

GY

中华人民共和国广播电影电视行业标准

GY/T 168 - 2001

广播音频数据文件格式规范 广播波形格式 (BWF)

Specification of the format for audio data files in broadcasting——
The broadcast wave format (BWF)

2001-03-09 发布

2001-05-01 实施

国家广播电影电视总局 发布

目 次

前言

1 范围.....	1
2 引用标准.....	1
3 缩略语.....	1
4 广播波形格式 (BWF) 文件	1
5 采用 MPEG 音频的 BWF 文件	4
附录 A (标准的附录)RIFF WAVE (.WAV) 文件格式.....	7
附录 B (标准的附录)MPEG-1 音频.....	12

前 言

制定本标准是为了实现我国广播电台、电视台、唱片社及其它节目录制单位，基于不同计算机平台的音频工作站之间的数字音频文件录制和无缝交换。

本标准是根据我国广播电视的技术规范，参照 Rec. ITU-R BR. 1352 号建议书《信息技术媒体的音频节目资料交换文件格式》和该建议书的附录 1《广播波形格式规范 一种广播音频数据文件格式》、附录 2《广播波形格式规范 一种广播音频数据文件格式：MPEG 音频》，以及 EBU R85(1997)号技术建议《广播波形格式用于音频数据文件交换》、EBU R98(1999)号技术建议《广播波形格式文件 BWF 中 <CodingHistory>字段的格式》，在资源交换文件格式 RIFF 所定义的音频 WAVE 文件格式基础上，结合我国目前数字音频普遍采用 PCM 和 MPEG-1 编码方式的实际情况制定。

本标准的附录 A 和附录 B 是标准的附录。

本标准由全国广播电视标准化技术委员会归口。

本标准起草单位：电子科技大学、北京人民广播电台。

本标准主要起草人：黄晓革、章通、黄俊、周南、甘涛。

中华人民共和国广播电影电视行业标准

广播音频数据文件格式规范 广播波形格式 (BWF)

GY/T 168 - 2001

Specification of the format for audio data files in broadcasting

The broadcast wave format (BWF)

1 范围

本标准规定了一种专业广播领域内的数字音频文件格式和技术要求。

本标准适用于采用数字音频线性 PCM 和 MPEG-1 编码方式的广播电台、电视台、唱片社及其它节目录制单位基于不同计算机平台的音频工作站之间的数字音频文件录制和无缝交换。

2 引用标准

下列标准所包含的条文，通过在本标准中引用而构成为本标准的条文。本标准出版时，所示版本均为有效。所有标准都会被修订，使用本标准的各方应探讨使用下列标准最新版本的可能性。

GB/T 17975.3-2001 信息技术 运动图像及其伴音信号的通用编码 第3部分：音频
GY/T 156-2000 演播室数字音频参数

3 缩略语

3.1 RIFF

RIFF 是 resource interchange file format 的缩写，其意为资源交换文件格式。

3.2 WAVE

WAVE 是 waveform audio file format 的简写，其意为波形音频文件格式。

3.3 PCM

PCM 是 pulse code modulation 的缩写，其意为脉冲编码调制。

3.4 MPEG

MPEG 是 moving picture experts group 的缩写，其意为运动图像专家组。

3.5 BWF

BWF 是 broadcast wave format 的缩写，其意为广播波形格式。

4 广播波形格式 (BWF) 文件

4.1 广播波形格式 (BWF) 文件组成

广播波形格式 (BWF) 文件必须以 RIFF 的文件头开始，并至少包括下列数据块：

RIFF (' WAVE '	// RIFF 的文件头及 WAVE 标识符
<broadcast_audio_extension>	// 广播音频扩展数据块：音频序列的信息
<fmt_ck>	// 音频信号的格式：PCM 或 MPEG
[<fact_ck>]	// 仅 MPEG 格式要求的实际数据块

```
[<mpeg_audi_o_extensi_on>] // 仅 MPEG 格式要求的 MPEG 音频扩展数据块
<wave_data> ) // 音频数据
```

注：BWF 文件中出现的非上述的其它任何数据块被视作该 BWF 文件的自定义块，或者是将定义其它数据块以携带或指向某些应用所需的特定数据，例如：音频编辑和存档。其它应用程序可不必解释和运用这些自定义块，这些自定义中的数据完整性得不到保证。然而基于 BWF 的应用程序将尽可能传递这些自定义块。

4.2 RIFF 标准中已定义的数据块

有关技术规范中已定义了 RIFF 标准，本标准采用了其中一些已定义的数据块

它们是：

```
fmt_ck
```

```
fact_ck
```

这些数据块的解释参见附录 A。

4.3 广播音频扩展数据块

在广播音频扩展数据块中明确了用于音频资料数据文件交换所需的基本参数，其结构定义如下：

```
struct{
    DWORD chID; // 标识符：chID= ' bext ' */
    DWORD chSize; // 扩展数据块的长度 */
    BROADCAST_EXT chData; // 扩展数据块的数据 */
}
```

