

EBU – TECH 3285



Specification of the Broadcast Wave Format (BWF)

A format for audio data files in broadcasting

Version 2.0

Geneva
May 2011

EBU – TECH 3285



Specification of the Broadcast Wave Format (BWF)

Внимание!

Данный перевод **НЕ** претендует на аутентичность
и может содержать отдельные неточности.
Оригинал документа на сайте <https://tech.ebu.ch>

Спецификация Broadcast Wave Format (BWF)

Формат для файлов аудиоданных в вещании

Версия 2.0

**Женева
Май 2011**

Резюме

Broadcast Wave Format (BWF) – формат файлов для аудиоданных. Он может использоваться для бесшовного обмена аудиоматериалом между различной вещательной средой и между оборудованием на базе различных компьютерных платформ.

Помимо аудиоданных, файл BWF содержит минимальную информацию – или метаданные – которая считается необходимой для всех вещательных приложений. Broadcast Wave Format основан на формате аудиофайлов Microsoft WAVE, к которому EBU добавил порцию данных “Broadcast Audio Extension”.

BWF Версия 0

Спецификация Broadcast Wave Format для аудиоданных PCM (сейчас называется Версия 0) была опубликована в 1997 г. как EBU Tech 3285.

BWF Версия 1

Версия 1 отличается Версии 0 только тем, что 64 из 254 резервных байтов в Версии 0 используются для размещения SMPTE UMID [1].

BWF Версия 2

Версия 2 – существенная ревизия Версии 1, которая включает метаданные громкости (в соответствии с EBU R 128 [2]) и учитывает публикацию Дополнений 1 – 6 и другой соответствующей документации. Эта версия полностью совместима с Версиями 0 и 1, но пользователи, желающие гарантировать, чтобы их файлы соответствовали требованиям Рекомендации EBU R 128, должны гарантировать, чтобы их системы могли читать и писать метаданные громкости.

Система обозначений

Настоящий документ содержит как нормативный, так и информативный текст.

Весь текст является нормативным, кроме Введения, разделов, отмеченных как «информативные», или отдельных параграфов, начинающихся с «Примечания».

Нормативный текст описывает обязательные или непреложные элементы. Он содержит ключевые слова «должен», «следует» или «можно», определяемые следующим образом:

«Должен» или «не должен»: Указывает требования, которые нужно строго соблюдать и от которых не допускается отклонений для соответствия документу.

«Следует» или «не следует»: Указывает, что один из нескольких вариантов рекомендуется как особенно подходящий, не упоминая и не исключая других.

ИЛИ что определенный ход действий предпочтителен, но не обязателен.

ИЛИ что (в отрицательной форме) определенный вариант или ход действий не рекомендуется, но не запрещается.

«Можно» или «можно не»: Указывает ход действий, допустимый в рамках документа.

По умолчанию означает обязательные (в фразах, содержащих «должен») или рекомендуемые (в фразах, содержащих «следует») предустановки, которые могут быть опционально изменены пользователем или иметь другие опции в продвинутых приложениях. Обязательные установки по умолчанию должны поддерживаться. Поддержка рекомендуемых установок предпочтительна, но не обязательна.

Информативный текст потенциально полезен для пользователя, но не обязателен и может быть исключен, изменен или дополнен, не влияя на нормативный текст. Информативный текст не содержит ключевых слов соответствия.

Совместимая реализация включает все обязательные условия («должен») и все рекомендуемые условия («следует») в случае их реализации. Совместимая реализация не требует реализации опциональных условий («можно»).

Содержание

1. Введение.....	4
1.1 Совместимость версий	5
2. Файл Broadcast Wave Format	5
2.1 Содержание файла Broadcast Wave Format.....	5
2.2 Существующие порции данных, определенные как часть стандарта RIFF	5
2.3 Порция данных Broadcast Audio Extension	5
2.4 Обработка параметров громкости	8
2.5 Другая информация, специфичная для приложений	8
3. Ссылки	9
Приложение A: Формат файлов RIFF WAVE (.WAV)	10
A1. Формат волновых аудиофайлов (WAVE)	10
A1.1 Порция данных формата WAVE	10
A1.2 Категории формата WAVE	11
A2. Формат Pulse Code Modulation (PCM).....	11
A2.1 Упаковка данных для файлов PCM WAVE	11
A2.2 Формат данных выборок	12
A2.3 Примеры файлов PCM WAVE	12
A2.4 Хранение данных WAVE	12
A2.5 Порция данных Fact	13
A2.6 Другие опциональные порции данных	13
A3. Другие типы WAVE	13
A3.1 Общая информация	13
A3.2 Порция данных Fact	13
A3.3 Расширение формата WAVE	13

Спецификация Broadcast Wave Format (BWF)

Формат для файлов аудиоданных в вещании

Комитет EBU	Первый выпуск	Переработка	Переиздание
TC	1997	2001, 2011	

Ключевые слова: Broadcast Wave File, BWF, RIFF WAV, Метаданные, громкость

1. Введение

Broadcast Wave Format (BWF) основан на формате аудиофайлов Microsoft WAVE, который является типом файла, определенным в Microsoft "Resource Interchange File Format", RIFF [3]. Файлы WAVE содержат аудиоданные. Базовый компоновочный блок файлов RIFF – порция данных, содержащая специфическую информацию, поле идентификации и поле размера. Файл RIFF состоит из ряда порций данных.

