

DIN EN IEC 62942

ICS 33.160.01

Einsprüche bis 2019-11-06

Entwurf**Dateiformat für die professionelle Übertragung und den Austausch digitaler Audiodaten****(IEC 100/3143/CDV:2018);****Deutsche und Englische Fassung prEN IEC 62942:2018**File format for professional transfer and exchange of digital audio data (TA 6)
(IEC 100/3143/CDV:2018);

German and English version prEN IEC 62942:2018

Format de fichier pour le transfert et l'échange professionnels de données
audionumériques

(IEC 100/3143/CDV:2018);

Version allemande et anglaise prEN IEC 62942:2018

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2019-09-06 wird der Öffentlichkeit zur Prüfung und
Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses
Entwurfs besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal von DIN unter www.din.de/go/entwuerfe bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an dke@vde.com möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/go/stellungnahmen-norm-entwuerfe oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE, Stresemannallee 15, 60596 Frankfurt am Main.

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten
Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 80 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Nationales Vorwort

Die Deutsche Fassung des europäischen Dokuments prEN IEC 62942:2018 „Dateiformat für die professionelle Übertragung und den Austausch digitaler Audiodaten“ (Entwurf in der Umfrage) ist unverändert in diesen Norm-Entwurf übernommen worden.

Die Internationale Elektrotechnische Kommission (IEC) und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dem entsprechend ist das internationale Dokument IEC 100/3143/CDV:2018 „File format for professional transfer and exchange of digital audio data (TA 6)“ unverändert in den Entwurf prEN IEC 62942:2018 übernommen worden.

Da die Deutsche Fassung noch nicht endgültig mit der Englischen und Französischen Fassung abgeglichen ist, ist die englische Originalfassung des IEC-CDV entsprechend der diesbezüglich durch die IEC erteilten Erlaubnis beigefügt. Die Nutzungsbedingungen für den deutschen Text des Norm-Entwurfes gelten gleichermaßen auch für den englischen IEC-Text.

Das internationale Dokument wurde vom TC 100/TA 6 „Storage media, storage data structures, storage systems and equipment“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet.

Bei der Abstimmung zu dem Europäischen Schluss-Entwurf bei CENELEC und dem Internationalen Schluss-Entwurf bei IEC [Final Draft International Standard (FDIS)] sind jeweils nur „JA/NEIN“-Entscheidungen möglich, wobei „NEIN“-Entscheidungen fundiert begründet werden müssen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Für dieses Dokument ist das nationale Arbeitsgremium K 742 „Audio-, Video- und Multimediasysteme, -geräte und -komponenten“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE (www.dke.de) zuständig.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf ein Dokument ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils aktuellste Ausgabe des in Bezug genommenen Dokuments.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe des Dokuments.

Der Zusammenhang der zitierten Dokumente mit den entsprechenden deutschen Dokumenten ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. IEC ist nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

Dateiformat für die professionelle Übertragung und den Austausch digitaler Audiodaten

Inhalt		Seite
Einleitung		4
1 Anwendungsbereich		5
2 Normative Verweisungen		5
3 Begriffe		5
4 BWF-Datei		7
4.1 Vorhandene Segmente werden als Teil des RIFF-Formats definiert		7
4.2 Zusätzliche Segmente		7
4.3 Inhalt eines BWFF		7
4.4 Broadcast Audio Extension-Segment		7
Anhang A (normativ) RIFF-WAVE-Dateiformat		11
A.1 Allgemeines		11
A.2 Resource Interchange File Format (RIFF)		11
A.2.1 Segment (en: chunk; ck)		11
A.2.2 RIFF-Formen		12
A.3 Waveform Audio File Format (WAVE)		12
A.3.1 WAVE-Formatsegment		13
A.3.2 WAVE-Formatkategorien		14
A.4 Speicherung der WAVE-Daten		16
Anhang B (normativ) Segmentreihenfolge		17
Anhang C (normativ) Dateinamenskonventionen		18
C.1 Allgemeines		18
C.2 Dateinamenlänge		18
C.3 Dateinamenerweiterung		18
C.4 Zeichensatz für Dateinamen		18
Anhang D (informativ) Mehrkanal-Verwendung		20
D.1 Allgemeines		20
D.2 Packen von Mehrkanal-Audiodaten		20
D.3 Kanaluordnungen in Mehrkanal-Dateien		21
D.3.1 Verbreitung und Archivierung		21
D.3.2 Produktionsaufzeichnungen		21
Anhang E (informativ) Andere Audiocodierungen		22
E.1 Allgemeines		22
E.2 MPEG-Dateien		22
Anhang F (normativ) Erweitertes Dateiformat (BWF-E)		23
F.1 Einleitung		23

	Seite
F.2 Überschreiten der 4-Gigabyte-Grenze	23
F.2.1 Allgemeines	23
F.2.2 64-Bit Resource Interchange File Format (RF64)	24
F.3 Kompatibilität zwischen BWF und BWF-E	27
F.3.1 Allgemeines	27
F.3.2 Initialisierung als BWF	28
F.3.3 Übergang zu BWF-E	28
F.4 Strukturen von RIFF/WAVE und RF64/WAVE	28
F.4.1 Für das RIFF/WAVE-Format spezifische Segmente und Strukturen	28
F.4.2 Für das RF64/WAVE (BWF-E) Format spezifische Segmente und Strukturen	29
Anhang G (normativ) bext-Segmentversionen	30
G.1 Version 0	30
G.2 Version 1	30
G.3 Version 2	30
Anhang H (normativ) Lautheitsparameter	31
H.1 Behandlung der Lautheitsparameter	31
H.2 Verweisungen zu den Lautheitsparametern	32
H.3 Informative Verweisungen zur Lautheit	32
Anhang I (informativ) Formatfestlegung für den "Unique" Source Identifier (USID) zur Verwendung im Feld <OriginatorReference>	33
I.1 USID	33
I.2 USID-Beispiele	33
Anhang J (informativ) Spezifikation des Formats für das Feld <CodingHistory>	34
J.1 Einleitung	34
J.2 Syntax	34
J.3 Beispiele für Felder der Codierungshistorie	34
Anhang K (normativ) Universal Broadcast Audio Extension-Segment	36
K.1 Allgemeines	36
K.2 Inhalt eines BWFF mit 'ubxt'-Segment	36
K.3 Universal Broadcast Audio Extension-Segment	36
Literaturhinweise	40
Bilder	
Bild D.1 – Packen von Daten für 24-Bit-Mono PCM Audiodaten	20
Bild D.2 – Packen von Daten für 16-Bit-Stereo (2-Kanal) PCM-Audiodaten	20
Bild D.3 – Packen von Daten für 24-Bit 4-Kanal PCM-Audiodaten	20
Bild D.4 – Packen einer 4-Bit-Abtastung	21
Bild F.1 – Herkömmliches RIFF/WAVE-Format	24
Bild F.2 – Erweitertes RF64/WAVE-Format	26
Bild F.3 – Kompatible RIFF/WAVE-Struktur	27