其中扩展数据块的数据结构 BROADCAST_EXT 定义如下：

```
typedef struct broadcast_audi_o_extensi_on
{
    CHAR Description[256]; // ASCII 码：音序列的描述*/
    CHAR Originator[32]; // ASCII 码：创作者的姓名*/
    CHAR OriginatorReference[32]; // ASCII 码：创作者的有关信息*/
    CHAR OriginationDate[10]; // ASCII 码：创作日期<<yyyy: mm: dd>>*/
    CHAR OriginationTime[8]; // ASCII 码：创作时间<<hh: mm: ss>>*/
    DWORD TimeReferenceLow; // 参考时间低位字*/
    DWORD TimeReferenceHigh; // 参考时间高位字*/
    WORD Version; // BWF 的版本，无符号的二进制表示*/
    CHAR Reserved[254]; // 保留字，设为 Null (空字符)*/
    CHAR CodingHistory[]; // ASCII 码：编码历史*/
} BROADCAST_EXT
```

具体字段说明：

Description: ASCII 码字符串（最长 256 个字符），包含了该音序列的说明。由于某些应用程序需要提供简短说明，建议开始 64 个字符用作简短描述，剩余 192 个字符用作详细说明。如字符串长度不足 256 个字符，剩余位用空字符(00)_{hex} 填补。

Originator: ASCII 码字符串（最长 32 个字符），包含了音频文件的创作者姓名的说明。如不足 32 个字符，剩余位用空字符(00)_{hex} 填补。

OriginatorReference: ASCII 码字符串 (最长 32 个字符), 包含了创作者的参考信息。如不足 32 字符, 剩余位用空字符 (00)_{hex} 填补。

OriginationDate: 10 个字符的 ASCII 码字符串, 包含了音频序列的生成日期。

格式为: 年-月-日

年为 4 个字符: 0000 到 9999

月为 2 个字符: 01 到 12

日为 2 个字符: 01 到 28, 29, 30 或 31

当中的间隔符为以下几种: 连字符(-), 下划线(_), 冒号(:), 空格(), 停止符(.).

OriginationTime: 8 个字符的 ASCII 码字符串, 包含了音频序列产生的时间。

格式为: 时-分-秒

时: 00 到 24

分: 00 到 59

秒: 00 到 59

当中的间隔符为以下几种: 连字符(-), 下划线(_), 冒号(:), 空格(), 停止符(.).

TimeReference: 参考时间字段, 该字段包含了该序列的时间码, 是 64 比特数值, 为从午夜零点开始到产生音频序列时的第一次取样之间的取样个数, 可用于精确地产生时间码。每秒的取样个数, 取决于取样频率, 它在<fmt_ck> (格式数据块) 中的字段<nSamplesPerSec> 中定义。

Version: 版本字段, 无符号的二进制数给出 BWF 的版本号, 特别是指保留字段的内容。初始值设置为 0。

Reserved: 保留字段, 为扩展保留的 254 个字节, 如果版本字段 Version 值为 0, 这 254 个字节必须被设为 Null (空) 值。

CodingHistory: 包含多行以回车符 CR/LF 结尾的 ASCII 码字符串, 每行字符串记录了每次编码处理过程, 每一次新的编码操作都将在该字段中增加一行新的包含有关该次编码参数变量值的字符串, 每行字符串的具体语法见表 1。

表 1 CodingHistory 字段语法

参数	变量串<可选值>	单位	备注
编码算法	A=<ANALOGUE ,PCM ,MPEG1L1 ,MPEG1L2 , MPEG1L3 ,MPEG2L2 ,MPEG2L3>		
取样频率	F=<11025 , 16000 , 22050 , 24000 , 32000 , 44100 , 48000>	[Hz]	
比特率	B=< MPEG 2(ISO/IEC 13818-3)允许的 任意比特率>	[kbit/s per channel]	仅用于 MPEG 编码
字长	W=<8 , 12 , 14 , 16 , 18 , 20 , 22 , 24>	[bits]	对于 MPEG 而言, 是指输入 MPEG 编码器的 PCM 码的字长
模式	M=<mono , stereo , dual -mono , joint-stereo>		

表 1 CodingHistory 字段语法(完)

参数	变量串<可选值>	单位	备注
文字,任意字符	T=<某些应用中需要明确的任意 ASCII 码文字字符串,如:标示号;编码类型;A/D 类型。注意该字符串中不能有逗号(ASCII 码为 2C _{hex})>		
注:变量字符串以逗号(ASCII 码为 2C _{hex})分开,每行以回车符 CR/LF(ASCII 码为 0D _{hex} 0A _{hex})结尾。			

5 采用 MPEG 音频的 BWF 文件

利用 BWF 文件传送 MPEG 音频,需要增加以下信息:

- 格式数据块的扩展;
- 实际(fact)数据块;
- MPEG 扩展数据块。

MPEG 音频数据已组织在 WAVE 文件中。格式数据块的扩展和实际数据块含有 MPEG 编码所需的相关信息,也作为 WAVE 格式的一部分,其原理在附录 B 中详细介绍。对于 MPEG 第二层,需要提供更多的有关信号的编码信息,这些信息在 MPEG 音频扩展(<mpeg_audio_extension>)数据块中携带。

5.1 MPEG 音频扩展(<mpeg_audio_extension>)数据块

MPEG 音频扩展数据块结构定义如下:

```
struct {
    DWORD    ckID ;                /* 该数据块标识符:ckID = 'mext' */
    DWORD    ckSize ;             /* 该数据块长度:ckSize = '000C' */
    MPEG_EXT ckData ;            /* 该数据块数据 */
}
```