В BWF к исходному формату WAVE применяются некоторые ограничения. Кроме того, файл BWF включает порцию данных <Broadcast Audio Extension>. Это проиллюстрировано на Рис 1.

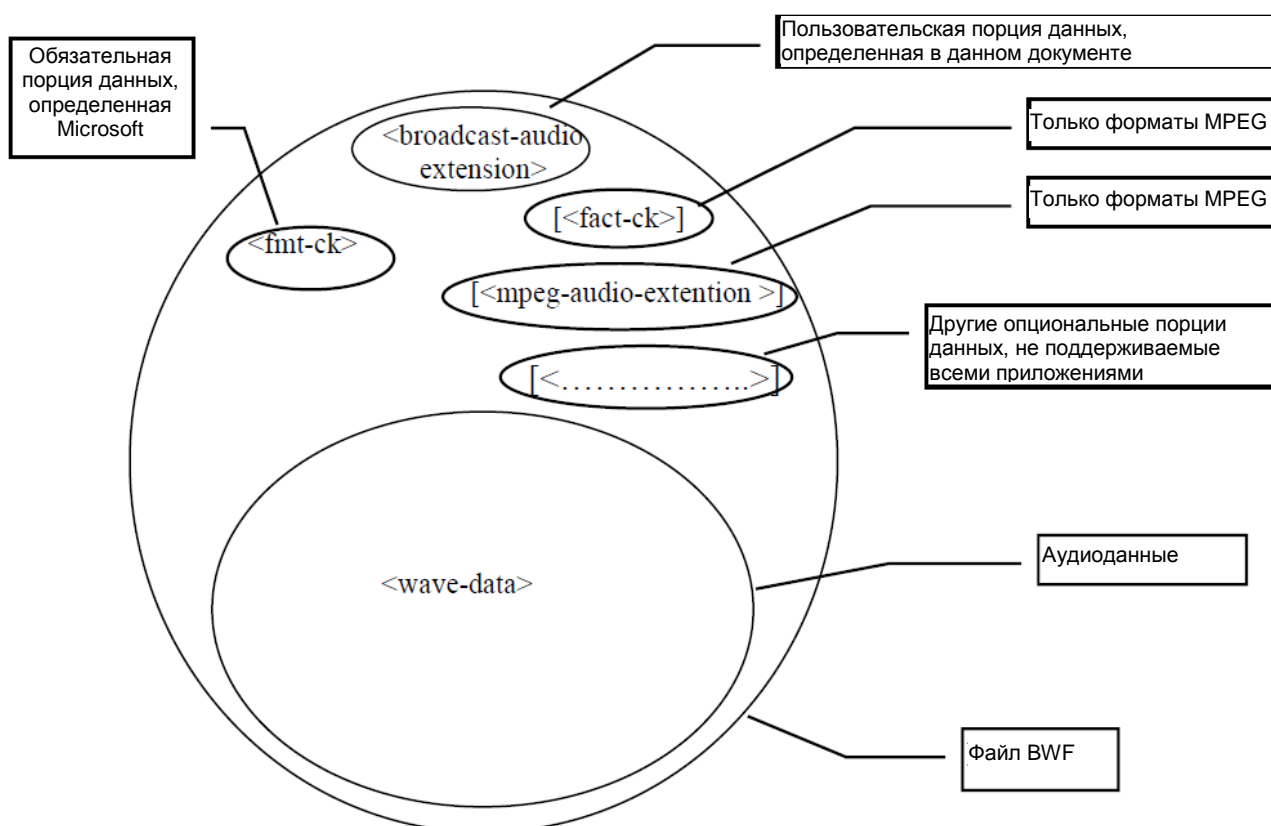


Рис. 1: Файл Broadcast Wave Format

Настоящий документ содержит спецификацию Версии 2 порции broadcast audio extension, которая используется во всех файлах BWF. Кроме того, в *Приложении А* можно найти информацию о форматах PCM Wave и RIFF и о том, как можно расширить RIFF другими типами аудиоданных.

