

## РЕКОМЕНДАЦИЯ МСЭ-R BR.1352-3

**Формат файлов для обмена материалом звуковых программ с метаданными на носителях, созданных на основе информационных технологий**

(Вопрос МСЭ-R 58/6)

(1998-2001-2002-2007)

**Сфера применения**

В настоящей Рекомендации содержится спецификация фрагмента расширения аудиоданных радиовещания<sup>1</sup> и описание его использования с аудиоданными с ИКМ кодированием, а также формата MPEG-1 или MPEG-2. Также представлена информация о формате файлов для обмена ресурсами RIFF и возможностях его расширения на другие типы данных.

Ассамблея радиосвязи МСЭ,

*учитывая,*

- a) что носители данных, созданные на основе информационных технологий, в том числе диски и ленты для хранения данных, проникли во все сферы производства звуковых данных для радиовещания – нелинейный монтаж, перегон в эфире и архивирование;
- b) что эта технология обуславливает значительные преимущества в части эксплуатационной гибкости, производственного потока и автоматизации станций и вследствие этого является эффективной для модернизации существующих студий и проектирования новых студийных установок;
- c) что принятие единого формата файлов для обмена сигналами в значительной степени способствовало бы обеспечению взаимодействия отдельного оборудования и удаленных студий, это упростило бы целесообразную интеграцию процессов монтажа, эфирного перегона и архивирования;
- d) что в файл должен включаться минимальный набор связанной с радиовещанием информации для документирования метаданных, относящихся к аудиосигналу;
- e) что для обеспечения совместимости приложений разной сложности должен быть согласован минимальный набор функций, являющийся общим для всех приложений, способных обрабатывать рекомендуемый формат файлов;
- f) что в Рекомендации МСЭ-R BS.646 определяется формат цифрового аудиосигнала, используемый в производстве аудиоматериала для радио- и телевидения;
- g) что необходимость в обмене аудиоматериалом возникает также при использовании систем кодирования ISO/IEC 11172-3 и ISO/IEC 13818-3 для сжатия сигнала;
- h) что совместимость с используемыми в настоящее время в промышленности форматами файлов может сократить объем промышленных работ, необходимых для реализации этого формата в оборудовании;
- j) что стандартный формат информации о применении кодирования упростил бы использование информации после обмена программами;
- k) что качество аудиосигнала затрагивается при обработке этого сигнала, в частности при применении нелинейного кодирования и декодирования в процессах снижения битовой скорости,

---

<sup>1</sup> Фрагмент – это основной структурный блок файла в формате для обмена ресурсами (RIFF) корпорации Microsoft®.

*рекомендует,*

**1** что для обмена звуковыми программами на созданных на основе информационных технологий носителях параметры звукового сигнала, частота дискретизации, разрядность кодирования и частотная коррекция должны быть установлены согласно соответствующим частям Рекомендации МСЭ-R BS.646;

**2** что для обмена звуковыми программами в формате линейной импульсно-кодовой модуляции (ИКМ) на носителях, созданных на основе информационных технологий, должен использоваться формат файлов, описанный в Приложении 1;

**3** что, если аудиосигнал кодируется с использованием систем кодирования ISO/IEC 11172-3 или ISO/IEC 13818-3, для обмена звуковыми программами на носителях, созданных на основе информационных технологий, должен использоваться формат файлов, описанный в Приложении 1 и дополненный в соответствии с Приложением<sup>2</sup>;

**4** что, если для переноса информации об аудиоматериале, собранном и обработанном цифровой звуковой станцией (DAW) записи, используется формат файлов, описанный в Приложениях 1 и/или 2, метаданные должны соответствовать спецификациям, представленным в Приложении 3.

## Приложение 1

### Спецификация волнового формата для радиовещания

#### Формат файлов аудиоданных в радиовещании

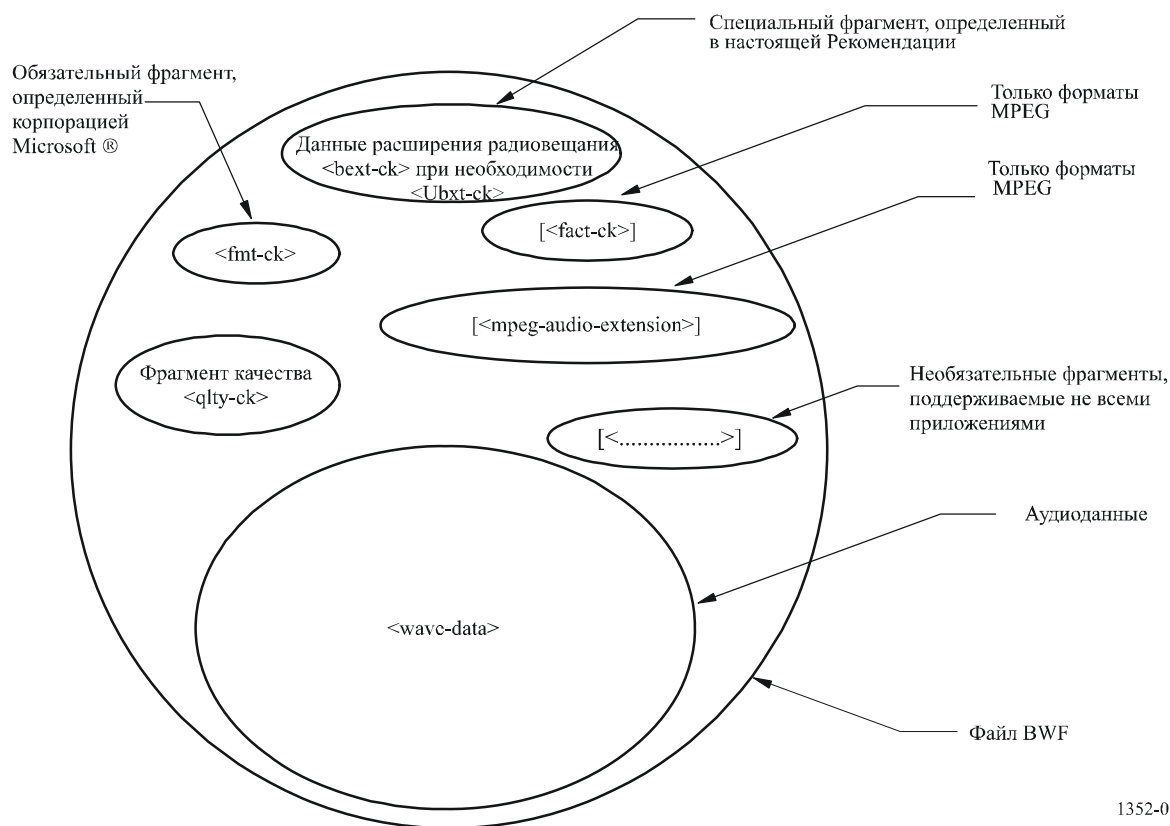
##### **1 Введение**

Основой волнового формата для радиовещания Broadcast Wave Format (BWF) является формат аудиофайлов WAVE корпорации Microsoft<sup>®</sup>, представляющий тип файлов, определенный в "Формате файлов для обмена ресурсами", RIFF Microsoft<sup>®</sup>. Файлы формата WAVE содержат аудиоданные. Основным структурным блоком формата файлов RIFF является "фрагмент" (chunk), и он содержит группу тесно связанных порций информации. Фрагмент содержит идентификатор фрагмента, целую величину, отражающую его длину в байтах, и переносимую информацию. Файл формата RIFF состоит из набора фрагментов.

В случае формата BWF в отношении исходного формата WAVE применяются определенные ограничения. Кроме того, файл BWF включает фрагмент расширения аудиоданных радиовещания <Broadcast Audio Extension>. Это показано на рисунке 1.

---

<sup>2</sup> Нельзя не отметить, что рекомендация в этом смысле может поставить в невыгодное положение разработчиков, использующих некоторые компьютерные платформы.

РИСУНОК 1  
Файл WBF

1352-01

В приложении содержится спецификация фрагмента расширения аудиоданных радиовещания, который используется во всех файлах WBF. Кроме того, в Дополнении 1 представлена информация о базовом формате RIFF и методах его расширения до других типов аудиоданных. В Дополнении 1 содержится также подробная информация о волновом формате с использованием ИКМ. Подробные спецификации расширения до других типов файлов, а также метаданных включены в Приложения 2 и 3.

### 1.1 Нормативные положения

Соблюдение положений данной Рекомендации носит добровольный характер. Однако в Рекомендации могут содержаться определенные обязательные положения (например, для обеспечения возможности взаимодействия или практической реализации), и соблюдение положений данной Рекомендации достигается в случае выполнения всех этих обязательных положений.

Для выражения необходимости выполнения требований используется синтаксис долженствования и соответствующие слова (такие, как "должен" и т. п.), а также их отрицательные эквиваленты. Использование этих слов не предполагает, что соблюдение положений данной Рекомендации является обязательным для какой-либо из сторон.

## 2 Файл волнового формата радиовещания Broadcast wave format (BWF)

### 2.1 Контент файла волнового формата радиовещания

Файл волнового формата радиовещания должен начинаться с обязательного заголовка "WAVE" формата RIFF корпорации Microsoft® и, по крайней мере, следующих фрагментов:

```

<WAVE-form>
RIFF('WAVE'
<fmt-ck> /* Формат аудиосигнала: ИКМ/MPEG */
<broadcast_audio_extension> /* информация о последовательности аудиоданных*/
<universal broadcast audio extension> /* ubxt требуется только для поддержки языка
многобайтовых символов */

<fact-ck> /* Фрагмент fact требуется только для форматов MPEG */
<mpeg_audio_extension> /* Фрагмент аудиорасширения MPEG требуется только
для форматов MPEG */

<wave-data> ) /* Звуковые данные */
<quality-chunk> /* Требуется только в случае, если необходима
информации о событиях, влияющих на качество*/

```

ПРИМЕЧАНИЕ 1. – В файл могут входить и другие фрагменты. Некоторые из них необязательно попадают в сферу применения настоящей Рекомендации. Приложения могут интерпретировать и использовать или не интерпретировать и не использовать эти фрагменты. Поэтому невозможно гарантировать целостность данных, содержащихся в таких неизвестных фрагментах. Однако отвечающие стандарту приложения передают неизвестные фрагменты прозрачным образом.

### 2.2 Существующие фрагменты, определенные как часть стандарта RIFF

Стандарт RIFF определен в документах, выпущенных корпорацией Microsoft®<sup>3</sup>. В данном приложении используются несколько уже определенных фрагментов, а именно:

```

fmt-ck
fact-ck.

```

Действующие описания этих фрагментов представлены для информации в Дополнении 1 к Приложению 1.

### 2.3 Фрагмент расширения аудиоданных радиовещания<sup>4</sup>

Дополнительные параметры, необходимые для обмена материалом между радиовещательными организациями, добавляются в специальный фрагмент расширения аудиоданных радиовещания "Broadcast Audio Extension", определенный ниже:

```

broadcast_audio_extension typedef struct {
    DWORD    ckID, /* (broadcastextension)ckID=bext. */
    DWORD    ckSize, /* размер фрагмента расширения */
    BYTE     ckData[ckSize], /* данные фрагмента */
}

```

```

typedef struct broadcast_audio_extension {
    CHAR Description[256], /* ASCII: "Описание звуковой последовательности" */
    CHAR Originator[32], /* ASCII: "Имя создателя оригинала" */
}

```

<sup>3</sup> Стандарт Microsoft "Формат файлов для обмена ресурсами", RIFF, размещен (в декабре 2005 года) по адресу: [http://www.tactilemedia.com/info/MCI\\_Control\\_Info.html](http://www.tactilemedia.com/info/MCI_Control_Info.html).

<sup>4</sup> См. в п. 2.4 определение фрагмента ubxt, необходимое для удобочитаемого представления информации фрагмента bext в многобайтовых наборах символов.

```

CHAR OriginatorReference[32], /* ASCII: "Ссылка на создателя оригинала" */
CHAR OriginationDate[10], /* ASCII: "гггг:мм:дд" */
CHAR OriginationTime[8], /* ASCII: "чч:мм:сс" */
DWORD TimeReferenceLow, /* Первый отсчет выборок начиная с полуночи, младшее слово */
DWORD TimeReferenceHigh, /* Первый отсчет выборок начиная с полуночи, старшее слово */
WORD Version, /* Версия BWF; двоичное число без знака */
BYTE UMID_0, /* Двоичный байт 0 поля UMID SMPTE */....
BYTE UMID_63, /* Двоичный байт 63 поля UMID SMPTE */
CHAR Reserved[190], /* 190 байтов, зарезервированных для будущего использования,
установлены в нулевое значение .NULL. */
CHAR CodingHistory[], /* ASCII: "Применение кодирования" */
} BROADCAST_EXT,

```

**Поле****Описание**

Description	<p>ASCII-строка (максимально 256 символов), содержащая описание последовательности в свободной форме. Для случая приложений, которые отображают только короткое описание, рекомендуется в первых 64 символах размещать резюме описания, а остальные 192 символа использовать для детального описания.</p> <p>Если длина строки составляет менее 256 символов, поле заканчивается символом нуля. (0x00)</p>
Originator	<p>ASCII-строка (максимально 32 символа), содержащая имя создателя оригинала/производителя звукового файла. Если длина строки составляет менее 32 символов, поле заканчивается символом нуля. (0x00)</p>
OriginatorReference	<p>ASCII-строка (максимально 32 символа), содержащая неразрешимую ссылку, выделенную организацией-создателем. Если длина строки составляет менее 32 символов, поле заканчивается символом нуля. (0x00)</p> <p>Стандартный формат "уникального" идентификатора источника (USID) для использования в поле ссылки на создателя оригинала OriginatorReference приведен в Добавлении 3 к Приложению 1.</p>
OriginationDate	<p>10 ASCII-символов, указывающие дату создания звуковой последовательности. Формат содержит следующие элементы: " 'год', '-', 'месяц', '-', 'день', '" , 4 символа для обозначения года и по 2 символа для остальных элементов.</p> <p>Формат указания года: 0000–9999.</p> <p>Формат указания месяца: 1–12.</p> <p>Формат указания дня: 1–31.</p> <p>В качестве разделителя для этих элементов должно использоваться тире, соответствующее формату ISO 8601. В некоторых унаследованных реализациях могут использоваться: ' _ ' подчеркивание, ' : ' двоеточие, ' ' пробел, ' . ' символ останова; воспроизводящее оборудование должно опознавать эти символы.</p>
OriginationTime	<p>8 ASCII-символов, указывающие время создания звуковой последовательности. Формат содержит следующие элементы: " 'час', '-', 'минута', '-', 'секунда' " – по два символа в каждом.</p> <p>Формат указания часов: 0–23.</p> <p>Формат указания минут и секунд: 0–59.</p> <p>В качестве разделителя для этих элементов должно использоваться тире, соответствующее формату ISO 8601. В некоторых унаследованных реализациях могут использоваться: ' _ ' подчеркивание, ' : ' двоеточие, ' ' </p>

