

RECOMENDACIÓN UIT-R BR.1352-3

**Formato de fichero para el intercambio de materiales
de programas de audio con metadatos por medios
basados en la tecnología de la información**

(Cuestión UIT-R 58/6)

(1998-2001-2002-2007)

Ámbito de aplicación

La presente Recomendación contiene las especificaciones relativas al segmento de extensión de audio de radiodifusión¹ y su utilización con la codificación MIC, y datos de audio MPEG-1 o MPEG-2. También se incluye en esta Recomendación información básica sobre el formato RIFF y la manera en que puede ampliarse a otros tipos de datos de audio.

La Asamblea de Radiocomunicaciones de la UIT,

considerando

- a) que los medios de almacenamiento basados en la tecnología de la información, incluidos los discos y cintas de datos, serán utilizados en todos los campos de la producción de audio para la radiodifusión, a saber, edición no lineal, reproducción a partir de la emisión y archivos;
- b) que esta tecnología ofrece ventajas importantes desde el punto de vista de la flexibilidad de funcionamiento, flujo de producción y automatización de la estación y que, en consecuencia, es interesante para la mejora de los estudios existentes y el diseño de nuevas instalaciones de estudios;
- c) que la adopción de un solo formato de fichero para el intercambio de señales simplificaría considerablemente la interoperabilidad de los equipos y estudios distantes, y facilitaría la integración deseable de la edición, la reproducción a partir de la emisión y el archivo;
- d) que se debe incluir un conjunto mínimo de información relacionada con la radiodifusión en el fichero para documentar la señal audio;
- e) que, con miras a asegurar la compatibilidad entre aplicaciones con complejidades diferentes, se debe acordar un conjunto mínimo de funciones, comunes a todas las aplicaciones capaces de tratar el formato de fichero recomendado;
- f) que la Recomendación UIT-R BS.646 define el formato de audio digital utilizado en la producción de audio para radiodifusión sonora y de televisión;
- g) que la necesidad de intercambiar materiales de audio se plantea también cuando se emplean los sistemas de codificación de las Normas ISO/CEI 11172-3 e ISO/CEI 13818-3 para comprimir la señal;

¹ Un segmento (chunk) es el componente básico de un fichero en el formato de fichero de intercambio de recursos (RIFF) de Microsoft ®.

- h) que la compatibilidad con los formatos de ficheros comerciales actualmente disponibles podría minimizar los esfuerzos de la industria para aplicar este formato en los equipos;
- j) que un formato normalizado para la información de historial de codificación simplificaría el empleo de la información tras el intercambio de programas;
- k) que la calidad de la señal de audio está influida por el tratamiento que haya tenido la señal, especialmente por la utilización decodificación y decodificación no lineales durante los procesos de reducción binaria,

recomienda

- 1 que, para el intercambio de programas de audio por medios basados en la tecnología de la información, los parámetros de la señal audio, la frecuencia de muestreo, la resolución de codificación y la preacentuación se fijen de acuerdo con las partes pertinentes de la Recomendación UIT-R BS.646;
- 2 que se utilice el formato de fichero especificado en el Anexo 1 para el intercambio de programas de audio en formato modulación por impulsos codificados MIC lineal por los medios basados en la tecnología de la información;
- 3 que cuando las señales audio sean codificadas aplicando los sistemas de codificación de las Normas ISO/CEI 11172-3 o ISO/CEI 13818-3, se utilice el formato de fichero especificado en el Anexo 1 y complementado en el Anexo 2 para el intercambio de programas de audio por los medios basados en la tecnología de la información²;
- 4 que, cuando se utilice el formato de fichero especificado en los Anexos 1 y/o 2 para cursar información sobre el material de audio recopilado y reordenado por una estación de trabajo audio digital (DAW, *digital audio workstation*), los metadatos cumplirán las especificaciones detalladas en el Anexo 3.

Anexo 1

Especificación del formato de onda de radiodifusión

Un formato para ficheros de datos de audio en radiodifusión

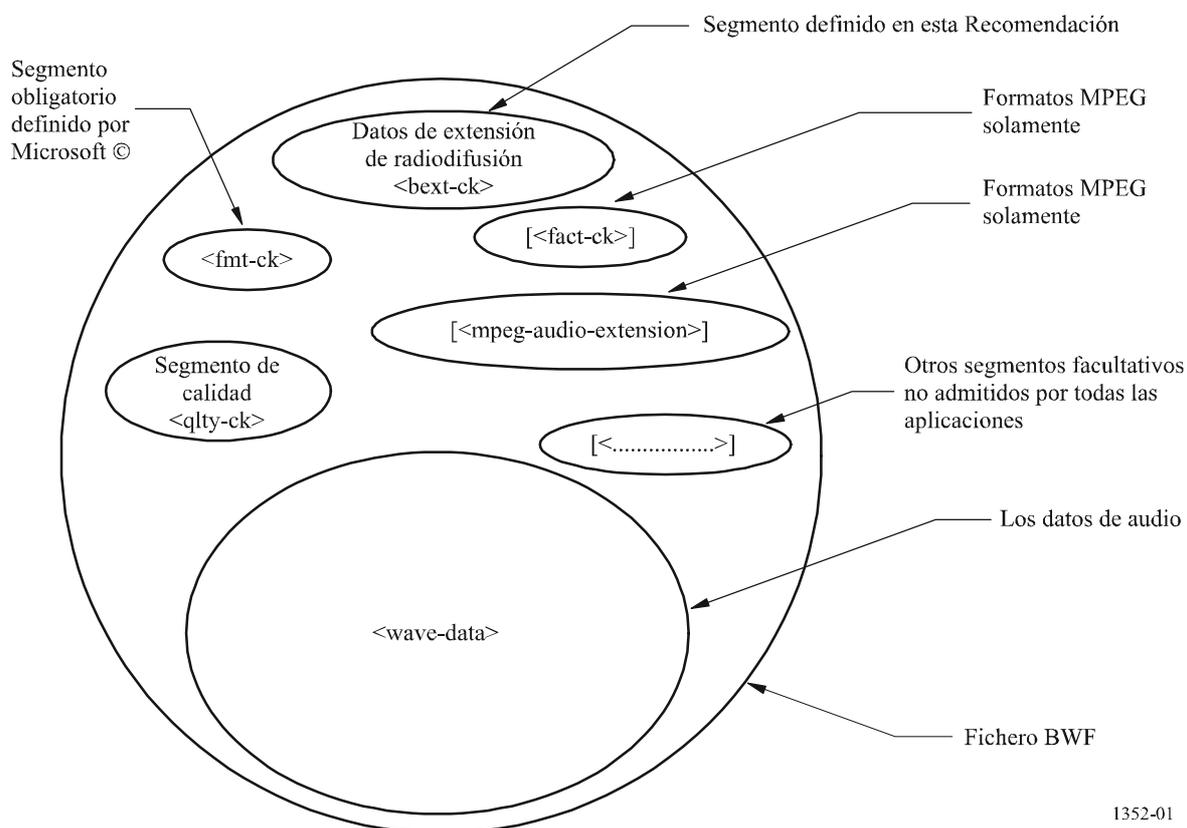
1 Introducción

El formato de onda de radiodifusión (BWF, *broadcast wave format*), se basa en el formato de fichero audio WAVE de Microsoft[®] que es un tipo de fichero especificado en el «Formato de fichero de intercambio de recursos» (RIFF, *Resource Interchange File Format*) de Microsoft[®]. Los ficheros WAVE contienen específicamente datos de audio. El componente básico del formato de fichero RIFF, denominado segmento (*chunk*), contiene un grupo de piezas de información estrechamente relacionadas. Consiste en un identificador de segmento, un valor entero que representa la longitud en octetos del segmento y la información. Un fichero RIFF se compone de una colección de segmentos.

² Se ha reconocido que una Recomendación en ese sentido podría penalizar a diseñadores que utilizan ciertas plataformas informáticas.

Para el BWF, se aplican algunas restricciones al formato WAVE original. Además, el fichero BWF incluye un segmento de extensión de audio de radiodifusión (<Broadcast audio extension>), que se ilustra en la Fig. 1 siguiente.

FIGURA 1
Fichero BWF



1352-01

Este Anexo contiene la especificación del segmento de extensión de audio de radiodifusión que se utiliza en todos los ficheros BWF. Además, en el Apéndice 1 figura información sobre el formato RIFF básico y cómo se puede ampliar a otros tipos de datos de audio. Los detalles del formato de onda con MIC figuran también en el Apéndice 1. Las especificaciones detalladas de la extensión a otros tipos de datos de audio y metadatos se incluyen en los Anexos 2 y 3 a esta Recomendación.

1.1 Disposiciones normativas

La observancia de esta Recomendación es voluntaria. Ahora bien, la Recomendación puede contener ciertas disposiciones obligatorias (para asegurar, por ejemplo, la aplicabilidad o la interoperabilidad), por lo que la observancia se consigue con el cumplimiento exacto y puntual de todas las disposiciones obligatorias.

La obligatoriedad de un elemento preceptivo o requisito se expresa mediante las frases «tener que, haber de, hay que + infinitivo» o el verbo principal en tiempo futuro simple de mandato, en modo afirmativo o negativo. El hecho de que se utilice esta formulación no entraña que la observancia se imponga a ninguna de las partes.

2 Fichero de formato de onda de radiodifusión

2.1 Contenido de un fichero de formato de onda de radiodifusión

Un fichero de formato de onda de radiodifusión (BWF) comenzará con el encabezamiento obligatorio Microsoft® RIFF «WAVE» y por lo menos los segmentos siguientes:

<WAVE-form>	
RIFF('WAVE')	
<fmt-ck>	/* Formato de la señal audio: MIC/MPEG */
<broadcast_audio_extension>	/* la información sobre la secuencia audio */
<universal broadcast audio extension>	/* el segmento ubxt se requiere sólo para admitir un lenguaje de múltiples bytes */
<fact-ck>	/* el segmento ampliado se requiere sólo para los formatos MPEG */
<mpeg_audio_extension>	/* el segmento de extensión de audio MPEG se requiere sólo para los formatos MPEG */
<wave-data>)	/* datos radiofónicos */
<quality-chunk>	/* se requiere sólo cuando se debe obtener información relativa a sucesos pertinentes que afectan a la calidad */

NOTA – Es posible que en el fichero estén presentes otros segmentos. Algunos de ellos pueden estar fuera del alcance de la presente Recomendación. Las aplicaciones pueden o no interpretar o utilizar estos segmentos, de modo que es imposible garantizar la integridad de los datos contenidos en ese tipo de segmento desconocidos. Sin embargo, las aplicaciones homologadas transferirán de manera transparente los segmentos desconocidos.

2.2 Segmentos existentes definidos como parte de la norma RIFF

La norma RIFF se define en documentos publicados por Microsoft®³ Corporation. Esta aplicación utiliza varios segmentos que ya están definidos, a saber:

fmt-ck (segmento de formato)
fact-ck (segmento ampliado)

En el Apéndice 1 al Anexo 1 figuran las descripciones actuales de estos segmentos para información.

2.3 Segmento de extensión de audio de radiodifusión⁴

Los parámetros suplementarios necesarios para el intercambio de material entre entidades de radiodifusión se añaden en un segmento específico «extensión de audio de radiodifusión» que se define como sigue:

```
broadcast_audio_extension typedef struct {
    DWORD    ckID;           /* (broadcastextension)ckID=bext. */
    DWORD    ckSize;        /* tamaño de segmento de extensión */
    BYTE     ckData[ckSize]; /* datos del segmento */
}
```

³ Formato de fichero de intercambio de recursos (RIFF) de Microsoft, disponible (2005-12) en http://www.tactilemedia.com/info/MCI_Control_Info.html.

⁴ Para la definición de segmento ubxt, que indica la información legible para el hombre del segmento bext en un conjunto de caracteres de múltiples bytes, véase el § 2.4.

```

typedef struct broadcast_audio_extension {
CHAR Description[256];           /* ASCII: «Descripción de la secuencia sonora» */
CHAR Originator[32];           /* ASCII: «Nombre del creador» */
CHAR OriginatorReference[32];  /* ASCII: «Referencia del creador» */
CHAR OriginationDate[10];      /* ASCII: «yyyy:mm:dd» */
CHAR OriginationTime[8];       /* ASCII: «hh:mm:ss» */
DWORD TimeReferenceLow;        /* Primer cómputo de muestras desde medianoche,
                               palabra baja */
DWORD TimeReferenceHigh;       /* Primer cómputo de muestras desde medianoche,
                               palabra alta */
WORD Version;                  /* Versión del BWF; número binario sin signo */
BYTE UMID_0;                   /* Byte 0 binario de SMPTE UMID */
BYTE UMID_63;                  /* Byte 63 binario de SMPTE UMID */
CHAR Reserved[190];           /* 190 bytes, reservado para uso futuro,
                               puesto a .NULL. */
CHAR CodingHistory[],          /* ASCII: «Historial de codificación» */
} BROADCAST_EXT,

```

Campo	Descripción
Description	<p>Cadena ASCII (256 caracteres como máximo) que contiene una descripción libre de la secuencia. Para asistir a las aplicaciones que sólo visualizan una breve descripción, se recomienda que un resumen de la descripción figure en los primeros 64 caracteres y que los últimos 192 caracteres se utilicen para detalles.</p> <p>Si la longitud de la cadena es menor que 256 caracteres, el último va seguido por un carácter nulo. (0x00)</p>
Originator	<p>Cadena ASCII (32 caracteres como máximo) que contiene el nombre del creador/productor del fichero audio. Si la longitud de la cadena es menor que 32 caracteres, el campo es terminado por un carácter nulo. (0x00)</p>
OriginatorReference	<p>Cadena ASCII (32 caracteres como máximo) que contiene una referencia unívoca asignada por la organización creadora. Si la longitud de la cadena es menor que 32 caracteres, el campo es terminado por un carácter nulo. (0x00)</p> <p>Un formato estándar para la información del identificador de fuente «único» (USID, «Unique» Source Identifier), para utilizar en el campo OriginatorReference se ofrece en el Apéndice 3 al Anexo 1</p>
OriginationDate	<p>10 caracteres ASCII que contienen la fecha de creación de la secuencia audio. El formato es «',año',-',mes',-',día,'» con 4 caracteres para el año y 2 caracteres para los otros ítems.</p> <p>Año se define de 0000 a 9999</p> <p>Mes se define de 1 a 12</p> <p>Día se define de 1 a 31</p> <p>El separador entre los ítems debe ser un guión conforme a la Norma ISO 8601. Ciertas implementaciones tradicionales pueden utilizar ‘_’ subrayado ‘:’ dos puntos ‘ ’ espacio ‘.’ punto; el equipo de reproducción debe reconocer estos caracteres separadores.</p>

