Embedding Metadata in Digital Audio Files

Introductory Discussion for the Federal Agencies Guideline

By the Federal Agencies Audio-Visual Working Group
http://www.digitizationguidelines.gov/audio-visual/

September 15, 2009

INTRODUCTORY DISCUSSION

APPENDIXES

What is this document?

This is one of three documents pertaining to the embedding of metadata in digital audio files prepared by the Federal Agencies Audio-Visual Working Group in 2009. The two companion documents are:

- *Guideline for Federal Agency Use of Broadcast WAVE Files (Version 1.0)*
http://www.digitizationguidelines.gov/audio-visual/documents/Embed_Guideline_090915.pdf
- Consultant's report on embedding options in digital audio files
http://www.digitizationguidelines.gov/audio-visual/documents/AVPS_Audio_Metadata_Overview_090612.pdf

Introductory Discussion for the Federal Agencies Guideline

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. Many if not all metadata elements turn up in more than one place, a good thing since redundancy supports long-term preservation. (Being in more than one place, however, can make it difficult to update metadata across the board, unless this is supported in an automated way by an organization's technical infrastructure.)

The general topic of embedded metadata—in the broadest sense—is being explored by the Federal Agencies Digitization Initiative. Who are all of the actors (humans and systems) that the metadata ought to support? Which metadata elements serve their needs? When will these needs arise during the content lifecycle? A subgroup from the federal agencies initiative is developing a set of use cases that pertain to embedded metadata. The overall use-case effort is intended to be generic, representing both the still image and the audio-visual Working Groups.

Limits to the Audio-Visual Working Group's current proposal

As the larger federal agencies activity proceeds, the current investigation by the Audio-Visual Working Group is addressing only a portion—albeit a pressing and important portion—of the larger topic. Here are some of our limits:

- Format type. The associated guideline pertains to embedded metadata in audio files that result from the reformatting of analog content.
- Content lifecycle. The guideline is concerned with the initial stages of production and archiving. Of course, some of the elements identified—e.g., the embedding of appropriate identifiers—will "pay off" in later stages of the life cycle.
- Files not packages. The Working Group uses the term digital file (http://www.digitizationguidelines.gov/term.php?term=digitalfile) to name a single computer file, e.g., a WAVE file. The term digital package (http://www.digitizationguidelines.gov/term.php?term=digitalpackage) is used to name a digital entity that represents a single intellectual or logical entity, e.g., a digital copy of a published long-playing phonograph record that is described in a single library bibliographic record or a digital copy of an oral history audiocassette that is a listed item in an archival

finding aid. A digital package may consist of one or more files.[1] The emphasis of the associated guideline is on the metadata to be embedded in files.

- Master files more than derivative files. The guideline is concerned with embedded data in master files (http://www.digitizationguidelines.gov/term.php?term=archivalmasterfile and http://www.digitizationguidelines.gov/term.php?term=productionmasterfile) and, to a lesser degree, with derivative files (http://www.digitizationguidelines.gov/term.php?term=derivativefile).
- Administrative and descriptive metadata, including identifiers. The focus for the guideline is on administrative (http://www.digitizationguidelines.gov/term.php?term=metadataadministrative) and descriptive metadata (http://www.digitizationguidelines.gov/term.php?term=metadatadescriptive). This is not to discount the importance of technical metadata (http://www.digitizationguidelines.gov/term.php?term=metadatatechnical) but, for audio files, this category seems to be well established. See appendix A for more information on technical metadata in WAVE files.

Constraints imposed by WAVE and BWF file specifications, future explorations

The options for embedding administrative and descriptive metadata are limited by the WAVE and BWF formats' specifications. The underlying structure has been inherited from the 1991 Microsoft-IBM RIFF specification[2] that calls for an extensible structure made up of chunks. A number of chunks are specified in this document, including what is called the INFO list chunk (and "subchunks") for descriptive and administrative metadata. Over the years, various organizations have added new chunks. In 2001, the European Broadcasting Union (EBU) added chunks as part of its standardization of the Broadcast WAVE format (a WAVE subtype often referred to as BWF).[3] The most widely used BWF chunk is *bext* (Broadcast Extension), which allows for additional metadata to serve the needs of broadcasters. The data elements available in the INFO and *bext* chunks are listed in Appendix B.

In 2003, the EBU drafted an additions chunk for XML data called the axml, and a corollary schema called aXML.[4] Although of interest and potentially very useful for archiving, the axml

[1] This limit means that this set of guidelines will not address the specialized metadata that may be used to package multichannel (http://www.digitizationguidelines.gov/term.php?term=multitrackaudio), multitrack (http://www.digitizationguidelines.gov/term.php?term=multitrackaudio), and multisegment recordings. *Multichannel* and *multitrack* are defined by the glossary entries in the URLs cited; *multisegment* refers to such things as single, lengthy performances that have been broken into segments, each of which is represented by its own file. This topic receives insightful discussion in the final report from the Sound Directions project carried out by Indiana and Harvard Universities; see http://www.dlib.indiana.edu/projects/sounddirections/papersPresent/index.shtml.

