ARSC TECHNICAL COMMITTEE

A Study of Embedded Metadata Support in Audio Recording Software: Summary of Findings and Conclusions

October, 2011

Authored by:
Chris Lacinak, AudioVisual Preservation Solutions
Walter Forsberg, Graduate, Moving Image Archiving and Preservation Masters Program, New York University

Edited by:
Mike Casey, Media Preservation Initiative, Indiana University
Marcos Sueiro Bal, WNYC Radio

Study performed by:
Chris Lacinak, AudioVisual Preservation Solutions
Peter Oleksik, AudioVisual Preservation Solutions
Walter Forsberg, Graduate, Moving Image Archiving and Preservation Masters Program, New York University

With contributions from:
Tom Endres, BMS/Chace
Tommy Sjöberg, Folkmusikens hus
Bruce Gordon, Harvard University
Preston Cabe, George Blood Audio & Video

The study design and this report were reviewed by the ARSC Technical Committee which consisted of the following members:

Mike Casey, Co-Chair
Adrian Cosentini, Co-Chair
Peter Alyea
Marcos Sueiro Bal
George Blood
Bruce Gordon
Gary Galo
Richard L. Hess
Eric Jacobs
Bill Klinger
Chris Lacinak
Brad McCoy
John Narus
Doug Pomeroy
Marie O'Connell
Dennis Rooney
Tommy Sjöberg
John Spencer
Konrad Strauss
Charles Van Winkle
Seth Winner

Table of Contents

Executive Summary

In 2010, the ARSC Technical Committee (TC) completed a study of support for embedded metadata within and across a variety of audio recording software applications. The study was coordinated, and much of it carried out, by AudioVisual Preservation Solutions. This work addressed two primary questions: How well does embedded metadata persist, and is its integrity maintained, within any given file as it is handled by various applications over time? How well is embedded metadata handled during the process of creating a derivative?

The term 'embedded metadata' can be defined as metadata that is stored inside the same file, or container, that also stores the essence or target content to which the metadata refers. Embedding metadata in audio, video, and still image files is now widely recognized as an essential strategy for managing, handling, interpreting, preserving, and accessing media holdings. For example, end-users rely on embedded metadata for search and retrieval of audiovisual content within computer operating systems and in applications such as iTunes, producers require it for management of media and workflows in the production process, and archivists utilize it to administer collections of content.

Three tests were designed for this study. They were developed to evaluate interoperability and semantic shifts, persistence and integrity through editing operations, and persistence and integrity through derivative creation.

Results from this study demonstrate that few of the standard metadata chunks are supported in their entirety by any software application. Rather, applications tend to display and provide access to selected fields of their choosing from each chunk standard. In general, the Broadcast Wave Format `bext` chunk is the most widely supported, followed by selected fields within the `LIST-INFO` chunk. Least supported were the XML-based chunks—there was some support for selected fields within `iXML`, but no support for `axml`, while `XMP` was only supported by its creator, Adobe.

Most troubling are the findings associated with application metadata management, where it seems to be the rule rather than the exception for applications to automatically *erase* chunks and fields that they do not support after common user actions such as metadata or audio editing are performed. Embedded metadata does not persist nor is its integrity maintained consistently across the audio software applications studied.

These issues have major implications for the use of embedded metadata over time and across workflows. The findings of this study raise serious concerns, particularly for the archiving and preservation communities who rely on embedded metadata for interpretation and management of digital files representing preserved content into the future.

The ARSC Technical Committee recommends further action to not only raise awareness but advocate for greater interoperability as well as development of an interchange environment

that supports the operating principles of persistence and integrity of embedded metadata. These issues influence the selection of hardware and software systems in organizations concerned with preservation and archiving, which includes a growing number of corporations, government, non-profits, and academic institutions. Additionally, the increasing integration of production, post-production, and archive workflows requiring embedded metadata means that these issues impact a sizeable number of users with economic influence that should be of interest to system manufacturers.

NOTE: The applications tested in this study were selected based on what was readily available to participating members of the ARSC TC. This document is not intended, nor should it be used, as a guide for software selection. Between testing and publication of this report, several of these applications will see new releases that may alter the referenced behaviors. This study provides a snapshot of embedded metadata functionality at a particular point in time to raise awareness of general issues relating to persistence and integrity.