Tabellen

Tabelle 1 – Feldinhaltsdefinitionen für 'bext'	8
Tabelle A.1 – Segmentbeschreibung	12
Tabelle A.2 – Formatsegment – common fields	14
Tabelle A.3 – WAVE-Formatkategorien	14
Tabelle A.4a – Packen von Daten für 16-Bit-Mono-PCM	15
Tabelle A.4b – Packen von Daten für 16-Bit-Stereo-PCM	15
Tabelle A.5 – PCM-Datenformat	16
Tabelle A.6 – PCM-Datenformat – 16-Bit	16
Tabelle A.7 – Beispiele für das PCM WAVE-Format-Segment	16
Tabelle J.1 – CodingHistory-Parameter	34
Tabelle K.1 – ubxt Feldinhaltsdefinitionen	38

Einleitung

Das Broadcast Wave Format File (BWFF) basiert auf dem WAVE Audio File Format von Microsoft, das eine Dateiart darstellt, die im Resource Interchange File Format (RIFF) von Microsoft festgelegt ist. WAVE-Dateien enthalten spezifisch Audiodaten. Der Basisblock einer RIFF-Datei ist ein Segment, das spezifische Angaben, ein Identifikationsfeld und ein Größenfeld enthält. Eine RIFF-Datei enthält eine Anzahl von Segmenten.

Das BWFF enthält insbesondere ein <Broadcast Audio Extension> Segment zum Übertragen bestimmter Metadaten, die für Rundfunk und professionelle Verwendung wichtig sind. Für einen zuverlässigen Austausch gelten einige Einschränkungen für das Format der Audiodaten.

Das Broadcast Wave Format wurde zuerst unter Verwendung von ASCII-Text für alle Felder entwickelt. Im Verlauf der Weiterentwicklung des Formats wurde später die Verwendung von Mehrbyte-Zeichen vorgeschlagen, um das Format zu internationalisieren. Es wurde davon ausgegangen, dass die Verwendung von Mehrbyte-Zeichensätzen im vorhandenen Format bei der Analyse von Mehrbyte-Metadaten durch Anwendungen, die ASCII-Text erwarten, Probleme mit der Kompatibilität verursachen würden. Die Eigenständigkeit visuell lesbarer und maschinenlesbarer Metadaten wurde erstellt und ein neues "Universal"-Segment eingerichtet, um mittels Mehrbyte-Zeichensätzen internationalisierte, visuell lesbare Metadaten ohne Kompatibilitätsprobleme zu übertragen. Dies wird in Anhang K beschrieben.

Dieses Dokument beinhaltet die Spezifikation des Broadcast Audio Extension-Segments und seine Verwendung mit PCM-codierten Audiodaten. Grundlegende Informationen zum RIFF-Format und dessen möglicher Erweiterung um andere Arten von Audiodaten sind in Anhang E aufgeführt. Einzelheiten zum PCM WAVE-Format sind in Anhang A enthalten.

1 Anwendungsbereich

Diese Internationale Norm legt ein Dateiformat zum Austausch von Audiodaten zwischen konformen Einrichtungen fest. Sie ist in erster Linie für Audioanwendungen zur professionellen Aufzeichnung, Produktion, Postproduktion und Archivierung bestimmt.

Sie ist abgeleitet vom EBU Broadcast Wave Format, aber auch kompatibel mit abweichenden Spezifikationen einschließlich ITU-R BR.1352-3-2007 und BWF-J der Japan Post Production Association.

Ein optional erweitertes Format, BWF-E, unterstützt die 64-Bit-Adressierung, um Dateigrößen von mehr als 4 GByte zu ermöglichen.

2 Normative Verweisungen

Die folgenden Dokumente werden im Text in solcher Weise in Bezug genommen, dass einige Teile davon oder ihr gesamter Inhalt Anforderungen des vorliegenden Dokuments darstellen. Bei datierten Verweisungen gilt nur die in Bezug genommene Ausgabe. Bei undatierten Verweisungen gilt die letzte Ausgabe des in Bezug genommenen Dokuments (einschließlich aller Änderungen).

SMPTE ST 330-2011; *SMPTE standard for television – Unique Material Identifier (UMID)*

ISO/IEC 646:1991; *Information technology – ISO-7-bit coded character set for information exchange*

ISO 8601; *Data elements and interchange formats – Information interchange – Representation of dates and times*

ISO/IEC 10646:2012; *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*

IETF RFC 3629; *UTF-8, a transformation format of ISO 10646*

3 Begriffe

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: verfügbar unter <http://www.electropedia.org/>
- ISO Online browsing platform: verfügbar unter <http://www.iso.org/obp>

3.1

Resource Interchange File Format

RIFF

eine Dateidarstellung, auf der das WAVE-Dateiformat beruht

3.2

Segment (en: chunk; ck)

Datenpaket in den RIFF-Dateien, das zugehörige Daten beinhaltet

3.3

ASCII

mit ISO/IEC 646 konformer 7-Bit-Zeichencode

3.4

Waveform Audio File Format

WAVE

ein Audiodateiformat, das auf der RIFF-Dateistruktur beruht

3.5

Broadcast Wave Format File BWFF

WAVE-Datei, die das in dieser Norm beschriebene bext-Segment beinhaltet

3.6

Broadcast Extension-Segment

bext

ein Erweiterungssegment für WAVE

3.7

Universal Broadcast Audio Extension-Segment

ubxt

die visuell lesbaren Informationen des bext-Segments in Mehrbyte-Sprachen

3.8

UMID

eindeutiger Materialbezeichner nach SMPTE ST 330

3.9

Broadcast Wave Format, Extended

BWF-E

ein optional erweitertes Format, das einen RIFF-Kopf durch einen RF64-Kopf ersetzt, um die 64-Bit-Adressierung zu unterstützen, die Dateigrößen von mehr als 4 GByte ermöglicht

3.10

RF64

eine Ersatzstruktur für den RIFF-Dateityp, die eine 64-Bit-Adressierung unterstützt

3.11

CHAR

8-Bit-Integer mit Vorzeichen, der ganze Zahlen von -128 bis +127 repräsentiert. Äquivalenter C Typ: "signed char"

3.12

BYTE

vorzeichenloser 8-Bit-Integer, der ganze Zahlen von 0 bis 255 repräsentiert. Äquivalenter C Typ: "unsigned char"

