

ANSI/CTA Standard

Digital Audiobook File Format and Player Requirements

ANSI/CTA-2003-C R-2013

(Formerly ANSI/CEA-2003-C R-2013)

February 2007



NOTICE

Consumer Technology Association (CTA)™ Standards, Bulletins and other technical publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for his particular need. Existence of such Standards, Bulletins and other technical publications shall not in any respect preclude any member or nonmember of the Consumer Technology Association from manufacturing or selling products not conforming to such Standards, Bulletins or other technical publications, nor shall the existence of such Standards, Bulletins and other technical publications preclude their voluntary use by those other than Consumer Technology Association members, whether the standard is to be used either domestically or internationally.

Standards, Bulletins and other technical publications are adopted by the Consumer Technology Association in accordance with the American National Standards Institute (ANSI) patent policy. By such action, the Consumer Technology Association does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the Standard, Bulletin or other technical publication.

Note: The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights.

By publication of this standard, no position is taken with respect to the validity of this claim or of any patent rights in connection therewith. The patent holder has, however, filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Details may be obtained from the publisher.

This document does not purport to address all safety problems associated with its use or all applicable regulatory requirements. It is the responsibility of the user of this document to establish appropriate safety and health practices and to determine the applicability of regulatory limitations before its use.

This document is copyrighted by the Consumer Technology Association and may not be reproduced, in whole or part, without written permission. Federal copyright law prohibits unauthorized reproduction of this document by any means. Organizations may obtain permission to reproduce a limited number of copies by entering into a license agreement. Requests to reproduce text, data, charts, figures or other material should be made to the Consumer Technology Association.

(Formulated under the cognizance of the CTA **R6 Portable, Handheld and In-Vehicle Electronics Committee.**)

Published by
©CONSUMER TECHNOLOGY ASSOCIATION 2015
Technology & Standards Department
www.CTA.tech

All rights reserved

FOREWORD

This standard was developed under the auspices of the Consumer Electronics Association (CEA) R6 Mobile Electronics Committee.

CD/MP3 information from CEA-2003, CEA-2003-A, and CEA-2003-B are retained. CEA-2003-C is backward compatible with and supercedes CEA-2003, CEA-2003-A, and CEA-2003-B.



(This page intentionally left blank)

CONTENTS

1 Scope.....	1
2 References.....	1
2.1 Normative References.....	1
2.1.1 Normative Reference List	1
2.1.2 Normative Reference Acquisition	2
2.2 Informative References	2
3 Overview	3
4 MultiAudio Audiobook Extensions	4
4.1 Table of Contents	4
4.2 Description	7
4.2.1 Tag.Identifier	7
4.2.2 Tag.OrdinalNumber	7
4.2.3 Tag.Reserved	7
4.2.4 Tag.Length.....	7
4.2.5 Data	7
5 Audiobook Extendable Structure (AB_ExtStruct)	7
5.1 Tag.....	8
5.2 Offset to Extra Data	8
5.3 Fixed Length Chunk (AB_FixedChunk).....	8
5.3.1 Example	8
5.4 Variable Length Chunk (AB_VariableChunk)	9
5.4.1 Example	9
5.5 Extra Data	10
6 Audiobook Header (AB_Header)	10
6.1 Tag.....	10
6.2 AB_Header Fixed Chunk.....	10
6.2.1 Tag.....	11
6.2.2 Audiobook Spec Major Version.....	11
6.2.3 Audiobook Spec Minor Version	11
6.2.4 Number of Audiobooks (N_Bk)	11
6.2.5 Offset to Playlist Indexes	11
6.3 AB_Header Variable Chunk	11
6.3.1 Tag.....	11
6.3.2 Playlist Indexes	12
6.4 AB_Header Extra Data	12
7 Audiobook Book (AB_Book).....	12
7.1 Tag.....	12
7.2 AB_Book Fixed Chunk.....	12
7.2.1 Tag.....	13
7.2.2 Text Format	13
7.2.3 Flags.....	13
7.2.4 TOC_Playlist Index	13
7.2.5 Offset to Book Title.....	13
7.2.6 Offset to Author	13
7.2.7 Offset to Narrator	13
7.2.8 Offset to Copyright Statement.....	14
7.2.9 Offset to Audiobook UUID	14
7.2.10 Offset to Book ISBN Identifier	14

7.2.11 Offset to Book ISSN Identifier	14
7.2.12 Offset to Book DOI Identifier	14
7.2.13 Offset to Book Index Number	14
7.2.14 Offset to Book Shelf Number	14
7.2.15 Offset to Book Part Number	14
7.2.16 Offset to Other Identifier	14
7.2.17 Offset to First Audiobook Level	14
7.2.18 Offset to Playlist Content Location Identifier	14
7.3 AB_Book Variable Chunk	14
7.3.1 Tag.....	15
7.3.2 Book Title.....	15
7.3.3 Author	15
7.3.4 Narrator	16
7.3.5 Copyright Statement	16
7.3.6 Book UUID	16
7.3.7 Book ISBN Identifier	16
7.3.8 Book ISSN Identifier	16
7.3.9 Book DOI Identifier	16
7.3.10 Book Index Number	16
7.3.11 Book Shelf Number	16
7.3.12 Book Part Number	16
7.3.13 Other Identifier	16
7.3.14 First Audiobook Level	17
7.3.15 Playlist Content Location Identifier	17
7.4 AB_Book Extra Data.....	17
8 Audiobook Level (AB_Level)	17
8.1 Tag.....	18
8.2 AB_Level Fixed Chunk.....	18
8.2.1 Tag.....	18
8.2.2 Text Format	18
8.2.3 Flags.....	19
8.2.4 Number of Sub-Levels.....	19
8.2.5 Level Number	19
8.2.6 TOC_TrackEntry Index.....	19
8.2.7 Offset to Sub-Level Chunk.....	19
8.2.8 Offset to Level Name	19
8.2.9 Offset to Level Description	19
8.2.10 Book Time Offset	19
8.2.11 Time Start Offset.....	19
8.2.12 Time Duration.....	19
8.2.13 Offset to Parent AB_Level	19
8.2.14 Offset to Previous AB_Level	19
8.2.15 Offset to Next AB_Level.....	20
8.3 AB_Level Variable Chunk	20
8.3.1 Tag.....	20
8.3.2 Level Name	20
8.3.3 Level Description.....	20
8.3.4 Sub-Level Chunk.....	20
8.4 AB_Level Extra Data.....	20
9 MPV Audiobook Profiles	20
9.1 XML Schema Namespace Identifier.....	21
9.2 Use of MPV-Music Profile Identifier.....	21
9.3 Audiobook File.....	21
9.4 Audiobook Profile Metadata	22
9.5 MPV-Audiobook Profile Metadata	23

9.6 Dublin Core Metadata Usage.....	29
9.7 Audiobook Profile Using MPV-Music Playlists.....	30
9.7.1 Number of Playlists per MPV File.....	31
9.7.2 Metadata Usage.....	31
9.7.3 Background Usage	33
9.7.4 Foreground Usage.....	33
9.7.5 Related and Rendition Assets	33
9.8 Groups of Assets in Playlists.....	33
9.9 MPV-Audiobook Schema Details	35
9.9.1 XML Schema Definition.....	35
9.9.2 Audiobook Metadata Usage Definitions.....	38
9.10 MPV-Audiobook-Level Metadata Using <mpvp:Album>	39
9.11 <mpvm:AudioWithStills> Music and Stills Asset.....	47
9.12 MPV-Audiobook Extensions to MPV-Core.....	48
9.12.1 Audiobook Manifest File Types & Extensions.....	48
9.12.1.1 File Extension.....	48
9.12.1.2 Apple Macintosh File Type.....	49
9.12.2 Audiobook Manifest MIME Media Type	49
9.13 Media Types and File Formats	49
9.14 MPV-Audiobook and CEA-2003-B.....	49
9.15 CEA-2003-B Binary to XML Formats—Tag Comparison	49
10 Conforming Player and Digital Audiobook File Format Structure Requirements.....	51
10.1 Player Requirements.....	51
10.2 Digital Audiobook File Format Structure Requirements	51
11 Player and Digital Audiobook File Format Structure Functionality Recommendations	52
11.1 Player Functionality Recommendations	52
11.2 Digital Audiobook File Format Structure Functionality Recommendations	52
12 Digital Audiobook File Format Structures and Data Constraints	52
13 Additional Production Options.....	53
13.1 ID3 tagging within MP3 files	53
13.2 M3U Playlists.....	53
13.3 DAISY Production Options	53
14 Certification	53
Annex A Audiobook Structure Example (Informative).....	54
A.1 AB_Level Structures.....	60
A.1.1 Level 0	60
A.1.2 Level 1	60
A.1.3 Level 2	61
A.1.4 Level 3	62
Annex B Encoding Types and Identifiers (Normative).....	64
Annex C Use Case and Implementation Examples: Spoken Word Audio (Informative)	65