	пробел, ‘.’ символ останова; воспроизводящее оборудование должно опознавать эти символы.
TimeReference	Это поле содержит временной код последовательности. Это 64-битовое значение, содержащее первый отсчет выборок после полуночи. Число выборок в секунду зависит от частоты дискретизации, определенной в поле <code>&lt;nSamplesPerSec&gt;</code> фрагмента <code>&lt;fmt-ck&gt;</code> .
Version	Двоичное число без знака, обозначающее версию BWF. Для указания Версии 1 значение этого поля устанавливается равным 0x0001.
UMID	64 байта, содержащие расширенное UMID, которое определяется полем SMPTE 330M. Если основное используемое UMID составляет 32 символа, последние 32 символа должны быть заполнены нулями. Если UMID отсутствует, все 64 символа должны быть заполнены нулями.  ПРИМЕЧАНИЕ. – Длина UMID кодируется в начале самого UMID.
Reserved	190 байтов, зарезервированных для расширения. Эти 190 байтов должны быть установлены в нулевое значение.
Coding History	Блок переменной длины ASCII-символов, содержащий 0 или более строк, каждая из которых заканчивается символами <code>&lt;CR&gt;&lt;LF&gt;</code> . Первым неиспользуемым символом должен быть символ нуля (0x00). Каждая строка должна содержать описание процесса кодирования, применявшегося к аудиоданным.  Каждое новое приложение кодирования должно добавлять новую строку, содержащую соответствующую информацию.  Стандартный формат данных применения кодирования представлен в Дополнении 2 к Приложению 1.  Данная информация должна содержать тип звука (ИКМ или MPEG) и его конкретные параметры:  ИКМ: режим (моно, стерео), разрядность выборки (8, 16 битов) и частота дискретизации,  MPEG: частота дискретизации, битовая скорость, Layer (I или II) и режим (моно, стерео, квазистерео или двойной канал).  Рекомендуется, чтобы производители кодеров обеспечивали наличие ASCII-строки для использования в данных о применении кодирования.

#### 2.4 Фрагмент универсального расширения аудиоданных радиовещания

Информация, содержащаяся во фрагменте расширения аудиоданных радиовещания Broadcast Audio Extension (bext), определенном в п. 2.3, может дополнительно переноситься в специальном фрагменте, называемом универсальным расширением аудиоданных радиовещания (Universal Broadcast Audio Extension), или “ubxt”, который предназначен для удобочитаемого представления информации фрагмента bext во многобайтовых языках. Базовая структура этого фрагмента метаданных та же, что и структура фрагмента bext. Четыре удобочитаемых элемента, uDescription, uOriginator, uOriginatorReference и uCodingHistory, описываются в формате UTF-8 (Формат преобразования универсального набора символов, 8-битовая форма) вместо формата ASCII. Размер первых трех элементов в восемь раз превышает размер данных соответствующего элемента фрагмента bext. Структура фрагмента ubxt определяется следующим образом:

```
typedef struct chunk_header {
    DWORD   ckID;           /* (universal broadcast extension)ckID=ubxt */
    DWORD   ckSize;        /* размер фрагмента расширения */
    BYTE    ckData[ckSize]; /* данные фрагмента */
}
```

```

} CHUNK_HEADER;
typedef struct universal_broadcast_audio_extension {
    BYTE    uDescription[256*8];    /* UTF-8: "описание звуковой последовательности" */
    BYTE    uOriginator[32*8];     /* UTF-8: "Имя создателя оригинала" */
    BYTE    uOriginatorReference[32*8]; /* UTF-8: "Ссылка на создателя оригинала" */
    CHAR    OriginationDate[10];   /* ASCII: "гггг:мм:дд" */
    CHAR    OriginationTime[8];    /* ASCII: "чч:мм:сс" */
    DWORD   TimeReferenceLow;      /* Первый отсчет выборок начиная с полуночи, младшее слово */
    DWORD   TimeReferenceHigh;     /* Первый отсчет выборок начиная с полуночи, старшее слово */
    WORD    Version;               /* Версия BWF; двоичное число без знака */
    BYTE    UMID_0;                /* Двоичный байт 0 поля UMID SMPTE */
    BYTE    UMID_63;               /* Двоичный байт 63 поля UMID SMPTE */
    CHAR    Reserved[190];         /* 190 байтов, зарезервированных для будущего использования, установлены в нулевое значение "NULL" */
    BYTE    uCodingHistory[];      /* UTF-8: "Информация о кодировании" */
} UNIV_BROADCAST_EXT;

```

Поле	Описание
uDescription	UTF-8-строка, 2048 байтов или менее, содержащая описание последовательности. Если данные отсутствуют или длина строки составляет менее 2048 байтов, первый неиспользуемый байт должен быть нулевым символом (0x00).
uOriginator	UTF-8-строка, 256 байтов или менее, содержащая имя создателя звукового файла. Если данные отсутствуют или длина строки составляет менее 256 байтов, первый неиспользуемый байт должен быть нулевым символом (0x00).
uOriginatorReference	UTF-8- строка, 256 байтов или менее, содержащая ссылку, выделенную организацией-создателем. Если данные отсутствуют или длина строки составляет менее 256 байтов, первый неиспользуемый байт должен быть нулевым символом (0x00).
OriginationDate	10 ASCII-символов, указывающих дату создания звуковой последовательности. Формат содержит следующие элементы: " 'год', '-', 'месяц', '-', 'день', '" , 4 символа для обозначения года и по 2 символа для остальных элементов. Формат указания года: 0000–9999. Формат указания месяца: 1–12. Формат указания дня: 1–31. В качестве разделителя для этих элементов должно использоваться тире, соответствующее формату ISO 8601. В некоторых унаследованных реализациях могут использоваться: ' _ ' подчеркивание, ' : ' двоеточие, ' ' пробел, ' . ' символ останова; воспроизводящее оборудование должно опознавать эти символы.

OriginationTime	<p>8 ASCII-символов, указывающих время создания звуковой последовательности. Формат содержит следующие элементы: "‘час,’-‘,минута,’-‘,секунда’" – по два символа в каждом.</p> <p>Формат указания часов: 0–23.</p> <p>Формат указания минут и секунд: 0–59.</p> <p>В качестве разделителя для этих элементов должно использоваться тире, соответствующее формату ISO 8601. В некоторых унаследованных реализациях могут использоваться: ‘_’ подчеркивание, ‘:’ двоеточие, ‘ ’ пробел, ‘.’ символ останова; воспроизводящее оборудование должно опознавать эти символы.</p>
TimeReference	<p>Это поле содержит временной код последовательности. Это 64-битовое значение, содержащее первый отсчет выборок после полуночи. Число выборок в секунду зависит от частоты дискретизации, определенной в поле &lt;nSamplesPerSec&gt; фрагмента &lt;fmt-ck&gt;.</p>
Version	<p>Двоичное число без знака, обозначающее версию BWF. Для указания Версии 1 значение этого поле устанавливается равным 0x0001.</p>
UMID	<p>64 байта, содержащие расширенное UMID, которое определяется полем SMPTE 330M. Если основное используемое UMID составляет 32 символа, последние 32 символа должны быть заполнены нулями. Если UMID отсутствует, все 64 символа должны быть заполнены нулями.</p> <p>ПРИМЕЧАНИЕ. – Длина UMID кодируется в начале самого UMID.</p>
Reserved	<p>190 байтов, зарезервированных для расширения. Эти 190 байтов должны быть установлены в нулевое значение.</p>
uCoding History	<p>Блок переменной длины ASCII-символов, содержащий 0 или более строк, каждая из которых заканчивается символами &lt;CR&gt;&lt;LF&gt;. Первым неиспользуемым символом должен быть символ нуля (0x00).</p> <p>Каждая строка должна содержать описание процесса кодирования, применявшегося к аудиоданным. Каждое новое приложение кодирования должно добавлять новую строку, содержащую соответствующую информацию.</p> <p>Стандартный формат данных о применении кодирования представлен в Дополнении 2 к Приложению 1.</p> <p>Данная информация должна содержать тип звука (ИКМ или MPEG) и его конкретные параметры:</p> <p>ИКМ: режим (моно, стерео), разрядность выборки (8, 16 битов) и частота дискретизации,</p> <p>MPEG: частота дискретизации, битовая скорость, Layer (I или II) и режим (моно, стерео, квазистерео или двойной канал).</p>

ПРИМЕЧАНИЕ 1. – Все элементы, за исключением uDescription, uOriginator, uOriginatorReference и uCodingHistory, должны иметь то же содержимое, что и соответствующие элементы фрагмента bext п. 2.3.

ПРИМЕЧАНИЕ 2. – Если данное кодовое значение в UTF-8 не входит в поднабор (определенный в главе 12 ISO/IEC 10646:2003), поддерживаемый какой-либо частью обрабатывающего оборудования, это значение должно оставаться неизменным и игнорироваться в процессе обработки.



## Дополнение 1 к Приложению 1 (информационное)

### Формат файлов RIFF WAVE (.WAV)

Представленная в настоящем Дополнении информация взята из документов, в которых содержатся спецификации формата файлов RIFF Microsoft®. Эта информация приводится только для информации.

#### 1 Формат звуковых файлов, волновая форма (WAVE)

Формат WAVE определяется следующим образом. Программы должны ожидать (и игнорировать) любые неизвестные фрагменты, как и во всех случаях форматов RIFF. Вместе с тем фрагмент <fmt-ck> должен всегда следовать до формата <wave-data>, и наличие обоих этих фрагментов в файле типа WAVE является обязательным.

```
<WAVE-form> ->
    RIFF ( 'WAVE'
        <fmt-ck>           // Фрагмент Format
        [<fact-ck>]       // Фрагмент Fact
        [<other-ck>]     // Другие необязательные фрагменты
        <wave-data> )    // Звуковые данные
```

Фрагменты WAVE описываются в следующих ниже разделах.

#### 1.1 Фрагмент формата WAVE

Фрагмент формата WAVE <fmt-ck> описывает формат <wave-data>. Фрагмент <fmt-ck> определяется следующим образом:

```
<fmt-ck> ->fmt( <common-fields>
                <format-specific-fields> )
<common-fields> ->
    struct{
        WORD wFormatTag,           /* Категория формата */
        WORD nChannels,           /* Число каналов */
        DWORD nSamplesPerSec,     /* Частота дискретизация */
        DWORD nAvgBytesPerSec,   /* Для оценки буфера */
        WORD nBlockAlign,        /* Размер блока данных */
    }
```

Поля в части <common-fields> данного фрагмента определяются следующим образом:

Поле	Описание
wFormatTag	Число, указывающее категорию формата файла WAVE. Это значение определяет содержимое части <format-specific-fields> фрагмента <fmt-ck> и интерпретацию данных волновой формы.
nchannels	Число каналов, представленных в данных волновой формы, 1 – для моно и 2 – для стерео.
nSamplesPerSec	Частота дискретизации (количество выборок сигнала в секунду), с которой должен воспроизводиться каждый канал.
nAvgBytesPerSec	Среднее количество байтов в секунду, т. е. скорость, с которой должны передаваться данные волновой формы. Используя это значение, программное обеспечение воспроизведения может оценить размер буфера.

nBlockAlign Выравнивание блока (в байтах) данных волновой формы. Программному обеспечению воспроизведения необходимо одновременно обрабатывать несколько байтов <nBlockAlign> данных, поэтому значение <nBlockAlign> может использоваться для выравнивания буфера.

Поле <format-specific-fields> содержит нуль или более байтов параметров. Содержание параметра зависит от категории формата WAVE – подробно см. в следующих разделах. Программное обеспечение воспроизведения должно допускать (и игнорировать) любые неизвестные параметры <format-specific-fields>, которые могут появиться в конце данного поля.

## 1.2 Категории формата WAVE

Категория формата файла WAVE определяется значением поля <wFormatTag> фрагмента 'fmt'. Представление данных в <wave-data> и содержимое <format-specific-fields> фрагмента 'fmt' обуславливаются категорией формата.

В настоящее время определены в том числе следующие открытые непатентованные категории формата WAVE:

wFormatTag	Значение	Категория формата
WAVE_FORMAT_PCM	(0x0001)	Формат Microsoft® (ИКМ)
WAVE_FORMAT_MPEG	(0x0050)	MPEG-1 Audio (только аудио)

ПРИМЕЧАНИЕ. – Несмотря на то что корпорацией Microsoft® зарегистрированы другие форматы WAVE, в настоящее время для BWF используются только перечисленные выше категории. Подробная информация о формате ИКМ WAVE представлена в следующем ниже разделе 2. Общая информация об остальных форматах WAVE приведена в разделе 3. Подробная информация о формате MPEG WAVE содержится в Приложении 2. В будущем могут быть определены другие форматы WAVE.

## 2 Формат ИКМ

Если поле <wFormatTag> фрагмента <fmt-ck> установлено в значение WAVE\_FORMAT\_PCM, тогда данные волновой формы содержат выборки, представляющие формат ИКМ. Для данных волновой формы ИКМ поля <format-specific-fields> определяются следующим образом:

```
<PCM-format-specific> ->
    struct{
        WORD nBitsPerSample, /* Размер выборки */
    }
```

Поле <nBitsPerSample> определяет количество битов данных, используемых для представления каждой выборки сигнала в каждом канале. Если используется несколько каналов, размер отсчета является одинаковым для всех каналов.

Содержимое поля <nBlockAlign> должно соответствовать значению, полученному согласно следующей формуле и округленному до старшей целой величины:

$$nchannels \times BytesPerSample.$$

Значение BytesPerSample должно рассчитываться путем округления значения nBitsPerSample до следующего целого байта. Если слово выборки аудиосигнала меньше целого числа байтов, старшие значащие биты выборки аудиосигнала помещаются в позиции старших значащих битов слова данных; неиспользуемые биты данных, соседние с младшим значащим битом, должны быть установлены равными нулю.