Подробные спецификации расширения BWF другими типами аудиоданных и другими порциями, ратифицированными EBU, опубликованы в Дополнениях к настоящему документу; см. раздел Ссылки.

В частности, EBU ратифицировал Multi-channel Broadcast Wave Format, MBWF, который определяет средство расширения формата RF64 метаданными BWF. Эта разработка позволяет использовать файлы большего размера (более 4 Gb) и вмещает аудиофайлы более чем с двумя каналами [4].

Кроме того, Общество инженеров звукозаписи ратифицировало "Cart Chunk" для систем передачи как AES46 [5].

1.1 Совместимость версий

Версия 0 BWF была опубликована в 1997 г.

Версия 1, опубликованная в июле 2001 г., отличается от Версии 0 лишь тем, что 64 из 254 резервных байтов в Версии 0 используются для SMPTE UMID и поле <Version> соответственно изменено.

Версия 1 обратно совместима с Версией 0. Это значит, что программное обеспечение, разработанное для чтения файлов Версии 0, будет корректно интерпретировать файлы Версии 1, игнорируя лишь поле UMID.

Изменение также совместимо «вверх». Это значит, что программа Версии 1 сможет корректно читать файлы Версии 0. В идеале ПО Версии 1 должно читать поле <Version> для определения присутствия UMID. Однако, если номер Версии не читается, программа прочтет все нули в поле UMID в файле Версии 0. Это не будет являться действительным UMID и будет проигнорировано. Это было заменено в Версии 2 в мае 2011 г.

Версия 2 отличается от Версии 1 лишь тем, что 10 из 190 резервных байтов в Версии 1 используются для информации о громкости файла и поле <Version> соответственно изменено.

Версия 2 обратно совместима с Версиями 1 и 0. Это значит, что программное обеспечение, разработанное для чтения файлов Версии 1 и 0, будет корректно интерпретировать файлы с тем исключением, что программа Версии 0 будет игнорировать UMID и информацию громкости, а программа Версии 1 будет игнорировать информацию громкости. Следовательно, пользователи таких устройств будут терять метаданные, если не будут приняты специальные меры. Кроме того, первые устройства с возможностью BWF не смогут работать с большими файлами RF64 и MBWF и могут не распознавать порции данных, определенные после 2001 г.

Изменение также совместимо «вверх». Это значит, что программа Версии 2 сможет корректно читать файлы Версии 0 и 1. Программа должна читать поле <Version> для определения присутствия UMID и метаданных громкости.

2. Файл Broadcast Wave Format

2.1 Содержание файла Broadcast Wave Format

Файл Broadcast Wave Format должен начинаться с обязательного заголовка Microsoft RIFF “WAVE” и минимум следующих порций данных:

```
<WAVE-form> ->
  RIFF( 'WAVE'
    <broadcast_audio_extension> //информация об аудио последовательности
    <fmt-ck> //Формат аудиосигнала: PCM/MPEG
    [<fact-ck>] //Порция Fact нужна только для форматов MPEG
    [<mpeg_audio_extension>] //Порция MPEG Audio Extension нужна только для
      форматов MPEG
    <wave-data> ) //звуковые данные
```

Любые дополнительные порции данных, присутствующие в файле и не указанные в настоящем документе и Дополнениях к нему, в EBU Tech 3306 или в AES46, считаются приватными. Приложения не должны интерпретировать или использовать эти порции. Таким образом, целостность данных, содержащихся в любых порциях, не указанных в документах или списках порций данных, не гарантирована. Однако BWF-совместимые приложения должны передавать эти порции.

2.2 Существующие порции данных, определенные как часть стандарта RIFF

Стандарт RIFF определен в документах, изданных Microsoft Corporation [3]. Это приложение использует ряд порций данных, которые уже определены. Это следующие порции:

```
fmt-ck
fact-ck
```

Описания этих порций дано для информации в Приложении А.

