

ITU-R BR.1352-3建议书

音频节目材料与元数据在信息技术媒体上交换的文件格式

(ITU-R 58/6号研究课题)

(1998-2001-2002-2007)

范围

本建议书规定广播音频扩展块¹ 以及其使用PCM编码和MPEG-1或MPEG-2音频数据的规范。还包括RIFF格式的基本信息以及将其扩展到其他类型音频数据的过程。

国际电信联盟无线电通信大会，

考虑到

- a) 基于信息技术的存储介质，包括数据光盘和磁带，已经渗透到采用无线广播的音频产品的所有领域，即非线性编辑、空中播放和存档；
- b) 本技术在运营灵活性、生产流程和站台自动化方面表现出强大的优势，并且因此在现有演播室的升级以及新的演播室设施的设计方面广受欢迎；
- c) 信号交换采用单一的文件格式极大简化了个别装备和远端演播室之间的互操作能力，它还便于编辑、空中播放和存档的综合操作；
- d) 记录与音频信号相关的元数据的文件中必须包括广播相关信息的最小集；
- e) 为保证复杂程度不同的应用程序之间的兼容性，必须对功能的最小集协商一致，它对所有能够处理被推荐文件格式的应用是共同的；
- f) ITU-R BS.646建议书定义用于无线电和电视广播的音频产品中使用的数字音频格式；
- g) 当ISO/IEC11172-3和ISO/IEC13818-3编码系统用于压缩信号时，也出现了交换音频材料的需求；
- h) 与现有的商业文件格式的兼容性可以最大程度地减少在设备中实现此格式所付出的行业代价；
- j) 历史信息的编码的标准格式将简化节目交换之后信息的使用；
- k) 音频信号的品质受信号处理过程的影响，特别是在比特率降低过程中受非线性编码和解码的使用的影响。

¹ “块”表示微软®资源交换文件格式（RIFF）中一个文件的基本结构块。

建议

- 1 对于信息技术媒体上的音频节目交换，应按照ITU-R BS.646建议书的相关规定设置音频信号参数、抽样频率、编码分辨率和预加重；
- 2 对于信息技术媒体上线性脉冲编码调制（PCM）格式的音频节目的交换，应采用附件1中规定的文件格式；
- 3 当音频信号的编码采用ISO/IEC11172-3或ISO/IEC13818-3编码系统时，在信息技术媒体²上音频节目的交换采用附件1中规定的并通过附件2补充的文件格式。
- 4 当采用附件1和/或2规定的文件格式，承载通过获取工作站（数字音频工作站（DAW））收集并计算的音频资料信息时，元数据应符合附件3中详述的规定。

附件 1

广播波格式的规范

广播中音频数据文件的格式

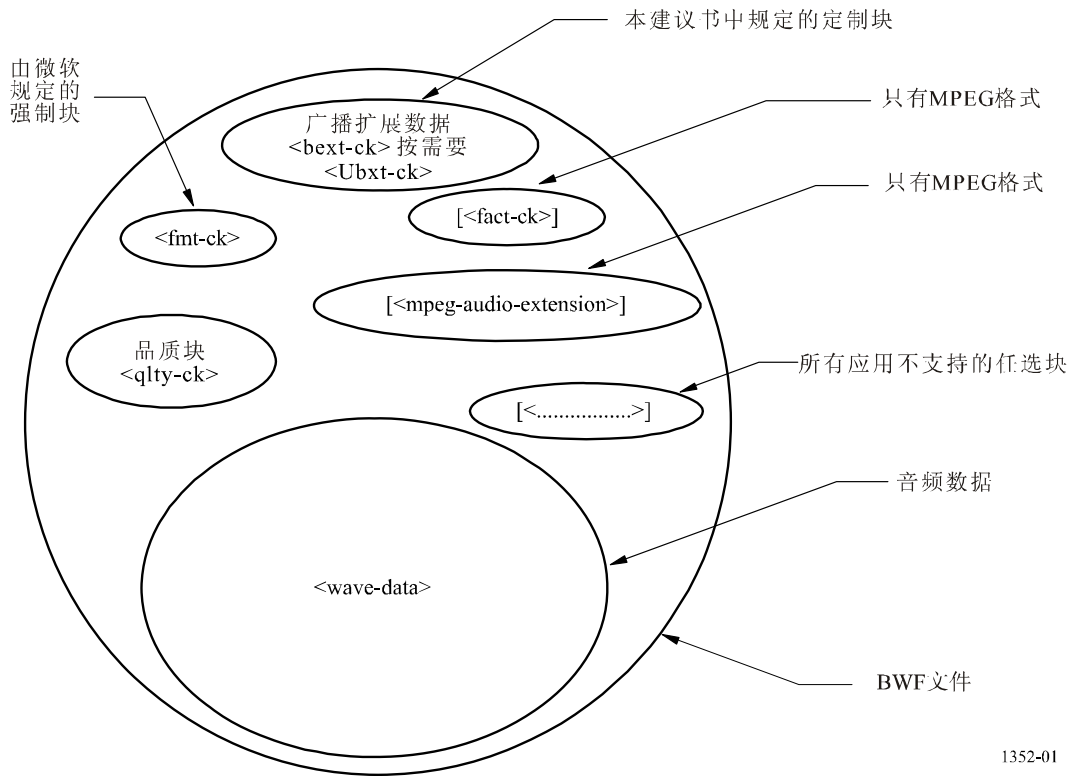
1 引言

广播波格式（BWF）基于微软®WAVE音频文件格式，它属于微软®“资源交换文件格式”RIFF中规定的文件类型中的一种。WAVE文件明确包括音频数据。RIFF文件格式的基本结构块，即简称为块，包括一组密切相关的信息元。它包括块标识符，一个代表字节的长度和承载的信息的整数值。一个RIFF文件由块的组合构成。

对于BWF，除BWF文件包括一个<广播音频扩展>块外，还对初始WAVE格式适用某些限制。如图1所示。

² 应当承认，在此意义上该建议书可能对那些使用某些计算机平台的开发者不利。

图1
BWF文件



1352-01

附件包括所有BWF文件中采用的广播音频扩展块的规范。另外，基本RIFF格式的信息以及其扩展到其他类型的音频数据的过程在附录1中描述。附录1还详述了PCM波格式。其他类型的音频数据扩展以及元数据的详细规范见附件2和3。

1.1 正式规定

遵守本建议书的规定采取自愿的原则。但是，本建议书也包括某些强制性条款（以保证例如互操作能力或适用性），因此当满足了所有强制条款后即认为遵守了本建议书。

使用“务必”或其他一些强制性的语言，例如“必须”，以及一些贬义的同义词来表述那些强制性条款。使用这样的词语并不意味着要求任何人遵守本建议书。

2 广播波格式（BWF）文件

2.1 广播波格式文件的内容

广播波格式文件的开始必须采用强制微软®RIFF “WAVE” 标题以及至少包括如下块：

```

<WAVE-form>
RIFF ('WAVE'
<fmt-ck> /*音频信号的格式：PCM/MPEG */
<broadcast_audio_extension> /*音频序列上的信息*/
<通用广播音频扩展> /*只支持多字节语言时要求ubxt */
<fact-ck> /*只有MPEG格式要求实际块*/
<mpeg_audio_extension > /*只有MPEG格式要求MPEG音频扩展块*/

<wave-data> ) /*声音数据 */
< quality-chunk > /*只当需要涉及影响品质的相关事件的信息
时要求此块*/

```

注 1 – 另外的块也可以出现在文件中。有些可能超出了本建议书讨论的范围。这些应用可能或可能不说明或使用这些块，因此不能保证该未知块中包括的数据的完整性。但是，符合要求的应用必须在未知块上透明地传送。

2.2 定义为RIFF 标准部分的现有的块

RIFF 标准在微软®³发布的文件中规定。此应用使用已经定义的一些块。其中有：

```

fmt-ck
fact-ck

```

这些块的当前描述信息见附件1的附录1。

2.3 广播音频扩展块⁴

在特定的“广播音频扩展”块中，增加用于广播电台之间交换材料所需要的额外参数如下：

```

broadcast_audio_extension typedef struct {
    DWORD ckID, /* (broadcastextension) ckID=bext.*/
    DWORD ckSize, /* 扩展块大小 */
    BYTE ckData[ckSize], /* 块的数据 */
}

typedef struct broadcast_audio_extension {
    CHAR Description[256], /* ASCII：“声音序列描述” /

```

³ 微软公司资源交换文件格式，RIFF (2005-12) 网址：
http://www.tactilemedia.com/info/MCI_Control_Info.html.

⁴ ubxt 块的定义参见第2.4节，以表示在多字节字符集中bext块的人类可读信息。

```

CHAR Originator[32],          /* ASCII：“始发方名称” /
CHAR OriginatorReference[32], /* ASCII：“始发方证明” /
CHAR OriginationDate[10],    /* ASCII：“yyyy：mm：dd” /
CHAR OriginationTime[8],    /* ASCII：“hh：mm：ss” /
DWORD TimeReferenceLow,     /* 自午夜起计算的第一个范例，低字*/
DWORD TimeReferenceHigh,   /* 自午夜起计算的第一个范例，高字*/
WORD 版本,                  /* BWF的版本；无符号二进制数 */
BYTE UMID_0,                /* SMPTE UMID 的二进制字节 0 */...
BYTE UMID_63,               /* SMPTE UMID的二进制字节 63 */
CHAR 保留[190],             /* 190字节，保留做将来使用，
                             设置为 .NULL.* /
CHAR CodingHistory[],       /* ASCII：“历史编码”
} BROADCAST_EXT,

```

字段**描述**

描述

包括对序列进行描述的ASCII串（最多256个字符）。为帮助只显示简短描述的应用，建议在第一个64字符中包括一个概述，最后192字符再用于详述。

如果串的长度小于256字符，则最后一个之后接一个空字符。（0x00）

始发方

包括音频文件的始发方/制造商名称的ASCII串（最大32字符）。如果串的长度小于32字符，字段由空字符结束。（0x00）

OriginatorReference

包括由始发机构分配的明确证明的ASCII串（最大32字符）。如果串的长度小于32字符，字段以空字符结束。（0x00）

OriginatorReference字段中使用的“唯一的”源标识符（USID）信息的标准格式在附件1的附录3中给出。

OriginationDate

包括音频序列的创建日期的10ASCII字符。格式是«‘，年’，-，’月，’-‘，日，’»年采用4字符其他项采用2字符。

年从0000至9999定义。

月从1至12定义。

日从1至31定义。

按照ISO8601，各项之间的间隔符应采用连字符。某些传统的实现可采用‘_’下划线，‘：’冒号，‘ ’空白，‘.’结束，音响设备应该认可这些分隔符字符。

OriginationTime

包括音频序列的创建时间的8ASCII字符。格式是«‘小时，’-‘，分钟，’-‘，秒’»每项2字符。

小时从0至23定义。

分钟和秒从0至59定义。

按照ISO8601，各项之间的间隔符应采用连字符。某些传统的实现可采用‘_’下划线，‘:’冒号，‘ ’空白，‘.’结束，音响设备应该认可这些分隔符字符。

TimeReference	此字段包含序列的时间码。它的长度是64比特，包括自午夜起计算的第一个抽样计数。每秒的抽样数取决于抽样频率，在来自<fmt-ck>的字段<nSamplesPerSec>中定义。
版本	BWF的版本采用一个无符号二进制数表示。对于版本1，设置为0x0001。
UMID	包括由SMPTE 330M定义的扩展的UMID的64字节。如果采用32字节的基本UMID，最后32字节应用0填充。如果没有UMID，64字节应用0填充。 注 – UMID的长度在UMID的开头编码。
保留	预留190个字节用于扩展。这190个字节应设置为0。
编码历史	包括0或多个串的块大小可变的ASCII字符，每个均以<CR><LF>结束，第一个未使用的字符应为空字符（0x00）。每串应包括适用于音频数据编码过程的描述。 每个新的编码应用应加入具有适当信息的新串。 历史信息的编码的标准格式在附件1的附录2。 此信息必须包括具有其特定参数的声音类型（PCM 或 MPEG）： PCM：模式（单声道、立体声）、抽样大小（8、16比特）和抽样频率， MPEG：抽样频率、比特率、层（I或II）以及模式（单声道、立体声、联合立体声或双通道）， 建议编码器的制造商提供在编码历史中使用的ASCII串。

