Document Name			
Document Title	Significant Properties Testing Report: Audio Recordings		
Work Package			
Author(s) & project role	Gareth Knight, Digital Curation Specialist		
Date	12 February 2010	**Filename**	audio-significantproperties-v11
Access	☑ Project and JISC internal		☑ General dissemination

Document History		
Version	**Date**	**Comments**
1.1	12 February 10	Copied text into revised template
1.0	30 March 09	Final version of significant properties testing report
0.2	20 October 08	First draft of Significant properties testing report
0.1	13 February 08	Preliminary list of properties and criteria outlined in 'Framework for the definition of significant Properties'

Contributors

The following people have made direct or indirect contribution to this report: Adrian Brown, Mike Coyne, Stephen Grace, Lynne Montague and Mike Stapleton.

Intended Audience

This document is written for use by the InSPECT project team, the JISC community and those interested in digital preservation.

Table of Contents

Project Overview 3
Purpose of the report 3
1. Introduction 3
1.1. Overview of audio 3
1.2. Structure of an audio object 3
1.3. Overview of metadata standards 3
1.4. Application of the Performance model 3
2. Testing requirements 3
2.1. Significant properties that must be maintained 3
2.2. Assessment of significant properties 3
2.2.1. Audio stream 3
2.2.2. Embedded metadata 3
2.2.3. Summary 3
3. Methodology 3
3.1. Representation Formats 3
3.2. Software tools 3
Experiment 3
3.3. Sample data to be analysed 3
3.4. Testing Environment 3
3.5. Experiment testing 3
3.6. Experiment 3
3. Conclusion 3
Appendix 1: Software Tools 3
FFIdent 3
ID3Lib 3
Java Metadata Collection (JMDC) 3
JHOVE 3
Kaa Metadata Modules 3
Lib Extractor 3
NLNZ Metadata Extractor 3
XENA 3
Java Sound API 3
MPEG7audioenc 3
MPEG-7 Low Level Audio Descriptors Extractor 3
MPEG-7 Spoken Content Demonstrator 3
Hachoir 3
Meta Track 3
Gnormalize 3
OggConvert 3
SoX 3
Header Investigator 3
MetaFlac 3
Appendix 2: Audio object description 3
Short1.wav 3
Short2.wav 3
Voice1.wav 3
Voice2.wav 3
Appendix 3: Metadata elements contained in the Bext chunk of Microsoft Broadcast Wave 3
References 3

Project Overview

Significant properties are those aspects of a digital record that must be preserved over time in order for the Information Object to remain accessible and meaningful. The InSPECT Project is funded by JISC to investigate methods for maintaining the authenticity of digital resources across digital environments and transformation processes. It has produced a framework for the analysis of significant properties and creating a set of reports that outline its application to four object types - types – audio recordings, raster images, structured text and e-mail – that will contribute and advance strategies for the characterisation and maintenance of significant properties over time.

Purpose of the report

This report examines the notion of significant properties as it applies to digital audio. It seeks to identify the significant properties of audio that must be maintained by examining each of its constituent elements and analyzing their designated function. It goes on to examine strategies that may be utilized to maintain access to audio assets in the long-term. Finally, it outlines a set of experiments that were performed by the project team to identify and evaluate tools that may be utilized to convert significant properties from one form to another.

1. Introduction

1.1. Overview of audio

Sound in its original (analogue) state is a series of air vibrations (compressions and rarefactions), which are captured by our ears and then converted to electronic impulses for interpretation. Sound waves are commonly measured by their frequency and amplitude. The ability to hear sound is subject to a range of factors, including the receptive capabilities of the listener and the medium through which it is transmitted. The content of an audio recording and the functions it is required to perform are diverse. A recording may consist of one or more people talking, a music performance, or indeed any type of sound.

The storage and management of audio recordings has been an area for concern for almost 100 years. Early recordings were stored on analogue media systems, such as wire record device, gramophone records and magnetic tape. Since the 1960s, there has been an increasing amount of audio being stored in digital formats. A digital audio system stores sound as a series of electrical on/off pulses that can be subsequently interpreted by a digital-to-analogue-converter and converted into air vibrations, in order to be heard by the listener. Institutions located in government, academia, as well as the commercial sector are collecting an increasing amount of audio recordings. These may be stored to fulfil business functions, providing a record of events that occurred for short-term use, or as cultural artefacts that must be maintained in the long-term.

1.2. Structure of an audio object

As a digital asset, an audio file may be considered a compound object that is able to encapsulate two ore more distinct types of information. The type of information and the method in which it is structured will often vary between audio recordings stored in different Representation Format (see section 3.1). The attributes of an audio object may be separated into several layers that provide different levels of granularity (as shown on figure 1), each layer of which may possess different attributes.

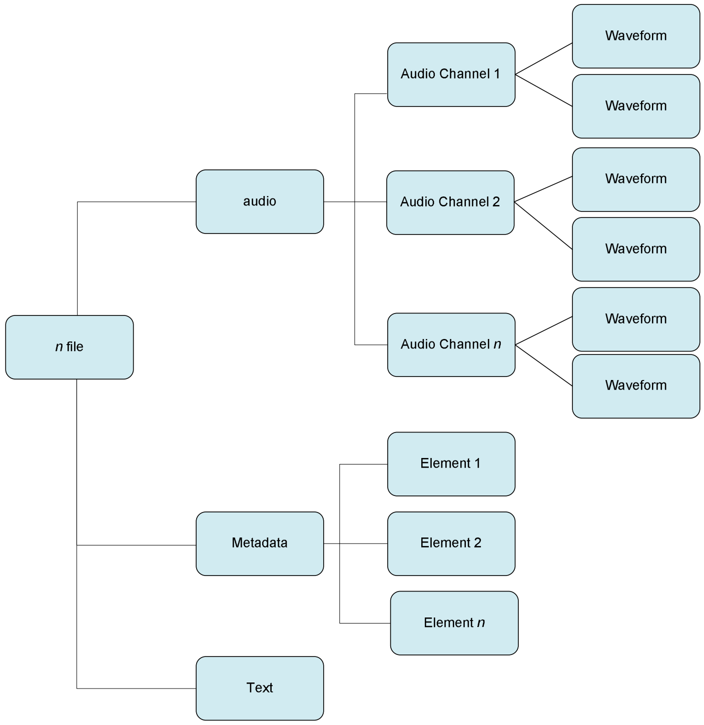


Figure 1: Structure of an audio object

Each component possesses attributes that must be maintained to ensure that it remains authentic over subsequent manifestations. Although they shares a common relationship and contributes information for the interpretation of the audio recording in its entirety, a preservation strategy may be adopted in which each component is managed separately. For example, the audio component may be converted from MP3 to MS Wav, metadata may be exported to an XML record and textual information may be exported to a text file.

Digital audio may be encapsulated in one of several container formats, including RIFF, OGG and MPEG. The specification for each format differs in the type of information that they may encapsulate. However, they typically allow 2 – 3 types of encoded information to be provided:

1. Audio bit-stream

The audio bitstream represents the primary target for preservation and is the entity to which other components embedded in the audio object will relate. Encoding algorithms used for the storage of audio data may be classified into one of several classes, each of which differ in the method that they store information. Examples include:

- A continuous waveform composed of samples taken at specific time intervals (e.g. PCM, MP3)
- An instruction set that indicate the musical notes to be reproduced (e.g. Midi)
- Disparate waveforms that are processed and reproduced in a non-sequential manner (e.g. Modules)
- Music notation (e.g. CMusic)

The objective of each class is broadly similar – to store and reproduce audio, however the approach taken by each is almost entirely incompatible. Similar to the relationship between Vector and Raster encoding formats, it is possible to convert audio from one class to another. However, it would likely result in information loss. The analysis of Representation formats in this report is limited to those encoded as continuous waveform.

2. Metadata

Metadata provides textual information about the audio object, which may indicate the title and composer of the audio recording for resource discovery, the actions that have been performed on it and associated rights information.

3. Text and other information
A digital object may contain textual information created for different purposes, such as a transcript of an interview, song lyrics, or hyperlinks. For example, the Lyrics3 tagging system (http://www.id3.org/Lyrics3v2) for the MP3 format

1.3. Overview of metadata standards

The metadata standards relevant to the storage and preservation of digital audio fulfil many objectives, describing the intellectual content of a resource (resource discovery), providing information about ownership and rights management (administrative), recording the relationship between an object and its siblings (structural) and documenting its internal composition (technical). Standards range from the generic elements of Dublin Core, to highly complex, granular schemas, such as the draft AES-x098 standard for audio objects. The following section provides an overview of relevant data dictionaries and schemas.

Dublin Core
Dublin Core is an international standard for the definition of resource discovery (ISO 15836). It provides a core set of 15 semantic elements for the description of a resource that may have use across a broad range of domains and subject disciplines. Dublin Core metadata may be easily searched and shared between institutions of different types. However, its generic design makes it ill-suited to high-detailed descriptions.

AES-X098 Audio Engineering Society
AES-098 is a set of standards maintained by the Audio Engineering Society to support the curation and preservation of audio objects in analogue and digital form. Development began in 1999 with an announced completion date late 2008, though no publication has been made at the time that this project report was written[1]. It will consist of a set of reports that establish standards for the description of audio objects (AES-X098A), technical information on the structure of audio objects (AES-X098B) and process history of audio objects (AES-X098C). The AES-X098B specification is likely to have particular relevance for the InSPECT project in the long-term. Although the final version of the standard is forthcoming, a draft schema has been implemented in JHOVE which provides some indication of the metadata elements that are considered to be useful.

PREMIS
PREMIS is a data dictionary and associated XML schema for defining the technical composition of a digital object and the activities that have been performed on it. PREMIS 2.0, published in March 2008 implemented several changes from the earlier iteration, including an expanded rights metadata framework, a revised metadata schema and, of particular notability for the InSPECT project, the provision of extension elements to embed further granularity into each metadata category. The extension may be utilised to provide a more flexible structure within which significant properties can be defined and described.

PBCore
The PBCore[2] (Public Broadcasting Metadata Dictionary) is a Dublin Core application Profile that was created to enable the interchange of information between public broadcasting organisations in the USA. PBCore has 53 elements arranged in 15 containers and 3 sub-containers, all organized within four content classes (PBCoreIntellectualContent, PBCoreIntellectualProperty, PBCoreInstantiation & PBCoreExtensions). The element set specifies a number of elements of relevance to the significant properties of the audio recording, including formatTimeStart, formatBitDepth and formatDuration.

[1] Information on the publication status of the standard is available at http://www.aes.org/standards/b_policies/project-status.cfm
[2] A list of PBCore metadata elements can be found at http://www.pbcore.org/PBCore/UserGuide.html

1.4. Application of the Performance model

To determine the significant properties of a digital Record, a consistent, formal method of identifying the important aspects is required. The National Archives of Australia (2002) has developed a 'Performance Model', which has been adopted by the InSPECT Project.
The principle of the model is that the process of rendering the Information Object in a form that can be understood by a user requires some interaction between the underlying data object and interpretative software. The model is comprised of three components:

1. Source: the encoded data object that contains the text, still images, moving images, or other content for interpretation;
2. Process: the method in which the encoded data is interpreted, e.g. a software tool, an algorithm;
3. Performance: the recreation of the Information Object in a form that can be understood by the user.

The constituent components of an audio recording may be 'performed' using a number of methods when processed in different software applications (figure 2).