2.3 Порция данных Broadcast Audio Extension

Дополнительные параметры, необходимые для обмена материалом между вещателями, следует добавлять в специальной порции “Broadcast Audio Extension”, определенной следующим образом:

```

broadcast_audio_extension typedef struct {
    DWORD    ckID;                /* (broadcastextension)ckID=bext. */
    DWORD    ckSize;              /* размер порции расширения */
    BYTE     ckData[ckSize];      /* данные порции */
}
typedef struct broadcast_audio_extension {
    CHAR     Description[256];     /* ASCII : «Описание звуковой последовательности» */
    CHAR     Originator[32];       /* ASCII : «Имя создателя» */
    CHAR     OriginatorReference[32]; /* ASCII : «Код создателя» */
    CHAR     OriginationDate[10];  /* ASCII : «yyyy:mm:dd» */
    CHAR     OriginationTime[8];   /* ASCII : «hh:mm:ss» */
    DWORD    TimeReferenceLow;     /* Первый номер выборки после полуночи, младшее слово */
    DWORD    TimeReferenceHigh;    /* Первый номер выборки после полуночи, старшее слово */
    WORD     Version;              /* Версия BWF; двоичное число без знака */
    BYTE     UMID_0                /* Бинарный байт 0 SMPTE UMID */
    ....
    BYTE     UMID_63              /* Бинарный байт 63 of SMPTE UMID */
    WORD     LoudnessValue;        /* WORD : «Интегрированное значение громкости файла в
    LUFs (умноженное на 100) » */
    WORD     LoudnessRange;        /* WORD : «Диапазон громкости файла в LU (умноженный на
    100) » */
    WORD     MaxTruePeakLevel;     /* WORD : «Максимальный уровень реальных пиковых значе-
    ний файла, выраженный как dBTP (умноженный на 100) » */
    WORD     MaxMomentaryLoudness; /* WORD : «Высшее значение мгновенного уровня громкости
    файла в LUFs (умноженное на 100) » */
    WORD     MaxShortTermLoudness; /* WORD : «Высшее значение краткосрочного уровня громко-
    сти файла в LUFs (умноженное на 100) » */
    BYTE     Reserved[180];        /* 180 байтов, зарезервированных на будущее, установлены
    на "NULL" */
    CHAR     CodingHistory[];      /* ASCII : « История кодирования » */
} BROADCAST_EXT

```

Поле	Описание
<u>Description</u>	Строка ASCII (максимум 256 символов), содержащая свободное описание последовательности. Чтобы помочь приложениям, которые отображают лишь краткое описание, рекомендуется, чтобы в первых 64 знаках содержалось резюме описания, а последние 192 знаков использовались для деталей. Если длина строки менее 256 знаков, за последним должен идти нулевой символ (00).
<u>Originator</u>	Строка ASCII (максимум 32 символов), содержащая имя создателя / продюсера аудиофайла. Если длина строки менее 32 знаков, поле должно заканчиваться нулевым символом.
<u>OriginatorReference</u>	Строка ASCII (максимум 32 символов), содержащая однозначный код, выданный организацией-производителем. Если длина строки менее 32 знаков, поле должно заканчиваться нулевым символом. <i>Примечание: EBU определил формат для поля OriginatorReference. См. Рекомендацию EBU R 99 [6].</i>
<u>OriginationDate</u>	10 символов ASCII, содержащих дату создания аудио последовательности. Формат должен быть « 'год', '-', 'месяц', '-', 'день', » с 4 знаками для года и 2 знаками для остальных пунктов. Год определяется с 0000 по 9999 Месяц определяется с 1 по 12 День определяется с 1 по 28, 29, 30 или 31 Разделитель между пунктами может быть любым, но рекомендуется использовать один из следующих символов: ' ' дефис ' _ ' подчеркик ':' двоеточие ' ' пробел '.' точка