[2] RIFF is documented in *Multimedia Programming Interface and Data Specifications 1.0* (1991, IBM Corporation and the Microsoft Corporation). This document is available at several Web sites, including http://www.kk.iij4u.or.jp/~kondo/wave/mpidata.txt and http://www.tactilemedia.com/info/MCI_Control_Info.html.

[3] BWF is documented in "BWF – A Format for Audio Data Files in Broadcasting," ver. 1, Tech 3285 (Geneva: Switzerland: European Broadcasting Union, July 2001), http://www.ebu.ch/CMSimages/en/tec_doc_t3285_tcm6-10544.pdf.

[4] "Specification of the Broadcast Wave Format; A Format for Audio Data Files in Broadcasting; Supplement 5: <axml> Chunk," http://www.ebu.ch/CMSimages/en/tec_doc_t3285_s5_tcm6-10485.pdf.

chunk does not seem to have been widely adopted by broadcasters and digital audio workstation manufacturers and the Working Group is not aware of any archive-oriented implementations.

The accompanying report from AudioVisual Preservation Solutions describes the axml chunk and some other metadata chunks that have been defined for the RIFF/WAVE format. On paper, these chunks overcome many of the limitations of the BWF bext chunk and the RIFF List INFO chunk. Their low levels of adoption, however, make them less appealing than one would wish. Nevertheless, the Working Group will explore them in the future as possible improved solutions to the metadata-embedding needs of preservation-oriented archives.

Appendix A
Technical Metadata in WAVE Files

The WAVE Format Chunk

The key document that governs WAVE-specific technical metadata is the RIFF (Resource Interchange File Format) specification: *Multimedia Programming Interface and Data Specifications 1.0* (1991, IBM Corporation and the Microsoft Corporation). This publication specifies the content for the WAVE Format chunk (<fmt-ck>), which in turn specifies the format of the actual waveform sound data (<wave-data>).

Field	**Description**
FormatTag	A number indicating the WAVE format category of the file. The content of the <format-specific-fields> portion of the fmt chunk, and the interpretation of the waveform data, depend on this value. You must register any new WAVE format categories. See Registering Multimedia Formats in Chapter 1, Overview of Multimedia Specifications, for information on registering WAVE format categories. Wave Format Categories, following this section, lists the currently defined WAVE format categories. If the wFormatTag field of the <fmt-ck> is set to WAVE_FORMAT_PCM, then the waveform data consists of samples represented in pulse code modulation (PCM) format.[5] For PCM waveform data, the <format-specific-fields> includes the BitsPerSample data element (next item listed).
BitsPerSample	Specifies the number of bits of data used to represent each sample of each channel. If there are multiple channels, the sample size is the same for each channel.
Channels	The number of channels represented in the waveform data, such as 1 for mono or 2 for stereo.
SamplesPerSec	The sampling rate (in samples per second) at which each channel should be played.
AvgBytesPerSec	The average number of bytes per second at which the waveform data should be transferred. Playback software can estimate the buffer size using this value.

[5] The *Multimedia Programming Interface and Data Specifications 1.0* lists the following WAVE format categories with their FormatTag, Value, and Format Category:
WAVE_FORMAT_PCM (0x0001) Microsoft Pulse Code Modulation (PCM) format
IBM_FORMAT_MULAW (0x0101) IBM mu-law format
IBM_FORMAT_ALAW (0x0102) IBM a-law format
IBM_FORMAT_ADPCM (0x0103) IBM AVC Adaptive Differential Pulse Code Modulation format
Many additional categories are lists in the documentation for the JHOVE WAVE module (http://hul.harvard.edu/jhove/wave-hul.html).

BlockAlign	The block alignment (in bytes) of the waveform data. Playback software needs to process a multiple of BlockAlign bytes of data at a time, so the value of BlockAlign can be used for buffer alignment.

JHOVE output

An indication of the presence of and the values for these technical metadata elements is provided by JHOVE (version 1.x; JSTOR/Harvard Object Validation Environment, http://hul.harvard.edu/jhove/). The table below blocks out an excerpt for a file from the American Folklife Center at the Library of Congress; the entire JHOVE report is provided at the end of this appendix. The JHOVE reporting organizes and tags the output to maximize the ease with which the data can be translated into the XML schema structure for the Audio Engineering Society standard AES-X098B, Administrative and structural metadata for audio objects.[6]