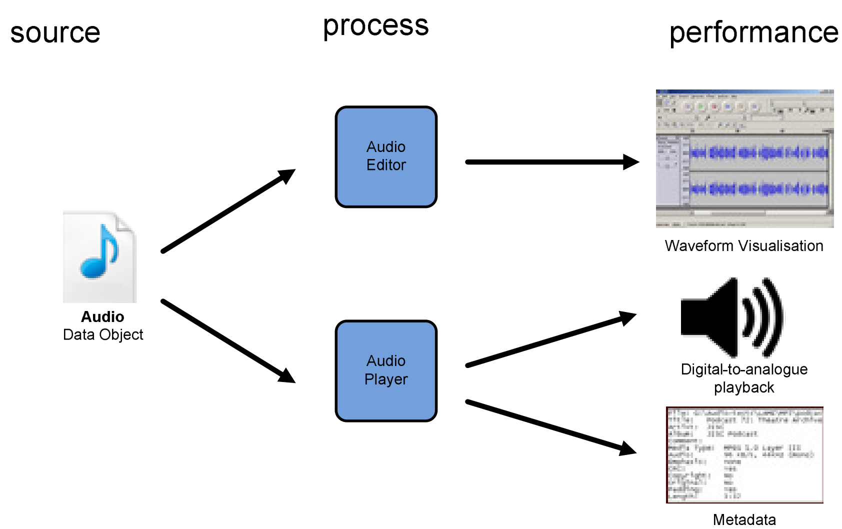


Figure 2 Application of the Performance Model to emails

A key concept in the Performance model is the recognition that the method in which the Source is processed will vary between members of the Designated Community and are likely to change over time as a result of the evolving technological environment. The recreation of the audio recording in an audible form is essential to understanding the intrinsic information contained within the audio stream. This raises the question, what elements must be retained for the audio recording to remain renderable and understandable?

2. Testing requirements

2.1. Significant properties that must be maintained

The identification of properties of a digital object that are worthy of preservation is not a simple task that can be analysed based upon a set of universal rules. A set of rules defined for one category of digital object may prove to be too restrictive when applied to unusual variations, or inappropriate for other object types. Instead, the InSPECT Project team has developed a methodology to identify factors that establish the authenticity and integrity of the Information Object through a combined technical and epistemological approach.

During the process of investigating the creation, storage and use of digital objects in a research environment it was found that the classification of significant properties was influenced by four key elements:

1. The form that the creator has chosen to express an intellectual or artistic idea and the method that they have used to communicate information
2. The function for which the digital object has been created to perform or the aims and objectives that its use will achieve.
3. The method in which information is encoded and stored in a digital environment, influenced by the encoding format and data standards in use.
4. The interpretation of the audience – the intended recipient of the audio recording or an unknown future user – that is accessing the information to achieve an objective.

The challenge for the curator is to identify and characteristics of a digital object that enable them to fulfil the required function of maintaining its authenticity and integrity throughout the digital lifecycle. This requires several questions to be considered:

1. What intellectual content is intended for communication by the creator and how is it represented in the source object?
2. What information is available to establish the provenance of the information?
3. What information is required or desired by the designated community to interpret the intellectual content in context?

The curator may be able to answer some, but not all of the questions that need to be asked. The first question may initially be considered quite simple (e.g. it may be suggested that the sound itself is the primary information intended for communication). However, different scenarios considered for question 3 may introduce additional complexity that requires some thought. A digitisation project allocated the task of digitising a vinyl disc collection will wish to record the sound waves that are stored on the disc. However, they may also wish to communicate the sound that the vinyl disc makes when played, in order to communicate the time period in which the audio was created. A similar interpretation can be made of the long-term value of different types of provenance information contained within the digital object. For 'born digital' objects that represent the original recording, the file creation date may provide the only method for establishing the provenance of the recording and is beneficial for communication to an end user. However, for digitised data (i.e. that which originated from an analogue source) the file creation date is one of several dates associated with its management and may be considered less useful for an end user.

2.2. Assessment of significant properties

To develop a list of the properties that may be significant for establishing the authenticity and integrity of a digital audio recording, the evaluator reviewed several specifications that are in use for the storage and description of digital audio. The review included the draft AES-X098B specification, the HUL DRS administrative metadata for digital audio schema[3], PBCore[4] and the Library of Congress AudioMD schema[5], as well as preservation guidance provided by the Indiana University Digital Library Sound Directions project[6], Council on Library and Information Resources & Library of Congress[7], Arts & Humanities Data Service[8] and CDP Digital Audio Working Group[9]. The AES-X098B standard, in particular was seen as being particularly valuable for the definition of significant properties for digital audio objects. However, the specification had not been published during the time period in which the InSPECT project was funded. In its absence, the evaluator reviewed each

[3] An data dictionary is available at http://preserve.harvard.edu/resources/audiometadata.pdf.
[4] A list of PBCore metadata elements can be found at http://www.pbcore.org/PBCore/UserGuide.html
[5] An element list is available at http://www.loc.gov/rr/mopic/avprot/DD_AMD.html
[6] Casey, M. & Gordon, B. (2007). Sound Directions: Best Practices for Audio Preservation. http://www.dlib.indiana.edu/projects/sounddirections/papersPresent/index.shtml
[7] Council on Library and Information Resources & and Library of Congress (2006). Capturing Analog Sound for Digital Preservation: http://www.clir.org/pubs/abstract/pub137abst.html
[8] Knight. G & McHugh, J (2005). Preservation Handbook: Digital Audio. http://ahds.ac.uk/preservation/audio-preservation-handbook.pdf
[9] CDP Digital Audio Working Group Digital Audio Best Practices (2005). Digital Audio Best Practices. Version 2.0. http://www.bcr.org/cdp/best/digital-audio-bp.pdf

specification in turn, identified commonalities and attempted to classify each unit by the function it performed in isolation or in conjunction with other units. The evaluator subsequently cross-matched the function of each element to the outlined questions and used the results to establish a set of properties that were considered essential.

For the purpose of analysis, the InSPECT project team examined audio objects at the second and third layer of analysis, as indicated in Figure 1. The assessor examined the audio stream in its entirety and the requirements of each channel. However, it was considered out of scope of the work to progress to analyse the properties of the waveform itself, e.g. by recording the peaks for each second.

2.2.1. Audio stream

The digital audio steam contains the sound that must be rendered to be understood. As a result of the literary review of preservation advice, the following requirements were established:

1. The number of distinct channels within the audio stream is unaltered;
2. The allocation of each channel is unaltered;
3. The duration of each audio channel is unaltered
4. The sound quality of the audio is equivalent or higher than the original;

The evaluator sought to identify characteristics that enabled a digital curator to monitor each function. The assessment of properties for the audio stream is derived from the draft AES-X098B Audio Standard for digital objects and draws upon the Harvard University Library 'DRS Administrative metadata for digital audio files' specification. In the absence of a final document describing the X-098B standard, the evaluator reviewed third-party implementations created by JHOVE and the Library of Congress implementation[10]. The analysis identified 7 elements that enabled a Curator to monitor an audio recording over subsequent manifestations for possible change.

Name	Definition	Function classification	Function description	Applicability
Duration	The intended length of the sound recording in Time-code character format (TCF).	Content: Length	Indicates if the audio is complete in its intended length. It may provide a simple indicator to identify if sound has been lost, as a result of mis-configuration or corruption.	All
Bit depth	The number of bits of information stored for each sample. Bit depth corresponds to the resolution of each sample. It limit quantities such as dynamic range and signal-to-noise ratio.	Rendering	The bit depth may provide an indicator of the audio quality. A reduction in the bit depth value of a converted object may indicate that the dynamic range of sample has been reduced and, as a result some quality loss has occurred.	Bit depth is primarily applied to lossless encoding, such as PCM. Lossy formats, such as MP3 assign a bit value to each individual sample.
Sample rate	The number of samples per second (or per other unit) at which each channel should be played.	Rendering	The sample rate may provide an indicator of audio quality. A greater number of samples may be recorded at higher sampling frequencies, e.g. 44.1kz or higher	All

[10] A draft implementation of the AES schema is available at http://hul.harvard.edu/ois/xml/xsd/drs/audioObject.xsd

			indicates CD quality, 8kHz is equivalent to telephone quality. A reduction in sample rate of a converted object may indicate that quality loss has occurred. However, an increase in sample rate may not be a source of concern.	
Number of channels	A numeric value that indicates the number of distinct streams within an audio object. An audio recording that contains two or more channels may output each channel through a different speaker or other output.	Content	The value is not essential, duplicating information that can be found by examining the Channel Number. However, it may be useful to store a value for curators to validate that content loss has not occurred. A comparison of a source and destination object that identifies a reduction in the channel number may indicate quality loss (one or more channels has been lost) or reduction (the channels have been merged and can no longer be treated independently.	All
Sound field	The aural space to which the channels are mapped, e.g. mono, stereo	Context	Sound field indicates the intended environment in which the designated community should experience the audio recording. The value complements the sound map location. The corruption of the sound field (e.g. surround-to-stereo, stereo-to-mono) may cause software to combine audio streams and output them to a limited number of speakers.	All
Channel Assignment: Channel Number	A unique ID assigned to each audio channel.	Context	An identifier assigned to each channel.	All
Channel Assignment: Sound map location	Maps a sound channel to a designated output.	Structure	The output location of a channel may communicate intrinsic information that supports the interpretation of the sound. It has only limited	All

			use when handling a mono recording. However, its value will increase when handling a greater number of audio channels. The corruption or loss of the sound map location may cause confusion to the listener	

Table 1: A core set of properties for describing the high-level composition of an audio stream

Each property outlined in Table 1 will require rules to be defined that indicate acceptable variation in the recorded value between different manifestations of an audio recording. The need to distinguish between the acceptable values for a ‘high-quality’ digital master and acceptable values for a ‘low-quality’ derivative intended for dissemination is particularly important. It is likely that the majority of users will require content-specific properties, such as Duration to remain the same between different manifestations – a reduction in the duration of an audio recording may indicate that information has been lost. However, It is possible that the audio quality may be changed without affecting the underlying information. For example, the sample rate may be reduced from 44.1Hz to 22Hz and stereo audio may be merged into a single channel. Although a lower value may denote that the audio recording in a specific derivative has been encoded at a lower quality. It is not necessary to presume that this is unwanted quality reduction for all scenarios. The Significant Properties Data Dictionary enables the curator to specify an Upper and Lower specification limit for each property, indicating the allowable deviation from the target value where a characteristic continues to be an acceptable representation of the information content. For example, the ideal sample rate of an digital master may be 48000Hz, while the allowable tolerance level specified by the curator may indicate that a numeric value between 44100Hz and a hypothetical maximum of 96000Hz is acceptable for a digital master, or a numeric value between 11000Hz and 22000Hz is acceptable for a distribution manifestation.

2.2.2. Embedded metadata

Metadata embedded within the audio object may provide textual information about the audio recording, indicating its purpose, who created it and when, the actions that have been performed on it, associated rights and other information.