I. Introduction

The term 'embedded metadata' can be defined as metadata that is stored inside the same file, or container, that also stores the essence or target content to which the metadata refers. Embedding metadata in audio, video, and still image files is now widely recognized as an essential strategy for managing, handling, preserving and accessing media. In 2009, the Federal Agencies Digitization Guidelines Initiative's Audio Visual Working Group published a document titled *Embedding Metadata in Digital Audio Files*[1] that explored this topic within the context of archives and endorsed its use. The following is an excerpt from page 2 of the referenced document:

> **Why embed metadata?**
> Embedded metadata can provide information to and support functionality for various persons and systems at a variety of points in the content life cycle. For example, it can help the digitizing unit or organization as it produces and preserves content. It can serve persons or systems who receive content that is disseminated by the digitizing unit or organization. Some metadata elements are especially valuable to internal actors, some to external, and some to both.
>
> Embedded metadata, of course, is rarely an agency's only metadata. In most archiving and preservation programs, workflow and archiving are supported by one or more databases, cataloging systems, finding aids, and the like, each of which contains metadata.

The field of photography is a paramount example of collective awareness and advocacy amongst practitioners using embedded metadata.[2] [3] Since the late-1990s, photographers and imaging professionals have worked together to establish and widely adopt the International Press Telecommunications Council (IPTC) Photo Metadata Standard, a robust standard enabling rich sets of embedded metadata to be leveraged within workflows and applications that handle

[1] http://www.digitizationguidelines.gov/audio-visual/documents/Embed_Intro_090915.pdf
[2] http://www.metadataworkinggroup.org/
[3] http://www.photometadata.org

digital images. In the audio industry, adoption of embedded metadata has taken a different path. The uptake of MP3 as the predominant format for audio distribution and online access demanded a mechanism that enabled metadata to travel with the files as a means for identification of artists, albums, song titles, etc. This led to the development of what are known as 'ID3 tags' for embedding metadata inside of MP3 files. However, neither MP3 nor this set of metadata proved sufficiently robust for archiving and preservation, and wide adoption in the end user community did not carry over into typical professional workflows for quite some time.

While the ability to embed metadata in WAVE audio files has existed for many years via the `LIST-INFO` chunk, adoption has been extremely limited. Initially published in 1997,[4] the Broadcast Wave Format (BWF) standard extended the data architecture of WAVE files by adding the `bext` (Broadcast Extension) chunk, which accommodates the minimum set of metadata considered necessary for broadcast applications. BWF subsequently became the de facto standard for audio archiving and preservation best practices, recommended in publications from organizations such as ARSC,[5] the International Association of Sound and Audiovisual Archives (IASA),[6] and the Sound Directions Project.[7]

As audio metadata awareness and practices continue to evolve, organizations concerned with archiving, preservation, and access are finding that the `LIST-INFO` and `bext` chunks are not sufficient. This led to development of the `axml` chunk[8] as a supplement to BWF, standardized in 2003, with the expressed intention of facilitating the storage of any valid XML data (i.e. schema agnostic). This differs from the `bext` chunk, which contains a prescribed set of fields. `axml` development was seen as particularly valuable for embedding metadata defined in standards such as EBU 3293[9]—known as the 'EBU Core' metadata set, and EBU 3295—known as 'P_Meta' (the former is Dublin Core-based and archive-centric, though explicitly stated for use beyond archives).

In addition to the introduction of the `axml` chunk, the `iXML` chunk was established through a consortium of manufacturers

> to be used as a RIFF (embedded tagged data) chunk inside a Broadcast Wave file (although it can be optionally included in other file types), to supersede the metadata currently written in the standard Broadcast Wave 'bext' chunk description field in a non-standardised way by several manufacturers. `iXML` is intended to offer a standardised specification to communicate all information currently in use, and to provide an extensible framework for manufacturers to add new private, or public data, and for the specification to expand in a completely forward and backwards compatible manner.[10]

---

4 http://tech.ebu.ch/publications/tech3285
5 http://www.arsc-audio.org/pdf/ARSCTC_preservation.pdf
6 IASA-TC04, 2nd Edition, Guidelines on the Production and Preservation of Digital Audio Objects, March 2009 found at http://www.iasa-web.org/tc04/audio-preservation
7 http://www.dlib.indiana.edu/projects/sounddirections/papersPresent/index.shtml
8 http://tech.ebu.ch/publications/tech3285s5
9 http://tech.ebu.ch/lang/en/MetadataSpecifications
10 http://www.gallery.co.uk/ixml/introduction.html from the main page at http://www.ixml.info/

At the time of this writing, the Audio Engineering Society (AES) is developing `iXML` as a standard.[11] To date, `iXML` has seen adoption in both consumer and professional level field recorders and other recording equipment. Similar to `bext`, `iXML` is both a prescribed set of fields (expressed as XML), and a chunk.