2.4 通用广播音频扩展块

广播音频扩展（bext）块（第2.3节中定义）中包括的信息还可以由所谓的“通用广播音频扩展”或“ubxt”块承载，以表示多字节语言中bext块的人类可读信息。此元数据块的基本结构与bext块的相同。4个人类可读项目：uDescription、uOriginator、uOriginatorReference和uCodingHistory用UTF-8（8比特UCS传输格式）描述，而不采用ASCII。前三项具有8倍bext块中对应项的数据大小。ubxt块的结构定义如下：

```
typedef struct chunk_header {
    DWORD   ckID;           /* （通用广播扩展）ckID=ubxt */
    DWORD   ckSize;        /* 扩展块大小 */
};
```

```

    BYTE ckData[ckSize];           /* 块的数据 */
} CHUNK_HEADER;

typedef struct universal_broadcast_audio_extension {
    BYTE uDescription[256*8];      /* UTF-8：“声音序列描述” */
    BYTE uOriginator[32*8];       /* UTF-8：“始发方名称” */
    BYTE uOriginatorReference[32*8]; /* UTF-8：“始发方证明” */
    CHAR OriginationDate[10];     /* ASCII：“yyyy: mm:dd” */
    CHAR OriginationTime[8];     /* ASCII：“hh:mm:ss” */
    DWORD TimeReferenceLow;      /* 自午夜起计算的第一个抽样计数，低字 */
    DWORD TimeReferenceHigh;     /* 自午夜起计算的第一个抽样计数，高字 */
    WORD 版本;                  /* BWF的版本；无符号二进制数 */
    BYTE UMID_0;                 /* SMPTE UMID的二进制字节0 */
    BYTE UMID_63;                /* SMPTE UMID的二进制字节63 */
    CHAR 保留[190];             /* 190字节，保留做将来使用，设置为“NULL” */
    BYTE uCodingHistory[];       /* UTF-8：“编码历史” */
} UNIV_BROADCAST_EXT;

```

字段	描述
uDescription	UTF-8串，2 048字节或更少，包括序列描述。如果未得到数据或如果串的长度小于2 048字节，第一个未使用字节必须是空字符（0x00）
uOriginator	UTF-8串，256字节或更少，包括音频文件的始发方名称。如果未得到数据或如果串的长度小于256字节，第一个未使用字节必须是空字符（0x00）
uOriginatorReference	UTF-8串，256字节或更少，包括一个始发机构分配的证明。如果未得到数据或如果串的长度小于256字节，第一个未使用字节必须是空字符（0x00）
OriginationDate	包括音频序列的创建日期的10 ASCII 字符。格式是«‘，年’，-，’月，’-‘，日，’»年采用4字符其他项2字符。 年从0000至9999定义 月从1至12定义 日从1至31定义 按照ISO8601，各项之间的间隔符应采用连字符。某些传统的实现可采用‘_’下划线，‘:’冒号，‘ ’空白，‘.’结束；音响设备应该认可这些分隔符字符。
OriginationTime	包括音频序列的创建时间的8ASCII字符。格式是«‘小时，’-‘，分钟，’-‘，秒’»每项2字符。 小时从0至23定义。

分钟和秒从0到59定义。

按照ISO 8601，各项之间的间隔符应采用连字符。某些传统的实现可采用‘_’下划线，‘:’冒号；‘ ’空白；‘.’结束；音响设备应该认可这些分隔符字符。

TimeReference	此字段包含序列的时间码。它的长度是64比特，包括自午夜起计算的第一个抽样计数。每秒的抽样数取决于抽样频率，在下列字段中定义。来自 <fmt-ck>中的<nSamplesPerSec>。
版本	BWF的版本采用一个无符号二进制数表示。对于版本1，设置为0x0001。
UMID	包括由SMPTE 330M定义的扩展的UMID的64字节。如果采用32字节的基本UMID，最后32字节应用0填充。如果没有UMID，64字节应用0填充。 注 – UMID的长度在UMID的开头编码。
保留	预留190个字节用于扩展。这190个字节应设置为0。
uCoding History	包括0或多个串的UTF-8字符的大小可变块，每个均以<CR><LF>结束。第一个未使用字节必须是空字符（0x00）。 每个串必须包括对适用于音频数据的编码过程的描述。每个新的编码应用必须增加一个具有适当信息的新串。 历史信息的编码的标准格式详见附件1的附录2。 此信息必须包括具有其特定参数的声音类型（PCM 或 MPEG）： PCM： 模式（单声道、立体声）、抽样大小（8，16bits）和抽样频率， MPEG： 抽样频率、比特率、层（I或II）以及模式（单声道、立体声、联合立体声或双通道），

注 1 – 除uDescription、uOriginator、uOriginatorReference和uCodingHistory 之外的所有各项的内容必须与bext块的相应各项的内容相同，见第2.3节。

注 2 – 当UTF-8中给定的代码不属于处理设备支持的子集时（如ISO/IEC 10646：2003第12章定义），该值不得改变并在处理时忽略。

附件1的附录1 (资料性)

RIFF WAVE (.WAV) 文件格式

本附录的信息取自微软® RIFF文件格式说明文件。只作为信息性参考。

1 波形音频文件格式 (WAVE)

WAVE形式定义如下。节目必须预计（并忽略）所有相遇的具有全部RIFF格式的未知块。但是，<fmt-ck>通常必须在<wave-data>前出现，这些块中的双方在WAVE文件中是必备的。

```
<WAVE-form> ->
    RIFF ( 'WAVE'
        <fmt-ck>           // 格式块
        [<fact-ck>]       // 实际块
        [<other-ck>]      // 其他任选块
        <wave-data> )     // 声音数据
```

WAVE块在下列各节中描述：

1.1 WAVE 格式块

WAVE 格式块 <fmt-ck> 规定<wave-data>的格式。<fmt-ck> 定义如下：

```
<fmt-ck> ->fmt ( <common-fields>
    <format-specific-fields> )
<common-fields> ->
    struct{
        WORD wFormatTag,           /* 格式类别 */
        WORD nChannels,           /* 通路数 */
        DWORD nSamplesPerSec,     /* 抽样率 */
        DWORD nAvgBytesPerSec,    /* 缓冲评估*/
        WORD nBlockAlign,         /* 数据块大小*/
    }
```

块的<common-fields>部分中的字段描述如下：

字段	描述
wFormatTag	指示文件的WAVE格式类别的一个号码。<fmt-ck>的<format-specific-fields> 部分的内容和波形数据的解释取决于该值。
nchannels	在波形数据中代表通路数，如1代表单声道或2代表立体声。
nSamplesPerSec	抽样率（在每秒抽样中），在此对每个通路复制。

nAvgBytesPerSec 每秒平均字节数，在此发送波形数据。播放软件可以使用此值估算缓冲区大小。

nBlockAlign 波形数据的块校准（字节）。播放软件每次需要处理多个数据的 $\langle nBlockAlign \rangle$ 字节，因此 $\langle nBlockAlign \rangle$ 用于缓冲校准。

$\langle \text{format-specific-fields} \rangle$ 由0或多个参数的字节组成。哪个参数出现取决于WAVE格式类别，详细描述参见以下章节。播放软件应允许（并忽略）在此字段结束时出现的任何 $\langle \text{format-specific-fields} \rangle$ 参数。

1.2 WAVE 格式种类

WAVE文件的格式类别由‘fmt’块的 $\langle wFormatTag \rangle$ 字段的内容规定。 $\langle \text{wave-data} \rangle$ 中的数据表示以及‘fmt’块 $\langle \text{format-specific-fields} \rangle$ 的内容取决于格式类别。

当前定义的开放非专有 WAVE 格式种类如下：

wFormatTag	值	格式类别
WAVE_FORMAT_PCM	(0x0001)	微软® (PCM) 格式
WAVE_FORMAT_MPEG	(0x0050)	MPEG-1 音频 (只有音频)

注 – 虽然其他WAVE 格式注册了微软®, 目前只有上述格式采用BWF。PCM WAVE 格式的详细描述在下述的第2部分中。其他WAVE 格式的通用信息在第3节中描述。MPEG WAVE 格式的详细描述参见附件 2。其他WAVE 格式可能在今后规定。

2 PCM 格式

如果 $\langle \text{fmt-ck} \rangle$ 的 $\langle wFormatTag \rangle$ 字段设置为 WAVE_FORMAT_PCM，波形数据由 PCM 格式表示的抽样组成。对于 PCM 波形数据， $\langle \text{format-specific-fields} \rangle$ 定义如下：

```

<PCM-format-specific> ->
    struct{
        WORD nBitsPerSample,    /*抽样大小*/
    }

```

$\langle nBitsPerSample \rangle$ 字段规定：用于表示每通路的每个抽样所用的数据比特数。如果有多个通路，每个通路的抽样大小是相同的。

$\langle nBlockAlign \rangle$ 字段等于下列公式，四舍五入到下一个整数

$$nchannels \times BytesPerSample$$

BytesPerSample 的值的计算采用将 nBitsPerSample 四舍五入到下一个整数字节。当音频抽样字小于字节的整数时，音频抽样的最高有效位设置于数据字的最高有效位上，最低有效位相邻的未使用的数据应被设置为 0。

对于 PCM 数据，‘fmt’块的 $\langle nAvgBytesPerSec \rangle$ 字段等于下列公式。

$$nSamplesPerSec \times nBlockAlign$$

注 1 – 最初的WAVE规范允许2个通路的最低有效位的单个字节，例如 2个通路的 20比特抽样打包到5个共享字节。本建议书规定每个音频抽样的整数字节，以减少具体实施中的模糊性并实现最大限度的互换兼容性。

2.1 PCM WAVE文件的数据打包

在单通路的WAVE文件中，抽样是连续存储的。对于立体声WAVE文件，通路 0 表示左边通路，通路1表示右边通路。多通路WAVE文件中，抽样是隔行交织的。

下表表示8比特单声道和立体声WAVE文件的数据打包：

8比特单声道 PCM的数据打包

抽样 1	抽样 2	抽样 3	抽样 4
通路0	通路 0	通路 0	通路 0

8比特立体声 PCM的数据打包

抽样 1		抽样 2	
通路 0 (左)	通路 1 (右)	通路 0 (左)	通路 1 (右)

下表表示16比特单声道和立体声WAVE文件的数据打包：

16比特单声道 PCM的数据打包

抽样 1		抽样 2	
通路 0 低阶字节	通路 0 高阶字节	通路 0 低阶字节	通路 0 高阶字节

16比特立体声 PCM的数据打包

抽样 1			
通路 0 (左)	通路 0 (左)	通路 1 (右)	通路 1 (右)
低阶字节	高阶字节	低阶字节	高阶字节

2.2 抽样的数据格式

每个抽样包含在整数*i*中。*i*的大小是包含特定抽样大小所需要的最小的字节数。最低有效字节最先存储。表示抽样幅度的比特存储在 *i* 的最高有效位。其余比特设置为0。

例如，如果抽样大小（<nBitsPerSample>中记录）是12比特，则在2字节整数中存储每个抽样。第1个（最低有效）字节的最低有效4比特置为0。各种大小的PCM波形抽样的数据格式和最大和最小值如下：

