The Broadcast Wave Format
– an introduction

R. Chalmers
(EBU Technical Department)

This article provides a brief introduction to the new Broadcast Wave Format (BWF) file which has been developed by the EBU – in close collaboration with the audio industry – to facilitate the interchange of programme material between audio workstations.

Another article by Lars Jonsson [1] describes the use of BWF files in Swedish Radio.

1. Introduction

The Broadcast Wave Format has been developed by EBU Project Group P/DAPA (Digital Audio Production and Archiving), in close co-operation with the audio industry. It aims to provide a means of exchanging programme material between audio workstations in the form of special files. This has proved difficult in the past because of:

⇨ differences in the native file formats of different computer platforms;

⇨ the different operating systems chosen by different manufacturers;

⇨ the amount of metadata (data about the data) demanded to accompany the audio data.
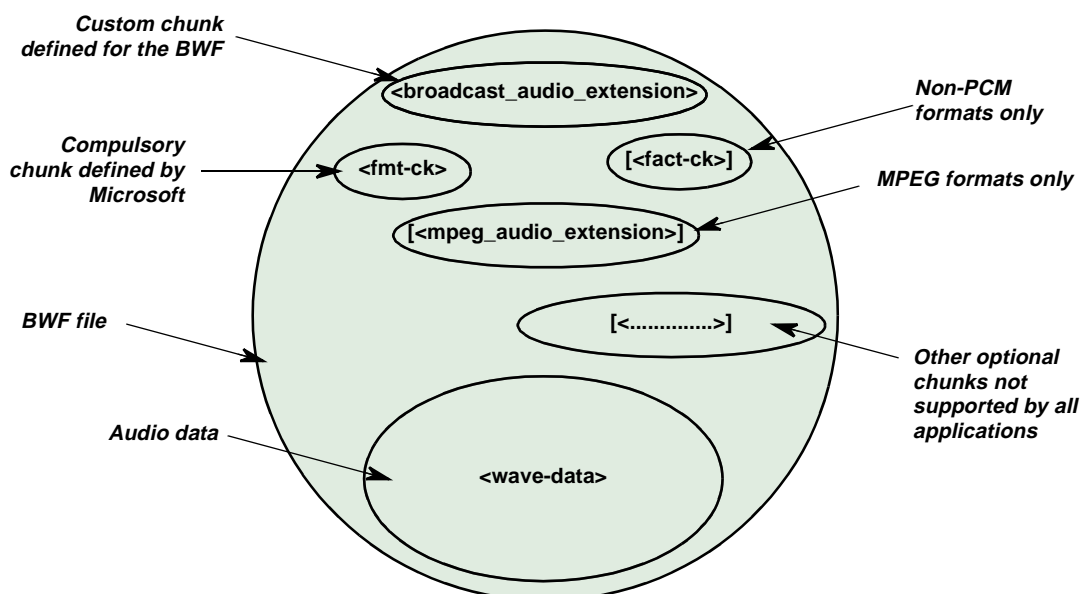
As far as possible, given that a choice had to be made between incompatible technologies, the BWF has been designed to be neutral and not to favour any particular equipment. It can be used as a native format on equipment and this implementation is highly recommended. Indeed, at the present time, some 25 manufacturers have adopted this approach. However, its use as a native format on equipment is not compulsory: the minimum requirement sought by the broadcasters is simply that the native file format used on any particular equipment can be converted into BWF files for the exchange of programme material. This could be done "on the fly", while the file was being read from the source machine and stored on the destination machine.

2. The next stages

2.1. World-wide co-operation

The BWF was originally developed by the EBU from a European perspective, but it has attracted world-wide interest as it presents an opportunity to break the current log-jam in the

Custom chunk
defined for the BWF

<broadcast_audio_extension>

Non-PCM
formats only

Compulsory
chunk defined by
Microsoft

<fmt-ck>

[<fact-ck>]

MPEG formats only

[<mpeg_audio_extension>]

BWF file

[<.............>]

Other optional
chunks not
supported by all
applications

Audio data

<wave-data>

Figure 1
Contents of a Broadcast Wave Format file.

development of a common file format for audio. Recently, the AES working group, SC-06-01, adopted the basic BWF file in principle. This AES working group has also agreed to collaborate with EBU Project Group P/AFT (Audio File Transfer) in the further development of the BWF, thus bringing the AES's vast experience of professional audio into the BWF development programme. This collaboration closely resembles the earlier joint work carried out on the AES/EBU digital audio interface [2], which is now the norm on all professional digital audio equipment.