At the same time, `XMP`, which is a standard originally developed by Adobe for digital images, is being advanced as an ISO standard.[12] Aside from its use in digital images, Adobe created a chunk within WAVE files, labeled '`_PMX`' which is intended to store `XMP` data. `XMP` has a number of defined schemas that can be used, but it is also extensible.

These standards collectively provide the capability to embed sufficient metadata to meet the needs of various applications and organizations, including archives. However, the archive community recognizes that metadata support varies dramatically from application to application. In addition, drastically different operating principles may be used across applications, or even within versions of the same application. Of primary interest to archives are the following questions:

1. How well does embedded metadata persist, and is its integrity maintained, within any given file as it is handled by various applications over time?

2. How well is embedded metadata handled during the process of creating a derivative?

This study was undertaken to answer these questions, quantifying variations while identifying issues and problems. Ultimately, our goal is not just to raise awareness but to advocate for greater interoperability and the development of an interchange environment that supports the operating principles of persistence and integrity of embedded metadata. Increased awareness and documentation of these issues is likely to influence the selection of hardware and software systems in organizations concerned with preservation and archiving, which is a dramatically growing number encompassing corporations, government, non-profits, and academic institutions. Combined with the increasing integration of production, post-production, and archive workflows requiring embedded metadata, this means that these issues impact a sizable number of users.

With this in mind, a byproduct of this study is the creation of freely available and documented test methods for practitioners to use in the evaluation of hardware and software system metadata management.

---

[11] AES-X155: (D155) AES standard for audio metadata - Production recording metadata set (iXML)
[12] ISO/DIS 16684-1 XMP specification -- Part 1: Data model, serialization and core properties

II. Test Description

Three tests were used in this study. They were designed as follows:

1. Interoperability and Semantic Shifts: This evaluation uses a reference WAVE file containing extensive embedded metadata including the `bext`, `LIST-INFO`, `axml`, `XMP` and `iXML` chunks. The reference WAVE file is opened in a number of software applications to evaluate which fields are displayed in the application's interface. In addition to this, any semantic shifts are documented. Semantic shifts are defined here as the displaying of a field's value under a field name which differs from the intent of the original field. As an example: if an application presents the `LIST-INFO` chunk field '`itch`'(the technician who digitized the audio) as a field labeled '`artist`', this would be considered a semantic shift. It should be emphasized that this test analyzes metadata display only.

2. Persistence and Integrity Through Editing Operations: A four-part test, this evaluation considers how various applications handle embedded metadata when basic metadata and audio editing operations are performed and the file is saved. The first two sub-tests analyze the results of editing and adding embedded metadata and saving the file. The third sub-test analyzes the results of performing an audio edit and saving the file. The final sub-test analyzes the results of simply performing the '`save as`' function. The primary focus for evaluation in all tests is identifying whether existing metadata persists unaltered.
    a. Impact of Editing Existing Chunks: This test analyzes the results of editing embedded metadata and saving the file.
    b. Impact on Existing Chunks When Creating New Chunks: The test analyzes the impact to existing chunks in a file when new chunks are added and the file is saved.
    c. Impact on Metadata of Audio-Only Editing: This test analyzes the results of performing an audio edit and saving the file.
    d. Impact on Metadata of '`Save As`' Function: This test analyzes the results of performing the '`save as`' function.

3. Persistence and Integrity Through Derivative Creation: This evaluation tests how various applications handle embedded metadata, when creating a derivative file from a WAVE file. Target derivative file formats tested include: MP3, FLAC, and WAVE.

A test method was drafted for each test and submitted to the ARSC TC for review and feedback. The test methods used are available at:
http://www.avpreserve.com/wp-content/uploads/2011/10/ARSC_Metadata_Test_Method.pdf

An associated set of reference files were created for these tests, containing variations of populated `bext`, `LIST-INFO`, `XMP`, `iXML`, and `axml` chunks. These files were created using BWF MetaEdit[13] and are available at:
http://www.avpreserve.com/wp-content/uploads/2011/10/ARSC_BWF_Metadata_Test1.zip
http://www.avpreserve.com/wp-content/uploads/2011/10/ARSC_BWF_Metadata_Test2.zip
http://www.avpreserve.com/wp-content/uploads/2011/10/ARSC_BWF_Metadata_Test3.zip

The applications tested in this study were:

- WaveLab v. 6.10 on Windows XP Professional
- Adobe Audition v. 3.0 on Windows XP Professional
- Audacity v. 1.3.11 on Mac OSX 10.5.8
- BIAS Peak v. 5.2.1 and 6.2.0 on Mac OSX 10.4.11 and Mac OSX 10.6.2 respectively
- Avid Pro Tools LE v. 8.0.1 on Mac OSX 10.5.8
- Apple iTunes v. 9.0.2 on Mac OSX 10.5.8
- Steinberg Nuendo v. 4.10 on 10.4.11

Wave files produced or altered in tests 2 and 3 were analyzed using BWF MetaEdit to evaluate the persistence and integrity of embedded metadata. For FLAC and MP3 files produced in test 3 Phil Harvey's ExifTool was used, employing the Verbose function (command line: '`exiftool –V5 [file location]`') to inspect which embedded metadata persisted through the derivative-making process.