OriginationTime	<p>8 caracteres ASCII que contienen la hora de creación de la secuencia audio. El formato es «‘hora,’-‘,minuto,’-‘,segundo’» con 2 caracteres por ítem.</p> <p>Hora se define de 0 a 23.</p> <p>Minuto y segundo se definen de 0 a 59.</p> <p>El separador entre los ítems debe ser un guión conforme a la Norma ISO 8601. Ciertas implementaciones tradicionales pueden utilizar ‘_’ subrayado ‘:’ dos puntos ‘ ’ espacio ‘.’ punto; el equipo de reproducción debe reconocer estos caracteres separadores.</p>
TimeReference	<p>Este campo contiene el código temporal de la secuencia. Es un valor de 64 bits que contiene el primer cómputo de muestras desde medianoche. El número de muestras por segundo depende de la frecuencia de muestreo que se define en el campo <nSamplesPerSec> (número de muestras por segundo) de <fmt-ck> (segmento de formato).</p>
Version	<p>Un número binario sin signo que indica la versión del BWF. En la versión 1, se pone a 0x0001.</p>
UMID	<p>64 bytes que contienen un UMID ampliado definido por SMPTE 330M. Si se utiliza un UMID básico de 32 bytes, los últimos 32 bytes se rellenarán con ceros. Si no se dispone de ningún UMID, los 64 bytes se rellenarán con ceros.</p> <p>NOTA – La longitud del UMID se codifica al comienzo del propio UMID.</p>
Reserved	<p>190 bytes reservados para extensión. Estos 190 bytes deben ponerse a cero.</p>
CodingHistory	<p>Bloque de caracteres ASCII de tamaño variable que comprende 0 o más cadenas, cada una de ellas terminada por <CR><LF>. El primer carácter no utilizado será un carácter nulo (0x00). Cada cadena debe contener una descripción del proceso de codificación aplicado a los datos de audio.</p> <p>Cada nueva aplicación de codificación tiene que añadir una nueva cadena con la información apropiada.</p> <p>En el Apéndice 2 del Anexo 1 figura un formato normalizado para la información del historial de codificación.</p> <p>Esta información debe contener el tipo de sonido (MIC o MPEG) con sus parámetros específicos:</p> <p>MIC: modo (monofónico, estereofónico), tamaño de la muestra (8, 16 bits) y frecuencia de muestreo.</p> <p>MPEG: frecuencia de muestreo, velocidad binaria, capa (I o II) y el modo (monofónico, estereofónico, estereofónico asociado o canal dual).</p> <p>Se recomienda que los fabricantes de los codificadores proporcionen una cadena ASCII para utilizarla en la historia de codificación.</p>

2.4 Segmento de extensión de audio de radiodifusión universal

La información contenida en el segmento de extensión de audio de radiodifusión (bext) definido en el § 2.3 puede ser transmitido también mediante un segmento específico llamado «extensión de audio de radiodifusión universal» o segmento «ubxt», que indica la información legible para el hombre del segmento bext en lenguajes de múltiples bytes. La estructura básica de este segmento de metadatos es la misma que la correspondiente al segmento bext. Cuatro ítems legibles para el

hombre, uDescription, uOriginator, uOriginatorReference y uCodingHistory, se describen en UTF-8 (*formato de transformación único de 8 bits*) y no ASCII. Los primeros tres ítems tienen 8 veces el tamaño de los datos de los ítems correspondientes en el segmento bext. La estructura del segmento ubxt se define como sigue:

```
typedef struct chunk_header {
    DWORD   ckID;                /* (extensión de radiodifusión universal)ckID=ubxt */
    DWORD   ckSize;              /* tamaño del segmento de extensión */
    BYTE   ckData[ckSize];      /* datos del segmento */
} CHUNK_HEADER;

typedef struct universal_broadcast_audio_extension {
    BYTE   uDescription[256*8];  /* UTF-8 : «Descripción de la secuencia sonora» */
    BYTE   uOriginator[32*8];   /* UTF-8 : «Nombre del creador» */
    BYTE   uOriginatorReference[32*8]; /* UTF-8 : «Referencia del creador» */
    CHAR   OriginationDate[10]; /* ASCII : «yyyy:mm:dd» */
    CHAR   OriginationTime[8];  /* ASCII : «hh:mm:ss» */
    DWORD   TimeReferenceLow;   /* Primer cómputo de muestras desde medianoche,
                                palabra baja */
    DWORD   TimeReferenceHigh;  /* Primer cómputo de muestras desde medianoche,
                                palabra alta */

    WORD   Version;            /* Versión del BWF; número binario sin signo */
    BYTE   UMID_0;             /* Byte 0 binario de SMPTE UMID */...
    BYTE   UMID_63;           /* Byte 63 binario de SMPTE UMID */
    CHAR   Reserved[190];     /* 190 bytes, reservado para uso futuro, puesto a
                                «NULL» */

    BYTE   uCodingHistory[];   /* UTF-8 : «Historial de codificación» */
} UNIV_BROADCAST_EXT;
```

Campo	Descripción
uDescription	Cadena UTF-8, 2 048 bytes o menos, que contiene una descripción de la secuencia. Si no se dispone de datos o si la longitud de la cadena es menor que 2 048 bytes, el primer byte no utilizado será un carácter nulo (0x00)
uOriginator	Cadena UTF-8, 256 bytes o menos, que contiene el nombre del creador del fichero audio. Si no se dispone de datos o si la longitud de la cadena es menor que 256 bytes, el primer byte no utilizado será un carácter nulo (0x00).
uOriginatorReference	Cadena UTF-8, 256 bytes o menos, que contiene una referencia asignada por la organización creadora. Si no se dispone de datos o si la longitud de la cadena es menor que 256 bytes, el primer byte no utilizado será un carácter nulo (0x00).

OriginationDate	<p>10 caracteres ASCII que contienen la fecha de creación de la secuencia audio. El formato es « ‘,año’,-,’mes,’-‘,día,’ » con 4 caracteres para el año y 2 caracteres para los otros ítems.</p> <p>Año se define de 0000 a 9999</p> <p>Mes se define de 1 a 12</p> <p>Día se define de 1 a 31</p> <p>El separador entre los ítems debe ser un guión conforme a la Norma ISO 8601. Ciertas implementaciones tradicionales pueden utilizar ‘_’ subrayado, ‘:’ dos puntos ‘ ’ espacio, ‘.’ punto; el equipo de reproducción debe reconocer estos caracteres separadores.</p>
OriginationTime	<p>8 caracteres ASCII que contienen la hora de creación de la secuencia audio. El formato es « ‘hora,’-‘,minuto,’-‘,segundo’ » con 2 caracteres por ítem.</p> <p>Hora se define de 0 a 23.</p> <p>Minuto y segundo se definen de 0 a 59.</p> <p>El separador entre los ítems debe ser un guión conforme a la Norma ISO 8601. Ciertas implementaciones tradicionales pueden utilizar ‘_’ subrayado, ‘:’ dos puntos ‘ ’ espacio, ‘.’ punto; el equipo de reproducción debe reconocer estos caracteres separadores.</p>
TimeReference	<p>Este campo contiene el código temporal de la secuencia. Es un valor de 64 bits que contiene el primer cómputo de muestras desde medianoche. El número de muestras por segundo depende de la frecuencia de muestreo que se define en el campo <nSamplesPerSec> (número de muestras por segundo) de <fmt-ck> (segmento de formato).</p>
Version	<p>Un número binario sin signo que indica la versión del BWF. En la Versión 1, se pone a 0x0001.</p>
UMID	<p>64 bytes que contienen un UMID ampliado definido por SMPTE 330M. Si se utiliza un UMID básico de 32 bytes, los últimos 32 bytes se rellenarán con ceros. Si no se dispone de ningún UMID, los 64 bytes se rellenarán con ceros.</p> <p>NOTA – La longitud del UMID se codifica al comienzo del propio UMID.</p>
Reserved	<p>190 bytes reservados para extensión. Estos 190 bytes deben ponerse a cero.</p>
uCoding History	<p>Bloque de caracteres UTF-8 de tamaño variable que comprende 0 o más cadenas, cada una de ellas terminadas por <CR><LF>. El primer byte no utilizado será un carácter nulo (0x00).</p> <p>Cada cadena debe contener una descripción del proceso de codificación aplicado a los datos de audio. Cada nueva aplicación de codificación tiene que añadir una nueva cadena con la información apropiada.</p> <p>En el Apéndice 2 del Anexo 1 figura un formato normalizado para la información del historial de codificación.</p> <p>Esta información debe contener el tipo de sonido (MIC o MPEG) con sus parámetros específicos:</p> <p>MIC: modo (monofónico, estereofónico), tamaño de la muestra (8, 16 bits) y frecuencia de muestreo,</p>

MPEG: frecuencia de muestreo, velocidad binaria, Capa (I o II) y el modo (monofónico, estereofónico, estereofónico asociado o canal dual).

NOTA 1 – Todos los ítems, excepto uDescription, uOriginator, uOriginatorReference y uCodingHistory, tendrán el mismo contenido que cada ítem correspondiente del segmento bext (véase el § 2.3).

NOTA 2 – Cuando el valor de un código dado en UTF-8 no pertenece al subconjunto (tal como se define en el Capítulo 12 de la Norma ISO/CEI 10646:2003) admitido por un componente del equipo de procesamiento, el valor no se modificará y será ignorado en el procesamiento.

Apéndice 1 al Anexo 1 (con carácter informativo)

Formato de fichero RIFF WAVE (.WAV)

La información de este Apéndice se ha tomado de los documentos sobre el formato de fichero RIFF de Microsoft®. Se incluye para información solamente.

1 Formato de fichero audio de forma de onda (WAVE)

La forma WAVE se define como sigue. Los programas deben esperar (y pasar por alto) cualesquiera segmentos desconocidos encontrados, con todas las formas RIFF. Sin embargo, <fmt-ck> debe aparecer siempre antes que <wave-data> (datos de onda), y estos dos segmentos son obligatorios en un fichero WAVE.

```
<WAVE-form> ->
    RIFF ( 'WAVE'
        <fmt-ck>           // Segmento de formato
        [<fact-ck>]       // Segmento ampliado
        [<other-ck>]      // Otros segmentos facultativos
        <wave-data> )     // Datos radiofónicos
```

Los segmentos WAVE se describen en las secciones siguientes:

1.1 Segmento de formato WAVE

El segmento de formato WAVE <fmt-ck> especifica el formato de <wave data>. <fmt-ck> se define como sigue:

```
<fmt-ck> ->fmt( <common-fields>
    <format-specific-fields> )
<common-fields> ->
    struct{
        WORD wFormatTag;           /* Categoría de formato */
        WORD nChannels;           /* Número de canales */
        DWORD nSamplesPerSec;     /* Velocidad de muestreo */
        DWORD nAvgBytesPerSec;    /* Para estimación de la memoria intermedia */
        WORD nBlockAlign;         /* Tamaño de bloque de datos */
    }
```

Los campos en la porción <common fields> (campos comunes) del segmento son los siguientes:

Campo	Descripción
wFormatTag	Un número que indica la categoría del formato WAVE del fichero. El contenido de la porción <format-specific-fields> (campos específicos de formato) del segmento 'fmt' y la interpretación de los datos de la forma de onda, dependen de este valor.
nchannels	El número de canales representado en los datos de forma de onda, a saber, 1 para monofonía o 2 para estereofonía.
nSamplesPerSec	La velocidad de muestreo (en muestras por segundo) a la cual se debe reproducir cada canal.
nAvgBytesPerSec	El número medio de octetos por segundo en que se deben transferir los datos de forma de onda. El soporte lógico de reproducción puede estimar el tamaño de la memoria intermedia utilizando este valor.
nBlockAlign	La alineación de bloques (en octetos) de los datos de forma de onda. El soporte lógico de reproducción necesita procesar a la vez un múltiplo de octetos de datos <nBlockAlign> (número de alineación de bloque), con el fin de que el valor de <nBlockAlign> se pueda utilizar para la alineación de la memoria intermedia.

El campo (<format-specific-fields>) consta de cero, uno o más octetos de parámetros. Los parámetros que aparecen dependen de la categoría del formato WAVE: para más detalles, véanse las secciones siguientes. El soporte lógico de reproducción debe permitir (y pasar por alto) cualesquiera parámetros <format-specific-fields> desconocidos que aparezcan al final de este campo.

1.2 Categorías de formato WAVE

La categoría de formato de un fichero WAVE es especificada por el valor del campo <wFormatTag> (rótulo de formato) del segmento 'fmt'. La representación de datos en <wave data>, y el contenido de <format-specific-fields> del segmento 'fmt', dependen de la categoría del formato.

Entre las categorías de formato WAVE abiertas que no son privadas actualmente definidas cabe citar las siguientes:

wFormatTag	Valor	Categoría de formato
WAVE_FORMAT_PCM	(0x0001)	Formato MIC Microsoft®
WAVE_FORMAT_MPEG	(0x0050)	Audio MPEG-1 (audio solamente)

NOTA – Aunque otros formatos WAVE están registrados en Microsoft®, sólo los formatos anteriores se utilizan actualmente con BWF. En el § 2 se dan detalles sobre el formato PCM WAVE. En el § 3 figura información general sobre otros formatos WAVE. Los detalles del formato MPEG WAVE se indican en el Anexo 2. En el futuro se pueden definir otros formatos WAVE.

2 Formato MIC

Si el campo <wFormatTag> del <fmt-ck> se pone a WAVE_FORMAT_PCM, los datos de forma de onda consisten en muestras representadas en formato MIC. Para los datos de forma de onda MIC, <format-specific-fields> se define como sigue:

```
<PCM-format-specific> ->
    struct{
        WORD nBitsPerSample;    /* Tamaño de muestra */
    }
```

El campo <nBitsPerSample> (número de bits por muestra) especifica el número de bits de datos utilizado para representar cada muestra de cada canal. Si hay múltiples canales, el tamaño de la muestra es igual para cada canal.