An audio object may contain one of several different metadata formats, influenced by the representation format in use and the creator’s choices. Several metadata formats have been developed to fulfil a specific purpose (e.g. ID3 metadata embedded within an MP3 file). However, some metadata formats are intended to store different types of metadata. The Bext chunk embedded within a Broadcast Wave file is one such example, which may contain a mixture of technical and descriptive metadata. Although it is feasible to extract the information, the mixture of different types of content can prove to be problematic for a curator, potentially resulting in the storage and distribution of metadata with an object (e.g. stored as MP3, AIFF) that refers to the technical composition of the BWF-encoded original. To establish if the Functional Analysis based methodology implemented by the project can be applied to metadata, the evaluator consulted the public documentation provided for the BWF bext format[11], identified a set of 13 elements that may be contained within an audio object and attempted to classify each by the function it may perform. A full outline of the metadata elements, indicating the justification for the decisions on significance is provided in Appendix 3. Table 2 specifies 6 elements that were considered essential for preservation.

Name	**Definition**	**Function classification**	**Function description**
Description	An ASCII string that contains a free text description of the	Context	If completed, it may provide qualitative information that establishes the provenance of the audio recording.

[11] See ‘EBU Tech 3285 - Specification of the Broadcast Wave Format (BWF) - Version 1 - first edition (2001)‘, available at http://www.ebu.ch/CMSimages/en/tec_doc_t3285_tcm6-10544.pdf

	sound sequence.		
Originator	An ASCII string that may contain the name of the creator of the audio.	Context	If completed, it may provide qualitative information that establishes the provenance of the audio recording.
OriginatorReference	An ASCII string that contains a non ambiguous reference allocated by the originating organization[12]	Context	If completed, it may provide qualitative information that establishes the provenance of the audio recording.
OriginationDate	Indicates the creation date of the audio sequence. Format is YYYY-MM-DD	Context	If completed it may establish the creation point of the original recording, which is useful for establishing its provenance. However, it may be considered unnecessary for some users if it indicates the date at which a digital manifestation of an analogue original was created.
OriginationTime	The time that the audio sequence was created. Format is hh-mm-ss.	Context	If completed it may establish the creation point of the original recording, which is useful for establishing its provenance. However, it may be considered unnecessary for some users if it indicates the date at which a digital manifestation of an analogue original was created.
Coding History[13]	An ASCII text field of non-restricted length that may be used to describe the encoding process applied to each manifestation of the Information Object. Each entry is terminated by a Carriage Return+Line Feed. Recommendations for a Coding History format are provided in EBU Recommendation R98-1999	Context: provenance	The field may be beneficial for curators who wish to understand the activities performed on a digital object, particularly if it has been provided by a third-party and no other information is available. However, the Coding History is not essential for the performance of the audio recording or understanding the context of its creation.
Quality Report[14]	A text field that may be used to describe events that affect the quality of the recording sound signal. Each event is	Context: provenance	The field may be beneficial for curators who wish to understand the activities performed on a digital object, particularly if it has been provided by a third-party and no other information is available. However, the Coding History

[12] See EBU Technical Recommendation R99-1999. 'Unique' Source identifier (USID) for use in the OriginatorReference field of the Broadcast Wave Format. https://www.ebu.ch/CMSimages/en/tec_text_r99-1999_tcm6-4689.pdf

[13] See European Broadcasting Union (199). EBU Technical Recommendation R98-1999. Format for the <CodingHistory> field in Broadcast Wave Format files, BWF, available at www.ebu.ch/CMSimages/en/tec_text_r98-1999_tcm6-4709.pdf

[14] Coding History, Quality Report and Cue Sheet are described in detail in European Broadcasting Union (2001). BWF Summary 2 - Capture Report, available at http://www.ebu.ch/CMSimages/en/tec_doc_t3285_s2_tcm6-10482.pdf

	listed with details of the type of event, exact time stamps, priority, event status and other quality parameters.		is not essential for the performance of the audio recording or understanding the context of its creation.
Cue Sheet	A list that identifies one or more events within the sound recording, e.g. the beginning of an aria or the starting point of an important speech. Each event is recorded using a time code and description	Context: provenance	The field may provide contextual information that enables the listener to identify specific components of the audio stream.

Table 2: A core set of properties for the bext metadata chunk

The application of a function analysis based methodology was effective in analysing metadata elements, although problems were encountered when applying the method to unique identifiers. The Unique Media Identifier (UMID) is a system identifier that is unique to each manifestation of an object. Although it is a characteristic of the manifestation, it cannot be easily classified as representation information of the digital object or a significant property of the information object.

2.2.3. Summary

The suggested list of significant properties of audio that need to be maintained, within the scope and definition of the InSPECT project is:

1. Duration
2. Bit depth
3. Sample rate
4. Number of channels
5. Sound field
6. Sound map location for each channel

If the audio recording contains BEXT formatted metadata the following information should be retained:

7. Description
8. Originator
9. OriginatorReference
10. OriginationDate
11. OriginationTime
12. Coding History
13. Quality Report
14. Cue Sheet

3. Methodology

3.1. Representation Formats

Representation format is a general term that describes the method in which information is stored. In its abstract form, a representation format may be applied to many types of information. Restrictions on the type and extent of information are imposed when handling representation formats intended for a specific purpose. To provide a simple example, a representation format for image data is unlikely to be able to contain audio. Limitations may be imposed, even if information is stored in a representation format of the correct type. Specific properties of the information content may be degraded or removed when it is stored in a representation format.

Many of the representation format developed for the storage of audio data are compound objects that may contain several types of information. An Audio object may encapsulate several data streams, each of which conform to different standards and fulfil different functions, e.g. an audio stream and associated metadata. The ability of an encoding format to store the significant properties of digital audio, as defined in this paper, is dependent upon the design principles adopted when it was created. Audio data may be stored in an uncompressed, lossless compressed, or lossy compressed format. Encoding formats that fit into the first and second category maintain the quality of the audio recording, but require a greater amount of storage space. In contrast, lossy compression formats reduce file size by removing 'redundant' or 'irrelevant' information that is not considered to be in the audible range of most people.

3.1.1. Common representation formats

There are many Representation formats used for the storage of digital audio. This section provides a brief overview of several widely used formats that are addressed in this report:

- **FLAC:** FLAC[15] (Free Lossless Audio Codec) is a non-proprietary file format for the storage of audio as lossless, compressed data. It was developed and maintained by Xiph.Org Foundation, which promote it as a lossless replacement to the popular MP3 format. FLAC refers to an encoding format for the storage of audio streams and a 'Native Flac' container format for the storage of disparate information. A FLAC encoded audio stream may be embedded in several container formats, including Native Flac and Ogg.

- **MP3:** MPEG-1 Audio Layer 3, commonly shortened to "MP3" is an audio encoding format that uses a lossy compression algorithm to reduce the amount of data required to reproduce an audio recording. The compression is based upon the principle of 'perceptual coding' which discards or reduces accuracy of sections of sound that are considered beyond the hearing of most people. An MP3 audio file frequently contains an additional ID3 metadata component.

- **Vorbis:** Vorbis is an open source, lossy audio codec that is commonly used in conjunction with the Ogg container format. The codec development was led by the Xiph.Org Foundation, which created it as a replacement for the MP3 format. The latest official version of the codec is v1.2.0 which was published on July 25, 2007.

- **Microsoft Waveform (.wav)**: MS Wave is a container format intended for the storage of audio bitstreams. It was developed by Microsoft and IBM as an application of the Resource Interchange File Format (RIFF). An MS Wave file can store compressed and uncompressed audio data, the latter method being the most frequently used. The Linear Pulse Code Modulation (LPCM) is a popular example of a non-compressed, lossless algorithm that may be encoded in an MS Wave file. The encoding format maintains all samples of an audio recording, which has resulted in its adoption by several institution as an appropriate format for data curation.

---

[15] The specification and software tools to manipulate and playback the format are available at http://flac.sourceforge.net/

- **Broadcast Wave (BWF):** The Broadcast Wave Format (BWF) is an extension of the Microsoft Wave format. The specification was first published by the European Broadcasting Union in 1997, and was later revised in 2001 and 2003. The Broadcast Wave specification allows the inclusion of one or more of several extension chunks, including bext, iXML, qlty (Quality), mext (MPEG audio extension), levl (Peak Envelope), link and axml that allow the embedding of different types of metadata.

- **ID3:** ID3 is a metadata container format that may be embedded within a number of audio encoding formats, including, MP3, AIFF and MP4. There are two incompatible versions of the format - ID3v1 was developed in 1996 and is the most commonly used format for MP3s. It enables a standard set of metadata elements to be stored within the audio file. Supported elements include Title, Artist, Album, Year, Comment, Track, Genre, Speed, Start-time and End-time (the final 3 are found in the ID3v1 'extended tag' set). The ID3v1 specification lacks internationalisation support, indicating that text strings must be encoded in ISO-8859-1. The ID3v2 specification was released in 1998 and has received several updates and addendums. The standard specifies the storage of a number of 'frames', each of which contain metadata intended for a specific purpose, e.g. copyright, lyrics and cover art.

3.2. Software tools

3.2.1. Requirements

The criteria for identification and selection of software tools are intended to be inclusive, considering a range of software available on many different software platforms and published under different types of licence. General criteria for the selection of software tools

1. *Task:* Able to identify some or all properties of an Information Object that are considered to be significant;
2. *Task:* Able to extract significant properties of source format and store them in an open, well documented destination format;
3. *Environment:* Can be compiled or operated on a number of computing operating systems;
4. *Environment:* Can be implemented in a processing workflow;
5. Distribution: Are publicly available as a full product or in demo form for testing;
6. *Legal:* Provide clear guidance on the licence for use of the software in a production environment. Particular preference given to open source licence models;
7. *Documentation:* Are well documented.

3.2.2. Software tools available

The ability to identify, extract and convert the significant properties of an digital audio object require a combination of mainstream software tools that are able to analyse representation formats and bespoke development to combine software into an integrated workflow. The project team identified several software tools that were able to process audio objects and selected a subset for testing.

3.2.2.1. Characterisation tools

For the characterisation task, the project selected tools based upon the supported formats, type of information extracted and the level of detail.

- **MP3Info:** MP3Info[16] is a technical information viewer and ID3 1.x tag editor that supports the MP3 file format. The tool is available as source code and binary variants for Linux and MS Windows operating systems. The project tested version 0.8.5a of the tool, released on November 14, 2006.

- **JHOVE:** Jhove (JSTOR/Harvard Object Validation Environment) is a characterisation tool developed by JSTOR and Harvard University Library. It is able identify files conformant to a limited number of formats, measure their compliance to an existing standard and extract representation information for storage in an XML or text format. The tool takes a modular approach providing support for a limited, but extensible list of text, raster image and audio formats and variations, most notably AIFF and WAVE.

[16] Source code and binary distributions may be downloaded at http://www.ibiblio.org/mp3info/

- **SOXI:** SOXI is the characterisation component of the Sound eXchange (SoX) software tool. It is able to extract and display information from an audio header and provide a limited amount of technical metadata. Although the information provided is limited, it recognises a large number of audio formats[17].

The project team were unable to locate software that provided a detailed analysis for all of the analysed formats - the capabilities of each tool vary considerably, differing in the amount of information provided and the format in which it is presented. To assist with the analysis, the investigator merged the ouptu of each tool into a single text record for analysis.

3.2.2.2. Conversion tools

For the conversion task, the project selected the following software tools:

- **JHOVE:** A characterisation tool developed by JSTOR and Harvard University Library. The characterisation function provided by JHOVE makes it well equipped to extract metadata embedded within audio objects.

- **SoX (Sound eXchange):** SoX is a cross-platform command line utility licensed under the GNU General Public Licence (GPL) that can convert common audio format and apply various types of effects to audio recordings. SOX is distributed as source code and compiled binary for various types of operating system. However, the latter lacks MP3 support due to licence restrictions associated with the format. To integrate MP3 support, the experimenter obtained the SOX source code and compiled it with the LAME[18] (MP3 encoder) and MAD[19] (MPEG Audio Decoder) libraries using Visual Studio 2008.