[13] An open source, freely available tool developed by AudioVisual Preservation Solutions for the Federal Agencies Digitization Audio-Visual Working Group, Available at http://sourceforge.net/projects/bwfmetaedit/

III. Test Result Summary

Test 1 – Display of Embedded Metadata in Application Interfaces

Figure 1 is a simplified version of the findings, identifying which applications display fields from the identified chunks in this study.

Chunk	Metadata Field/Tag	Wavelab 6.10	Audition 3.0	Audacity 1.3.11	Peak 5.1.2	Peak 6.0	ProTools LE 8.0.1	iTunes 9.0.2	Nuendo 4.0
BEXT	Description	√	√	x	x	√	√	x	√
BEXT	Originator	√	√	x	x	√	x	x	√
BEXT	OriginatorReference	√	√	x	x	√	~	x	√
BEXT	OriginationDate	√	√	x	x	√	~	x	√
BEXT	TimeReference (translated)	√	√	x	x	√	~	x	~
BEXT	TimeReference	√	√	x	x	x	x	x	x
BEXT	UMID	x	√	x	x	√	x	x	√
BEXT	Coding History	√	√	x	x	x	x	x	√
LIST-INFO	IARL	x	x	x	x	x	x	x	x
LIST-INFO	IART	√	√	√	x	√	x	x	x
LIST-INFO	ICMS	x	x	x	x	x	x	x	x
LIST-INFO	ICMT	√	√	√	x	√	x	x	x
LIST-INFO	ICOP	√	√	√	x	√	x	x	x
LIST-INFO	ICRD	√	√	√	x	x	x	x	x
LIST-INFO	IENG	√	√	x	x	x	x	x	x
LIST-INFO	INGR	√	√	x	x	√	x	x	x
LIST-INFO	IKEY	√	√	x	x	x	x	x	x
LIST-INFO	IMED	x	√	x	x	x	x	x	x
LIST-INFO	INAM	√	√	√	x	√	x	x	x
LIST-INFO	IPRD	x	√	x	x	x	x	x	x
LIST-INFO	ISBJ	√	√	x	x	x	x	x	x
LIST-INFO	ISFT	√	√	√	x	x	x	x	x
LIST-INFO	ISRC	x	√	x	x	x	x	x	x
LIST-INFO	ISRF	√	√	x	x	x	x	x	x
LIST-INFO	ITCH	x	√	x	x	x	x	x	x
XMP	All	x	√	x	x	x	x	x	x
axml	All	x	x	x	x	x	x	x	x
iXML	All	x	x	x	x	x	x	x	x

Figure 1: This is a simplified chart demonstrating which applications display fields from the `bext`, `LIST-INFO`, `axml` and `iXML` chunks in their interface. A green '√' indicates that the application displays the associated field. A red 'x' indicates that it does not. A yellow '~' indicates an approximation which is discussed below.

Figure 1 and detailed semantic findings can be downloaded at:
http://www.avpreserve.com/wp-content/uploads/2011/10/ARSC_BWF_Metadata_Test1_Results.zip

Generally speaking there is wildly varying support for display of bext and LIST-INFO fields. Of the software tested in this study, Adobe Audition offers the most comprehensive display of metadata. Since Adobe is the creator of XMP, it is not surprising that Audition displays this metadata, but it is the only application in this study that does. Display of axml and iXML are not supported by any of the applications tested.

In reviewing semantics there were no significant variances in the field labels used in the application interfaces compared to the intended meaning of the associated field. Adobe Audition's interface offers an interesting exception in this regard based on the sophistication of its metadata management, which performs mapping functions between bext and LIST-INFO to XMP and also maps between prospective formats such as MP3 and WAVE. This results in some semantic shifts associated with the mappings, although this would be true of any application that had this level of sophistication. The documentation on semantic findings demonstrates some of the mappings available within Audition.