抽样大小	数据格式	最大值	最小值
1到8比特	无符号整数	255 (0xFF)	0
9或更多bits	有符号整数i	i的最大正值	i的最大负值

例如，8比特和16比特PCM波形数据的最大、最小和中间值如下：

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

2.3 PCM WAVE文件的范例

PCM WAVE文件的范例，11.025 kHz抽样率、单声道、8比特每抽样：

```
RIFF ('WAVE' fmt (1, 1, 11025, 11025, 1, 8)
      data (<wave-data>))
```

PCM WAVE文件的范例，22.05 kHz 抽样率、立体声、8比特每抽样：

```
RIFF ('WAVE' fmt (1, 2, 22050, 44100, 2, 8)
      data (<wave-data>))
```

PCM WAVE文件范例，44.1 kHz抽样率、单声道、20比特每抽样：

```
RIFF ('WAVE' INFO (INAM ("O Canada" Z))
      fmt (1, 1, 44100, 132300, 3, 20)
      data (<wave-data>))
```

2.4 WAVE数据的存储

<wave-data>包括波形数据定义如下：

```
<wave-data> -> { <data-ck> }
<data-ck> -> data (<wave-data>)
```

2.5 实际块

<fact-ck> 实际块存储关于WAVE文件内容的重要信息。此块定义如下：

```
<fact-ck> -> fact (<dwFileSize:DWORD>) /*抽样数*/
```

此块不是PCM文件所要求的。

实际块将被扩展以包括所有未来WAVE格式所要求的其他信息。增加的字段将出现在<dwFileSize>字段之后。具体应用使用块大小字段以确定所出现的字段。

2.6 其他任选块

规定其他块使用WAVE 格式。这些块的详细说明在WAVE格式规范中给出并随时发布更新。

注 1 – WAVE 格式可支持WAVE文件中包括的承载特定信息的其他任选块。如附件1的第2.1节的注1所述，在广播波格式文件中，这些被认为是专用块并被不能对其进行识别的应用所忽略。

3 其他WAVE类型

下列信息源于微软®数据标准。它概述了包含其他类型WAVE格式的基本WAVE文件（用于PCM音频）的扩展。

3.1 一般信息

所有新定义的WAVE类型必须包括一个 <fact-ck> 和一个扩展的波格式描述（在 <fmt-ck> 格式块中）。类型WAVE_FORMAT_PCM 的RIFF WAVE文件不需要额外块也不需要扩展的波格式描述。

3.2 实际块

此块存储有关WAVE文件内容的文件相关信息。通常在抽样中规定文件的长度。

WAVE 格式扩展

扩展的波格式结构加入<fmt-ck>，用于定义所有非PCM 格式波数据，并描述如下。一般扩展的波形格式结构用于所有非PCM 格式。

```
typedef struct waveformat_extended_tag {
    WORD    wFormatTag,           /* 格式类型 */
    WORD    nChannels,           /* 通路数（即单声道、立体声...） */
    DWORD   nSamplesPerSec,      /* 抽样率 */
    DWORD   nAvgBytesPerSec,     /* 缓冲评估 */
    WORD    nBlockAlign,         /* 数据块大小 */
    WORD    wBitsPerSample,      /* 单声道数据的每抽样比特数 */
    WORD    cbSize,              /* 额外部分的字节计算 */
} WAVEFORMATEX;
```

字段	描述
wFormatTag	定义WAVE文件类型。
nChannels	波中的通路数，1代表单声道，2代表立体声。
nSamplesPerSec	波文件抽样率的频率。应该为48000或44100等。它也在实际块中用于抽样大小入口以确定数据的持续时间。
nAvgBytesPerSec	平均数据率。播放软件可以采用<nAvgBytesPerSec> 值估计缓冲大小。
nBlockAlign	<data-ck>中数据的块校准（字节）。播放软件需要同时处理多个

数据的<nBlockAlign>字节，因此<nBlockAlign>值可用于缓冲校准。

wBitsPerSample	这是每抽样每通路的比特数。假定每通路具有相同的抽样解决方案。如果不需要此字段，则应设置为0。
cbSize	WAVE 格式标题中额外信息的字节大小，不包括WAVEFORMATEX结构的大小。

注 – <cbSize> 字段之后的字段，包括 <wFormatTag>中定义WAVE 格式所需要的特定信息。所有可以用于BWF 的WAVE 格式将分别规定作为对本建议书的补充。

附件1的附录2 (资料性)

<CodingHistory>字段的格式规范

引言

<bext>块中的<CodingHistory> 字段定义为包括编码过程历史的串的集合。每当编码历史发生变化时应增加新的行。每行应包括编码的每个参数的串变量。每行应由CR/LF结束。编码历史串的格式描述如下。

语法

每行的语法描述如下：

参 数	可变串 <允许的选项>
编码算法	A=<ANALOGUE, PCM, MPEG1L1, MPEG1L2, MPEG1L3, MPEG2L1, MPEG2L2, MPEG2L3>
抽样频率 (Hz)	F=<16000, 22050, 24000, 32000, 44100, 48000>
比特率 (kbit/s每通路)	B=< MPEG 2 (ISO/IEC 13818-3) 是允许的所有比特率>
字长度 (bits)	W=<8, 12, 14, 16, 18, 20, 22, 24>
模式	M=<单声道、立体声、双单声道、联合立体声>
文本、空闲串	T=<室内使用的空闲ASCII文本串。此串应不包括逗号 (ASCII 2Chex)。内容的范例：ID-No; 编解码器类型; A/D类型>

可变串应用逗号 (ASCII 2Chex) 隔开。每行由CR/LF终止。

可变 B= 只用于MPEG编码。

可变 W=对于MPEG编码，应用于指示到MPEG编码器的PCM输入的字长。

编码历史字段的范例

范例 1

A=PCM, F=48000, W=16, M=立体声, T=最初的, CR/LF

A=MPEG1L2, F=48000, B=192, W=16, M=立体声, T=PCX9, CR/LF

范例1的解释

系列1

最初的文件记录为采用PCM编码的线性BWF文件:

- 抽样频率: 48 kHz
- 编码分辨率: 16比特每抽样
- 模式: 立体声
- 状态: 最初的编码

系列2

最初的文件已经转换为MPEG-1 层 II BWF文件, 采用下列参数:

- 抽样频率: 48 kHz
- 每秒每通路比特: 192 kbit/s
- 编码分辨率: 16比特
- 模式: 立体声
- 编码器: PCX9 (Digigram)

范例2用于模拟材料的数字化过程

A=ANALOGUE, M=立体声, T=StuderA816; SN1007; 38; Agfa_PER528, <CR/LF>

A=PCM, F=48000, W=18, M=立体声, T=NVision; NV1000; A/D, <CR/LF>

A=PCM, F=48000, W=16, M=立体声, T=PCX9; DIO, <CR/LF>

范例2的解释

系列1

模拟磁带, 类型Agfa PER528, 通过录音机播放, Studer模式A816, 序列号1007:

- 磁带速率: 38 cm/s
- 模式: 立体声

系列2

数字化录音采用A/D转换类型NVision NV1000:

- 抽样频率: 48 kHz
- 编码分辨率: 18比特每抽样
- 模式: 立体声

系列3

录音存储为BWF文件, 线性PCM编码采用PCX9接口卡的数字输入:

- 抽样频率: 48 kHz
- 编码分辨率: 16比特每抽样
- 模式: 立体声

附件1的附录3 (资料性)

<OriginatorReference>字段中使用的“唯一的” 源标识符 (USID) 的格式定义

USID

<OriginatorReference>中的USID的产生采用几个独立的随机源，以保证其在缺少单个分配授权情况下的唯一性。一种有效和容易利用随机性的方法是结合用户、机器和时间特定信息加上一个随机数。这些元素是：

CC 国家代码：（2字符）基于ISO 3166⁵ 标准[ISO, 1997]。

OOOO 组织机构代码：4字符。

NNNNNNNNNNNN 序号：（录音模式和序号中提取的12字符），确定机器的类型和序号。

HHMMSS OriginationTime: 来自BWF的<OriginationTime>字段的（6字符）。

结合其他正式和非正式的信息源，这些元素应足以确定“人类使用”形式中特定的录音。

另外，USID包括：

RRRRRRRR 采用适当的随机算法的录音机局部产生随机数（8字符）。

此元素有助于分别识别的文件（例如）立体声通路或多轨录音中的磁轨（在同时产生时）。

USID的范例

范例1

USID由TascamDA88，S/N396FG347A生成，由RAI，无线电视台意大利运营，时间：12:53:24

UDI格式：CCOOOONNNNNNNNNNNNNHHMMSSRRRRRRRR

UDI范例：ITRAI0DA88396FG34712532498748726

范例2

USID由xxxxxx，S/N ssssssss生成，由YLE，芬兰广播公司运营，时间：08:14:48

UDI格式：CCOOOONNNNNNNNNNNNNHHMMSSRRRRRRRR

UDI范例：FIYLE0xxxxxxsssssss08144887724864

⁵ 表示国家名称及其分支机构的ISO 3166-1:1997代码 – 第1部分：国家代码
(参见: <http://www.din.de/gremien/nas/nabd/iso3166ma/index.html>)

附件1的附录4 (资料性)

对BWF任选的峰值包络电平块 <levl -ck> 的定义

当音频文件在工作站之间交换时，如果可以得到文件中关于峰值音频信号电平的数据，就可以加快文件的打开、显示和处理。将<levl>块附加到广播波格式（BWF）文件[1]上，为存储并转发通过音频子抽样得到的信号峰值数据提供了一个标准。此块中的数据可以用于提供文件中音频要素的包络。这也允许在不太降低正确率的情况下快速显示音频文件的应用。

另外，还可以发送各峰值的峰值，即第一个音频抽样，其绝对值是整个音频文件的最大值。音频应用可利用此信息实时规格化文件而无需对整个文件进行扫描文件（因为发送方已经执行了此过程）。

1 术语

将音频信号分为若干块。每个音频块产生一个**峰帧**。每个峰帧有n个**峰值**，其中n是峰通路数。每个峰值可以由一个（仅为正）或两个（一正一负）**峰点**组成。

1.1 峰值的产生

音频信号被分为恒定大小的抽样块。为每个通路建议的块大小的默认值为256个抽样。

对每个通路的抽样进行评估以找到峰点（最大值）。建议分别从正和负的抽样寻找峰点，但只可采用绝对值（正或负）。所有的峰点均是无符号值。

峰点被舍入到两种格式之一，8或16比特。通常情况下，8比特格式就足够了。16比特格式应覆盖需要较高精度的所有情况。

每个通路的格式化的峰值组合到峰帧中。对于每个与音频抽样次序相同的通路，每个峰帧由正和负的峰点（或绝对峰点）组成。

这些峰帧作为峰包络块中的数据被承载。每个峰包络块以一个标题开始，其中包括允许对峰数据进行解释的信息。

各峰值的峰值是第一个音频抽样，其绝对值是整个音频文件的最大值。对各峰值的峰值的正值进行存储而非将各峰值的峰值存储为抽样值。换言之，存储音频抽样帧索引。因此一个应用就可以得知在哪可以读到音频文件中各峰值的峰值。存储一个峰值可能更困难一些，因为这取决于音频抽样的二进制格式（整数、浮动、双倍……）。