2.2.   Transport layers

The BWF is a file format, and is compatible with the ISO-OSI layer model for information exchange. It is independent of the transport layer used to connect the equipment, and can be used over networks in real or non-real time, or carried on a suitable physical support such as disc or tape.

The standards and protocols for networks, data discs and data tapes have been – and will continue to be – developed by the information technology industry. The EBU does not expect to be active in developing such systems. However, because the BWF files are large compared to the files used in an office environment and they need a high data-rate for real-time transfer, the EBU expects to recommend suitable networks and data supports for BWF files. When manufacturers have adopted these, the overall aim of the EBU will have been achieved, that is to allow any two items of audio equipment to exchange files without any restrictions.

2.3.   More metadata

The metadata included in the BWF has been deliberately restricted to the minimum needed to exchange the audio data. Much other information can be added to help during the produc-

tion, post-production, broadcasting and archival stages of the life of a programme. However the EBU project group quickly realized that this information was usually highly specific to the application and the equipment. The group therefore only included the information which was expected to be common to all applications. The next stage will be to develop suitable methods of carrying more information, either as part of different types of BWF files or as separate but linked files. The first developments are expected to be a *project chunk*, to exchange information during simple editing, and an *archive chunk*.

3. Some details of the BWF

3.1. Relation of BWF to WAVE

The Broadcast Wave Format is based on the Microsoft WAVE audio file format. The WAVE file is one of a number of types of file specified in the Microsoft *Resource Interchange File Format* (RIFF). A WAVE file is a type of RIFF file that specifically contains audio data.

RIFF files are made up from basic building blocks called "chunks". A chunk is a block of data bytes, usually containing a specific type or types of information. Each chunk also contains an identification field and a size field. This allows the reading software to identify any chunks it is able to process, and to pass over and ignore any chunks that it does not understand.

The BWF uses as much of the basic RIFF WAVE structure as possible. Only one new chunk needs to be added to a *linear PCM* WAVE file to make it into a BWF file. A second new chunk needs to be added to an *MPEG* WAVE file to make a BWF file.

The WAVE format supports a number of different audio formats but not all types of WAVE files can become valid BWF files. At the moment, only linear PCM and MPEG code files are supported in BWF (see *Fig. 1*).

3.2. EBU documents on BWF

The EBU has just published – in document Tech. 3285 [3] – the specification of the BWF and the broadcast audio extension chunk <bext> which is used in all BWF files. In addition, Tech. 3285 gives information on the basic RIFF format and shows how the WAVE and BWF formats can be extended to other types of audio data. Details of the PCM Wave format are given in Appendix A of Tech. 3285. Detailed specifications of how to create BWF files for other specific types of audio data will be published in separate Supplements to Tech. 3285. At present the

Abbreviations

AES	Audio Engineering Society	**LS**	Least significant
ASCII	American Standard Code for Information Interchange	**MPEG**	(ISO) Moving Picture Experts Group
		OSI	Open systems interconnection
BWF	(EBU) broadcast wave format	**PCM**	Pulse code modulation
ISO	International Organization for Standardization	**RIFF**	(Microsoft) resource interchange file format

only other type of audio data recognized is MPEG and this specification is given in Tech. 3285 Supplement 1.

In addition to the specification of the BWF files, the EBU has recommended – in EBU document R85 [4] – the audio content of BWF files when they are used for the exchange of programme material without prior consent. The phrase "without prior consent" means that EBU Members will normally accept such material. For instance, document R85 recommends a sampling frequency of 48 kHz. If the receiving organization is willing to accept material which has been sampled at 44.1 kHz, this can still be supplied as a valid BWF file but only after this has been mutually agreed.