<u>OriginationTime</u>	<p>8 символов ASCII, содержащих время создания аудио последовательности. Формат должен быть « 'час'-'минута'-'секунда' » с 2 знаками в каждом пункте.</p> <p>Часы определяются с 0 по 23. Минуты и секунды определяются с 0 по 59.</p> <p>Разделитель между пунктами может быть любым, но рекомендуется использовать один из следующих символов:</p> <p>'-' дефис '_' подчеркик ':' двоеточие ' ' пробел '.' точка</p>
<u>TimeReference</u>	<p>Эти поля должны содержать тайм-код последовательности. Это 64-битное значение, которое содержит первый номер выборки после полуночи. Число выборок в секунду зависит от частоты дискретизации, которая определяется в поле <nSamplesPerSec> из <format chunk>.</p>
<u>Version</u>	<p>Двоичное число без знака, указывающее версию BWF. Это число особенно актуально для передачи UMID и информации громкости. Для Версии 1 оно должно быть установлено на 0001h, а для Версии 2 – на 0002h.</p>
<u>UMID</u>	<p>64 байта, содержащие UMID (уникальный идентификатор материала) по стандарту SMPTE 330M [1]. Если используется только 32-байтный "basic UMID", последние 32 байта должны быть установлены на ноль. (Длина UMID указана внутренне.)</p>
<u>LoudnessValue</u>	<p>16-битное целое число без знака, равное <i>round</i> (100 x значение интегрированной громкости файла в LUFS).</p>
<u>LoudnessRange</u>	<p>16-битное целое число без знака, равное <i>round</i> (100 x диапазон громкости файла в LU).</p>
<u>MaxTruePeakLevel</u>	<p>16-битное целое число без знака, равное <i>round</i> (100 x максимальный уровень реальных пиковых значений файла в dBTP).</p>
<u>MaxMomentaryLoudness</u>	<p>16-битное целое число без знака, равное <i>round</i> (100 x высшее значение мгновенного уровня громкости файла в LUFS).</p>
<u>MaxShortTermLoudness</u>	<p>16-битное целое число без знака, равное <i>round</i> (100 x высшее значение краткосрочного уровня громкости файла в LUFS).</p> <p><i>Примечание: Термины громкости объясняются в Рекомендации EBU R 128 [2], EBU Tech 3341 [7] и EBU Tech 3342 [8], поддерживаемых EBU Tech 3343 и EBU Tech 3344</i></p>
<u>Reserved</u>	<p>180 байт, зарезервированные для расширения. Если поле Version установлено на 0001h или 0002h, эти 180 байт должны быть установлены на значение NULL (ноль).</p>
<u>CodingHistory</u>	<p>Неограниченные символы ASCII, содержащие набор строк, оканчивающихся CR/LF. Каждая строка должна содержать описание процесса кодирования, примененного к аудиоданным. Каждое новое приложение кодирования должно добавлять новую строку с соответствующей информацией.</p> <p>Эта информация должна содержать тип звука (PCM или MPEG) с его параметрами:</p> <p>PCM : режим (mono, stereo), размер выборки (8, 16 бит) и частота дискретизации, MPEG : частота дискретизации, битрейт, уровень (I или II) и режим (mono, stereo, joint stereo или dual channel),</p> <p>Рекомендуется, чтобы производители кодером обеспечивали строку ASCII для использования в истории кодирования.</p> <p><i>Примечание: EBU определил формат для истории кодирования, который упростит интерпретацию информации в этом поле. См. Рекомендацию EBU R 98 [9].</i></p>
<p><i>Примечание: EBU определил формат для файлов BWF, имеющих более двух аудио каналов, MBWF. См. EBU Tech 3306 [4].</i></p>	

2.4 Обработка параметров громкости

Параметры громкости представлены целыми числами, но сохраняют точность двух десятичных разрядов при умножении на 100 перед округлением. Функция округления, которая должна применяться, определяется следующим образом:

$$\text{целочисленное представление} = \text{целая часть} (x + \text{sgn}(x) \cdot 0.5)$$

где x – представленное значение, умноженное на 100

и где $\text{sgn}()$ – сигнум-оператор. $\text{sgn}(x) = -1$, если $x < 0$; 0 , если $x = 0$; 1 , если $x > 0$.

Этот метод округления общеизвестен как «округление до ближайшего, в сторону от нуля», т.к. если дробная часть числа – 5 (посредине между целыми числами), то округление идет вверх для положительных чисел и вниз для отрицательных.

Примеры

Отрицательные числа:

Переменное значение	Вычисление	Значение, передаваемое в BWF (десятичное / шестнадцатеричное)
-22.644	целое число $[(-22.644 \times 100) + \text{sgn}(-22.644 \times 100) \cdot 0.5]$	-2264/ F728h
-22.645	целое число $[(-22.645 \times 100) + \text{sgn}(-22.645 \times 100) \cdot 0.5]$	-2265/ F727h
-22.646	целое число $[(-22.646 \times 100) + \text{sgn}(-22.646 \times 100) \cdot 0.5]$	-2265/ F727h

Положительные числа:

Переменное значение	Вычисление	Значение, передаваемое в BWF (десятичное / шестнадцатеричное)
12.764	целое число $[(12.764 \times 100) + \text{sgn}(12.764 \times 100) \cdot 0.5]$	1276/ 04FCh
12.765	целое число $[(12.765 \times 100) + \text{sgn}(12.765 \times 100) \cdot 0.5]$	1277/ 04FDh
12.766	целое число $[(12.766 \times 100) + \text{sgn}(12.766 \times 100) \cdot 0.5]$	1277/ 04FDh

Если какой-либо параметр громкости не используется, их 16-битные целые значения должны быть установлены на 7FFFh, которое не входит в диапазон значений параметров.

Для LoudnessValue, MaxTruePeakLevel, MaxMomentaryLoudness и MaxShortTermLoudness диапазон действительных значений – с D8F1h по FFFFh (в соответствии с эквивалентными значениями плавающей точки с -99.99 по -0.01) и с 0000h по 270Fh (с 0.00 по 99.99). Самый старший разряд 16-битного шестнадцатеричного числа – это знаковый разряд; следовательно, значения между 8000h и FFFFh представляют отрицательные числа.

Для LoudnessRange диапазон действительных значений – с 0000h по 270Fh (с 0.00 по 99.99).

Следовательно, в случае 7FFFh известно, что данный параметр надо игнорировать.

Если какие-либо параметры имеют значения вне действительного диапазона (не только 7FFFh) при чтении порции данных, то их тоже следует игнорировать.