其中 MPEG 音频扩展数据块的数据结构 MPEG_EXT 定义如下:

```
typedef struct mpeg_audio_extension{
    WORD     SoundInformation ;    /* 附加音频信息 */
    WORD     FrameSize ;          /* 每帧长度 */
    WORD     AncillaryDataLength ; /* 辅助数据的长度 */
    WORD     AncillaryDataDef ;   /* 辅助数据的类型 */
    CHAR     Reserved[4] ;        /* 保留字段,设为 Null(空字符) */
} MPEG_EXT ;
```

具体字段说明:

SoundInformation: 16 比特数表示音频文件附加信息,对 MPEG 层 2(或层 1):

Bit0: '1' 表示同类音频数据。

'0' 表示非同类音频数据。

Bit1 和 Bit2 为同类音频文件提供附加信息:

Bit1: '0' 表示在文件中使用补充位(Padding bit),其为 '0' 和 '1'交替。

‘1’表示补充位在整个文件中设置为0。

Bit2：‘1’表示文件中包含了一帧补充位为‘1’、取样频率为22.05kHz或44.1kHz的序列。

Bit3：‘0’表示使用自由格式。

‘1’表示使用非自由格式音频帧。

Bit4至15：未作定义。

FrameSize：16比特数表示音频文件一个正常帧中的字节数。这个字段仅对同类文件有意义，非同类文件应设为0。如没有使用补充位，即在整个文件中补充位保持不变，则字段<FrameSize>值与格式数据块中字段<nBlockAlign>的值相同。如果使用了补充位，音频数据长度是变化的，字段<FrameSize>内容是补充位为0的每帧长度，补充位为‘1’时的帧长等于FrameSize+1（层1为4字节）。

事实上当<nBlockAlign>设置为‘1’时，帧长将根据补充位决定为FrameSize或FrameSize+1。

AncillaryDataLength：16比特数表示的音频文件中辅助数据的最小字节长度，这个值是从音频帧末尾算起的。

AncillaryDataDef：16比特数指定辅助数据的内容：

Bit0设置为‘1’：辅助数据中表示了左声道能量。

Bit1设置为‘1’：辅助数据中一个自定义比特内部自由使用。

Bit2设置为‘1’：辅助数据中表示了右声道能量。

Bit3设置为‘0’：为APR数据保留待用。

Bit4设置为‘0’：为DAB数据保留待用。

Bit5设置为‘0’：为J52数据保留待用。

Bit6至15设置为‘0’：保留待用。

注：辅助数据中各项内容按照AncillaryDataDef中的定义顺序，从后往前排列，第一项储存在辅助数据的末尾，第二项储存在第一项之前。

- 对于单声道文件，bit2通常设置为‘0’，bit0与单声道帧的能量相关；
- 对于立体声文件，当bit2为‘0’且bit0为‘1’时，表示左右声道能量的最大值；
- 帧能量储存在2个字节中，为用于编码该帧的取样点中绝对值的最大值，为15比特的高位收尾（Big Endian）格式值。

Reserved：4个保留字节，必须设置为Null（空），在以后的运用中，Null作为默认值以保持兼容性。

5.2 同类(Homogeneous)音频数据

对于同类音频数据，除了以下情况，音频文件的MPEG帧应具有相同的帧头：

- 如果取样频率为44.1kHz或22.05kHz，或比特率字段设置为自由格式（‘0000’），则补充位可在‘0’和‘1’二者之间变化；
- 模式字段（M）可在立体声（M=‘00’）、相关立体声（M=‘01’）和双声道（M=‘10’）之间选择。如果模式字段为单声道（M=‘11’），则在整个文件中应保持一致；
- 对于相关立体声（M=‘01’），模式扩展字段（Mx）在整个文件中可变；

- 自定义比特 (Pr) 在文件中可以变化。

要特别注意除了取样频率为 44.1kHz 或 22.05kHz 和自由格式外，文件中的所有帧应有相同长度。

自由格式允许文件中的帧长为 N 或 N+P 字节。

当长度为 N 时，补充位设置为 '0'

当长度为 N+P 时，补充位设置为 '1'

N 值取决于比特率和取样率

对于层 1 压缩：P= '4'

对于层 2 压缩：P= '1'

5.3 非同类音频数据

一个非同类音频文件包含一个无任何限制的 MPEG 帧序列。

附录 A
(标准的附录)
RIFF WAVE (.WAV) 文件格式

本附录资料来自 RIFF 文件格式规范，仅作为资料引用。

A1 波形音频文件格式 (WAVE)

A1.1 WAVE 文件格式定义

一般定义为：

```

RIFF ( ' WAVE '                               // RIFF 的文件头及 WAVE 标识符
      <fmt_ck>                                  // 格式数据块
      [<fact_ck>]                               // 实际数据块
      [<other_ck>]                             // 其它可选数据块
      <wave_data> )                            // 音频数据

```

其中有三个部分是任何 WAVE 文件必须包含的：

- RIFF 的文件头及 WAVE 标识符；
- 格式数据块 (<fmt_ck>)，包含了解释音频数据时所需的指导信息；
- 波形音频文件数据 (<wave_data>)，包含音频数据。

如果音频数据不是 PCM 格式，WAVE 文件中还应增加<fact_ck>，提供更多的音频数据编码信息。象对待所有其它 RIFF 文件一样，程序若遇到它不能识别的块，可以跳过并忽略该块。