注：

- 标题只使用DWORD（4字节值）或4字节的倍数，以避免在不同编辑器中校准结构的问题。
- 标题共有128字节以避免缓存错位。

2 峰包络块

峰包络、<levl>、块由一个标题接着峰点数据组成。块的总长度是可变的，取决于音频内容、块大小和如何对峰数据进行格式化。

```
typedef struct peak_envelope
{
    CHAR        ckID[4],           /* {'l', 'e', 'v', 'l'} */
    DWORD       ckSize,           /* 块大小 */
    DWORD       dwVersion,        /* 版本信息 */
    DWORD       dwFormat, ;       /* 峰点格式 */
                                   1 = 无符号字符
                                   2 = 无符号短码
    DWORD       dwPointsPerValue, /* 1 = 只有正的峰值点
                                   2 = 正和负的峰值点 */
    DWORD       dwBlockSize,      /* 每值的帧 */
    DWORD       dwPeakChannels,   /* 通路数 */
    DWORD       dwNumPeakFrames, /* 峰帧数 */
    DWORD       dwPosPeakOfPeaks, /* 音频抽样帧索引/* 或
    0xFFFFFFFF 如未知*/
    DWORD       dwOffsetToPeaks,  /* 通常应等于此标题的大小，但也可以大于 */
    CHAR        strTimestamp[28], /* ASCII：峰值数据的时间标记 */
};
```

2.1 “levl”块的单元

ckID 这是4字符阵列{“l”，“e”，“v”，“l”}⁶ 块识别。

ckSize 剩余的块的大小。（不包括ckID和ckSize中使用的8字节）

dwVersion peak_envelope块的版本。它以0000开始。

dwFormat 峰包络数据的格式。两种格式描述如下⁷：

dwFormat	值	描述
LEVL_FORMAT_UINT8	1	每个峰点的无符号字符
LEVL_FORMAT_UINT16	2	每个峰点的无符号短码整数

dwPointsPerValue 这表示每个峰值的峰点数。可以为1或者2。

⁶ 定义DWORD ckID = “levl” 通常不是唯一的。不同的 C-编辑器产生不同的字符顺序。因此我们定义字符ckID[4] = {“l”，“e”，“v”，“l”}。

⁷ 因为所有支持“levl”块的音频应用必须实现所有可能的格式，只允许两种格式。在多数情况下，无符号字符(8 bit) 格式就足够了。无符号短格式 (16 bit)应覆盖所有需要更高精度的情况。

dwPointsPerValue = 1

每个峰值由一个峰点组成。在每个区中峰点是**dwBlockSize**音频抽样的最大绝对值：

$$\max\{\text{abs}(X_1), \dots, \text{abs}(X_n)\}$$

注 – 在此情况下，所表现的波形通常相对水平轴是对称的。

dwPointsPerValue = 2

每个峰值由2个峰点组成。第一个峰点对应区中**dwBlockSize**音频抽样的最高正值。第二个峰点对应区中**dwBlockSize**音频抽样的负峰值。

建议采用2个峰点（**dwPointsPerValue = 2**），因为将显示不对称波形（例如DC偏移）。

dwBlockSize 这是用于产生每个峰帧的音频抽样数。此数是可变的。默认和建议的区大小为256。

dwPeakChannels 峰通路数⁸。

dwNumPeakFrames 峰帧数。峰帧数为通过下列公式计算并四舍五入得出的整数：

$$\text{dwNumPeakFrames} = \frac{(\text{numAudioFrame} + \text{dwBlockSize})}{\text{dwBlockSize}}$$

或：

$$\text{dwNumPeakFrames} = \frac{\text{numAudioFrame}}{\text{dwBlockSize}}$$

其中 **numAudioFrame** 是音频数据每通路中的音频抽样数。

例如，对于峰比值（区大小）256，表示：

0	音频抽样	-> 0 峰帧
1	音频抽样	-> 1 峰帧
256	音频抽样	-> 1 峰帧
257	音频抽样	-> 2 峰帧
7582	音频抽样	-> 30 峰帧。

dwPosPeakOfPeaks 音频应用可以采用此信息对文件进行规格化而不需要扫描整个文件。（因为此过程已经通过发送方执行）。其益处在于性能的提升以及可能实时对文件进行规格化。

各峰值的峰值为第一个音频抽样，其绝对值是整个音频文件的最大值。

存储各峰的正峰值，而非将存储各峰值的峰值作为抽样值。换言之，存储音频抽样帧索引。因此一个应用就可以得知在哪可以读到音频文件中的各峰值。存储峰值可能更加困难因为这取决于音频抽样的二进制格式（整数、浮动、双倍）。

⁸ 通常峰通路数等于音频通路数。如果该数值为1，则每个音频通路所显示的波形是一样的。

如果值为0xFFFFFFFF，则表示各峰值未知。

dwOffsetToPeaks 峰值数据从标题开始偏移。这通常等于标题的大小，也可能更高些。用此值保证峰值数据在DWORD边界上开始。

strTimeStamp 包括创建峰值数据的时间标记的串。格式化如下：⁹

“YYYY : MM : DD : hh : mm : ss : uuu”

其中：

YYYY: 年
MM: 月
DD: 日
hh: 小时
mm: 分钟
ss: 秒
uuu: 毫秒

范例：“2000: 08: 24: 13: 55: 40: 967”

2.2 峰点的格式

峰值由1或2个峰点组成，标记为**dwPointsPerValue**。标记**dwFormat**指示代表每个峰帧中的峰点数的格式。

		dwPointsPerValue	
		= 1	= 2
dwFormat		数字相当于绝对峰值	第一个数字相当于正的峰值 第二个数字相当于负的峰值 (注“负”峰值存储为“正”数)
= 1	levl_format_uint8	无符号字符 (0...255)	无符号字符 (0...255) 无符号字符 (0...255)
= 2	levl_format_uint16	无符号短码 (0...65535)	无符号短码 (0...65535) 无符号短码 (0...65535)

2.3 多通路峰值文件

对于多通路音频文件，每个通路的单个峰值是交叉的。一组交叉的峰值称为一个峰帧。峰帧中峰值的次序相当于RIFF音频数据帧内抽样点的安排。

⁹ 此格式的优势在于不再有时间限制并可以方便读取。（其他格式使用一个DWORD，表示自从1970以来的秒，约125年后达到其限值。）

2.4 与音频文件同步

如果下列2个条件中的任一个满足，则峰文件必须重建：

时间标记比音频文件的时间标记老。

峰帧数不对应于音频文件中的抽样帧数。

2.5 字节顺序

由于广播波格式文件（BWF）是RIFF 格式的扩展，所有数字都存储为短字节。

附件1的附录5 （资料性）

对BWF任选的链接块 <link-ck>的定义

引言

广播波格式（BWF）文件允许的最大文件大小为4千兆字节，虽然实际上许多RIFF/波应用只支持最大文件为2千兆字节。对于超过此限制的音频数据，必须将音频信息分为多个BWF文件。<link>块为几个文件上传的无缝音频输出提供链接数据。

1 术语

文件集	一组属于一个连续的音频信号的链接文件。
文件名	为文件集中的每个文件给出名称。
文件清单	文件集中文件名的清单
“实际”属性	在文件清单中为当前（或“实际”）文件标出文件名的属性。文件清单中所有其他文件名标为“其他”。
文件标识符	一个任选标识符，它对于文件集的所有文件都应该是相同的。
“专用”单元	块中一个附加单元，存储文件清单的专有信息。
<link>块	文件集所有文件中包括的块。它包括一个标题，接着一个文件清单并任选地包括一个文件标识符和“专用”单元。块中的数据以XML 1.0 格式 ¹⁰ 存储，即用于数据交换的普遍格式。

¹⁰ Extensible Markup Language (XML) 1.0 W3C 建议书 10-February-1998 <http://www.w3.org/TR/1998/REC-xml-19980210>.

2 链接块结构

2.1 概要

<link>块的组成包括一个标题，接着以XML（eXtensible Markup Language）格式存储的链接信息。块的总长度是可变的。

```
typedef struct link
{CHAR   CkID[4],      /* {'l', 'i', 'n', 'k'} */
      DWORD   CkSize,    /* 块大小 */
      CHAR   XmlData[], /* XML中的链接信息 */
}
Link_chunk,
```

字段 **DescriptionckID** 这是用于块识别的4字符阵列 { 'l', 'i', 'n', 'k' }¹¹。

CkSize 块的数据段的长度（不包括ckID和ckSize使用的8字节）

XmlData 此缓冲区包括XML（ASCII字符）中的链接信息。

2.2 <xmlData>可变数据字段中的XML数据结构

数据结构是分级的。数据存储在文本串中。对于严格的语法规则，还要增加一个DTD（数据转换文件）。

```
<LINK>
      <FILE type="...">
          <FILENUMBER>...</FILENUMBER>
          <FILENAME>...</FILENAME>
      </FILE>
      .....
      Possible further FILE elements
      .....
      <ID>...</ID>    optional
      <PRIVATE>      optional
      .....
      ..... implementation dependent
      </PRIVATE>
</LINK>
```

LINK 这是XML数据的根元素。LINK包括一个或多个采用文件描述的FILE元素。它也可以包括标识符ID和/或一个PRIVATE元素。

ID 标识符ID对给定的文件集的所有文件是共同的。它以XML 1.0规范的#PCDATA定义所允许的字符的文本串方式存储，它包括所有的ASCII字符、空白等。

PRIVATE PRIVATE元素可能包括由任何XML数据（如更多的元素或#PCDATA）组成的与实现相关的信息。

FILE FILE可以包括FILENUMBER元素和FILENAME元素。在清单中的文件描述

¹¹ DWORD ckID = “链接”的定义不应该是唯一的。不同的C-编辑器产生不同的字符顺序。因此规定字符 ckID[4] = { 'l', 'i', 'n', 'k' }。

该块所属于的文件时，类型属性应为“实际”。所有其他文件应具有的类型属性为“其他”。该文件的文件名应与文件清单中出现的名称相同。

FILENUMBER 文件应该按照其在文件集中的年代顺序编号。整数（ASCII字符）应从1开始。

FILENAME 文本串存储的格式与ID相同。

2.3 <link>块XML结构的DTD

DTD（文件类型定义）在XML 1.0规范中描述，作为XML结构的语法定义。<link>块的不同元素的格式和属性描述如下，包括子元素和其多样性。

元素LINK应包括一个或多个子元素FILE（‘+’指示一个或多个），它也包括一个子元素ID和一个子元素PRIVATE（‘?’指示一个或没有）。

每个元素FILE应包括一个子元素FILENUMBER和一个子元素FILENAME。应规定其类型属性，可以为“实际”或“其他”。

子元素FILENUMBER，FILENAME和ID必须包括文本串（XML中称为#PCDATA）。

子元素PRIVATE可以包括任何定义的元素。如果PRIVATE需要包括已经定义的之外的元素，DTD必须做相应的修改。

```

<!ELEMENT LINK          (FILE+, ID?, PRIVATE?) >
<!ELEMENT FILE          (FILENUMBER, FILENAME) >
<!ATTLIST FILE          type ("actual" | "other") #REQUIRED>
<!ELEMENT FILE          NUMBER (#PCDATA) >
<!ELEMENT FILE          NAME (#PCDATA) >
<!ELEMENT ID            (#PCDATA) >
<!ELEMENT PRIVATE      ANY>

```

3 链接的文件的更名

如果有一个或多个文件名变化了，属于整个文件集的<link>块中相应的FILENAME条目也应修改。

一个此范例中连续的声音信号已经分成由3个BWF文件组成的文件集，所谓“Sinatra_1.wav”，“Sinatra_2.wav”和“Sinatra_3.wav”。这3个文件的<link>块的XML结构除类型属性外是相同的。

3.1 <link> chunk of “Sinatra_1.wav”

```

<LINK>
  <FILE type="actual">
    <FILENUMBER>1</FILENUMBER>
    <FILENAME>Sinatra_1.wav</FILENAME>
  </FILE>
  <FILE type="other">
    <FILENUMBER>2</FILENUMBER>
    <FILENAME>Sinatra_2.wav</FILENAME>
  </FILE>
  <FILE type="other">
    <FILENUMBER>3</FILENUMBER>
    <FILENAME>Sinatra_3.wav</FILENAME>
  </FILE>
</ID>73365869</ID>
</LINK>