Tables

Table 1 Table of Contents Data Structure	5
Table 2 Audiobook MultiAudio Chunk Format.....	7
Table 3 AB_ExtStruct Format	8
Table 4 Example—AB_FixedChunk Structure	8

Table 5 Example—AB_FixedChunk Structure Revised	9
Table 6 Example—AB_FixedChunk Fields	10
Table 7 Example—AB_VariableChunk Data	10
Table 8 AB_Header Format	10
Table 9 AB_Header Fixed Chunk Format	11
Table 10 AB_Header Variable Chunk Format	11
Table 11 AB_Book Format	12
Table 12 AB_Book Fixed Chunk Format	13
Table 13 AB_Book Variable Chunk Format	15
Table 14 AB_Level Structure Required Field Settings	17
Table 15 Playlist Content Location Identifier Field Settings	17
Table 16 AB_Section Format	18
Table 17 AB_Level Fixed Chunk Format	18
Table 18 AB_Level Variable Chunk Format	20
Table 19 Schema Namespace Identifier	21
Table 20 MPV-Music Profile Identifier	21
Table 21 MPV Audiobook Profile Properties	23
Table 22 MPV-Audiobook Profile Metadata	24
Table 23 Dublin Core Metadata (DC-NMF)	29
Table 24 Metadata Properties Usage	33
Table 25 Audiobook Media types for mpv:Audio	49
Table 26 CEA-2003-B Binary & XML Formats—Tag Comparison	50
Table 27 CEA-2003-B Binary & XML Formats—Tag Comparison (Continued)	51
Table 28 Examples of ID3 Tags	53
Table 29 File 1—Example Audio File TOC_TrackEntry Structure	54
Table 30 File 2—Example Audio File TOC_TrackEntry Structure	54
Table 31 File 3—Example Audio File TOC_TrackEntry Structure	55
Table 32 File 4—Example Audio File TOC_TrackEntry Structure	55
Table 33 Example Audiobook TOC_Playlist structure	56
Table 34 Example AB_Book Structure	56
Table 35 Key to Figure 4 Elements	58
Table 36 AB_Level Structure at Level 0	60
Table 37 First AB_Level at Level 1	60
Table 38 Second AB_Level at Level 1	61
Table 39 Third AB_Level at Level 1	61
Table 40 First AB_Level at Level 2—a Child of Table 27	61
Table 41 Second AB_Level at Level 2—a Child of Table 27	62
Table 42 First AB_Level at Level 2—a Child of Table 39	62
Table 43 First AB_Level at Level 2—a Child of Table 39	62
Table 44 First AB_Level at Level 3—a Child of Table 39	63
Table 45 Second AB_Level at Level 3—a Child of Table 39	63

Figures

Figure 1 Example of MPV Music Profile Identifier Usage	21
Figure 2 MPV-Music Playlist Using <mpvp:Album> Example	31
Figure 3 Example of MPV Asset Groups	35
Figure 4 XML Schema for AB:Audiobook Properties	38
Figure 5 MPV-Audiobook Example	48
Figure 6 Example Audiobook Table of Contents	54
Figure 7 Example—Audio Files Partitioning the Table of Contents	55
Figure 8 Example—AB_Level Structure Organization in AB_Book Variable Length Chunk Area	57
Figure 9 View of An Audiobook AB_Level Hierarchy	59
Figure 10: TOC.AUB Example	70
Figure 11: TOC.MUM Example	71

Digital Audiobook File Format and Player Requirements

1 Scope

CEA-2003-C defines requirements and provides recommendations to publishers, software developers, content providers, and hardware manufacturers for the data structure, usability requirements, playback systems and delivery systems for audiobooks in digital file format. It should be noted that throughout CEA-2003-C, the term audiobook is defined as any audio file or collection of audio files of primarily spoken word content that is played in a linear order. Therefore, spoken word audio with occasional music, a narration of newspaper articles, or other similar spoken word audio, would additionally be considered audiobooks under this standard.

2 References

2.1 Normative References

The following standards contain provisions that, through reference in this text, constitute normative provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed in Section 2.1.1.

2.1.1 Normative Reference List

CEA-2004, Audiobook Media and Player Compatibility, December, 2003

ID3 Specifications

ISO 9660:1988, Information processing -- Volume and file structure of CD-ROM for information interchange

ISO/IEC 11172-3:1993, Information technology -- Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s -- Part 3: Audio (MPEG-1 Audio)

ISO/IEC-11578:1996, Information technology -- Open Systems Interconnection -- Remote Procedure Call (RPC), Application Layer

ISO/IEC 14496-3:2002, Information technology -- Coding of audio-visual objects -- Part 3: Audio (MPEG-4 Audio)

ISO/IEC 13818-7:1998, Information technology -- Generic coding of moving pictures and associated audio information -- Part 7: Advanced Audio Coding (AAC)

OSTA MPV Audiobook Profile Specification, Version 1.0, September 1, 2003

NOTE-OSTA MPV Audiobook Profile Specification, Version 1.0 is referred to as "OSTA MPV" Throughout CEA-2003-C

OSTA MultiAudio Specification, Version 1.10, September 16, 2002

NOTE—OSTA MultiAudio Specification, Version 1.10 is referred to as "OSTA MultiAudio" throughout CEA-2003-C.

MPEG-4 Specifications

OGG VORBIS Specifications

WMA Specifications

WAV Specifications

ATRAC3 Standards

NOTE—The audio formats listed above are non-inclusive. Additional audio formats may be identified in the future. Any codec or audio format may be utilized for CEA-2003-C compliant audiobooks. Inclusion, or lack thereof, in the above list does not preclude compliance.

2.1.2 Normative Reference Acquisition

CEA:

Global Engineering Documents, World Headquarters, 15 Inverness Way East, Englewood, CO USA 80112-5776; Phone 800-854-7179; Fax 303-397-2740; Internet: <http://global.ihs.com/>; Email: global@ihs.com

ISO and ISO/IEC:

Global Engineering Documents, World Headquarters, 15 Inverness Way East, Englewood, CO USA 80112-5776; Phone 800-854-7179; Fax 303-397-2740; Internet: <http://global.ihs.com/>; Email: global@ihs.com

OSTA:

Optical Storage Technology Association, 19925 Stevens Creek Blvd., Cupertino, CA USA 95014; Phone 408-253-3695; Fax 408-253-9938; Internet: <http://www.osta.org>; Email: info@osta.org

OGG VORBIS:

Xiph.org Foundation, 1408 Adams St. NE, Albuquerque, NM USA 87110; Internet: <http://www.xiph.org>, <http://www.vorbis.com>; Email: info@xiph.org

WMA, WAV:

Microsoft Corporation, Microsoft Technology Center, 1250 Charleston Road, Mountain View, CA USA 94043; Phone 650-693-8515; Fax 650-693-8557; Internet: <http://www.microsoft.com>; Email: info@microsoft.com

MPEG-4 Audio:

Fraunhofer IIS, Am Wolfsmantel 33, 91058 Erlangen, Germany; Phone +49 (0) 91 31/7 76-0, Fax +49 (0) 91 31/7 76-9 99; Internet: <http://www.iis.fraunhofer.de/amm/techinf/mpeg4/audio.html>; Email: info@iis.fraunhofer.de

ATRAC3:

Sony Corporation of America, 550 Madison Avenue, New York, NY USA 10022; Phone 212-833-6800; Internet: <http://www.sony.net/Products/ATRAC3>; Email: info@sony.com

TwinVQ:

Transform-Domain Weighted Interleave Vector Quantization, Internet: <http://www.twinvq.org>; Email: info@twinvq.org

DOI: International DOI Foundation (Washington&Oxford); Director, International DOI Foundation, Linacre House, Jordan Hill, Oxford, OX2 8DP, UK. Email: contact@doi.org; Web: www.doi.org.

ISBN: U.S. ISBN Agency, 630 Central Avenue, New Providence, NJ, 07974. Email: isbn-san@bowker.com; Web: www.isbn.org.

ISSN: Library of Congress, National Serials Data Program, 101 Independence Ave., SE, Washington, DC, 20540-4160. Email: issn@loc.gov; Web: lcweb.loc.gov/issn.