El campo <nBlockAlign> debe ser igual a la siguiente fórmula, redondeada hasta el siguiente número entero:

$$nchannels \times BytesPerSample$$

El valor de BytesPerSample debe calcularse redondeando nBitsPerSample hasta el siguiente número entero. Cuando la palabra de muestra audio es menor que un número entero de bytes, los bits más significativos de la muestra audio se colocan en los bits más significativos de la palabra de datos, y los bits de datos no utilizados adyacentes al bit menos significativo se pondrán a cero.

Para datos MIC, el campo <nAvgBytesPerSec> (número medio de bytes por segundo) del segmento 'fmt' debe ser igual a la siguiente fórmula:

$$nSamplesPerSec \times nBblockAlign$$

NOTA – La especificación WAVE original permite, por ejemplo, el empaquetamiento en 5 bytes de muestras de 20 bits tomadas de dos canales, compartiéndose un solo byte para los bits menos significativos de ambos canales. Esta Recomendación especifica un número completo de bytes por muestra audio con el fin de reducir ambigüedades en las implementaciones y alcanzar la máxima compatibilidad en los intercambios.

2.1 Empaquetado de datos para ficheros PCM WAVE

En un fichero WAVE monocal, las muestras se almacenan consecutivamente. Para ficheros WAVE estereofónicos, el canal 0 representa el canal izquierdo y el canal 1 representa el canal derecho. En ficheros WAVE multicanal, las muestras están entrelazadas.

En los siguientes diagramas se observa el empaquetado de datos para ficheros WAVE monofónicos y estereofónicos de 8 bits:

Empaquetado de datos para MIC monofónica de 8 bits

Muestra 1	Muestra 2	Muestra 3	Muestra 4
Canal 0	Canal 0	Canal 0	Canal 0

Empaquetado de datos para MIC estereofónica de 8 bits

Muestra 1		Muestra 2	
Canal 0 (izquierda)	Canal 1 (derecha)	Canal 0 (izquierda)	Canal 1 (derecha)

En los siguientes diagramas se observa el empaquetado de datos para ficheros WAVE monofónicos y estereofónicos de 16 bits:

Empaquetado de datos para MIC monofónica de 16 bits

Muestra 1		Muestra 2	
Canal 0 octeto de orden inferior	Canal 0 octeto de orden superior	Canal 0 octeto de orden inferior	Canal 0 octeto de orden superior

Empaquetado de datos para MIC estereofónica de 16 bits

Muestra 1			
Canal 0 (izquierda)	Canal 0 (izquierda)	Canal 1 (derecha)	Canal 1 (derecha)
octeto de orden inferior	octeto de orden superior	octeto de orden inferior	octeto de orden superior

2.2 Formato de datos de las muestras

Cada muestra está contenida en un entero i . El tamaño de i es el número más pequeño de octetos requeridos para contener el tamaño de muestra especificado. El octeto menos significativo se almacena primero. Los bits que representan la amplitud de la muestra se almacenan en los bits más significativos de i , y los bits restantes se ponen a cero.

Por ejemplo, si el tamaño de la muestra (grabada en $\langle n\text{BitsPerSample} \rangle$ (número de bits por muestra)) es 12 bits, cada muestra se almacena en un entero de 2 octetos. Los cuatro bits menos significativos del primer octeto (menos significativo) se ponen a cero. El formato de datos y los valores máximos y mínimos de las muestras de forma de onda MIC de varios tamaños son como sigue:

Tamaño de muestra	Formato de datos	Valor máximo	Valor mínimo
Uno a ocho bits	Entero sin signo	255 (0xFF)	0
Nueve o más bits	Entero con signo i	Valor positivo máximo de i	Valor más negativo de i

Por ejemplo, los valores máximo, mínimo y medio para datos de forma de onda MIC de 8 bits y de 16 bits son como sigue:

Formato	Valor máximo	Valor mínimo	Valor medio
8-bit MIC	255 (0xFF)	0	128 (0x80)
16-bit MIC	32767(0x7FFF)	-32768(-0x8000)	0

2.3 Muestra de ficheros PCM WAVE

Ejemplo de un fichero PCM WAVE con velocidad de muestreo de 11,025 kHz, monofónico, 8 bits por muestra:

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

Ejemplo de un fichero PCM WAVE con velocidad de muestreo de 22,05 kHz, estereofónico, 8 bits por muestra:

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

Ejemplo de un fichero PCM WAVE con velocidad de muestreo de 44,1 kHz, monofónico, 20 bits por muestra:

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

2.4 Almacenamiento de datos WAVE

<wave data> contiene los datos de forma de onda y se define como sigue:

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

2.5 Segmento ampliado

El segmento ampliado <fact-ck> almacena información importante sobre el contenido del fichero WAVE. Este segmento se define como sigue:

```
<fact-ck> -> fact( <dwFileSize:DWORD> ) /* Número de muestras */
```

Este segmento no se requiere para ficheros MIC.

El segmento ampliado se extenderá para incluir cualquier otra información requerida por futuros formatos WAVE. Los campos añadidos aparecerán después del campo <dwFileSize> (tamaño de fichero). Las aplicaciones pueden utilizar el campo de tamaño del segmento para determinar los campos que están presentes.

2.6 Otros segmentos facultativos

Se especifican algunos otros segmentos para utilización en el formato WAVE. Los detalles de estos segmentos figuran en la especificación del formato WAVE y cualesquiera actualizaciones publicadas ulteriormente.

NOTA – El formato WAVE puede admitir otros segmentos facultativos que pueden ser incluidos en ficheros WAVE para transportar información específica. Como se indica en la Nota 1 del § 2.1 del Anexo 1, en el fichero de formato de onda de radiodifusión se considera que éstos son segmentos privados y serán omitidos por aplicaciones que no pueden interpretarlos.

3 Otros tipos de WAVE

La siguiente información se ha extraído de Microsoft® Data Standards. Destaca las extensiones necesarias de los ficheros WAVE básicos (utilizados para audio MIC) con el fin de tratar otros tipos de formato WAVE.

3.1 Información general

Todos los tipos WAVE recientemente definidos deben contener un <fact-ck> (segmento ampliado) y una descripción de formato de onda ampliada dentro del segmento de formato <fmt-ck>: Los ficheros RIFF WAVE del tipo WAVE_FORMAT_PCM no tienen que tener el segmento suplementario ni la descripción de formato de onda ampliada.

3.2 Segmento ampliado

Este segmento almacena información dependiente del fichero sobre el contenido del fichero WAVE. Especifica actualmente la longitud del fichero en muestras.

Extensión del formato WAVE

La estructura de formato de onda ampliada añadida a <fmt-ck> se utiliza para definir todos los datos de onda de formato que no son MIC, y se describe como sigue. La estructura de formato de forma de onda ampliada general se utiliza para todos los formatos que no son MIC.

```
typedef struct waveformat_extended_tag {
    WORD    wFormatTag;          /* tipo de formato */
    WORD    nChannels;          /* número de canales (monofónicos, estereofónicos) */
    DWORD   nSamplesPerSec;     /* velocidad de muestreo */
    DWORD   nAvgBytesPerSec;    /* para estimación de la memoria intermedia */
    WORD    nBlockAlign;        /* tamaño de bloque de datos */
    WORD    wBitsPerSample;     /* número de bits por muestra de datos monofónicos */
    WORD    cbSize;             /* cómputo en octetos del tamaño suplementario */
} WAVEFORMATEX;
```

Campo	Descripción
wFormatTag	Define el tipo de fichero WAVE.
nChannels	Número de canales en la onda, 1 para monofonía, 2 para estereofonía.
nSamplesPerSec	Frecuencia de la velocidad de muestreo del fichero. Debe ser 48000 ó 44100, etc. Esta velocidad es utilizada también por la entrada de tamaño de muestra en el segmento ampliado para determinar la duración de los datos.
nAvgBytesPerSec	Velocidad media de datos. El soporte lógico de reproducción puede estimar el tamaño de la memoria intermedia utilizando el valor <nAvgBytesPerSec> (número medio de octetos por segundo).
nBlockAlign	La alineación de bloques (en octetos) de los datos de <data-ck> (segmento de datos). El soporte lógico de reproducción tiene que procesar a la vez un múltiplo de octetos de datos (<nBlockAlign>), de modo que el valor de <nBlockAlign> se pueda utilizar para la alineación de la memoria intermedia.
wBitsPerSample	Éste es el número de bits por muestra por canal. Se supone que cada canal tenga la misma resolución de muestreo. Si este campo no es necesario, se debe poner a cero.
cbSize	El tamaño en octetos de la información suplementaria en el encabezamiento del formato WAVE no incluye el tamaño de la estructura WAVEFORMATEX.

NOTA – Los campos que siguen al campo <cbSize> (tamaño) contienen información específica necesaria para el formato WAVE definido en el campo <wFormatTag> (rótulo de formato). Cualesquiera formatos WAVE que puedan ser utilizados en el BWF se especificarán en Suplementos a la presente Recomendación.

Apéndice 2 al Anexo 1 (con carácter informativo)

Especificación del formato para el campo <CodingHistory>

Introducción

El campo <CodingHistory> del segmento <bext> se define como una colección de cadenas que contienen un historial de los procesos de codificación. Debe añadirse una nueva fila siempre que varíe el historial de codificación. Cada fila debe contener una cadena variable para cada parámetro de la codificación. Cada fila debe terminar por CR/LF. A continuación se da un formato para las cadenas del historial de codificación.

Sintaxis

La sintaxis de cada fila debe ser la siguiente:

Parámetro	Cadena variable <allowed option>
Algoritmo de codificación	A=<ANALÓGICO, PCM, MPEG1L1, MPEG1L2, MPEG1L3, MPEG2L1, MPEG2L2, MPEG2L3>
Frecuencia de muestreo (Hz)	F=<16000,22050,24000,32000,44100,48000>
velocidad binaria (kbit/s por canal)	B=<cualquier velocidad binaria permitida MPEG 2 (ISO/CEI 13818-3)>
Longitud de palabra (bits)	W=<8, 12, 14, 16, 18, 20, 22, 24>
Modo	M=<monoaural, estéreo, mono-dual, estéreo-mixto>
Texto, cadena libre	T=<cadena libre de texto ASCII- para utilización interna. Esta cadena no debe contener comas (ASCII 2Ch _{hex}). Ejemplos de contenido: No-ID; tipo o de códec; tipo A/D>

Las cadenas variables deben ir separadas por comas (ASCII 2Ch_{hex}). Cada fila debe terminar por CR/LF.

Variable B = Se utiliza únicamente para codificación MPEG.

Variable W = Para codificación MPEG, debe utilizarse a fin de indicar la longitud de la palabra de la entrada MIC al codificador MPEG.

Ejemplos de campos de historial de codificación

Ejemplo 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

Interpretación del ejemplo 1

Línea 1:

El fichero original se registra como fichero BWF lineal con codificación MIC y:

- Frecuencia de muestreo: 48 kHz
- Resolución de codificación: 16 bits por muestra
- Modo: estéreo
- Estado: codificación original

Línea 2:

El fichero original se ha convertido en un fichero BWF de Capa 2 MPEG-1 utilizando los parámetros:

- Frecuencia de muestreo: 48 kHz
- bits por segundo por canal: 192 kbit/s
- Resolución de codificación: 16 bits
- Modo: estéreo
- Codificador: PCX9 (digigram)

Ejemplo 2 para un proceso de digitalización de material analógico

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>

Interpretación del ejemplo 2**Línea 1:**

Se reprodujo la cinta magnética analógica, tipo Agfa PER528, en un grabador modelo Studer A816, número de serie 1007:

- Velocidad de cinta: 38 cm/s
- Modo: estéreo

Línea 2:

La grabación era una digitalización utilizando un convertidor A/D tipo NVision NV1000 con:

- Frecuencia de muestreo: 48 kHz
- Resolución de codificación: 18 bits por muestra
- Modo: estéreo

Línea 3:

La grabación se almacenó como fichero BWF con codificación MIC lineal, utilizando la entrada digital de una tarjeta de interfaz PCX9 con:

- Frecuencia de muestreo: 48 kHz
- Resolución de codificación: 16 bits por muestra
- Modo: estéreo

Apéndice 3 al Anexo 1 (con carácter informativo)

Definición del formato de «Unique» Source Identifier (USID) para utilizar en el campo <OriginatorReference>

USID

El USID en <OriginatorReference> se genera mediante varias fuentes independientes de aleatorización a fin de garantizar su unicidad en ausencia de una única autoridad de asignación. Una forma eficaz y fácil de utilizar el método de aleatorización consiste en combinar la información específica de usuario, de máquina y de tiempo con un número aleatorio. Los elementos son:

CC Código de país: (2 caracteres) basado en la norma ISO 3166⁵ [ISO, 1997].

OOOO Código de organización: 4 caracteres

NNNNNNNNNNNNNN Número de serie: (12 caracteres extraídos del número de serie y número de modelo del grabador). Sirve para identificar el tipo y número de serie de la máquina

HHMMSS OriginationTime: (6 caracteres) del campo <OriginationTime> del BWF.

Estos elementos bastarán para identificar una grabación determinada en forma útil para las personas, junto con otras fuentes de información, formales e informales.

Además, el USID contiene:

RRRRRRRRR Número aleatorio: (8 caracteres) generado localmente por el grabador utilizando algún algoritmo aleatorio razonable.

Este elemento sirve para identificar por separado los ficheros, como los canales estéreo o las pistas en las grabaciones multipista que sean hechos al mismo tiempo.

Ejemplos de USID

Ejemplo 1

USID generado por un aparato Tascam DA88, S/N 396FG347A, utilizado por la Radiotelevisión Italiana (RAI) a las 12:53:24 horas

Formato UDI: CCOOOONNNNNNNNNNNNNHHMMSSRRRRRRRRR

Ejemplo UDI: ITRAI0DA88396FG34712532498748726

Ejemplo 2

USID generado por un xxxxxxxx, S/N ssssssss, utilizado por la Radiodifusora finlandesa YLE, a las 08:14:48 horas

Formato UDI: CCOOOONNNNNNNNNNNNNHHMMSSRRRRRRRRR

Ejemplo UDI: FIYLE0xxxxxxxssssss08144887724864

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

Apéndice 4 al Anexo 1 (con carácter informativo)

Definición de un segmento facultativo de nivel de cresta de la envolvente <levl -ck> para el BWF

Cuando se intercambian ficheros audio entre estaciones de trabajo, se puede acelerar la apertura, visualización y procesamiento de un fichero si éste dispone de datos sobre los niveles de cresta de la señal audio. Al añadir un segmento <levl> a un fichero de formato de onda de radiodifusión (BWF) se consigna una norma de almacenamiento y transferencia de datos sobre las crestas de la señal obtenidas mediante el submuestreo audio. Estos datos en el segmento pueden utilizarse para prever la envolvente de las señales esenciales de audio en el fichero, gracias a lo cual una aplicación de audio podrá visualizar rápidamente los ficheros audio sin perder demasiada precisión.