JHOVE report tag	**JHOVE value or note**	**Equivalent WAVE <fmt-ck> tag**	**Value, decimal**	**Comment**
Profile	PCMWAVEFORMAT	FormatTag	1	Tells us that this is LPCM encoding and thus sets up for the use of appropriate encoding-specific tags.
FormatRegion	[Has a set of sub-tags: BitDepth, SampleRate, CompressionCode, AverageBytesPerSecond, BlockAlign, Data (and DataLength)]	None	--	The writers have not determined what this is with certainty; repeats data also provide elsewhere. Developed to serve the needs of AES X098-B; seemingly not relevant to this discussion at this time.
		Channels	2	Provided in JHOVE as indicated at bottom of table
BitDepth	24	BitsPerSample	24	
SampleRate	96000.0	SamplesPerSec	96000	
CompressionCode	PCM audio in integer format	FormatTag (same as above)	1	
AverageBytesPerSecond	576000	AvgBytesPerSec	576000	SampleRate x number of channels x bytes per sample. Not used in AES-X098B.
BlockAlign	6	BlockAlign		Value in WAVE not known to compilers, probably 6. Not used in AES-X098B.
Channel information				
NumChannels	2	Channels		
Stream:				
ChannelNum	0			
ChannelAssignment	LEFT			
Stream:				
ChannelNum	1			
ChannelAssignment	RIGHT			

[6] As of October 2008, the standard remains in draft form; a copy for Working Group reference only (not for broader dissemination) is available here: http://home.comcast.net/~cfle/AVdocs/AES_x98b-sec-pwd-070914.pdf.

Complete JHOVE output for the file summarized in the preceding table.

```
Jhove (Rel. 1.1, 2006-09-05)
 Date: 2008-07-22 12:38:29 EDT
 RepresentationInformation: [pathname redacted]/afs5168am.wav
  ReportingModule: WAVE-hul, Rel. 1.2 (2005-09-26)
  LastModified: 2008-03-13 07:58:24 EDT
  Size: 417664048
  Format: WAVE
  Status: Well-Formed and valid
  SignatureMatches:
   WAVE-hul
  MIMEtype: audio/x-wave
  Profile: PCMWAVEFORMAT
  WAVEMetadata:
   AESAudioMetadata:
    AnalogDigitalFlag: FILE_DIGITAL
    SchemaVersion: 1.02b
    Format: WAVE
    AudioDataEncoding: PCM audio in integer format
    ByteOrder: LITTLE_ENDIAN
    FirstSampleOffset: 44
    Use:
     UseType: OTHER
     OtherType: JHOVE_validation
    PrimaryIdentifier: [pathname redacted]/afc1941014/afs5168/afs5168am.wav
     IdentifierType: FILE_NAME
    Face:
     TimeLine:
      StartTime:
       FrameCount: 30
       TimeBase: 1000
       VideoField: FIELD_1
       CountingMode: NTSC_NON_DROP_FRAME
       Hours: 0
       Minutes: 0
       Seconds: 0
       Frames: 0
       Samples:
        SampleRate: S44100
        NumberOfSamples: 0
       FilmFraming: NOT_APPLICABLE
        Type: ntscFilmFramingType
      Duration:
       FrameCount: 30
       TimeBase: 1000
       VideoField: FIELD_1
       CountingMode: NTSC_NON_DROP_FRAME
       Hours: 0
       Minutes: 12
       Seconds: 5
       Frames: 3
       Samples:
        SampleRate: S96000
        NumberOfSamples: 1024
```

```
       FilmFraming: NOT_APPLICABLE
        Type: ntscFilmFramingType
      NumChannels: 2
      Stream:
       ChannelNum: 0
       ChannelAssignment: LEFT
      Stream:
       ChannelNum: 1
       ChannelAssignment: RIGHT
    FormatList:
     FormatRegion:
      BitDepth: 24
      SampleRate: 96000.0
   CompressionCode: PCM audio in integer format
   AverageBytesPerSecond: 576000
   BlockAlign: 6
   Data:
    DataLength: 417663744
   ListInfo:
    Comments: Courtesy of The American Folklife Center, The Library of
Congress, Washington D.C.
    Copyright: See Copyright Restriction Statement.
    CreationDate: 2007-10-25
    Engineer: Safe Sound Archive
    Name: AFC1941/014: AFS5168a
    Software: Sony Sound Forge 8.0
```

Appendix B.

Data Elements in the WAVE (RIFF) INFO List and BWF bext Chunks

WAVE INFO chunk and list of elements

The INFO chunks are inherited by WAVE and BWF from the specification for their parent, the Resource Interchange File Format (RIFF; see http://www.digitalpreservation.gov/formats/fdd/fdd000025.shtml). The following descriptive overview was collected on December 22, 2008, from an online document titled "Inside the RIFF Specification" by Hamish Hubbard, dated September 1, 1994 (http://www.ddj.com/184409308).

- LIST chunks are the only chunks apart from RIFF chunks that may contain their own subchunks . . . . LIST chunks are usually subchunks of RIFF chunks themselves. Like RIFF chunks, LIST chunks have a four-character code in the first four bytes of their data area. This code specifies the list type (analogous to a RIFF chunk's form type) . . . .
- For example, a LIST chunk of list type INFO may contain subchunks such as INAM (the name of the data stored in the file) and ICRD (creation date). LIST chunks of type INFO are optional in current RIFF forms, but their use is recommended. The LIST chunks' subchunks can store much more information about the file than is available from the filename and date stamp. These LIST subchunks share a common format: Each contains one ASCII Z (NULL terminated) string.