UUID: *The Open Group – DCE 1.1 Remote Procedure Call (RPC) : Appendix A – Universal Unique Identifier* The Open Group, 44 Montgomery St., Suite 960, San Francisco, CA, 94104-4704; Email: OGSpecs@opengroup.org; Web: www.opengroup.org

2.2 Informative References

- DAISY Specifications, and DAISY 3, ANSI/NISO Z39.86-2002, Specifications for the Digital Talking Book; Daisy Consortium, <http://www.daisy.org/publications/specifications.asp>

- ID3 Specifications, <http://www.id3.org/develop.html>

3 Overview

CEA-2003-C defines the audiobook file format structure for digital audiobook media. Goals in creating CEA-2003-C include compatibility with music industry and multimedia standards as well as effectively presenting and navigating an audiobook. CEA-2003-C is a compilation standard that straddles early binary architectures represented by earlier versions of CEA-2003, and newer XML architectures represented by the Optical Storage Technology Association's (OSTA) MusicPhotoVideo™ MPV). A conforming playback system and conforming audiobook content provides a wonderful and highly functional reading and listening experience. The goal of CEA-2003-C is to create a broad, extensible standard for audiobook publishers, audiobook device manufacturers, and audiobook software developers in order to create the best listening experience for the audiobook consumer. An overview follows.

OSTA MultiAudio defines a binary file to be placed in the root of the file format structure on a CD/MP3. The TOC.MAU serves as the table of contents for the associated MP3 files on a CEA-2003-C compliant CD/MP3. Conforming MultiAudio players read this file to obtain the information the player needs to present the content to the listener.

OSTA MPV defines an eXtensible Markup Language (XML) schema to be placed in the root of the file format structure. The INDEX.AUB serves as the table of contents for the associated audiobook files, regardless of media, codec, or delivery type. Conforming MPV players read this file to obtain the information the player needs to present the content to the listener. Audiobooks conforming to this standard MUST utilize the audiobook-specific file extension, .AUB. In addition, audiobooks conforming to this standard may utilize the same MPV file extension specified by the MPV Music Profile, .MUM, in order to create additional compatibility for MPV-compliant players that do not recognize the audiobook extension.

Similarly, while OSTA MPV was primarily designed for the music industry and CD-ROM media, it provides "extensions" for enhancing the format. CEA-2003-C "extends" OSTA MPV to support audiobooks and does not require a specific media or content delivery system. The audiobook extensions are defined as the MultiAudio binary "TOC.MAU" for CD/MP3 audiobooks, and the MPV XML "INDEX.AUB" for all digital audiobooks.

Therefore, throughout CEA-2003-C, "digital audiobook file format structure" is used to describe the entire audiobook content, regardless of media or method of delivery. "Delivery system" is used to represent any system used for playback of the content, whether it is actual physical media or another type of delivery system, such as downloaded or streaming content. CEA-2003-C defines required MPV data and audiobook-specific *Extra Data* requirements. The conforming player, by reading the INDEX.AUB or TOC.MAU, is presented with the data needed to effectively navigate and play the content in the manner the publisher intended. Conforming players and conforming audio content shall meet the requirements contained in Section 11. In addition, conforming players and conforming audio content are encouraged but not required to conform to the recommendations of Section 11.

The player identifies and reads the INDEX.AUB or TOC.MAU file in the root of the digital audiobook file format structure. A conforming player recognizes the audiobook identifier in the OSTA MPV *Extra Data* field, and processes the audiobook data as defined in Sections 4 through 7 accordingly. OSTA MPV provides specific identifiers to tell the player the type of encoded audio file that is in use. OSTA MPV also provides an "unknown" identifier, which signals the device to determine the type of encoding used. Once an audiobook and its encoding method have been detected, the playback system can reference navigational points and informational fields and present them to the listener. Excessive delays in playback or moving to the next portion of the audiobook should be minimized; excessive delays are not acceptable according to this specification.

One or more audiobooks may reside on the system, but if there is more than one audiobook, they shall each reside in separate directories within the digital audiobook file format structure. The MPV file contains one or more "playlists" for each audiobook. The playlist identifies the correct linear reading order of the files to be played for each book, and the location of the digital audiobook files. A simple digital audiobook file format structure residing on physical media may need only one playlist; a more complex digital

audiobook file format structure that includes choices of delivery, such as local and streaming, may have multiple playlists to coincide with each method of delivery.

Navigational levels are defined by the content publisher within the INDEX.AUB or TOC.MAU file, and may be as simple or as complex as desired. *Extra Data* fields identify the levels and “time offsets” to navigational points within files or file locations. Pointers to the “parent” and “child” locations are also defined within *Extra Data* fields. In this manner, an unlimited amount of levels and navigation points may be identified. Within the binary TOC.MAU file, Level 0 navigation is the top level; for example, the audiobook itself, or a collection of several audiobooks contained in separate folders. Level 1 navigation is the first level of nested pointers defined, and is required for compliance. In a very simple example, a digital audiobook file format structure could contain one audiobook comprised of one digital file. Each of the chapter locations are defined as Level 1 navigation points within the TOC.MAU. The conforming player is able to present the user with Level 1 navigation by moving to each of the defined offsets.

More complex digital audiobook file format structures can be achieved by nesting levels within the INDEX.AUB. A digital audiobook file format structure can contain several layers or levels, with offsets to the next navigation point, parent level, and child level. As a more complex example, a digital audiobook file format structure could contain two audiobooks, each in a separate folder and comprised of ten chapters each, with each chapter encoded as a separate digital file. Navigation would include pointers to each separate audiobook, and enable the player to choose between one or the other. Navigation markers would point to the beginning of each chapter, allowing navigation from chapter to chapter within the audiobook. Nested levels would point to pages within the chapters.

NOTE—Books, chapters and pages are used as Level name examples; however, there is no naming requirement for levels in order to be compliant with CEA-2003-C.

The only constraint on the audio publisher is to encode MP3 files with a “fixed bit rate” because variable bit rate encoding makes it impossible for low-powered players to calculate the data offset from the time offset.

NOTE--Audiobook content providers are strongly encouraged to name the audio files using an ASCII collated sequence. This means that the unaware player would still play the audio files in order and the navigation from file to file would be whatever design the audio publisher chose for structuring the files, i.e. one chapter per file would yield chapter navigation; one section per file would yield section navigation. The correct linear reading order would then most likely be preserved.

There is audiobook-specific content, such as title, author, and narrator, as well as time information that allows players to calculate elapsed time and time remaining. *Extra Data* fields associated with each file provide the player with this information so that it can be presented to the listener. Additional enhancements are encouraged, but not required with *Level Class Identifiers* and the separate *Bookmark.AUB* file for rewriteable media. The *Level Class Identifiers* will allow a playback device to handle distinct audiobook levels in customized ways, depending on the type of content, or sub grouping, of those levels. The *Bookmark.AUB* file contains provisions for both an autoresume point, as well as user-defined bookmarking capabilities, and is encouraged for transfer from one device to another of such. Playback device designers may also consider incorporating multi-user enhancements with bookmarking capabilities.

For a player that reads binary files, Sections 4-8 detail the MultiAudio Audiobook Extensions, Headers, Playlists and Levels in the MultiAudio Format. For a player that is capable of reading XML, Section 10 discusses in detail the MPV Schema for audiobooks in the MPV format. For players that can read both binary files and XML schema, the XML should be used.

4 MultiAudio Audiobook Extensions

This section defines audiobook extended data structures based on OSTA MultiAudio.

4.1 Table of Contents

The data structure for the table of contents shall be as defined in Table 1.

Section	Description
4.2	Description
Table 2	audiobook MultiAudio Chunk
4.2.1	Tag.Identifier
4.2.2	Tag.OrdinalNumber
4.2.3	Tag.Reserved
4.2.4	Tag.Length
4.2.5	Tag.Data
5	Audiobook Extendable Structure (AB_ExtStruct)
5.1	Tag
5.2	Offset to <i>Extra Data</i>
5.3	Fixed Length Chunk (AB_FixedChunk)
5.3.1	Example
5.4	Variable Length Chunk (AB_VariableChunk)
5.4.1	Example
5.5	Extra Data
6	Audiobook Header (AB_Header)
6.1	Tag
6.2	AB_Header Fixed Chunk
6.2.1	Tag
6.2.2	Audiobook Spec Major Version
6.2.3	Audiobook Spec Minor Version
6.2.4	Number of Audiobooks
6.2.5	Offset to Playlist Indexes
6.3	AB_Header Variable Chunk
6.3.1	Tag
6.3.2	Playlist Indexes
6.4	Extra Data
7	Audiobook (AB_Book)
7.1	Tag
7.2	AB_Book Fixed Chunk
7.2.1	Tag
7.2.2	Text Format
7.2.3	Flags
7.2.4	<i>TOC Playlist</i> Index
7.2.5	Offset to Book Title
7.2.6	Offset to Author
7.2.7	Offset to Narrator
7.2.8	Offset to Copyright Statement
7.2.9	Offset to Audiobook UUIDs
7.2.10	Offset to Book ISBN Identifier