2.5 Другая информация, специфичная для приложений

EBU определил другие порции данных для передачи или указания на данные, специфичные для определенных приложений, например, многоканальный звук, Dolby Metadata или любые данные XML. См. раздел Ссылки или веб-сайт EBU technical (<http://tech.ebu.ch>).

3. Ссылки

- [1] SMPTE ST 330:2004 Television – Unique Material Identifier (UMID)
- [2] EBU R 128 Loudness normalisation and permitted maximum level of audio signals
- [3] MS RIFF Microsoft Resource Interchange File Format, RIFF – part of the Multimedia Registration Kit, rev 3.0: <http://support.microsoft.com/kb/q120253/>
- [4] EBU Tech 3306 MBWF / RF64 : An Extended File Format for Audio
- [5] AES46 AES standard for network and file transfer of audio – Audio-file transfer and exchange – Radio traffic audio delivery extension to the broadcast-WAVE-file format
- [6] EBU R 99 'Unique' Source Identifier (USID) for use in the OriginatorReference field of the Broadcast Wave Format
- [7] EBU Tech 3341 Loudness Metering: 'EBU Mode' metering to supplement Loudness normalisation in accordance with EBU R 128
- [8] EBU Tech 3342 Loudness Range: A descriptor to supplement Loudness normalisation in accordance with EBU R 128
- [9] EBU R 98 Format for CodingHistory field in Broadcast Wave Format files, BWF

Дальнейшее чтение

- EBU R 85: Use of the Broadcast Wave Format for the Exchange of Audio Data Files
- EBU R 111: Multichannel Use of the BWF Audio File Format (MBWF)
- EBU Tech 3285 Supplement 1: MPEG Audio
- EBU Tech 3285 Supplement 2: Capturing Report (includes Quality and Cue-sheet data)
- EBU Tech 3285 Supplement 3: Peak-Envelope Chunk
- EBU Tech 3285 Supplement 4: Link Chunk
- EBU Tech 3285 Supplement 5: XML Data Chunk
- EBU Tech 3285 Supplement 6: Dolby Metadata Chunk
- EBU Tech 3343: Practical Guidelines for Production and Implementation in accordance with EBU R 128
- EBU Tech 3344: Practical Guidelines for Distribution of Programmes in accordance with EBU R 128

Приложение А: Формат файлов RIFF WAVE (.WAV)

Информация в этом приложении взята из спецификации формата файлов Microsoft RIFF. Были внесены небольшие поправки для ясности. **Это включено только для информации.**

Полную информацию см. в последней версии Microsoft Multimedia Registration Kit [3].

[Примечание EBU: Пояснительные записки EBU выделены курсивом в квадратных скобках]

A1. Формат волновых аудиофайлов (WAVE)

Форма WAVE определяется следующим образом. Программы должны ожидать (и игнорировать) любые неизвестные порции данных, как и во всех формах RIFF. Однако **<fmt-ck>** всегда должно идти перед **<wave-data>**, и обе эти порции в файле WAVE обязательны.

```
<WAVE-form> ->
  RIFF( 'WAVE'
    <fmt-ck>           // Формат
    [<fact-ck>]       // Порция Fact
    [<other-ck>]      // Другие опциональные порции
    <wave-data> )     // Данные формы сигнала
```

Порции данных WAVE описаны в следующих разделах.