В случае данных ИКМ содержимое поля <nAvgBytesPerSec> фрагмента 'fmt' должно соответствовать значению, полученному согласно следующей формуле:

$$nSamplesPerSec \times nBblockAlign.$$

ПРИМЕЧАНИЕ 1. – Исходная спецификация WAVE допускает, например, упаковку 20-битовых выборок от двух каналов в 5 байтов, совместно использующих один байт для наименее значащих битов этих двух каналов. В настоящей Рекомендации определяется целое число байтов на выборку аудиосигнала, с тем чтобы уменьшить уровень неоднозначности в реализациях и обеспечить максимальную совместимость при обмене.

## 2.1 Упаковка данных для файлов WAVE с ИКМ

Для файлов WAVE с одним каналом выборки запоминаются последовательно. Для файлов стереозвука WAVE канал 0 представляет левый канала, а канал 1 – правый канал. В файлах WAVE с несколькими каналами выборки чередуются.

Следующий алгоритм показывает упаковку данных для файлов WAVE монофонических и стереофонических сигналов с разрядностью выборки 8 битов:

### Упаковка данных в случае 8-битовых выборок, монофонический сигнал, ИКМ

Выборка 1	Выборка 2	Выборка 3	Выборка 4
Канал 0	Канал 0	Канал 0	Канал 0

### Упаковка данных в случае 8-битовых выборок, стереофонический сигнал, ИКМ

Выборка 1		Выборка 2	
Канал 0 (левый)	Канал 1 (правый)	Канал 0 (левый)	Канал 1 (правый)

Следующие диаграммы показывают упаковку данных для файлов WAVE моно- и стереосигналов с разрядностью дискретизации 16 битов:

### Упаковка данных в случае 16-битовых выборок, монофонический сигнал, ИКМ

Выборка 1		Выборка 2	
Канал 0 младший байт	Канал 0 старший байт	Канал 0 младший байт	Канал 0 старший байт

### Упаковка данных в случае 16-битовых выборок, стереофонический сигнал, ИКМ

Выборка 1			
Канал 0 (левый) младший байт	Канал 0 (левый) старший байт	Канал 1 (правый) младший байт	Канал 1 (правый) старший байт

## 2.2 Формат данных выборок

Каждая выборка содержится в целом числе  $i$ . Размер  $i$  – это наименьшее число байтов, требуемое для размещения выборки конкретного размера. Первым запоминается наименее значащий бит. Биты, отображающие амплитуду выборки, сохраняются в наиболее значащих битах  $i$ , остальные биты устанавливаются равными нулю.

Например, если размер выборки (записанный в <nBitsPerSample>) составляет 12 битов, тогда каждая выборка хранится в состоящем из двух байтов целом числе. Четыре наименее значащих бита первого (наименее значащего) байта устанавливаются равными нулю. Формат данных и максимальное и минимальное значения для выборок волновой формы ИКМ разных размеров представлены ниже:

Размер выборки	Формат данных	Максимальное значение	Минимальное значение
От одного до восьми битов	Целое число без знака	255 (0xFF)	0
Девять и более битов	Целое число $i$ со знаком	Наибольшее положительное значение $i$	Наибольшее отрицательное значение $i$

Например, максимальное, минимальное значения и значение средней точки для данных волновой формы ИКМ с разрядностью выборки 8 и 16 битов следующие:

Формат	Максимальное значение	Минимальное значение	Значение средней точки
ИКМ, разрядность выборки 8 битов	255 (0xFF)	0	128 (0x80)
ИКМ, разрядность выборки 16 битов	32 767(0x7FFF)	-32 768(-0x8000)	0

### 2.3 Примеры файлов формата WAVE с ИКМ

**Пример** файла формата WAVE с ИКМ, частота дискретизации 11,025 кГц, монофонический звук, разрядность выборки – 8:

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

**Пример** файла формата WAVE с ИКМ, частота дискретизации 22,05 кГц, стереофонический звук, разрядность выборки – 8:

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

**Пример** файла формата WAVE с ИКМ, частота дискретизации 44,1 кГц, монофонический звук, разрядность выборки – 22:

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

### 2.4 Хранение данных WAVE

Поле <wave-data> содержит данные волновой формы. Это поле определяется следующим образом:

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

### 2.5 Фрагмент Fact

Во фрагменте <fact-ck> хранится важная информация о содержании WAVE-файла. Этот фрагмент определяется следующим образом:

```
<fact-ck> -> fact( <dwFileSize:DWORD> ) /*Количество выборок*/
```

Этот фрагмент не требуется для файлов с ИКМ.

Фрагмент `fact` будет расширяться для включения любой другой информации, необходимой для будущих форматов WAVE. Добавляемые поля будут размещаться после поля `<dwFileSize>`. Приложения могут использовать поле размера этого фрагмента для определения того, какие поля представлены.

## 2.6 Другие необязательные фрагменты

Для использования в формате WAVE определен ряд других необязательных фрагментов. Подробная информация об этих фрагментах содержится в спецификации формата WAVE и выпущенных позже обновлениях.

ПРИМЕЧАНИЕ 1. – Формат WAVE может поддерживать другие необязательные фрагменты, которые могут быть включены в WAVE-файлы для переноса конкретной информации. Как указано в Примечании 1 к п. 2.1 Приложения 1, в файлах волнового формата радиовещания они рассматриваются как частные фрагменты и игнорируются приложениями, которые не могут их интерпретировать.

## 3 Другие форматы WAVE

Следующая далее информация взята из стандартов данных Microsoft®. Она определяет формат необходимых расширений базовых файлов формата WAVE (используемых для аудиосигналов с ИКМ) для охвата других типов формата WAVE.

### 3.1 Общая информация

Все вновь определяемые типы WAVE должны содержать и фрагмент `<fact-ck>`, и описание расширенного волнового формата во фрагменте `<fmt-ck>`. Для WAVE-файлов RIFF типа WAVE\_FORMAT\_PCM не требуется ни этот дополнительный фрагмент, ни описание расширенного волнового формата.

### 3.2 Фрагмент `fact`

В этом фрагменте хранится определяемая файлом информация о содержимом WAVE-файла. В настоящее время в нем содержится определение длины файла в выборках.

### Расширение формата WAVE

Расширенная структура волнового формата, добавляемая к `<fmt-ck>`, используется для определения всех волновых данных не ИКМ-формата и описывается следующим образом. Общая структура расширенного волнового формата используется для всех не ИКМ-форматов.

```
typedef struct waveformat_extended_tag {
    WORD    wFormatTag,          /* тип формата */
    WORD    nChannels,          /* число каналов (т. е. моно, стерео...) */
    DWORD   nSamplesPerSec,     /* частота дискретизации */
    DWORD   nAvgBytesPerSec,    /* для оценки размера буфера */
    WORD    nBlockAlign,        /* размер блока данных */
    WORD    wBitsPerSample,     /* количество битов на выборку для данных монозвука */
    WORD    cbSize,             /* счет в байтах дополнительного размера */
} WAVEFORMATEX;
```

Поле	Описание
wFormatTag	Определяет тип WAVE-файла.
nChannels	Число каналов в потоке, 1 – моно, 2 – стерео.
nSamplesPerSec	Частота дискретизации волнового файла. Должна составлять 48 000 или 44 100 и т. д. Эта частота используется также записью размера выборки фрагмента <code>fact</code> для определения длительности данных.

nAvgBytesPerSec	Средняя скорость данных. Используя значение <code>&lt;nAvgBytesPerSec&gt;</code> , программное обеспечение воспроизведения может оценить размер буфера.
nBlockAlign	Выравнивание блока (в байтах) данных в <code>&lt;data-ck&gt;</code> . Программному обеспечению воспроизведения необходимо одновременно обрабатывать несколько байтов <code>&lt;nBlockAlign&gt;</code> данных, поэтому значение <code>&lt;nBlockAlign&gt;</code> может использоваться для выравнивания буфера.
wBitsPerSample	Это количество битов для представления каждой выборки в каждом канале. Принимается, что разрядность дискретизации в каждом канале одинаковая. Если данное поле не требуется, следует установить его значение равным нулю.
cbSize	Размер в байтах дополнительной информации в заголовке формата WAVE, не включая размер структуры WAVEFORMATEX.

ПРИМЕЧАНИЕ. – Поля, следующие за полем `<cbSize>`, содержат специальную информацию, необходимую для формата WAVE, который определен в поле `<wFormatTag>`. Все форматы WAVE, которые могут использоваться в BWF, будут описаны в отдельных Добавлениях к настоящей Рекомендации.

## Дополнение 2 к Приложению 1 (информационное)

### Спецификация формата поля `<CodingHistory>`

#### Введение

Поле данных о применении кодирования `<CodingHistory>` фрагмента `<hex>` определяется как набор цепочек, содержащих хронологию процессов кодирования. При изменении данных о применении кодирования должна добавляться новая строка. Каждая строка должна содержать строковую переменную для каждого параметра кодирования. Все строки должны заканчиваться символом CR/LF. Формат строк данных о применении кодирования представлен ниже.

#### Синтаксис

Все строки должны соответствовать следующему синтаксису:

Параметр	Переменная цепочка <допустимый вариант>
Алгоритм кодирования	A=<ANALOGUE, PCM, MPEG1L1, MPEG1L2, MPEG1L3, MPEG2L1, MPEG2L2, MPEG2L3>
Частота дискретизации (Гц)	F=<16000,22050,24000,32000,44100,48000>
Битовая скорость (кбит/с на канал)	V=<любая битовая скорость, разрешенная в MPEG 2 (ISO/IEC 13818-3)>
Длина слова (биты)	W=<8, 12, 14, 16, 18, 20, 22, 24>
Режим	M=<моно, стерео, двоянный моно, квазистерео>
Текст, произвольные цепочки	T=<произвольная цепочка ASCII-текста для внутреннего использования. Эта цепочка не должна содержать запятых (ASCII 2C <sub>hex</sub> ). Пример содержимого цепочки: ID-No; тип кодека; тип А/Ц преобразователя>

Цепочки переменных должны разделяться запятой (ASCII 2C<sub>hex</sub>). Каждая строка должна завершаться символом CR/LF.

Переменная V= используется только для кодирования типа MPEG.

Переменная W= В случае кодирования типа MPEG должна использоваться для указания длины слова ИКМ-входа в MPEG-кодер.

## Примеры полей данных о применении кодирования

### Пример 1

A=PCM,F=48000,W=16,M=stereo,T=original,CR/LF

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

### Интерпретация примера 1

#### Строка 1

Исходный файл записывается как линейный файл типа BWF с кодированием типа ИКМ со следующими параметрами:

- Частота дискретизации: 48 кГц
- Разрядность кодирования: 16 битов в выборке
- Режим: стерео
- Статус: исходный код

#### Строка 2

Исходный файл конвертируется в файл типа MPEG-1 Layer II BWF с использованием следующих параметров:

- Частота дискретизации: 48 кГц
- Битов в секунду в каждом канале: 192 кбит/с
- Разрядность кодирования: 16 битов
- Режим: стерео
- Кодер: PCX9 (Digigram)

### Пример 2 для процесса преобразования аналогового материала в цифровую форму

A=ANALOGUE,M=stereo,T=StuderA816; SN1007; 38; Agfa\_PER528,<CR/LF>

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

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

### Интерпретация примера 2

#### Строка 1

Было осуществлено воспроизведение записанного на магнитной пленке типа Agfa PER528 аналогового материала на магнитофоне Studer, модель A816, серийный номер 1007:

- Скорость протяжки ленты: 38 см/с
- Режим: стерео

#### Строка 2

Запись была преобразована в цифровую форму с использованием А/Ц преобразователя типа NVision NV1000 со следующими параметрами:

- Частота дискретизации: 48 кГц
- Разрядность кодирования: 18 битов в каждой выборке
- Режим: стерео

#### Строка 3

Запись сохранена в виде файла BWF с линейным ИКМ-кодированием при использовании цифрового входа интерфейсной карты PCX9 со следующими параметрами:

- Частота дискретизации: 48 кГц
- Разрядность кодирования: 16 битов в каждой выборке
- Режим: стерео

### Дополнение 3 к Приложению 1 (информационное)

#### Определение формата "уникального" идентификатора источника (USID) для использования в поле <OriginatorReference>

#### USID

Идентификатор USID в поле создателя оригинала <OriginatorReference> генерируется при использовании нескольких независимых источников рандомизации в целях обеспечения его уникальности в отсутствие единого органа по распределению. Эффективность и простота использования метода рандомизации достигается путем сочетания конкретной информации о пользователе, машине и времени и случайного числа. Такими элементами являются следующие:

CC	Код страны: (2 символа) на основе стандарта ISO 3166 <sup>5</sup> [ISO, 1997].
OOOO	Код организации: 4 символа.
NNNNNNNNNNNN	Серийный номер: (12 символов, соответствующих модели и серийному номеру магнитофона). Он должен указывать тип и серийный номер аппарата.
HHMMSS	Время создания: (6 символов). Из поля <OriginationTime> формата BWF.

Этих элементов должно быть достаточно для указания конкретной записи в "пригодной для человека" форме в сочетании с другими источниками информации, формальными и неформальными.

Кроме того, USID содержит:

RRRRRRRR	Случайное число (8 символов), генерируемое локально магнитофоном с использованием приемлемого алгоритма случайных чисел.
----------	--

Данный элемент служит для отдельного указания файлов, таких как стерео каналы или дорожки в многодорожечных записях, созданных в одно время.

#### Примеры идентификаторов USID

##### Пример 1

USID, созданный магнитофоном Tascam DA88, серийный номер 396FG347A, под управлением RAI, телерадиокомпании Италии, время создания: 12:53:24

Формат UDI : CCOOOONNNNNNNNNNNNNHHMMSSRRRRRRRR

Пример UDI : ITRAI0DA88396FG34712532498748726

##### Пример 2

USID, созданный аппаратом xxxxxxx, серийный номер ssssssss, под управлением YLE, вещательной компании Финляндии, время создания: 08:14:48

Формат UDI : CCOOOONNNNNNNNNNNNNHHMMSSRRRRRRRR

Пример UDI : FIYLE0xxxxxxxsssss08144887724864

---

<sup>5</sup> ISO 3166-1:1997 Codes for the representation of names of countries and their subdivisions –Part 1: Country codes (см.: <http://www.din.de/gremien/nas/nabd/iso3166ma/index.html>).



## Дополнение 4 к Приложению 1 (информационное)

### Определение необязательного фрагмента огибающей пиков Level <levl-ck> для BWF

При обмене звуковыми файлами между рабочими станциями наличие в файле данных о пиковых уровнях звуковых сигналов может ускорить открытие, отображение и обработку файла. Добавление фрагмента <levl> к файлу в волновом формате радиовещания Broadcast Wave Format (BWF) [1] обеспечивает стандарт хранения и передачи данных о пиковых уровнях сигнала, полученных путем субдискретизации звуковых сигналов. Эти данные фрагмента могут использоваться для создания огибающей звукоядра в файле. Это позволит звуковому приложению быстро отображать файлы без значительных потерь точности.