```

3.2 <link> chunk of “Sinatra_2.wav”

```

<LINK>
  <FILE type="other">
    <FILENUMBER>1</FILENUMBER>
    <FILENAME>Sinatra_1.wav</FILENAME>
  </FILE>
  <FILE type="actual">
    <FILENUMBER>2</FILENUMBER>
    <FILENAME>Sinatra_2.wav</FILENAME>
  </FILE>
  <FILE type="other">
    <FILENUMBER>3</FILENUMBER>
    <FILENAME>Sinatra_3.wav</FILENAME>
  </FILE>
</LINK>
  <ID>73365869</ID>

```

3.3 <link> chunk of “Sinatra_3.wav”

```

<LINK>
  <FILE type="other">
    <FILENUMBER>1</FILENUMBER>
    <FILENAME>Sinatra_1.wav</FILENAME>
  </FILE>
  <FILE type="other">
    <FILENUMBER>2</FILENUMBER>
    <FILENAME>Sinatra_2.wav</FILENAME>
  </FILE>
  <FILE type="actual">
    <FILENUMBER>3</FILENUMBER>
    <FILENAME>Sinatra_3.wav</FILENAME>
  </FILE>
</LINK>
  <ID>73365869</ID>

```

附件1的附录6 (标准的)

文件名约定

1 概述

音频文件的一般互换意味着：可以在与起始系统非常不同的计算机和操作系统类型上播放。不适当的文件名可能意味着：该文件不能被目的地系统识别。例如，某些操作系统限制文件名中的字符数。而另一些则不能容许多字节字符。某些字符在某些操作系统中有特别的意义并应该避免。这些导则都旨在为一般国际互换确定最佳的实施方案。

2 文件名长度

BWF文件名不应超过31字符，包括文件名扩展。

3 文件名扩展

BWF 文件必须采用相同的4字符文件名扩展。“**.wav**”，作为常规的**WAVE**文件。这允许音频内容不采用附加软件就可以在大部分计算机上播放。实际的执行过程也应接受其他扩展，如“**.bwf**”，可能已经在差错中采用。

4 文件名字符集

用于国际交换的文件名只应采用**ASCII**（**ISO/IEC646**）7比特字符，范围为32至126（十进制）。

字符	十进制值	十六进制值
(空白)	32	0x20
...
~ (tilda)	126	0x7E

此外，下面的字符是在某些文件系统中保留用于特殊功能，并且不应在文件名中使用：

字符	十进制值	十六进制值
“	34	0x22
*	42	0x2A
/	47	0x2F
:	58	0x3A
<	60	0x3C
>	62	0x3E
?	63	0x3F
\	92	0x5C
	124	0x7C

此外，下列字符不应用于文件名的第一或最后一个字符：

字符	十进制值	十六进制值
(空白)	32	0x20
(周期)	46	0x2E

附件2

承载 MPEG-1 音频的广播波格式规范

广播中音频数据文件的格式

1 引言

本附件包括的规范用于BWF以承载只有MPEG音频的信号。对于MPEG音频，必须在本建议书的主要部分中规定的基本块中增加下列信息：

- 格式块的扩展；
- 实际块；
- MPEG_extension 块。

格式块的扩展和实际块规定作为WAVE 格式的一部分，并且相关的信息在附件2附录1中给出。

MPEG_extension块的规范在附件2的第2节中给出。

本建议书的主要部分包括用于所有BWF的广播音频扩展块的规范。基本RIFF格式的信息在附件 2的附录1中给出。

2 MPEG 音频

微软®已经详细规定了在WAVE文件中如何组织MPEG音频数据。格式块的扩展和实际块承载规定MPEG编码选项所需要的更多信息。一般原理在附件1的附录1中描述，详细的规定参见附件2的附录1。对于MPEG层II，已经发现还需要承载信号编码的额外信息。这在<MPEG 音频扩展>块中承载，并由MPEG层2音频相关群发展。此块规定如下。

2.1 MPEG 音频扩展块

MPEG 音频扩展块定义如下：

```
typedef struct {
    DWORD    ckID,           /* (mpeg_extension) ckID='mext' */
    DWORD    ckSize,        /* size of extension chunk :
                             cksize =000C*/
    BYTE    ckData[ckSize], /* data of the chunk */
}

typedef struct mpeg_audio_extension {
    WORD SoundInformtion,   /* more information about sound */
    WORD FrameSize,        /* nominal size of a frame */
    WORD AncillaryDataLength, /* Ancillary data length */
    WORD AncillaryDataDef,  /* Type of ancillary data */
    CHAR Reserved [4],     /* "NULL" */
} MPEG_EXT ;
```

Champ	描述
SoundInformation	<p>16比特给定关于声音文件的附加信息：</p> <p>对于 MPEG 层 II（或层 I）：</p> <p>Bit 0: ‘1’ 均匀声音数据 ‘0’ 非均匀声音数据</p> <p>Bits 1 和 2 用于均匀声音文件的附加信息：</p> <p>Bit 1: ‘0’ 填充位用于文件中，可以在‘0’或‘1’之间交替 ‘1’ 填充位在整个文件中，设置为 ‘0’</p> <p>Bit 2: ‘1’ 文件包括一个帧序列，填充位设置为 ‘0’ 和抽样频率等于 22.05或44.1 kHz</p> <p>注 1 – 此文件不符合MPEG 标准（2.4.2.3节中padding_bit的定义），但是可以视为可变比特率的一种特殊情况。不需要 MPEG 解码器对此比特流进行解码，因为大多数解码器将执行此功能。比特率将略低于标题的指示。</p> <p>Bit 3: ‘1’ 使用空闲格式 ‘0’ 非空闲格式音频帧。</p>
FrameSize	<p>标称帧的16比特字节数。</p> <p>此字段只对均匀文件有意义，否则它置为 ‘0’。</p> <p>如果不使用填充位，即它在声音文件的所有帧中保持恒定，字段<FrameSize> 包括的值与格式块中的字段<nBlockAlign> 相同。如果采用填充位并且在声音数据中出现可变长度，<FrameSize> 包括帧的大小，填充位设置为 ‘0’。填充位设置为 ‘1’ 的帧的长度为1或多个字节（4字节用于层I），即<FrameSize+1>。</p> <p>事实上<nBlockAlign> 设置为 ‘1’ 表示可变帧长度（FrameSize 或 FrameSize+1）具有可变填充位。</p>
AncillaryDataLength	<p>16比特数字，给出在全部声音文件中辅助数据的已知字节的最小数目。此值与音频帧结束相关。</p>
AncillaryDataDef	<p>16比特值规定了辅助数据的内容：</p> <p>Bit 0 设置为 ‘1’: 辅助数据中左通路的能力</p> <p>Bit 1 设置为 ‘1’: 一个专用字节，为辅助数据内部使用空闲</p> <p>Bit 2 设置为 ‘1’: 辅助数据中右通路的能力</p> <p>Bit 3 设置为 ‘0’: 保留为ADR 数据将来使用</p> <p>Bit 4 设置为 ‘0’: 保留为DAB 数据将来使用</p> <p>Bit 5 设置为 ‘0’: 保留为J 52 数据将来使用</p> <p>Bit 6至 15 设置为 ‘0’: 保留为将来使用</p>

注:

- 辅助数据中出现的各项按时与AncillaryDataDef中的比特数相同的次序。第1项存储在辅助数据结尾，第2项在第1项之前，从后向前排列。
- 对于单声道文件，比特2通常设置为‘0’，比特0涉及单声道帧的能力。
- 对于立体声文件，如果比特2等于‘0’和比特0等于‘1’，能力涉及左和右的最大能力。
- 能力存储在2字节中并相当于编码帧所采用的最大抽样的绝对值。在Big Endian格式中这是15比特值。

保留 4字节保留做将来使用。这些4字节必须设置为空。在任何未来的应用中，空值将用于默认值以保证兼容性。

附件2的附录1 (资料性)

RIFF WAVE (.WAV) 文件格式

本附录给出的规范针对包括MPEG音频的WAVE文件必须的额外信息。

本附录中的信息源于微软®RIFF文件格式的规范文件。它只用于资料性信息。

1 MPEG-1 音频 (只有音频)

1.1 实际块

此块为所有WAVE_FORMAT_PCM之外的WAVE格式所必须。它存储关于WAVE数据内容的文件相关信息。它规定抽样中数据的时间长度。

注1 - 也参见附件1的附录1，第2.5节。

1.2 WAVE 格式标题

```
#define WAVE_FORMAT_MPEG (0x0050)

typedef struct mpeg1waveformat_tag {
    WAVEFORMATEX          wfx ;
    WORD                  fwHeadLayer ;
    DWORD                 dwHeadBitrate ;
    WORD                  fwHeadMode ;
    WORD                  fwHeadModeExt ;
    WORD                  wHeadEmphasis ;
    WORD                  fwHeadFlags ;
    DWORD                 dwPTSLow ;
    DWORD                 dwPTSHigh ;
} MPEG1WAVEFORMAT ;
```

字段	描述
wFormatTag	此项必须设置为WAVE_FORMAT_MPEG。[0x0050]
nChannels	波中的通路数，1代表单声道、2代表立体声。
nSamplesPerSec	波文件的抽样频率（Hz）：32 000，44 100或48 000等。注意，如果数据的抽样频率可变，则此字段应设置为0。强烈建议，台式应用采用固定抽样频率。
nAvgBytesPerSec	平均数据速率；如果层 III下采用可变比特率编码，这可能不是一个法定的MPEG比特率。
nBlockAlign	<p><data-ck>中的数据块校准（字节），对于具有固定音频帧长度的音频流，块校准等于帧长度。对于其帧长度可变的流，<nBlockAlign>应设置为1。</p> <p>抽样频率为32或48 kHz时，MPEG音频帧的大小是比特率的函数。如果音频流采用恒定的比特率，音频帧的大小就不变。因此，下列公式适用：</p> <p>层 I: $nBlockAlign = 4 * (int) (12 * BitRate / SamplingFreq)$</p> <p>层 II 和 III: $nBlockAlign = (int) (144 * BitRate / SamplingFreq)$</p> <p>范例 1: 对于层I, 抽样频率为32 000 Hz以及比特率为256 kbit/s, $nBlockAlign = 384$ 字节。</p> <p>如果音频流包括的帧具有不同的比特率，则帧的长度在流内变化。当采用抽样频率为44.1 kHz时，可变的帧长度也出现：为维护数据速率在标称值上，MPEG音频帧的大小相比较上述给出的公式定时地增加一个“时隙”（层I中4字节，层 II和III中1字节）。在这两种情况下，块校准的概念是无效的。因此<nBlockAlign>的值必须设置为1，以便MPEG-可知应用可以描述是否数据为区域校准的。</p>
<p>注 – 通过将每个音频帧标题中的padding_bit设置为相同值（0或1），可能在44.1 kHz上构建具有恒定长度的音频帧。注意，结果流的比特率将不再精确地对应于帧标题中的标称值，因此某些解码器不能正确地对该流进行解码。出于标准化和兼容性方面的考虑，不推荐使用此方法。</p>	
WBitsPerSample	不使用；设置为0。
CbSize	WAVEFORMATEX 结构之后扩展信息的字节大小。对于标准WAVE_FORMAT_MPEG格式，这是22（0x0016）。如果增加了额外字段，此值将增加。
fwHeadLayer	<p>MPEG音频层，如下列标志所定义：</p> <p>ACM_MPEG_LAYER1 – 层I。</p> <p>ACM_MPEG_LAYER2 – 层II。</p> <p>ACM_MPEG_LAYER3 – 层III。</p>