A1.2 WAVE 格式数据块(<fmt_ck>)

格式数据块<fmt_ck>描述了<wave_data>块的格式，其定义如下：

```

fmt ( <common_fi el ds>
      <format_speci fi c_fi el ds> )

```

其中

<common_fi el ds> 定义为：

```

struct{
    WORD    wFormatTag;
    WORD    nChannel s;
    DWORD   nSampl ePerSec;
    DWORD   nAvgBytesPerSec;
    WORD    nBl ockAl i gn;
}

```

<common_fi el ds>部分中的各字段定义为：

字段名	描 述
wFormatTag	标明 WAVE 文件的格式种类。格式数据块<fmt_ck>中<format_specific_fields>的值和<wave_data>块数据的解释方法都取决于该值。
nChannels	波形数据所表示的声道数，单声道是 1，立体声是 2。
nSamplesPerSec	各通道回放的取样频率。
nAvgBytesPerSec	每秒平均的字节数。回放软件可以根据该值来预测缓冲区的大小。
nBlockAlign	波形数据的块对齐校准（以字节为单位），回放软件需要同时多个<nBlockAlign>个字节的数据，所以该值用于实现缓冲区的对齐校准。

<format_specific_fields>包含了零个或更多的参数字节。这些参数取决于波形格式的种类，参看下述部分。回放软件的设计中允许在<format_specific_fields>域末尾出现的任何不认识的参数，并能忽略它。

A1.3 WAVE 格式种类

WAVE 文件的格式种类由格式数据块<fmt_ck>中<wFormatTag>字段的值来定，<wave_data>的数据表示方法和格式数据块<fmt_ck>中<format_specific_fields>的值都取决于该值。

当前定义的 WAVE 格式种类有：

wFormatTag	值	格式种类
WAVE_FORMAT_PCM	(0x0001)	脉冲编码调制(PCM)格式
WAVE_FORMAT_MPEG	(0x0050)	MPEG-1 音频格式

A2 脉冲编码调制 (PCM) 格式

如果<fmt_ck>中的<wFormatTag>字段被设置为 WAVE_FORMAT_PCM，则取样所组成的波形数据都以脉冲编码调制 (PCM) 格式表示。

对于 PCM 波形数据，<format_specific_fields>定义如下：

```
struct{
    WORD nBitsPerSample          /*量化比特值*/
}
```

<nBitsPerSample>字段为表示每个样点的量化比特值，如果是多个通道，则每个通道的量化比特数是一样的。

对于 PCM 数据而言，<nAvgBytesPerSec>字段应该等于按下列公式计算出的值：

$$(nChannels \times nBitsPerSecond \times nBitsPerSample) / 8$$

<nBlockAlign>字段应该等于按下列公式计算出的值：

$$(nChannels \times nBitsPerSample) / 8$$

A2.1 PCM WAVE 文件的数据包

在单声道的 WAVE 文件中, 取样点连续存放。而在立体声的 WAVE 文件中, 通道 0 表示左声道, 通道 1 表示右声道。多于两声道的目前尚未定义, 在多通道 WAVE 文件中, 取样点是交叉存放的。

以下表格显示 8 比特单声道和立体声 WAVE 文件的数据包格式:

8 比特单声道 PCM 的数据包

Sample 1	Sample 2	Sample 3	Sample 4
Channel 0	Channel 0	Channel 0	Channel 0

8 比特立体声 PCM 的数据包

Sample 1		Sample 2	
Channel 0 (左)	Channel 1 (右)	Channel 0 (左)	Channel 1 (右)

以下表格显示 16 比特单声道和立体声 WAVE 文件的数据包格式:

16 比特单声道 PCM 的数据包

Sample 1		Sample 2	
Channel 0 低字节	Channel 0 高字节	Channel 0 低字节	Channel 0 高字节

16 比特立体声 PCM 的数据包

Sample 1			
Channel 0 (左) 低字节	Channel 0 (左) 高字节	Channel 1 (右) 低字节	Channel 1 (右) 高字节

A2.2 取样点数据格式

每个样本包含于一个整数 i , i 的大小为能容纳规定的样本量化比特值 ($n\text{BitsPerSample}$) 所需的最小字节数。最低有效字节首先存放, 代表样本值幅度的比特存放在 i 的最高有效位, 剩余位设置为 0。例如, 如果样本量化比特值是 12 比特, 占 2 个字节, 第一个字节的最低有效 4 比特设置为 0。PCM 的各种量化比特的最大、最小值和数据格式如下:

量化比特	数据格式	最大值	最小值
1-8 比特	无符号实数	255 (0xFF)	0
9 比特或 9 比特以上	符号实数	i 的最大正值	i 的最小负值

例如: PCM 8 比特和 16 比特量化的最大、最小、中间值如下:

格式	最大值	最小值	中间值
8 比特 PCM	255 (0xFF)	0	128 (0x80)
16 比特 PCM	32767(0x7FFF)	-32768(-0x8000)	0

A2.3 PCM WAVE 文件举例

例：PCM WAVE 文件，48kHz 的取样频率，立体声，每样值 16 比特量化：

```
RIFF( ' WAVE ' fmt ( 1, 2, 48000, 96000, 2, 16)
      data(<wave_data> ) )
```