Кроме того, существует возможность передачи данных о пике пиков, являющимся *первой* выборкой звукового сигнала, абсолютное значение которого является максимальным значением всего звукового файла. Звуковое приложение может использовать эту информацию для нормализации файла в реальном времени без необходимости сканирования всего файла (поскольку это уже было сделано отправителем).

#### 1 Терминология

Звуковой сигнал подразделяется на блоки. Для каждого звукового блока генерируется один **кадр пиков**. Для каждого кадра пиков существует *n* **пиковых значений**, где *n* – число каналов с пиками. Каждое пиковое значение может содержать одну (только положительную) или две (одну положительную и одну отрицательную) **пиковые точки**.

##### 1.1 Генерирование пиковых значений

Звуковой сигнал подразделяется на блоки выборок постоянного размера. Размер одного блока по умолчанию, и рекомендуемый, составляет 256 выборок из каждого канала.

Выборки каждого канала оцениваются для нахождения пиковых точек (максимальных значений). Рекомендуется, чтобы определялись отдельные пиковые точки для положительных и отрицательных выборок, в противном случае может использоваться только абсолютное значение (либо положительное, либо отрицательное). Все пиковые точки являются значениями без знака.

Пиковые точки округляются до одного из двух форматов – 8 или 16 битов. В большинстве случаев достаточным является 8-битовый формат. 16-битовый формат должен охватывать любые случаи, когда необходима более высокая точность.

Форматированные пиковые точки для каждого канала собираются в кадры пиков. Каждый кадр пиков содержит положительную и отрицательную пиковые точки (или абсолютную пиковую точку) для каждого канала, сохраняя порядок следования выборок звукового сигнала.

Эти кадры пиков переносятся как данные во фрагменте огибающей пиков Peak Envelope. Фрагмент огибающей пиков начинается с заголовка, в котором содержится информация, позволяющая интерпретировать данные о пиках.

**Пик пиков** является *первой* выборкой звукового сигнала, абсолютное значение которой является максимальным значением всего звукового файла. Вместо сохранения пика пиков как значения выборки, сохраняется *позиция* пика пиков. Другими словами, сохраняется индекс кадра выборки звукового сигнала. Приложение "понимает", где считывать пик пиков в звуковом файле. Было бы значительно сложнее сохранять пиковое значение, поскольку оно зависит от двоичного формата выборок звукового сигнала (целые, с плавающей запятой, двойные...).

## ПРИМЕЧАНИЯ:

- В заголовке используются только DWORD (4-байтовые значения) или несколько 4-байтовых значений во избежание сложностей выравнивания структуры в разных компиляторах.
- Общий размер заголовка составляет 128 байтов, с тем чтобы избежать смещения кэшей.

## 2 Фрагмент огибающей пиков Peak envelope

Фрагмент огибающей пиков, *<levl>*, состоит из заголовка, за которым следуют данные о пиковых точках. Общая длина фрагмента должна быть переменной в зависимости от звукового контента, размера блоков и метода форматирования данных о пиках.

```
typedef struct peak_envelope
{
    CHAR    ckID[4],           /* {'l','e','v','l'} */
    DWORD   ckSize,           /* размер фрагмента */
    DWORD   dwVersion,       /* информация о версии */
    DWORD   dwFormat,;       /* формат пиковой точки */
                                1 = символ без знака
                                2 = короткий формат без знака
    DWORD   dwPointsPerValue, /* 1 = только положительная пиковая точка
                                2 = положительная И отрицательная пиковые точки */
    DWORD   dwBlockSize,     /* кадров для значения */
    DWORD   dwPeakChannels,  /* число каналов */
    DWORD   dwNumPeakFrames, /* число кадров пиков */
    DWORD   dwPosPeakOfPeaks, /* индекс кадра выборки звукового сигнала */ или
                                0xFFFFFFFF, если неизвестен*/
    DWORD   dwOffsetToPeaks, /* всегда должно быть равным размеру этого заголовка,
но
                                Может быть выше */
    CHAR    strTimestamp[28], /* ASCII: временная отметка данных пика */

```

### 2.1 Элементы фрагмента "levl"

- ckID** Это 4-символьное матричное {"l", "e", "v", "l"}<sup>6</sup>, обозначение фрагмента.
- ckSize** Размер оставшейся части фрагмента (не включает 8 байтов, используемых элементами ckID и ckSize.)
- dwVersion** Версия фрагмента peak\_envelope. Начинается с 0000.
- dwFormat** Формат данных огибающей пиков. Допустимы два формата<sup>7</sup>:

dwFormat	Значение	Описание
LEVL_FORMAT_UINT8	1	символ без знака для каждой пиковой точки
LEVL_FORMAT_UINT16	2	короткое целое без знака для каждой пиковой точки

- dwPointsPerValue** Обозначает количество пиковых точек в каждом пиковом значении. Значением этого элемента может быть 1 или 2.

<sup>6</sup> Определение DWORD ckID = "levl" необязательно уникально. Разные С-компиляторы образуют разные порядки следования символов. Поэтому взамен мы определяем символ ckID[4] = {"l", "e", "v", "l"}.

<sup>7</sup> Поскольку все звуковые приложения, которые поддерживают фрагмент "levl", должны реализовывать все возможные форматы, разрешаются только два формата.

В большинстве случаев достаточно формата символа без знака (8 битов). Короткий формат без знака (16 битов) должен охватывать все случаи, когда требуется более высокая точность.

**dwPointsPerValue = 1**

Каждое пиковое значение содержит одну пиковую точку. Пиковая точка является максимумом абсолютных значений выборки звукового сигнала **dwBlockSize** в каждом блоке:

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

ПРИМЕЧАНИЕ. – В данном случае отображаемая волновая форма всегда будет симметричной относительно горизонтальной оси.

**dwPointsPerValue = 2**

Каждое пиковое значение содержит две пиковые точки. Первая пиковая точка соответствует наивысшему *положительному* значению выборки звукового сигнала **dwBlockSize** в блоке. Вторая пиковая точка соответствует *отрицательному* пику выборки звукового сигнала **dwBlockSize** в блоке.

Рекомендуется использовать две пиковые точки (**dwPointsPerValue = 2**), поскольку в этом случае будут правильно отображаться несимметричные волновые формы (например, смещение по постоянному току).

**dwBlockSize** Это число выборок звукового сигнала, используемое для генерирования каждого кадра пиков. Это число переменное. Значение по умолчанию, и рекомендуемое, составляет 256.

**dwPeakChannels** Число каналов пиков<sup>8</sup>.

**dwNumPeakFrames** Число кадров пиков. Числом кадров пиков является целая величина, полученная путем округления в меньшую сторону результата расчета по следующей формуле:

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

или путем округления в большую сторону результата следующего расчета:

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

где numAudioFrame – это число выборок звукового сигнала в каждом канале звуковых данных.

Например, для отношения пиков (размер блока), равного 256, это означает:

0	выборки звукового сигнала	-> 0 кадров пиков
1	выборка звукового сигнала	-> 1 кадр пиков
256	выборки звукового сигнала	-> 1 кадр пиков
257	выборки звукового сигнала	-> 2 кадра пиков
7 582	выборки звукового сигнала	-> 30 кадров пиков.

**dwPosPeakOfPeaks** Звуковое приложение может использовать эту информацию для нормализации файла без необходимости сканирования всего файла (поскольку это уже было сделано отправителем). Преимущество заключается в увеличении производительности, а также в возможности нормализации файла в реальном времени.

**Пик пиков** – это *первая* выборка звукового сигнала, абсолютное значение которой является максимальным значением всего звукового файла.

Вместо сохранения пика пиков как значения выборки, сохраняется *позиция* пика пиков. Другими словами, сохраняется индекс кадра выборки звукового сигнала. Приложение "понимает", где

<sup>8</sup> Обычно число каналов пиков равняется числу звуковых каналов. Если это число равно единице, для каждого звукового канала будет отображаться та же волновая форма.

считывать пик пиков в звуковом файле. Было бы значительно сложнее сохранять пиковое значение, поскольку оно зависит от двоичного формата выборок звукового сигнала (целые, с плавающей запятой, двойные).

Если значение составляет 0xFFFFFFFF, это означает, что пик пиков неизвестен.

**dwOffsetToPeaks** Смещение данных о пике относительно начала заголовка. Обычно оно равняется размеру заголовка, но может быть и выше. Это значение может использоваться для обеспечения того, что данные о пике начинаются на границе DWORD.

**strTimeStamp** Строка, содержащая отметку времени создания данных о пиках. Формат строки следующий<sup>9</sup>:

"ГГГГ:ММ:ДД:чч:мм:сс:мсмсмс",

где:

ГГГГ: год  
 ММ: месяц  
 ДД: день  
 чч: часы  
 мм: минуты  
 сс: секунды  
 мсмсмс: миллисекунды

Например: "2000:08:24:13:55:40:967"

## 2.2 Формат пиковой точки

Пиковое значение состоит из одной или двух точек, имеющих признак **dwPointsPerValue**. Признак **dwFormat** указывает формат чисел, представляющих пиковые точки в каждом кадре пиков.

		dwPointsPerValue	
		= 1	= 2
<b>dwFormat</b>		Число соответствует абсолютному пику	Первое число соответствует положительному пику Второе число соответствует отрицательному пику (Заметим, что "отрицательный" пик хранится как "положительное" число)
= 1	levl_format_uint8	символ без знака (0...255)	символ без знака (0...255) символ без знака (0...255)
= 2	levl_format_uint16	короткое целое без знака (0...65535)	короткое целое без знака (0...65535) короткое целое без знака (0...65535)

## 2.3 Файлы многоканальных пиков

В случае многоканальных звуковых файлов происходит перемежение отдельных пиковых значений по каждому каналу. Набор перемежающихся пиковых значений называется кадром пиков. Порядок следования пиковых значений внутри кадра пиков соответствует размещению точек выборок внутри кадра звуковых данных RIFF.

<sup>9</sup> Преимуществом этого формата является отсутствие ограничений по времени и его удобочитаемость. (В других форматах используется элемент DWORD, означающий секунды с 1970 года, предел которого будет достигнут примерно через 125 лет.)

## 2.4 Синхронизация со звуковым файлом

Файл пиков должен быть создан заново, если существует любое из следующих условий:

Отметка времени старше отметки времени звукового файла.

Число кадров пиков не соответствует числу кадров выборок в звуковом файле.

## 2.5 Порядок следования байтов

Поскольку файл волнового формата радиовещания Broadcast Wave Format file (BWF) – это расширение формата RIFF, числа сохраняются в формате следования байтов, начиная с младшего.

# Дополнение 5 к Приложению 1 (информационное)

## Определение необязательного фрагмента <link-ck> для BWF

### Введение

Файлы волнового формата радиовещания Broadcast Wave Format (BWF) допускают максимальный размер файла, равный 4 Гбайтам, хотя на практике многие приложения RIFF/Wave поддерживают максимальный размер файлов, составляющий только 2 Гбайта. В случае звуковых данных, превышающих эти пределы, необходимо разделить звуковую информацию на несколько файлов BWF. Фрагмент <link> обеспечивает данные для соединения, что позволяет осуществлять бесшовное распределение звуковых выходных данных по нескольким файлам.

### 1 Терминология

<b>Набор файлов</b>	Набор связанных файлов, относящихся к одному непрерывному звуковому сигналу.
<b>Имя файла</b>	Имена, дающиеся каждому файлу в наборе файлов.
<b>Список файлов</b>	Список имен файлов в наборе файлов.
<b>Атрибут "actual" (фактический)</b>	Атрибут, с помощью которого в списке файлов маркируется имя файла, являющего текущим (или "фактическим"). Все прочие имена файлов в списке маркируются признаком "other" (другой).
<b>Идентификатор файла</b>	Необязательный идентификатор, который должен быть одинаковым для всех файлов в наборе файлов.
<b>Элемент "private" (частный)</b>	Дополнительный элемент во фрагменте для хранения в списке файлов информации, являющейся собственностью фирмы.
<b>Фрагмент &lt;link&gt; (соединение)</b>	Фрагмент, содержащийся во всех файлах набора файлов. Он содержит заголовок, за которым следует список файлов и необязательный идентификатор файлов, а также элемент "private". Данные фрагмента хранятся в формате XML 1.0 <sup>10</sup> – широко распространенном формате для обмена данными.

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

## 2 Структура фрагмента соединения Link

### 2.1 Обзор

Фрагмент соединения <link> содержит заголовок, за которым следует информация для соединения, хранящаяся в формате XML (расширенный язык разметки). Общая длина фрагмента переменная.

```
typedef struct link
{CHAR    ckID[4],      /* {'l','i','n','k'} */
  DWORD   ckSize,     /* размер фрагмента */
  CHAR    XmlData[ ], /* информация для соединения в формате XML */
}
Link_chunk,
```

<b>Поле</b>	<b>Description</b>	ckID	Это 4-символьное матричное {'l', 'i', 'n', 'k'} <sup>11</sup> обозначение фрагмента.
<b>CkSize</b>			Размер секции данных фрагмента (не включает 8 байтов, используемых элементами ckID и ckSize.)
<b>XmlData</b>			Этот буфер содержит информацию для соединения в формате XML (ASCII-символы).

### 2.2 Структура данных XML в поле переменных данных <xmlData>

Структура данных имеет иерархический характер. Данные хранятся в текстовых строках. Для точного описания синтаксиса добавляется DTD (документ передачи данных).

```
<LINK>
  <FILE type="...">
    <FILENUMBER>...</FILENUMBER>
    <FILENAME>...</FILENAME>
  </FILE>
  .....
  Возможные будущие элементы FILE
  .....
  <ID>...</ID>      необязательный
  <PRIVATE>         необязательный
  ..... зависит от реализации
  </PRIVATE>
</LINK>
```

<b>LINK</b>	Это корневой элемент данных XML. LINK содержит один или более элементов FILE с описанием файла. Он может также содержать идентификатор ID и/или элемент PRIVATE.
<b>ID</b>	Идентификатор ID является общим для всех файлов в данном наборе файлов. Он хранится в виде текстовой строки символов, соответствующей определению #PCDATA спецификации XML 1.0, которая включает все видимые ASCII-символы, пробелы и т. д.
<b>PRIVATE</b>	Элемент PRIVATE может содержать информацию, определяемую реализацией и включающую любые данные XML (например, дальнейшие элементы или #PCDATA).
<b>FILE</b>	Элемент FILE содержит элемент FILENUMBER и элемент FILENAME. Атрибут типа должен быть 'actual' (фактический), в случае если файл в списке описывает файл, которому принадлежит данный фрагмент. Все остальные файлы должны

<sup>11</sup> Определение DWORD ckID = "link" необязательно уникально. Разные С-компиляторы образуют разные порядки следования символов. Поэтому взамен мы определяем символ ckID[4] = {'l', 'i', 'n', 'k'}.