Por otra parte, se puede enviar la cresta de crestas, que es la *primera* muestra audio cuyo valor absoluto es el valor máximo de todo el fichero audio. Una aplicación audio puede utilizar esta información para normalizar un fichero en tiempo real sin tener que explorarlo completamente (puesto que ya lo ha hecho el emisor).

1 Terminología

La señal audio se divide en bloques, y para cada bloque se genera una **trama de cresta**. Para cada trama de cresta hay **n valores de cresta**, donde n es el número de canales de cresta. Cada valor de cresta puede consistir en un **punto de cresta** (sólo positivo) o en dos **puntos de cresta** (uno positivo y otro negativo).

1.1 Obtención de valores de cresta

La señal audio se divide en bloques de muestras de tamaño constante. El tamaño de los bloques por defecto, y el más recomendado, es de 256 muestras de cada canal.

Para hallar los puntos de cresta (valores máximos) se evalúan las muestras de cada canal. Se recomienda obtener puntos de cresta separados para muestras positivas y negativas pero sólo podrá utilizarse alternativamente el valor absoluto (positivo o negativo). Todos los puntos de cresta son valores sin signo.

Los puntos de cresta se redondean a uno de los dos formatos, de 8 ó 16 bits. En la mayoría de los casos, el formato de 8 bits es suficiente. El formato de 16 bits se aplicará a todos los casos en que se necesite una mayor precisión.

Los puntos de cresta formateados para cada canal se reúnen en tramas de cresta. Cada trama de cresta contiene los puntos de cresta positivos y negativos (o el punto de cresta absoluto) para cada canal en el mismo orden de las muestras audio.

Esas tramas de cresta son transmitidas como los datos en el segmento de cresta de la envolvente, el cual se inicia con un encabezamiento donde figura información que permite la interpretación de los datos de cresta.

La **cresta de crestas** es la *primera* muestra audio cuyo valor absoluto es el valor máximo de todo el fichero audio. En vez de almacenar la cresta de crestas como un valor de cresta, se almacena la *posición* de la cresta de crestas. En otras palabras, se almacena el índice de una trama de la muestra audio. Una aplicación sabe luego *dónde* leer la cresta de crestas en el fichero audio. Resultaría más difícil almacenar un valor de cresta ya que éste depende del formato binario de las muestras audio (enteros, flotantes, dobles ...).

NOTAS:

- El encabezamiento sólo utiliza DWORD (valores de 4 bytes) o múltiplos de 4 bytes para evitar problemas en la alineación de estructuras en diferentes compiladores.
- El tamaño total del encabezamiento es de 128 bytes para evitar un alineamiento incorrecto de la memoria *cache*.

2 Segmento de cresta de la envolvente

El segmento de cresta de la envolvente, *<levl>*, consiste en un encabezamiento seguido por los datos de los puntos de cresta. La longitud total del segmento variará según el contenido audio, el tamaño de los bloques y la forma en que están formateados los datos de cresta.

```
typedef struct peak_envelope
{
    CHAR          ckID[4],          /* {'l','e','v','l'} */
    DWORD         ckSize,          /* tamaño del segmento */
    DWORD         dwVersion,       /* información de la versión */
    DWORD         dwFormat,;       /* formato de un punto de cresta */
                                   1 = carácter sin signo
                                   2 = entero corto sin signo
    DWORD         dwPointsPerValue, /* 1 = sólo punto de cresta positivo
                                   2 = puntos de cresta positivos Y negativos */
    DWORD         dwBlockSize,     /* tramas por valor */
    DWORD         dwPeakChannels,  /* número de canales */
    DWORD         dwNumPeakFrames, /* número de tramas de cresta */
    DWORD         dwPosPeakOfPeaks, /* índice de trama de muestra audio/* o
                                   0xFFFFFFFF, si se conoce */
    DWORD         dwOffsetToPeaks, /* será generalmente igual al tamaño de este
                                   encabezamiento, pero también podría ser mayor */
    CHAR          strTimestamp[28], /* ASCII: sello temporal de los datos de cresta */
};
```

2.1 Elementos del segmento «levl»

ckID	Conjunto de 4 caracteres {«l», «e», «v», «l»} ⁶ que identifica el segmento.
ckSize	Tamaño de la parte restante del segmento. (No incluye los 8 bytes utilizados por ckID y ckSize.)
dwVersion	Versión del segmento peak_envelope. Comienza con 0000.

⁶ La definición de DWORD ckID = «levl» no será la única. El orden de los caracteres será diferente según los diferentes compiladores-C . Por lo tanto, damos aquí la siguiente definición del carácter ckID[4] = {«l», «e», «v», «l»}

dwFormat Formato de los datos de cresta de la envolvente. Se admiten dos formatos⁷:

dwFormat	Valor	Descripción
LEVL_FORMAT_UINT8	1	carácter sin signo para cada punto de cresta
LEVL_FORMAT_UINT16	2	entero corto sin signo para cada punto de cresta

dwPointsPerValue Indica el número de puntos de cresta por valor de cresta que puede ser uno o dos.

dwPointsPerValue = 1

Cada valor de cresta consiste en un punto de cresta, que es el máximo de los valores absolutos de las muestras audio **dwBlockSize** en cada bloque:

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

NOTA – En este caso, la forma de onda visualizada será siempre simétrica con respecto al eje horizontal.

dwPointsPerValue = 2

Cada valor de cresta consiste en dos puntos de cresta. El primero corresponde al valor *positivo* más alto de las muestras audio **dwBlockSize** en el bloque. El segundo corresponde a la cresta *negativa* de las muestras audio **dwBlockSize** en el bloque.

Se recomienda utilizar dos puntos de cresta (**dwPointsPerValue = 2**) dado que las formas de ondas asimétricas (por ejemplo, una separación DC) se recibirán correctamente.

dwBlockSize Número de muestras audio utilizadas para obtener cada trama de cresta. Este número es variable. El tamaño de los bloques por defecto, y el más recomendado, es de 256 muestras.

dwPeakChannels Número de canales de cresta⁸.

dwNumPeakFrames Número de tramas de cresta. El número de tramas de cresta es el entero que se obtiene redondeando por defecto el cálculo siguiente:

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

o redondeando por exceso el cálculo siguiente:

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

donde numAudioFrame es el número de muestras audio en cada canal de los datos de audio.

⁷ Dado que las aplicaciones audio que admiten el segmento «levl» tendrán que implementar todos los formatos posibles, sólo se admiten dos formatos.

En la mayoría de los casos, el formato del carácter sin signo (8 bits) es suficiente. El formato del entero breve sin signo (16 bits) se aplicará a todos los casos en que se necesite una mayor precisión.

⁸ Generalmente, el número de canales de cresta es igual al número de canales audio. Si ese número es uno, se obtendrá la misma forma de onda para cada canal audio.

Por ejemplo, para una relación de cresta (tamaño del bloque) de 256, esto significa lo siguiente:

0	muestras audio	-> 0 trama de cresta
1	muestra audio	-> 1 trama de cresta
256	muestras audio	-> 1 trama de cresta
257	muestras audio	-> 2 tramas de cresta
7582	muestras audio	-> 30 tramas de cresta.

dwPosPeakOfPeaks

Una aplicación audio puede utilizar esta información para normalizar un fichero sin tener que explorarlo completamente (puesto que ya lo ha hecho el emisor). La ventaja es que aumenta la calidad de funcionamiento y también la posibilidad de normalizar un fichero en tiempo real.

La **cresta de crestas** es la *primera* muestra audio cuyo valor absoluto es el valor máximo de todo el fichero audio.

En vez de almacenar la cresta de crestas como un valor de cresta, se almacena la *posición* de la cresta de crestas. En otras palabras, se almacena el índice de una trama de la muestra audio. Una aplicación sabe luego *dónde* leer la cresta de crestas en el fichero audio. Resultaría más difícil almacenar un valor de cresta ya que éste depende del formato binario de las muestras audio (enteros, flotantes, dobles ...).

Si el valor es 0xFFFFFFFF, esto significa que se desconoce la cresta de crestas.

dwOffsetToPeaks

Desviación de los datos de cresta a partir del comienzo del encabezamiento. Generalmente, esto es igual al tamaño del encabezamiento, pero podría ser mayor. Puede utilizarse para garantizar que los datos de cresta comienzan en una frontera DWORD.

strTimeStamp

Cadena que contiene el sello temporal de la creación de los datos de cresta. Su formato es el siguiente⁹:

«YYYY:MM:DD:hh:mm:ss:uuu»

donde:

YYYY: año
 MM: mes
 DD: día
 hh: horas
 mm: minutos
 ss: segundos
 uuu: milisegundos

Ejemplo: «2000:08:24:13:55:40:967»

⁹ La ventaja de este formato es que no tiene límites de tiempo y es de fácil lectura. (Otros formatos utilizan un valor DWORD que cuenta los segundos transcurridos desde 1970 y que alcanza su límite al cabo de unos 125 años.)

2.2 Formato de un punto de cresta

Un valor de cresta está constituido por uno o dos puntos de cresta, señalados como **dwPointsPerValue**. La bandera **dwFormat** indica el formato de los números que representan los puntos de cresta en cada trama de cresta.

		dwPointsPerValue	
		= 1	= 2
dwFormat		El número corresponde a la cresta absoluta	El primer número corresponde a la cresta positiva El segundo número corresponde a la cresta negativa (Obsérvese que la cresta «negativa» se almacena como número « positivo»)
= 1	levl_format_uint8	<i>carácter sin signo (0...255)</i>	<i>carácter sin signo (0...255)</i> <i>carácter sin signo (0...255)</i>
= 2	levl_format_uint16	<i>corto sin signo (0...65535)</i>	<i>corto sin signo (0...65535)</i> <i>corto sin signo (0...65535)</i>

2.3 Ficheros de cresta multicanal

En los ficheros audio multicanal se intercalan los valores de cresta de cada canal. Se llama trama de cresta al conjunto de valores de cresta intercalados. El orden de los valores de cresta en el interior de una trama de cresta corresponde a la ubicación de los puntos de la muestra dentro de la trama de datos de audio RIFF.

2.4 Sincronización con el fichero audio

Se deberá efectuar la reconstrucción del fichero de cresta si se cumple una de las dos condiciones siguientes:

El sello temporal es más antiguo que el sello temporal del fichero audio.

El número de tramas de cresta no corresponde al número de tramas de la muestra en el fichero audio.

2.5 Orden de los bytes

Dado que el fichero de formato de onda de radiodifusión (BWF) es una extensión del fichero RIFF, todos los números se almacenan en el formato correspondiente al sistema *little-endian*.

Apéndice 5 al Anexo 1 (con carácter informativo)

Definición de un segmento facultativo de enlace <link-ck> para el BWF

Introducción

Aunque el fichero de formato de onda de radiodifusión (BWF) tiene en cuenta un tamaño de fichero máximo de 4 Gbytes, en la práctica muchas aplicaciones RIFF/de onda admitirán únicamente un tamaño de fichero máximo de 2 Gigabytes. Para los datos de audio que exceden estos límites, es necesario dividir la información audio en más de un fichero BWF. El segmento <link> proporciona datos vinculantes para una salida audio sin discontinuidades a través de varios ficheros.

1 Terminología

Conjunto de ficheros (File-set)	Conjunto de ficheros vinculantes pertenecientes a una señal audio continua.
Nombre de fichero (Filename)	Nombre dado a cada fichero en el conjunto de ficheros.
Lista de ficheros (File list)	Lista de los nombres de fichero en el conjunto de ficheros.
Atributo «real» ("Actual" attribute)	Atributo que marca el nombre de fichero en la lista de ficheros indicando que es el fichero actual (o «real»). Los demás nombres de ficheros en la lista quedan marcados como «otros».
Identificador de fichero (File identifier)	Identificador optativo que debe ser el mismo para todos los ficheros de un conjunto de ficheros.
Elemento «privado» ('Private' element)	Elemento adicional en el segmento que almacena la información no pública (o de dominio privado) en la lista de ficheros.
Segmento <link> (<link> chunk)	Segmento contenido en todos los ficheros de un conjunto de ficheros. Consta de un encabezamiento seguido de una lista de ficheros, y también puede contener un identificador de fichero y un elemento «privado». Los datos en el segmento se almacenan en el formato XML 1.0 ¹⁰ , formato de amplia utilización para el intercambio de datos.

2 Estructura del segmento de enlace

2.1 Panorama general

El segmento <link> consiste en un encabezamiento seguido de la información vinculante almacenada en formato XML (lenguaje de etiquetado extensible). La longitud total del segmento será variable.

¹⁰ Lenguaje de etiquetado extensible (XML, *Extensible Markup Language*) 1.0, Recomendación W3C, 10 de febrero de 1998 (<http://www.w3.org/TR/1998/REC-xml-19980210>).

```
typedef struct link
{CHAR   CkID[4],      /* {'l','i','n','k'} */
  DWORD CkSize,      /* size of chunk */
  CHAR  XmlData[ ],  /* link-up information in XML */
}
Link_chunk,
```

Field	Description	Es el conjunto de 4 caracteres {'l', 'i', 'n', 'k'} ¹¹ para la identificación del segmento.
CkSize	Tamaño de la sección de datos del segmento (no incluye los 8 bytes utilizados por ckID y ckSize).	
XmlData	Esta memoria tampón contiene la información vinculante en XML (caracteres ASCII).	