Pro Tools presented the most interesting scenario in this test. The test files created for this study were stereo interleaved files but Pro Tools only supports mono files. Upon opening, Pro Tools split each stereo interleaved test file into two single mono files. Further testing should be performed using mono reference files, but this enabled an unanticipated opportunity to review how Pro Tools manages embedded metadata in the process of splitting a stereo file. The results of this review include the following:

- The axml, XMP and LIST-INFO chunks were removed completely.
- The iXML chunk persisted in the files but was significantly altered. The following fields and values were maintained: <take>, <scene>, <tape>, <speed>, <timecode_flag>, <timecode_rate>, <ubits>, <project>, <circled>, <track_list>, <track_count>, <channel_index>, <interleave_index>, and track <name>. However the original iXML <note> value was replaced with the value embedded in the bext 'description' field. Within the data maintained, Pro Tools mapped channel specific data appropriately – parsing the left channel data to the file representing the left channel and the right channel data to the file representing the right channel.
- Some of the bext fields were altered or replaced completely.
    - The 'description' field is labeled in the Pro Tools interface as 'file comment.' The original value for 'description' was augmented with iXML fields that Pro Tools automatically mapped there. These were <ubits>, <scene>, <take>, <tape>, <framerate>, <speed>, and track <name>.
    - The 'UMID' value was erased without a replacement value. Interestingly, Pro Tools created a new chunk in the file labeled umid.
    - 'Origination Date' was displayed and the original value was maintained despite the fact that new files were created in this process.
    - 'Originator Reference' was labeled 'Unique ID' in the Pro Tools interface and again was displayed, but any existing value was replaced with a completely different value generated by Pro Tools in the import process.
- Pro Tools automatically embedded the following chunks: minf, elm1, regn, umid and DGDA.

Nuendo also presented a couple of functional and semantic issues:

- Nuendo displayed `bext` fields at the file level but access for editing and entry into `bext` fields appeared to be available only with the fields '`description`', '`Originator`' (displayed as '`Author`'), and '`Originator Reference`' (displayed as '`Reference`'), and only at the global level within the application preferences. In other words, entering values into these fields within the preferences embedded them into all files created.
- Nuendo displayed '`Time Reference`' as '`Timecode,`' which is a bit different semantically. The intent of '`Time Reference`' is to indicate the time stamp of the first sample of audio. Timecode is generally continuous and not a single timestamp.

Test 2: Persistence and Integrity Through Editing Operations

Whereas Test 1 focused on display of fields within applications, Test 2 focused on what happens to the metadata in the file itself after editing, creating new chunks, or using the '`Save As`' function. Detailed findings can be downloaded at:
http://www.avpreserve.com/wp-content/uploads/2011/10/ARSC_BWF_Metadata_Test2_Results.zip

Test 2A: Impact of Editing Existing Chunks

WaveLab 6.10
Editing metadata values in the `bext` and/or `LIST-INFO` chunks caused loss of some previously embedded metadata in those chunks—any embedded metadata fields and values in the `bext` and `LIST-INFO` chunks not displayed/supported by WaveLab did not persist. These include the `bext` fields '`UMID`' and the `LIST-INFO` fields, '`IARL`', '`ICMS`', '`IMED`', '`IPRD`', '`ISRC`' and '`ITCH`'. Metadata in the `XMP`, `iXML`, and `axml` chunks appeared to persist.

Audition 3.0
Editing metadata values did not appear to impact the persistence of metadata. The only aberration appeared to be in the `bext` '`Coding History`' field, where the following text was added to the end of the previously embedded value:
"`A=PCM,F=96000,W=32,M=stereo`". Embedded metadata in the `XMP`, `iXML`, and `axml` chunks appeared to persist unaffected by alterations/additions to metadata in the `bext` and `LIST-INFO` chunks.

As the only application under test to support/display metadata in the `XMP` chunk, Audition does not appear to discard any metadata in the `XMP`, `iXML`, and `axml` chunks when an edit is made to the `XMP` chunk.

However, reviewing the checksum of the data chunk (where audio is stored) within the BWF file revealed that this action altered the audio. A null test confirmed that there was in fact a difference between the original audio and the audio in the file containing the edited metadata.

Audacity 1.3.11
Editing metadata values in any of the six available `LIST-INFO` fields caused loss of previously embedded metadata in any `LIST-INFO` fields not displayed/supported by Audacity. Metadata in the `bext`, `XMP`, `iXML`, and `axml` chunks was deleted.

Peak 5.1.2
Peak 5.1.2 does not support any of the metadata chunks under consideration in this study.

Peak 6.0
Throughout every sub-test in Test 2 Peak consistently removed any metadata not directly accessible through its interface.

- In all cases existing `XMP`, `axml`, and `iXML` chunks were removed in their entirety.
- In all cases all `LIST-INFO` fields which are not supported in Peak 6.0 were deleted.
- In all cases the '`timereference`' field was altered, reducing the '`samplecount`' value by 168 samples.
- When a field in the `LIST-INFO` chunk was edited all `bext` data persisted.
- In all cases an `smpl` chunk was added to the files.