- **FFMPEG:** A cross-platform command line utility licensed under the GNU General Public Licence (GPL) that is able to record, convert and stream digital audiovisual formats. It is composed of several open source and free software libraries, including libavcodec and libavformat.

Experiment

3.3. Sample data to be analysed

To demonstrate the identification, extraction and conversion of properties in a production environment the project team obtained data samples from several sources which were used as the basis for analysis. Prior to data selection, it was established that the data should represent real-world examples, i.e. audio recordings created in a production environment, as opposed to audio created in a controlled environment for analysis purposes. Specifically, audio recordings were selected that had been created by different software applications and stored in different file formats (bwf, wav, mp3).

The final test set is made up of audio recordings stored as follows:

- 4x Broadcast Wave (BWF)
- 3x Microsoft Wave
- 2x MP3

These files were obtained from the following sources:

Microsoft Wave
The Microsoft Wave files were obtained from the 'Imago Lecture Notes' collection. The audio recordings were created by Trevor Wishart and describe the compositional techniques used in the

[17] A full list of audio formats supported by SoX and SOXI is available at http://sox.sourceforge.net/soxformat.html
[18] http://lame.sourceforge.net/
[19] http://www.underbit.com/products/mad/

composition of 'Imago' and were deposited with the Art & Humanities Data Service (http://ahds.ac.uk/catalogue/collection.htm?uri=pa-1039-1) for publication.

Filename	Encoding
imago_lecturenotes_01.wav	RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, stereo 48000 Hz
imago_lecturenotes_02.wav;	RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, stereo 48000 Hz
imago_lecturenotes_10.wav	RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, stereo 48000 Hz

Table 3: A list of MS Wave files examined by the project

Broadcast Wave

The Broadcast Wave files were obtained from the European Broadcasting Union web site (http://www.sr.se/utveckling/tu/bwf/). The files are published by the EBU for use by software developers when implementing the BWF specification. The files are encoded as follows:

Filename	Encoding
Short1.wav	PCM format, 2 channel, 48000 Hz, 192000 bytes/sec, 16 bit
Short2.wav	An MPEG-1 Layer 2 encoded audio stream wrapped in a RIFF wrapper
Voice1.wav	PCM format, 2 channel, 48000 Hz, 192000 bytes/sec, 4 bytes/frame, 16 bit, 1 min 38.853 sec duration
Voice2.wav	RIFF 'WAVE', MPEG format, 2 channel, 48000 Hz, 48000 bytes/sec, 1152 bytes/frame, MPEG-1 Layer 2, 384000 bits/sec, Stereo

Table 4: A list of BWF files examined by the project

Further technical details on the BWF files can be found in Appendix 2.

MP3

The MP3 files were obtained from the JISC web site (http://www.jisc.ac.uk/news/podcasts.aspx). The files were created from a lossless master format and the quality reduced for distribution purposes.

Filename	Encoding
podcast72yvonneklein.mp3	MPEG 1.0 Layer III, 96 KB/s, 44KHz (mono), 3:32 duration & ID3 metadata
podcast72yvonneklein.mp3	MPEG 1.0 Layer III, 96 KB/s, 44KHz (mono), 3:32 duration & ID3 metadata

Table 5: A list of MP3 files examined by the project

Digital copies of both files may be obtained from http://www.jisc.ac.uk/news/stories/2009/01/podcast70annthunhurst.aspx and http://www.jisc.ac.uk/news/stories/2009/03/podcast72yvonneklein.aspx.

3.4. Testing Environment

All software testing was performed on a Dell GX260 fitted with a 32-bit Pentium 4 2GHz CPU, 1GB RAM and installed with Microsoft Windows XP Professional (version 2002) Service Pack 3.

3.5. Experiment testing

The following experiments, with the exception of Experiment 5 consisted of four distinct stages, outlined in Figure 3.

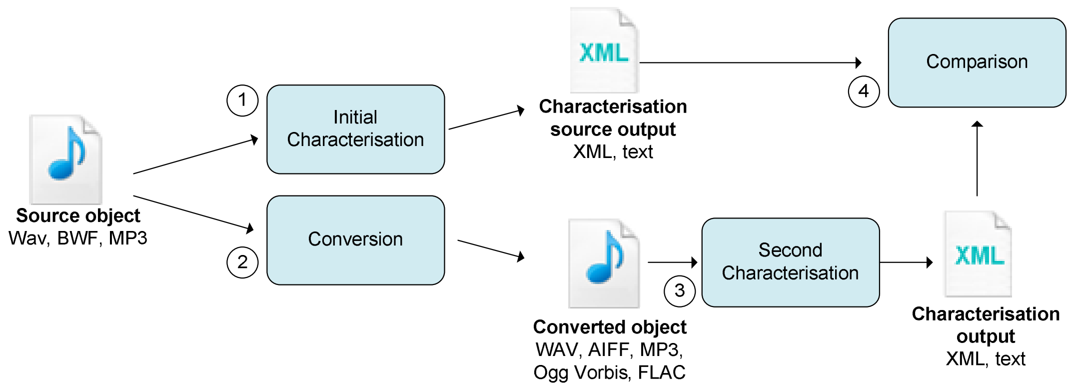


Figure 3: Illustration of the automated experiment procedure

1. **Initial characterisation:** The first characterisation stage examines the source object and extracts appropriate representation information. The information is utilized as a base line against which later characterisation activities are compared.

2. **Conversion:** Each source object is converted into several different file formats, including FLAC, OGG Vorbis, MP3, AIFF and MS Wave.

3. **Second characterisation:** The second characterisation stage examines the converted objects and extracts appropriate representation information. The project utilized several tools: JHOVE to examine AIFF, Broadcast Wave and MS Wave; MP3Info and SOXI for MP3; and SOXI for FLAC and OGG Vorbis.

4. **Comparison:** The result of the format conversion is evaluated through a combination of automated and manual comparison. A comparison is made between Representation information extracted from the source and converted object and an auditory assessment is made of each recording to identify noticeable differences.

3.6. Experiment

3.6.1. Experiment 1: Convert MS Wave to other formats using FFMPEG

For the first experiment we converted the collected MS Wave and Broadcast Wave audio recordings to four alternative formats – AIFF, FLAC, OGG and MP3 – using FFMPEG. FFMPEG is able to handle several audiovisual formats as input and output, but it does not handle additional metadata embedded within the file. The canonical manifestation of FFMPEG is distributed as source code only. However, there are a number of binary distributions available for different platforms that are linked from the project. We tested the Windows binary for FFmpeg SVN-r16586-Sherpya, available from http://ffmpeg.arrozcru.org/builds/.

Format conversion
Format conversion was performed using the Windows XP command line interpreter. The conversion from WAV to AIFF, FLAC and MP3 was performed using the following command:

ffmpeg -i [input_filename.wav] -sameq –f [format] [output_filename.ext]

For the WAV-to-OGG conversion, it was necessary to specify the encoding format.

ffmpeg -i [input_filename.wav] -sameq –acodec vorbis [output_filename.ogg]

The format conversion did not produce any errors or other reports.

Characterisation
The conversion strategy adopted for the experiment was validated through an automated comparison of the significant properties of the audio recording. The project team were unable to locate a single software tool capable of performing the required level of analysis for each of the encoding formats

being handled in the experiment. Therefore, it was necessary to combine the functionality of several tools to evaluate the format conversion. The initial characterisation was performed using JHOVE, which supports several variations of the WAVE format. Following the conversion, the technical properties of the derivatives were analysed and measured using a combination of JHOVE, SOXI and MP3Info. A comparison of the significant properties found in the source WAVE files and the converted manifestation are provided in table 6 – 9.

	Format				
	Wav [source]	AIFF	MP3	FLAC	OGG
Duration (hh:mm:ss)	00:02:53	00:02:53	00:02:53	00:02:53	00:02:53
Bit depth	16	16	-	16	16
Sample rate	48000	48000	48000	48000	48000
No. of channels	2	2	2	2	2
Sound field			Joint stereo		
Sound map location	Left, Right	Left, Right		?	?

Table 6: Analysis results for the conversion of imago_lecturenotes_01.wav to AIFF, MP3, FLAC and OGG using FFMPEG

	Format				
	Wav [source]	*AIFF*	*MP3*	*FLAC*	*OGG*
Duration (hh:mm:ss:msms)	00:00:14	00:00:14	00:00:14	00:00:14.45	00:00:14:44
Bit depth	16	16	-		16
Sample rate	48000	48000	48000		48000
No. of channels	2	2	2		2
Sound field			Joint stereo		
Sound map location	Left. Right	Left. Right			?

Table 7: Analysis results for the conversion of imago_lecturenotes_02.wav to AIFF, MP3, FLAC and OGG using FFMPEG

The duration of the Voice1.wav and Voice 2.wav were obtained from descriptive metadata embedded within each file.

	Format				
	Wav [source]	*AIFF*	*MP3*	*FLAC*	*OGG*
Duration (hh:mm:ss:msms)	00:01:38.853	00:01:38.85	00:01:39	00:01:38.85	00:01:38.84
Bit depth	16	16	-	16	16
Sample rate	48000	48000	48000	48000	48000
No. of channels	2	2	2	2	2
Sound field			Joint stereo		
Sound map location	Left, Right	Left, Right		?	?

Table 8: Analysis results for the conversion of Voice1.wav to AIFF, MP3, FLAC and OGG using FFMPEG

	Format				
	Wav [source]	AIFF	MP3	FLAC	OGG
Duration (hh:mm:ss:msms)	00:01:41.088	00:01:41	00:01:41	00:01:41.09	00:01:41.08
Bit depth	16	16	-	16	16
Sample rate	48000	48000	48000	48000	48000
No. of channels	2	2	2	2	2
Sound field					
Sound map location	Left, Right	Left, Right	?	?	?

Table 9: Analysis results for the conversion of Voice2.wav to AIFF, MP3, FLAC and OGG using FFMPEG

The significant properties of an audio recording were correctly maintained by FFMPEG when converting the Broadcast WAVE and Microsoft Wav format to AIFF, MP3, FLAC and OGG. However, there were some minor differences in the analysis results obtained for the bit depth and duration. The former is caused by the inherent characteristics of lossy encoding formats, which utilise a variable bit depth for each sample. The latter may be an issue for concern, though is likely to be caused by the variable handling of milliseconds in different software applications.

3.6.2. Experiment 2: Convert MP3 to other formats using FFMPEG

For the second experiment we converted the collected MP3 Podcast recordings to four alternative formats – AIFF, FLAC, OGG and WAV – using FFMPEG. As in Experiment 1, we tested the Windows binary for FFmpeg SVN-r16586-Sherpya, available from http://ffmpeg.arrozcru.org/builds/.

Format conversion
Format conversion was performed using the Windows XP command line interpreter. The conversion from WAV to AIFF, FLAC and MP3 was performed using the following command:

ffmpeg -i [input_filename.mp3] -sameq –f [format] [output_filename.ext]

The format conversion to WAV, AIFF and FLAC completed successfully without any errors or other reports. However, the MP3-to-OGG conversion requires the tester to specify the sample rate to be used, potentially due to the lossy-to-lossy conversion being performed. The experimenter used the same sample rate (48,000Hz) as found in other audio recordings. However, the conversion technique may result in some unnoticed quality loss occurring.