Table 1 Table of Contents Data Structure

Section	Description
7.2.11	Offset to Book ISSN Identifier
7.2.12	Offset to Book DOI Identifier
7.2.13	Offset to Book Index Number
7.2.14	Offset to Book Shelf Number
7.2.15	Offset to Book Part Number
7.2.16	Offset to Other Identifier
7.2.17	Offset to First Audiobook Level
7.2.18	Offset to Playlist Content Location Identifier
7.3	AB_Book Variable Chunk
7.3.1	Tag
7.3.2	Book Title
7.3.3	Author
7.3.4	Narrator
7.3.5	Copyright Statement
7.3.6	Book UUID
7.3.7	Book ISBN Identifier
7.3.8	Book ISSN Identifier
7.3.9	Book DOI Identifier
7.3.10	Book Index Number
7.3.11	Book Shelf Number
7.3.12	Book Part Number
7.3.13	Other Identifier
7.3.14	First Audiobook Level
7.3.15	Playlist Content Location Identifier
7.4	Extra Data
8	Audiobook Level (AB_Level)
8.1	Tag
8.2	AB_Level Fixed Chunk
8.2.1	Tag
8.2.2	Text Format
8.2.3	Flags
8.2.4	Number of Sub-Levels
8.2.5	Level Number
8.2.6	First TOC_TrackEntry Index
8.2.7	Offset to Sub-Level Chunk
8.2.8	Offset to Level Name
8.2.9	Offset to Level Description
8.2.10	Book Time Offset
8.2.11	Time Start Offset
8.2.12	Time Duration
8.2.13	Offset to Parent AB_Level
8.2.14	Offset to Previous AB_Level
8.2.15	Offset to Next AB_Level
8.3	AB_Level Variable Chunk
8.3.1	Tag
8.3.2	Level Name
8.3.3	Level Description
8.3.4	Sub-Level Chunk
8.4	Extra Data

Table 1 Table of Contents Data Structure (Continued)

4.2 Description

All structures defined in CEA-2003-B were extensions of OSTA MultiAudio. These structures are retained in CEA-2003-C. Each structure resides in the *Extra Data* area of the MultiAudio structures so that it does not interfere with the original intent of OSTA MultiAudio, but enhances OSTA MultiAudio with additional metadata that applies to a collection of audiobooks. All audiobook structures are contained within a MultiAudio *Chunk* and use the *Extendable Structure* format described in Section 5.

All audiobook structures are contained with a MultiAudio *Chunk* structure. The *Chunk* shall have the values indicated in Table 2.

#	Structure	RBP	Length	Name	Contents
1	Tag	0	4	Identifier	9FFF0000h
2		4	2	Ordinal Number	Not Defined
3		6	2	Reserved	0
4		8	4	Length	UInt32
5	Data	12	Variable	audiobook Structure	AB_ExtStruct

Table 2 Audiobook MultiAudio Chunk Format

4.2.1 Tag.Identifier

The identifier value should uniquely identify the MultiAudio *Chunk* as being an audiobook structure. The value 9FFF0000h shall designate this is an audiobook structure. A parser would then look for a *Chunk* with this identifier value in a MultiAudio structures *Extra Data* area to determine if audiobook structure is defined.

Since OSTA MultiAudio defines the range [80000000h – FFFFFFFFh] for public use, this means that other specifications may also use the value 9FFF0000h for their *Chunk* identifier. To ensure that the *Chunk* contains an audiobook structure, a parser shall then check the first 4 bytes of the *Chunk's* data area for a valid audiobook structure identifier. The first 4 bytes of the data area shall contain an audiobook structure *Tag.Identifier* field, which shall be defined by each audiobook structure.

4.2.2 Tag.OrdinalNumber

The ordinal number of the chunk shall be the index of the chunk within the array of chunk structures that make up a MultiAudio structure *Extra Data* area.

4.2.3 Tag.Reserved

The tag.reserved field shall be reserved for future OSTA use. This field is not defined at present, and shall not contain data.

4.2.4 Tag.Length

The Tag.Length field shall contain the length of the *Chunk* structure within the length of the *Chunk.Tag* and *Chunk.Data* area.

4.2.5 Data

The data area of the *Chunk* structure shall contain a valid audiobook structure, which shall follow the *AB_ExtStruct* format.

5 Audiobook Extendable Structure (AB_ExtStruct)

The audiobook extendable structure defines a binary structure that allows easy extendability. This allows new fields to be added to the structure, while remaining backwards compatible with previous versions of the structure. All audiobook structures shall conform to the Extendable Structure format contained in Table 3.

#	RBP	Length	Name	Contents
1	0	12	Tag	tag
2	12	4	Offset to Extra Data	UInt32
3	16	L_Fc	Fixed Length Chunk	AB_FixedChunk
4	16 + L_Fc	L_Vc	Variable Length Chunk	AB_VariableChunk
5	16 + L_Fc + L_Vc	L_Ex	Extra Data	Extra Data

Table 3 AB_ExtStruct Format

5.1 Tag

Tag.Identifier field shall be set to a unique identifier that identifies the audiobook structure.

Tag.OrdinalNumber shall be set to the ordinal number of the structure. By default, the ordinal number should be set to the value 0, unless the structure is part of a list of structures.

Tag.Length shall be set to the length of the *Extendable Structure* in bytes. Since the structure is a *MultiAudio Chunk* structure, this length shall be evenly divisible by 4.

5.2 Offset to Extra Data

To ensure that the *AB_ExtStruct* can be further extended, an extra data area is included. If there is extra data included in the structure, the offset to the extra data area should contain the offset in bytes from the start of the *AB_ExtStruct* to the start of the *Extra Data Chunk*. If there is no *Extra Data Chunk*, this field should be set to the value 0.

5.3 Fixed Length Chunk (AB_FixedChunk)

The Fixed Length Chunk is defined as a Chunk structure, and provides for the extendibility of the *AB_ExtStruct*. The following values should be set:

AB_FixedChunk.Tag.Identifier shall be set to the hex value 9FFF0001h to designate that it is an *AB_FixedChunk*.

AB_FixedChunk.Tag.OrdinalNumber shall be set to the value 0.

AB_FixedChunk.Tag.Length shall be set to the length of the *AB_FixedChunk* in bytes. The length of this structure is the key to the extendibility of the structure. As new fields are added to the structure, this length grows. Software or devices that interpret this structure should first check this length to identify the version of the structure. To remain backwards compatible, fields defined in this structure should never be removed or redefined. All new fields should be added to the end of the structure.

Since the structure is defined as a *MultiAudio Chunk*, the length shall be evenly divisible by 4.

5.3.1 Example

The following is an example of how the structure is extended and interpreted by parsers. An example of the *AB_FixedChunk* structure is defined in Table 4.

#	RBP	Length	Name	Contents
1	0	12	<i>AB_FixedChunk</i> Tag	Tag
2	12	4	Field1	UInt32
3	16	2	Field2	UInt16
4	18	2	Field3	UInt16

Table 4 Example—AB_FixedChunk Structure

The *AB_FixedChunk.Tag.Length* field of this structure would be set to the value 20, which designates a length of 20 bytes. All devices and software that can parse this structure would understand the values in Fields1-3, and that the total length of the structure is 20 bytes.

At a later date, a new field is added to the structure. The structure then is described in Table 5.

#	RBP	Length	Name	Contents
1	0	12	<i>AB_FixedChunk</i> Tag	Tag
2	12	4	Field1	UInt32
3	16	2	Field2	UInt16
4	18	2	Field3	UInt16
5	20	4	Field4	UInt32

Table 5 Example—AB_FixedChunk Structure Revised

The previous software or device that was designed to handle Fields1-3 would ignore the new Field4. New software or devices that are designed to read the new field would still be able to interpret the previous three fields. If the newer software or device were to parse the previous, older structure, it would first check the length of the *FixedChunk.Tag.Length* field and realize that this structure does not define any fields greater than Field3, since the length would be only 20 bytes. Therefore, the newer software or device should gracefully handle this situation and initialize Field4 with a default value, or ignore it completely.

5.4 Variable Length Chunk (AB_VariableChunk)

The Variable Length Chunk (*AB_VariableChunk*) structure is designed to work with the Fixed Length Chunk (*AB_FixedChunk*) to handle field values that are of a variable length, such as a string. The *AB_FixedChunk* would define what values are stored in the *AB_VariableChunk* by providing offsets to the values in the *AB_VariableChunk*.

