

[HomePage](#) > [Technical](#) > [Reference Data](#)

- [The Union](#)
- [Members](#)
- [President/Presidency](#)
- [Secretary General](#)
- [Radio](#)
- [Television](#)
- [Eurovision Operations](#)
- [Legal](#)
- [Technical](#)
 - [Publications](#)
 - [Reference data](#)
 - [Project Groups](#)
 - [Seminars & Events](#)
 - [Organisation](#)
 - [Related activities](#)
 - [FAQ](#)
 - [Contacts](#)
 - [Members only](#)
- [Multimedia](#)
- [HR Group](#)
- [Intl. Training](#)
- [Calendar](#)
- [Sister Unions](#)
- [EBU Lounge](#)
- [Contact us](#)

Broadcast Wave Format (BWF) user guide

Introduction

The EBU introduced the Broadcast Wave Format in 1996 to allow files to be exchanged between the increasing number of digital audio workstations used in radio and television production. The format was developed in collaboration with the industry and since its introduction it has been implemented in equipment from a large number of manufacturers.

In late 2000, the EBU agreed to make provision for BWF files to include a SMPTE UMID. This development had led to the introduction of version 1 of the specification.

What is the BWF?

The increasing use of computer technology and equipment in broadcasting creates a need for seamless exchange of audio material between different broadcast environments and equipment based on different computer platforms. The EBU has specified a file format which contains the minimum information that is considered necessary for all broadcast applications. The basic information, together with the audio data, is organized as "Broadcast Wave Format", BWF, files. From the Broadcast Wave Format file, using an object-oriented approach, a higher level descriptor can be used to reference other files containing more complex sets of information which can be assembled for the different specialised kinds of application. However for the exchange of audio files between EBU Members without prior consent, certain restrictions are imposed on the audio signals contained in the BWF to simplify the process of exchange.

What is version 1 of the BWF?

Version 1 differs from version 0 only in that 64 of the 254 reserved bytes in version 0 are now used to contain a SMPTE UMID and the reserved and version fields are changed accordingly.

Version 1 is backwards compatible with Version 0. This means that software designed to read version 0 files will interpret Version 1 files correctly, except that it will ignore the UMID field.

The change is also forward compatible. This means that Version 1 software will be able to read Version 0 files correctly. Ideally, Version 1 software should read the version field to determine if a UMID is present. However if the Version number is not read, software will read all zeros in the UMID field in a Version 0 file. This will not be a valid UMID and will be ignored.

Why don't my MPEG BWF files play on my PC?

One reason may be that your audio player may not support MPEG 1 layer II. With the increasing popularity of MP3, some players no longer support other layers of MPEG. If you have recently updated your operating system you may have updated your earlier player. Try downloading an *older* version, for example Windows Media Player 6.4, or installing a different player.

Another reason may be that your player does not recognise the extension "wav" for MPEG files. Some players expect MPEG files to have the file extension "mp2". You may be able to set up your player to accept the extension "wav" for MPEG files. However, this may then prevent it playing linear BWF files! This is of course a fault with the player which should check the file header not the extension.

Is the BWF a unique format, specially designed for broadcasters?

No, the BWF is a development of the existing WAVE format, used on many DAWs and computers. A WAVE file is an audio file which is one type of the more general RIFF (Resource Interchange File Format) file. RIFF has been developed by the IBM and Microsoft corporations.

Where is the BWF specified?

The BWF is specified in EBU document . The specification of the MPEG version is given in . These documents also contain information from the basic specification of WAVE files.

How does a BWF file differ from an ordinary WAVE file?

Briefly, a BWF file is a restricted subset of possible WAVE files but it also contains an extra "chunk" to carry information about the content.

How is a BWF file restricted compared to an ordinary WAVE file?

A BWF file can only contain two specified types of audio data, Linear PCM and MPEG coded. A WAVE file can contain a much wider range, many very specific to some software. BWF files are intended for use in radio and television production. The basic audio format is linear PCM, 16 bits, sampled at 48 kHz, which is the recommended audio format for production. See [EBU Recommendation R68-2000](#). A supplement to the basic specification also allows the files to contain MPEG compressed audio data. At the moment, PCM and MPEG are the only audio formats allowed.

Is a BWF with audio sampled at 44.1 kHz still a proper BWF?

Yes, but you can't expect another organisation to accept it without getting their prior consent. See [EBU Recommendation R68-2000](#).

What is a "chunk"?