例：PCM WAVE 文件，44.1kHz 的取样频率，立体声，每样值 20 比特量化：

```
RIFF( ' WAVE ' fmt ( 1, 2, 48000, 144000, 3, 20)
      data(<wave_data> ) )
```

A2.4 WAVE 数据存储

<wave_data>包含波形数据，其定义如下：

```
data ( <wave_data> )
```

A2.5 实际数据块<fact_ck>

<fact_ck>块包含了 WAVE 文件的重要信息，定义如下：

```
fact ( < dwFileSize : DWORD> ) //取样次数
```

PCM 文件不需要这个块。

在<dwFileSize>字段后可新增字段，以扩展实际数据块，使其包含将来 WAVE 格式所需的任何其它信息，应用程序根据块大小来决定当前有哪些字段。

A2.6 其它可选择数据块

WAVE 格式中还指定运用了一些其它块，它们的详细资料在 WAVE 格式规范和今后更新发布中给出。

注：WAVE 格式可支持其它可选数据块以携带特殊信息，正如本标准正文 4.1 节所述，在广播波形格式（BWF）中，这些块被看作自定义块，如应用程序不能解释，将被忽略。

A3 其它 WAVE 类型

A3.1 通用信息

所有新定义的 WAVE 类型必须包含一个实际数据块（<fact_chunk>）和一个扩展的波形格式描述的格式数据块（<fmt_ck>），PCM 格式（WAVE_FORMAT_PCM）的 RIFF WAVE 文件既不需要额外实际数据块，也不需要波形格式描述。

A3.2 实际（fact）数据块

这个块储存了有关 WAVE 文件内容的信息，当前它指定了以取样点数量度的文件长度。

A3.3 WAVE 格式扩展

为定义所有非 PCM 格式的波形数据，需在<fmt_ck>块中新增扩展波形格式结构，该结构的一般描述为：

```
typedef struct waveformat_extended_tag {
    WORD wFormatTag; /* 格式类型*/
    WORD nChannels; /* 通道数（如单声道，立体声）*/
    WORD nSamplesPerSec; /* 取样率 */
    WORD nAvgBytesPerSec; /* 用于缓冲区大小估计 */
    WORD nBlockAlign; /* 数据块长度 */
}
```

```

    WORD wBitsPerSample;    /* 单声道数据中每个样本的比特数 */
    WORD cbSize;            /* 该扩展部分长度的字节数 */
} WAVEFORMATEX;

```

字段名	描述
wFormatTag:	定义 WAVE 文件类型。
nChannels:	波形文件的通道数，单声道为 1，立体声为 2。
nSamplesPerSec:	波形文件的取样率，应为 48000 或 44100 等，它也用于实际数据块中取样长度的登录，并决定数据的时间长度。
nAvgBytesPerSec:	平均数据率，回放软件用该值来估计缓冲区大小。
nBlockAlign:	数据块中数据以块对齐校准（以字节为单位），回放软件需要同时处理多个 <nBlockAlign> 个字节的数据，所以该值用于实现缓冲区的对齐校准。
wBitsPerSample:	每通道每个样本的比特数，每一通道被认为有相同的取样分辨率，若不需要该值，它应设置为“0”。
cbSize:	WAVE 格式头中额外信息的字节数，不包含 WAVEFORMATEX 结构的长度。

注：在字段<cbSize>之后包含了定义在字段<wFormatTag>的 WAVE 所需的专门信息，任何可以用于 BWF 的 WAVE 格式已在本标准的第 5 节中作了说明。

附录 B
(标准的附录)
MPEG-1 音频

B1 实际数据块

除 WAVE_FORMAT_PCM 格式的所有 WAVE 格式，都有实际数据块(fact_ck)，它储存了 WAVE 数据内容的相关信息。它通常以取样个数为单位标明数据的时间长度。

B2 WAVE 格式的文件头

```
#define WAVE_FORMAT_MPEG( 0x0050 )
    typedef struct mpeg1waveformat_tag{
        WAVEFORMATEX    wfx;
        WORD             fwHeadLayer;
        DWORD            dwHeadBi trate;
        WORD             fwHeadMode;
        WORD             fwHeadModeExt;
        WORD             wHeadEmphasi s;
        WORD             fwHeadFl ags;
        DWORD            dwPTSLow;
        DWORD            dwPTSHi gh;
    }MPEG1WAVEFORMAT;
```

wFormatTag：应设置为 WAVE_FORMAT_MPEG[0x0050]。

nChannels：波形数据所表示的通道数，单声道是 1，立体声是 2。

nSamplesPerSec：波形数据文件的取样率：32000，44100 或 48000 等。注意如数据的取样频率是变化的，这个段应设为 0。在桌面应用中，极力推荐采用固定取样频率。

注：广播中通常取样频率为 48kHz。

nAvgBytesPerSec：平均数据率，如果在第三层压缩编码中可变化比特率时，平均数据率可能不是一个合法的 MPEG 比特率。

nBlockAlign：在数据块 (<data_ck>) 中的数据块校准对齐值 (以字节为单位)。在有固定帧长的音频数据码流中，该值等于帧长。在变化帧长的码流中，该值应设为 1。取样频率为 32kHz 或 48kHz 的 MPEG 音频文件的帧长是比特率的函数，如果音频码流为固定比特率，音频帧长是不变化的，下面给出相应公式：