3.13

INT

16-Bit-Integer mit Vorzeichen, der ganze Zahlen von -32768 bis +32767 repräsentiert. Äquivalenter C Typ: "signed short int". Mehrbyte-Datentypen sind Little-Endian

3.14

WORD

vorzeichenloser 16-Bit-Integer, der ganze Zahlen von 0 bis +65535 repräsentiert. Äquivalenter C Typ: "unsigned short int". Mehrbyte-Datentypen sind Little-Endian

3.15

LONG

32-Bit-Integer mit Vorzeichen, der ganze Zahlen von -2.147.483.648 bis +2.147.483.647 repräsentiert. Äquivalenter C Typ: "signed long int". Mehrbyte-Datentypen sind Little-Endian

3.16

DWORD

vorzeichenloser 32-Bit-Integer, der ganze Zahlen von 0 bis +4294967295 repräsentiert. Äquivalenter C Typ: "unsigned long int". Mehrbyte-Datentypen sind Little-Endian

4 BWF-Datei

4.1 Vorhandene Segmente werden als Teil des RIFF-Formats definiert

Diese Spezifikation verwendet eine Anzahl bereits definierter RIFF-Segmente (siehe Anhang A). Diese sind:

```
<fmt-ck>           Format Chunk
<wave-data>        Audio data chunk
```

4.2 Zusätzliche Segmente

In der Datei befinden sich möglicherweise zusätzliche Segmente. Einige von ihnen können außerhalb des Anwendungsbereichs dieser Norm liegen. Anwendungen können diese Segmente interpretieren oder benutzen oder auch nicht, so dass die Integrität der in solchen unbekannten Segmenten enthaltenen Daten nicht garantiert werden kann. Konforme Anwendungen sollten jedoch unbekannte Segmente mit deren unveränderten Inhalten weiterleiten.

4.3 Inhalt eines BWFF

Ein BWFF muss den RIFF-“WAVE“-Kopf und zumindest folgende Segmente beinhalten:

```
<WAVE-form>
RIFF('WAVE'
    <fmt-ck>           /* Format of the audio signal: PCM/MPEG */
    <broadcast_audio_extension> /* information on the audio sequence */
    <wave-data> )      /* sound data */
```

4.4 Broadcast Audio Extension-Segment

Zusätzliche Parameter, die für den Austausch von Material zwischen den Sendern notwendig sind, werden in einer speziellen Broadcast Audio Extension, dem bext-Segment, hinzugefügt. Die Struktur des bext-Segments muss wie folgt festgelegt werden:

```
typedef struct chunk_header {
    DWORD ckID;           /* (broadcastextension)ckID=bext */
    DWORD ckSize;         /* size of extension chunk */
    BYTE ckData[ckSize];  /* data of the chunk */
}   CHUNK_HEADER;

typedef struct broadcast_audio_extension {
    CHAR Description[256]; /* ASCII : "Description of the sound
                           sequence" */
    CHAR Originator[32];   /* ASCII : "Name of the originator" */
    CHAR OriginatorReference[32]; /* ASCII : "Reference of the originator"
                                   */
    CHAR OriginationDate[10]; /* ASCII : "yyyy-mm-dd" */
    CHAR OriginationTime[8];  /* ASCII : "hh:mm:ss" */
    DWORD TimeReferenceLow;   /* First sample count since midnight, low
                               word */
```

```

DWORD TimeReferenceHigh;      /* First sample count since midnight,
                               high word */

WORD Version;                 /* Version of the BWF; unsigned binary
                               number. See annex G */

BYTE UMID_0;                  /* Binary byte 0 of SMPTE UMID */

....

BYTE UMID_63;                 /* Binary byte 63 of SMPTE UMID */

INT LoudnessValue;            /* Integrated Loudness Value of the file
                               in LKFS (multiplied by 100) see annex H
                               */

INT LoudnessRange;            /* Loudness Range of the file in LU
                               (multiplied by 100) see annex H */

INT MaxTruePeakLevel;         /* Maximum True Peak Level of the file
                               expressed as dBTP (multiplied by 100) see
                               annex H */

INT MaxMomentaryLoudness;     /* Highest value of the Momentary
                               Loudness Level of the file in LKFS
                               (multiplied by 100) see annex H */

INT MaxShortTermLoudness;     /* Highest value of the Short-Term
                               Loudness Level of the file in LKFS
                               (multiplied by 100) see annex H */

BYTE Reserved[180];           /* 180 bytes, reserved for future use,
                               set to "NULL" */

CHAR CodingHistory[];         /* ASCII : « History coding » */
} BROADCAST_EXT

```

Der Inhalt der Felder im bext-Segment muss wie in Tabelle 1 festgelegt werden. Zu beachten ist, dass in Anwendungen, für die ASCII-Text für visuell lesbare Informationen ungeeignet ist – beispielsweise, wenn ein anderer Zeichensatz als ISO 646 notwendig ist – diese auf andere Weise zu übertragen sind, wie beispielsweise in einem hierfür vorgesehenen Metadaten-Segment, das dem BWFF angefügt wird. Siehe auch Anhang K, Universal Broadcast Audio Extension-Segment (ubxt).

Bis auf Description, Originator, OriginatorReference und CodingHistory sollten alle Positionen denselben Inhalt wie jede der entsprechenden Positionen des gegebenenfalls vorhandenen ubxt-Segments aufweisen (siehe Anhang K). Wenn maschinenlesbare Daten im 'bext'-Segment aktualisiert werden, sollten die entsprechenden maschinenlesbaren Daten im 'ubxt'-Segment ebenfalls in identischer Weise aktualisiert werden.

Tabelle 1 – Feldinhaltsdefinitionen für 'bext' (1 von 3)

Description	visuell lesbar	<p>ASCII-Zeichenkette, 256 Zeichen oder weniger, beinhaltet eine Beschreibung der Sequenz. Wenn Zeilenumbrüche verwendet werden, müssen die Zeilen mit <CR><LF> beendet werden. Wenn keine Daten verfügbar sind oder wenn die Länge der Zeichenkette weniger als 256 Zeichen beträgt, muss das erste nicht benutzte Zeichen ein Füllzeichen sein (00₁₆).</p> <p>Um Anwendungen zu unterstützen, die nur eine kurze Beschreibung anzeigen, sollten die ersten 64 Zeichen eine Zusammenfassung der Beschreibung beinhalten. Die letzten 192 Zeichen dürfen für Einzelheiten verwendet werden.</p>
-------------	-------------------	--

Tabelle 1 (2 von 3)

