BBC

Research & Development

White Paper

WHP 233

September 2012

Archive Preservation File Format: DigiBeta System

Philip de Nier

BRITISH BROADCASTING CORPORATION

White Paper WHP 233

Archive Preservation File Format: DigiBeta System

Philip de Nier

Abstract

This white paper describes the changes to the Archive Preservation MXF file format made as part of the BBC Archive Preservation Project to support Digital Betacam (DigiBeta) and similar standard definition tape formats. The changes are relative to the earlier D3 Preservation File Format.

Additional key words: Ingex, LTO, Infax

White Papers are distributed freely on request.

Authorisation of the Chief Scientist or General Manager is required for publication.

White Paper WHP 233

Archive Preservation File Format: DigiBeta System

Philip de Nier

1 Introduction

BBC Research and Development developed the D3 Preservation system for BBC Information and Archives to transfer around 100000 D3 tapes to files over a six year period starting November 2007. The 'D3 Preservation File Format' White Paper [1)] describes the MXF file format that was used.

In November 2009 the BBC Digital Media Initiative project requested that R&D extend the preservation system to support Digital Betacam (DigiBeta) tapes. This system would also be used by I&A in the next phase of their preservation project after completing the D3 transfer. The extended system was completed in January 2010 and has been used to ingest hundreds of hours of content into the new DMI Fabric production environment and archive system.

I&A started (initially through BBC Studios & Post Production) to use the extended system, commonly known as the DigiBeta system, for preservation of DigiBeta tapes around January 2012.

The following sections describe the changes that were made to the MXF file format for the DigiBeta system.

2 Footer Partition Status

The MXF file's footer partition status is set to Closed and Complete at the first write of the file. The D3 system set it to Open and Complete and updated it to Closed and Complete just before transferring to LTO. In the DigiBeta system the LTO transfer is optional and so the file is assumed to be closed.

3 DigiBeta Dropout Event Metadata

The DigiBeta system includes a DigiBeta dropout detector that analyses the video and tries to detect artefacts that are associated with errors in the data encoded using a DCT-based compression scheme. Errors in the compressed data could be a result of tape defects or playback faults.

The dropout detector analyses each field of the video image and calculates a normalised strength value that is an indication of how likely the images contains dropout errors. The maximum strength value from both fields is compared with a threshold value and if it exceeds that threshold then the value is stored in the MXF file.

The raw strength value is normalised using a lower and upper threshold value. The lower and upper threshold values are set in a system configuration file and have been set empirically such that images with clear dropouts have strength values above the upper threshold and images with clear no dropouts have strength values below the lower threshold.

The raw strength value is normalised using this formula:

$$\text{strength} = 100 \times (\text{raw\_strength} - \text{lower\_threshold}) / (\text{upper\_threshold} - \text{lower\_threshold})$$

The normalised strength values are stored in the MXF file in the same way as the VTR errors and PSE failures. An event track in the header metadata contains a sequence of DigiBeta Dropout Framework sets. The set key for the DigiBeta Dropout Framework set is as follows:

06.0e.2b.34.02.53.01.01.0d.04.01.01.01.04.00.00

The DigiBeta Dropout Framework set contains the normalised strength value (Table 1).

Name	Meaning	Type	UL
APP Strength	Normalised Digibeta dropout strength	Int32	06.0e.2b.34.01.01.01.01.0d.04.01.01.01.04.01.00

Table 1 DigiBeta Dropout Framework set properties

A DigiBeta Dropout Count property is added to the Preface set to provide a total of DigiBeta dropout errors detected (Table 2).

Name	Meaning	Type	UL
APP DigiBeta Dropout Count	Total number of DigiBeta dropouts detected	UInt32	06.0e.2b.34.01.01.01.01.0d.04.01.01.40.01.03.00

Table 2 Preface set DigiBeta dropout count extension

4 Timecode Break Event Metadata

The validity or state of the VITC and LTC timecodes is often a good indicator of where a programme item starts/stops or where tape or playback errors are likely to be.

A valid timecode is defined to be timecode that equals the previous timecode plus 1 frame. A timecode break event is defined to be when a timecode transitions from a valid to an invalid state and vice versa. For example, a single dropped frame will result in 2 timecode break events: one event when the timecode becomes invalid (doesn't equal the previous timecode plus 1) and another event when the timecode returns to being valid.