The following introductory statement and element list is from the RIFF specification, the Multimedia Programming Interface and Data Specifications 1.0, issued as a joint design by IBM Corporation and Microsoft Corporation, August 1991. In this document the block of data is called *INFO list chunk* (singular) and in another section there is a reference to the *list chunk type* as having *subchunks*. Regarding the "values" that can associated with each subchunk or element, the writers of this document believe that there is no particular limit on the length of the null-terminated coded chunks. We also believe that these subchunks are not repeatable. Comments from specialists in the field will be welcome.

The INFO list is a registered global form type that can store information that helps identify the contents of the chunk. This information is useful but does not affect the way a program interprets the file; examples are copyright information and comments. An INFO list is a LIST chunk with list type INFO. The following shows a sample INFO list chunk:

```
LIST('INFO'   INAM("Two Trees"Z)
         ICMT("A picture for the opening screen"Z) )
```

An INFO list should contain only the following chunks. New chunks may be defined, but an application should ignore any chunk it doesn't understand. The chunks listed below may only appear in an INFO list. Each chunk contains a ZSTR, or null-terminated text string.

Chunk ID	Description
IARL	Archival Location. Indicates where the subject of the file is archived.
IART	Artist. Lists the artist of the original subject of the file. For example, Michaelangelo.

ICMS	Commissioned. Lists the name of the person or organization that commissioned the subject of the file. For example, Pope Julian II.
ICMT	Comments. Provides general comments about the file or the subject of the file. If the comment is several sentences long, end each sentence with a period. Do not include newline characters.
ICOP	Copyright. Records the copyright information for the file. For example, Copyright Encyclopedia International 1991. If there are multiple copyrights, separate them by a semicolon followed by a space.
ICRD	Creation date. Specifies the date the subject of the file was created. List dates in year-month-day format, padding one-digit months and days with a zero on the left. For example, 1553-05-03 for May 3, 1553.
ICRP	Cropped. Describes whether an image has been cropped and, if so, how it was cropped. For example, lower right corner.
IDIM	Dimensions. Specifies the size of the original subject of the file. For example, 8.5 in h, 11 in w.
IDPI	Dots Per Inch. Stores dots per inch setting of the digitizer used to produce the file, such as 300.
IENG	Engineer. Stores the name of the engineer who worked on the file. If there are multiple engineers, separate the names by a semicolon and a blank. For example, Smith, John; Adams, Joe.
IGNR	Genre. Describes the original work, such as, landscape, portrait, still life, etc.
IKEY	Keywords. Provides a list of keywords that refer to the file or subject of the file. Separate multiple keywords with a semicolon and a blank. For example, Seattle; aerial view; scenery.
ILGT	Lightness. Describes the changes in lightness settings on the digitizer required to produce the file. Note that the format of this information depends on hardware used.
IMED	Medium. Describes the original subject of the file, such as, computer image, drawing, lithograph, and so forth.
INAM	Name. Stores the title of the subject of the file, such as, Seattle From Above.
IPLT	Palette Setting. Specifies the number of colors requested when digitizing an image, such as 256.
IPRD	Product. Specifies the name of the title the file was originally intended for, such as Encyclopedia of Pacific Northwest Geography.
ISBJ	Subject. Describes the contents of the file, such as Aerial view of Seattle.
ISFT	Software. Identifies the name of the software package used to create the file, such as Microsoft WaveEdit.
ISHP	Sharpness. Identifies the changes in sharpness for the digitizer required to produce the file (the format depends on the hardware used).
ISRC	Source. Identifies the name of the person or organization who supplied the original subject of the file. For example, Trey Research.
ISRF	Source Form. Identifies the original form of the material that was digitized, such as slide, paper, map, and so forth. This is not necessarily the same as IMED.
ITCH	Technician. Identifies the technician who digitized the subject file. For example, Smith, John.

The BWF bext chunk, list of elements, and discussion of CodingHistory

The list of elements and a descriptive definition that follows is taken from this standards document from the European Broadcasting Union (EBU): “BWF – A Format for Audio Data Files in Broadcasting,” ver. 1, Tech 3285 (Geneva: Switzerland: European Broadcasting Union, July 2001), http://www.ebu.ch/CMSimages/en/tec_doc_t3285_tcm6-10544.pdf. The limits on the extent of the values permitted for each tag are provided in the definitions. The writers’ understanding is that these fields are not repeatable.

