合法
原版标志	1	原版
强调方式	00	未定义

第 1397H 开始的三个字节是 54 41 47，存放的是字符“TAG”，表示此文件有 ID3 V1.0 信息。

139AH 开始的 30 个字节存放歌名，前 4 个非 00 字节是 54 45 53 54，表示“TEST”；

13F4H 开始的 4 个字节是 04 19 14 03，存放年份“04/25/2003”；

最后 1 个字节是 4E，表示音乐类别，代号为 78，即“Rock&Roll”；

其它字节均为 00，未存储信息。

六、资料

<http://www.id3.org/>