The timecode break events are stored in the MXF file in the same way as VTR error and PSE failure events. An event track in the header metadata contains a sequence of Timecode Break Framework sets. The set key for the Timecode Break Framework set is as follows:

06.0e.2b.34.02.53.01.01.0d.04.01.01.01.05.00.00

The Timecode Break Framework set has a Timecode Type property which indicates which of the timecode types, VITC and/or LTC, are associated with the event (Table 3). Bit 0 of the Timecode Type property is set for VITC and bit 1 is set for LTC.

Name	Meaning	Type	UL
APP Timecode Type	Flags which timecode types are associated with this event	UInt16	06.0e.2b.34.01.01.01.01.0d.04.01.01.01.05.01.00

Table 3 Timecode Break Framework set properties

A Timecode Break Count property is added to the Preface set to provide a total of timecode break events detected (Table 4).

Name	Meaning	Type	UL
APP Timecode Break Count	Total number of Timecode break events	UInt32	06.0e.2b.34.01.01.01.01.0d.04.01.01.40.01.04.00

Table 4 Preface set Timecode break event count extension

5 Frame Checksum

The audio and video data is expected to be stored and transferred between systems many times and so a checksum is useful to be able to detect storage and transfer errors. Also, only part of the audio and video frame data may be stored and transferred, possibly in different file containers, and therefore calculating a checksum of every individual frame would be useful.

A CRC-32 checksum is calculated for every video and audio frame and stored in the MXF file. The CRC-32 checksum type is the same as the type defined in the PNG specification [3)]. See section

5.5 Cyclic Redundancy Code algorithm in the PNG specification as well as the example implementation in Annex D.

The checksum array is stored in the System Scheme 1 system item set in the essence container that is also used for the VITC and LTC timecodes. The local tag for the checksum array is set to 0xffff and the value consists of an array header (array length and array element length) followed by 32-bit big endian checksum values.

For example, the system item bytes for a VITC and LTC value equal to 09:58:10:12 and checksums for video and 4 audio elements are as follows (bytes are shown in hexadecimal notation and the start of each item is highlighted in bold text):

```
Key               : 06.0e.2b.34.02.53.01.01.0d.01.03.01.14.02.01.00
Len               : 83.00.00.3c
Timecode array    : 01.02
Local len         : 00.18
Array len         : 00.00.00.02
Array element len : 00.00.00.08
VITC element      : 12.10.58.09.00.00.00.00
LTC element       : 12.10.58.09.00.00.00.00
Checksum array    : ff.ff
Local len         : 00.1c
Array len         : 00.00.00.05
Array element len : 00.00.00.04
Video checksum    : 8b.cf.fa.3c
Audio 1 checksum  : 89.45.12.55
Audio 2 checksum  : 6f.89.01.06
Audio 3 checksum  : 32.cc.10.9a
Audio 4 checksum  : 32.cc.10.9a
```

6 10-bit Uncompressed Video

The DigiBeta system allows a choice between storing 8-bit (the same as the D3 system) and 10-bit uncompressed video. The v210 [4)] format is used for 10-bit uncompressed video.

The 10-bit video frame size is 1105920 bytes and is derived as follows:

Frame size = (width + 47) / 48 x 128 x height = (720 + 47) / 48 x 128 x 576 = 1105920

The Component Depth property in the CDCI Essence Descriptor set is equal to 10.

The 2011 revision of the MXF specification [5)] defined a set of picture coding labels in section G.2.25 that can be used to unambiguously identify uncompressed picture component layouts. The Picture Coding Label property in the CDCI Essence Descriptor set is equal to the value below to specify that the v210 format is used:

06.0e.2b.34.04.01.01.0a.04.01.02.01.01.02.02.01

Note that, as stated in [1], the MXF file format for the DigiBeta system still conforms to the 2004 version of the MXF specification [2].

7 Miscellaneous Changes

The Aspect Ratio property in the CDCI Essence Descriptor can be 4:3 or 16:9.

8 References

1) Philip de Nier and Phil Tudor, D3 Preservation File Format, BBC WHP 167, July 2008
2) SMPTE, Material Exchange Format (MXF), SMPTE 377M-2004
3) W3C, Portable Network Graphics (PNG) Specification (Second Edition)", 10 November 2003, http://www.w3.org/TR/2003/REC-PNG-20031110
4) Apple Quicktime, 'v210' 4:2:2 Compression Type, https://developer.apple.com/quicktime/icefloe/dispatch019.html#v210
5) SMPTE, Material Exchange Format (MXF), SMPTE ST 377-1:2011