A "chunk" is a self contained collection of data in a RIFF file. It contains a header, which gives its type and length, followed by data arranged in fixed or variable length fields.

What is the extra chunk in the BWF?

The "Broadcast extension" chunk, coded "bext", is contained in all BWF files. It contains the minimum information expected to be needed by all applications in broadcast production. It contains information on the title, origination, date, time, etc. of the audio content. A BWF file containing MPEG audio data also includes a further extra chunk "mext".

Should it be BEXT or bext, MEXT or mext?

Fairly late in the development, it was decided that the lower case was correct i.e. <bext> and <mext> not <BEXT> and <MEXT>. It appears that IBM/Microsoft originally intended to use upper case for registered chunks. However it seems that there have been no new chunks registered since 1994, according to the documentation on the Microsoft website.

We advise software developers to use lower case chunk names when creating files but be prepared to read either case.

Is the file extension for a BWF file ".bwf"?

No, even if this is what you might expect. The extension should be ".wav" because this is mandatory for all WAVE files. The BWF is a special WAVE file. If you use any other extension, some computers may not be able to play out the audio. This also implies that any earlier extensions have to be replaced.

If my software cannot interpret the "bext" chunk, will I still be able to play the file?

Yes, if it is a PCM audio file. All the information necessary to interpret the audio data is carried in the "Format" chunk which is part of the basic WAVE file. All WAVE compatible software should be able to interpret this information and play the PCM audio.

Will my WAVE software always play BWF files?

Not necessarily. All WAVE compliant software should always play linear PCM wave files. However, not all WAVE software recognises MPEG compression coding. This applies to ordinary WAVE files as well as to BWF files.

How does my software know that a BWF file contains MPEG audio?

This information is carried in the WAVE "format" and "fact" chunks which are part of the WAVE standard. Additional information for MPEG layer 2 audio is carried in an extra chunk "mext" in MPEG BWF files.

Can I use other types of audio coding in BWF files?

No. Linear PCM and MPEG are the only recognized formats for audio data in BWF files. See the restrictions on BWF files.

How long is a BWF file?

The WAVE specification allocates 32 bits to the file length field. Thus, a BWF file is limited to 4 GBytes total, including all the non-audio chunks. (Note, some software and/or operating systems may, incorrectly, treat the file length field as a signed number. If so, this would unnecessarily reduce the file length to half.)

What is the longest audio sequence that I can put in a BWF file?

This depends on the sample frequency, the number of channels and the resolution (bits per sample) of the audio signal. For example, a stereo (two-channel) file, sampled at 48 kHz and using 16 bits per channel, can contain up to 6 hours of audio.

Can I add extra information or metadata about my programme to a BWF file?

You can add any valid chunk to a BWF file. However, the extra chunks will only be interpreted by applications which are programmed to do so. Other applications will ignore the contents of these chunks. The EBU intends to register a limited number of extra chunks for specific application in broadcasting. The current types are given below.

What information do I put in the "bext" and "mext" chunks, and what form should it have?

The "bext" and the "mext" chunks contain various fields of data. Many of these fields are fully defined in the specification. For example, the *Date* field in the "bext" chunk and the *SoundInformation* field in the "mext" chunk have defined formats. For other fields, such as *OriginatorReference*, the specification only covers the type of data and the length. However the EBU Members have developed recommended formats for the data in some of these fields. See below.

When a BWF file is imported from another software system, how should the contents of the "bext" chunk be treated?

This depends on whether the audio software works with a file structure, or is a system connected to a database. Different designs must be used in each case.

If the receiving system is based on a file structure, but with no database, selected fields from the "bext" chunk of a file incoming from another system should be displayed in a pop-up window.

In an audio software system that works with a database, the audio files are indexed and the metadata contained in the "bext" chunk is stored and retrieved from the database. For an incoming file, the content of the various fields of the "bext" chunk are interpreted and the corresponding fields of the database in the receiving system, are updated accordingly.

How should the metadata from the "bext" chunk be stored in a user's database?