иметь атрибут типа "other" (другой). Имя файла данного файла должно быть тем же, что и указанное в списке файлов.

**FILENUMBER** Файлы должны нумероваться последовательно в соответствии с их хронологическим порядком в наборе файлов. Должны использоваться целые числа (ASCII-символы), начиная с числа 1.

**FILENAME** Текстовая строка, хранящаяся в том же формате, что и ID.

### 2.3 DTD для структуры XML фрагмента <link>

DTD (определение типа документа) определяется в спецификации XML 1.0 как описание синтаксиса структуры XML. Ниже описаны формат и атрибуты разных элементов фрагмента <link>, включая субэлементы и их многообразие.

Элемент LINK должен содержать один или более субэлементов FILE ('+' указывает, содержится один или более), он может содержать субэлемент ID и субэлемент PRIVATE ('?' указывает, содержится один или ни одного).

Каждый элемент FILE должен содержать один субэлемент FILENUMBER и один субэлемент FILENAME. Должен быть определен атрибут типа, которым может быть "actual" (фактический) или "other" (другой).

Субэлементы FILENUMBER, FILENAME и ID должны содержать текстовые строки (в XML называемые #PCDATA).

Субэлемент PRIVATE может содержать любые определенные элементы. Если в PRIVATE необходимо ввести элементы, отличные от определенных, следует соответствующим образом изменить DTD.

<!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 Переименование связанных файлов

Если меняется одно или более имен файлов, должны изменяться соответствующие записи FILENAME в каждом фрагменте <link>, принадлежащем всему набору файлов.

Непрерывный звуковой сигнал в данном примере был разбит на образующие набор файлов три файла BWF с именами "Sinatra\_1.wav", "Sinatra\_2.wav" и "Sinatra\_3.wav". Структуры XML фрагментов <link> этих трех файлов идентичны, за исключением атрибута типа.

### 3.1 Фрагмент <link> файла "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 Фрагмент `<link>` файла "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>
</ID>73365869</ID>
</LINK>

```

3.3 Фрагмент `<link>` файла "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>
</ID>73365869</ID>
</LINK>

```

## Дополнение 6 к Приложению 1 (информационное)

### Соглашения об именах файлов

#### 1 Общее

Общий обмен звуковыми файлами означает, что должна обеспечиваться возможность их воспроизведения на компьютере и в операционной системе, типы которых могут весьма отличаться от системы, где был изготовлен оригинал. Ненадлежащее имя файла может означать невозможность его распознавания системой назначения. Например, некоторые операционные системы ограничивают число символов в имени файла. Другие не могут обрабатывать многобайтовые символы. Некоторые символы в ряде операционных систем имеют специальное значение и их использование не допустимо. Целью настоящих руководящих принципов является определение передового опыта для общего международного обмена.

#### 2 Длина имени файла

Длина имени файла формата BWF не должна превышать 31 символа, включая расширение имени файла.



### 3 Расширение имени файла

Файлы BWF должны иметь то же четырехсимвольное расширение имени файла – ".wav", что и традиционный файл формата WAVE. Это обеспечит возможность воспроизведения звукового контента на большинстве компьютеров без использования дополнительного программного обеспечения. Практические реализации должны также допускать и другие расширения, такие как ".bwf", которые могут быть следствием ошибки.

### 4 Набор символов для имени файла

В именах файлов для международного обмена должны использоваться только 7-битовые ASCII (ISO/IEC 646) символы в диапазоне от 32 до 126 (десятичные).

Символ	Десятичное значение	Шестнадцатеричное значение
(пробел)	32	0x20
...	...	...
~ (тильда)	126	0x7E

Кроме того, следующие символы зарезервированы в некоторых файловых системах для специальных функций и не должны использоваться в именах файлов:

Символ	Десятичное значение	Шестнадцатеричное значение
“	34	0x22
*	42	0x2A
/	47	0x2F
:	58	0x3A
<	60	0x3C
>	62	0x3E
?	63	0x3F
\	92	0x5C
	124	0x7C

Кроме того, следующие символы не должны использоваться в качестве первого или последнего символа в имени файла:

Символ	Десятичное значение	Шестнадцатеричное значение
(пробел)	32	0x20
(точка)	46	0x2E

## Приложение 2

### Спецификация волнового формата радиовещания с MPEG-1 аудио

#### Формат для файлов аудиоданных в радиовещании

#### 1 Введение

В настоящем Приложении содержится спецификация применения BWF для переноса только звуковых сигналов в формате MPEG. В случае MPEG аудио к основным фрагментам, определенным в основной части настоящей Рекомендации, необходимо добавлять следующую информацию:

- расширение фрагмента format;
- фрагмент fact;
- фрагмент MPEG\_extension.

Расширения фрагмента format и фрагмента fact определены как часть формата WAVE, и соответствующая информация представлена в Дополнении 1 к Приложению 2.

Спецификация фрагмента MPEG\_extension представлена в п. 2 Приложения 2.

Основная часть настоящей Рекомендации содержит спецификацию фрагмента расширения аудиоданных радиовещания, который используется во всех BWF. Информация о базовом формате RIFF представлена в Дополнении 1 к Приложению 2.

#### 2 MPEG аудио

Корпорация Microsoft® определила порядок организации аудиоданных MPEG в файлы WAVE. Расширения фрагмента format и фрагмент fact переносят дополнительную информацию, необходимую для описания опций кодирования MPEG. Общие принципы изложены в Дополнении 1 к Приложению 1, и подробная информация содержится в Дополнении 1 к Приложению 2. В случае формата MPEG Layer II было обнаружено, что необходима дополнительная информация для переноса данных о кодировании сигнала. Такие данные содержатся во фрагменте расширения MPEG аудио <MPEG Audio Extension>, разработанном специальной группой по вопросам MPEG Layer 2 Audio. Описание данного фрагмента представлено ниже.

##### 2.1 Фрагмент расширения MPEG аудио

Фрагмент расширения MPEG аудио определяется следующим образом:

```
typedef struct {
    DWORD ckID,          /* (mpeg_extension)ckID='mext' */
    DWORD ckSize,       /* размер фрагмента расширения:
                        cksize =000C*/
    BYTE  ckData[ckSize], /* данные фрагмента */
}

typedef struct mpeg_audio_extension {
    WORD SoundInformtion, /* дополнительная информация о звуке */
    WORD FrameSize,      /* номинальный размер кадра */
    WORD AncillaryDataLength, /* Длина вспомогательных данных */
    WORD AncillaryDataDef, /* Тип вспомогательных данных */
    CHAR Reserved [4],    "NULL"*/
} MPEG_EXT ;
```

<b>Поле</b>	<b>Описание</b>
SoundInformation	<p>16 битов, содержащих дополнительную информацию о звуковом файле:</p> <p>Для MPEG Layer II (или Layer I):</p> <p>Бит 0: '1' Однородные звуковые данные  '0' Неоднородные звуковые данные.</p> <p>Биты 1 и 2 используются для дополнительной информации в случае однородных звуковых файлов:</p> <p>Бит 1: '0' Бит-заполнитель используется в файле, поэтому может менять значения между '0' и '1'  '1' Бит-заполнитель устанавливается равным '0' во всем файле</p> <p>Бит 2: '1' Файл содержит последовательность кадров с битом-заполнителем, установленным равным '0', и частотой дискретизации, равной 22,05 или 44,1 кГц.</p> <p>Бит 3: '1' Используется произвольный формат  '0' Аудиокадр не произвольного формата.</p>
FrameSize	<p>16-битовое число байтов номинального кадра.</p> <p>Это поле имеет значение только для однородных файлов, иначе оно устанавливается равным '0'.</p> <p>Если бит-заполнитель не используется, т. е. он остается постоянным во всех кадрах звукового файла, тогда поле &lt;FrameSize&gt; содержит то же значение, что и поле &lt;nBlockAlign&gt; фрагмента формата. Если бит-заполнитель используется и звуковые данные имеют переменную длину, поле &lt;FrameSize&gt; содержит размер кадра с битом-заполнителем, установленным равным '0'. Длина кадра с битом-заполнителем, установленным равным '1', ровно на один байт больше (четыре байта для Layer I), т. е. &lt;FrameSize+1&gt;.</p> <p>Установка значения &lt;nBlockAlign&gt; равным '1' означает переменную длину кадров (FrameSize и FrameSize+1) с переменным битом-заполнителем.</p>
AncillaryDataLength	<p>16-битовое число, обозначающее минимальное число известных байтов для вспомогательных данных во всем звуковом файле. Это значение указывается с конца аудиокадра.</p>
AncillaryDataDef	<p>Это 16-битовое число определяет содержимое вспомогательных данных следующим образом:</p> <p>бит 0 установлен равным '1': во вспомогательных данных содержатся данные о мощности левого канала</p> <p>бит 1 установлен равным '1': байт "частный", доступен для внутреннего использования во вспомогательных данных</p> <p>бит 2 установлен равным '1': во вспомогательных данных содержатся данные о мощности левого канала</p> <p>бит 3 установлен равным '0': зарезервирован для будущего использования для данных ADR</p> <p>бит 4 установлен равным '0': зарезервирован для будущего использования для данных DAB</p>

ПРИМЕЧАНИЕ 1. – Такой файл не соответствует стандарту MPEG (п. 2.4.2.3, определение бита-заполнителя), но может рассматриваться как частный случай переменной битовой скорости. Нет необходимости, чтобы декодер MPEG декодировал такой битовый поток, поскольку большинство декодеров должны выполнять эту функцию. Битовая скорость должна быть немного ниже указанной в заголовке.

- бит 5 установлен равным '0': зарезервирован для будущего использования для данных J 52
- биты с 6 по 15 установлены равными '0': зарезервированы для будущего использования.

#### ПРИМЕЧАНИЯ:

- Элементы, представленные во вспомогательных данных, следуют в том же порядке, что и биты в AncillaryDataDef. Первый элемент сохраняется в конце вспомогательных данных, второй сохраняется перед первым и т. д., двигаясь от конца к началу.
- В случае файла монозвука бит 2 всегда устанавливается равным '0', а бит 0 относится к мощности кадра монозвука.
- В случае файла стереозвука, если бит 2 равен '0' и бит 0 равен '1', мощность относится к максимальной мощности левого и правого канала.
- Мощность хранится в 2 байтах и соответствует абсолютному значению максимальной выборки, используемой для кодирования кадра. Это 15-битовое значение в формате с порядком следования байтов, начиная со старшего.

Reserved 4 байта зарезервированы для будущего использования. Эти 4 байта должны быть установлены равными нулю. В любом будущем использовании нулевое значение должно будет использоваться для значения по умолчанию в целях поддержания совместимости.

## Дополнение 1 к Приложению 2 (информационное)

### Формат файлов RIFF WAVE (.WAV)

В настоящем Дополнении представлена спецификация дополнительной информации, необходимой для файлов WAVE, содержащих MPEG Audio.

Представленная в настоящем Дополнении информация взята из документов, в которых содержатся спецификации формата файлов RIFF Microsoft®. Эта информация приводится только для информации.

#### 1 MPEG-1 аудио (только аудио)

##### 1.1 Фрагмент Fact

Этот фрагмент требуется для всех форматов WAVE, отличных от WAVE\_FORMAT\_PCM. В нем сохраняется определяемая файлом информация о содержимом данных WAVE. На данный момент он определяет длительность по времени данных в выборках.

ПРИМЕЧАНИЕ 1. – См. также п. 2.5 Дополнения 1 к Приложению 1.

##### 1.2 Заголовок формата WAVE

```
#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;
```

<b>Поле</b>	<b>Описание</b>
wFormatTag	Это поле должно быть установлено равным WAVE_FORMAT_MPEG. [0x0050]
nChannels	Число каналов в потоке, 1 – для моно, 2 – для стерео.
nSamplesPerSec	Частота дискретизации (Гц) волнового файла: 32 000, 44 100 или 48 000 и т. д. Следует заметить, что если частота дискретизации данных является переменной, это поле должно быть установлено равным нулю. Настоятельно рекомендуется для настольных приложений использовать постоянную частоту дискретизации.
nAvgBytesPerSec	Средняя скорость данных; это может не соответствовать допустимой битовой скорости MPEG, если используется кодирование с переменной битовой скоростью в режиме Layer III.
nBlockAlign	<p>Выравнивание блока (в байтах) данных в &lt;data-ck&gt;. Для аудиопотоков, имеющих фиксированную длину аудиокадра, выравнивание блока равняется длине кадра. Для потоков с переменной длиной кадров значение &lt;nBlockAlign&gt; должно быть установлено равным 1.</p> <p>При частоте дискретизации 32 или 48 кГц размер аудиокадра MPEG является функцией битовой скорости. Если аудиопоток имеет постоянную битовую скорость, размер аудиокадров не меняется. Следовательно, применяются следующие формулы:</p> <p>Layer I: <math>nBlockAlign = 4 * (int)(12 * BitRate / SamplingFreq)</math></p> <p>Layers II и III: <math>nBlockAlign = (int)(144 * BitRate / SamplingFreq)</math></p> <p>Пример 1: Для формата Layer I с частотой дискретизации 32 000 Гц и битовой скоростью 256 кбит/с поле nBlockAlign = 384 байта.</p> <p>Если аудиопоток содержит кадры с разной битовой скоростью, тогда длина кадров в пределах потока меняется. Переменная длина кадров также применяется при использовании частоты дискретизации 44,1 кГц: в целях поддержания номинального значения скорости передачи данных размер аудиокадра MPEG периодически увеличивается на один “интервал” (4 байта для Layer I, 1 байт для Layer II и III) по сравнению с приведенными выше формулами. В этих двух случаях концепция выравнивания блока не применима. Следовательно, значение &lt;nBlockAlign&gt; должно быть установлено равным 1, так чтобы поддерживающие MPEG приложения могли сообщать, являются ли данные выровненными блоками.</p>
	<p>ПРИМЕЧАНИЕ. – Возможно построить аудиопоток, имеющий аудиокадры постоянной длины при частоте 44,1 кГц, путем установки бита-заполнителя в каждом заголовке аудиокадра, равным одному значению (либо 0 или 1). Следует отметить, вместе с тем, что битовая скорость результирующего потока не будет точно соответствовать номинальному значению в заголовке кадра, и, следовательно, некоторые декодеры могут не иметь возможности правильно декодировать этот поток. В интересах стандартизации и совместимости этот подход не рекомендуется.</p>
WBitsPerSample	Не используется; устанавливается равным нулю.
CbSize	Размер в байтах расширенной информации после структуры WAVEFORMATEX. В случае формата стандарта WAVE_FORMAT_MPEG этот размер равен 22 (0x0016). Если вставляются дополнительные поля, это значение должно увеличиваться.
fwHeadLayer	<p>Аудиослой MPEG, определяемый следующими флагами:</p> <p>ACM_MPEG_LAYER1 – Layer I.</p> <p>ACM_MPEG_LAYER2 – Layer II.</p> <p>ACM_MPEG_LAYER3 – Layer III.</p>

Некоторые допустимые потоки MPEG могут содержать кадры разных слоев. В этом случае вышеуказанные флаги должны применяться вместе, объединенные по схеме ИЛИ, так чтобы драйвер мог определить, какие слои присутствуют в данном потоке.

dwHeadBitrate

Битовая скорость данных в битах в секунду. Это значение должно соответствовать стандартной битовой скорости согласно спецификации MPEG; не все битовые скорости действительны для всех режимов и слоев, см. таблицы 1 и 2. Следует заметить, что в этом поле записана фактическая битовая скорость, а не код заголовка кадра MPEG. Если битовая скорость является переменной или нестандартной, тогда это поле должно быть установлено равным нулю. Рекомендуется по возможности избегать кодирования с переменной битовой скоростью.

fwHeadMode

Режим потока, определяемый следующими флагами:

ACM\_MPEG\_STEREO – стерео.