The *AB_VariableChunk* is defined as a MultiAudio *Chunk* structure. Therefore, the length of the structure shall be evenly divisible by 4 to ensure that it lies on a 4-byte aligned memory boundary. To ensure this alignment property, all fields in the *AB_VariableChunk* shall have a length that is evenly divisible by 4. Therefore, a field shall be padded with bytes to ensure that its length is evenly divisible by 4. All padding bytes shall be set to the value 0. To calculate the length of the padding, use the following equation:

$$\text{Number of Padding bytes} = \text{int}((\text{TempLen} + 3)/4) * 4 - \text{TempLen}$$

where

TempLen = Size in bytes of the field before padding

For example, if the *AB_VariableChunk* contains two strings (String1 = "ABC\0") and (String2 = "ABCDEF\0"), where the character '\0' designates a termination character, the first string (String1) would not have to be padded, since its length is 4 bytes. The second string (String2) would have to be padded with a single additional byte to change its length from 7 bytes to 8.

The *AB_VariableChunk* shall have the following values set:

- AB_VariableChunk.Tag.Identifier* should be set to the hex value 9FFF0002h to designate that this is a variable length chunk.
- AB_VariableChunk.Tag.OrdinalNumber* should be set to the value 1.
- AB_VariableChunk.Tag.Length* should be set to the length of this structure including the tag.

The following is example of how the Fixed Length Chunk and the Variable Length Chunk work together to provide an extendable structure.

5.4.1 Example

In this example, the *AB_FixedChunk* defines the fields in Table 6.

#	RBP	Length	Name	Contents
1	0	12	<i>AB_FixedChunk</i> Tag	tag
2	12	2	String1 offset	UInt32
3	16	2	Variable Field1 offset	UInt16
4	18	2	Variable Field1 Length (L_F1)	UInt16

Table 6 Example—*AB_FixedChunk* Fields

In this example, the *AB_VariableChunk* would then contain the data indicated in Table 7.

#	RBP	Length	Name	Contents
1	0	12	<i>AB_VariableChunk</i> Tag	tag
2	12	L_Str1	String1 (NULL terminated)	byte
3	12 + L_Str1	L_F1	Variable Field1	byte

Table 7 Example—*AB_VariableChunk* Data

The *AB_VariableChunk* contains two fields, String1 and Variable Field1. Since String1 is a NULL terminated string, there is no need to provide a length in bytes of the string, since a parser would recognize the NULL character as being the end of the data. The Variable Field1 contains an arbitrary value of length L_F1 in bytes. Since this variable does not have the luxury of the NULL termination, the length of the field is included in the *AB_FixedChunk*.

The *AB_FixedChunk* would contain offsets from the start of the *AB_FixedChunk* to the data inside the *AB_VariableChunk*. If String1 were not included in the *AB_VariableChunk*, the offset in the *AB_FixedChunk* would be set to 0. Therefore, the *AB_VariableChunk* does not define how the data is ordered, since it can be placed in an arbitrary order. This is defined in the *AB_FixedChunk*. Therefore, the *AB_FixedChunk* is similar to the index of a book, and the *AB_VariableChunk* contains the actual data, which would be the pages and chapters of the book.

5.5 Extra Data

Extra Data shall be defined as a *Chunk* containing an array of *Chunk* based structures. This area is used to further extend the structure.

6 Audiobook Header (AB_Header)

The audiobook Header is located in the *Extra Data* field of the OSTA MultiAudio *TOC_Header* structure. The *AB_Header* is contained within a *Chunk* structure, as described in Section 4.2. The ordinal number of the *Chunk* shall not be fixed. The *AB_Header* structure is defined as an *AB_ExtStructure*, as defined in Section 4.2. The *AB_Header* can be used to determine whether a TOC file contains any audiobooks, as well as where they are located. The *AB_Header* format shall be as indicated in Table 8.

#	RBP	Length	Name	Contents
1	0	12	Tag	Tag
2	12	4	Offset to Extra Data	UInt32
3	16	L_Fc	<i>AB_Header</i> Fixed Chunk	<i>AB_FixedChunk</i>
4	16 + L_Fc	L_Vc	<i>AB_Header</i> Variable Chunk	<i>AB_VariableChunk</i>
5	16 + L_Fc + L_Vc	L_Ex	Extra Data	ExtraData

Table 8 *AB_Header* Format

6.1 Tag

The *Tag.Identifier* shall be set to the hex value 44484241h (ABHD) to designate an *AB_Header* structure.

The *Tag.OrdinalNumber* shall be set to the value 0.

The *Tag.Length* shall be set to the length of the *AB_Header* structure, which includes the length of the Tag.

6.2 *AB_Header* Fixed Chunk

The *AB_Header Fixed Chunk* shall contain the format defined in Table 9.

#	RBP	Length	Name	Contents
1	0	12	Tag	tag
2	12	2	Audiobook Spec Major Version	UInt16
3	14	2	Audiobook Spec Minor Version	UInt16
4	16	2	Number of Audiobooks (N_Bk)	UInt16
5	18	2	Offset to Playlist Indexes	UInt16

Table 9 AB_Header Fixed Chunk Format

6.2.1 Tag

The *Tag.Identifier* sub-field shall be set to the hex value 9FFF0001h to designate that it is an *AB_FixedChunk*.

The *Tag.OrdinalNumber* sub-field shall be set to the value 0.

The *Tag.Length* shall be set to the length of the *AB_Header Fixed Chunk* structure. CEA-2003-C specifies this length as 20 bytes.

6.2.2 Audiobook Spec Major Version

This field shall contain the major version number of the audiobook specification. If the major version number changes, the new version of the specification is not required to be backward compatible with the previous versions of the spec.

For CEA-2003-C, this field shall be set to the hex value 0001h, which designates that this is version 1.X and has structure compliant with CEA-2003-C.

NOTE—Since CEA-2003-C is backward compatible with CEA-2003, CEA-2003-A and CEA-2003-B, version 1.0 is used for all versions of the standard.

6.2.3 Audiobook Spec Minor Version

This field shall contain the minor version number of the audiobook specification. If the minor version number changes, the newer version of the specification is required to be backwards compatible with the previous versions of the spec.

For CEA-2003-C, this field shall be set to the hex value 0001h, which designates that this is version X.1.

6.2.4 Number of Audiobooks (N_Bk)

This field specifies the number of audiobooks that can be found in the MultiAudio TOC file.

6.2.5 Offset to Playlist Indexes

The Offset to Playlist Indexes shall be defined as Offset from the start of the *AB_Header* structure into the *AB_Header Variable Chunk* that contains an ordered list of MultiAudio *TOC_Playlist* indexes. See Section 6.2.

6.3 AB_Header Variable Chunk

The *AB_Header Variable Chunk* shall contain the fields defined in Table 10.

#	RBP	Length	Name	Contents
1	0	12	Tag	Tag
2	12	N_Bk * 2	Playlist Indexes	UInt16

Table 10 AB_Header Variable Chunk Format

6.3.1 Tag

Tag.Identifier sub-field should be set to the hex value 9FFF0002h to designate that this is an *AB_VariableChunk*.

The *Tag.OrdinalNumber* sub-field shall be set to the value 1.

The *Tag.Length* shall be set to the length of the *AB_Header Variable Chunk* structure in bytes.

6.3.2 Playlist Indexes

The Playlist Indexes shall contain a list of *UInt16* values that represent the indexes of the MultiAudio *TOC_Playlist* structures, which contain an *AB_Book* structure, defined in their *Extra Data* field. Each audiobook on the media shall have an associated *TOC_Playlist* structure, with a single *AB_Book* structure found in the *Extra Data* field. See Section 7 for more information.

TOC_Playlist indexes are used instead of byte offsets to the *AB_Book* structures to ensure that TOC can change without breaking the audiobook structures. Since parsers that are not audiobook aware may edit the contents of the TOC, such as adding new *TOC_TrackEntry* structures, it may cause the byte offsets to become invalid. Therefore, it is up to the audiobook parser to map the associated indexes to the appropriate byte offsets in the file.

Since the *Playlist Indexes* field may have a length that it is not evenly divisible by 4, it may be necessary to include padding. See Section 5.4 for more information about padding.

6.4 AB_Header Extra Data

AB_Header Extra Data shall be defined as a *Chunk* containing an array of *Chunk* based structures. This area is used to further extend the *AB_Header* structure.

7 Audiobook Book (AB_Book)

An audiobook is a collection of audio files that represent logical units of primarily spoken word audio. The audiobook may be represented by a single audio file, or by multiple files that constitute smaller units of the entire book. To play an audiobook, these units are played in order from first to last. Therefore, an audiobook can be represented as an ordered playlist of audio files.

Since a book is a playlist of files, the OSTA MultiAudio *TOC_Playlist* structure is used to represent an audiobook. Any MultiAudio compliant player could then play an audiobook by simply recognizing it as a *TOC_Playlist* and playing it. This allows players that do not recognize CEA-2003-C data structures to still play an audiobook with minimal navigation ability. To extend the navigation, as well as provide additional metadata, the *AB_Book* structure was designed.