Below are some examples of how manufacturers have inserted the information from the "bext" chunk into the fields of their databases. The examples are mainly taken from network based radio on-air systems with simple two-track editing facilities:

Field	Comments
Description	The 256 characters can be named "Title" or similar in the database. This field holds the working name of the file. For instance "Summitbriefing". This field should not be the name of the file.
Originator	This field can, for instance, be the name of the reporter or the producer of the file, or the artist or orchestra, if it is a music recording. This field can have the name "Reporter", "Producer", "Client" or "Artist" or similar in the database, depending on the most common use for a majority of files in the production area where the database is used.
OriginatorReference	A format for this field is described below. This long string could be kept as a separate Unique ID field in the database not be the name of the file.
OriginationDate	This field should be the date of the creation of the audio file. When the actual original recording is being made, the date is retrieved from the computer's clock and stored in this field.
OriginationTime	This field should be treated similarly to the date. The time is that retrieved from the real-time clock in the computer exactly when the recording begins. This will make it possible to seek files based on date and time-of-day. The accuracy depends on the stability of the real-time clock in the computer used for recording the file. Date and time should be inserted into the corresponding fields in the database. OriginationTime is not necessarily the same information as the "time" field in the file directory.
TimeReference	This field is a count from midnight in samples to the first sample of the audio sequence. This number can be used for time code generation if the audio section of the computer being used for recording the file has a very accurate and time-stable sampling frequency. This feature might be omitted from BWF files that are not used with accompanying video. If used, this number can be transferred to and from the database.
CodingHistory	A format for this field has been agreed but is not compulsory. The strings for each stage in the coding history can be extracted and kept in field(s) of the database. When the next copy of the file is being made, these fields can be retrieved from the database and used to generate the coding history field of the new "bext" chunk.

What actions should be made when saving a BWF file?

It depends on the software concept:

In an audio software system using a database for indexing files and storing the metadata, a database window should appear when a new file is saved. Some of the fields needed in the "bext" chunk can be created automatically by the computer and entered into the chunk. Some of the information can be entered manually via this database window or it may be taken from default values or from an automatic numbering system.

In an audio software system using a file system with no database system involved, a dialogue box should appear in the "save as" window when a new files is saved. This box should allow manual entry of the information into the fields of the "bext" chunk as described above.

The BWF specification suggests splitting the *Description* field in "bext" into two sections.

Can these two sections be saved as two separate strings?

Yes, the short and detailed descriptions can be saved as separate strings.

If the short *description* string is shorter than 64 characters, should this field be padded with spaces or NULLs?

Padding with Nulls is preferred.

The "bext" chunk seems to be open ended but in the sample BWF files, the *CodingHistory* field is padded with Nulls to a length of 256 characters. How should this field be used?

The length of the *CodingHistory* field is given by the overall chunk size (ckSize) field of the "bext" chunk. The coding history field can therefore exceed 256 bytes. The field need only contain as

many bytes as needed with at least one Null at the end. For most files, padding to 256 characters with Nulls is simpler.

Optional chunks

Note: Optional chunks can be added to BWF files for specific uses. These extra chunks are not necessarily interpreted by all BWF applications and, if not, they will be ignored.

Capturing Report or "Quality" chunk

This optional chunk is intended to contain information on the audio quality of the audio content of a BWF. Frequently this data is gathered at the digitisation stage and can remain with the file as a permanent record of the audio quality and any known defects.

It is published as a supplement to the [specification of the BWF](#), see [supplement 2 to EBU Tech 3285](#).

Peak Envelope or "Levels" chunk

This optional chunk is intended to contain data on the peak audio level of the audio contents of the BWF file. It contains a series of values giving the maximum levels in each of the blocks of the signal. It also contains the location of the maximum of the peaks. This data is intended to assist applications to open files without delay.

It is published as a supplement to the specification of the BWF, see [supplement 3 to EBU Tech 3285](#).

Audio Decision List

This system is intended to enable simple audio projects to be transferred between different equipment and different platforms. The development has been led by the AES and is published as AES31-3.

Proposed optional chunks for BWF files

A number of optional chunks have been proposed and the EBU has invited comments from industry and potential users. Please send any comments to the [EBU Technical Department](#). The EBU will decide if the proposed chunks are accepted as sufficiently widely applicable to be registered for general use.

Future developments

The EBU, the AES and industry have proposed other optional chunks to be used with BWF files for specific applications. These chunks are still under consideration.

Multichannel audio:

This proposal will define a format for carrying multichannel PCM audio signals in BWF files, together with an information chunk.

Film Chunk: This chunk is intended for audio files used in film production. The development is being led by the AES. Information on AES developments is available from the AES [website](#).

References

[EBU Recommendation R98-1999](#) - Format for CodingHistory field in Broadcast Wave Format files, BWF

[EBU Recommendation R99-1999](#) - 'Unique' Source Identifier (USID) for use in the Originator Reference field of the Broadcast Wave Format

[Articles in the EBU Technical Review](#)

Further questions?

Please send any further questions to the [EBU Technical Department](#)