2.2 Estructura de datos XML en el campo datos variables <xmlData>

La estructura de datos es jerárquica. Los datos se almacenan en cadenas de texto. Para la especificación exacta de la sintaxis, se añade un DTD (documento de transferencia de datos).

```
<LINK>
    <FILE type="...">
        <FILENUMBER>...</FILENUMBER>
        <FILENAME>...</FILENAME>
    </FILE>
    .....
    Otros posibles elementos FILE
    .....
    <ID>...</ID>      facultativo
    <PRIVATE>         facultativo
    ..... depende de la implementación
    </PRIVATE>
</LINK>
```

LINK	Elemento raíz de los datos XML. LINK (enlace) contiene uno o más elementos FILE (fichero) con la descripción del fichero. También puede contener el elemento ID (identificador) y/o un elemento PRIVATE (privado).
ID	El identificador ID es común a todos los ficheros de un conjunto de ficheros dado. Se almacena como cadena de caracteres texto permitida por la definición #PCDATA de la especificación XML 1.0, que incluye todos los caracteres ASCII visibles, espacios, etc.
PRIVATE	El elemento PRIVATE puede contener información que depende de la implementación y que puede estar formada por cualquier tipo de datos XML (como otros elementos o #PCDATA).
FILE	El elemento FILE contiene el elemento FILENUMBER (número de fichero) y el elemento FILENAME (nombre de fichero). El atributo tipo debe ser «real» cuando el fichero de la lista describa el fichero al que pertenece el segmento. Todos los demás ficheros tendrán el atributo tipo «otro». El nombre del fichero será el mismo que aparece en la lista de ficheros.

¹¹ La definición DWORD ckID = «enlace» no será la única. El orden de los caracteres será diferente según los diferentes compiladores-C. Por lo tanto, damos aquí la siguiente definición del carácter ckID[4] = {'l', 'i', 'n', 'k'}.

FILENUMBER Los ficheros se enumerarán en forma secuencial según su orden cronológico en el conjunto de ficheros. Se utilizarán números enteros (caracteres ASCII) que comienzan con el número 1.

FILENAME Cadena de texto almacenada en el mismo formato que el ID.

2.3 DTD para la estructura XML del segmento <link>

En la especificación XML 1.0 se describe DTD (definición de tipo de documento) como una definición de la sintaxis de una estructura XML. Se describen a continuación el formato y los atributos de los diferentes elementos del segmento <link>, incluidos los subelementos y su multiplicidad.

El elemento LINK debe contener uno o más subelementos FILE ('+' indica uno o más), puede contener un subelemento ID y un subelemento PRIVATE ('?' indica uno o ninguno).

Cada elemento FILE debe contener un subelemento FILENUMBER y un subelemento FILENAME. Se debe especificar un atributo tipo, que puede ser «real» u «otro».

Los subelementos FILENUMBER, FILENAME e ID deben contener cadenas de textos (llamada #PCDATA en XML).

El subelemento PRIVATE puede contener cualquiera de los elementos definidos. Si el subelemento PRIVATE necesita contener otros elementos distintos de los definidos, el DTD debe ser modificado en consecuencia.

<!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 Nueva denominación de ficheros vinculantes

Si se cambian uno o más nombres de fichero, se cambiarán las entradas FILENAME correspondientes en cada uno de los segmentos <link> pertenecientes al conjunto completo de ficheros.

En este ejemplo, la señal sonora continua se ha dividido en un conjunto de ficheros de tres ficheros BWF llamados «Sinatra_1.wav», «Sinatra_2.wav» y «Sinatra_3.wav». Las estructuras XML de los segmentos <link> de los tres ficheros son idénticas excepto con respecto al atributo tipo.

3.1 Segmento <link> de «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 Segmento <link> de «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 Segmento <link> de «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>

```

Apéndice 6 al Anexo 1 (con carácter normativo)

Convenciones de los nombres de fichero

1 Aspectos generales

Por intercambio general de ficheros audio se entiende la reproducción de ese tipo de ficheros en computadores y tipos de sistemas operativos que pueden ser muy diferentes al sistema original. Un nombre de fichero inadecuado podría indicar que el sistema destinatario no puede reconocer el fichero. Por ejemplo, algunos sistemas operativos informáticos imponen límites al número de caracteres en un nombre de fichero. Otros no pueden admitir caracteres de múltiples bytes. En ciertos sistemas operativos algunos caracteres tienen un significado especial y deben evitarse. La finalidad de esta guía de carácter orientativo es determinar las mejores prácticas en relación con el intercambio internacional general.

2 Longitud del nombre de fichero

Los nombres de fichero BWF no deben exceder los 31 caracteres, incluida la extensión nombre de fichero.

3 Extensión nombre de fichero

Los ficheros BWF utilizarán la misma extensión nombre de fichero de cuatro caracteres, «.wav», como fichero WAVE convencional. De este modo, se podrá reproducir el contenido audio en la mayoría de los computadores sin necesidad de un programa adicional. Las implementaciones prácticas deben también aceptar otras extensiones como, por ejemplo, «.bwf», que pueden haber sido utilizadas por error.

4 Conjunto de caracteres nombre de fichero

Los nombres de fichero para el intercambio internacional deben utilizar solamente caracteres ASCII de 7 bits (Norma ISO/CEI 646) en la gama comprendida entre 32 y 126 (valores decimales).

Carácter	Valor decimal	Valor hexadecimal
(espacio)	32	0x20
...
~ (tilde)	126	0x7E

Además, se reservan los caracteres siguientes para funciones especiales en ciertos sistemas de fichero y no deben utilizarse en nombres de fichero:

Carácter	Valor decimal	Valor hexadecimal
“	34	0x22
*	42	0x2A
/	47	0x2F
:	58	0x3A
<	60	0x3C
>	62	0x3E
?	63	0x3F
\	92	0x5C
	124	0x7C

Por otra parte, no deben utilizarse los siguientes caracteres para el primer o último carácter en un nombre de fichero:

Carácter	Valor decimal	Valor hexadecimal
(espacio)	32	0x20
(punto)	46	0x2E

Anexo 2

Especificación del formato de onda de radiodifusión con audio MPEG-1 Un formato para ficheros de datos de audio en radiodifusión

1 Introducción

Este Anexo contiene la especificación de la utilización del BWF para transportar señales audio MPEG solamente. Para audio MPEG, es necesario añadir a los segmentos básicos especificados en la parte principal de ésta, la siguiente información:

- una extensión al segmento de formato;
- un segmento ampliado;
- un segmento MPEG_extension.

La extensión al segmento de formato y al segmento ampliado se especifican como parte del formato WAVE y la información pertinente figura en el Apéndice 1 al Anexo 2.

La especificación del segmento MPEG_extension se presenta en el § 2 del Anexo 2.

La parte principal de esta Recomendación contiene la especificación del segmento de extensión de audio de radiodifusión que se utiliza en todos los BWF. La información sobre el formato RIFF básico figura en el Apéndice 1 al Anexo 2.

2 Audio MPEG

Microsoft[©] ha especificado cómo se pueden organizar los datos de audio MPEG en ficheros WAVE. Una extensión del segmento de formato y un segmento ampliado transportan información adicional necesaria para especificar las opciones de codificación MPEG. Los principios generales se indican en el Apéndice 1 al Anexo 1 y los detalles en el Apéndice 1 al Anexo 2. Para la capa II de MPEG se ha encontrado que hay que transportar información suplementaria sobre la codificación de la señal. Ésta es transportada en el segmento <extensión de audio MPEG> (<**MPEG Audio Extension**>), elaborado por el Grupo de Audio de la capa 2 de MPEG. Este segmento se especifica a continuación.

2.1 Segmento de extensión de audio MPEG

El segmento de extensión de audio MPEG se define como sigue:

```
typedef struct {
    DWORD    ckID;           /* (mpeg_extension)ckID='mext' */
    DWORD    ckSize;        /* tamaño de segmento de extensión:
                           cksize =000C*/
    BYTE    ckData[ckSize]; /* datos del segmento */
}
typedef struct mpeg_audio_extension {
    WORD    SoundInformation; /* más información sobre sonido */
    WORD    FrameSize;       /* tamaño nominal de la trama */
    WORD    AncillaryDataLength; /* Longitud de datos auxiliares */
    WORD    AncillaryDataDef; /* Tipo de datos auxiliares*/
    CHAR    Reserved [4];    /* Reservado para uso futuro; puesto a
                           "NULL"*/
} MPEG_EXT ;
```

Campo	Descripción
SoundInformation	<p>16 bits que dan información adicional sobre el fichero de sonido:</p> <p>Para la capa II (o capa I) MPEG:</p> <p style="margin-left: 40px;">Bit 0: ‘1’ Datos de sonido homogéneos ‘0’ Datos de sonido no homogéneos</p> <p>Los bits 1 y 2 se utilizan para información adicional para ficheros de sonido homogéneos:</p> <p style="margin-left: 40px;">Bit 1: ‘0’ Bit de relleno que se utiliza en el fichero para poder alternar entre ‘0’ o ‘1’ ‘1’ Bit de relleno puesto a ‘0’ en todo el fichero</p> <p style="margin-left: 40px;">Bit 2: ‘1’ El fichero contiene una secuencia de tramas con el bit de relleno puesto a ‘0’ y frecuencia de muestreo igual a 22,05 ó 44,1 kHz</p> <p>NOTA 1 – Este fichero no cumple la norma MPEG (§ 2.4.2.3, definición de bit de relleno), pero se puede considerar como un caso especial de velocidad binaria variable. No se necesita un decodificador MPEG para decodificar este tren de bits, pues la mayoría de los decodificadores realizarán esta función. La velocidad binaria será ligeramente más baja que la indicada en el encabezamiento.</p> <p style="margin-left: 40px;">Bit 3: ‘1’ Se utiliza formato libre ‘0’ Ninguna trama de audio en formato libre.</p>
FrameSize	<p>Número de octetos de 16 bits de una trama nominal.</p> <p>Este campo tiene significado solamente para ficheros homogéneos; en los demás casos se pone a ‘0’.</p> <p>Si no se utiliza el bit de relleno, es decir, que permanece constante en todas las tramas del fichero de sonido, el campo <FrameSize> (tamaño de trama) contiene el mismo valor que el campo <nBlockAlign> en el segmento de formato. Si se utiliza el bit de relleno y se producen longitudes variables en los datos de sonido, <FrameSize> contiene el tamaño de una trama con el bit de relleno puesto a ‘0’. La longitud de una trama con el bit de relleno puesto a ‘1’ es un octeto más (cuatro octetos para la capa I), es decir, <FrameSize+1>.</p> <p>El hecho de que <nBlockAlign> se pone a ‘1’ significa longitudes de trama variables (Frame Size o Frame Size + 1) con bit de relleno variable.</p>
AncillaryDataLength	<p>Número de 16 bits que indica el número mínimo de octetos conocidos para datos auxiliares en el fichero de sonido completo. El valor es relativo con respecto al fin de la trama audio.</p>
AncillaryDataDef	<p>Este valor de 16 bits especifica el contenido de los datos auxiliares con:</p> <p>Bit 0 puesto a ‘1’: Alimentación del canal izquierdo presente en datos auxiliares</p> <p style="margin-left: 40px;">Bit 1 puesto a ‘1’: Un octeto privado, libre para uso interno en datos auxiliares</p> <p style="margin-left: 40px;">Bit 2 puesto a ‘1’: Alimentación del canal derecho presente en datos auxiliares</p> <p style="margin-left: 40px;">Bit 3 puesto a ‘0’: Reservado para uso futuro para datos ADR</p> <p style="margin-left: 40px;">Bit 4 puesto a ‘0’: Reservado para uso futuro para datos DAB</p>

Bit 5 puesto a '0': Para uso futuro para datos J 52

Bit 6 a 15 puesto a '0': Para uso futuro.

NOTAS

- Los ítems presentes en datos auxiliares siguen el mismo orden que los números de bit en definición de datos auxiliares (AncillaryDataDef). El primer ítem se almacena al final de los datos auxiliares, el segundo ítem se almacena justamente detrás del primero, etc.
- Para un fichero monofónico, el bit 2 se pone siempre a '0' y el bit 0 se relaciona con la alimentación de la trama monofónica.
- Para un fichero estereofónico, si el bit 2 equivale a '0' y el bit 0 equivale a '1', la alimentación se relaciona con el máximo de alimentación a la izquierda y a la derecha.
- La alimentación se almacena en 2 octetos y corresponde al valor absoluto de la muestra máxima utilizada para codificar la trama. Éste es un valor de 15 bits en formato Big Endian.

Reserved 4 octetos reservados para uso futuro. Estos 4 octetos se deben poner a nulo. En algún uso futuro, el valor nulo se empleará para el valor por defecto con miras a mantener la compatibilidad.

Apéndice 1 al Anexo 2 (con carácter informativo)

Formato de fichero RIFF WAVE (.WAV)

Este Apéndice especifica la información suplementaria necesaria para un fichero WAVE que contiene audio MPEG.

La información de este Apéndice se ha tomado de los documentos de especificación del formato de fichero RFF de Microsoft[®]. Se incluye para información solamente.

1 **Audio MPEG-1 (audio solamente)**

1.1 **Segmento ampliado**

Este segmento se requiere para todos los formatos WAVE que no sean WAVE_FORMAT_PCM. Almacena información dependiente del fichero sobre el contenido de los datos WAVE. Especifica actualmente la longitud temporal de los datos en las muestras.

NOTA 1 – Véase también el Apéndice 1 al Anexo 1, § 2.5.

1.2 **Encabezamiento de formato 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 ;

```

Campo	Descripción
wFormatTag	Se debe poner a WAVE_FORMAT_MPEG. [0x0050]
nChannels	Número de canales, 1 para monofonía, 2 para estereofonía.
nSamplesPerSec	Frecuencia de muestreo (Hz) del fichero WAVE: 32 000, 44 100 ó 48 000, etc. Obsérvese, no obstante, que si la frecuencia de muestreo de los datos es variable, este campo se debe poner a cero. Se recomienda encarecidamente que se utilice una frecuencia de muestreo fija para aplicaciones de computadores de mesa.
nAvgBytesPerSec	Velocidad de datos media; pudiera no ser igual a la velocidad binaria MPEG si se utiliza codificación de velocidad binaria variable en la capa III.
nBlockAlign	<p>La alineación de bloques (en octetos) de los datos en <data-ck>. Para trenes de audio que tienen una longitud de trama de audio fija, la alineación de bloques es igual a la longitud de la trama. Para los trenes en los que la longitud de trama varía, <nBlockAlign> se debe poner a 1.</p> <p>Con una frecuencia de muestreo de 32 ó 48 kHz, el tamaño de una trama de audio MPEG depende de la velocidad binaria. Si un tren de audio utiliza una velocidad binaria constante, el tamaño de las tramas de audio no varía. Por consiguiente, se aplican las siguientes fórmulas:</p> <p>Capa I: $nBlockAlign = 4 * (int)(12 * BitRate / SamplingFreq)$</p> <p>Capas II y III: $nBlockAlign = (int)(144 * BitRate / SamplingFreq)$</p> <p>Ejemplo 1: Para la capa I, con una frecuencia de muestreo de 32 000 Hz y una velocidad binaria de 256 kbit/s, $nBlockAlign = 384$ octetos.</p> <p>Si un tren de audio contiene tramas con diferentes velocidades binarias, la longitud de las tramas varía dentro del tren. Se producen también longitudes de trama variables cuando se utiliza una frecuencia de muestreo de 44,1 kHz: para mantener la velocidad de datos al valor nominal, el tamaño de una trama audio MPEG se aumenta periódicamente en un «intervalo» (4 octetos en la capa I, 1 octeto en las capas II y III) en comparación con las fórmulas indicadas anteriormente. En estos dos casos, el concepto de alineación de bloques no es válido. Por consiguiente, el valor de <nBlockAlign> debe ponerse a 1, de modo que las aplicaciones MPEG puedan indicar si los datos están alineados en bloque o no.</p>

NOTA 1 – Obsérvese que es posible construir un tren de audio que tiene tramas de audio de longitud constante a 44,1 kHz poniendo el bit de relleno en el encabezamiento de cada trama de audio al mismo valor (0 ó 1). Obsérvese, no obstante que la velocidad binaria del tren resultante no corresponderá exactamente con el valor nominal del encabezamiento de trama, por lo que algunos decodificadores pueden no ser capaces de decodificar el tren correctamente. En aras de la normalización y de la compatibilidad, se desaconseja este método.