4. Deconstruction of a BWF file

Each BWF file consists of a file header followed by a number of chunks. Some chunks are compulsory and some are optional (see *Fig. 1*). The BWF file is a special class of WAVE file.

All WAVE files contain the header and at least:

⇨ a *format chunk* that contains coded instructions on how to interpret the audio data;

⇨ a *data chunk* that contains the audio data.

If the audio data is not in PCM form, there will also be a *fact chunk* which contains further information on the nature of the audio data.

In theory, the chunks can be in any order but, even in a relatively short audio file, the data chunk is by far the largest. Therefore, especially in a large file, the chunks that contain information on the audio data are more useful if they come before the data chunk.

A BWF file also includes a special *broadcast extension chunk.* This contains a minimum set of fields which store information about the file that is relevant to all broadcasting applications (see *Table 1*).

Field	Content
Description	Short and longer titles
Originator	Name of the originator or producer
Reference	A reference number issued by the originator
Date and time	When the programme was created
Coding history	A log of the signal coding, e.g. linear PCM, MPEG, etc.

Figure 1
Contents of the broadcast extension chunk.

However, it does not include any information needed to interpret the audio data and, thus, a BWF file containing a broadcast programme can be played on any WAVE-compliant application. Obviously, in this case the contents of the broadcast extension chunk will not be read.

Other optional chunks can also be included in WAVE and BWF files. A number of types of chunks have been registered for useful applications. If these are included in a BWF file, the reading software will interpret any that it understands and ignore any others.

How does the software application know what to do?

⇨   All WAVE files start with the header which consists of:

⇨   hex code 52 49 46 46 (ASCII "RIFF");

⇨   a 4-byte length code which gives the total number of bytes contained in the remainder of the file;

⇨   hex code 57 41 56 45 (ASCII "WAVE").

Without these header codes, the reading software will not recognize a WAVE file.

Following the header codes are all the various chunks which make up the file. Each of these chunks consists of:

⇨   a chunk identifier (ID);

⇨   a chunk length;

⇨   the chunk data.

Examples of two chunks are given in *Table 2*.

Broadcast extension chunk			
Chunk identifier (ID)	Hex code 62 65 78 74	ASCII code "bext"	Broadcast extension
Chunk length	Hex code F0 02 00 00	LS byte first	752 bytes
Chunk data	xx xx	752 bytes of data	As per EBU Tech. 3285
Format chunk			
Chunk identifier (ID)	Hex code 66 6D 74 20	ASCII code "fmt"	Format
Chunk length	Hex code 10 00 00 00	LS byte first	16 bytes
Chunk data	xx xx	16 bytes of data	As per RIFF WAVE spec.

Table 2
Examples of the broadcast extension chunk and the format chunk.

If an application reads a chunk ID and length code, but does not recognize the ID, it can at least proceed by the number of bytes in the length code and find the next chunk ID.

5.   Length and capacity of a BWF file

The maximum length of a BWF file is slightly less than 2^{32} bytes, that is 4 GB. For a linear PCM stereo audio file, sampled at 48 kHz with 16-bit resolution, there are 4 bytes per sample period. This gives a maximum capacity of 10^9 samples or a duration of over 6 hours of audio. This is longer than most recording media normally used by broadcasters (CD, DAT, 6.3 mm

tape, etc.). The use of MPEG compression will extend this duration, depending on the compression factor chosen.

Bibliography

[1]   Jonsson, L.: **The use of BWF files in Swedish Radio**
        EBU Technical Review No. 274, Winter 1997.

[2]   EBU document Tech. 3250 (1992): **Specification of the digital audio interface (the AES / EBU interface).**

[3]   EBU document Tech. 3285 (1997): **Specification of the Broadcast Wave Format: a format for audio data files in broadcasting** (Supplement 1 to this document gives detailed specifications of how to create BWF files for MPEG audio).

[4]   EBU Technical Recommendation R85-1997: **Use of the Broadcast Wave Format for the exchange of audio data files**.