A1.1 Порция данных формата WAVE

Порция **<fmt-ck>** формата WAVE указывает формат **<wave-data>**. **<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 дискретизации
    DWORD         nAvgBytesPerSec; // Для информации буфера
    WORD          nBlockAlign;    // Размер блока данных
  }
```

Поля в части **<common-fields>** порции данных следующие:

Поле	Описание
wFormatTag	Число, показывающее категорию формата файла WAVE. Содержание части <format-specific-fields> порции 'fmt' и интерпретация данных о форме сигнала зависят от этого значения.
nchannels	Число каналов, представленных в данных о форме сигнала: 1 – моно, 2 – стерео. <i>[Примечание EBU: EBU определил Multi-channel Broadcast Wave Format [4], где требуется более двух каналов звука.]</i>
nSamplesPerSec	Частота дискретизации (выборка в секунду), с которой должен воспроизводиться каждый канал.
nAvgBytesPerSec	Среднее число байтов в секунду, с которым должны передаваться данные о форме сигнала. Программа воспроизведения может вычислить размер буфера с помощью этого значения.
nBlockAlign	Выравнивание блоков (в байтах) в данных о форме сигнала. Программа воспроизведения должна обрабатывать множество байтов <nBlockAlign> одновременно, поэтому значение <nBlockAlign> может использоваться для выравнивания буфера.

<format-specific-fields> состоит из нуля или более параметров. Параметры зависят от категории формата WAVE, как описано ниже. Программа воспроизведения должна быть написана с учетом возможности (и игнорирования) любых неизвестных параметров **<format-specific-fields>**, которые встречаются в конце этого поля.

A1.2 Категории формата WAVE

Категория формата файла WAVE указывается значением поля <wFormatTag> в порции 'fmt'. Представление данных в <wave-data> и содержание <format-specific-fields> в порции 'fmt' зависят от категории формата.

Среди нынешних открытых, непатентованных категорий формата WAVE есть следующие:

wFormatTag	Значение	Категория формата
WAVE_FORMAT_PCM	(0001h)	Формат Microsoft Pulse Code Modulation (PCM)
WAVE_FORMAT_MPEG	(0050h)	MPEG-1 Audio (только аудио)

[Примечание EBU: Хотя в Microsoft зарегистрированы и другие форматы WAVE, с BWF в настоящее время используются только вышеперечисленные. Детали формата PCM WAVE даны в следующем разделе. Общая информация о других форматах WAVE дана в разделе 3. Детали формата MPEG WAVE даны в Дополнении 1 к настоящему документу, а детали расширения Broadcast Wave Format до многоканального звука – в EBU Tech 3306. Другие форматы WAVE могут быть определены в будущих Дополнениях.]

A2. Формат Pulse Code Modulation (PCM)

Если поле <wFormatTag> в <fmt-ck> установлено на WAVE_FORMAT_PCM, то данные о форме сигнала состоят из выборок, представленных в формате импульсно-кодовой модуляции (PCM). Для данной формы сигнала PCM <format-specific-fields> определяется следующим образом:

<PCM-format-specific> ->

```
struct{
    WORD nBitsPerSample;    // Размер выборки
}
```

Поле <nBitsPerSample> указывает число битов данных, используемых для представления каждой выборки каждого канала. Если каналов несколько, размер выборки одинаков для каждого канала.

Для данных PCM поле <nAvgBytesPerSec> в порции 'fmt' должно быть равно следующей формуле с округлением до ближайшего целого числа:

$$\frac{nChannels \times nSamplesPerSecond \times nBitsPerSample}{8}$$

Поле <nBlockAlign> должно быть равно следующей формуле с округлением до ближайшего целого числа:

$$\frac{nChannels \times nBitsPerSample}{8}$$

[Примечание EBU: Эта формула не всегда дает правильный ответ. Строго говоря, число байтов на выборку (nBitsPerSample / 8) следует сначала округлить. Затем это целое число следует умножить на <nChannels> (которое всегда является целым числом) для получения <nBlockAlign>. Его, в свою очередь, нужно умножить на <nSamplesPerSecond> для получения <nAvgBytesPerSec>.]

A2.1 Упаковка данных для файлов PCM WAVE

В одноканальном файле WAVE выборки хранятся последовательно. В стерео файлах WAVE канал 0 представляет левый канал, а канал 1 – правый. Преобразование позиции громкоговорителей для файла BWF более чем с двумя каналами определено в EBU Tech 3306. В многоканальных файлах WAVE выборки перемежаются.

Следующие диаграммы показывают упаковку данных для 8-битных моно и стерео файлов WAVE:

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

Упаковка данных для 8-битного моно PCM

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

Упаковка данных для 8-битного стерео PCM

Следующие диаграммы показывают упаковку данных для 16-битных моно и стерео файлов WAVE:

Выборка 1	Выборка 2	Выборка 3	Выборка 4
Канал 0 младший байт	Канал 0 старший байт	Канал 0 младший байт	Канал 0 старший байт

Упаковка данных для 16-битного моно PCM

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

Упаковка данных для 16-битного стерео PCM

A2.2 Формат данных выборок

Каждая выборка содержится в целом числе *i*. Размер *i* – наименьшее число байтов, необходимое для содержания определенного размера выборки. Первым сохраняется самый младший байт. Биты, представляющие амплитуду выборки, хранятся в самых старших разрядах *i*, а остальные биты установлены на ноль.

Например, если размер выборки (записанный в <nBitsPerSample>) 12 бит, то каждая выборка хранится в двухбайтном целом числе. Четыре самых младших разряда первого (самого младшего) байта установлены на ноль. Формат данных и максимальные и минимальные значения выборок формы сигнала PCM различного размера следующие:

Размер выборки	Формат данных	Максимальное значение	Минимальное значение
1 – 8 бит	Целое число без знака	255 (FFh)	0
9 или более бит	Целое число со знаком <i>i</i>	Наибольшее положительное значение <i>i</i>	Наименьшее отрицательное значение <i>i</i>

Например, максимальное, минимальное и среднее значения для 8-битных и 16-битных данных формы сигнала PCM следующие:

Формат	Максимальное значение	Минимальное значение	Среднее значение
8-bit PCM	255 (FFh)	0	128 (80h)
16-bit PCM	32767 (7FFFh)	-32768 (-8000h)	0

A2.3 Примеры файлов PCM WAVE

Пример файла PCM WAVE с частотой дискретизации 11.025 kHz, моно, 8 бит на выборку:

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

Пример файла PCM WAVE с частотой дискретизации 22.05 kHz, стерео, 8 бит на выборку:

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

Пример файла PCM WAVE с частотой дискретизации 44.1 kHz, моно, 20 бит на выборку:

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

A2.4 Хранение данных WAVE

<wave-data> содержит данные о форме сигнала. Он определяется следующим образом:

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

A2.5 Порция данных Fact

Порция <fact-ck> хранит важную информацию о содержании файла WAVE. Эта порция определяется следующим образом:

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

Эта порция данных не требуется для файлов PCM.

Порция "fact" будет расширяться для включения любой другой информации, необходимой для будущих форматов WAVE. Дополнительные поля появятся после поля <dwFileSize>. Приложения могут использовать поле размера порции данных для определения присутствия полей.

A2.6 Другие опциональные порции данных

Определен ряд других порций данных для использования в формате WAVE. Детали этих порций даны в спецификации формата WAVE и в ее обновлениях.

[Примечание EBU: Формат WAVE может поддерживать другие опциональные порции данных, которые могут быть включены в формат WAVE для передачи определенной информации. Как говорилось в п.2.1, порции, присутствующие в файле Broadcast Wave Format и не указанные в настоящем документе, Tech 3306, Дополнениях к ним и в AES46, считаются приватными и будут игнорироваться приложениями, которые не могут их интерпретировать.]

A3. Другие типы WAVE

[Примечание EBU: следующая информация взята из Microsoft Multimedia Registration Kit [3]. Она содержит необходимые расширения базового файла WAVE (для звука PCM) для включения других типов формата WAVE.]

A3.1 Общая информация

Все заново определенные типы WAVE должны содержать и <fact chunk>, и расширенное описание формата формы сигнала в порции данных <fmt-ck>. WAVE файлы RIFF типа WAVE_FORMAT_PCM не должны иметь лишней порции данных и расширенного описания формата формы сигнала.

A3.2 Порция данных Fact

Эта порция хранит обусловленную файлом информацию о содержании файла WAVE. В настоящее время она указывает длину файла в выборках.

[Примечание EBU: См. п. A2.5]

A3.3 Расширение формата WAVE

Структура расширенного формата, добавленная в <fmt-ck>, используется для определения всех данных формы сигнала не-PCM формата и описывается следующим образом. Общая структура расширенного формата используется для всех форматов не-PCM.