某些法定的MPEG流可以包括不同层的帧。在此情况下，上述标志将OR，因此一个驱动程序可以确定哪层出现在流中。

dwHeadBitrate

数据的比特率，以每秒比特计。此值必须是符合MPEG规范的标准比特率；并非所有比特率对所有的模式和层有效。参见表1和2。注意，本字段记录了实际的比特率，非MPEG帧标题代码。如果比特率可变，或如果它是非标准比特率，则此字段应设置为0。建议在可能的情况下避免可变比特率编码。

fwHeadMode

流模式，如下列标记所定义：

ACM_MPEG_STEREO – 立体声。

ACM_MPEG_JOINTSTEREO – 联合立体声。

ACM_MPEG_DUALCHANNEL – 双通路（例如，双语流）。

ACM_MPEG_SINGLECHANNEL – 单通路。

一些法定的MPEG流可以包括不同模式的帧。在此情况下，上述标记可以OR，驱动程序可以指示流中所出现的模式。此情形特别用于联合立体声编码，编码器可以根据信号特征将其用于立体声与联合立体声之间的动态转换。在此情况下，ACM_MPEG_STEREO和ACM_MPEG_JOINTSTEREO标记均应设置。

fwHeadModeExt

包括联合立体声编码的额外参数；不用于其他模式。参见表3。一些法定的MPEG流可以包括不同模式扩展的帧。在此情况下，表3中的值可以OR在一起。注意，fwHeadModeExt只用于联合立体声编码；对于其他模式（单通路、双通路或立体声），它应设置为0。

一般情况下，编码器将根据信号特征将各种可能的mode_extension值动态转换。因此，对于标准的联合立体声编码，此字段应设置为0x000f。但是，如果需要限制编码器为特别的联合立体声编码类型，此字段可用于规定允许的类型。

wHeadEmphasis

描述解码器不强调必须的内容。这表示编码前对流执行的操作。参见表4。

fwHeadFlags

在音频帧标题中设置相应的标记：

ACM_MPEG_PRIVATEBIT – 设置专用位。

ACM_MPEG_COPYRIGHT – 设置版权位。

ACM_MPEG_ORIGINALHOME – 设置最初的/主页位。

ACM_MPEG_PROTECTIONBIT – 设置保护位，并将16比特差错保护码插入到每帧中。

ACM_MPEG_ID_MPEG1 – 将IDbit设为1，定义该流为MPEG-1音频流。此标记必须明确设置以保持与未来MPEG音频扩展

(即MPEG-2) 的兼容性。

编码器将使用这些标记值在每个MPEG 音频帧的标题中设置相应的比特。当描述一个编码的数据流时，这些标记在每个帧标题中表示标记的逻辑OR。如果在流中的帧标题中设置一个或多个版权位，则应设置ACM_MPEG_COPYRIGHT标记。因此，这些标记值对每个音频帧不是必须有效。

dwPTSLow 此字段（与下列字段一起）由第1个音频流的帧的表示时间标记（PTS）组成，源自MPEG系统层。dwPTSLow包括33比特PTS的32LSB。可以采用PTS将音频流与相关视频流重新组合。如果音频流与系统层无关，则此字段应设置为0

dwPTSHigh 此字段（与前字段一起）由第1个音频流的帧的表示时间标记（PTS）组成，源自MPEG系统层。dwPTSHigh的LSB包括33比特PTS的MSB。可以采用PTS将音频流与相关视频流重新组合。如果音频流与系统层无关，则此字段应设置为0

注 – 前两个字段可以按单个64比特整数处理；任选地，dwPTSHigh字段可以按标记处理以确定是否设置或清除MSB。

表 1
允许的比特率 (bit/s)

MPEG帧标题代码	层I	层II	层III
'0000'	空闲格式	空闲格式	空闲格式
'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
'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 2

层 II 允许的模式 – 比特率组合

比特率 (bit/s)	允许的模式
32000	单通路
48000	单通路
56000	单通路
64000	所有模式
80000	单通路
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, 双通路
256000	立体声, 强度立体声, 双通路
320000	立体声, 强度立体声, 双通路
384000	立体声, 强度立体声, 双通路

表 3

模式扩展

fwHeadModeExt	MPEG 帧标题代码	层 I 和 II	层 III
0x0001	'00'	强度立体声中的子带4-31	非强度或MS-立体声编码
0x0002	'01'	强度立体声中的子带8-31	强度立体声
0x0004	'10'	强度立体声中的子带12-31	MS-立体声
0x0008	'11'	强度立体声中的子带16-31	强度和 MS-立体声编码

表 4

重点字段

wHeadEmphasis	MPEG 帧标题代码	要求解除重点
1	'00'	非重点
2	'01'	50/15 μ s 重点
3	'10'	保留
4	'11'	ITU-T J.17建议书

1.3 数据字段中使用的标记

fwHeadLayer

为<fwHeadLayer> 字段定义下列标记，对于编码，应该设置这些标记之一，使编码器得知所使用的层。对于解码，驱动程序可以检查这些标记以确定是否能够对流进行解码。注意，法定的MPEG流可以使用单个流内不同帧中的不同的层。因此，可以设置为几种标记。

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

fwHeadMode

下列标记为<fwHeadMode> 字段规定。对于编码，应该设置这些标记之一，因此编码器得知所采用的模式；对于联合立体声编码，典型的 ACM_MPEG_STEREO 和 ACM_MPEG_JOINTSTEREO 标记均应设置，因此编码器只在它比立体声更有效时才可以使用联合立体声编码。对于解码，驱动程序可以检查这些标记以确定是否能够对流进行解码。注意，法定的MPEG流可以使用单个流内不同帧中的不同的层。因此，可以设置为几种标记。

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

fwHeadModeExt

表3为<fwHeadModeExt> 字段定义标记。此字段只用于联合立体声编码；对于其他编码模式，此字段应设置为0。对于联合立体声编码，这些标记指示允许编码器使用的联合立体声编码的类型。通常，一个编码器将动态地选择最适合输入信号的模式扩展；因此，应用中典型的情况是将此字段设置为0x000f，因此编码器可以在所有可能性中选择；但是，它也可以通过清除一些标记限制编码器。对于编码流，此字段指示流中出现的MPEG *mode_extension* 字段的值。

fwHeadFlags

下列标记为 <fwHeadFlags> 字段定义。这些标记应在编码前设置，因此适当的比特设置在MPEG 帧标题中。当描述一个已编码的MPEG 音频流时，这些标记表现每个音频帧的标题中相应比特的逻辑OR，如果该比特设置在任何帧中，它设置在<fwHeadFlags> 字段内。如果一项应用的预编码的MPEG音频比特流围绕一个RIFF WAVE标题，它负责分解该比特流并设置此字段内的标记。

```
#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        (0x0010)
```

1.4 音MPEG 文件中的频数据

<data chunk> 由一个MPEG-1音频序列组成，定义见ISO11172规范第3部分（音频）。此序列由一个比特流组成，存储在数据块中作为字节排列。在一个字节中，MSB为流的第一个比特，LSB是最后一个比特。数据不是反向字节。例如，下列数据包括典型音频帧标题的第1个16比特（从左到右）：

Syncword	ID	层	ProtectionBit	...
11111111111111	1	10	1	...

此数据应以下列次序存储在字节内：

Byte0	Byte1	...
FF	FD	...

1.4.1 MPEG 音频帧

一个MPEG音频序列由一系列的音频帧组成，每个以帧标题开始。此帧标题内的大部分字段对应于上述规定的MPEG1WAVEFORMAT结构中的字段。对于编码，这些字段可以设置为MPEG1WAVEFORMAT结构，并且当编码时，驱动程序可以使用此信息在帧标题中设置适当的比特。对于解码，驱动程序可以检查这些字段以确定是否能够对流进行解码。

1.4.2 编码

对MPEG音频流进行编码的驱动程序应读取MPEG1WAVEFORMAT结构的标题字段，并在MPEG帧标题中设置相应的比特。如果驱动程序还需要其他信息，必须从配置对话框或通过驱动程序回叫功能得到此信息。对于更多的信息，参见辅助数据部分。

如果预编码MPEG音频流配有RIFF标题，其应用功能是将比特流分入其各组成部分中并设置MPEG1WAVEFORMAT结构中的字段。如果抽样频率或比特率指标在整个数据流中不是恒定的，驱动程序应将相应的MPEG1WAVEFORMAT字段（<nSamplesPerSec> 和<dwHeadBitrate>）设置为0，如上所述。如果流包括的帧多于一层，它应为流中出现的所有层在<fwHeadLayer>设置标记。因为字段如<fwHeadFlags>可能从帧到帧变化，在设置和测试这些标记时必须使用告警：一般情况下，一个应用不应依赖对每个帧有效。当设置这些标记时，应支持下列指导原则：

- ACM_MPEG_COPYRIGHT应设置，如果流中的所有帧均有版权位集的话。
- ACM_MPEG_PROTECTIONBIT应设置，如果流中的所有帧均有保护位集的话。
- ACM_MPEG_ORIGINALHOME应设置，如果流中的所有帧均有最初的/主页位集的话。如果对流进行复制，此比特应被清除。
- ACM_MPEG_PRIVATEBIT应设置，如果流中的所有帧均有专用位集的话。
- ACM_MPEG_ID_MPEG1应设置，如果流中的所有帧均有ID集的话。对于MPEG-1流，ID比特应总被设置；但是，MPEG的未来扩展（如MPEG-2多通路格式）可以有清除的ID比特。

如果MPEG音频流源自系统层MPEG流，或如果流将会组合到系统层中，则可以采用PTS字段。PTS是用于不同字段同步的MPEG系统层中的字段，MPEG PTS字段是33比特，因此，RIFF WAVE格式标题将其值存储在两个字段中：<dwPTSLow> 包括PTS的32LSB，<dwPTSHigh> 包括MSB。这两个字段可一起采用作为一个64比特整数；任选地，<dwPTSHigh>字段可以作为标记测试，以确定是否设置或清除MSB。当从系统层提取一个音频流时，驱动程序应将PTS字段设置为音频数据的第1帧的PTS。这还可用于将流再组合入系统层中。PTS字段不应用于其他任何目的。如果音频流与MPEG系统层不相关，则PTS字段应设置为0。

1.4.3 解码

驱动程序可以测试MPEG1WAVEFORMAT结构的字段以确定是否能够对流进行解码。但是，驱动程序必须得知某些字段，如 <fwHeadFlags> 字段，可能不是对比特流中每个帧均是一致的。驱动程序不应再使用MPEG1WAVEFORMAT结构的字段执行实际的解码。解码参数应完全取自MPEG数据流。

驱动程序可检查<nSamplesPerSec> 字段以确定是否它支持特定的抽样频率。如果MPEG流包括具有可变抽样率的数据，则<nSamplesPerSec> 字段将设置为0。如果驱动程序不能处理此类型的数据流，则它不应试图解码此数据，应立即放弃。

1.5 辅助数据

MPEG 音频帧中的音频数据不可填充整个帧。所有其余数据称为辅助数据。此数据可以具有所需要的任何格式，并可用于传送任何类型的附加信息。如果驱动程序希望支持辅助数据，它必须可以传送数据至（和自）呼叫应用。驱动程序可以将回叫功能用于此目的。基本上，每当它将辅助数据传送到应用（即解码）或它需要更多的辅助数据（编码）时，驱动程序可以呼叫一个特定的回呼功能。

驱动程序应知道并非所有应用希望处理辅助数据。因此，只有在应用明确请求时，驱动程序才提供此业务。驱动程序可以定义允许和禁止回叫功能的定制消息。为更加灵活，可以为编码和解码操作分别定义消息。