Characterisation
The conversion strategy adopted for the experiment was validated through an automated comparison of the significant properties of the audio recording. As noted, the project team were unable to locate a single software tool able to analyse each encoding format in the experiment. Therefore, it was necessary to combine the functionality of several tools to evaluate the format conversion. The initial characterisation of each of the MP3 recordings was performed using MP3Info. Following the conversion, the technical properties of the derivatives were analysed and measured using a combination of JHOVE and SOXI. A comparison of the significant properties found in the source WAVE files and the converted manifestation are provided in table 10 – 11.

	Format				
	MP3 [source]	*AIFF*	*WAV*	*FLAC*	*OGG*
Duration (hh:mm:ss)	00:06:15	00:06:15	00:06:15	00:06:15.64	00:06:15
Bit depth	-	16	16	16	-
Sample rate	44100	44100	44100	44100	44100
No. of channels	1	1	1	1	1
Sound field	Mono			?	?
Sound map location	?	"Unknown"	"Unknown"	?	?

Table 10: Analysis results for the conversion of podcast70annthunhurst.mp3 to AIFF, WAV, FLAC and OGG using FFMPEG

	Format				
	MP3 [source]	AIFF	WAV	FLAC	OGG
Duration (hh:mm:ss)	00:03:32	00:03:32	00:03:32	00:03:32.40	00:03:32
Bit depth	-	16	16	16	-
Sample rate	44100	44100	44100	44100	44100
No. of channels	1	1	1	1	1
Sound field				?	?
Sound map location		"UNKNOWN"	"UNKNOWN"	?	?

Table 11: Analysis results for the conversion of podcast72yvonneklein.mp3 to AIFF, WAV, FLAC and OGG using FFMPEG

The significant properties of an audio recording were correctly recognised by FFMPEG and were maintained when converting MP3 format to AIFF, WAV and FLAC. However, the tool required the quality level to be manually configured when converting from MP3 to OGG. As noted in Experiment 1, there were some minor differences in the analysis results obtained for the bit depth and duration. The former is caused by the inherent characteristics of lossy encoding formats, which utilise a variable bit depth for each sample. The latter may be an issue for concern, though is likely to be caused by the variable handling of milliseconds in different software applications.

3.6.3. Experiment 3: Convert Broadcast Wave (BWF) and MS Wave to other formats using SoX

For the third experiment we converted the collected MS Wave and Broadcast Wave audio recordings to four alternative formats – AIFF, FLAC, OGG and MP3 – using SOX. SoX is a cross-platform command line utility that can perform format conversion and apply various types of effects to audio recordings. SOX is distributed as source code and compiled binary for various types of operating system. However, the latter lacks MP3 support due to licence restrictions associated with the format. To integrate MP3 support, the experimenter obtained the SOX source code and compiled it with the LAME[20] (MP3 encoder) and MAD[21] (MPEG Audio Decoder) libraries using Visual Studio 2008.

Format conversion
The software tool was configured through the Windows XP command line interpreter, to set the input and output filenames. The tool automatically recognised the quality level of the source file and output at the same quality level in the destination format.

Characterisation
The conversion strategy adopted for the experiment was validated through an automated comparison of the significant properties of the audio recording. The initial characterisation was performed using JHOVE, which supports several variations of the WAVE format. Following the conversion, the technical properties of the derivatives were analysed and measured using a combination of JHOVE, SOXI and MP3Info. It was found that the software tool had automatically inserted a "Processed by SoX" text entry into the Comment field of each object.

A comparison of the significant properties found in the source WAVE files and the converted manifestation are provided in table 12 – 15.

	Format				
	Wav [source]	AIFF	MP3	FLAC	OGG
Duration (hh:mm:ss:msms)	00:01:38.853	00:01:38.85	00:01:39.85	00:01:39.85	00:01:39.85
Bit depth	16	16	-	16	16
Sample rate	48000	48000	48000	48000	48000
No. of channels	2	2	2	2	2
Sound field			Joint stereo	?	?
Sound map location	Left, Right	Left, Right		?	?

Table 12: Analysis results for the conversion of Voice1.wav to AIFF, MP3, FLAC and OGG using SOX

	Format				
	Wav [source]	AIFF	MP3	FLAC	OGG
Duration (hh:mm:ss:msms)	00:01:41.088	00:01:41	00:01:41	00:01:41.09	00:01:41.08
Bit depth	-	16	-	16	16
Sample rate	48000	48000	48000	48000	48000
No. of channels	2	2	2	2	2

[20] http://lame.sourceforge.net/
[21] http://www.underbit.com/products/mad/

Sound field					
Sound map location	Left, Right	Left, Right	?	?	?

Table 13: Analysis results for the conversion of Voice2.wav to AIFF, MP3, FLAC and OGG using SOX

	Format				
	Wav [source]	AIFF	MP3	FLAC	OGG
Duration (hh:mm:ss)	00:02:53	00:02:53	00:02:53	00:02:53	00:02:53
Bit depth	16	16	-	16	16
Sample rate	48000	48000	48000	48000	48000
No. of channels	2	2	2	2	2
Sound field			Joint stereo		
Sound map location	Left, Right	Left, Right		?	?

Table 14: Analysis results for the conversion of imago_lecturenotes_01.wav to AIFF, MP3, FLAC and OGG using SOX

	Format				
	Wav [source]	AIFF	MP3	FLAC	OGG
Duration (hh:mm:ss:msms)	00:00:14	00:00:14	00:00:14	00:00:14.45	00:00:14:44
Bit depth	16	16	-		16
Sample rate	48000	48000	48000		48000
No. of channels	2	2	2		2
Sound field			Joint stereo		
Sound map location	Left. Right	Left. Right			?

Table 15: Analysis results for the conversion of imago_lecturenotes_02.wav to AIFF, MP3, FLAC and OGG using SOX

The significant properties of an audio recording were correctly recognised by SOX and were maintained when converting Broadcast Wave and MS Wave format to AIFF, MP3, OGG and FLAC. As noted in other Experiments, there were some minor differences in the analysis results obtained for the bit depth and duration.

3.6.4. Experiment 4: Convert MP3 to other formats using SoX

For the fourth experiment we converted the collected MP3 Podcast recordings to four alternative formats – AIFF, FLAC, OGG and WAV – using SOX. As in Experiment 3, we utilised the compiled version of SOX that integrated MP3 encoding/decoding.

Format conversion
The software tool was configured through the Windows XP command line interpreter, to set the input and output filenames. Similar to Experiment 2, the tool required the bit depth to be manually configured when converting the MP3 to an alternative format.

Characterisation
The conversion strategy adopted for the experiment was validated through an automated comparison of the significant properties of the audio recording. As noted, the project team were unable to locate a single software tool able to analyse each encoding format in the experiment. Therefore, it was necessary to combine the functionality of several tools to evaluate the format conversion. The initial characterisation of each of the MP3 recordings was performed using MP3Info. Following the conversion, the technical properties of the derivatives were analysed and measured using a combination of JHOVE and SOXI.

A comparison of the significant properties found in the source MP3 files and the converted manifestation are provided in table 16 – 17.

	Format				
	MP3 [source]	AIFF	WAV	FLAC	OGG
Duration (hh:mm:ss)	00:03:32:48	00:03:32:48	00:03:32:48	00:03:32.40	FLAC
Bit depth	-	16	16	16	00:03:32.40
Sample rate	44100	44100	44100	44100	16
No. of channels	1	1	1	1	44100
Sound field				?	1
Sound map location		“UNKNOWN”	“UNKNOWN”	?	?

Table 16: Analysis results for the conversion of podcast72yvonneklein.mp3 to AIFF, WAV, FLAC and OGG using SOX

	Format				
	MP3 [source]	AIFF	WAV	FLAC	OGG
Duration (hh:mm:ss)	00:06:15:69	00:06:15:69	00:06:15:69	00:06:15.64	
Bit depth	-	16	16	16	
Sample rate	44100	44100	44100	44100	
No. of channels	1	1	1	1	
Sound field	Mono			?	
Sound map location	?	“Unknown”	“Unknown”	?	

Table 17: Analysis results for the conversion of podcast70annthunhurst.mp3 to AIFF, WAV, FLAC and OGG using SOX

The significant properties of an audio recording were correctly recognised by SOX and were maintained when converting Mp3 to AIFF, WAV, OGG and FLAC. However, the tool required the quality level to be manually configured when converting from MP3 to other formats. As noted in other Experiments, there were some minor differences in the analysis results obtained for the bit depth and duration.

3.6.5. Experiment 5: Extract BEXT metadata from Broadcast WAVE using JHOVE

For the fifth experiment we sought to discover if JHOVE was able to extract all descriptive metadata embedded in the extension chunk of a Broacast Wave object. The Broadcast Wave specification allows the inclusion of one or more of several extension chunks, including bext, iXML, qlty (Quality), mext (MPEG audio extension), levl (Peak Envelope), link and axml that each fulfil different objectives within the object. Although many of the metadata elements within the various chunks contain technical metadata that will have little use when the audio recording has been converted to another format, it may also contain text that describes the audio recording and establishes its provenance.

Characterisation
To establish a baseline for the experiment, we selected the four Broadcast Wave files[22] and examined their content using three tools: BWF Chunk Viewer[23], Wav-Info[24] and a hex viewer. The latter was used to establish if there was additional information contained in the BWF files that had not been identified by specialist tools. Although examination of each file in its entirety is time-consuming, it confirmed that there was no additional text-based information.

The analysis identified eight elements - description, originator, originator reference, origination date, origination time, sample count, UMID and coding history - within the ‘bext’ chunk that provided provenance information about the recording, indicating its purpose and origin. Table 18 – 21 indicate

[22] The Broadcast Wave files were obtained from http://www.sr.se/utveckling/tu/bwf/
[23] The BWF Chunk Viewer is a Windows application available at http://www.sr.se/utveckling/tu/bwf/.
[24] Wav-Infois a "Property Sheet Shell Extension available from http://www.softpedia.com/downloadTag/WAV+Info

the metadata contained within the files. A full analysis of the four Broadcast Wave files may be found in Appendix 2.