```
typedef struct waveformat_extended_tag {
    WORD    wFormatTag;          /* тип формата */
    WORD    nChannels;          /* число каналов (т.е. моно, стерео...) */
    DWORD   nSamplesPerSec;     /* частота дискретизации */
    DWORD   nAvgBytesPerSec;    /* для информации буфера */
    WORD    nBlockAlign;        /* размер блока данных */
    WORD    wBitsPerSample;     /* Число бит на выборку моно данных */
    WORD    cbSize;             /* Число байтов лишнего размера */
} WAVEFORMATEX;
```

Поле	Примечания
wFormatTag	Определяет тип файла WAVE.
nChannels	Число каналов в форме сигнала, 1 – моно, 2 – стерео
nSamplesPerSec	Частота дискретизации файла. Должна быть 48000 или 44100 и т.д. Эта частота также используется полем размера выборки в порции fact для определения длины данных во времени.
nAvgBytesPerSec	Средняя скорость передачи данных. Программа воспроизведения может вычислить размер буфера с помощью значения <nAvgBytesPerSec>.
nBlockAlign	Выравнивание блоков (в байтах) данных в <data-ck>. Программа воспроизведения должна обрабатывать множество байтов данных <nBlockAlign> одновременно, поэтому значение <nBlockAlign> может использоваться для выравнивания буфера.
wBitsPerSample	Это число битов на выборку в канале данных. Предполагается, что каждый канал имеет одинаковое разрешение выборок. Если это поле не нужно, оно должно быть установлено на ноль.
cbSize	Размер в байтах дополнительной информации в заголовке формата WAVE, не включая размер структуры WAVEFORMATEX.
<i>[Примечание EBU:</i>	<i>поля после поля <cbSize> содержат специфическую информацию, необходимую для формата WAVE, определенного в поле <wFormatTag>. Любые форматы WAVE, которые можно использовать в BWF, указаны в отдельных Дополнениях к настоящему документу, опубликованных EBU.]</i>