Every audiobook on the media shall be represented in the MultiAudio TOC as a separate *TOC_Playlist*. In the *TOC_Playlist* structure for each audiobook, there shall be one *AB_Book* structure located in the *Extra Data* area.

The *AB_Book* structure is defined as an *AB_ExtStructure* that can be found in a *Chunk* structure (as defined in Section 4.2 within the *Extra Data* field of a MultiAudio *TOC_Playlist*). The ordinal number of this *Chunk* is not defined.

#	RBP	Length	Name	Contents
1	0	12	Tag	tag
2	12	4	Offset to Extra Data	UInt32
3	16	L_Fc	<i>AB_Book</i> Fixed Chunk	<i>AB_FixedChunk</i>
4	16 + L_Fc	L_Vc	<i>AB_Book</i> Variable Chunk	<i>AB_VariableChunk</i>
5	16 + L_Fc + L_Vc	L_Ex	Extra Data	ExtraData

Table 11 *AB_Book* Format

7.1 Tag

The *Tag.Identifier* shall be set to the hex value 4B424241h (ABBK) to designate an *AB_Book* structure.

The *Tag.OrdinalNumber* shall be set to the value 0.

The *Tag.Length* shall be set to the length of the *AB_Book* structure, which includes the length of the Tag.

7.2 AB_Book Fixed Chunk

The *AB_Book Fixed Chunk* shall contain the format indicated in Table 12.

#	RBP	Length	Name	Contents
1	0	12	Tag	tag
2	12	2	Text Format	UInt16
3	14	2	Flags	UInt16
4	16	2	<i>TOC_Playlist</i> Index	UInt16
5	18	2	Offset to Book Title	UInt16
6	20	2	Offset to Author	UInt16
7	22	2	Offset to Narrator	UInt16
8	24	2	Offset to Copyright Statement	UInt16
9	26	2	Offset to Audiobook UUID	UInt16
10	28	2	Offset to Book ISBN Identifier	UInt16
11	30	2	Offset to Book ISSN Identifier	UInt16
12	32	2	Offset to Book DOI Identifier	UInt16
13	34	2	Offset to Book Index Number	UInt16
14	36	2	Offset to Book Shelf Number	UInt16
15	38	2	Offset to Book Part Number	UInt16
16	40	2	Offset to Other Identifier	UInt16
17	42	2	Offset to First Audiobook Level	UInt16
18	44	4	Offset to Playlist Content Location Identifier	UInt32

Table 12 AB_Book Fixed Chunk Format

7.2.1 Tag

The *Tag.Identifier* sub-field shall be set to the hex value 9FFF0001h to designate that it is an *AB_FixedChunk*.

The *Tag.OrdinalNumber* sub-field shall be set to the value 0.

Tag.Length shall be set to the length of the *AB_Book Fixed Chunk* structure in bytes. For CEA-2003-C, the length is 48 bytes.

7.2.2 Text Format

If this field is set to 0, then all text fields shall be interpreted as ASCII strings.

If this field is set to 1, then all text fields shall be interpreted as UNICODE strings.

7.2.3 Flags

This field shall be set to the value 0 for CEA-2003-C.

7.2.4 TOC_Playlist Index

This field shall contain the index of the *TOC_Playlist* in the OSTA MultiAudio TOC which represents the audiobook and which contains the *AB_Book* structure.

7.2.5 Offset to Book Title

The Offset to Book Title shall be defined as Offset from the start of the *AB_Book* structure to the title of the book within the *AB_Book Variable Chunk* field. Since the *Book Title* is a required field, this offset shall always be set. See Section 7.3.2.

7.2.6 Offset to Author

The Offset to Author shall be defined as Offset from the start of the *AB_Book* structure to the author of the book within the *AB_Book Variable Chunk* field. Since the *Author* is a required field, this offset shall always be set. See Section 7.3.3.

7.2.7 Offset to Narrator

The Offset to Narrator shall be defined as the Offset from the start of the *AB_Book* structure to the narrator of the book within the *AB_Book Variable Chunk* field. Since the *Narrator* is a required field, this offset shall always be set. See Section 7.3.4.

7.2.8 Offset to Copyright Statement

The Offset to Copyright Statement shall be defined as Offset from the start of the *AB_Book* structure to the copyright statement of the book within the *AB_Book Variable Chunk* field. If the book has no copyright statement, this field should be set to the value 0. See Section 7.3.5.

7.2.9 Offset to Audiobook UUID

The Offset to Audiobook UUID (universally unique identifier) shall be defined as the Offset from the start of the *AB_Book* structure to the audiobooks unique identifier. If the audiobook does not have a unique identifier, this field shall be set to the value 0. See Section 7.3.6.

7.2.10 Offset to Book ISBN Identifier

The Offset to Book ISBN (International Standard Book Number) Identifier shall be defined as the Offset from the start of the *AB_Book* structure to the ISBN of the book within the *AB_Book Variable Chunk* field. If the book has no ISBN, this field should be set to the value 0. See Section 7.3.7.

7.2.11 Offset to Book ISSN Identifier

The Offset to Book ISSN (International Standard Serial Number) Identifier shall be defined as the Offset from the start of the *AB_Book* structure to the ISSN identifier string of the book within the *AB_Book Variable Chunk* field. If the book has no ISSN, this field should be set to the value 0. See Section 7.3.9.

7.2.12 Offset to Book DOI Identifier

The Offset to Book DOI (Digital Object Identifier) Identifier shall be identified as the Offset from the start of the *AB_Book* structure to the DOI identifier string of the book within the *AB_Book Variable Chunk* field. If the book has no DOI, this field should be set to the value 0. See Section 7.3.9.

7.2.13 Offset to Book Index Number

The Offset to Book Index Number shall be defined as the Offset from the start of the *AB_Book* structure to the book index number within the *AB_Book Variable Chunk* field. If the book has no index number, this field should be set to the value 0. See Section 7.3.10.

7.2.14 Offset to Book Shelf Number

The Offset to Book Shelf Number shall be defined as Offset from the start of the *AB_Book* structure to a book shelf number string within the *AB_Book Variable Chunk* field. If the book has no shelf number, this field should be set to the value 0. See Section 7.3.11.

7.2.15 Offset to Book Part Number

The Offset to Book Part Number shall be defined as the Offset from the start of the *AB_Book* structure to the book part number string within the *AB_Book Variable Chunk* field. If the book has no ISSN, this field should be set to the value 0. See Section 7.3.12.

7.2.16 Offset to Other Identifier

The Offset to Other Identifier shall be defined as the Offset from the start of the *AB_Book* structure to an additional, undefined, string identifier for the book within the *AB_Book Variable Chunk* field. If the book has no additional identifier, this field should be set to the value 0. See Section 7.3.13.

7.2.17 Offset to First Audiobook Level

The Offset to First Audiobook Level shall be defined as the Offset from the start of the *AB_Book structure* to the first *AB_Level* structure within the *AB_Book Variable Chunk* field. All audiobooks shall contain one *AB_Level* structure to define the entire audiobook, so this offset shall always be set. See Section 7.3.14.

7.2.18 Offset to Playlist Content Location Identifier

The Offset to Playlist Content Location Identifier shall be defined as Offset from the start of the *AB_Book* structure to the Playlist Content Location Identifier within the *AB_Book Variable Chunk* field. If the field is not present or null, this field should be set to the value 0. See Section 7.3.14.

7.3 AB_Book Variable Chunk

The *AB_Book Variable Chunk* shall contain the format indicated in Table 13.

#	RBP	Length	Name	Contents
1	0	12	Tag	tag
2	12	L_Bt	Book Title	byte
3	12 + L_Bt	L_Au	Author	byte
4	12 + L_Bt + L_Au	L_Na	Narrator	byte
5	12 + L_Bt + L_Au + L_Na	L-Cs	Copyright Statement	byte
6	12 + L_Bt + L_Au + L_Na + L-Cs	L_Ui	Book UUID	byte
7	12 + L_Bt + L_Au + L_Na + L-Cs + L_Ui	L_Is	Book ISBN Identifier	byte
8	12 + L_Bt + L_Au + L_Na + L-Cs + L_Ui + L_Is	L_In	Book ISSN Identifier	byte
9	12 + L_Bt + L_Au + L_Na + L-Cs + L_Ui + L_Is + L_In	L_Do	Book DOI Identifier	byte
10	12 + L_Bt + L_Au + L_Na + L-Cs + L_Ui + L_Is + L_In + L_Do	L_In	Book Index Number	byte
11	12 + L_Bt + L_Au + L_Na + L-Cs + L_Ui + L_Is + L_In + L_Do + L_In	L_Sn	Book Shelf Number	byte
12	12 + L_Bt + L_Au + L_Na + L-Cs + L_Ui + L_Is + L_In + L_Do + L_In + L_Sn	L_Pn	Book Part Number	byte
13	12 + L_Bt + L_Au + L_Na + L-Cs + L_Ui + L_Is + L_In + L_Do + L_In + L_Sn + L_Pn	L_Oi	Other Identifier	byte
14	12 + L_Bt + L_Au + L_Na + L-Cs + L_Ui + L_Is + L_In + L_Do + L_In + L_Sn + L_Pn + L_Oi	L_Sc	First Audiobook Level	byte
15	12 + L_Bt + L_Au + L_Na + L-Cs + L_Ui + L_Is + L_In + L_Do + L_In + L_Sn + L_Pn + L_Oi + L_Li + L_Sc	L_Li	Content Location Identifier	Uint32

Table 13 AB_Book Variable Chunk Format

7.3.1 Tag

The *Tag.Identifier* sub-field should be set to the hex value 9FFF0002h to designate that this is an *AB_VariableChunk*.