Originator	visuell lesbar	ASCII-Zeichenkette, 32 Zeichen oder weniger, beinhaltet den Namen des Erstellers der Audiodatei. Wenn keine Daten verfügbar sind oder wenn die Länge der Zeichenkette weniger als 32 Zeichen beträgt, muss das erste nicht benutzte Zeichen ein Füllzeichen sein (00 ₁₆).
OriginatorReference	visuell lesbar	ASCII-Zeichenkette, 32 Zeichen oder weniger, beinhaltet eine Referenz, die von der erstellenden Organisation zugewiesen wurde. Siehe Anhang I. Wenn keine Daten verfügbar sind oder wenn die Länge der Zeichenkette weniger als 32 Zeichen beträgt, muss das erste nicht benutzte Zeichen ein Füllzeichen sein (00 ₁₆).
OriginationDate	visuell lesbar	ASCII-Zeichenkette, 10 Zeichen, beinhaltet das Datum der Erstellung der Audiosequenz. Format: <i>yyyy-mm-dd</i> <i>yyyy</i> = 4 Zeichen für das Jahr müssen einen Wert zwischen 0000 und 9999 beinhalten – = 1 Zeichen <i>mm</i> = 2 Zeichen für den Monat müssen einen Wert zwischen 01 und 12 beinhalten – = 1 Zeichen <i>dd</i> = 2 Zeichen für den Tag müssen einen Wert zwischen 01 und 31 beinhalten Alle Komponenten müssen vorhanden sein. Bindestrich-Zeichen, “–”, müssen in Übereinstimmung mit ISO 8601 als Trennzeichen in einem Datumsausdruck verwendet werden. Zur Kompatibilität mit alternativen Umsetzungen sollte das Wiedergabegerät auch die folgenden Trennzeichen erkennen: “_” Unterstrich “:” Doppelpunkt “ ” Leerzeichen “.” Punkt.
OriginationTime	visuell lesbar	ASCII-Zeichenkette, 8 Zeichen, beinhaltet den Zeitpunkt der Erstellung der Audiosequenz in Stunden, Minuten und Sekunden. Wenn keine Daten verfügbar sind, muss der Standardwert 00:00:00 sein. Format: <i>hh:mm:ss</i> <i>hh</i> = 2 Zeichen für Stunden müssen einen Wert zwischen 00 und 23 beinhalten, wenn eine Zeit angegeben ist : = 1 Zeichen <i>mm</i> = 2 Zeichen für Minuten müssen einen Wert zwischen 00 - 59 beinhalten, wenn eine Zeit angegeben ist : = 1 Zeichen <i>ss</i> = 2 Zeichen für Sekunden müssen einen Wert zwischen 00 und 59 beinhalten Alle Komponenten müssen vorhanden sein. Doppelpunkt-Zeichen, “:”, müssen in Übereinstimmung mit ISO 8601 als Trennzeichen in einem Uhrzeitausdruck verwendet werden. Zur Kompatibilität mit alternativen Umsetzungen sollte das Wiedergabegerät auch die folgenden Zeichen erkennen: “_” Unterstrich “–” Bindestrich “ ” Leerzeichen “.” Punkt.

Tabelle 1 (3 von 3)

TimeReference	maschine nlesbar	Dieses Feld muss die Abtastungsadresszählung [time code] der Sequenz beinhalten. Dies ist ein vorzeichenloser 64-Bit-Wert, der die Abtastungszählung seit Mitternacht der ersten Abtastung in den Audiodaten beinhaltet. Die Anzahl der Abtastungen pro Sekunde hängt von der Abtastfrequenz ab, die im Feld <code><nSamplesPerSec></code> ab <code><fmt-ck></code> festgelegt ist. Der Standardwert ist Null und entspricht Mitternacht.
Version	maschine nlesbar	Eine vorzeichenlose Binärzahl zur Angabe der Version des BWF. Für Version 1 muss diese auf 0001_{16} und für Version 2 muss sie auf 0002_{16} gesetzt werden. Diese ist auf 0002_{16} gesetzt. Siehe Anhang G
UMID	maschine nlesbar	64 Byte, die eine Extended UMID nach SMPTE ST 330 beinhalten. Wenn eine 32-Byte Basic UMID verwendet wird, müssen die letzten 32 Byte mit Nullen gefüllt werden. Wenn keine UMID verfügbar ist, müssen die 64 Byte mit Nullen gefüllt werden. ANMERKUNG Die Länge der UMID ist am Kopf der UMID selbst codiert
LoudnessValue	maschine nlesbar	Der integrierte Lautheitswert der Datei in LKFS (multipliziert mit 100). Ein 16-Bit-Integer mit Vorzeichen, bildet den Integer von $(100 \times \text{Wert der integrierten Lautheit der Datei in LKFS}) \pm 0,5$. Siehe Anhang H zu weiteren Einzelheiten des Umrechnungsverfahrens.
LoudnessRange	maschine nlesbar	Ein 16-Bit-integer mit Vorzeichen, repräsentiert den Lautheitsbereich der Datei in LU. Siehe Anhang H.
MaxTruePeakLevel	maschine nlesbar	Ein 16-Bit-Integer mit Vorzeichen, bildet den Integer von $(100 \times \text{Wert des größten echten Spitzenpegels der Datei in dBTP}) \pm 0,5$. Siehe Anhang H zu weiteren Einzelheiten des Umrechnungsverfahrens.
MaxMomentaryLoudness	maschine nlesbar	Ein 16-Bit-Integer mit Vorzeichen, bildet den Integer von $(100 \times \text{den höchsten Wert des augenblicklichen Lautheitspegels der Datei in LKFS}) \pm 0,5$. Siehe Anhang H zu weiteren Einzelheiten des Umrechnungsverfahrens.
MaxShortTermLoudness	maschine nlesbar	Ein 16-Bit-Integer mit Vorzeichen, bildet den Integer von $(100 \times \text{höchster Wert des Kurzzeit-Lautheitspegels der Datei in LKFS}) \pm 0,5$. Siehe Anhang H zu weiteren Einzelheiten des Umrechnungsverfahrens.
Reserved	maschine nlesbar	180 Byte sind zur Erweiterung reserviert. Diese 180 Byte müssen auf Null gesetzt sein.
CodingHistory	visuell lesbar	Ein Block variabler Größe aus ASCII-Zeichen mit 0 oder mehr Zeichenketten, von denen jede durch <code><CR><LF></code> abgeschlossen wird. Das erste unbenutzte Zeichen muss ein Füllzeichen sein (00_{16}). Jede Zeichenkette muss eine Beschreibung eines für die Audiodaten angewendeten Codierungsprozesses beinhalten. Jede neue Codierungsanwendung sollte eine neue Zeichenkette mit den entsprechenden Informationen anfügen. Siehe Anhang J.