WBitsPerSample	No utilizado, puesto a cero.
CbSize	El tamaño en octetos de la información ampliada después de la estructura WAVEFORMATEX. Para el formato WAVE_FORMAT_MPEG normalizado, éste es 22 (0x0016). Si se añaden campos suplementarios, este valor aumentará.
fwHeadLayer	<p>La capa audio MPEG, definida por las siguientes banderas:</p> <p style="padding-left: 40px;">ACM_MPEG_LAYER1 – capa I.</p> <p style="padding-left: 40px;">ACM_MPEG_LAYER2 – capa II.</p> <p style="padding-left: 40px;">ACM_MPEG_LAYER3 – capa III.</p> <p>Algunos trenes MPEG legales pueden contener tramas de diferentes capas. En este caso, las banderas anteriores deben ser puestas a OR (O exclusiva) juntas, de modo que un servidor (driver) pueda determinar qué capas están presentes en el tren.</p>
dwHeadBitrate	La velocidad binaria de los datos, en bits por segundo. Este valor debe ser una velocidad binaria normalizada de acuerdo con la especificación MPEG; no todas las velocidades binarias son válidas para todos los modos y capas. Véanse los Cuadros 1 y 2. Obsérvese que este campo registra la velocidad binaria real, no el código de encabezamiento de trama MPEG. Si la velocidad binaria es variable, o si no es una velocidad binaria normalizada, este campo se debe poner a cero. Se recomienda que se evite la codificación de velocidad binaria variable, cuando sea posible.
fwHeadMode	<p>Modo tren, definido por las siguientes banderas:</p> <p style="padding-left: 40px;">ACM_MPEG_STEREO – estereofonía.</p> <p style="padding-left: 40px;">ACM_MPEG_JOINTSTEREO – estereofonía asociada.</p> <p style="padding-left: 40px;">ACM_MPEG_DUALCHANNEL – canal dual (por ejemplo, un tren bilingüe).</p> <p style="padding-left: 40px;">ACM_MPEG_SINGLECHANNEL – monocal.</p> <p>Algunos trenes MPEG pueden contener tramas de diferentes modos. En este caso, las banderas anteriores deben ser puestas a OR juntas de modo que un servidor pueda indicar los modos que están presentes en el tren. Esta situación es particularmente probable con codificación de estereofonía asociada, pues los codificadores pueden encontrar útil conmutar dinámicamente entre estereofonía y estereofonía asociada, de acuerdo con las características de la señal. En este caso, se deben fijar ambas banderas ACM_MPEG_STEREO y ACM_MPEG_JOINTSTEREO.</p>
fwHeadModeExt	<p>Contiene parámetros suplementarios para codificación de estereofonía asociada; no se utiliza para otros modos. Véase el Cuadro 3. Algunos trenes MPEG legales pueden contener tramas de diferentes extensiones de modo. En este caso, los valores del Cuadro 3 pueden ser puestas a OR juntos. Obsérvese que fwHeadModeExt se utiliza solamente para codificación de estereofonía asociada; para otros modos (monocal, canal dual o estereofonía) se debe poner a cero.</p> <p>En general, los codificadores conmutarán dinámicamente entre los distintos valores de extensión de modo posibles de acuerdo con las características de la señal. Por consiguiente, para la codificación de estereofonía asociada normal, este campo se debe poner a 0x000f. Sin</p>

embargo, si es conveniente limitar el codificador a un tipo determinado de codificación de estereofonía asociada, este campo se puede utilizar para especificar los tipos admisibles.

wHeadEmphasis	Describe la desacentuación requerida por el decodificador; esto implica la acentuación realizada en el tren antes de la codificación. Véase el Cuadro 4.
fwHeadFlags	Fija las banderas correspondientes en el encabezamiento de la trama audio: <ul style="list-style-type: none"> ACM_MPEG_PRIVATEBIT - fija el bit privado. ACM_MPEG_COPYRIGHT - fija el bit de derechos de autor. ACM_MPEG_ORIGINALHOME - fija el bit original/interno. ACM_MPEG_PROTECTIONBIT - fija el bit de protección e inserta un código de protección contra errores de 16 bits en cada trama. ACM_MPEG_ID_MPEG1 - fija el bit ID a 1, que define el tren como un tren de audio MPEG-1. <i>Esta bandera se debe fijar siempre explícitamente para mantener la compatibilidad con futuras extensiones de audio MPEG (es decir, MPEG-2).</i> <p>Un codificador utilizará el valor de estas banderas para fijar los bits correspondientes en el encabezamiento de cada trama de audio MPEG. Cuando se describe un tren de datos codificados, estas banderas representan una OR lógica de las banderas fijadas en cada encabezamiento de trama. Es decir, si el bit de derechos de autor se fija en uno o más encabezamientos de trama del tren, se fijará la bandera ACM_MPEG_COPYRIGHT. Por tanto, el valor de estas banderas no es necesariamente válido para cada trama audio.</p>
dwPTSLow	Este campo (junto con el siguiente) consiste en el sello de hora de presentación (PTS, <i>presentation time stamp</i>) de la primera trama del tren de audio de la capa de sistema MPEG. dwPTSLow contiene los 32 bits menos significativos (LSB) del PTS de 33 bits. El PTS se puede utilizar para facilitar la reintegración de un tren de audio con un tren vídeo asociado. Si el tren audio no está asociado con una capa de sistema, este campo se debe poner a cero.
dwPTSHigh	Este campo (junto con el anterior) consiste en el sello de hora de presentación (PTS) de la primera trama del tren de audio de la capa de sistema MPEG. El bit menos significativo (LSB) de dwPTSHigh contiene el bit más significativo del PTS de 33 bits. El PTS se puede utilizar para facilitar la reintegración de un tren de audio con un tren de vídeo asociado. Si el tren de audio no está asociado con una capa de sistema, este campo se debe poner a cero.

NOTA 2 – Los dos campos anteriores pueden ser tratados como un solo entero de 64 bits; facultativamente, el campo dwPTSHigh puede ser probado como una bandera para determinar si el bit más significativo está fijado o no.

CUADRO 1

Velocidades binarias admisibles (bit/s)

Códigoe encabezamiento de trama MPEG	Capa I	Capa II	Capa III
'0000'	formato libre	formato libre	formato libre
'0001'	32000	32000	32000
'0010'	64000	48000	40000
'0011'	96000	56000	48000
'0100'	128000	64000	56000
'0101'	160000	80000	64000
'0110'	192000	96000	80000
'0111'	224000	112000	96000
'1000'	256000	128000	112000
'1001'	288000	160000	128000
'1010'	320000	192000	160000
'1011'	352000	224000	192000
'1100'	384000	256000	224000
'1101'	416000	320000	256000
'1110'	448000	384000	320000
'1111'	prohibido	prohibido	prohibido

CUADRO 2

Combinaciones de velocidades binarias de modos admisibles para la capa II

Velocidad binaria (bit/s)	Modos admisibles
32000	monocanal
48000	monocanal
56000	monocanal
64000	todos los modos
80000	monocanal
96000	todos los modos
112000	todos los modos
128000	todos los modos
160000	todos los modos
192000	todos los modos
224000	estereofónico, estereofónico de intensidad, canal dual
256000	estereofónico, estereofónico de intensidad, canal dual
320000	estereofónico, estereofónico de intensidad, canal dual
384000	estereofónico, estereofónico de intensidad, canal dual

CUADRO 3

Extensión de modos

fwHeadModeExt	Código de encabezamiento de trama MPEG	Capas I y II	Capa III
0x0001	'00'	subbandas 4-31 en estereofonía de intensidad	ninguna codificación de estereofonía de intensidad o estereofonía MS
0x0002	'01'	subbandas 8-31 en estereofonía de intensidad	estereofonía de intensidad
0x0004	'10'	subbandas 12-31 en estereofonía de intensidad	estereofonía MS
0x0008	'11'	subbandas 16-31 en estereofonía de intensidad	codificación de estereofonía de intensidad y estereofonía MS

CUADRO 4

Campode acentuación

wHeadEmphasis	Código de encabezamiento de trama MPEG	Desacentuación requerida
1	'00'	ninguna acentuación
2	'01'	acentuación 50/15 μ s
3	'10'	reservado
4	'11'	Rec. UIT-T J.17

1.3 Banderas utilizadas en campos de datos

fwHeadLayer

Las siguientes banderas se definen para el campo <fwHeadLayer>. Para la codificación, una de estas banderas se debe fijar de manera que el codificador sepa qué capa ha de utilizar. Para la decodificación, el servidor puede comprobar estas banderas para determinar si es capaz de decodificar el tren. Obsérvese que un tren MPEG legal puede utilizar diferentes capas en diferentes tramas dentro de un solo tren. Por consiguiente, se puede fijar más de una de estas banderas.

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

fwHeadMode

Las siguientes banderas se definen para el campo <fwHeadMode>. Para la codificación, una de estas banderas se debe fijar de modo que el codificador sepa qué capa [¿modo ?] ha de utilizar; para la codificación de estereofonía asociada, típicamente las banderas ACM_MPEG_STEREO y ACM_MPEG_JOINTSTEREO se fijarán de modo que el codificador pueda utilizar la codificación de estereofonía asociada solamente cuando es más eficaz que la codificación de estereofonía. Para la decodificación, el servidor puede comprobar estas banderas para determinar si es capaz de

decodificar el tren. Obsérvese que un tren MPEG legal puede utilizar diferentes capas en diferentes tramas dentro de un solo tren. Por consiguiente, se puede fijar más de una de estas banderas.

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

fwHeadModeExt

El Cuadro 3 define banderas para el campo <fwHeadModeExt>. Este campo se utiliza solamente para codificación de estereofonía asociada; para otros modos de codificación, este campo se debe poner a cero. Para codificación de estereofonía asociada, estas banderas indican los tipos de codificación de estereofonía asociada que un codificador puede utilizar. Normalmente, un codificador seleccionará de manera dinámica la extensión de modo más apropiada para la señal de entrada; en consecuencia, una aplicación fijaría típicamente este campo a 0x000f de modo que el codificador pueda seleccionar entre todas las posibilidades; sin embargo, es posible limitar al codificador suprimiendo algunas de estas banderas. Para un tren codificado, este campo indica los valores del campo MPEG *mode_extension* que están presentes en el tren.

fwHeadFlags

Las siguientes banderas se definen para el campo <fwHeadFlags>. Estas banderas se deben fijar antes de la codificación de manera que los bits apropiados se fijen en el encabezamiento de la trama MPEG. Cuando se describe un tren de audio MPEG codificado, estas banderas representan una OR lógica de los bits correspondientes en el encabezamiento de cada trama audio. Es decir, si el bit está fijado en cualquiera de las tramas, está fijado en el campo <fwHeadFlags>. Si una aplicación coloca un encabezamiento RIFF WAVE alrededor de un tren de bits de audio MPEG precodificado, es responsable de analizar el tren de bits y fijar las banderas en este campo.

```
#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 Datos de audio en ficheros MPEG

El <segmento de datos> (<**data chunk**>) consiste en una secuencia de audio MPEG-1 definida por la Norma ISO 11172, Parte 3 (Audio). Esta secuencia consiste en un tren de bits, que se almacena en el segmento de datos como un conjunto de octetos. Dentro de un octeto, el bit más significativo (MSB) es el primer bit del tren, y el bit menos significativo (LSB) es el último bit. Los datos *no* son invertidos en el octeto. Por ejemplo, los siguientes datos consisten en los primeros 16 bits (de izquierda a derecha) de un encabezamiento de trama de audio típico:

syncword	ID	Layer	ProtectionBit	...
11111111111111	1	10	1	...

Estos datos serían almacenados en octetos en el orden siguiente:

```
Byte0 Byte1 ...
FF    FD    ...
```

1.4.1 Tramas de audio MPEG

Una secuencia de audio MPEG consiste en una serie de tramas de audio, cada una de las cuales comienza con un encabezamiento de trama. La mayoría de los campos dentro de este encabezamiento de trama corresponden con campos en la estructura MPEG1WAVEFORMAT definida

anteriormente. Para la codificación, estos campos se pueden fijar en la estructura MPEG1WAVEFORMAT y el servidor puede utilizar esta información para fijar los bits apropiados en el encabezamiento de trama cuando codifica. Para la decodificación, el servidor puede comprobar estos campos con el fin de determinar si es capaz de decodificar el tren.

1.4.2 Codificación

Un servidor que codifica un tren de audio MPEG debe leer los campos de encabezamiento en la estructura MPEG1WAVEFORMAT y fijar los bits correspondientes en el encabezamiento de la trama MPEG. Si el servidor requiere alguna otra información, debe obtenerla de una casilla de diálogo de configuración, o a través de una función de devolución de llamada (callback). Para más información, véase más adelante el punto sobre datos auxiliares.

Si un tren de audio MPEG precodificado tiene un encabezamiento RIFF, es función de la aplicación separar el tren de bits en sus partes componentes y fijar los campos en la estructura MPEG1WAVEFORMAT. Si la frecuencia de muestreo o el índice de velocidad binaria no es constante en todo el tren de datos, el servidor debe fijar a cero los correspondientes campos MPEG1WAVEFORMAT <nSamplesPerSec> (número de muestras por segundo) y <dwHeadBitrate> (velocidad binaria de encabezamiento), como se describe anteriormente. Si el tren contiene tramas de más de una capa, debe fijar las banderas en <fwHeadLayer> (capa de encabezamiento) para todas las capas que están presentes en el tren. Como algunos campos tales como <fwHeadFlags> (banderas de encabezamiento) pueden variar de una trama a otra, se debe tener cuidado de fijar y probar estas banderas; en general, una aplicación no debe contar con que sean válidas para cada trama. Cuando se fijan estas banderas, se deben seguir las directrices siguientes:

- ACM_MPEG_COPYRIGHT se debe fijar si cualquiera de las tramas en el tren tiene fijado el bit de derecho de autor.
- ACM_MPEG_PROTECTIONBIT se debe fijar si cualquiera de las tramas en el tren tiene fijado el bit de protección.
- ACM_MPEG_ORIGINALHOME se debe fijar si cualquiera de las tramas en el tren tiene fijado el bit original/interno. Este bit puede ser suprimido si se hace una copia del tren.
- ACM_MPEG_PRIVATEBIT se debe fijar si cualquiera de las tramas en el tren tiene fijado el bit privado.
- ACM_MPEG_ID_MPEG1 se debe fijar si cualquiera de las tramas en el tren tiene fijado el bit ID. Para trenes MPEG-1, el bit ID debe estar siempre fijado; sin embargo, futuras ampliaciones de MPEG (tales como el formato multicanal MPEG-2) pueden tener el bit ID suprimido.

Si el tren de audio MPEG se tomó de un tren MPEG de capa de sistema, o si el tren ha de ser integrado en la capa de sistema, se pueden utilizar los campos de PTS. El PTS es un campo en la capa de sistema MPEG que se utiliza para la sincronización de los distintos campos. El campo MPEG PTS tiene 33 bits y, por consiguiente, el encabezamiento de formato RIFF WAVE almacena el valor en dos campos: <dwPTSLow> (PTS bajo) contiene los 32 LSB del PTS, y <dwPTSHigh> (PTS alto) contiene el MSB. Estos dos campos pueden tomarse juntos como un entero de 64 bits; facultativamente, el campo <dwPTSHigh> puede ser probado como una bandera para determinar si el MSB está fijado o no. Cuando se extrae un tren de audio de una capa de sistema, el servidor debe fijar los campos PTS al PTS de la primera trama de los datos de audio, que se pueden utilizar ulteriormente para reintegrar el tren en la capa de sistema. *Los campos PTS no se deben utilizar para ninguna otra finalidad.* Si el tren de audio no está asociado con la capa de sistema MPEG, los campos PTS se deben poner a cero.

1.4.3 Decodificación

Un servidor puede probar los campos en la estructura MPEG1WAVEFORMAT para determinar si es capaz de decodificar el tren. Sin embargo, el servidor debe saber que algunos campos, tales como el campo <fwHeadFlags>, pueden no concordar para cada trama el tren de bits. El servidor no debe utilizar nunca los campos de la estructura MPEG1WAVEFORMAT para realizar la decodificación real. Los parámetros de decodificación se deben tomar totalmente del tren de datos MPEG.

El servidor puede verificar el campo <nSamplesPerSec> para determinar si admite la frecuencia de muestreo especificada. Si el tren MPEG contiene datos con una velocidad de muestreo variable, el campo <nSamplesPerSec> se pondrá a cero. Si el servidor no puede tratar este tipo de tren de datos, no debe intentar decodificar los datos, sino debe, inmediatamente, dejar de funcionar.

1.5 Datos auxiliares

Los datos de audio en una trama de audio MPEG pueden no llenar toda la trama. Cualesquiera datos restantes se denominan *datos auxiliares*. Estos datos pueden tener cualquier formato deseado, y se pueden utilizar para transferir información adicional de cualquier clase. Si un servidor desea admitir los datos auxiliares, debe tener una facilidad para transferir los datos a y desde la aplicación llamante. El servidor puede utilizar una función de devolución de llamada para esta finalidad. Básicamente, el servidor puede utilizar una función de devolución de llamada especificada cuando tiene datos auxiliares para transferir a la aplicación (es decir, decodificar) o cuando requiere más datos auxiliares (codificar).

Los servidores deben saber que no todas las aplicaciones desearán procesar los datos auxiliares. Por consiguiente, un servidor sólo debe proporcionar este servicio cuando es solicitado explícitamente por la aplicación. El servidor puede definir un mensaje individualizado que habilite e inhabilite la facilidad de devolución de llamada. Se podrán definir mensajes distintos para las operaciones de codificación y de decodificación, para mayor flexibilidad.

Obsérvese que este método puede no ser apropiado para todos los servidores o todas las aplicaciones; se incluye solamente como una ilustración sobre cómo se pueden admitir datos auxiliares.

NOTA – En <MPEG_Audio_Extension_chunk> figura más información sobre datos auxiliares que se debe utilizar para los ficheros MPEG conformes al formato de onda de radiodifusión. Véase el § 2 del texto principal del Anexo 2.

Referencias Bibliográficas

ISO/CEI 11173-3: MPEG 1.

ISO/CEI 13818-3: MPEG 2.

NOTA – Los documentos de Microsoft[®] están disponibles en la siguiente dirección Internet: <http://www.microsoft.com>.

Anexo 3

Especificación del BWF

Un formato para ficheros de datos de audio en radiodifusión

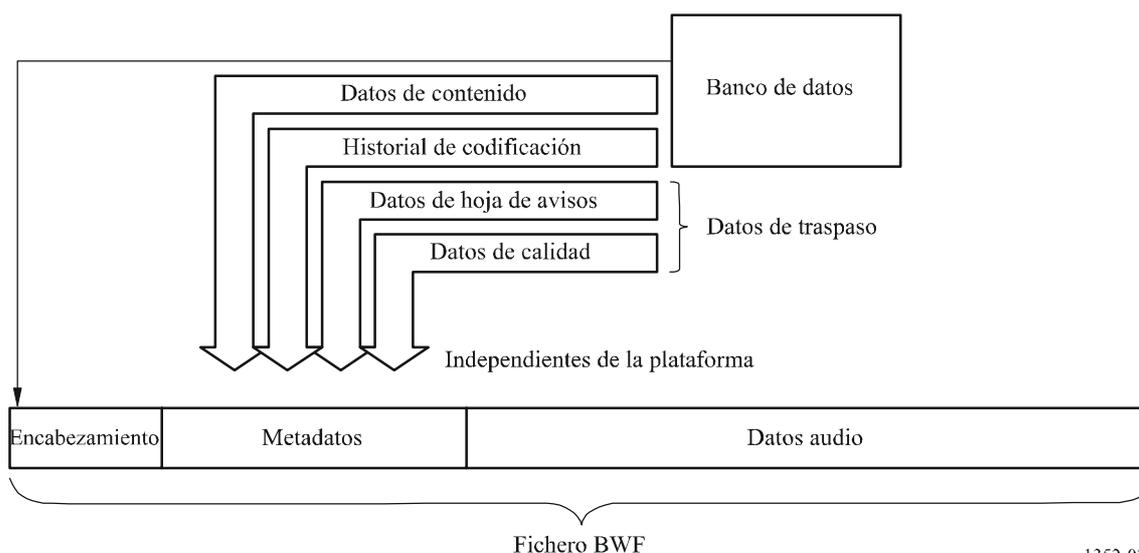
ESPECIFICACIONES DE METADATOS

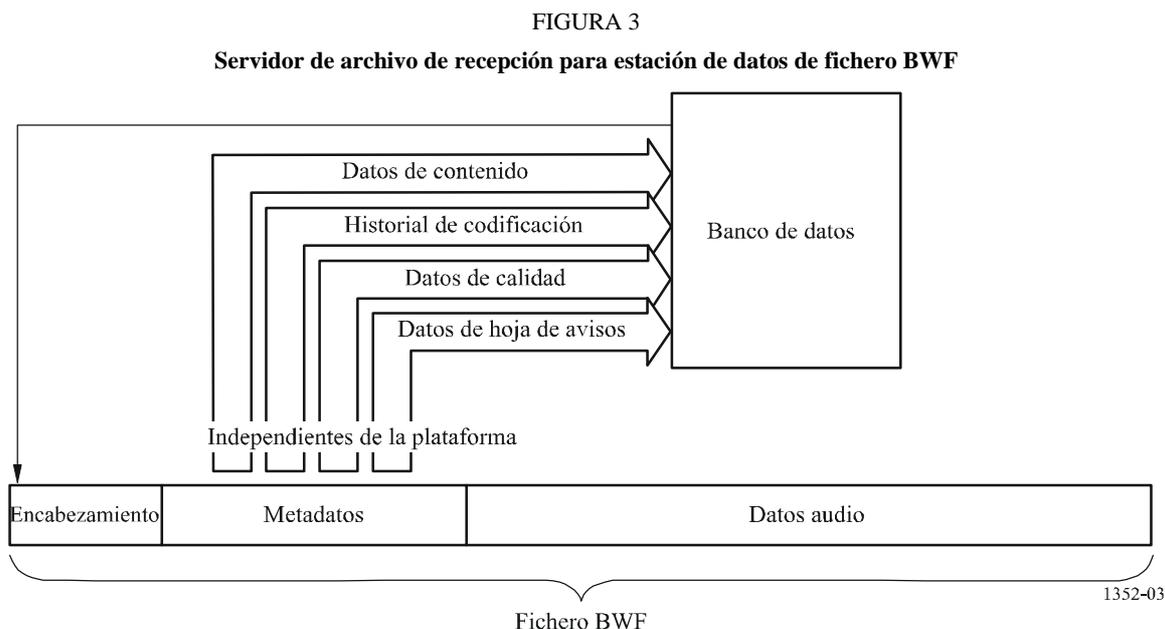
1 Introducción

Este Anexo contiene la especificación de la utilización del BWF para cursar información sobre el material de programas de audio recopilado y reordenado por una DAW (véase la Fig. 2). El fichero BWF se utiliza como contenedor independiente de la plataforma para la señal de sonido y todos los metadatos pertinentes. El servidor de archivo de recepción es capaz de extraer la información necesaria del fichero y utilizarla a conveniencia; por ejemplo, introducirla en el banco de datos, etc. (Fig. 3).

FIGURA 2

Recogida de datos por una estación de trabajo en un fichero BWF





Este Anexo especifica un nuevo segmento para cursar información que no está aún presente en un fichero BWF básico y especifica también la forma en que deben utilizarse los segmentos actuales del BWF.

Hay que tener cautela al editar ficheros BWF que contengan informes de calidad. Si un sistema de edición combina más de un fichero BWF, la lista de decisión de edición (EDL, *edit decision list*) debe apuntar a las partes adecuadas de los segmentos de historial de codificación y de calidad de cada fichero fuente BWF. Además, si se obtiene un nuevo fichero a partir de partes de otros ficheros, debe producirse para el nuevo un nuevo segmento de historial de codificación y de calidad.

2 Informe de captura

Para salvaguardar archivos que guardan portadoras únicas analógicas o digitales originales, es importante volver a grabar la señal de sonido original con calidad plena en los ficheros de BWF. Un informe de captura contiene información sobre toda la cadena de procesamiento, desde el dominio analógico al digital, o para las transferencias desde el dominio digital (por ejemplo, de CD o DAT).

El informe de captura se establece, junto con los datos procedentes del análisis de la señal de audio, formando parte de los metadatos del fichero BWF.

El informe de captura consta de tres partes:

- Campo de historial de codificación en el segmento <bext> del fichero BWF. Contiene detalles de toda la cadena de transmisión, por ejemplo, del tipo de cinta magnética, disco compacto o casete DAT hasta el fichero BWF (historial de la señal de sonido).
- El informe de calidad del segmento <qlty>. Contiene información que describe todos los sucesos pertinentes que afectan a la calidad de la señal de sonido grabada en el segmento de datos de onda. Cada suceso, ya sea reconocido por el operador o por el computador, se enumera con detalles del tipo de evento, sellos temporales exactos, prioridad y estado del evento. También se informa de parámetros generales de calidad, etc.
- La hoja de avisos en el segmento <qlty> es una lista de sucesos marcada con sellos temporales exactos y una descripción adicional de la señal de sonido, por ejemplo, el inicio de un área o el punto de partida de un parlamento importante. De esta manera, los archivistas pueden completar los metadatos del banco de datos con útiles informáticos.

2.1 Sintaxis del informe de captura

- El informe de captura consta de cadenas de caracteres ASCII (ISO 646) [ISO/IEC, 1991] dispuestas en filas de hasta 256 caracteres.
- Cada fila debe terminar por <CR/LF> (ASCII 0Dh, 0Ah).
- Una fila puede contener una o más cadenas variables separadas por comas (ASCII 2Bh).
- Las cadenas variables van en caracteres ASCII y no deben contener comas.
- Los puntos y comas (ASCII 3Bh) deben utilizarse como separadores dentro de las cadenas variables.

3 Campo CodingHistory en el segmento <bext>

Las cadenas utilizadas en el campo de historial de codificación se especifican en el Apéndice 2 del Anexo 1. A continuación se repite esta información por conveniencia.

A=<ANALOGUE,>	Información sobre el trayecto de la señal de sonido analógica
A=<PCM,>	Información sobre el trayecto de la señal de sonido digital
F=<48000, 441000, etc.>	Frecuencia de muestreo [Hz]
W=<16, 18, 20, 22, 24, etc.>	Longitud de palabra [bits]
M=<mono, stereo, 2-channel>	Modo
T=<free ASCII-text string>	Texto para comentarios

4 Segmento de calidad

El segmento de calidad se define en el texto en cursiva del § 4.1:

4.1 Elementos de segmento de calidad

FileSecurityReport: Este campo contiene el FileSecurityCode del segmento. QualityChunk
Es un valor de 32 bits que contiene la checksum [0 ...231].

FileSecurityWave: Este campo contiene el FileSecurityCode de los datos de la onda BWF.
Es un valor de 32 bits que contiene la checksum [0 ...231].

Quality-chunk typedef struct {

DWORD ckID; / (segmento de calidad) ckID='qlty' */*

DWORD ckSize; / tamaño del segmento de calidad */*

BYTE ckData[ckSize]; / datos del segmento */*

}

typedef struct quality_chunk {

DWORD FileSecurityReport; / código de FileSecurityCode del informe de calidad */*

DWORD FileSecurityWave; / Código de FileSecurityCode de los datos de onda BWF */*

CHAR BasicData[]; / ASCII: « Datos básicos » */*

CHAR StartModulation[]; / ASCII: « Datos de inicio de modulación » */*

CHAR QualityEvent[]; / ASCII: « Datos de evento de calidad » */*