The *Tag.OrdinalNumber* sub-field shall be set to the value 1.

The *Tag.Length* shall be set to the length of the *AB_Book Variable Chunk* structure in bytes.

7.3.2 Book Title

Every audiobook shall contain a book title field. This field shall contain the *Book Title* according to the character set defined in the *Text Format* field located in the *AB_Book Fixed Chunk*. If there is no book title, the book title field shall be filled with a string termination character. It may be necessary to pad this field. See Section 5.4 for more information on padding.

If the *AB_Book* structure is located in a MultiAudio *TOC_Playlist* structure, the *Book Title* shall be the same string as *Playlist Name* field in the *TOC_Playlist*.

If the *AB_Book* structure is located in a MultiAudio *TOC_TrackEntry* structure, the *Book Title* shall be the same string as the *Album Name* field in the *TOC_TrackEntry*.

7.3.3 Author

Every audiobook shall contain an author field. This field shall contain the *Author* according to the character set defined in the *Text Format* field located in the *AB_Book Fixed Chunk*. If there is no author, the *Author* field shall be filled with a string termination character. It may be necessary to pad this field. See Section 5.4 for more information on padding.

If the *AB_Book* structure is located in a MultiAudio *TOC_TrackEntry* structure, the *Author* shall be the same string as the *Songwriter Name* field in the *TOC_TrackEntry*.

7.3.4 Narrator

Every audiobook shall contain a narrator field. This field shall contain the *Narrator* according to the character set defined in the *Text Format* field located in the *AB_Book Fixed Chunk*. If there is no author, the *Narrator* field shall be filled with a string termination character. It may be necessary to pad this field. See Section 5.4 for more information on padding.

If the *AB_Book* structure is located in a MultiAudio *TOC_TrackEntry* structure, the *Narrator* shall be the same string as the *Performer Name* field in the *TOC_TrackEntry*.

7.3.5 Copyright Statement

This field shall contain the copyright statement according to the character set defined by the *Text Format* field in the *AB_Book Fixed Chunk* structure. If there is no copyright statement, this field shall not be present. It may be necessary to pad this field. See 5.4 for more information on padding.

7.3.6 Book UUID

A UUID format was defined in the Open Software Foundation's Distributed Computer Environment RPC standard also available as ISO-11578, which defines UUIDs in an appendix.

This field contains a UUID in a 36-byte string format according to that specification. If this field is present, the UUID is not NULL (00h) terminated. Since the field shall be exactly 36 bytes, there is no need for padding.

An audiobook need not include this field. This field is recommended, although not required, so devices, as well as publishers, can uniquely identify a digital audiobook.

7.3.7 Book ISBN Identifier

This field shall contain the ISBN of the audiobook formatted as a string of *d-characters*, as defined by the MultiAudio specification. If the book has no ISBN, this field shall not be present. It may be necessary to pad this field. See 5.4 for more information on padding.

7.3.8 Book ISSN Identifier

This field shall contain the ISSN of the audiobook formatted as a string of *d-characters*, as defined by the MultiAudio specification. If the book has no ISSN, this field shall not be present. It may be necessary to pad this field. See Section 5.4 for more information on padding.

7.3.9 Book DOI Identifier

This field shall contain the DOI of the audiobook formatted as a string of *d-characters*, as defined by the MultiAudio specification. If the book has no DOI, this field shall not be present. It may be necessary to pad this field. See Section 5.4 for more information on padding.

7.3.10 Book Index Number

This field shall contain the audiobook index number formatted as a string of *d-characters*, as defined by the MultiAudio specification. If the book has no index number, this field shall not be present. It may be necessary to pad this field. See Section 5.4 for more information on padding.

7.3.11 Book Shelf Number

This field shall contain the audiobook shelf number formatted as a string of *d-characters*, as defined in OSTA MultiAudio. If the book has no shelf number, this field shall not be present. It may be necessary to pad this field. See Section 5.4 for more information on padding.

7.3.12 Book Part Number

This field shall contain the audiobook part number formatted as a string of *d-characters*, as defined by the MultiAudio specification. If the book has no part number, this field shall not be present. It may be necessary to pad this field. See Section 5.4 for more information on padding.

7.3.13 Other Identifier

This field shall contain an additional audiobook identifier formatted as a string of *d-characters*, as defined by the MultiAudio specification. This additional identifier is not defined by the specification, and can be set to any value that helps uniquely identify an audiobook. If the book has no additional identifier, this field

shall not be present. It may be necessary to pad this field. See Section 5.4 for more information on padding.

7.3.14 First Audiobook Level

Every *AB_Book* structure shall contain one *AB_Level* structure that defines the entire book. The book is then divided into smaller sub-levels of this level. The *AB_Level* structure that represents the entire audiobook shall have the fields set as indicated in Table 14.

Section #	AB_Level Field	Value
8.2.1	Tag.OrdinalNumber	0
8.2.5	Level Number	0
8.2.13	Offset to Parent <i>AB_Level</i>	. 0
8.2.10	Book Time Offset	0
8.2.11	Time Start Offset	0
8.2.12	Time Duration	Length of audiobook in milliseconds
8.2.14	Offset to Previous <i>AB_Level</i>	0
8.2.15	Offset to Next <i>AB_Level</i>	0

Table 14 AB_Level Structure Required Field Settings

See Section 8 for more information on the *AB_Level* structure.

7.3.15 Playlist Content Location Identifier

This field shall contain the Playlist Content Location Identifier. This identifier is a number indicating the location of files for this playlist (locally, online as downloadable content, online as streaming content or any combination). This number can be derived from Table 15. If the field is null or not present, value 0 (local) should be assumed.

Playlist Content Location Identifier	Value
Local	0
Undefined	1
No content	2
Download content	3
Streaming content	4
Mixed content – Local/Download	5
Mixed content – Local/Streaming	6
Mixed content – Download/Streaming	7
Mixed content – Local/Download/Streaming	8

Table 15 Playlist Content Location Identifier Field Settings

7.4 AB_Book Extra Data

AB_Book Extra Data shall be defined as a *Chunk* containing an array of *Chunk* based structures. This area is used to further extend the *AB_Book* structure.

8 Audiobook Level (AB_Level)

An *AB_Level* is a new structure that is not present in OSTA MultiAudio. An *AB_Level* represents a logical unit of an audiobook, with a start time and duration in milliseconds of the audio file or group of audio files. Each level can represent an infinite number of smaller sub-level sections. Therefore, each level may represent the entire book, a chapter, a sub-section, or even a page. Each audiobook contains a top-level parent that is defined in the *AB_Book* structure. This section represents the entire audiobook. Within this top-level section, an infinite number of sub-sections are defined.

If an *AB_Level* represents a group of audio files, it shall contain child sub-levels that reference each audio file's *TOC_TrackEntry* structure in the group. An *AB_Level* references an audio file using the *First TOC_TrackEntry Index* field in the *AB_Level Fixed Chunk Field*. Since this field only contains one *TOC_TrackEntry* index, it would be impossible for a single *AB_Level* structure to reference multiple audio files without defining child sub-levels. For example, if an audiobook were divided into 3 separate audio files, it would be impossible to have the top-level *AB_Level* structure (*Level Number* = 0) define no child

sub-levels. At the very least, this top-level structure shall have 3 sub-levels that reference each audio file's *TOC_TrackEntry* structure.

The *AB_Level* structure is defined as an *AB_ExtStructure*. There is no need to contain the *AB_Level* structure within a *Chunk* structure as defined in Table 2, because it is not directly contained within a MultiAudio structures *Extra Data* field. *AB_Section* format shall be as defined in Table 16.