Anhang A (normativ)

RIFF-WAVE-Dateiformat

A.1 Allgemeines

Die Informationen in diesem Anhang folgen den Spezifikationsdokumenten des Microsoft RIFF-Dateiformats. Siehe Literaturhinweise.

A.2 Resource Interchange File Format (RIFF)

Das Resource Interchange File Format (RIFF) ist eine mit Tags markierte Dateistruktur, die zur Verwendung auf Multimediaplattformen entwickelt wurde. In diesem Abschnitt wird ein "Chunk" und die "RIFF Form" beschrieben.

A.2.1 Segment (en: chunk; ck)

Der Basisblock einer RIFF-Datei wird als ein Segment bezeichnet. Mittels der C Syntax kann ein solches Segment wie folgt definiert werden:

```
typedef    unsigned long  DWORD;
typedef    unsigned char  BYTE;
typedef    DWORD FOURCC;      /* Four-character code */
typedef    FOURCC CKID;      /* Four-character-code chunk identifier */
typedef    DWORD CKSIZE;     /* 32-bit unsigned size value */
typedef    struct {          /* Chunk structure */
    CKID    ckID;            /* Chunk type identifier */
    CKSIZE  ckSize;          /* Chunk size field (size of ckData) */
    BYTE    ckData[ckSize]  /* Chunk data */
};
} CK;
```

Ein "FOURCC" wird dargestellt als eine Sequenz von ein bis vier alphanumerischen ASCII-Zeichen, die bei Bedarf auf der rechten Seite mit Leerzeichen (ASCII-Zeichenwert 32), ohne eingebettete Leerzeichen, aufgefüllt werden.

Beispielsweise wird der Vier-Zeichen-Code (FOURCC) "FOO" als eine Sequenz von vier Byte: "F", "O", "O", " " in aufsteigenden Adressen gespeichert. Für rasche Vergleiche darf ein Vier-Zeichen-Code auch als eine 32-Bit-Zahl behandelt werden.

Die drei Teile des Segments sind in der nachfolgenden Tabelle A.1 dargestellt.

Tabelle A.1 – Segmentbeschreibung

Teil	Beschreibung
ckID	Ein Vier-Zeichen-Code, der die Darstellung der Daten von 'ckData' identifiziert. Ein Programm, das eine RIFF-Datei liest, kann jedes Segment überspringen, dessen Identifikationskennung es nicht erkennt; es überspringt einfach die Anzahl an Byte, die von ckSize festgelegt sind und das gegebenenfalls vorhandene Füllbyte.
ckSize	Ein vorzeichenloser 32-Bit-Wert, der die Größe von ckData identifiziert. Dieser Größenwert beinhaltet nicht die Größe der Felder von ckID oder ckSize oder das Füllbyte am Ende von ckData.
ckData	Binäre Daten fester oder variabler Größe. Der Beginn von ckData ist in Bezug auf den Beginn der RIFF-Datei wortausgerichtet. Wenn die Segmentgröße eine ungerade Anzahl von Bytes ist, wird auf ckData folgend ein Füllbyte mit dem Wert Null geschrieben. Der Wert von ckSize beinhaltet nicht das Füllbyte.

Darstellbar ist ein Segment mit folgender Notation (in diesem Beispiel sind ckSize und das Füllbyte impliziert):

`<ckID> ( <ckData> )`

RIFF-Segment, enthält möglicherweise ineinander verschachtelte Segmente oder Untersegmente (en: subchunks).

Alle anderen Arten von Segmenten speichern ein einzelnes Element der Binärdaten in <ckData>.

A.2.2 RIFF-Formen

Eine RIFF-Form ist ein Segment mit einer "RIFF"-Segment-Identifikationskennung. Dieser Begriff bezieht sich auch auf ein Dateiformat, das dem RIFF-Grundaufbau folgt. Ein "WAVE" ist für einen "RIFF Form Type" registriert.

Unter Verwendung der Notation zur Darstellung eines Segments, sieht eine RIFF-Form wie folgt aus:

`RIFF ( <formType> <ck>... )`

Die ersten vier Byte einer RIFF-Form bilden die Segment-Identifikationskennung mit den Werten "R", "I", "F", "F". Das Feld ckSize ist notwendig, zur Vereinfachung wird es jedoch in der Notation ausgelassen.

Das erste DWORD der Segmentdaten im "RIFF"-Segment (oben als <formType> gezeigt) ist ein Vier-Zeichen-Code zur Identifizierung der Datendarstellung oder des *Form-Typs* der Datei. Dem Form-Typ-Code folgt eine Reihe von Untersegmenten. Welche Untersegmente vorhanden sind, hängt vom Form-Typ ab.

Die Definition einer bestimmten RIFF-Form beinhaltet gewöhnlich folgendes:

- Einen eindeutigen Vier-Zeichen-Code zur Identifizierung des Form-Typs
- Eine Liste der erforderlichen Segmente
- Eine Liste der optionalen Segmente
- Möglicherweise eine erforderliche Reihenfolge für die Segmente

A.3 Waveform Audio File Format (WAVE)

Die WAVE-Form ist wie folgt festgelegt. Wie bei allen RIFF-Formen, müssen Programme damit rechnen, auf unbekannte Segmente zu treffen (und diese ignorieren). Dabei muss <fmt-ck> immer vor <wave-data> auftreten und beide Segmente sind in einer WAVE-Datei erforderlich.

```

<WAVE-form> ->
    RIFF ( "WAVE"
        <fmt-ck>                                /* Format chunk */
        [<fact-ck>]                             /* Fact chunk */
        [cue-ck]                                /* Cue points */
        [<playlist-ck>]                         /* playlist */
        [<assoc-data-list>]                     /* Associated data list */
        <wave-data> )                          /* Wave data */

```

Die WAVE-Segmente sind in den nachfolgenden Abschnitten beschrieben.

ANMERKUNG In dieser Spezifikation werden <fact-ck>, <cue-ck>, <playlist-ck> und <assoc-data-list> nicht benutzt.

A.3.1 WAVE-Formatsegment

Das WAVE-Formatsegment <fmt-ck> legt das Datenformat für <wave-data> fest.

Das Segment <fmt-ck> muss wie folgt definiert werden:

```

<fmt-ck> ->    fmt( <common-fields>
                  <format-specific-fields> )

<common-fields> ->
    struct{
        WORD wFormatTag;                /* Format category */
        WORD nChannels;                 /* Number of channels */
        DWORD nSamplesPerSec;           /* Sampling frequency */
        DWORD nAvgBytesPerSec;          /* For buffer estimation */
        WORD nBlockAlign;               /* Data block size */
    }

```

Der Inhalt der Felder im Teil <common-fields> des Segments muss wie in Tabelle A.2 festgelegt werden.