Description.	ASCII string (maximum 256 characters) containing a free description of the sequence. To help applications which only display a short description, it is recommended that a résumé of the description is contained in the first 64 characters, and the last 192 characters are use for details. If the length of the string is less than 256 characters, the last one is followed by a null character. (00)
Originator.	ASCII string (maximum 32 characters) containing the name of the originator/producer of the audio file. If the length of the string is less than 32 characters, the field is ended by a null character.
OriginatorReference.	ASCII string (maximum 32 characters) containing a non ambiguous reference allocated by the originating organization. If the length of the string is less than 32 characters, the field is ended by a null character. Note: The EBU has defined a format for the OriginatorReference field. See EBU Recommendation R99-1999.
OriginationDate.	Ten ASCII characters containing the date of creation of the audio sequence. The format is .yyyy-mm-dd. (year-month-day). Year is defined from 0000 to 9999; Month is define from 1 to 12; Day is defined from 1 to 28,29,30 or 31. The separator between the items can be anything but it is recommended that one of the following characters is used: .-. hyphen ._. underscore .:. colon . . space ... stop [Editor’s note: it is not possible to use the syntax of ISO 8601 datetime for this element. Thus date and time cannot be expressed in terms of UTC.]
OriginationTime.	Eight ASCII characters containing the time of creation of the audio sequence. The format is .hh-mm-ss. (hours-minutes-seconds). Hour is defined from 0 to 23. Minute and second are defined from 0 to 59. The separator between the items can be anything but it is recommended that one of the following characters is used: .-. hyphen ._. underscore .:. colon . . space ... stop [Editor’s note: it is not possible to use the syntax of ISO 8601 datetime for this element. Thus date and time cannot be expressed in terms of UTC.]
TimeReference.	This field contains the timecode of the sequence. It is a 64-bit value which contains the first sample count since midnight. The number of samples per second depends on the sample frequency which is defined in the field

	<nSamplesPerSec> from the <format chunk>. [Editor's note: This metadata element records what is sometimes called a *timestamp*, i.e., the start time for a given file, in terms of a timeline of the sort often provided by digital audio workstation software. Timelines are often set up to begin at 00:00:00 ("midnight"). Files can be placed on the timeline in terms of their sequence using TimeReference. In <bext>, the time value is provided in terms of sample count for timeline location for the first sample in the file. The Working Group understands that this element has special value in the case of multitrack or multisegment content, to be explored further at another time.]
Version.	An unsigned binary number giving the version of the BWF, particularly the contents of the Reserved field. For Version 1, this is set to 0001h. UMID 64 bytes containing a UMID (Unique Material Identifier) to the SPMTE 330M standard [3]. If only a 32 byte .basic UMID. is used, the last 32 bytes should be set to zero. (The length of the UMID is given internally.) Note: The EBU intends to publish guidance on the use of the UMID in audio files.
Reserved.	190 bytes reserved for extensions. If the Version field is set to 0001h, these 190 bytes must be set to a NULL (zero) value.
CodingHistory.	Non-restricted ASCII characters, containing a collection of strings terminated by CR/LF. Each string contains a description of a coding process applied to the audio data. Each new coding application is required to add a new string with the appropriate information. This information must contain the type of sound (PCM or MPEG) with its specific parameters: -- PCM : mode (mono, stereo), size of the sample (8, 16 bits) and sample frequency: -- MPEG : sample frequency, bit-rate, layer (I or II) and the mode (mono, stereo, joint stereo or dual channel). It is recommended that the manufacturers of the coders provide an ASCII string for use in the coding history. Note: The EBU has defined a format for CodingHistory which will simplify the interpretation of the information provided in this field. See EBU Recommendation R98-1999.

Special note on CodingHistory

The Coding History element is interesting because it offers a place and a method for recording process or digital-provenance metadata (http://www.digitizationguidelines.gov/term.php?term=metadataprocess). The use of this field to record some of the reformatting history for given item has been outlined in the final report from the Sound Directions project carried out by Indiana and Harvard Universities (http://www.dlib.indiana.edu/projects/sounddirections/papersPresent/index.shtml). The report describes the use of Coding History at Indiana University and their discussion is presented in the box that follows.

The Coding History field is designed to hold data on the digitizing process including signal chain specifics, sample rate and bit depth, and other elements. It is defined as a collection of strings, each presented on a separate line, containing a history of the coding processes applied to the file. Each variable within a string is separated by a comma. A new line is added when the coding history related to the file is changed, and each line should end with a carriage return and line feed which are automatically added by WaveLab. According to the EBU, each line should contain these elements, as appropriate to the coding history being described:
-- Coding algorithm. String begins with "A=" For example: A=ANALOG, PCM, MPEG1L3, and others
-- Sampling frequency. String begins with "F="
-- Bit-rate, for MPEG coding only. String begins with "B="
-- Word length. String begins with "W="
-- Mode—this corresponds to sound field, such as mono, stereo, or dual-mono. String begins with "M="
-- Text, free string—a free ASCII-text string for in-house use. The EBU suggests documenting devices in the signal chain and analog source recording formats in this field. String begins with "T="

At Indiana, we include three lines of coding history in our BWF files for the digitization of analog recordings. The first documents the analog source recording, the second contains data on digitization chain, while the third records information on the storage of the file.