Element	**Description**
Description:	BWF version one testfile from the European Broadcasting Union and Swedish Radio 2004
Originator:	SR
OriginatorReference:	SESRXLAJHPLAP10WA105837099748726
OriginationDate:	2004-03-05
OriginationTime:	10:58:37
Sample Count	1984500000 since midnight
UMID	See Appendix 3
Coding History:	A=PCM,F=48000,W=16,M=stereo A=PCM,F=48000,W=16,M=stereo, T=D.A.V.I.D. Gmbh Audio Conversion Software V7.00

Table 18: Bext metadata embedded within Short.wav

Element	**Description**
Description:	BWF version one testfile from the European Broadcasting Union and Swedish Radio 2004
Originator:	SR
OriginatorReference:	SESRXLAJHPLAPOWAV090951887248640
OriginationDate:	2004-03-05
OriginationTime:	09:09:51
Sample Count:	1984500000 since midnight
UMID:	See Appendix 2
Coding History:	A=PCM,F=48000,W=16,M=stereo A=MPEG1L2,F=48000,B=192,W=16,M=stereo,T=D.A.V.I.D. GmbH Audio Conversion Software V7.00

Table 19: Bext metadata embedded within Short2.wav

Voice1.wav

Element	**Description**
Description:	BWF version one testfile from The European Broadcasting Union and Swedish Radio 2004
Originator:	SR
OriginatorReference:	TEST file #1
OriginationDate:	2004-03-05
OriginationTime:	10:31:58
Sample Count:	1984500000 since midnight
Version	1
UMID	See Appendix 2
Coding History:	A=PCM,F=48000,W=16,M=stereo A=PCM,F=48000,W=16,M=stereo,T=D.A.V.I.D. GmbH Audio Conversion Software V7.00

Table 20: Bext metadata embedded within Voice1.wav

Voice2.wav

Element	**Description**
Description:	BWF version one testfile from The European Broadcasting Union and Swedish Radio 2004
Originator:	SR
OriginatorReference:	TEST file #2
OriginationDate:	2004-03-04
OriginationTime:	11:12:01
Sample Count:	2018713617 since midnight
UMID	See Appendix 2
Coding History:	A=PCM,F=48000,W=16,M=stereo

	A=MPEG1L2,F=48000,B=192,W=16,M=stereo,T=D.A.V.I.D. GmbH Audio Conversion Software V7.00

Table 21: Bext metadata embedded within Voice2.wav

Format conversion
The conversion of embedded metadata from Broadcast Wave to XML is a two-stage process of text extraction and formatting. To perform the conversion, we tested JHOVE 1.2 (2009-02-10) with Java 6 (Update 7) in a Microsoft Windows XP environment. JHOVE was executed through the Java Platform SE binary. Each of the BWF files were selected in-turn and the results were saved in the JHOVE XML format.

An analysis of each of the XML files indicated a common set of metadata elements had been extracted from the BWF files and stored in a consistent metadata scheme. JHOVE utilised the same element names for the majority of elements (i.e. originatorReference in the BWF file was labelled originatorReference in the JHOVE output) with the exception of ‘Sample Count’ which had been relabelled timeReference.

Table 22 indicates the metadata elements that were extracted by JHOVE.

Metadata element	Extracted by JHOVE
Description:	Y
Originator:	Y
OriginatorReference:	N
OriginationDate:	Y
OriginationTime:	Y
Sample Count:	Y
UMID:	Y
Coding History:	Y

Table 22: bext metadata elements extracted by JHOVE

The OriginatorReference field may be populated by a Unique Source Identifier (USID) which may indicate the country, organization and recording device from which the audio originated, as well as the date and time it was created[25].

3. Conclusion

There were surprisingly few variations in the audio objects when the original and converted audio files were compared. In all of the experiments, the sample rate and number of channels remained the same. There was some minor variation in the duration reported for converted files which differed from that measured or obtained from the source file. However, an examination of the same audio file using different software suggests that the variation was caused by different handling of milliseconds by each tool.

The primary difference between original and converted objects was caused by the encoding algorithm and the capabilities of the container format to store different types of metadata. When converting from a lossless to lossy format or visa versa, it was possible to measure duration, sample rate and no. of channels. However, the variable bit depth of lossy encoding made it difficult to identify if quality loss had occurred. A possible workaround to this issue may be to record the bit depth of each sample and record the highest value.

Recommendations:

- Although a large number of tools are available to analyse audio recordings, they do not provide the level of granularity required to compare significant properties between different formats. It is recommended that the JHOVE (or more likely JHOVE2) is extended to identify and analyse a larger number of audio formats. Alternatively, resources may be allocated to

[25] See EBU Technical Recommendation R99-1999 ‘Unique’ Source Identifier (USID) for use in the OriginatorReference field of the Broadcast Wave Format, available https://www.ebu.ch/CMSimages/en/tec_text_r99-1999_tcm6-4689.pdf

the development of other open source able to analyse each format at a greater level of granularity.

- Recommend that the JHOVE wav-hul plugin is modified to identify and extract the BWF bext 'OriginatorReference' element.

- Recommend that a large collection of audio recordings (accompanied by appropriate representation information) are gathered and made available for use in tool development and analysis.

Appendix 1: Software Tools

The project examined a number of software tools capable of analyzing representation formats used for the storage of emails. To document the process it adopted the format adopted by the CAIRO project for its tool survey[26].

FFIdent

Tool Name	FFIdent
Source URL	http://schmidt.devlib.org/ffident/index.html
Formats supported	Recognition of several formats with support for extensions. MP3 and MIDI recognized in first version at source URL
Technology Base	Java
Operating system	Platform-independent
Dependencies	
License	LGPL
Category	Format identification
Description	FFIdent is a Java library written to identify and extract basic information for various file types. The first version recognizes 27 encoding formats using header information, such as magic number and common structural information (e.g. the <title> tag in an HTML file or hex code in a common location in a binary file).
Output methods	Text or other formats – putput can be configured by the software developer
Notes	FFIdent is, as yet incapable of extracting detailed representation information on the format encoding or significant properties of the information content.

ID3Lib

Tool Name	id3lib
Source URL	http://www.id3lib.org/
Formats supported	ID3v1 and ID3v2 (MP3)
Technology Base	C / C++ / Visual Basic
Operating system	Cross-platform (POSIX-compliant and MS Windows versions available)
Dependencies	
License	LGPL
Category	Metadata extractor Metadata transformer
Description	id3lib is an open-source, cross-platform software development library for reading, writing, and manipulating ID3v1 and ID3v2 tags. ID3 is a metadata container. It is most commonly used in the MP3 audio file format, allowing information such as the title, artist, album, track number and other descriptions to be stored in the file itself.
Output methods	
Notes	The project appears to have been abandoned. The latest release is version 3.8.3, which was uploaded on March 2, 2003.

Java Metadata Collection (JMDC)

Tool Name	Java Metadata Collection (JMDC)
Source URL	http://www.buckazoid.com/jmdc/ http://sourceforge.net/projects/jmdc/
Formats supported	FLAC metadata, Ogg Vorbis (forthcoming)
Technology Base	Java
Operating system	Platform-independent
Dependencies	None

[26] Further details of the format can be found on p11 of the Cairo Tools Survey, located at http://cairo.paradigm.ac.uk/projectdocs/index.html

License	BSD
Category	Metadata extraction
Description	The Java Metadata Collection is a set of Java API's for metadata access and manipulation.
Output methods	
Notes	The project appears to have abandoned. The latest release is version 0.10, which was uploaded on April 5, 2006. A subsequent message labeled May 20, 2006 indicates forthcoming support for Ogg Vorbis. However, no further releases have been made.

JHOVE

Tool Name	JHOVE
Source URL	http://hul.harvard.edu/jhove/
Formats supported	AIFF 1.3, AIFF-C, Microsoft WAVE (PCM, Format Ex, Format Extension)
Technology Base	Java
Operating system	Platform-independent
Dependencies	
License	LGPL
Category	Format identifier, format validator, metadata extractor
Description	
Output methods	Text, XML, HTML
Notes	Funding has been granted to the JHOVE2 project and development work began in 2008.

Kaa Metadata Modules

Tool Name	Kaa Metadata Modules
Source URL	http://doc.freevo.org/2.0/Kaa
Formats supported	ac3, dts, flac, mp3 (with id3 tag support), ogg, pcm, m4a, wma
Technology Base	
Operating system	
Dependencies	libdvdread (optional; for dvd parsing)
License	GPL
Category	Metadata extractor
Description	Kaa modules are based on parts from Freevo and modules created for MeBox. Kaa's modules provide specific media-related functionality, such as retrieving metadata on arbitrary media files (kaa.metadata, previously called mmpython), Python wrappers for Imlib2, Xine, and Evas, and many other high level APIs for easily creating applications that deal with video and audio. The Kaa metadata module can identify limited Representation Information of the encoded object, such as codec, and significant properties of the information object, length, resolution, subtitles, as well as embedded metadata formats.
Output methods	Unknown
Notes	

Lib Extractor

Tool Name	Lib Extractor
Source URL	http://gnunet.org/libextractor
Formats supported	FLAC, MP3 (ID3v1 and ID3v2), Ogg Vorbis, Real Media, WAV (and other format)
Metadata supported	
Technology Base	
Operating system	
Dependencies	
License	
Category	
Description	Libextractor is a library used to extract meta-data from files of arbitrary

	type. It is designed to use helper-libraries to perform the actual extraction, and to be trivially extendable by linking against external extractors for additional file types.
Output methods	Text
Notes	See http://www.linuxjournal.com/article/7552 for a tutorial

NLNZ Metadata Extractor

Tool Name	NLNZ Metadata Extractor
Source URL	http://meta-extractor.sourceforge.net/
Formats supported	MS Wave, MP3 and other formats
Elements recognised	Channels, Bitrate, resolution, time (hours:minutes:seconds), hardware
Technology Base	Java
Operating system	Platform-independent
Dependencies	
License	APL v2
Category	Format identifier, metadata extractor
Description	
Output methods	XML, text
Notes	Format identification is limited to an analysis of the format extension and subsequent treatment of the information as the basis for further analysis. The tool successfully extracted MS Wav data. However, problems were encountered when attempting to extract metadata for MP3s.

XENA

Tool Name	XENA
Source URL	http://xena.sourceforge.net/
Formats supported	AIFF, WAV, FLAC, MP3
Technology Base	
Operating system	
Dependencies	
License	GPL v2
Category	Format conversion
Description	
Output methods	
Notes	

Java Sound API

Tool Name	Java Sound API
Source URL	http://java.sun.com/products/java-media/sound/ http://java.sun.com/j2se/1.5.0/docs/api/javax/sound/sampled/AudioFileFormat.html http://java.sun.com/javase/6/docs/technotes/guides/sound/index.html
Formats supported	Wave, AU. AIFF, AIFF-C, SND
Technology Base	Java
Operating system	Cross-platform
Dependencies	
License	
Category	
Description	
Output methods	
Notes	

MPEG7audioenc

Tool Name	MPEG7audioenc
Source URL	http://mpeg7audioenc.sourceforge.net/

	http://mpeg7audioenc.sf.net (GUI version)
Formats supported	Wav, AU, AIFF, MP3
Technology Base	Java
Operating system	Cross-platform
Dependencies	
License	LGPL
Category	Content encoding, metadata extraction
Description	A Java library that may be used to encode audio and describe its content using descriptors of the MPEG-7 standard.
Metadata supported	No. of audio channels, sample rate, bits per sample, total number of samples, file size and raw information on each sample.
Output methods	XML, MPEG7
Notes	The encoding tool supports other MPEG7 descriptors through the use of an appropriate XSD.

MPEG-7 Low Level Audio Descriptors Extractor

Tool Name	MPEG-7 Low Level Audio Descriptors Extractor
Source URL	http://mpeg7lld.nue.tu-berlin.de/
Formats supported	MP3, WAV
Technology Base	Unknown – online service
Operating system	Web-based interface
Dependencies	
License	
Category	Format conversion, metadata extraction
Description	MPEG-7 Extractor obtains 17 Low Level Descriptors (LLDs) defined within the MPEG-7 standard.
Output methods	MPEG-7
Notes	The online tool places restrictions on the content that can be submitted – file size must be less than 1 MByte for WAV and less than 300 KByte for MP3 and the audio file has to contain only one audio channel.