注意，此方法不适用于所有的驱动程序或所有应用。只在说明如何支持辅助数据时才使用。

注 1 – 关于辅助数据更多的信息参见<MPEG_Audio_Extension_chunk>，其应用于符合广播波格式的MPEG文件。见附件2正文的第2部分。

参考文献

ISO/IEC 11173-3: MPEG 1。

ISO/IEC 13818-3: MPEG 2。

注 – 微软® 文件可以从下列网址获得：<http://www.microsoft.com>。

附件3

BWF的规范

广播中音频数据文件的格式

元数据规范

1 引言

本附件包括BWF的使用规范，以承载由DAW收集并计算的音频材料信息（见图2）。对于声音信号和所有其他相关的元数据，BWF文件用作与平台无关的存储器。接收存档服务器能够将所需要的信息从文件中取得并必须时使用它；例如，将其输入到数据库中等（见图3）。

图2
工作站将数据收集到BWF文件中

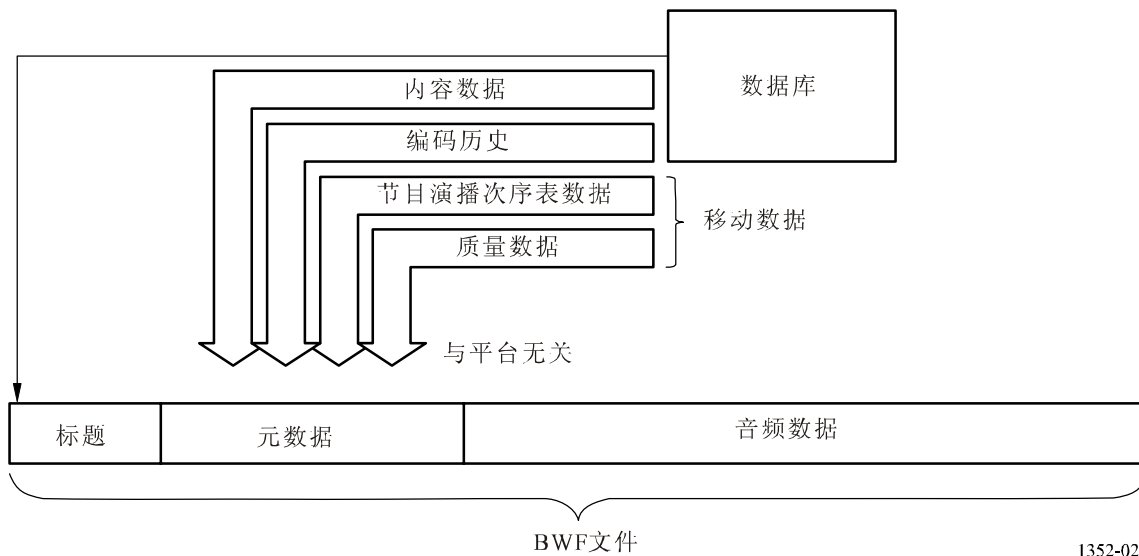
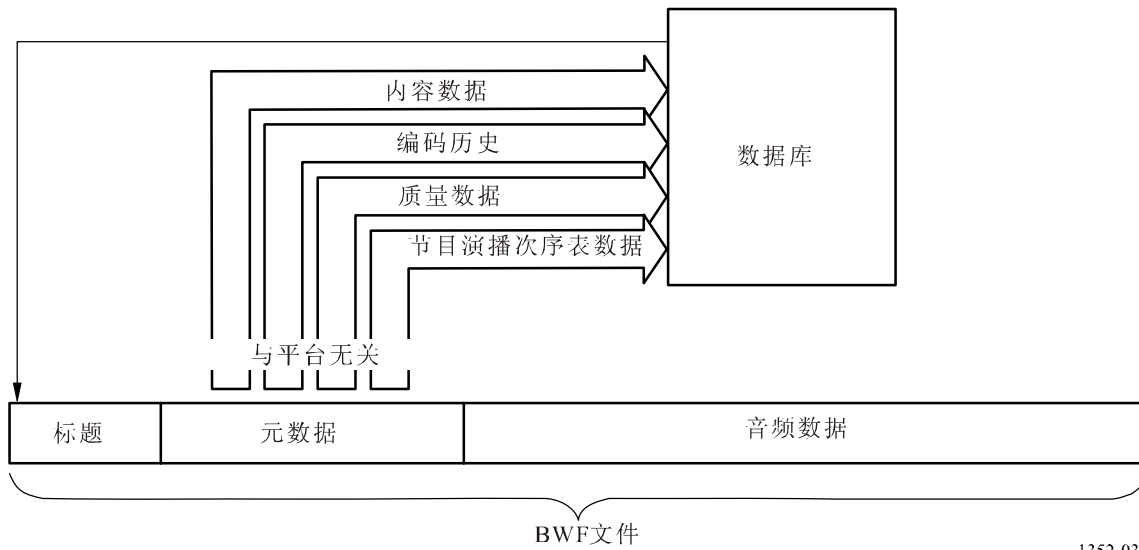


图3
接收存档服务器从BWF文件中提取数据



本附件规定新的块，承载在基本BWF文件未出现过的信息，并规定如果使用BWF中现有的块。

当编辑包括质量报告的BWF文件时应当小心。如果编辑系统结合了多个BWF文件，编辑结果表（EDL）应指向编码历史的适当部分和每个BWF源文件的质量块。此外，如果新文件源自其他文件部分，则应为新文件产生新的编码历史和质量块。

2 获取报告

为保证最初的单个模拟或数字承载者存档，将最初完整的的质量的声音信号重新记录入BWF文件。获取报告包括的信息涉及从模拟到数字域的完整处理链，或数字域内的传输（例如，源自CD或DAT）。

与音频信号分析的数据一起，作为BWF文件的元数据的一部分，形成获取报告。

获取报告由三部分组成：

- BWF文件的<bext>块内的CodingHistory字段。包括整个传输链的详述，例如，从磁带的类型、光盘或DAT磁带直到BWF文件（声音信号的历史）。
- <qlty>块中的质量报告。包括的信息涉及在波数据块中影响所记录声音信号质量的所有相关事件。每个事件，不论是否被运营商或计算机认可，均将详细列出其事件类型、精确的时间标记、优先级和事件状态。还将报告全部质量参数等。
- <qlty>块中的节目演播次序表是一个事件列表，标记有精确的时间标记和对声音信号更多的描述，例如，唱段的开始或重要讲话的起点。因此存档程序能够通过计算机辅助工具完成数据库的元数据。

2.1 获取报告的语法

- 获取报告由ASCII (ISO646) [ISO/IEC, 1991]字符串组成, 排列行最高至256字符。
- 每行由<CR/LF> (ASCII 0Dh, 0Ah) 终止。
- 一行可包括一个或多个由逗号分隔的可变串 (ASCII 2Bh) 。
- 可变串为ASCII字符并应不包含逗号。
- 分号 (ASCII3Bh) 在可变串中用作分隔符。

3 <bext>块中的CodingHistory 字段

编码历史字段中使用的串在附件 1的附录2中规定。为方便起见将此信息重复如下。

A=<ANALOGUE,>	模拟声音信号通道信息
A=<PCM,>	数字声音信号通道信息
F=<48000,441000,等。>	抽样频率 [Hz]
W=<16,18,20,22,24等。>	字长度 [bits]
M=< mono, stereo, 2-channel >	模式
T=<free ASCII-text string>	文本注释

4 质量块

质量块在第4.1节的斜体文本中定义:

4.1 质量块的元素

FileSecurityReport: 此字段包括 QualityChunk的FileSecurityCode
它是 32比特值包括校验和 [0231]。

FileSecurityWave: 此字段包括BWF 波数据的FileSecurityCode
它是 32比特值包括校验和[0231]。

```
Quality-chunk typedef struct {
    DWORD    ckID ;                /* ( quality-chunk) ckID='qlty' */
    DWORD    ckSize ;             /* size of quality chunk */
    BYTE     ckData[ckSize] ;     /* data of the chunk */
}
typedef struct quality_chunk {
    DWORD FileSecurityReport ;     /* FileSecurityCode of quality report */
    DWORD FileSecurityWave ;      /* FileSecurityCode of BWF wave data */
    CHAR BasicData[ ] ;          /* ASCII : « Basic data » */
    CHAR StartModulation[ ] ;    /* ASCII : « Start modulation data » */
    CHAR QualityEvent[ ] ;      /* ASCII : « Quality event data » */
    CHAR EndModulation[ ] ;     /* ASCII : « End modulation data » */
    CHAR QualityParameter[ ] ;  /* ASCII : « Quality parameter data » */
    CHAR OperatorComment[ ] ;   /* ASCII : « Comments of operator » */
    CHAR CueSheet[ ] ;          /* ASCII : « Cue sheet data » */
} quality_chunk
```

BasicData: 获取的基本数据。

B= 包括关于声音材料的基本数据的ASCII串。

- 档案号 (AN) : 档案号 (最大32字符)。
- 标题 (TT) : 标题/取自声音数据 (最多256个字符)。
- 持续时间 (TD) : 包括声音序列的持续时间的10ASCII 字符。
格式: « hh: mm: ss: d »
小时 hh: 0...23
分钟 mm: 0...59
秒 ss: 0...59
1/10s d: 0...9
- 日期 (DD) : 包括数字化的日期的10 ASCII 字符。
格式: « yyyy: mm: dd »
年 yyyy: 0000...9999
月 mm: 0...12
日 dd: 0...31
- 运营商 (OP) : 包括执行数字化操作的人的姓名的ASCII串 (最大64字符)。
- 复制站 (CS) : 包括创建文件所采用的工作站的类型和序列号的ASCII串 (最大64字符)。
- StartModulation: 最初记录的调制开始 (SM)。
- SM= 包括从文件开始计的声音信号的开始时间的10ASCII字符。
格式: « hh: mm: ss: d »
小时 hh: 0...23
分钟 mm: 0...59
秒 ss: 0...59
1/10 s d: 0...9
- 抽样计数 (SC) : 从文件开始计的SM点的抽样地址码 (调制的十六进制开始)。
格式: « #####H »
0H..... FFFFFFFFH (0..... 4.295×10^9)
- 注释 (T) : 包括注释的ASCII串。
- QualityEvent 描述声音信号中每个质量事件的信息。每个事件使用一个QualityEvent串。
- Q= 包括质量事件的ASCII串 (最多256个字符)。
- 事件编号 (M) : 运营商人工启动的编号标志
格式: « M### » ###: 001...999
- 事件编号 (A) : 系统自动启动的编号标志
格式: « A### » ###: 001...999
- 优先级 (PRI) : 质量事件的优先级
格式: « # » #: 1 (LO)5 (HI)
- 时间标记 (TS) : 包括从文件开始计的质量事件的时间标记的10 ASCII 字符。

	格式: « hh:mm:ss:d »	
	小时	hh: 0...23
	分钟	mm: 0...59
	秒	ss: 0...59
	1/10 s	d: 0...9
事件类型 (E):	描述事件类型的ASCII串 (最大16字符), 例如 “点击”, “AnalogOver”, “Transparency” 或超过限制的 QualityParameter (如下定义), 例如 “QP: 方位: L-20.9smp”。	
状态 (S):	包括事件的处理状态的ASCII串 (最大16字符), 例如 “不清楚”、“检查”、“恢复”、“删除”。	
注释 (T):	包括注释的ASCII串。	
抽样计数 (SC):	从文件开始计的TS点的抽样地址代码 (十六进制ASCII)。 格式: « #####H » 0H..... FFFFFFFFH (0..... 4.295×10^9)	
QualityParameter	描述声音信号的质量参数 (QP)。	
P=	包括质量参数的ASCII串 (最多256个字符)。	
参数 (QP):	MaxPeak:	-xx.x dBFSL; -yy.y dBFSR [-99.9...-00.0]
	MeanLevel:	-xx.x dBFSL; -yy.y dBFSR [-99.9...-00.0]
	相互关系:	$\pm x.x$ [-1.0.....+1.0]
	动态:	xx.x dBL; yy.y dBR [00.0..... 99.9] (动态范围)
	ClippedSamples:	xxxx smpL; yyyy smpR [0.....9999]
	SNR:	xx.x dBL; yy.y dBR [00.0.....99.9] (信噪比)
	带宽:	xxxxx HzL; yyyyyy HzR [0.....20000]
	方位:	L \pm xx.x smp [-99.9....+99.9]
	平衡:	L \pm x.x dB [-9.9.....+9.9]
	DC-偏移:	x.x %L; y.y %R [0.0.....9.9]
	讲话:	xx.x% [0.0.....99.9]
	立体声:	xx.x% [0.0.....99.9]
	(L = 左通路, R = 右通路)	
质量因素 (QF):	概述声音文件的质量因素[1.....5 (最好), 0 = 未定义]	
检查员 (IN):	包括检查声音文件的人的姓名的ASCII串 (最大64字符)。	

