Specification of the Broadcast Wave Format
A format for audio data files in broadcasting

Supplement 4: <link> Chunk

1 Introduction

The Broadcast Wave Format (BWF) specification **[1]** allows a maximum file size of 4 Gigabyte although in practice many RIFF / Wave applications will only support a maximum file size of 2 Gigabyte. For audio data in excess of these limits it is necessary to split the audio information into more than one BWF file. The <link> chunk provides link-up data for a seamless audio output spread over several files.

2 Terminology

File-set	The set of linked files belonging to one continuous audio signal.
Filename	The names given to each file in the file-set.
File list	A list of the Filenames in the File-set
"Actual" attribute	An attribute flagging the filename in the file list as being the current (or "actual") file. All other filenames in the file list are flagged as "other" (see example in section 5.).
File identifier	An optional identifier which shall be the same for all files of a file-set.
'Private' element	An additional element in the chunk to store proprietary information in the file list.
<link> chunk	A chunk contained in all the files of a file-set. It contains a header followed by a file list and optionally a file identifier and "private" element. The data in the chunk is stored in XML 1.0 format, a widespread format for data exchange **[2]**.

3 Link chunk Structure

3.1 Overview

The <link> chunk consists of a header followed by the link-up information stored in XML (eXtensible Markup Language) **[2]** format. The overall length of the chunk will be variable.

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

3.2 Elements of the 'link' chunk

ckID This is the 4 character array {'l', 'i', 'n', 'k'}[1] for chunk identification.

CkSize This is the size of the data section of the chunk (not including the 8 bytes used by ckID and ckSize.)

XmlData This buffer contains the link-up information in XML (ASCII characters).

3.3 XML data structure in <xmlData> variable data field

The data structure is hierarchical. Data are stored in text strings. For the exact syntax specification a DTD (data transfer document) is added. This is described further in section 3.4.

---

[1] **Remark:** The definition DWORD ckID = "link" would not be unique. Different C-compilers produce different orders of the characters. Therefore we define char ckID[4] = {'l', 'i', 'n', 'k'} instead.

```
<LINK>
        <FILE type="…">
                <FILENUMBER>...</FILENUMBER>
                <FILENAME>...</FILENAME>
        </FILE>
        .......
        Possible further FILE elements
        .......
        <ID>...</ID>        optional
        <PRIVATE>           optional
                ..... implementation dependent
        </PRIVATE>
</LINK>
```

LINK	This is the root element of the XML data. LINK contains one or more FILE elements with the file description. It may also contain an identifier ID and/or a PRIVATE element.
ID	The identifier ID is common for all files of a given file-set. It is stored as a text string of characters permitted by the #PCDATA definition of the XML 1.0 specification, which includes all visible ASCII characters, spaces etc.
PRIVATE	The PRIVATE element may contain implementation-dependent information consisting of any XML data (such as further elements or #PCDATA).
FILE	The FILE element contains the FILENUMBER element and the FILENAME element. The type attribute shall be 'actual' in the case that the file in the list describes the file to which the chunk belongs. All other files shall have the type attribute 'other'. The filename of the file shall be the same as it appears in the file list.
FILENUMBER	Files shall be numbered sequentially according to their chronological order in the file-set. Integer numbers (ASCII characters) beginning with number 1 shall be used.
FILENAME	Text string stored in the same format as the ID.

3.4 DTD for XML structure of the <link> chunk

The DTD (document type definition) is described in the XML 1.0 specification as a definition of the syntax of an XML structure. The format and the attributes of the different elements of the <link> chunk are described below, including sub-elements and their multiplicity.

Element LINK shall contain one or more sub-elements FILE ('+' indicates one or more), it may contain a sub-element ID and a sub-element PRIVATE ('?' indicates one or none).

Each element FILE shall contain one sub-element FILENUMBER and one sub-element FILENAME.

A type attribute shall be specified, which may be either “actual” or “other”.

Sub-elements FILENUMBER, FILENAME and ID must contain text strings (called #PCDATA in XML).

Sub-element PRIVATE may contain any of the defined elements. If PRIVATE needs to contain elements other than the defined ones, the DTD must be modified accordingly.

```
<!ELEMENT LINK              (FILE+, ID?, PRIVATE?)>
<!ELEMENT FILE              (FILENUMBER, FILENAME)>
<!ATTLIST FILE              type (“actual” | “other”) #REQUIRED>
<!ELEMENT FILENUMBER        (#PCDATA)>
<!ELEMENT FILENAME          (#PCDATA)>
<!ELEMENT ID                (#PCDATA)>
<!ELEMENT PRIVATE           ANY>
```

4 Renaming of linked files

If one or more filenames is changed, the corresponding FILENAME entries in each of the <link> chunks belonging to the whole file-set shall be changed too.

5 Example

The continuous sound signal in this example has been split into a file-set of three BWF files called “Caruso_1.wav”, “Caruso_2.wav” and “Caruso_3.wav”. The XML structures of the <link> chunks of the three files are identical except for the type attribute.

5.1 <link> chunk of “Caruso_1.wav”:

```
<LINK>
        <FILE type="actual">
                <FILENUMBER>1</FILENUMBER>
                <FILENAME>Caruso_1.wav</FILENAME>
        </FILE>
        <FILE type="other">
                <FILENUMBER>2</FILENUMBER>
                <FILENAME>Caruso_2.wav</FILENAME>
        </FILE>
        <FILE type="other">
                <FILENUMBER>3</FILENUMBER>
                <FILENAME>Caruso_3.wav</FILENAME>
        </FILE>
        <ID>73365869</ID>
</LINK>
```

5.2 <link> chunk of “Caruso_2.wav”:

```
<LINK>
        <FILE type="other">
                <FILENUMBER>1</FILENUMBER>
                <FILENAME>Caruso_1.wav</FILENAME>
        </FILE>
        <FILE type="actual">
                <FILENUMBER>2</FILENUMBER>
                <FILENAME>Caruso_2.wav</FILENAME>
        </FILE>
        <FILE type="other">
                <FILENUMBER>3</FILENUMBER>
                <FILENAME>Caruso_3.wav</FILENAME>
        </FILE>
        <ID>73365869</ID>
</LINK>
```

5.3 <link> chunk of “Caruso_3.wav”:

```
<LINK>
        <FILE type="other">
                <FILENUMBER>1</FILENUMBER>
                <FILENAME>Caruso_1.wav</FILENAME>
        </FILE>
        <FILE type="other">
                <FILENUMBER>2</FILENUMBER>
                <FILENAME>Caruso_2.wav</FILENAME>
        </FILE>
        <FILE type="actual">
                <FILENUMBER>3</FILENUMBER>
                <FILENAME>Caruso_3.wav</FILENAME>
        </FILE>
        <ID>73365869</ID>
</LINK>
```

6 Bibliography

[1] EBU document Tech 3285: Specification of the Broadcast Wave Format – A format for audio files in broadcasting.

[2] Extensible Markup Language (XML) 1.0 W3C Recommendation 10-February-1998 http://www.w3.org/TR/1998/REC-xml-19980210.