MPEG-7 Spoken Content Demonstrator

Tool Name	MPEG-7 Spoken Content Demonstrator
Source URL	http://mpeg7spkc.nue.tu-berlin.de/
Formats supported	Wav, MP3
Technology Base	Unknown
Operating system	Web-based
Dependencies	
License	LGPL
Category	Metadata extraction
Description	The demonstration tool extracts an MPEG-7 SpokenContent description from an input speech signal. The MPEG-7 SpokenContent Description Scheme (DS) is a standardized representation of the output of an Automatic Speech Recognition (ASR) system, which is output in an MPEG-7 XML format.
Output methods	MPEG-7 XML
Notes	

Hachoir

Tool Name	Hachoir
Source URL	http://hachoir.org/wiki/hachoir-metadata
Formats supported	AIFF, MPEG1,2,2.5, Real Audio and Sun/NeXT audio
Technology Base	POSIX, GTK interface
Operating system	Linux
Dependencies	
License	

Category	Metadata extraction
Description	Hachoir is a tool to extract metadata from multimedia files (sound, video, archives, etc.)
Metadata supported	Title, album, duration, genre, track number, creator, creation date, producer (software), mime type, endian (little/big), channel (mono, stereo), sample rate, compression, bit rate, format name and version. See http://hachoir.org/wiki/hachoir-metadata/examples
Output methods	Text, other formats (with appropriate scripts)
Notes	The tool uses the following applications: jpeginfo, ogginfo, mkvinfo and mp3info

Meta Track

Tool Name	Meta Track
Source URL	http://projects.gnome.org/tracker/
Description	A tool designed to extract information and metadata about personal data so that it can be searched easily and quickly.
Formats supported	
Technology Base	POSIX
Operating system	Linux
Dependencies	
License	
Category	
Metadata supported	
Output methods	
Notes	

Gnormalize

Tool Name	Gnormalize
Source URL	http://gnormalize.sourceforge.net/
Description	An audio converter and CD ripper with ReplayGain normalization algorithms, a metadata (tag) editor and an audio player. It uses gtk2-perl under GNU/Linux.
Formats supported	MP3, MP4 (M4A or AAC), MPC (MPP or MP+ - Musepack), OGG, APE (Monkey's Audio), FLAC, Audio CD and WAV
Metadata supported	
Output methods	MP3, MP4, MPC, OGG, APE, FLAC and WAV
Category	Format conversion
License	
Technology Base	
Operating system	POSIX
Dependencies	gtk2-perl
Notes	Uses various plug-ins to perform functionality.

OggConvert

Tool Name	OggConvert
Source URL	http://oggconvert.tristanb.net/
Description	A utility for converting audio and video files into the Vorbis audio format (or Theora and Dirac video formats).
Formats supported	MP3, Wav and others
Metadata supported	
Output methods	Ogg
Category	
License	LGPL
Technology Base	
Operating system	Linux-based OS, Windows (experimental)
Dependencies	Python, version 2.4 or newer, GTK+ 2.4 or newer, GStreamer 0.10.11

	(newer versions strongly recommended), The GStreamer "Base" plugin set Python GTK bindings, Python Glade bindings, Python GStreamer binding
Notes	

SoX

Tool Name	SoX
Source URL	http://sourceforge.net/projects/sox/ http://sox.sourceforge.net/
Description	SoX is a sound processing and format conversion tool
Formats supported	Apple/SGI AIFF, SUN .au, PCM, u-law, A-law, MP3 (with optional libmad and libmp3lame libraries), MP4, AAC, AC3, WAVPACK, AMR-NB files (with optional ffmpeg library), Ogg Vorbis (with optional Ogg Vorbis libraries), FLAC files (with optional libFLAC), Microsoft .WAV files PCM, u-law, A-law, MS ADPCM, IMA ADPCM, GSM, RIFX (big endian) and others
Metadata supported	
Output methods	See above
Category	Format conversion, metadata extraction
License	GPL, LGPL
Technology Base	
Operating system	Windows, Linux, MacOS
Dependencies	Format specific plugins
Notes	

Header Investigator

Tool Name	Header Investigator
Source URL	http://railjonrogut.com/HeaderInvestigator.htm
Description	A Windows-based tool that allows the user to display and edit the header of a WAV file. A user can examine the encoding properties of an WAV audio recording and manipulate the header information without changing the audio bitstream itself. This may be useful if the header sample rate is a mismatch to the actual sample rate of the data – which results in the recording being replaced at an incorrect pitch.
Formats supported	Wave and BWF
Metadata supported	Sample rate, No. of channels, Resolution, Bits per sample
Output methods	Wave Header
Category	Metadata edit
License	Unknown
Technology Base	
Operating system	Windows
Dependencies	
Notes	

MetaFlac

Tool Name	MetaFlac
Source URL	http://flac.sourceforge.net/
Description	A command line driven tool to view and edit information embedded within a FLAC audio file
Formats supported	FLAC
Metadata supported	MD5, minimum and maximum block size, minimum and maximum frame size, sample rate, channels, bits per second, total number of samples
Output methods	Screen, user redirection
Category	Metadata extraction, metadata editor
License	GPL
Technology Base	
Operating system	Platform independent
Dependencies	

Notes	

Appendix 2: Audio object description

The project analysed several digital objects during the performance of the case study. The following section outlines the technical and descriptive composition of the four files stored in Broadcast Wave Format.

Short1.wav

Element	Description
Data	
Duration:	12.360 sec
BEXT	
Description:	BWF version one testfile from the European Broadcasting Union and Swedish Radio 2004
Originator:	SR
OriginatorReference:	SESRXLAJHPLAP10WA105837099748726
OriginationDate:	2004-03-05
OriginationTime:	10:58:37
TimeReferenceLow:	76491120
TimeReferenceHigh	0
Sample Count	1984500000 since midnight
Version	1
UMID	Basic UMID (32 bytes) Universal Label: Object Identifier: 0x00 Label Size: 0x00 Designation: ISO 0x00 Designation: SMPTE 0x00 Registry categories: Dictionaries 0x00 Registry categories: Metadata dictionaries 0x00 Structure: Dictionary standard 0x00 Version number: 0x00 Identifiers and locators: 0x00 Globally unique identifiers: 0x00 UMID type: 0x00 Number creation method Material number: 0x00 Instance number: 0x00 Length: 0x00 Instance number 0x00,0x00,0x00 Material Number Time Snap 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00 Random number 0x00,0x00 Machine Node 0x00,0x00,0x00,0x00,0x00,0x00
Coding History:	A=PCM,F=48000,W=16,M=stereo A=PCM,F=48000,W=16,M=stereo, T=D.A.V.I.D. Gmbh Audio Conversion Software V7.00
Unidentified tag	1
Format	
wFormatTag	1
nChannels	2
nSamplesPerSec	48000
nAvgBytesPerSample:	192000
nBlockAlign:	4
wBitsPerSample:	16
cbSize	7
FACT	
dwSampleLength:	593280
Labl	

	No labl information present.

Short2.wav

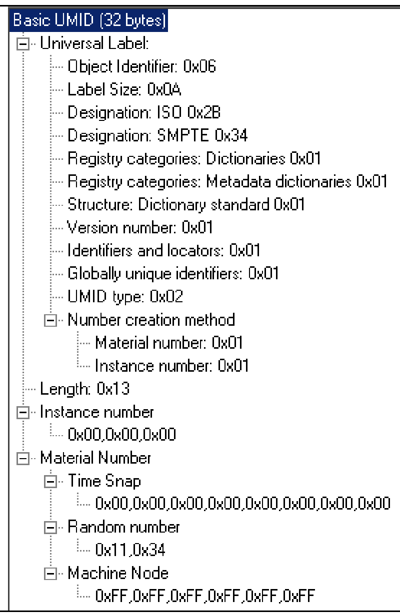
Element	**Description**
Data	
Duration:	12.336 sec duration
BEXT	
Description:	BWF version one testfile from the European Broadcasting Union and Swedish Radio 2004
Originator:	SR
OriginatorReference:	SESRXLAJHPLAPOWAV090951887248640
OriginationDate:	2004-03-05
OriginationTime:	09:09:51
TimeReferenceLow:	76491120
TimeReferenceHigh	0
Sample Count:	1984500000 since midnight
Version	1
UMID	Basic UMID (32 bytes) Universal Label: Object Identifier: 0x00 Label Size: 0x00 Designation: ISO 0x00 Designation: SMPTE 0x00 Registry categories: Dictionaries 0x00 Registry categories: Metadata dictionaries 0x00 Structure: Dictionary standard 0x00 Version number: 0x00 Identifiers and locators: 0x00 Globally unique identifiers: 0x00 UMID type: 0x00 Number creation method Material number: 0x00 Instance number: 0x00 Length: 0x00 Instance number 0x00,0x00,0x00 Material Number Time Snap 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00 Random number 0x00,0x00 Machine Node 0x00,0x00,0x00,0x00,0x00,0x00
Coding History:	A=PCM,F=48000,W=16,M=stereo A=MPEG1L2,F=48000,B=192,W=16,M=stereo,T=D.A.V.I.D. GmbH Audio Conversion Software V7.00
Unidentified tag	1
Format	
wFormatTag	80 (Wave_Format_MPEG)
nChannels	2
nSamplesPerSec	48000
nAvgBytesPerSample:	48000
nBlockAlign:	1152
wBitsPerSample:	0
cbSize:	22
fwHeadLayer:	2
dwHeadBitRate:	384000
fwHeadMode:	1
fwHeadModeExt:	0
wHeadEmphasis:	1
fwHeadFlags	16
dwPTSlow:	0
dwPTSHigh:	0

MEXT	
SoundInformation:	3
FrameSize (Bytes per frame):	1152
AncillaryDataLength:	5
AncillaryDataDef:	7
Reserved:	0
FACT	
dwSampleLength:	592128

Voice1.wav

Element	Description
Data	
Duration:	1 min 38.853 sec
BEXT	
Description:	BWF version one testfile from The European Broadcasting Union and Swedish Radio 2004
Originator:	SR
OriginatorReference:	TEST file #1
OriginationDate:	2004-03-05
OriginationTime:	10:31:58
TimeReferenceLow:	76491120
TimeReferenceHigh	0
Sample Count:	1984500000 since midnight
Version	1
UMID	Basic UMID (32 bytes) Universal Label: Object Identifier: 0x06 Label Size: 0x0A Designation: ISO 0x2B Designation: SMPTE 0x34 Registry categories: Dictionaries 0x01 Registry categories: Metadata dictionaries 0x01 Structure: Dictionary standard 0x01 Version number: 0x01 Identifiers and locators: 0x01 Globally unique identifiers: 0x01 UMID type: 0x02 Number creation method Material number: 0x01 Instance number: 0x01 Length: 0x13 Instance number 0x00,0x00,0x00 Material Number Time Snap 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00 Random number 0x12,0x34 Machine Node 0xFF,0xFF,0xFF,0xFF,0xFF,0xFF
Coding History:	A=PCM,F=48000,W=16,M=stereo A=PCM,F=48000,W=16,M=stereo,T=D.A.V.I.D. GmbH Audio Conversion Software V7.00
Unidentified tag	1
Format	
wFormatTag	1 (Wave_Format_PCM)
nChannels	2
nSamplesPerSec	48000
nAvgBytesPerSample:	192000
nBlockAlign:	4
wBitsPerSample:	16
cbSize:	17