层 1 压缩：
$$nBlockAlign = 4 * (int)(12 * 比特率 / 取样频率)$$

层 2 和层 3 压缩：
$$nBlockAlign = (int)(144 * 比特率 / 取样频率)$$

例 1：层 1 压缩，取样频率为 32000Hz，比特率为 256Kbit/s，

nBlockAlign=384 字节 (bytes)

如果一个音频码流中包含有不同比特率的帧，则码流中的帧长是变化的。当采用 44.1kHz 频率取样时，帧长也会发生变化。为了把数据传输率保持在额定值，为符合上面给出的公式，MPEG 音频文件帧长应周期性地增加 1 个时隙 (slot) (层 1 压缩为 4 个字节，层 2、层 3 压缩为 1 个字节)。在这两种情况下，块对齐方式无效。<nBlockAlign>的值应设为 '1'，以便处理 MPEG 的应用程序判断数据是否对齐。

注：可以通过在音频文件帧头加入有相同值 ('1' 或 '0') 的填充比特位，来建立一个 44.1kHz 有恒定帧长的音频码流，这个码流比特率会和帧头中的额定值不完全相同，因此一些解码器可能不能准确解码，且考虑到标准化和兼容性要求，不推荐采用这种方法。

wBitsPerSample：不用，设为 0。

cbSize：在 WAVEFORMATEX 结构后面的扩展信息的字节数，WAVE_FORMAT_MPEG 格式中，它是 22 (0x0016)。如果增加了额外字段，cbSize 的值应增加。

fwHeadLayer：MPEG 的音频压缩级由如下标志定义：

ACM_MPEG_LAYER1：层 1

ACM_MPEG_LAYER2：层 2

ACM_MPEG_LAYER3：层 3

一些合法的 MPEG 码流可能包含不同层的帧，这种情况下，以上标志位可用逻辑与式表达，以便驱动器判断码流中有哪些压缩层。

dwHeadBitrate：数据比特率，每秒比特数。它应是符合 MPEG 规范的一个标准比特率。并非所有比特率对所有模式所有层都有效。见表 B1 和表 B2。这个字段记录了实际比特率，而非 MPEG 帧的头代码。如果比特率是个变化值或它并非一个标准比特率，则这个段应设置为 0。推荐尽可能避免用变化的比特率编码。

表 B1 有效比特率 (bits/s)

MPEG 帧头编码	层 1	层 2	层 3
'0000'	自由格式	自由格式	自由格式
'0001'	32000	32000	32000
'0010'	64000	48000	40000
'0011'	96000	56000	48000
'0100'	128000	64000	56000
'0101'	160000	80000	64000
'0110'	192000	96000	80000
'0111'	224000	112000	96000
'1000'	256000	128000	112000
'1001'	288000	160000	128000

表 B1 有效比特率 (bits/s) (完)

MPEG 帧头编码	层 1	层 2	层 3
'1010'	320000	192000	160000
'1011'	352000	224000	192000
'1100'	384000	256000	224000
'1101'	416000	320000	256000
'1110'	448000	384000	320000
'1111'	禁止	禁止	禁止

表 B2 第 2 层有效模式 - 比特率组合

比特率 (bits/sec)	有效模式
32000	单声道
48000	单声道
56000	单声道
64000	所有模式
80000	单声道
96000	所有模式
112000	所有模式
128000	所有模式
160000	所有模式
192000	所有模式
224000	立体声, 强度立体声, 双声道
256000	立体声, 强度立体声, 双声道
320000	立体声, 强度立体声, 双声道
384000	立体声, 强度立体声, 双声道

fwHeadMode : 码流的模式由如下标志位定义 :

ACM_MPEG_STEREO : 立体声

ACM_MPEG_JOINTSTEREO : 相关立体声

ACM_MPEG_DUALCHANNEL : 双声道(例如: 双语言码流)

ACM_MPEG_SINGLECHANNEL : 单声道

一些合法的 MPEG 码流可能包含不同模式的帧, 这种情况下, 以上标志位可或 (OR) 在一起, 以便驱动程序判断码流中有哪些模式, 这种情况与相关立体声编码相似: 编码器能根据信号的特点, 动态地选择相关立体声或立体声, 这种情况下应设置 ACM_MPEG_STEREO 和 ACM_MPEG_JOINTSTEREO 两个

标志。

fwHeadModeExt : 包含了相关立体声编码的某些附加参数,其它模式中并不使用,见表 B3。一些合法的 MPEG 码流可能包含不同模式扩展的帧,这种情况下,表 B3 的值可或(OR)在一起。fwHeadModeExt 只用于相关立体声编码,在其它模式(单声道,双声道或立体声)中,它的值应设定为 0。通常,编码器能根据信号的特点动态地选择可能的 mode_extension 值,因此,对相关立体声,这个域应设置为 0x000f,但是,如果要限制编码的特殊类型,这个域应设置为相应的值。