文件状态 (FS) :	描述状态“准备传输?”的ASCII字符串。 [Y(es)/ N(o)/ U: 文件准备好/未准备/FS未定义]。												
OperatorComment	运营商注释												
T=	包括注释的ASCII串 (最多256个字符)。												
EndModulation	调制结束。												
EM=	包括声音信号的调制结束时间的10ASCII字符。 格式: « hh:mm:ss:d » <table> <tr> <td>小时</td> <td>hh:</td> <td>0...23</td> </tr> <tr> <td>分钟</td> <td>mm:</td> <td>0...59</td> </tr> <tr> <td>秒</td> <td>ss:</td> <td>0...59</td> </tr> <tr> <td>1/10 s</td> <td>d:</td> <td>0...9</td> </tr> </table>	小时	hh:	0...23	分钟	mm:	0...59	秒	ss:	0...59	1/10 s	d:	0...9
小时	hh:	0...23											
分钟	mm:	0...59											
秒	ss:	0...59											
1/10 s	d:	0...9											
抽样计数 (SC) :	EM 点的抽样地址代码 (十六进制 ASCII)。 格式: « #####H » 0H.....FFFFFFFFH (0.....4.295 × 10 ⁹)												
注释 (T) :	包括注释的ASCII串。												
CueSheet	节目演播次序表数据。												
C=	包括提示点的ASCII串 (最多256个字符)。												
提示号 (N) :	由系统自动产生的提示点号。 格式: « N### » ###: 001...999												
时间标记 (TS) :	包括提示点的时间标记的10 ASCII 字符。 格式: « hh: mm: ss: d » <table> <tr> <td>小时</td> <td>hh:</td> <td>0...23</td> </tr> <tr> <td>分钟</td> <td>mm:</td> <td>0...59</td> </tr> <tr> <td>秒</td> <td>ss:</td> <td>0...59</td> </tr> <tr> <td>1/10s</td> <td>d:</td> <td>0...9</td> </tr> </table>	小时	hh:	0...23	分钟	mm:	0...59	秒	ss:	0...59	1/10s	d:	0...9
小时	hh:	0...23											
分钟	mm:	0...59											
秒	ss:	0...59											
1/10s	d:	0...9											
文本 (T) :	包括描述提示点注释的ASCII串。 例如, “咏叹调开始”。												
抽样计数 (SC) :	TS 点的抽样地址代码 (十六进制 ASCII) 格式: « #####H » 0H..... FFFFFFFFFH (0...4.295 × 10 ⁹)												

5 获取报告的范例

5.1 模拟材料的数字化过程

(<bext>块的CodingHistory 字段中包括的基本信息)

系列

01 A=ANALOGUE, M= 立体声, T=Studer A816; SN1007; 38; No./telcom;
Agfa PER528 <CR/LF>

02 A=PCM, F=48000, W=18, M=立体声, T=NVision NV 1000; A/D<CR/LF>

03 A=PCM, F=48000, W=16, M=立体声, T=nodither; DIO<CR/LF>

(质量块中的QualityReport)

系列号:

01 <FileSecurityReport>

02 <FileSecurityWave>

03 B=CS=QUADRIGA2.0; SN10012, OP=运营商名称<CR/LF>

04 B=AN=存档数, TT=声音的标题<CR/LF>

05 B=DD= yyyy: mm: dd, TD=hh: mm: ss: d<CR/LF>

06 SM=00: 00: 04: 5, T=磁带噪声改变氛围, SC=34BC0H<CR/LF>

07 Q=A001, PRI=2, TS=00: 01: 04: 0, E=点击, S=不清楚, SC=2EE000H<CR/LF>

08 Q=A002, PRI=3, TS=00: 12: 10: 3, E=DropOut, S=检查, SC=216E340H<CR/LF>

09 Q=A003, PRI=4, TS=00: 14: 23: 0, E=Transparency, S=检查, SC=2781480H<CR/LF>

10 Q=M004, PRI=1, TS=00: 18: 23: 1, E=PrintThrough, S=检查, SC=327EF40H<CR/LF>

11 Q=A005, PRIG, TS=00: 20: 01: 6, E= Click0n, S=不清楚, T=需要恢复,
SC=3701400H<CR/LF>

12 Q=A006, PRI=5, TS=00: 21: 20: 3, E=QP: 方位: L=-20.9smp, S=不清楚,
SC=3A9B840H<CR/LF>

13 Q=A007, PRI=3, TS=00: 21: 44: 7, E=AnalogOver, S=检查, SC=3BB9740H<CR/LF>

14 Q=A008, TS=00: 22: 11: 7, E=C1ickOff, SC=3BB9740H<CR/LF>

15 Q=A009, PRI=1, TS=00: 28: 04: 0, E=DropOut, S=删除, SC=4D16600H<CR/LF>

16 EM=00: 39: 01: 5, T=掌声渐弱, SC=6B2F740H<CR/LF>

17 P=QP: MaxPeak: -2.1dBFS; -2.8dBFSR<CR/LF>

18 P=QP: MeanLevel: -11.5dBFS; 8.3dBFSR<CR/LF>

19 P=QP: Correlation: +0.8<CR/LF>

20 P=QP: Dynamic: 51.4dB; 49.6dBR<CR/LF>

21 P=QP: ClippedSamples: Osmpl; Osmpr<CR/LF>

22 P=QP: SNR: 32.3dB; 35.1dBR<CR/LF>

23 P=QP: 带宽: 8687HzL; 7943HzR<CR/LF>

24 P=QP: 方位: L-6.2smp<CR/LF>

25 P=QP: 平衡 L: +2.1dB<CR/LF>

26 P=QP: DC-偏移: 0.0%L; 0.0%R<CR/LF>

27 P=QP: 讲话: 64.2%<CR/LF>

28 P=QP: 立体声: 89.3%<CR/LF>

29 P=QF=2<CR/LF>

30 P=IN=检查员名称<CR/LF>

31 P=FS=N<CR/LF>

(质量块中的CueSheet)

系列号:

32 C=N001, TS=00:17:02:5, T=讲话开始, SC=2ECE6C0 H<CR/LF>

33 C=N002, TS=00:33:19:2, T=咏叹调开始, SC=5B84200H<CR/LF>

范例 1的解释**(CodingHistory中的基本信息)**

系列1: 模拟磁带类型Agfa PER528在录音机Stude A816上回放, 序列号1007, 采用电信扩展:

磁带速率: 38 cm/s

模式: 立体声

系列2: 对于数字化A/D转换类型NVision NV 1000采用:

抽样频率: 48 kHz

编码分辨率: 18比特每抽样

模式: 立体声

系列3: 最初的文件记录为线性BWF文件, PCM编码采用无抖动重新录音站的数字输入:

抽样频率: 48 kHz

编码分辨率: 16比特 每抽样

模式: 立体声

(质量块中的QualityReport)

系列1至2: 质量块和波数据的文件安全码。

系列3至5: 重新录音站QUADRIGA2.0, 运营商(OP)采用的序列号10012。磁带有存档数(AN)和标题(TT)按日期数字化(DD)。BWF文件中声音信号的持续时间是(TD)。

系列6: 在时间标记(TS)上开始调制(SM)和抽样计数(SC)与注释(T)。

系列7至15: 运营商(M)和/或系统控制(A)认可的事件(E)与优先级(PRI)和时间标记(TS)。事件状态(S)和注释(T)给出了更多信息。抽样计数(SC)给出了精确的时间标记。

系列16: 在时间标记上调制结束(EM)和抽样计数(SC)与注释(T)。

系列17至28: 波数据块中完整声音信号的质量参数(QP)。

系列29至31: 自动系统控制给出的概要质量因素(QF)和检查员(IN)名称, 并决定(FS)是否声音文件的质量为“准备传输”。

(质量块中的CueSheet)

系列32至33: 提示点标出讲话的开始和咏叹调的开始点。

5.2 光盘的获取过程

（<bext>块的CodingHistory 字段的基本信息）

系列号：

- 01 A=PCM， F=44100， W=16， M=立体声， T=SonyCDP-D500； SN2172； Mitsui CD-R74<CR/LF>
- 02 A=PCM， F=48000， W=24， M=立体声， T=DCS972； D/D<CR/LF>
- 03 A=PCM， F=48000， W=24， M=立体声， T=nodither； DIO<CR/LF>

（质量块中的QualityReport）

系列号：

- 01 <FileSecurityReport>
- 02 <FileSecurityWave>

等： 类似于上述第5.1节中的范例。

（质量块中的CueSheet）

类似于上述第5.1节中的范例。

范例2的解释

（CodingHistory中的基本信息）

- 系列1： CD可刻录型MitsuiCD-R74在CD播放机SonyCDP-D500上回放， 序列号2172：
- | | |
|--------|----------|
| 抽样频率： | 44.1 kHz |
| 编码分辨率： | 16比特每抽样 |
| 模式： | 立体声 |
- 系列2： 抽样率转换类型 DCS972采用：
- | | |
|--------|-------------------|
| 抽样频率： | 48 kHz（从44.1 kHz） |
| 编码分辨率： | 24比特每抽样 |
| 模式： | 立体声 |
- 系列3： 最初的文件记录为线性BWF文件， PCM编码采用无抖动重新录音站的数字输入：
- | | |
|--------|---------|
| 抽样频率： | 48 kHz |
| 编码分辨率： | 24比特每抽样 |
| 模式： | 立体声 |

（质量块中的QualityReport）

系列1至2： 质量块和波数据的文件安全码。

其他数据的使用按照类似于上述第5.1节的范例 1中的CD获取过程。

（质量块中的CueSheet）

节目演播次序表数据的使用按照类似于上述第5.1节的范例 1中的光盘获取过程。

5.3 DAT磁带的获取过程

(<bext>块的CodingHistory字段的基本信息)

系列号:

- 01 A=PCM, F=48000, W=16, M=立体声, T=SonyPCM-8500; SN1037;
TDKDA-R120 <CR/LF>
- 02 A=PCM, F=48000, W=16, M=立体声, T=无抖动; DIO<CR/LF>

(质量块中的QualityReport)

系列号:

- 01 <FileSecurityReport>
- 02 <FileSecurityWave>

等: 类似于上述第5.1节的范例。

(质量块中的CueSheet)

类似于上述第5.1节的范例。

范例3的解释

(CodingHistory中的基本信息)

系列1: DAT磁带类型TDKDA-8120在DAT录音机SonyPCM-8500上播放, 序列号1037:

抽样频率:	48 kHz
编码分辨率:	16比特每抽样
模式:	立体声

系列2: 最初的文件记录为线性BWF文件, PCM编码采用重新录音站无抖动数字输入:

抽样频率:	48 kHz
编码分辨率:	16比特每抽样
模式:	立体声

(质量块中的QualityReport)

系列1至2: 质量块和波数据文件安全码。

其他数据的使用按照类似于上述第5.1节的范例1中的DAT磁带的获取过程。

(质量块中的CueSheet)

节目演播次序表数据的使用按照类似于上述第5.1节的范例1中的DAT磁带的获取过程。