Tabelle A.2 – Formatsegment – common fields

Feld	Beschreibung
wFormatTag	Eine Zahl zur Angabe der WAVE-Formatkategorie der Datei. Der Inhalt des Teils <code><format-specific-fields></code> des Formatsegments und die Interpretation der Wellenform-Daten hängen von diesem Wert ab.
nchannels	Die Anzahl an Kanälen, die in den Wellenform-Daten repräsentiert sind, wie beispielsweise 1 für Mono oder 2 für Stereo.
nSamplesPerSec	Die Abtastfrequenz als Abtastungen pro Sekunde, mit der jeder Kanal wiedergegeben werden sollte.
nAvgBytesPerSec	Die durchschnittliche Anzahl an Byte pro Sekunde, mit der die Wellenform-Daten übertragen werden sollten. Wiedergabesoftware kann die Puffergröße mittels dieses Werts schätzen.
nBlockAlign	Die Blockausrichtung in Byte der Wellenform-Daten. Wiedergabesoftware benötigt zum Bearbeiten ein Mehrfaches von <code><nBlockAlign></code> Byte an Daten gleichzeitig, daher kann der Wert von <code><nBlockAlign></code> zur Pufferausrichtung verwendet werden.

Die `<format-specific-fields>` müssen Null oder mehr Byte an Parametern umfassen. Welche Parameter auftreten, hängt von der WAVE-Formatkategorie ab. Wiedergabesoftware sollte alle unbekannten `<format-specific-fields>` Parameter erlauben (und ignorieren), die am Ende dieses Felds auftreten.

A.3.2 WAVE-Formatkategorien

Die Formatkategorie einer WAVE-Datei muss über den Wert des Felds `<wFormatTag>` des Formatsegments festgelegt werden. Die Darstellung der Daten in `<wave-data>` und der Inhalt der `<format-specific-fields>` des Formatsegments muss von der Formatkategorie abhängen.

Die gegenwärtig festgelegten offenen, nicht proprietären WAVE-Formatkategorien sind in Tabelle A.3 aufgeführt.

Tabelle A.3 – WAVE-Formatkategorien

wFormatTag	Wert	Formatkategorie
WAVE_FORMAT_PCM	0001 ₁₆	Format der Pulsmodulation (PCM)

ANMERKUNG Obwohl andere WAVE-Formate existieren, werden derzeit nur die oben aufgeführten Formate mit dem BWFF verwendet. Einzelheiten zum PCM WAVE-Format sind in A.2 angegeben. Allgemeine Informationen zu anderen WAVE-Formaten sind in Anhang F aufgeführt.

A.3.2.1 PCM-Format

Wenn das Feld `<wFormatTag>` des `<fmt-ck>` auf `WAVE_FORMAT_PCM` gesetzt ist, müssen die Wellenform-Daten aus Abtastungen bestehen, die im PCM-Format dargestellt werden. Für die PCM Wellenform-Daten müssen die `<format-specific-fields>` wie folgt festgelegt werden:

`<PCM-format-specific> ->`

```
struct{
    WORD nBitsPerSample;    /* Sample size */
}
```

Das Feld <nBitsPerSample> muss die Anzahl an Datenbits festlegen, die zur Darstellung jeder Audio-Abtastung pro Kanal verwendet werden. Wenn mehrere Kanäle vorhanden sind, muss die Größe der Abtastung für jeden Kanal gleich sein.

Das Feld <nBlockAlign> sollte gleich der folgenden Formel sein, gerundet auf die nächste ganze Zahl:

$$nChannels \times BytesPerSample$$

Der Wert von BytesPerSample muss durch Aufrunden von nBitsPerSample auf das nächste ganze Byte berechnet werden. Ist das Audio-Datenwort kleiner als eine ganze Zahl von Byte, müssen die höchstwertigen Bits der Audio-Abtastung in den höchstwertigen Bits des Datenworts platziert werden; die nicht verwendeten Daten-Bits neben den niedrigstwertigen Bits müssen auf Null gesetzt werden.

Bei PCM-Daten muss das Feld <nAvgBytesPerSec> des Formatsegments gleich der folgenden Formel sein:

$$nSamplesPerSec \times nBlockAlign$$

ANMERKUNG: Die ursprüngliche WAVE-Spezifikation erlaubt beispielsweise, dass 20-Bit Abtastungen von 2 Kanälen in 5 Byte gepackt werden – die sich ein einziges Byte für die niedrigstwertigen Bits der zwei Kanäle teilen. Das vorliegende Dokument legt eine ganze Zahl von Byte je Audio-Abtastung fest, um die Mehrdeutigkeit der Ausführungen zu reduzieren und größte Austauschkompatibilität zu erreichen.

A.3.2.2 Packen von Daten für PCM WAVE-Dateien

In einer einkanaligen WAVE-Datei werden die Abtastungen aufeinander folgend gespeichert.

Bei Stereo-WAVE-Dateien muss Kanal 0 den linken Kanal und Kanal 1 den rechten Kanal darstellen. Die Tabellen A.4a und A.4b zeigen Beispiele für das Packen von Daten für 16-Bit-Mono- und 16-Bit-Stereo-WAVE-Dateien:

Tabelle A.4a – Packen von Daten für 16-Bit-Mono-PCM

Abtastung 1		Abtastung 2	
Kanal 0 niederwertiges Byte	Kanal 0 höherwertiges Byte	Kanal 0 niederwertiges Byte	Kanal 0 höherwertiges Byte

Tabelle A.4b – Packen von Daten für 16-Bit-Stereo-PCM

Abtastung 1			
Kanal 0 (links) niederwertiges Byte	Kanal 0 (links) höherwertiges Byte	Kanal 1 (rechts) niederwertiges Byte	Kanal 1 (rechts) höherwertiges Byte

In mehrkanaligen WAVE-Dateien müssen die Abtastungen in der Kanalreihenfolge verschachtelt werden (siehe Anhang E für Beispiele).

A.3.2.3 Datenformat der Abtastungen

Jede Abtastung ist in einem Integer i enthalten. Die Größe von i ist die kleinste Anzahl an Byte, die erforderlich ist, um die festgelegte Abtastgröße zu beinhalten. Das niedrigstwertige Byte muss als erstes gespeichert werden. Die Bits, die die Abtastamplitude darstellen, sind in den höchstwertigen Bits von i gespeichert und die verbleibenden Bits müssen auf Null gesetzt werden.

Wenn beispielsweise die in `nBitsPerSample` aufgezeichnete Abtastgröße 20 Bits beträgt, wird jede Abtastung in einem Integer von 3 Byte gespeichert. Die niedrigstwertigen vier Bits des ersten (niedrigstwertigen) Bytes werden auf Null gesetzt. Das Datenformat und die höchsten und niedrigsten Werte für die PCM-Wellenform-Abtastungen verschiedener Größen sind in Tabelle A.5 dargestellt. Ein entsprechendes Beispiel für 16-Bit-PCM-Datenwerte ist in Tabelle A.6 dargestellt.