Voice2.wav

Element	Description
Data	
Duration:	1 min 41.088 sec
BEXT	
Description:	BWF version one testfile from The European Broadcasting Union and Swedish Radio 2004
Originator:	SR
OriginatorReference:	TEST file #2
OriginationDate:	2004-03-04
OriginationTime:	11:12:01
TimeReferenceLow:	78532011
TimeReferenceHigh	0
Sample Count:	2018713617 since midnight
Version	1
UMID	
Coding History:	A=PCM,F=48000,W=16,M=stereo A=MPEG1L2,F=48000,B=192,W=16,M=stereo,T=D.A.V.I.D. GmbH Audio Conversion Software V7.00
Unidentified tag	1
Format	
wFormatTag	8 (wave_format_mpeg)
nChannels	2
nSamplesPerSec	480000
nAvgBytesPerSample:	480000
nBlockAlign:	1152
wBitsPerSample:	0
cbSize:	22
fwHeadLayer:	2
dwHeadBitRate:	384000
fwHeadMode:	1
fwHeadModeExt:	0
wHeadEmphasis:	1
fwHeadFlags	16
dwPTSlow:	0
dwPTSHigh:	0
MEXT	
SoundInformation:	3

FrameSize (Bytes per frame):	1152
AncillaryDataLength:	5
AncillaryDataDef:	7
Reserved:	0
FACT	
dwSampleLength:	4852224
Levl	
	No Levl information recorded
Aux	
Unidentified tag:	“t”

Appendix 3: Metadata elements contained in the Bext chunk of Microsoft Broadcast Wave

No	Name	Definition	Function classification	Function description	Significance summary	Applicability
	wFormatTag	A number indicating the WAVE format category of the file	Representation Information	A characteristic of the encoding format. The content of the <format-specific-fields> portion of the fmt chunk, and the interpretation of the waveform data by processing software depend on this value.	N	BWF, WAV
	nchannels	A numeric value that indicates the number of distinct streams within an audio object.	Content	A comparison of a source and destination object that identifies a reduction in the channel number may indicate quality loss (one or more channels has been lost) or reduction (the channels have been merged and can no longer be treated independently.	Y	All
	Average number of bytes per second	The average number of bytes per second at which the waveform data should be transferred.	Representation Information	The value is influenced by the type of encoding format in use. It may be used by playback software can estimate the buffer size using this value.	N	All
	Block alignment	The minimum atomic unit of data. E.g. the number of bytes used by a single sample.	Representation Information	A characteristic of the encoding format	N	PCM
1	DWORD ckSize	Indicates the size of the extension chunk within the file	Representation Information	ckSize is used for internal validation of the file. It is not considered to be a significant property due to the likelihood that the information will change when new	N	BWF

				information is added or become superfluous when converting to other formats. It might be considered a type of Representation Information.		
2	Description	An ASCII string that contains a free text description of the sound sequence.	Context	If completed, it may provide qualitative information that establishes the provenance of the audio recording.	Y	BWF
3	Originator	An ASCII string that may contain the name of the creator of the audio.	Context	If completed, it may provide qualitative information that establishes the provenance of the audio recording.	Y	BWF
4	OriginatorReference	An ASCII string that contains a non ambiguous reference allocated by the originating organization[27]	Context	If completed, it may provide qualitative information that establishes the provenance of the audio recording.	Y	BWF
5	OriginationDate	Indicates the creation date of the audio sequence. Format is YYYY-MM-DD	Context	If completed it may establish the creation point of the original recording, which is useful for establishing its provenance. However, it may be considered unnecessary for some users if it indicates the date at which a digital manifestation of an analogue original was created.	Y	BWF
6	OriginationTime	The time that the audio sequence was created. Format is hh-mm-ss.	Context	If completed it may establish the creation point of the original recording, which is useful for establishing its provenance. However, it may be considered unnecessary for some users if it indicates	Y	BWF

[27] See EBU Technical Recommendation R99-1999. 'Unique' Source identifier (USID) for use in the OriginatorReference field of the Broadcast Wave Format. https://www.ebu.ch/CMSimages/en/tec_text_r99-1999_tcm6-4689.pdf

				the date at which a digital manifestation of an analogue original was created.		
7	DWORD TimeReferenceLow	First sample count since midnight low word	Representation Information	TimeReferenceLow is used for decoding of BWF sound recording. may be used to calculate the length of a recording may dividing the TimeReferenceLow decimal value by the sampling rate.	N	BWF
8	DWORD TimeReferenceHigh	First sample count since midnight, high word	Representation Information	TimeReferenceHigh is used for decoding the audio in appropriate processing software. It is considered to be out of scope.	N	BWF
9	WORD Version	An unsigned binary number that indicates the BWF version.	Representation Information	A characteristic of the encoding format. May be beneficial when decoding the file and embedded data streams.	N	BWF
10	Unique Material Identifier	A unique identifier that conforms to the to the SPMTE 330M standard assigned to audiovisual content	System-wide identifier	An identifier may change between different manifestations of an	N	MXF, BWF, AAF

[28] See the Digital Preservation Europe briefing paper on UMID for further information.
http://www.digitalpreservationeurope.eu/publications/briefs/UMID_Unique%20Material%20Identifier.pdf

	(UMID)			Information Object[28].		
	Coding History[29]	An ASCII text field of non-restricted length that may be used to describe the encoding process applied to each manifestation of the Information Object. Each entry is terminated by a Carriage Return+Line Feed. Recommendations for a Coding History format are provided in EBU Recommendation R98-1999	Context: provenance	The field may be beneficial for curators who wish to understand the activities performed on a digital object, particularly if it has been provided by a third-party and no other information is available. However, the Coding History is not essential for the performance of the audio recording or understanding the context of its creation.	Y, for curatorial use	BWF
	Quality Report[30]	A text field that may be used to describe events that affect the quality of the recording sound signal. Each event is listed with details of the type of event, exact time stamps, priority, event status and other quality parameters.	Context: provenance	The field may be beneficial for curators who wish to understand the activities performed on a digital object, particularly if it has been provided by a third-party and no other information is available. However, the Coding History is not essential for the performance of the audio recording or understanding the context of its creation.	Y, for curatorial use.	BWF
	Cue Sheet	A list that identifies one or more events within the sound recording, e.g. the beginning of an aria or the starting point of an important speech. Each event is recorded using a time code and description	Context: provenance	The field may provide contextual information that enables the listener to identify specific components of the audio stream.	Y	BWF

[29] See European Broadcasting Union (199). EBU Technical Recommendation R98-1999. Format for the <CodingHistory> field in Broadcast Wave Format files, BWF, available at www.ebu.ch/CMSimages/en/tec_text_r98-1999_tcm6-4709.pdf

[30] Coding History, Quality Report and Cue Sheet are described in detail in European Broadcasting Union (2001). BWF Summary 2 - Capture Report, available at http://www.ebu.ch/CMSimages/en/tec_doc_t3285_s2_tcm6-10482.pdf

References

Anon (n.d). Draft AES Object schema. Retrieved on March 29, 2009 from: http://hul.harvard.edu/ois/xml/xsd/drs/audioObject.xsd

Anon (n.d.). Lame MP3 Encoder. Retrieved on March 29, 2009 from: http://lame.sourceforge.net/

Anon (2003). AudioMD Extension Schema Data Dictionary. Retrieved on March 29, 2009 from: http://www.loc.gov/rr/mopic/avprot/DD_AMD.html

Anon (2005). File formats. Retrieved on March 29, 2009 from: http://www.magicdb.org/

Anon (2005). User Guide to PBCore - Public Broadcasting Metadata Dictionary. Retrieved on March 29, 2009 from: http://www.pbcore.org/PBCore/UserGuide.html

Anon (2008). DRS METS Archive Tool (Dmart) for Audio Deposit. Retrieved on March 29, 2009 from: http://hul.harvard.edu/ois/systems/drs/dmart/current/

Anon (2008). Man page of SoX. Retrieved on March 29, 2009 from: http://sox.sourceforge.net/soxformat.html

Anon (2009-02-25) JHOVE - JSTOR/Harvard Object Validation Environment. Retrieved on March 29, 2009 from: http://hul.harvard.edu/jhove/

Anon (2009-03-27). AESSC: Project Status. Retrieved on March 29, 2009 from: http://www.aes.org/standards/b_policies/project-status.cfm

Casey, M. & Gordon, B. (2007). Sound Directions: Best Practices for Audio Preservation. Retrieved on March 29, 2009 from: http://www.dlib.indiana.edu/projects/sounddirections/papersPresent/index.shtml

CDP Digital Audio Working Group (2005). Digital Audio Best Practices. Version 2.0. Retrieved on March 29, 2009 from: http://www.bcr.org/cdp/best/digital-audio-bp.pdf

Coalson, J. (n.d.). FLAC - Free Lossless Audio Codec. Retrieved on March 29, 2009 from: http://flac.sourceforge.net/

Council on Library and Information Resources & and Library of Congress (2006). Capturing Analog Sound for Digital Preservation. Retrieved on March 29, 2009 from: http://www.clir.org/pubs/abstract/pub137abst.html

European Broadcasting Union (1997). Specification of the Broadcast Wave Format: A format for audio data files in broadcasting - Supplement 1 – MPEG audio. Retrieved on March 29, 2009 from: http://www.ebu.ch/CMSimages/en/tec_doc_t3285_s1_tcm6-10545.pdf

European Broadcast Union (1999). EBU Technical Recommendation R99-1999. ‘Unique’ Source Identifier (USID) for use in the OriginatorReference field of the Broadcast Wave Format. Retrieved on March 29, 2009 from: https://www.ebu.ch/CMSimages/en/tec_text_r99-1999_tcm6-4689.pdf

European Broadcasting Union (1999). EBU Technical Recommendation R98-1999. Format for the <CodingHistory> field in Broadcast Wave Format files, BWF. Retrieved on March 29, 2009 from: http://www.ebu.ch/CMSimages/en/tec_text_r98-1999_tcm6-4709.pdf

European Broadcasting Union (2001). EBU Tech 3285-2001 BWF - a format for audio data files in broadcasting V.1. Retrieved on March 29, 2009 from: http://www.ebu.ch/CMSimages/en/tec_doc_t3285_tcm6-10544.pdf

European Broadcasting Union (2001). BWF - A format for audio data files in broadcasting: Supplement 2 - Capturing Report. Retrieved on March 29, 2009 from: http://www.ebu.ch/CMSimages/en/tec_doc_t3285_s2_tcm6-10482.pdf

European Broadcasting Union (2001). BWF Summary 2 - Capture Report. Retrieved on March 29, 2009 from: http://www.ebu.ch/CMSimages/en/tec_doc_t3285_s2_tcm6-10482.pdf

Harvard University Library (2004). Administrative Metadata for Digital Audio Files. Retrieved on March 29, 2009 from: http://preserve.harvard.edu/resources/audiometadata.pdf

Knight. G & McHugh, J (2005). Preservation Handbook: Digital Audio. Retrieved on March 29, 2009 from: http://ahds.ac.uk/preservation/audio-preservation-handbook.pdf

Knight, G. (2008). Framework for the definition of significant propertiest. Retrieved on March 29, 2009 from: http://www.significantproperties.org.uk/outputs.html

Obrenovic, Z. Burger, T. Popolizio, P. & Troncy, R (2007). Multimedia Semantics: Overview of Relevant Tools and Resources. Retrieved on March 29, 2009 from: http://www.w3.org/2005/Incubator/mmsem/wiki/Tools_and_Resources

O'Neill, D. (2006-12-17). Lyrics3v2. Retrieved on March 29, 2009 from: http://www.id3.org/Lyrics3v2

Underbit Technologies (n.d.). MAD – MPEG Audio Decoder. Retrieved on March 29, 2009 from: http://www.underbit.com/products/mad/