For example:
A=ANALOG,M=mono,T=Studer A810; SN3690; 15 ips; open reel tape,
A=PCM,F=96000,W=24,M=mono,T=Benchmark; ADC1; SN00252; A/D,
A=PCM,F=96000,W=24,M=mono,T=Lynx; AES16; DIO,

Line 1 reads: an analog open reel tape with a mono sound field was played back on a Studer A810 tape machine with serial number 3690. Tape speed was 15 ips.

While the EBU document suggests including the tape brand and product number as the last element, we prefer a general designation of the format for several reasons: it is more useful to know the format than the specific brand and it avoids the need to interpret the brand information and playback machine data to identify the format. When a range of formats—analog cassettes, discs, DATs and others—are routinely digitized this interpreting might become unnecessarily difficult. In addition, the format remains constant through an entire collection (the brand and product number may or may not), providing one less element that requires data entry for each source recording.

Line 2 reads: the tape was digitized in mono mode using a Benchmark ADC1 A/D converter with serial number 00252 at 96 kHz sample rate with a bit depth of 24 bits.

Line 3 reads: the tape was stored as a 96/24 mono file using a Lynx AES16 digital input/output interface.

If we apply additional coding processes to produce a derivative file we add a fourth line in the header of the derivative file. For example:

A=PCM,F=44,100,W=16,M=mono,T=Steinberg; WaveLab 6; Resampler, Waves L2; Dither; DAW,

This line reads: A 16 bit, 44.1 kHz file was created using the WaveLab 6 Resampler and Waves L2 Dither in the Digital Audio Workstation.

Appendix C.
Identifiers: Types and Characteristics

Introduction

This appendix describes a varied landscape, in which different identifiers coexist and operate at varying levels of granularity and actionability. Identifiers from every part of this landscape are employed by federal agencies and other archives with little consistency, thus inhibiting the interoperability of our digital content. Some agencies use identifiers that link digital files to a metadata record in the system an agency uses to provide public access, while others use identifiers that connect solely to a local database, not accessible to the public. (Some content within an agency will be associated with both types of identifiers.) Some identifiers are attributes of the metadata while others are attributes of the digital content itself.

Identifier as attribute of a metadata record: LC example

Library of Congress Control Numbers (LCCNs) are the identifiers for the Library's bibliographic records (attribute of the metadata). They are not only "in" the record but the bibliographic records can be retrieved by the public using the PermaLink, a method for including the LCCN in an actionable URL. When the following URL is keyed into a browser, for example, the Permalink http://lccn.loc.gov/mp76000002 returns a presentation of the bibliographic record for the motion picture *Theodore Roosevelt's Return to New York, 1910*. By adding ../marcxml, ../mods, or ../dc to the URL, the system's action returns the same data in the form of a MARCXML record, a MODS record, or in the Dublin Core XML format. Thus, if an LCCN is embedded in digital content, it permits a user to pull up the bibliographic record.

Identifier as attribute of digital content itself: LC example

The bibliographic record for *Theodore Roosevelt's Return to New York, 1910* also contains an identifier for the digital content (attribute of the content). Scroll down in the display of the record to the MARC $856 field, and you will come to an additional actionable link in the form of a Handle identifier: http://hdl.loc.gov/loc.mbrsmi/trmp.4170. (Strictly speaking, this is a URL that incorporates the Handle.) Evoking this Handle brings back another presentation of the bibliographic record, this one in American Memory. That bib-record presentation includes further links based on the logical path- and file-name for the three files (at different resolutions and in different formats) that contain the movie, e.g., http://memory.loc.gov/mbrs/trmp/4170s1.mpg.

Examples in other agencies

Similar identifiers are in play in other agencies. At the National Archives, for instance, recent practice has made use of Record Group, Series, and Item identifiers for audio, motion picture, and video holdings, what the agency calls dynamic media. (At this writing, planning at NARA has begun for a new approach for digitized dynamic media items making use of non-mnemonic unique identifiers. All identifiers would be tracked in a related database and selected identifiers would also be embedded in the files.) Meanwhile, at least one participating agency assigns the widely used international ISRC identifier[7] to some of its digital content.

Digital packages, digital package parts, and digital files

Identifiers are associated with—*identify*—entities that exist at different levels of abstraction. The Working Group had adopted terms that refer to these levels, with definitions provided in the initiative's online glossary:

- Digital package
  - http://www.digitizationguidelines.gov/term.php?term=digitalpackage
- Digital package part
  - http://www.digitizationguidelines.gov/term.php?term=digitalpackagepart
- Digital file
  - http://www.digitizationguidelines.gov/term.php?term=digitalfile

How do these named entities relate to identifiers? The following paragraphs describe some archetypes; in actual practice, there are often nuanced shadings from one type to another.