ACM\_MPEG\_JOINTSTEREO – квазистерео.

ACM\_MPEG\_DUALCHANNEL – сдвоенный канал (например, двуязычный поток).

ACM\_MPEG\_SINGLECHANNEL – один канал.

Некоторые допустимые потоки MPEG могут содержать кадры разных слоев. В этом случае вышеуказанные флаги должны применяться вместе, объединенные по схеме ИЛИ, так чтобы драйвер мог определить, какие слои присутствуют в данном потоке. Такая ситуация особенно вероятна в случае кодирования квазистерео, так как для кодеров может оказаться целесообразным динамическое переключение между стерео и квазистерео в соответствии с характеристиками сигнала. В этом случае должны быть установлены оба флага – ACM\_MPEG\_STEREO и ACM\_MPEG\_JOINTSTEREO.

fwHeadModeExt

Содержит дополнительные параметры для кодирования квазистерео; не используется для других режимов. См. таблицу 3. Некоторые допустимые потоки MPEG могут содержать кадры с разными расширениями режимов. В этом случае значения таблицы 3 могут быть объединены по схеме ИЛИ. Следует отметить, что поле fwHeadModeExt используется только для кодирования квазистерео; в случае других режимов (один канал, двойной канал или стерео) значение поля должно быть установлено равным нулю.

В общем, декодеры должны динамически переключаться между различными возможными значениями mode\_extension в соответствии с характеристиками сигнала. Следовательно, для нормального кодирования квазистерео это поле должно быть установлено равным 0x000f. Вместе с тем, если желательно ограничить декодер конкретным типом кодирования квазистерео, это поле может использоваться для определения допустимых типов.

wHeadEmphasis

Описывает частотную коррекцию, требуемую декодером; это подразумевает внесение предсказаний в поток до процесса кодирования. См. таблицу 4.

fwHeadFlags

Наборы соответствующих флагов в заголовке аудиокадра:

ACM\_MPEG\_PRIVATEBIT – устанавливает бит "частный".

ACM\_MPEG\_COPYRIGHT – устанавливает бит "копирайт".

ACM\_MPEG\_ORIGINALHOME – устанавливает бит оригинал/основной.

АСМ\_MPEG\_PROTECTIONBIT – устанавливает бит защиты и вставляет 16-битовый код защиты от ошибок в каждый кадр.

АСМ\_MPEG\_ID\_MPEG1 – устанавливает бит ID равным 1, определяя поток как аудиопоток формата MPEG-1. *Этот флаг должен быть всегда установлен в явной форме для поддержания совместимости с будущими расширениями MPEG аудио (т. е. MPEG-2).*

Декодер должен использовать значение этих флагов для установления значений соответствующих битов в заголовке каждого аудиокadra MPEG. При описании кодированного потока данных эти флаги представляют объединение по логической схеме ИЛИ флагов, установленных в заголовке каждого кадра. Это значит, если бит копирайта установлен в одном или более заголовке кадра, тогда флаг АСМ\_MPEG\_COPYRIGHT должен быть установлен. Следовательно, значение этих флагов необязательно действительно для всех аудиокadров.

**dwPTSLow** В этом поле (вместе со следующим полем) содержится представление отметки времени (PTS) первого кадра аудиопотока из системного слоя MPEG. Поле dwPTSLow содержит 32 наименее значащих бита 33-битовой PTS. PTS может использоваться для целей реинтеграции аудиопотока со связанным видеопотоком. Если аудиопоток не связан с системным слоем, тогда это поле должно быть установлено равным нулю.

**dwPTSHigh** В этом поле (вместе с предыдущим полем) содержится представление отметки времени (PTS) первого кадра аудиопотока из системного слоя MPEG. LSB поля dwPTSHigh содержат MSB 33-битовой PTS. PTS может использоваться для целей реинтеграции аудиопотока со связанным видеопотоком. Если аудиопоток не связан с системным слоем, тогда это поле должно быть установлено равным нулю.

ПРИМЕЧАНИЕ. – Два предыдущих поля могут обрабатываться как единое 64-битовое целое; факультативно поле dwPTSHigh может проверяться как флаг для определения того, установлен или очищен MSB.

ТАБЛИЦА 1

## Допустимые битовые скорости (бит/с)

Код заголовка кадра MPEG	Layer I	Layer II	Layer III
'0000'	произвольный формат	произвольный формат	произвольный формат
'0001'	32 000	32 000	32 000
'0010'	64 000	48 000	40 000
'0011'	96 000	56 000	48 000
'0100'	128 000	64 000	56 000
'0101'	160 000	80 000	64 000
'0110'	192 000	96 000	80 000
'0111'	224 000	112 000	96 000
'1000'	256 000	128 000	112 000
'1001'	288 000	160 000	128 000
'1010'	320 000	192 000	160 000
'1011'	352 000	224 000	192 000
'1100'	384 000	256 000	224 000
'1101'	416 000	320 000	256 000
'1110'	448 000	384 000	320 000
'1111'	запрещен	запрещен	запрещен

ТАБЛИЦА 2

## Допустимые сочетания режима и битовой скорости для Layer II

Битовая скорость (бит/с)	Допустимые режимы
32 000	один канал
48 000	один канал
56 000	один канал
64 000	все режимы
80 000	один канал
96 000	все режимы
112 000	все режимы
128 000	все режимы
160 000	все режимы
192 000	все режимы
224 000	стерео, интенсивное стерео, двойной канал
256 000	стерео, интенсивное стерео, двойной канал
320 000	стерео, интенсивное стерео, двойной канал
384 000	стерео, интенсивное стерео, двойной канал

ТАБЛИЦА 3

## Расширение режима

fwHeadModeExt	Код заголовка кадра MPEG	Layer I и II	Layer III
0x0001	'00'	субполосы 4–31 в интенсивном стерео	вид кодирования, не являющийся интенсивным или MS-стерео
0x0002	'01'	субполосы 8–31 в интенсивном стерео	интенсивное стерео
0x0004	'10'	субполосы 12–31 в интенсивном стерео	MS-стерео
0x0008	'11'	субполосы 16–31 в интенсивном стерео	оба вида кодирования – интенсивное и MS-стерео

ТАБЛИЦА 4

## Поле предсказания

wHeadEmphasis	Код заголовка кадра MPEG	Требуемая частотная коррекция
1	'00'	без предсказания
2	'01'	предсказание 50/15 мкс
3	'10'	зарезервировано
4	'11'	Рекомендация МСЭ-T J.17



### 1.3 Флаги, используемые в полях данных

#### fwHeadLayer

Для поля <fwHeadLayer> определяются следующие флаги. В случае кодирования один из этих флагов должен быть установлен, с тем чтобы кодер "понимал", какой использовать слой. В случае декодирования драйвер может проверять эти флаги, с тем чтобы определить, возможно ли декодировать поток. Следует заметить, что в допустимом потоке MPEG могут использоваться разные слои в разных кадрах в пределах одного потока. Поэтому могут быть установлены несколько этих флагов.

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

#### fwHeadMode

Для поля <fwHeadMode> определяются следующие флаги. В случае кодирования один из этих флагов должен быть установлен, с тем чтобы кодер "понимал", какой использовать режим; в случае кодирования квазистерео, как правило, должны быть установлены оба флага – ACM\_MPEG\_STEREO и ACM\_MPEG\_JOINTSTEREO, с тем чтобы кодер мог применять кодирование квазистерео, только если это более эффективно по сравнению со стерео. В случае декодирования драйвер может проверять эти флаги, с тем чтобы определить, возможно ли декодировать поток. Следует заметить, что в допустимом потоке MPEG могут использоваться разные слои в разных кадрах в пределах одного потока. Поэтому могут быть установлены несколько этих флагов.

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

#### fwHeadModeExt

В таблице 3 определены флаги для поля <fwHeadModeExt>. Это поле используется только для кодирования квазистерео; в случае других режимов кодирования это поле должно быть установлено равным нулю. В случае кодирования квазистерео эти флаги указывают типы кодирования квазистерео, которые разрешены для использования кодером. Обычно кодер должен динамически выбирать то расширение режима, которое наиболее соответствует входному сигналу; следовательно, желательно, чтобы приложение устанавливало, как правило, это поле равным 0x000f, для того чтобы кодер мог выбирать из всех возможностей; вместе с тем, возможно ограничить кодер путем очистки ряда флагов. В случае кодированного потока это поле указывает значения поля MPEG *mode\_extension*, которые присутствуют в потоке.