表 B3 模式扩展

fwHeadModeExt	MPEG 帧头代码	层 1 和层 2	层 3
0x001	'00'	子带 4-31, 强度立体声	非强度或 MS 立体声
0x002	'01'	子带 8-31, 强度立体声	强度立体声
0x004	'10'	子带 12-31, 强度立体声	MS 立体声
0x008	'11'	子带 16-31, 强度立体声	强度或 MS 立体声

wHeadEmphasis : 提供解码器所需的去预加重信息,它表示在编码之前进行了预加重,见表 B4。

表 B4 预加重字段

wHeadEmphasis	MPEG 帧头代码	去加重要求
1	'00'	无预加重
2	'01'	50/15 μ s 预加重
3	'10'	保留
4	'11'	CCITTJ. 17

fwHeadFlags : 在音频帧头设定相应标志。

ACM_MPEG_PRIVATEBIT : 设定自定义位

ACM_MPEG_COPYRIGHT : 设定版权位

ACM_MPEG_ORIGINALHOME : 设定原始位

ACM_MPEG_PROTECTIONBIT : 设定保护位,同时在每帧中插入 16 比特纠错码。

ACM_MPEG_ID_MPEG1 : 设定 ID 位为 1,定义码流为 MPEG-1 音频码流,这个标志应明确设定以便与将来 MPEG 音频扩展兼容(即 MPEG2)。

编码器将根据这些状态位来设置 MPEG 音频帧头中相应位,当描述一个已编码的数据流时,每帧头的标志是这些标志的逻辑“或”结果,这就是说,

如果流中有 1 帧或多帧的版权为 1, ACM_MPEG_COPYRIGHT 位就应置 1。因此, 这些标志位并非对每个音频帧都是必要的。

dwPTSLow : 这个字段 (和下面介绍的字段一起) 构成了从 MPEG 系统层中提取出音频码流的第一帧的演出时间标记 (PTS)。dwPTSLow 包含 33 位 PTS 中的 32 个低比特位 (LSB)。PTS 将有助于音频流与相关视频流重新组合。如果音频流和系统层无关, 则应把该字段设为 0。

dwPTSHigh : 这个字段 (和上面介绍的字段一起) 构成了从 MPEG 系统层中提取出音频码流的第一帧的演出时间标记 (PTS)。dwPTSHigh 包含 33 位 PTS 中的高比特位 (MSB)。PTS 将有助于音频流与相关视频流重新组合。如果音频流和系统层无关则应把该字段设为 0。

注: 上面两字段可以看作一个 64 比特整数; dwPTSHigh 可以当作标志检测以确定 MSB 是否被设定或删除。

B3 数据段中标志的使用

fwHeadLayer

以下是<fwHeadLayer>段定义的标志位。在编码时应设定为其中一种标志以便让编码器知道采用哪一层编码。解码时, 驱动程序检测这些标志位以确定能否解码。一个合法 MPEG 流的不同帧可用不同层压缩, 因此, 可能设置多个标志位。

```
#define ACM_MPEG_LAYER1      (0x0001)
#define ACM_MPEG_LAYER2      (0x0002)
#define ACM_MPEG_LAYER3      (0x0004)
```

fwHeadMode

以下是<fwHeadMode>段定义的标志位。在编码时, 应设置其中一种标志位以便编码器知道用哪一种模式编码; 对相关立体声编码, 只有典型地将 ACM_MPEG_STEREO 和 ACM_MPEG_JOINTSTEREO 标志都设置, 以便当相关立体声编码比立体声编码更有效时, 编码器才能采用相关立体声编码。解码时, 驱动程序检测这些标志位以确定能否解码。一个合法 MPEG 码流中的不同帧可用不同层压缩, 因此, 可能有多个标志位被设置。

```
#define ACM_MPEG_STEREO      (0x0001)
#define ACM_MPEG_JOINTSTEREO (0x0002)
#define ACM_MPEG_DUALCHANNEL (0x0004)
#define ACM_MPEG_SINGLECHANNEL (0x0008)
```

fwHeadModeExt

表 B3 定义了<fwHeadModeExt>段的标志位。这个段仅为相关立体声编码使用, 采用其它模式时, 该段应设定为 0。在相关立体声编码中, 这些标志指出编码器允许使用的相关立体声编码类型。通常, 一个编码器能够动态地选择最适合输入信号的扩展模式。因此, 应用程序应将这个段设定为 0x000f, 以便编码器在所有可选模式中选择。同时可以通过清除一些标志位来限制编码器。对一个已编码的流, 这个段指出了码流中 MPEG_mode_extension 字段的值。

fwHeadFlags

以下是<fwHeadFlags>段定义的标志。这些标志应在编码前设定，以便在 MPEG 帧头中设定恰当的比特。当描述一个已编码的 MPEG 音频码流，这些标志位表示每个音频帧头相应位的逻辑或（OR），这就是说，只要任何一帧设置了标志位，<fwHeadFlags>将被设置。如果应用程序在预编码的 MPEG 音频比特流外，外包了一个 RIFF WAVE 头，它将负责作比特流的字段解析，并在这里设置相应标志。

```
#define ACM_MPEG_PRIVATEBIT      (0X0001)
#define ACM_MPEG_COPYRIGHT      (0X0002)
#define ACM_MPEG_ORIGINALHOME   (0X0004)
#define ACM_MPEG_PROTECTIONBIT  (0X0008)
#define ACM_MPEG_ID_MPEG1       (0X00010)
```

B4 MPEG 文件中的音频数据

在数据块<data chunk>中包含了一个 MPEG 音序列，这个序列在 GB/T 17975.3-2001《信息技术运动图像及其伴音信号的通用编码 第3部分：音频》中定义，它由一个比特流组成，作为字节数据组保存在数据块中。在一个字节中，MSB 是第一位，LSB 是最后一位。数据不是字节倒转，例如下面数据由一个典型音频帧头的前 16 比特（由左至右）组成：

```
SyncWord      ID      Layer      ProtectionBit  ...
111111111111  1      10      1              ...
```