Pro Tools LE 8.0.1
All issues discussed in Test 1 above apply here. The `bext` chunk was the only chunk modified by the tester, since Pro Tools only provided access to these fields.

- The '`description`' value was altered and this carried through appropriately, but the same behavior described in Test 1 took place. The same `iXML` fields were mapped into the '`description`' field. In addition, the new value that was placed in the `bext` '`description`' field was mapped to the `iXML` chunk within the `<note>` field.
- The '`originator reference`' value in the `bext` chunk was replaced with another value.
- The `axml`, `XMP` and `LIST-INFO` chunks were deleted.
- The same chunks mentioned in Test 1 were automatically embedded into the file.

iTunes 9.0.2
iTunes did not support any of the metadata chunks under consideration in this study.

Nuendo 4.0
The `bext` chunk was the only chunk modified by the test procedure, since Nuendo only provided access to these fields. The file produced by Nuendo for Test 2A had two `iXML` chunks which the software we used for assessing test results, BWF MetaEdit, rejected. Upon further investigation, the `iXML` chunks had some overlapping data as well as different data, making it unclear whether Nuendo had written two `iXML` chunks accidentally or not. There are no clear specifications that could be identified regarding the validity of a file with two `iXML` chunks. This raises an unanticipated point regarding chunk management that should be investigated further. Once this problem was solved, further review showed that no other fields had been altered.

Test 2B: Impact on Existing Chunks when Adding New Chunks

WaveLab 6.10
The results of this test were the same as those for test 2A.

Audition 3.0
There was no impact to other metadata chunks when adding chunks. However, as mentioned above this action caused an alteration to the data chunk, which is where the audio is stored.

Audacity 1.3.11
The results of this test were the same as those for test 2A.

Peak 5.1.2
Peak 5.1.2 did not support any of the metadata chunks under consideration in this study.

Peak 6.0
As in test 2A, BIAS Peak 6.0 deleted all data not displayed within the Peak interface.

Pro Tools LE 8.0.1
All issues discussed in Test 1 apply here as well. The `bext` chunk was the only chunk modified in the test, since Pro Tools only provided access to these fields.

- The '`description`' value was added by the tester, and carried through appropriately.
- The `bext` fields '`Originator`', '`OriginationDate`', '`OriginatorReference`', and '`OriginationTime`' were not populated by the tester, but were automatically populated by Pro Tools.
- The `axml`, `XMP` and `LIST-INFO` chunks were deleted by the application.
- The same chunks mentioned in Test 1 were automatically embedded into the file.

iTunes 9.0.2
iTunes 9.0.2 did not support any of the metadata chunks under consideration in this study.

Nuendo 4.0
Metadata added in the `bext` chunk did not persist. Metadata in the `LIST-INFO`, `XMP`, `iXML`, and `axml` chunks did persist. The tester reported complications and crashing in attempting to perform Test 2B which leads to an interest in additional testing.

Test 2C – Impact on Metadata of Audio-Only Editing

WaveLab 6.10
Editing the audio in the file did not cause any loss of metadata in any chunk, regardless of whether or not WaveLab supports/displays a specific metadata field.

Audition 3.0
Editing the audio in the file did not cause any loss of metadata in any chunk, regardless of whether or not Audition supports/displays a specific metadata field. The only aberration appears to be that in the `bext` '`Coding History`' field, the following text was added to the end of the previously embedded value: "`A=PCM,F=96000,W=32,M=stereo`".

Audacity 1.3.11
Audacity appeared to erase metadata embedded in the `XMP`, `iXML`, and `axml` chunks during simple editing operations. The `LIST-INFO` fields displayed/supported by Audacity were unaffected by simple editing operations.

Peak 5.1.2
Peak 5.1.2 did not support any of the metadata chunks under consideration in this study.

Peak 6.0
The results of this test were the same as those for tests 2A and 2B.

Pro Tools LE 8.0.1
The process used for generating a new file after performing the audio edit erased all embedded metadata, including the `iXML` chunk. Pro Tools automatically generated new metadata values for `bext` fields: '`Originator,`' '`OriginatorReference,`' '`OriginationDate,`' and '`OriginationTime.`' All other fields were left blank. All excess chunks which Pro Tools automatically embeds were present, with the exception of the `DGDA` chunk.

iTunes 9.0.2
iTunes 9.0.2 did not support any of the metadata chunks under consideration in this study.

Nuendo 4.0
This test was not performed on Nuendo 4.0.