#### fwHeadFlags

Для поля <fwHeadFlags> определяются следующие флаги. Эти флаги должны быть установлены до кодирования, с тем чтобы возможно было установить соответствующие биты в заголовке кадра MPEG. При описании кодированного аудиопотока MPEG эти флаги представляют объединение по логической схеме ИЛИ соответствующих битов в заголовке каждого аудиокadra. Это значит, что если данный бит установлен в каком-либо кадре, он устанавливается в поле <fwHeadFlags>. Если приложение свертывает заголовок RIFF WAVE вокруг предварительно кодированного битового аудиопотока MPEG, оно несет ответственность за анализ битового потока и установку флагов в этом поле.

```
#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 Аудиоданные в файлах MPEG

Фрагмент данных `<data chunk>` содержит аудиопоследовательность MPEG-1 согласно определению в спецификации ISO 11172, часть 3 (аудио). Эта последовательность состоит из битового потока, который сохраняется во фрагменте данных в виде матрицы байтов. В байте первым битом потока является MSB, а LSB является последним битом. В данных порядок следования байтов *не* обратный. Например, следующие данные состоят из первых 16 битов (слева направо) типичного заголовка аудиокадра:

Syncword	ID	Layer	ProtectionBit	...
11111111111111	1	10	1	...

Эти данные сохраняются в байтах со следующим порядком следования:

Byte0	Byte1	...
FF	FD	...

### 1.4.1 Аудиокадры MPEG

Аудиопоследовательность MPEG состоит из серии аудиокадров, каждый из которых начинается заголовком кадра. Большинство полей в этом заголовке кадра соответствуют полям в определенной выше структуре MPEG1WAVEFORMAT. В случае кодирования эти поля могут быть установлены в структуре MPEG1WAVEFORMAT, и драйвер может использовать эту информацию в процессе кодирования для установки соответствующих битов в заголовке кадра. В случае декодирования драйвер может проверять эти поля, с тем чтобы установить, возможно ли декодировать данный поток.

### 1.4.2 Кодирование

Драйвер, который кодирует аудиопоток MPEG, должен считывать поля заголовка в структуре MPEG1WAVEFORMAT и устанавливать соответствующие биты в заголовке кадра MPEG. Если драйверу требуется какая-либо дополнительная информация, он должен получить ее либо из диалогового окна конфигурации, либо с помощью возвращающей функции. Более подробную информацию см. в разделе "Вспомогательные данные", ниже.

Если предварительно закодированный аудиопоток MPEG обернут заголовком RIFF, разделение битового потока на компоненты и установка полей в структуре MPEG1WAVEFORMAT является функцией приложения. Если частота дискретизации или индекс битовой скорости не постоянны в пределах потока данных, драйвер должен установить соответствующие поля MPEG1WAVEFORMAT (`<nSamplesPerSec>` и `<dwHeadBitrate>`) равными нулю, как описано выше. Если поток содержит кадры нескольких слоев, драйвер должен установить флаги в `<fwHeadLayer>` для всех слоев, представленных в потоке. Поскольку такие поля, как `<fwHeadFlags>`, могут изменяться от кадра к кадру, следует проявлять осторожность при установке и проверке этих флагов; в общем, приложение не должно полагаться на действительность этих флагов для каждого кадра. При установке этих флагов следует придерживаться следующих правил:

- ACM\_MPEG\_COPYRIGHT должно устанавливаться, если в каком-либо из кадров в потоке установлен бит копирайта.
- ACM\_MPEG\_PROTECTIONBIT должно устанавливаться, если в каком-либо из кадров в потоке установлен бит защиты.
- ACM\_MPEG\_ORIGINALHOME должно устанавливаться, если в каком-либо из кадров в потоке установлен бит признака оригинала/основного. Этот бит может быть очищен, если сделана копия потока.
- ACM\_MPEG\_PRIVATEBIT должно устанавливаться, если в каком-либо из кадров в потоке установлен бит "частный".
- ACM\_MPEG\_ID\_MPEG1 должно устанавливаться, если в каком-либо из кадров в потоке установлен бит ID. Для потоков MPEG-1 бит ID должен быть установлен всегда; вместе с тем в будущих расширениях MPEG (таких как многоканальный формат MPEG-2) бит ID может быть очищен.

Если аудиопоток MPEG взят из потока MPEG системного слоя или если поток предназначен для интеграции в системный слой, тогда могут использоваться поля PTS. PTS – это поле в системном слое MPEG, которое используется для синхронизации различных полей. Поле PTS формата MPEG – это 33 бита, и, следовательно, в заголовке формата RIFF WAVE значение сохраняется в двух полях: <dwPTSLow> содержит 32 бита LSB отметки PTS, а <dwPTSHigh> содержит MSB. Эти два поля могут восприниматься вместе как 64-битовое целое; факультативно может проверяться поле <dwPTSHigh> как флаг для определения того, установлен или очищен MSB. При извлечении аудиопотока из системного слоя драйвер должен установить поля PTS равными PTS первого кадра аудиоданных. Это может позже использоваться для реинтеграции потока в системный слой. *Поля PTS не должны использоваться для какой-либо иной цели.* Если аудиопоток не связан с системным слоем MPEG, тогда поля PTS должны быть установлены равными нулю.

### 1.4.3 Декодирование

Драйвер может проверять поля в структуре MPEG1WAVEFORMAT, для того чтобы определить, возможно ли декодировать поток. Вместе с тем драйвер должен знать, что некоторые поля, такие как <fwHeadFlags>, могут не быть согласованными для каждого кадра битового потока. Драйвер никогда не должен использовать поля структуры MPEG1WAVEFORMAT для выполнения фактического декодирования. Параметры декодирования должны целиком быть взяты из потока данных MPEG.

Драйвер может проверить поле <nSamplesPerSec>, для того чтобы определить, поддерживает ли он указанную частоту дискретизации. Если поток MPEG содержит данные с переменной частотой дискретизации, тогда поле <nSamplesPerSec> должно быть установлено равным нулю. Если драйвер не может обработать этот тип потока данных, он не должен пытаться декодировать эти данные, а должен немедленно сформировать отказ.

### 1.5 Вспомогательные данные

Аудиоданные в аудиокadre MPEG могут не заполнить весь кадр. Любые оставшиеся данные называются *вспомогательными данными*. Такие данные могут быть любого желаемого формата и могут использоваться для переноса дополнительной информации любого вида. Если желательно, чтобы драйвер поддерживал вспомогательные данные, он должен быть оборудован средствами для передачи данных от и к вызывающему приложению. Для этой цели драйвер должен использовать возвращающую функцию. В принципе, драйвер может вызывать конкретную возвращающую функцию каждый раз при наличии вспомогательных данных, которые должны быть переданы приложению (т. е. при декодировании), или всякий раз, когда ему необходимы дополнительные вспомогательные данные (при кодировании).

Драйверы должны иметь информацию о том, что не все приложения будут обрабатывать вспомогательные данные. Следовательно, драйвер должен обеспечивать эту услугу только в случае явного ее запроса приложением. Драйвер может определить специальное сообщение, которое разрешает или запрещает применение средств обратного вызова. Могут определяться отдельные сообщения для обеспечения большей гибкости кодирования и декодирования.

Следует заметить, что этот метод может не быть пригодным для всех драйверов и приложений, он включен только в качестве иллюстрации того, как может быть обеспечена поддержка вспомогательных данных.

ПРИМЕЧАНИЕ 1. – Более подробная информация о вспомогательных данных содержится во фрагменте <MPEG\_Audio\_Extension\_chunk>, который должен использоваться для файлов MPEG, соответствующих формату Broadcast Wave. См. раздел 2 основной части Приложения 2.

### Ссылки

ISO/IEC 11173-3: MPEG 1.

ISO/IEC 13818-3: MPEG 2.

ПРИМЕЧАНИЕ. – Документы Microsoft® доступны по следующему адресу: <http://www.microsoft.com>.

## Приложение 3

### Спецификация формата BWF

#### Формат файлов аудиоданных в радиовещании

##### СПЕЦИФИКАЦИИ МЕТАДАННЫХ

### 1 Введение

В настоящем приложении представлена спецификация для использования формата BWF в целях переноса информации об аудиоматериале, собранном и обработанном цифровой звуковой станцией DAW (см. рисунок 2). Файл BWF используется как независимый от платформы контейнер для звукового сигнала и всех соответствующих метаданных. Принимающий сервер архивов способен извлекать требуемую информацию из такого файла и использовать ее необходимым образом: например, вводить ее в базу данных и т. д. (см. рисунок 3).

РИСУНОК 2

Сбор данных рабочей станцией в файл BWF

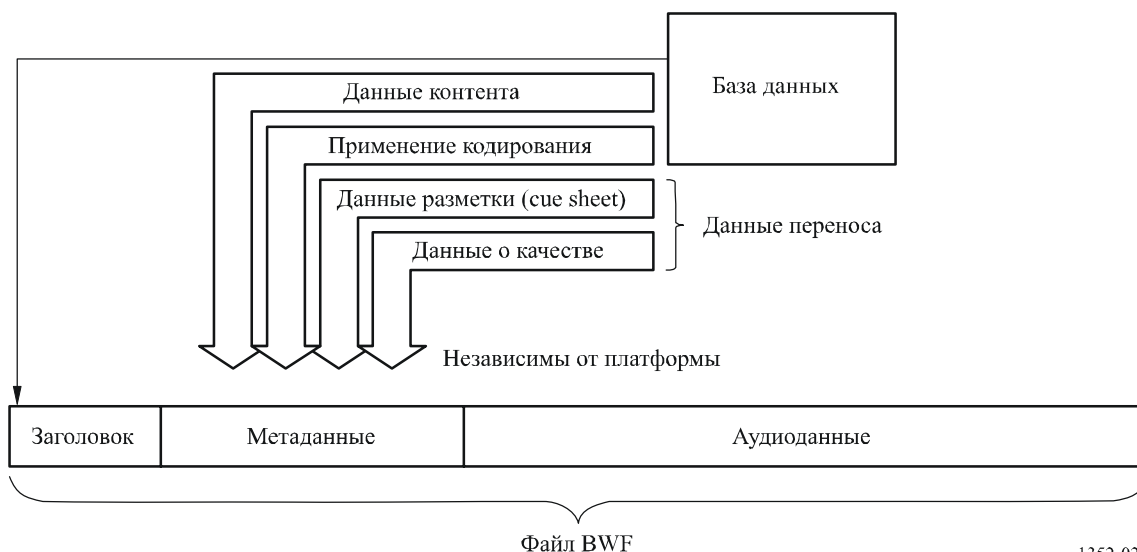
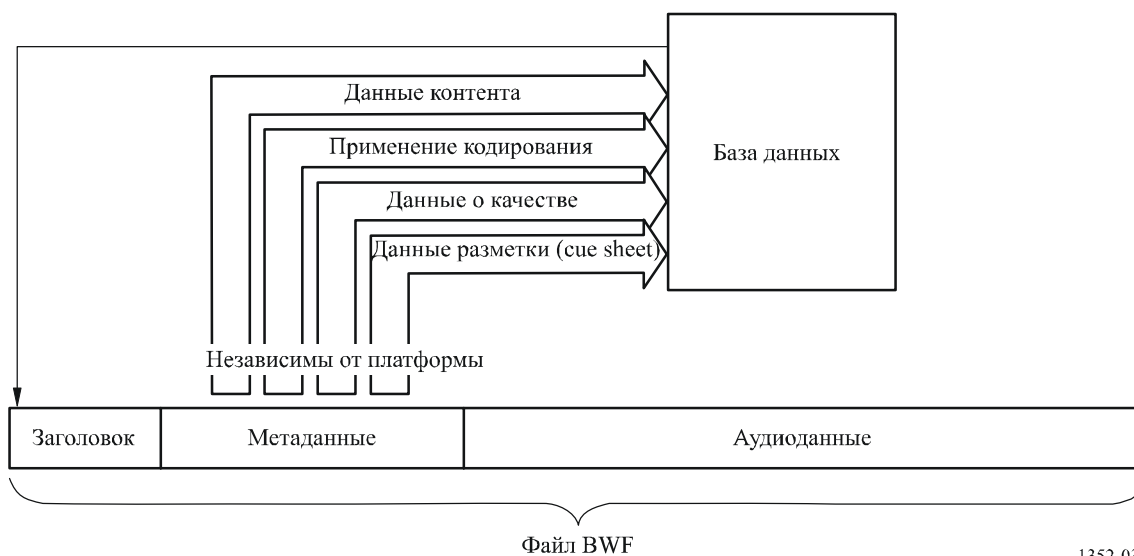


РИСУНОК 3

## Извлечение данных приемным сервером архивов из файла BWF



1352-03

В настоящем Приложении определяется новый фрагмент, предназначенный для переноса информации, не присутствующей в базовом файле BWF, а также описывается порядок использования существующих фрагментов BWF.

Следует соблюдать осторожность в случае, если в файлах BWF содержатся отредактированные отчеты о качестве. Если система редактирования объединяет более одного файла BWF, список редакторских правок (EDL) должен указывать соответствующие части фрагментов данных о применении кодирования и качества каждого исходного файла BWF. Кроме того, если новый файл воспроизводится из частей других файлов, для нового файла должны быть созданы новые фрагменты данных о применении кодирования и качества.

## 2 Отчет о записи

Для защиты хранимых в архивах аналоговых или цифровых оригиналов с одной несущей важно осуществить повторную запись оригинального звукового сигнала с полным сохранением качества в файлы BWF. Отчет о записи содержит информацию обо всей цепочке обработки – от аналоговой до цифровой среды – или информацию, необходимую для передачи в пределах цифровой среды (например, с CD или кассеты DAT).

Отчет о записи размещается вместе с данными анализа аудиосигнала, как часть метаданных файла BWF.

Отчет о записи составляют три части:

- Поле данных о применении кодирования CodingHistory фрагмента <bext> файла BWF. В этом поле содержится подробная информация обо всей цепочке передачи, например, от типа магнитной ленты, компакт-диска или кассеты DAT до файла BWF (процесс производства звукового сигнала).
- Отчет о качестве во фрагменте <qlty>. В этом отчете содержится информация, описывающая все события, оказавшие воздействие на качество записанного звукового сигнала во фрагменте волновых данных. Каждое событие, распознаваемое оператором или компьютером, вносится в список с подробной информацией о типе события, точными отметками времени, приоритетом или статусом события. Также регистрируются параметры общего качества и т. д.
- Разметка во фрагменте <qlty>, являющаяся списком событий, помеченных точными отметками времени, и содержащая дополнительное описание звукового сигнала, например начало арии или начальная точка важного выступления. Таким образом хранители архивов могут составить метаданные базы данных с помощью компьютерных инструментов.

## 2.1 Синтаксис отчета о записи

- Отчет о записи состоит из цепочек символов в формате ASCII (ISO 646) [ISO/IEC, 1991], организованных в строки длиной до 256 символов.
- Каждая строка должна завершаться символом <CR/LF> (ASCII 0Dh, 0Ah).
- Строка может содержать одну или более переменных цепочек, разделенных запятой (ASCII 2Bh).
- Переменные цепочки состоят из ASCII-символов и не должны содержать запятых.
- В пределах цепочки в качестве разделителя должно использоваться двоеточие (ASCII 3Bh).

## 3 Поле данных о применении кодирования CodingHistory фрагмента <bext>

Используемые в поле кодирования цепочки описаны в Дополнении 2 к Приложению 1. Эта информация воспроизведена ниже для удобства.

A=<ANALOGUE, .....>	Информация о тракте аналогового звукового сигнала
A=<PCM, .....>	Информация о тракте цифрового звукового сигнала
F=<48000, 441000, etc.>	Частота дискретизации [Гц]
W=<16, 18, 20, 22, 24, etc.>	Длина слова [биты]
M=<mono, stereo, 2-channel>	Режим
T=<free ASCII-text string>	Текст комментария

## 4 Фрагмент качества Quality

Фрагмент качества Quality определен в выделенном курсивом тексте в п. 4.1:

### 4.1 Элементы фрагмента качества Quality

**FileSecurityReport:** В этом поле содержится код защиты файла FileSecurityCode фрагмента QualityChunk.

Это – 32-битовое значение, содержащее контрольную сумму [0 ....231].

**FileSecurityWave:** В этом поле содержится код защиты файла FileSecurityCode данных BWF Wave.

Это – 32-битовое значение, содержащее контрольную сумму [0 ....231].

```
Quality-chunk typedef struct {
    DWORD    ckID;                /* (quality-chunk) ckID='qlty' */
    DWORD    ckSize;             /* размер фрагмента качества */
    BYTE     ckData[ckSize];     /* данные фрагмента */
}