Tabelle A.5 – PCM-Datenformat

Abtastgröße	Datenformat	Höchstwert	Niedrigstwert
>8 Bits	Integer i mit Vorzeichen	Größter positiver Wert von i	Größter negativer Wert von i

Tabelle A.6 – PCM-Datenformat – 16-Bit

Format	Höchstwert	Niedrigstwert	Mittelwert
16-Bit PCM	32767 (7FFF16)	-32768 (800016)	0

ANMERKUNG Ältere Wortgrößen von 8 Bits und kleiner verwenden üblicherweise ein anderes Schema

A.3.2.4 Beispiele für PCM WAVE-Dateien

Beispiele der Einstellungen des Format-Segments für vier Fälle sind in Tabelle A.7 dargestellt. Die Fälle sind:

- A** PCM WAVE-Datei mit Abtastrate 48 kHz, Mono, 16 Bits pro Abtastung
- B** PCM WAVE-Datei mit Abtastrate 44,1 kHz, Stereo, 16 Bits pro Abtastung
- C** PCM WAVE-Datei mit Abtastrate 96 kHz, Stereo, 24 Bits pro Abtastung
- D** PCM WAVE-Datei mit Abtastrate 44,1 kHz, Stereo, 20 Bits pro Abtastung

Tabelle A.7 – Beispiele für das PCM WAVE-Format-Segment

	Feld	A	B	C	D
Common fields	<code>wFormatTag</code>	1	1	1	1
	<code>nchannels</code>	1	2	2	2
	<code>nSamplesPerSec</code>	48000	44100	96000	48000
	<code>nAvgBytesPerSec</code>	96000	176400	576000	288000
	<code>nBlockAlign</code>	2	4	6	6
PCM-format-specific	<code>nBitsPerSample</code>	16	16	24	20

A.4 Speicherung der WAVE-Daten

Das Segment `<wave-data>` enthält die Wellenform-Daten als Little-Endian Ganzzahlen. Es muss wie folgt definiert werden:

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

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

Anhang B (normativ)

Segmentreihenfolge

Nach den RIFF-Regeln dürfen Segmente in beliebiger Reihenfolge in der Datei auftreten. Im spezifischen Fall einer WAVE-Datei, muss <fmt-ck> stets vor <data-ck> stehen. Andere Segmente können in beliebiger Reihenfolge stehen. Beispielsweise ist es möglicherweise für einige Anwendungen zweckmäßig, die Metadaten-Segmente nach statt vor dem Audiodaten-Segment zu schreiben.

Obwohl eine bestimmte Anwendung interne Konventionen in Bezug auf die Reihenfolge aufweisen darf, in der sie erwartet, Segmente zu schreiben, sollte sie dennoch in der Lage sein, Dateien anderer Systeme mit Segmenten in beliebiger anderer Reihenfolge wiederzugeben.

ANMERKUNG 1 Das erweiterte Format in Anhang F erfordert als besondere Ausnahme als erstes Segment in der Datei das "ds64"-Segment oder dessen Platzhalter, das "JUNK"-Segment.

ANMERKUNG 2 Das "JUNK"-Segment ist für eine konforme BWF-Datei nicht erforderlich, würde aber durch Anwendungen einbezogen, die eine Dateigröße von über 4 GByte aufnehmen möchten.

Anhang C (normativ)

Dateinamenskonventionen

C.1 Allgemeines

Der allgemeine Austausch von Audiodateien bedeutet, dass diese auf Computer- und Betriebssystemarten spielbar sein müssen, die sich stark vom Ursprungssystem unterscheiden können. Ein ungeeigneter Dateiname könnte bedeuten, dass die Datei nicht vom Zielsystem erkannt werden kann. Beispielsweise begrenzen einige Computerbetriebssysteme die Zeichenanzahl in einem Dateinamen. Andere sind nicht in der Lage, Mehrbyte-Zeichen aufzunehmen. Einige Zeichen haben eine besondere Bedeutung in bestimmten Betriebssystemen und sollten vermieden werden.

Diese Richtlinien sind dazu gedacht, bewährte Vorgehensweisen für den allgemeinen internationalen Austausch zu ermitteln.

C.2 Dateinamenlänge

Die Namen von BWFF-Dateien sollten 31 Zeichen, einschließlich der Dateinamenserweiterung, nicht überschreiten.

C.3 Dateinamenerweiterung

BWF-Dateien müssen mit ".wav" dieselbe Vier-Zeichen-Erweiterung des Dateinamens verwenden, wie eine herkömmliche WAVE-Datei. Dies ermöglicht das Abspielen des Audioinhalts auf den meisten Computern ohne zusätzliche Software. Praktische Ausführungen sollten auch andere Erweiterungen akzeptieren, wie beispielsweise ".bwf", die möglicherweise irrtümlich verwendet wurden.

C.4 Zeichensatz für Dateinamen

Dateinamen für den internationalen Austausch sollten nur ASCII (ISO/IEC 646) 7-Bit-Zeichen im Bereich von 32 bis 126 (dezimal) verwenden.

Zeichen	Dezimalwert	Hexadezimalwert
(Leerzeichen)	32	20 ₁₆
...	...	...
~ (Tilde)	126	7E ₁₆

Darüber hinaus sind folgende Zeichen für Sonderfunktionen bei bestimmten Dateisystemen reserviert und sollten nicht in Dateinamen verwendet werden:

Zeichen	Dezimalwert	Hexadezimalwert
"	34	22 ₁₆
*	42	2A ₁₆
/	47	2F ₁₆
:	58	3A ₁₆
<	60	3C ₁₆

Zeichen	Dezimalwert	Hexadezimalwert
>	62	3E ₁₆
?	63	3F ₁₆
\	92	5C ₁₆
	124	7C ₁₆

Zusätzlich sollten die folgenden Zeichen nicht für das erste oder letzte Zeichen in einem Dateinamen verwendet werden:

Zeichen	Dezimalwert	Hexadezimalwert
(Leerzeichen)	32	20 ₁₆
. (Punkt)	46	2E ₁₆

Anhang D (informativ)

Mehrkanal-Verwendung

D.1 Allgemeines

Bearbeitungssysteme benötigen üblicherweise Zugriff auf Mono-Dateien, um diese flexibel im Wechsel bearbeiten zu können. Einige Feldrecorder arbeiten allerdings effizienter, wenn sie mehrere Kanäle in einer einzigen Mehrkanal-Datei aufzeichnen. Daher sind viele Bearbeitungsanwendungen in der Lage, Aufzeichnungen von einer Mehrkanal-Quelle in einen kohärenten Satz von Mono-Dateien umwandeln.