Test 2D – Impact on Metadata of 'Save As' Function

WaveLab 6.10
Performing a '`Save As`' function on the file did not cause any loss of metadata in any chunk, regardless of whether or not WaveLab supports/displays a specific metadata field.

Audition 3.0
All embedded metadata displayed by Audacity persisted through '`Save As`' operations. Performing a '`Save As`' function on the file did not cause any loss of metadata in any chunk, regardless of whether or not Audition supports/displays a specific metadata field. The only aberration appears to be that in the `bext` '`Coding History`' field, the following was added the end of the previously embedded value: "`A=PCM,F=96000,W=32,M=stereo`".

The ‘save as’ function resulted in changes to the audio data in the same manner described in tests 2A and 2B.

Audacity 1.3.11
Audacity appeared to erase metadata embedded in the `XMP`, `iXML`, and `axml` chunks during ‘`Save As`’ operations.

Peak 5.1.2
Peak 5.1.2 did not support any of the metadata chunks under consideration in this study.

Peak 6.0
The results of this test were the same as those for tests 2A, 2B and 2C.

Pro Tools LE 8.0.1
The process used for generating a new file caused the behavior described in Test 2C.

iTunes 9.0.2
iTunes 9.0.2 did not support any of the metadata chunks under consideration in this study.

Nuendo 4.0
This test was not performed on Nuendo 4.0.

Test 3: Persistence and Integrity through Derivative Creation

Test 3 evaluated how embedded metadata is managed by each application in the creation of derivative files. MP3 and FLAC files store embedded metadata using structures that differ from BWF files. Therefore, the analysis identifies first the presence of a mapping scheme and, if present, what mapping strategy is employed and how well it works. Essentially this test aims to evaluate the extent to which applications support the inheritance of embedded metadata when performing transcoding operations. Detailed findings can be downloaded at:
http://www.avpreserve.com/wp-content/uploads/2011/10/ARSC_BWF_Metadata_Test3_Results.zip

WaveLab 6.10
Conversion to FLAC files was not possible in the WaveLab version used.
Using Phil Harvey’s ExifTool and the verbose extraction command (`-v5`) to evaluate the embedded metadata in the MP3 derivative file created, it appears that no metadata persisted from the original file.

The process of generating a derivative WAV file created an inaccuracy in the `bext` '`TimeReference.`' The '`samplecount`' persisted, but applications use this value to extrapolate the timestamp by dividing the '`samplecount`' by the sample rate. Thus, when the 96kHz sample rate file was converted to 44.1kHz file with the same '`samplecount`' value, the translated value went from being interpreted from the correct value of "`03:33:33.333`" to "`07:44:52.969`".[14] Metadata in the `XMP`, `iXML`, and `axml` chunks was unaltered.

Audition 3.0

Conversion to FLAC files was not possible in the Audition install used.

Before saving the test file as an MP3 or WAV 44.1 kHz derivative, the user is forced to first convert the sample rate of the audio under the '`Edit`' menu, by selecting the '`Adjust Sample Rate…`' option.

Analyzing the MP3 derivative using ExifTool revealed that embedded `XMP` metadata persisted, and that certain `LIST-INFO` fields were re-mapped to `ID3v1` and `ID3v2`.

Audition altered the '`samplecount`' value in order to maintain the correct timing. While this is an alteration, it is performed in order to maintain the integrity of the file's relationship with other files. This should be considered the correct practice. Metadata in the `XMP`, `iXML`, and `axml` chunks appeared to persist unaltered.

Audacity 1.3.11

Analyzing the FLAC derivative using ExifTool, it appears that only six fields of metadata persisted, all from the `LIST-INFO` chunk: '`ISFT`,' '`IART`,' '`ICMT`,' '`ICOP`,' '`ICRD`,' and '`INAM`'.

The derivative WAVE file retained only the embedded metadata in the `LIST-INFO` chunk displayed/supported by Audacity. All other embedded metadata did not persist.

Conversion to MP3 was not possible in the Audition install used.

Peak 5.1.2

Peak 5.1.2 does not support any of the metadata chunks under consideration in this study.

---

[14] The TimeReference value in the original file was '03:33:33.333'. At the original file's sample rate of 96,000 samples per second this equates to a sample count value of '1230079968'. In other words, if you divide 1230079968 by 96000 you get the number of seconds equivalent to 03:33:33.333. Although TimeReference is almost always displayed as a timestamp to users, it is actually stored as a sample count. When the transcoding was performed to create a file with a sample rate of 44,100 samples per second the sample count value remained the same. Therefore, the equation became 1230079968 divided by 44,100, generating a TimeReference timestamp of 07:44:52.969, creating an inaccuracy. For more information on TimeReference see http://www.avpreserve.com/wp-content/uploads/2011/08/AVPS_TimeReference_Primer.pdf

Peak 6.0
Peak 6.0 mapped the following `LIST-INFO` chunks into MP3 ID3v2 and FLAC tags: '`IART`' ('`Artist`'), '`INAM`' ('`Name (Title)`'), '`IGNR`' ('Genre') and '`ICMT`' ('`Comments`').