- Digital-package-level identifier.[8] In a library setting, packages generally correlate to what are called *manifestations* in the parlance of the Functional Requirements for Bibliographic Records (FRBR), and thus to identifiers like the LCCN (Library of Congress Control Number), ISBN, ISSN, ISRC, ISTC, ISWC, Superintendent of Documents number, etc.. In an archive, digital packages generally correlate to an *item* in, say, an EAD (Encoded Archival Description) finding aid, and to identifiers that reference the name for the collection, series, and item. Very often, this identifier will have been established prior to digitizing and will therefore be available for embedding by the digitization team.

- Digital-package-part identifier. This identifier has to do with structural metadata; it is intended to answer the "what sub-part am I in relationship to other parts of the overall package?" Here's a hypothetical example: a Library of Congress book-scanning project uses the string 00220008 to identify scan-exposure 22, representing the book page with the number 8 printed on it. This identifier is not typically part of a bibliographic record or finding aid but (in this example) is more or less self-explanatory within the object set. The identifier is still an abstraction; it is not "at the file level." Multiple image files (e.g., archival master, derivative viewing file, and thumbnail) may all represent the same object sub-part, e.g., page 8. Here's a hypothetical recorded sound example: gfc1972.cas14.sd2. This

[7] The International Standard Recording Code (http://www.ifpi.org/content/section_resources/isrc.html) is standardized in ISO 3901 and resulted from the work of International Federation of the Phonographic Industry (IFPI), based in London. The twelve-character ISRC consists of four elements: (1) ISO Country, e.g. GB for the UK, or US for the USA, DE for Germany, etc.; (2) Registrant Code, a three-character alpha-numeric unique reference; (3) Year of Reference, the last two digits of the current year, e.g. '03' for 2003; and (4) Designation Code, a five-digit unique number, e.g. '00013.' The ISO Country Code and the Registrant Code are issued by the National Agencies or by the International ISRC Agency; the rest of the identifier is then allocated by the entity wishing to identify their sound or music video recordings.

[8] Terminology varies for this conceptual entity. For example, the Sound Directions final report (section 4.2.3.4.1.1 *Source Audio Object Structure*, page 64) states, "Audio object is a general term used to describe digital files and physical audio carriers. This is the root of the hierarchy and contains metadata that pertains to the entire object. Each audio object is described by a single instance document in a strict one-to-one mapping."

identifier documents the fact that this sub-part is side 2 of original audiocassette 14, from the 1972 portion of the Galax Fiddlers Convention Collection. Another subpart in the set would be identified as gfc1972.cas14.sd1, reproducing cassette side 1. Generally speaking, the object-sub-part identifier is determined during the digitization process (including object-preparation stages) and the information is often available for embedding at production time.

- File identifier. This is an identifier assigned to the file itself, i.e., a kind of license tag that may or may not refer directly to descriptive information. For some digitizing projects, this is assigned by the Digital Asset Management system at the time of digitization. If the file identifier does not directly link to descriptive information, the systems in use generally include a database or other look-up tool (some may be "manual") that can be used to connect back to descriptive information.

  Identifiers that operate at the file level are found in METS documents. The METS Primer and Reference Manual states, "A <file> element may contain one or more <FLocat> elements which provide pointers to a content file and/or a <FContent> element which wraps an encoded version of the file." (p. 29) The FLocat elements has a LOCTYPE, for which the following types have been defined, some of which are persistent identifiers:
  ARK: Archival Resource Key
  URN: Uniform Resource Name
  URL: Uniform Resource Locator
  PURL: Persistent URL
  HANDLE: a CNRI Handle
  DOI: A Digital Object Identifier

  The following example is provided at the METS Web site, with a FLocat of the LOCTYPE URL.
  ```
  <mets:file ID="FID1" MIMETYPE="image/tiff" SEQ="1" CREATED="1999-06-17T00:00:00" ADMID="ADM1A" GROUPID="GID1">
  <mets:FLocat xlink:href="http://sunsite.berkeley.edu/masters/bkm00002773a.tif" LOCTYPE="URL"/>
  </mets:file>
  ```

  File identifiers may be pre-assigned or assigned at production time. They may be assigned by a production-management, digital-asset-management, or repository-ingestion system, or by manual means.

Multiple identifiers and the desirability of typing

As suggested by the preceding, there may be more than one identifier to be embedded in a file or object-packaging metadata. A comparative example exists for digital photography, in the specification for the International Press Telecommunications Council (IPTC)[9] photo metadata set. The specification provides element definitions for the following types:

---

[9] IPTC Standard Photo Metadata 2008, IPTC Core Specification Version 1.1, IPTC Extension Specification Version 1.0, Document Revision 2. International Press Telecommunications Council, 2008. http://iptc.cms.apa.at/std/photometadata/2008/specification/IPTC-PhotoMetadata-2008_2.pdf.