CHAR EndModulation[]; / ASCII: « Datos de fin de modulación » */*

CHAR QualityParameter[]; / ASCII: « Datos de parámetros de calidad » */*

CHAR OperatorComment[]; / ASCII: « Comentarios de operador » */*

CHAR CueSheet[]; / ASCII: « Datos de hoja de avisos » */*

} quality-chunk

BasicData:	Datos básicos de captura.
B=	Cadena ASCII que contiene datos básicos sobre el material de sonido.
Archive No. (AN):	Número de archivo (máximo 32 caracteres).
Title (TT):	Título/Toma de los datos de sonido (máximo 256 caracteres).
Duration (TD):	10 caracteres ASCII que contienen la duración de la secuencia de sonido. Formato: «hh:mm:ss:d» Horas hh: 0...23 Minutos mm: 0...59 Segundos ss: 0...59 1/10 de segundo d: 0...9
Date (DD):	10 caracteres ASCII que contienen la fecha de digitalización. Formato: «yyyy:mm:dd» Año yyyy: 0000...9999 Mes mm: 0...12 Día dd: 0...31
Operator (OP):	Cadena ASCII (máximo 64 caracteres) que contiene el nombre de la persona que efectúa la operación de digitalización.
Copying station (CS):	Cadena ASCII (máximo 64 caracteres) que contiene el tipo y el número de serie de la estación de trabajo utilizada para crear el fichero.
StartModulation:	Inicio de la modulación de la grabación original.
SM=	10 caracteres ASCII que contienen el instante de inicio de la señal de sonido tomada del principio del fichero. Formato: «hh:mm:ss:d» Horas hh: 0...23 Minutos mm: 0...59 Segundos ss: 0...59 1/10 de segundo d: 0...9
Sample count (SC):	Código de dirección de muestra del punto SM del principio del fichero (inicio hexadecimal de modulación). Formato: «#####H» 0H...FFFFFFFFH (0.....4,295 × 10 ⁹)
Comment (T):	Cadena ASCII que contiene comentarios.
QualityEvent	Información que describe cada evento de calidad en la señal de sonido. Se utiliza una cadena de evento de calidad para cada evento.
Q=	Cadena ASCII (máximo 256 caracteres) que contiene eventos de calidad.
Event number (M):	Marca numerada que crea manualmente el operador. Formato: «M###» ###: 001...999
Event number (A):	Marca numerada que crea automáticamente el sistema. Formato: «A###» ###: 001...999
Priority (PRI):	Prioridad del evento de calidad Formato: «#» #: 1 (LO).....5 (HI)

Time stamp (TS):	10 caracteres ASCII que contienen el sello temporal del evento de calidad del inicio del fichero.	
	Formato: «hh:mm:ss:d»	
	Horas	hh: 0...23
	Minutos	mm: 0...59
	Segundos	ss: 0...59
	1/10 de segundo	d: 0...9
Event type (E):	Cadena ASCII (máximo 16 caracteres) que describe el tipo de evento, por ejemplo, «Click», «AnalogOver», «Transparency» o parámetros de calidad (QualityParameters) (definidos más adelante) que exceden límites, por ejemplo, «QP:Azimuth:L-20.9smp».	
Status (S):	Cadena ASCII (máximo 16 caracteres) que contiene el estado de procesamiento del evento, por ejemplo, «unclear», «checked», «restored», «deleted».	
Comment (T):	Cadena ASCII que contiene comentarios.	
Sample count (SC):	Código de dirección de muestra del punto TS del comienzo del fichero (ASCII hexadecimal).	
	Formato: «#####H»	
	0H...FFFFFFFFH (0...4,295 × 10 ⁹)	
QualityParameter	Parámetros de calidad que describen la señal de sonido	
P=	Cadena ASCII (máximo 256 caracteres) que contiene parámetros de calidad.	
Parameters (QP):	MaxPeak: (Cresta máxima)	-xx.x dBFSL; [-99.9.....-00.0]
	MeanLevel: (Nivel medio)	-xx.x dBFSL;-yy.y dBFSR [-99.9.....-00.0]
	Correlation: (Correlación)	±x.x [-1.0.....+1.0]
	Dynamic: (Gama dinámica)	xx.x dBL; yy.y dBR [00.0.....99.9]
	ClippedSamples: (Muestras recortadas)	xxxx smpL; yyyy smpR [0.....9999]
	SNR: (Relación señal/ruido)	xx.x dBL; yy.y dBR [00.0.....99.9]
	Bandwidth: (Anchura de banda)	xxxxx HzL; yyyyy HzR [0.....20000]
	Azimuth: (Azimuth)	L±xx.x smp [-99.9.....+99.9]
	Balance: (Balance)	L±x.x dB [-9.9.....+9.9]

DC-Offset: (Separación DC)	x.x %L; y.y %R	[0.0.....9.9]
Speech: (Conversación)	xx.x%	[0.0.....99.9]
Stereo: (Estéreo)	xx.x%	[0.0.....99.9]
	(L = canal izquierdo, R = canal derecho)	
Quality factor (QF):	Factor de calidad resumen del fichero de sonido [1...5 (óptimo), 0 = no definido]	
Inspector (IN):	Cadena ASCII (máximo 64 caracteres) que contiene el nombre de la persona que inspecciona el fichero de sonido	
File status (FS):	Cadena de caracteres ASCII que describe el estado «¿dispuesto para transmisión?» [Y(sí)/N(no)/U: Fichero dispuesto/no dispuesto/FS no está definido]	
OperatorComment	Comentarios de operador	
T=	Cadena ASCII (máximo 256 caracteres) que contiene comentarios.	
EndModulation	Fin de modulación	
EM=	10 caracteres ASCII que contienen el final del tiempo de modulación de la señal de sonido. Formato: «hh:mm:ss:d» Horas h: 0...23 Minutos m: 0...59 Segundos s: ...59 1/10 de segundo d: 0...9	
Sample count (SC):	Código de dirección de muestra del punto EM del comienzo del fichero (ASCII hexadecimal). Formato: «#####H» 0H...FFFFFFFH (0.....4,295 × 10 ⁹)	
Comment (T):	Cadena ASCII que contiene comentarios.	
CueSheet	Datos de la hoja de aviso	
C=	Cadena ASCII (máximo 256 caracteres) que contiene puntos de aviso.	
Cue number (N):	Número del punto de aviso generado automáticamente por el sistema. Formato: «N###»###: 001...999	
Time stamp (TS):	10 caracteres ASCII que contienen el sello temporal del evento de calidad del inicio del fichero. Formato: «hh:mm:ss:d» Horas hh: 0...23 Minutos mm: 0...59 Segundos ss: 0...59 1/10 de segundo d: 0...9	

Text (T): Cadena ASCII que contiene la descripción de comentarios del punto de aviso,
por ejemplo, «Inicio de un aria».

Sample count (SC): Código de dirección de muestra del punto TS (inicio hexadecimal de modulación).
Formato: «#####H»
0H.....FFFFFFFFH ($0 \dots 4,295 \times 10^9$)

5 Ejemplos de informe de captura

5.1 Proceso de digitalización del material analógico

(información básica contenida en el campo CodingHistory del segmento <bext>)

Número
de línea

01 A=ANALOGUE, M=stereo, T=Studer A816; SN1007; 38; 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>

(Informe de calidad en el segmento de calidad)

Número
de línea

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, PRIG, TS=00:20:01:6, E=Click0n, 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; -2.8dBFSR<CR/LF>
18 P=QP:MeanLevel:-11.5dBFS; 8.3dBFSR<CR/LF>

- 19 P=QP:Correlation:+0.8<CR/LF>
 20 P=QP:Dynamic:51.4dBL;49.6dBR<CR/LF>
 21 PAP:ClippedSamples:Osmpl;Osmpr<CR/LF>
 22 P=QP:SNR:32.3dBL;35.1dBR<CR/LF>
 23 P=QP:Bandwidth:8687HzL;7943HzR<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%L;0.0%R<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>

(CueSheet en el segmento de calidad)

Número
de línea

- 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>

Interpretación del Ejemplo 1

(información básica en el campo CodingHistory)

- Línea 1: La cinta magnética analógica de tipo Agfa PER528 se reproduce en un grabador de cinta Studer A816 con número de serie 1007, utilizando un expansor telcom:
 Velocidad de cinta: 38 cm/s
 Modo: estéreo
- Línea 2: Para la digitalización se utiliza un convertidor A/D de tipo NVision NV 1000 con:
 Frecuencia de muestreo: 48 kHz
 Resolución de codificación: 18 bits
 Modo: estéreo
- Línea 3: El fichero original se graba como fichero BWF lineal con codificación MIC utilizando la entrada digital de la estación de regrabación sin yuxtaposición:
 Frecuencia de muestreo: 48 kHz
 Resolución de codificación: 16 bits por muestra
 Modo: estéreo

(QualityReport en el segmento de calidad)

- Líneas 1 a 2: Código de seguridad de fichero de segmento de calidad y de datos de onda.
- Líneas 3 a 5: El operador utiliza la estación de regrabación QUADRIGA2.0 con número de serie 10012
 (OP). La cinta tiene el número de archivo (AN) y el título (TT), y fue digitalizada en la fecha.
 (DD). La duración de la señal de sonido en el BWF es TD.

- Línea 6: Comienzo de la modulación (SM) en el sello temporal (TS) y cómputo de muestra (SC) con comentarios (T).
- Líneas 7 a 15: Eventos (E) reconocidos por el operador (M) y/o control de sistema (A) con prioridad (PRI) y un sello temporal (TS). El estado del evento (S) y los comentarios (T) dan más información. El cómputo de muestra (SC) da el sello temporal preciso.
- Línea 16: Fin de la modulación (EM) en el sello temporal y cómputo de muestra (SC), con comentarios (T).
- Líneas 17 a 28: Parámetros de calidad (QP de toda la señal de sonido en el segmento de datos de onda.
- Líneas 29 a 31: Factor de resumen (QF) dado por el control automático de sistema y el nombre del inspector IN, y la decisión (FS) de si la calidad del fichero de sonido está dispuesta para transmisión.

(CueSheet en el segmento de calidad)

- Líneas 32 a 33: Los puntos de avisos marcan el inicio de una conversación y el punto de comienzo de un área.

5.2 Proceso de captura de un disco compacto

(información básica en el campo CodingHistory del segmento <bext>)

Número
de línea

- 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>

(QualityReport en el segmento de calidad)

Número
de línea

- 01 <FileSecurityReport>
- 02 <FileSecurityWave>

etc: similar al ejemplo del § 5.1 anterior.

(CueSheet en el segmento de calidad)

similar al ejemplo del § 5.1

Interpretación del Ejemplo 2

(información básica en el campo CodingHistory)

- Línea 1: Un disco compacto que se puede grabar de tipo Mitsui CD-R74 se reproduce en un reproductor de disco compacto Sony CPD-D500 con número de serie 2172:
- | | |
|-----------------------------|---------------------|
| Frecuencia de muestreo: | 44,1 kHz |
| Resolución de codificación: | 16 bits por muestra |
| Modo: | estéreo |

Línea 2: Se utiliza un convertidor de velocidad de muestras de tipo DCS972 con:

Frecuencia de muestreo: 48 kHz (de 44,1 kHz)
 Resolución de codificación: 24 bits por muestra
 Modo: estéreo

Línea 3: El fichero original se graba como fichero BWF lineal con codificación MIC utilizando la entrada digital de la estación de grabación sin yuxtaposición:

Frecuencia de muestreo: 48 kHz
 Resolución de codificación: 24 bits por muestra
 Modo: estéreo

(QualityReport en el segmento de calidad)

Líneas 1 a 2: Código de seguridad de fichero de segmento de calidad y de datos de onda.

Los otros datos se utilizan conforme al proceso de captura del disco compacto similar al del Ejemplo 1 del § 5.1.

(CueSheet en el segmento de calidad)

Los datos de la hoja de avisos se utilizan conforme al proceso de captura del disco compacto similar al del Ejemplo del § 5.1.

5.3 Proceso de captura de una casete DAT

(información básica del campo CodingHistory del segmento <bext>)

Número
de línea

01 A=PCM, F=48000, W=16, M=stereo, T=SonyPCM-8500; SN1037; TDK DA-R120<CR/LF>
 02 A=PCM, F=48000, W=16, M=stereo, T=no dither; DIO<CR/LF>

(QualityReport en el segmento de calidad)

Número
de línea

01 <FileSecurityReport>
 02 <FileSecurityWave>

etc.: similar al ejemplo del punto § 5.1 anterior.

(CueSheet en el segmento de calidad)

similar al ejemplo del § 5.1 anterior.

Interpretación del Ejemplo 3

(información básica en el campo CodingHistory)

Línea 1: Una casete DAT de tipo TDK DA-8120 se reproduce en un registrador DAT Sony PCM-8500 con número de serie 1037:

Frecuencia de muestreo: 48 kHz
 Resolución de codificación: 16 bits por muestra
 Modo: estéreo

Línea 2: El fichero original se registra en un fichero BWF lineal con codificación MIC utilizando la entrada digital de la estación de regrabación sin yuxtaposición.

Frecuencia de muestreo: 48 kHz
Resolución de codificación: 16 bits por muestra
Modo: estéreo

(QualityReport en el segmento de calidad)

Líneas 1 a 2: Códigos de seguridad de fichero del segmento de calidad y de los datos de onda.

Los otros datos se utilizan conforme al proceso de captura de la casete DAT, similar al del Ejemplo 1 del § 5.1.

(CueSheet en el segmento de calidad)

Los datos de la hoja de avisos se utilizan conforme al proceso de captura de la casete DAT de forma similar a la del Ejemplo 1 del § 5.1