Metadata management with the derivative WAVE file followed the behaviors described in previous tests.

Pro Tools LE 8.0.1
The creation of MP3 and FLAC files was not possible on the Pro Tools install used.

Metadata management with the derivative WAVE file followed the behavior described in tests 2C and 2D.

iTunes 9.0.2
iTunes 9.0.2 did not support any of the metadata chunks under consideration in this study.

Nuendo 4.0
No derivative files were made by the tester.

IV. Conclusion and Discussion

Few metadata chunks were supported in their entirety by any software application. Rather, applications tended to display and provide access to selected fields of their choosing from each chunk standard.

The `bext` chunk was the most widely supported, followed by selected fields within the `LIST-INFO` chunk. Least supported were the XML-based chunks—there was some support for selected fields within `iXML`, but no support for `axml`, and support for `XMP` only by its creator, Adobe Systems.

Most troubling were the findings associated with application metadata management, where most applications automatically *erased* chunks and fields that they did not support after common user actions such as metadata or audio editing were performed. Embedded metadata did not persist nor consistently maintain its integrity across the audio software applications studied.

This study begins to quantify persistence and integrity issues relating to embedded metadata support and management within audio applications. These issues have major implications for the successful use of embedded metadata over time and across workflows. The impact of the problems this study documented reaches a wide audience, from rights holders to archives and production personnel to end-users. The value of embedded metadata is clear to the many communities that work with audio files. Despite this clarity, the findings of this study raise serious concerns, particularly for the archiving and preservation communities who rely on embedded metadata for interpretation and management of digital files representing preserved content into the future.

Metadata capture has long been a resource intensive and challenging proposition for archives. The unprecedented rate of growth in audiovisual content production that has taken place in the past decade creates management challenges of a different order of magnitude. However, the increased utilization of embedded metadata by systems that automatically populate files with technical, structural, descriptive, and administrative metadata provides new potential for its efficient capture and use, greatly reducing challenges related to both quantity and quality of metadata. This metadata is an aid to archives and conservators in the digital age for verifying the authenticity, integrity and provenance of files. It also enables increased accessibility and enhanced administrative capabilities.

End-users rely on embedded metadata for search and retrieval of audiovisual content on a daily basis within computer operating systems, applications such as iTunes, and on audio file-sharing sites such as archive.org. It is almost unthinkable for a musician to place audio files online without embedding metadata in them to identify at least the artist, song title, and rights information.

Producers rely on embedded metadata for management of media and workflows in the production process. This is evidenced by relatively wide support for `bext` and `iXML`, both of which have been adopted by the production community for identification and management purposes. This collective need has resulted in support from several system developers. However, this study demonstrates that this support is generally application-specific and does not accommodate the persistence and integrity of metadata across applications.

The legal system has also taken note of the significance of embedded metadata. In a 2009 Arizona Supreme Court Decision, the Court ruled in favor of a police officer who sued for employment discrimination. A letter drafted by his superiors regarding his performance was submitted as evidence in court. Through inspection of embedded metadata it was discovered that the letter had been backdated and was a falsification. Part of the Court's decision stated that "if a public entity maintains a public record in an electronic format, then the electronic version, including any embedded metadata, is subject to disclosure under our public records laws."[15]

Finally, the International Press Telecommunications Council's Photo Metadata Working Group recently issued a document entitled "Embedded Metadata Manifesto (2011)." This manifesto proposes five guiding principles for how metadata should be embedded and preserved in digital media files. They argue that not only photographers, but filmmakers, videographers, librarians and others share the problem of managing rapidly expanding collections of digital media assets. Embedding metadata is necessary to address this problem.[16]

The value of embedded metadata to these many communities, and the problems documented in this study, argue strongly for developing an operational environment that supports the persistence and integrity of embedded metadata across applications and time. The development and adoption by both manufacturers and users of operating principles and standards for handling embedded metadata is a key next step.

---

[15] Arizona Supreme Court. David Lake v. City of Phoenix. Arizona Supreme Court No. CV-09-0036-PR. http://www.supreme.state.az.us/opin/pdf2009/cv090036pr.pdf Consulted 20 May 2011.

[16] http://www.iptc.org/site/Photo_Metadata/Embedded_Metadata_Manifesto_%282011%29