- Digital Image GUID (p. 32). Globally unique identifier for this digital image. It is created and applied by the creator of the digital image at the time of its creation . This value shall not be changed after that time.
- Image Registry Entry (p. 35). Typically an id from a registry is negotiated and applied after the creation of the digital image. Both a Registry Item Id and a Registry Organization Id [are required] to record any registration of this digital image with a registry.
  - Related term: Item Id {registry entry detail} (p. 48). A unique identifier created by a registry and applied by the creator of the digital image. This value shall not be changed after being applied. This identifier is linked to a corresponding Registry Organization Identifier.
  - Related term: Organization Id {registry entry detail} (p. 49). An identifier for the registry which issued the corresponding Registry Image Id.
  - Related term: Registry Entry Details {data type} (p. 49). A structured datatype for an entry in a registry, includes the id for the image issued by the registry and the registry's id.
- Source Inventory Number {Artwork or Object detail} (p. 44). The inventory number issued by the organization or body holding and registering the artwork or object in the image.

Since there is a good chance that a given object or file will be associated with multiple identifiers, there will be considerable value in having a metadata encoding that allows for repeating elements and/or attributes, e.g., identifier_type, identifier_value. This approach would resemble that described for METS FLocat and LOCTYPE, above.

Meanwhile, another perspective on multiple identifiers is provided by the DLF wiki Best Practices for OAI Data Provider Implementations and Shareable Metadata (http://webservices.itcs.umich.edu/mediawiki/oaibp/index.php/IdentifyingTheResource). The following recommendation pertains to circumstances in which multiple versions of a digital object may exist, looking from external Dublin Core metadata toward the digital content:

- [I]n the case of digital objects, if the identifiers resolve to multiple versions of the resource, it is important to identify a single primary identifier that a service provider can label or use as the primary link to the resource. For example, only one <dc:identifier> element should be included with an actionable identifier (i.e. a URL). Additional <dc:identifier> elements might be included with a local identifier if not actionable (i.e. an end-user cannot click on the identifier to arrive at the resource).

Identifier characteristics

The many categories and types of identifiers possess varying and often overlapping characteristics.

- Actionable. Roughly speaking, this has to do with the provision of an automated response at retrieval time, sometimes called resolution: can you plug the identifier into a query box or URL slot, and have a machine retrieve and send you back information or some other response? Reponses may include descriptive information intended for a human user, a machine-readable version of that description, or to a file of some kind.

At the Library of Congress, Permalinks and Handles are actionable identifiers; see the examples provided at the beginning of this appendix.

Although actionable identifiers are almost always preferable, resolution is potentially expensive when it goes beyond a local system and especially if resolution involves more than redirection. Handles at the Library of Congress work by means of simple redirection and thus they are cheap. The version of the Handle adopted by publishers--the Digital Object Identifier (DOI)--requires a combination of redirection and centrally stored key metadata, and thus is more expensive to support. There may be value for a digitizing unit or organization in having a locally managed resolvable identifier and participating, as appropriate, in broader systems of identifiers that may not be actionable.

- Findable. Although actionable identifiers are desirable, non-actionable, "findable" identifiers can be very helpful in their stead. Non-actionable identifiers are sometimes called labels. (See, for example, the International DOI Foundation's use of this term in their FAQ: http://www.doi.org/faq.html.) Often you can use a search engine to look for information associated with an identifier. When a researcher writes about a specific work and uses something like the ISBN as a label, he or she has unambiguously identified the item under discussion even if neither the writer nor the reader try to resolve the ISBN through its naming authority.

- Local, global. Roughly speaking, this has to do with how widely an identifier is known and understood. Some identifiers may only be understood by the staff of the digitizing unit or organization, or may only function in a local system for local users, e.g., when content is managed by a local-service Digital Asset Management System. Other identifiers may be global, like the ISBN, DOI, or patent registration numbers. Some of these may be actionable, some not.

- Mnemonic, non-mnemonic. This refers to the "human-readability" of an identifier, which may vary by degree (some are easier to decipher than others) and the dependency on insider knowledge.
    - Example: The digital ID for the American Memory sound recording of the fiddle tune Chicken Reel by Red Harmon and Willard Brewer is AFCTS 4107b2. The string AFCTS stands for American Folklife Center Todd-Sonkin collection, and 4107b2 stands for original disc-recording number 4107, side B, cut 2.
    - In contrast, copyright registration numbers are not mnemonic, save for the prefix that defines the class. For example, VAu 598-675 (aka vau000598675) is from the Visual Materials class. But to know what the number 598-675 references, you must look it up. This example is post-1977 and can be the subject of a search (http://cocatalog.loc.gov/); earlier examples require a visit to the 3x5-inch-card catalog in the Copyright Office.

Comment on production workflow

Common library and archive practices today feature two basic options: (1) identifiers for reproduction files are based on an identification scheme used for or derived from the originals (presumably making it easy to hook up the reproductions later with systems used to describe or manage the originals); and (2) new (and possibly non-mnemonic) identifiers are created for each file, and thereafter a database (and/or sidecar or embedded metadata that will end up in a database) is used to connect key identification information to this new identifier. Libraries and archives typically want to use option 1, but asset management systems usually assume 2. An organization must decide which it will use for a particular workflow (or whether it will carry both through the process) and stick with it.