D.2 Packen von Mehrkanal-Audiodaten

Dateien mit mehreren Audiokanälen werden mit Hilfe einer einfachen Erweiterung des bestehenden Formats aufgebaut. Wie in den folgenden Beispielen dargestellt, ist das Kanalfeld in <fmt-ck> auf die Anzahl von Kanälen gesetzt und die Audiodaten sind sequentiell gepackt:

Abtastung 1			Abtastung 2		
Niederwertiges Byte	Mittelwertiges Byte	Höherwertiges Byte	Niederwertiges Byte	Mittelwertiges Byte	Hochwertiges Byte

Bild D.1 – Packen von Daten für 24-Bit-Mono PCM Audiodaten

Abtastung 1				Abtastung 2			
Kanal 1 (links)		Kanal 2 (rechts)		Kanal 1 (links)		Kanal 2 (rechts)	
Niederwertiges Byte	Höherwertiges Byte	Niederwertiges Byte	Höherwertiges Byte	Niederwertiges Byte	Höherwertiges Byte	Niederwertiges Byte	Höherwertiges Byte

Bild D.2 – Packen von Daten für 16-Bit-Stereo (2-Kanal) PCM-Audiodaten

Abtastung 1								Abtastung 2							
Kanal 1		Kanal 2		Kanal 3		Kanal 4		Kanal 1		Kanal 2		Kanal 3		Kanal 4	
Niederwertiges Byte	Mittelwertiges Byte	Höherwertiges Byte	Niederwertiges Byte	Niederwertiges Byte	Mittelwertiges Byte	Höherwertiges Byte	Niederwertiges Byte	Niederwertiges Byte	Mittelwertiges Byte	Höherwertiges Byte	Niederwertiges Byte	Niederwertiges Byte	Mittelwertiges Byte	Höherwertiges Byte	Niederwertiges Byte

Bild D.3 – Packen von Daten für 24-Bit 4-Kanal PCM-Audiodaten

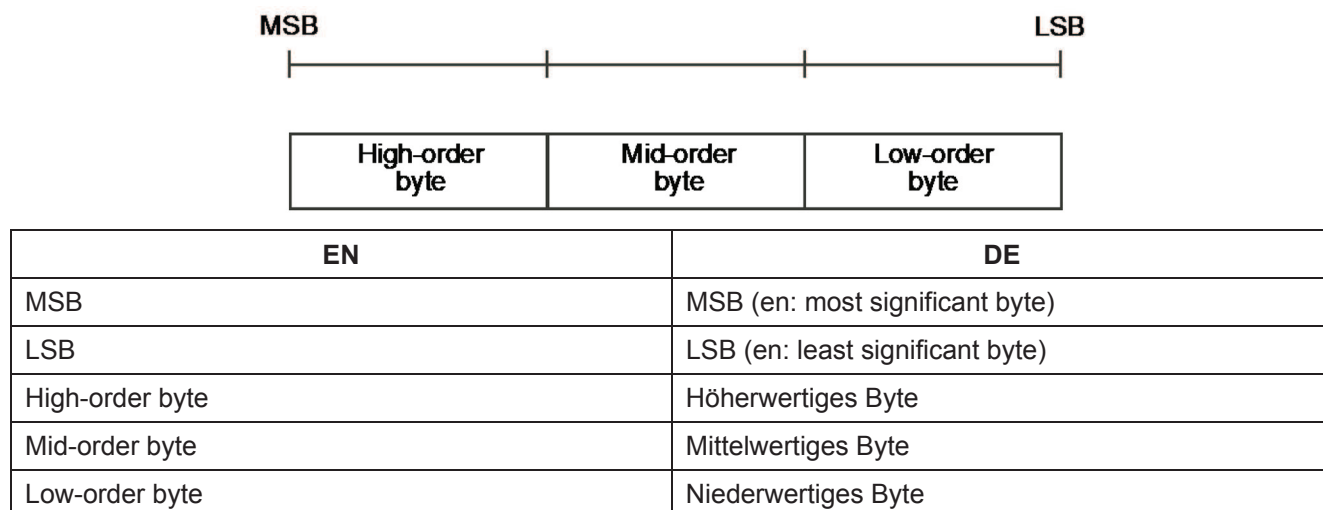


Bild D.4 – Packen einer 4-Bit-Abtastung

D.3 Kanalzuordnungen in Mehrkanal-Dateien

Die Kanalverwendung bleibt in dieser Norm unberücksichtigt. Es gibt zwei Hauptanwendungsbereiche.

D.3.1 Verbreitung und Archivierung

Stereo- oder Raumklangaufzeichnungen zur Archivierung oder Verbreitung von Zwischenelementen oder fertigen Mastern können die Audiokanäle nach einem aus einer Reihe vorbestimmter Schemata verwenden. Beispiele:

Rec. ITU-R BR.1384-2; Parameters for international exchange of multi-channel sound recordings with or without accompanying picture

EBU Technical Recommendation R91-2004; Track allocations and recording levels for the exchange of multichannel audio signals

SMPTE ST 2035-2009; Audio Channel Assignments for Digital Television Recorders (DTRs)

SMPTE ST 2035-2009 Amendment 1:2012; Audio Channel Assignments for Digital Television Recorders (DTRs) – Amendment 1

SMPTE ST 323-2004; Channel Assignments and Levels on Multichannel Audio Media (Film)

D.3.2 Produktionsaufzeichnungen

Produktionsaufzeichnungen können die Kanäle auf vielen verschiedenen Wegen einsetzen, die in einem herkömmlichen Aufzeichnungsbericht oder in einer geeigneten Form von zur Tonaufzeichnung zugehörigen Metadaten beschrieben sind. Der Satz an Audiokanälen kann zugehörige Aufzeichnungen in Mono, LR-Stereo, MS-Stereo oder Raumklangsätzen ermöglichen.

Anhang E (informativ)

Andere Audiocodierungen

E.1 Allgemeines

Alle Typen, die nicht PCM WAVE entsprechen, beinhalten ein Fact-Segment <fact-ck> und eine erweiterte WAVE-Formatbeschreibung im Format-Segment <fmt-ck>.

E.2 MPEG-Dateien

Einzelheiten des MPEG WAVE-Formats sind aufgeführt in ITU-R BR.1352-3 Annex 2, und Appendix 1 zu Annex 2, sowie in EBU T3285 – Supplement 1; Specification of the Broadcast Wave Format with MPEG-1 audio, A format for audio data files in broadcasting, siehe die Literaturhinweise.