typedef struct quality_chunk {
    DWORD FileSecurityReport;     /* FileSecurityCode отчета о качестве */
    DWORD FileSecurityWave;      /* FileSecurityCode данных BWF Wave */
    CHAR BasicData[ ];          /* ASCII: "Базовые данные" */
    CHAR StartModulation[ ];    /* ASCII: "Данные о начале модуляции" */
    CHAR QualityEvent[ ];       /* ASCII: "Данные о связанных с качеством событиях" */
    CHAR EndModulation[ ];      /* ASCII: "Данные о завершении модуляции" */
    CHAR QualityParameter[ ];   /* ASCII: "Данные параметров качества" */
    CHAR OperatorComment[ ];    /* ASCII: "Комментарии оператора" */
    CHAR CueSheet[ ];           /* ASCII: "Данные разметки (cue sheet)" */
} quality_chunk
```

**Базовые данные:** Базовые данные записи.

**B=** ASCII-строка, содержащая базовые данные о звуковом материале.

**Archive No. (AN):** Архивный номер (максимально 32 символа).

**Title (TT):** Название/разделение звуковых данных (максимально 256 символов).

- Duration (TD): 10 ASCII-символов, содержащих длительность по времени звуковой последовательности.  
 Формат: " чч:мм:сс:д "  
 Часы чч: 0...23  
 Минуты мм: 0...59  
 Секунды сс: 0...59  
 1/10 с д: 0...9
- Date (DD): 10 ASCII-символов, содержащих дату перевода в цифровую форму.  
 Формат: " гггг:мм:дд "  
 Год гггг: 0000...9999  
 Месяц мм: 0...12  
 День дд: 0...31
- Operator (OP): ASCII-строка (максимально 64 символа), содержащая имя лица, выполняющего операцию перевода в цифровую форму.
- Copying station (CS): ASCII-строка (максимально 64 символа), содержащая тип и серийный номер рабочей станции, использовавшейся для создания файла.
- StartModulation: Начало модуляции (SM) оригинальной записи.
- SM= 10 ASCII-символов, содержащих время начала звукового сигнала от начала файла.  
 Формат: " чч:мм:сс:д "  
 Часы чч: 0...23  
 Минуты мм: 0...59  
 Секунды сс: 0...59  
 1/10 с д: 0...9
- Sample count (SC): Код адреса выборки точки SM от начала файла (шестнадцатиричное представление начала).  
 Формат: " #####Н "  
 0Н..... FFFFFFFFH (0.....  $4,295 \times 10^9$ )
- Comment (T): ASCII-строка, содержащая комментарии.
- QualityEvent Информация, описывающая каждое событие, связанное с качеством, в течение звукового сигнала. Для каждого события используется одна строка QualityEvent.
- Q= ASCII-строка (максимально 256 символов), содержащая события, связанные с качеством.
- Event number (M): Нумерованная отметка, создаваемая оператором вручную.  
 Формат: " M### " ###: 001...999
- Event number (A): Нумерованная отметка, создаваемая системой автоматически.  
 Формат: " A### " ###: 001...999
- Priority (PRI): Приоритет связанного с качеством события.  
 Формат: " # " #: 1 (LO)..... 5 (HI)
- Time stamp (TS): 10 ASCII-символов, содержащих отметку времени связанного с качеством события от начала файла.  
 Формат: " чч:мм:сс:д "  
 Часы чч: 0...23  
 Минуты мм: 0...59  
 Секунды сс: 0...59  
 1/10 с д: 0...9

Event type (E):	ASCII-строка (максимально 16 символов), описывающая тип события, например, "Click" (щелчок), "AnalogOver" (наложение аналогового сигнала), "Transparency" (прозрачность) или Параметры качества QualityParameter (определены ниже), превышающие предельные значения, например, "QP:Azimuth:L-20.9smp".	
Status (S):	ASCII-строка (максимально 16 символов), содержащая статус обработки события, например, "unclear" (не установлено), "checked" (проверено), "restored" (восстановлено), "deleted" (удалено).	
Comment (T):	ASCII-строка, содержащая комментарии.	
Sample count (SC):	Код адреса выборки точки TS от начала файла (в шестнадцатичной форме ASCII). Формат: "#####H " 0H..... FFFFFFFFH (0..... $4,295 \times 10^9$ )	
QualityParameter	Параметры качества (QP), описывающие звуковой сигнал.	
P=	ASCII-строка (максимально 256 символов), описывающая параметры качества.	
Parameters (QP):	Макс. пик:            -xx.x dBFSL; -yy.y dBFSR	[ -99,9...-00,0]
	Средний уровень: -xx.x dBFSL; -yy.y dBFSR	[ -99,9...-00,0]
	Корреляция:           ±x.x	[ -1,0.....+1,0]
	Динамика:            xx.x dBL; yy.y dBR	[00,0..... 99,9]
	(динамический диапазон)	
	Выборки с ограничением: xxxx smpL; yyyu smpR	[0.....9999]
	SNR:                   xx.x dBL; yy.y dBR	[00,0.....99,9]
	(отношение сигнал-шум)	
	Ширина полосы:    xxxxx HzL; yyyuu HzR	[0.....20000]
	Азимут:               L±xx.x smp	[ -99,9....+99,9]
	Баланс:               L±x.x dB	[ -9,9.....+9,9]
	Смещение по постоянному току:  x.x %L; y.y %R	[0,0.....9,9]
	Речь:                   xx.x%	[0,0.....99,9]
	Сtereo:               xx.x%	[0,0.....99,9]
	(L = левый канал, R = правый канал)	
Quality factor (QF):	Суммарный коэффициент качества звукового файла [1..... 5 (наилучший), 0 = неопределенный].	
Inspector (IN):	ASCII-строка (максимально 64 символа), содержащая имя лица, проверяющего звуковой файл.	
File status (FS):	Строка ASCII-символов, описывающая статус "Готов к передаче?". [Y(es) / N(o) / U: File is ready/not ready/FS is undefined (Y(да) / N(нет) / U: Файл готов/не готов/FS не определен].	
OperatorComment	Комментарии оператора.	



T=	ASCII-строка (максимально 256 символов), содержащая комментарии.
EndModulation	Конец модуляции.
EM=	10 ASCII-символов, содержащих время окончания модуляции звукового сигнала. Формат: " чч:мм:сс:д " Часы чч: 0...23 Минуты мм: 0...59 Секунды сс: 0...59 1/10 с д: 0...9
Sample count (SC):	Код адреса выборки точки EM (в шестнадцатиричной форме ASCII). Формат: " #####Н " 0H.....FFFFFFFFH (0.....4,295 × 10 <sup>9</sup> )
Comment (T):	ASCII-строка, содержащая комментарии.
CueSheet	Данные разметки.
C=	ASCII-строка (максимально 256 символов), содержащая точки разметки.
Cue number (N):	Номер точки разметки, автоматически создаваемый системой. Формат: "N####" ###: 001...999
Time stamp (TS):	10 ASCII-символов, содержащие отметку времени данной точки разметки. Формат: " чч:мм:сс:д " Часы чч: 0...23 Минуты мм: 0...59 Секунды сс: 0...59 1/10 с д: 0...9
Text (T):	ASCII-строка, содержащая комментарии к точке разметки, например "Начало арии".
Sample count (SC):	Код адреса выборки точки TS (в шестнадцатиричной форме ASCII) Формат: " #####Н " 0H..... FFFFFFFFH (0...4,295 × 10 <sup>9</sup> ).

## 5 Примеры отчетов о записи

### 5.1 Процесс преобразования в цифровую форму аналогового материала

(базовая информация, содержащаяся в поле данных о применении кодирования CodingHistory фрагмента <bext>)

Строка

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

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

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

#### (Отчет о качестве во фрагменте качества)

Номер строки

01 <FileSecurityReport>

02 <FileSecurityWave>

03 B=CS=QUADRIGA2.0; SN10012, OP=name of operator<CR/LF>  
 04 B=AN=archive number, TT=title of sound<CR/LF>  
 05 B=DD= yyyy:mm.dd, TD=hh:mm:ss.d<CR/LF>  
 06 SM=00:00:04:5, T=tape noise changing to ambience, SC=34BC0H<CR/LF>  
 07 Q=A001, PRI=2, TS=00:01:04:0, E=Click, S=unclear, SC=2EE000H<CR/LF>  
 08 Q=A002, PRI=3, TS=00:12:10:3, E=DropOut, S=checked, SC=216E340H<CR/LF>  
 09 Q=A003, PRI=4, TS=00:14:23:0, E=Transparency, S=checked, SC=2781480H<CR/LF>  
 10 Q=M004, PRI=1, TS=00:18:23:1, E=PrintThrough, S=checked, SC=327EF40H<CR/LF>  
 11 Q=A005, PRI=1, TS=00:20:01:6, E=ClickOn, S=unclear, T=needs restoration,  
 SC=3701400H<CR/LF>  
 12 Q=A006, PRI=5, TS=00:21:20:3, E=QP:Azimuth:L=-20.9smp, S=unclear,  
 SC=3A9B840H<CR/LF>  
 13 Q=A007, PRI=3, TS=00:21:44:7, E=AnalogOver, S=checked, 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=deleted, SC=4D16600H<CR/LF>  
 16 EM=00:39:01:5, T=fade-out of applause, SC=6B2F740H<CR/LF>  
 17 P=QP:MaxPeak:-2.1dBFS; -2.8dBFS<CR/LF>  
 18 P=QP:MeanLevel:-11.5dBFS; 8.3dBFS<CR/LF>  
 19 P=QP:Correlation:+0.8<CR/LF>  
 20 P=QP:Dynamic:51.4dB; 49.6dB<CR/LF>  
 21 PAP:ClippedSamples:Osmpl;Osmpr<CR/LF>  
 22 P=QP:SNR:32.3dB; 35.1dB<CR/LF>  
 23 P=QP:Bandwidth:8687Hz; 7943Hz<CR/LF>  
 24 P=QP:Azimuth:L-6.2smp<CR/LF>  
 25 P=QP:Balance L:+2.1dB<CR/LF>  
 26 P=QP:DC-Offset:0.0%; 0.0%<CR/LF>  
 27 P=QP:Speech:64.2%<CR/LF>  
 28 P=QP:Stereo:89.3%<CR/LF>  
 29 P=QF=2<CR/LF>  
 30 P=IN=name of inspector<CR/LF>  
 31 P=FS=N<CR/LF>

**(Таблица разметки CueSheet во фрагменте качества)**

Номер строки

32 C=N001, TS=00:17:02:5, T=beginning of speech, SC=2ECE6C0 H<CR/LF>  
 33 C=N002, TS=00:33:19:2, T=start of aria, SC=5B84200H<CR/LF>

**Интерпретация примера 1****(базовая информация в поле данных о применении кодирования CodingHistory)**

- Строка 1: Воспроизведение аналоговой магнитной ленты типа Agfa PER528 на магнитофоне Stude A816, имеющем серийный номер 1007, с использованием расширителя Telcom:
- Скорость движения ленты: 38 см/с  
Режим: стерео
- Строка 2: Для преобразования в цифровую форму использовался А/Ц преобразователь типа NVision NV 1000 со следующими параметрами:
- Частота дискретизации: 48 кГц  
Разрядность кодирования: 18 битов в каждой выборке  
Режим: стерео
- Строка 3: Оригинальный файл записан как линейный файл WBF с ИКМ-кодированием и использованием цифрового входа станции перезаписи без добавления шумового сигнала:
- Частота дискретизации: 48 кГц  
Разрядность кодирования: 16 битов в каждой выборке  
Режим: стерео

**(Отчет о качестве во фрагменте качества)**

- Строки 1–2: Коды защиты файла фрагмента качества и волновых данных.
- Строки 3–5: Оператор использует станцию перезаписи QUADRIGA2.0, серийный номер 10012.  
(OP). Лента имеет архивный номер (AN) и заглавие (TT) и преобразована в цифровую форму в указанную дату.  
(DD). Длительность звукового сигнала в файле WBF составляет (TD).
- Строка 6: Начало модуляции (SM) в соответствующее временной метке время (TS) и отсчет выборок (SC) с комментарием (T).
- Строки 7–15: События (E) опознаются оператором (M) и/или управлением системы (A) с приоритетом (PRI) и в соответствующее временной метке время (TS). Статус события (S) и комментарии (T) предоставляют дополнительную информацию. Подсчет выборок (SC) обеспечивает точную отметку времени.
- Строка 16: Окончание модуляции (EM) в соответствующее временной метке время (SC) с комментарием (T).
- Строки 17–28: Параметры качества (QP) полного звукового сигнала во фрагменте волновых данных.
- Строки 29–31: Суммарный коэффициент качества (QF), выданный автоматическим управлением системы, и имя проверяющего (IN), а также решение (FS) о том, является ли качество звукового файла соответствующим "готов к передаче".

**(Таблица разметки CueSheet во фрагменте качества)**

- Строки 32–33: Точками разметки помечается начало выступления и начальная точка арии.

## 5.2 Процесс записи на компакт-диск

(базовая информация в поле данных о применении кодирования CodingHistory фрагмента <bext>)

Номер строки

01 A=PCM, F=44100, W=16, M=stereo, T=SonyCDP-D500; SN2172; Mitsui CD-R74<CR/LF>

02 A=PCM, F=48000, W=24, M=stereo, T=DCS972; D/D<CR/LF>

03 A=PCM, F=48000, W=24, M=stereo, T=nodither;DIO<CR/LF>

(Отчет о качестве QualityReport во фрагменте качества)

Номер строки

01 <FileSecurityReport>

02 <FileSecurityWave>

и т. д.: аналогично примеру в п. 5.1, выше.

(Таблица разметки CueSheet во фрагменте качества)

Аналогично примеру в п. 5.1, выше.

### Интерпретация примера 2

(базовая информация в поле данных о применении кодирования CodingHistory)

Строка 1: CD перезаписываемого типа Mitsui CD-R74 воспроизводится на CD проигрывателе Sony CDP-D500, серийный номер 2172:

Частота дискретизации: 44,1 кГц

Разрядность кодирования: 16 битов в каждой выборке

Режим: стерео

Строка 2: Используется конвертер частоты дискретизации DCS972 с параметрами:

Частота дискретизации: 48 кГц (от 44,1 кГц)

Разрядность кодирования: 24 бита в каждой выборке

Режим: стерео

Строка 3: Оригинальный файл записан как линейный файл BWF с ИКМ-кодированием и использованием цифрового входа станции перезаписи без добавления шумового сигнала:

Частота дискретизации: 48 кГц

Разрядность кодирования: 24 бита в каждой выборке

Режим: стерео

(Отчет о качестве QualityReport во фрагменте качества)

Строки 1–2: Коды защиты файла фрагмента качества и волновых данных.

Другие данные используются согласно процессу записи CD аналогично примеру 1 в п. 5.1, выше.

(Таблица разметки CueSheet во фрагменте качества)

Данные разметки используются согласно процессу записи компакт-диска аналогично примеру 1 в п. 5.1, выше.

### 5.3 Процесс записи цифровой аудиокассеты DAT

(базовая информация в поле данных о применении кодирования CodingHistory фрагмента <bext>)

Номер строки

01 A=PCM, F=48000, W=16, M=stereo, T=SonyPCM-8500; SN1037; TDKDA-R120 <CR/LF>

02 A=PCM, F=48000, W=16, M=stereo, T=no dither; DIO<CR/LF>

(Отчет о качестве QualityReport во фрагменте качества)

Номер строки

01 <FileSecurityReport>

02 <FileSecurityWave>

и т. д.: аналогично примеру в п. 5.1, выше.

(Таблица разметки CueSheet во фрагменте качества)

Аналогично примеру в п. 5.1, выше.

#### Интерпретация примера 3

(базовая информация в поле данных о применении кодирования CodingHistory)

Строка 1: Цифровая аудиокассета DAT типа TDK DA-8120 воспроизводится на магнитофоне Sony PCM-8500, серийный номер 1037:

Частота дискретизации: 48 кГц  
 Разрядность кодирования: 16 битов в каждой выборке  
 Режим: стерео

Строка 2: Оригинальный файл записан как линейный файл WFF с ИКМ-кодированием и использованием цифрового входа станции перезаписи без добавления шумового сигнала:

Частота дискретизации: 48 кГц  
 Разрядность кодирования: 16 битов в каждой выборке  
 Режим: стерео

(Отчет о качестве QualityReport во фрагменте качества)

Строки 1–2: Коды защиты файла фрагмента качества и волновых данных.

Другие данные используются согласно процессу записи кассеты DAT аналогично примеру 1 в п. 5.1, выше.

(Таблица разметы CueSheet во фрагменте качества)

Данные разметки используются согласно процессу записи кассеты DAT аналогично примеру 1 в п. 5.1, выше.

---