这个数据在字节中以下列顺序存放：

```
字节(Byte)0    字节(Byte)1    ...
FF              FD              ...
```

B4.1 MPEG 音频帧

一个 MPEG 音频序列由一系列音频帧组成。每个音频帧都以一个帧头开始。这个帧头的大多数字段与前面 MPEG1WAVEFORMAT 结构定义的字段相对应。在编码时，这字段被设定在 MPEG1WAVEFORMAT 结构中，驱动程序在编码时根据这些信息，在帧头设定适当的比特位。解码时，驱动程序能检测这些字段以决定能否对该码流解码。

B4.2 编码

对 MPEG 音频码流编码，驱动程序应读出 MPEG1WAVEFORMAT 结构中的头字段，并在 MPEG 帧头中设定相应比特。如果驱动程序需要其它信息则应从配置对话框或回调函数中提取。参考下面附加数据章节可获取更多信息。

如果一个预编码的 MPEG 音频码流封装在 RIFF 文件头内，应用程序将负责分析比特流并设定 MPEG1WAVEFORMAT 中的字段。如果数据码流中的取样频率或比特率参数并非恒定的，驱动程序应置 MPEG1WAVEFORMAT 相应字段（<nSamplesPerSec>和<dwHeadBtrate>）为‘0’。如果码流中包含多层压缩的不同的帧，它应在<fwHeadLayer>中设置相应层标志。因为<fwHeadLayer>中字段随不同的帧而变化，在设置和测试这些标志时应特别注意。通常应用程序并不认为每帧中这些标志都有效。设定这些标志应遵循如下原则：

- 如码流中任何一帧的版权位被设置，则 ACM_MPEG_COPYRIGHT 被置位；
- 如码流中任何一帧的保护位被设置，则 ACM_MPEG_PROTECTIONBIT 被置位；
- 如码流中任何一帧的原创位被设置，则 ACM_MPEG_ORIGINALHOME 被置位。若是拷贝的流，该位应清零；
- 如码流中任何一帧的自定义被设置，则 ACM_MPEG_PRIVATEBIT 被置位；
- 如码流中任何一帧的标识位被设置，则 ACM_MPEG_MPEG1 被置位，对 MPEG1，标识位（ID）始终被置位。对于以后的 MPEG 扩展（如 MPEG-2 多声道格式）可能会将标识位（ID）清零。

如果 MPEG 音频码流是从系统层 MPEG 码流中提取的，或者它将加入系统层，可以使用演出时间标记（PTS），PTS 是在 MPEG 系统层中用来同步各字段，MPEG PTS 字段有 33 位，故 RIFF WAVE 格式头用两个字段储存该值，<dwPTSLow>包含 PTS 的低 32 位（LSBs），<dwPTShigh>包含其高位（MSB）。这两个字段可合起来看作一个 64 位整数，<dwPTShigh>字段可作为检测 MSB 是否被置位或清零的标志，当从系统层中抽取音频流时，驱动程序应设置 PTS 字段为第一帧音频数据的 PTS 值，它将为以后音频流重新加入系统层时所用。除此，PTS 字段将不作其它用途，如果音频码流不与 MPEG 系统层相关联，那么 PTS 字段将被置 0。

B4.3 解码

驱动程序可检测 MPEG1 WAVEFORMAT 结构中的字段，以决定它能否对该码流解码，驱动程序必须识别一些字段，如<fwHeadFlags>，在比特流中的每一帧中可能并非一致。它绝不能够按 MPEG1WAVEFORMAT 结构中的字段做实际解码，解码的参数应都从 MPEG 数据中提取。

驱动程序可检查<nSamplesPerSec>字段以决定它是支持指定的取样频率。如果 MPEG 流包含多取样率的数据，<nSamplesPerSec>字段将被置 0。如果驱动程序不能处理数据流类型，它将不试图去解码，而是立即失败退出。

B5 辅助数据

在 MPEG 音频帧中音频数据可能没有填充整个帧，其中一些剩余数据叫作辅助数据。这些数据可能是任何需要的格式，可用于传播各类附加信息。

如果驱动程序希望能支持辅助数据，就必须具备在应用程序中传送这些数据的途径。驱动程序可用回调函数来实现，基本上，驱动程序在向应用程序传递辅助数据（如解码）或需要更多的辅助数据（如编码）时，可调用特定的回调函数。

驱动程序应清楚不是所有的应用程序都要处理辅助数据，因此驱动程序只提供应用程序明确需要的服务，驱动程序可定义一个用户消息，决定是否使用回调函数功能，为提高灵活性，编码和解码操作需定义独立的消息。

中 华 人 民 共 和 国
广 播 电 影 电 视 行 业 标 准
广 播 音 频 数 据 文 件 格 式 规 范
广 播 波 形 格 式 (B W F)

GY/T 168—2001

*

国家广播电影电视总局标准化规划研究所出版发行

责任编辑：王佳梅

查询网址：www.sarft.gov.cn

北京复兴门外大街二号

联系电话：(010) 66093424 66092645

邮政编码：100866

版权专有 不得翻印

定价 25.00 元