#	RBP	Length	Name	Contents
1	0	12	Tag	tag
2	12	4	Offset to Extra Data	UInt32
2	16	L_Fc	<i>AB_Level</i> Fixed Chunk	AB_FixedChunk
3	16 + L_Fc	L_Vc	<i>AB_Level</i> Variable Chunk	AB_VariableChunk
4	16 + L_Fc + L_Vc	L_Ex	Extra Data	ExtraData

Table 16 AB_Section Format

8.1 Tag

The *Tag.Identifier* shall be set to the hex value 564C4241h (ABLV) to designate an *AB_Level* structure.

The *Tag.OrdinalNumber* shall be set to the value 0.

The *Tag.Length* shall be set to the length of the *AB_Level* structure in bytes, which includes the length of the Tag.

8.2 AB_Level Fixed Chunk

The *AB_Level Fixed Chunk* shall contain the format indicated in Table 17.

#	RBP	Length	Name	Contents
1	0	12	Tag	tag
2	12	2	Text Format	UInt16
3	14	2	Flags	UInt16
4	16	2	Number of Sub-Level (N_Sc)	UInt16
5	18	2	Level Number	UInt16
6	20	2	First <i>TOC_TrackEntry</i> Index	UInt16
7	22	2	Offset to Sub-Level Chunk	UInt16
8	24	2	Offset to Level Name	UInt16
9	26	2	Offset to Level Description	UInt16
10	28	4	Book Time Offset	UInt32
11	32	4	Time Start Offset	UInt32
12	36	4	Time Duration	UInt32
13	40	4	Offset to Parent <i>AB_Level</i>	UInt32
14	44	4	Offset to Previous <i>AB_Level</i>	UInt32
15	48	4	Offset to Next <i>AB_Level</i>	UInt32

Table 17 AB_Level Fixed Chunk Format

8.2.1 Tag

The *Tag.Identifier* sub-field shall be set to the hex value 9FFF0001h to designate that it is an *AB_FixedChunk*.

The *Tag.OrdinalNumber* sub-field shall be set to the value 0.

The *Tag.Length* shall be set to the length of the *AB_Level Fixed Chunk* structure in bytes. For CEA-2003-C, the length is 52 bytes.

8.2.2 Text Format

If this field is set to 0, then all text fields shall be interpreted as ASCII strings.

If this field is set to 1, then all text fields shall be interpreted as UNICODE strings.

8.2.3 Flags

Flags shall be set to the value 0 for CEA-2003-C.

8.2.4 Number of Sub-Levels

This field shall contain the number of sub-levels that this level represents. If there are no sub-levels, this field shall be set to 0.

8.2.5 Level Number

This field shall contain the level number of the level. The top level shall have this value set to 0, while each sub-level having value of one greater than the parent. The *AB_Level* with the level 0 represents the entire audiobook, which can be found in the *AB_Book structure*.

8.2.6 TOC_TrackEntry Index

Each *AB_Level* represents a piece of an audio file, or a group of audio files. Each audio file is defined by a MultiAudio *TOC_TrackEntry* structure. This field shall contain the index of the *TOC_TrackEntry* structure within the MultiAudio TOC file that is the first audio file that this *AB_Level* encapsulates. If the *AB_Level* is a piece of one file, the index shall be the index of the *TOC_TrackEntry* of which the level is a piece. If the *AB_Level* represents a group of audio files, this field shall contain the *TOC_TrackEntry* index of the first audio file in the group.

8.2.7 Offset to Sub-Level Chunk

The Offset to Sub-Level Chunk shall be defined as Offset in bytes from the start of the *AB_Level* structure to the sub-level chunk within the *AB_Level Variable Chunk* field. If there are no sub-levels, this field shall be set to 0.

8.2.8 Offset to Level Name

The Offset to Level Name shall be defined as Offset from the start of the *AB_Level* structure to the *Level Name*. This field shall not be 0, since the *Level Name* is a required. See Section 8.3.2.

8.2.9 Offset to Level Description

The Offset to Level Description shall be defined as Offset from the start of the *AB_Level* structure to the *Level Description*. If the level has no description, this field shall be set to the value 0. See Section 8.3.3.

8.2.10 Book Time Offset

Book Time Offset shall designate the time offset in milliseconds from the start of the audiobook to the start of this level.

8.2.11 Time Start Offset

Time Start Offset shall designate the time offset in milliseconds from the start of an audio file to the start of this level. If the level represents an audio file, all sub-levels shall start at a value that is the same or greater than this value. If the level represents a group of audio files, this field shall be set to 0.

8.2.12 Time Duration

Time Duration shall designate the time length of this level in milliseconds. If the level represents an audio file, the duration shall not exceed the *Time Start Offset* of the next level or the length of the entire audio file in milliseconds. If the level represents a group of audio files, the duration shall not exceed the total time length of the audiobook in milliseconds or the *Book Time Offset* of the next level.

8.2.13 Offset to Parent *AB_Level*

Offset from the start of this *AB_Level* structure to the parent of this level in bytes. Since the parent level is always previous to this level within a MultiAudio file, this constitutes a negative offset. The field is defined as an unsigned value (*Uint32*), so the parser shall jump backward the designated number of bytes, instead of forward.

If the level has no Parent Level, this field shall be set to the value 0. This shall only occur at the highest level (*Level Number* = 0), which is the first *AB_Level* defined in the *AB_Book structure*.

8.2.14 Offset to Previous *AB_Level*

Offset in bytes from the start of the *AB_Level* structure to the previous level at the same *Level Number*. If this is the first level at the current *Level Number*, this field shall be set to 0. Since the previous level is always defined before the *AB_Level* in the MultiAudio file, this constitutes a negative offset. The parser shall therefore jump backwards the designated number of bytes in the file, instead of forward.

8.2.15 Offset to Next *AB_Level*

Offset in bytes from the start of the *AB_Level* structure to the next level at the same *Level Number*. If this is the last level at the current *Level Number*, this field shall be set to the value 0.

8.3 *AB_Level* Variable Chunk

The *AB_Level Variable Chunk* shall contain the format indicated in Table 18.

#	RBP	Length	Name	Contents
1	0	12	Tag	tag
3	12	L_Sn	Level Name	byte
4	12 + L_Sn	L_Sd	Level Description	byte
6	12 + L_Sn + L_Sd	L_Sc	Sub-Level Chunk	LevelChunk

Table 18 *AB_Level* Variable Chunk Format

8.3.1 Tag

The *Tag.Identifier* sub-field should be set to the hex value 9FFF0002h to designate that this is an *AB_VariableChunk*.

The *Tag.OrdinalNumber* sub-field shall be set to the value 1

The *Tag.Length* shall be set to the length of the *AB_Level* Variable Chunk structure in bytes.

8.3.2 Level Name

This field shall contain the name of the level according to the character set defined by the *Text Format* field in the *AB_Level* Fixed Chunk structure. The level name should be a short string that accurately and briefly describes what the level represents. If the level has no name, this field shall be filled with a string termination character. It may be necessary to pad this field. See Section 5.4 for more information on padding.

8.3.3 Level Description

This field shall contain the description of the level according to the character set defined by the *Text Format* field in the *AB_Level* Fixed Chunk structure. The level description is a string that further defines what the level represents. If the level has no description, this field shall not be present. It may be necessary to pad this field. See Section 5.4 for more information on padding.

8.3.4 Sub-Level Chunk

Sub-Level Chunk shall be defined as a *Chunk* containing an array of *AB_Level* structures that represent the sub-levels of the *AB_Level* structure. If the *AB_Level* has no sub-levels, this field shall not be present. The *Chunk* structure shall have the following fields defined:

- The *LevelChunk.Tag.Identifier* sub-field shall be set to the hex value 9FFF0003h to designate a *LevelChunk*.
- The *LevelChunk.Tag.OrdinalNumber* sub-field shall be set to the value 0.
- LevelChunk.Tag.Length* shall be set to the length in bytes of all sub-levels plus the length in bytes of the Tag.

8.4 *AB_Level* Extra Data

AB_Level Extra Data shall be defined as a *Chunk* containing an array of *Chunk* based structures. This area is used to further extend the *AB_Level* structure.

9 MPV Audiobook Profiles

Section 9 describes the XML-based version of audiobook requirements, and is based on an adaptation of the OSTA MPV Music specification.

The MPV Core Specification (MPV-Core) defines the essential framework of the MPV technology. The MPV Music Profile 1.0 (MPV-Music) extends MVP-Core to specify how to create, manage and play music playlists.

An MPV framework mechanism, the Profile, defines significant new functionality. MPV-Core Specification establishes requirements for MPV Profile use. MPV-Core requires that an MPV file declare which profiles

it implements, and declare the namespaces of the profiles. This allows a processing application to quickly determine whether a given MPV file meets its expectations for processing.

The Audiobook profile makes use of the MPV Music Profile 1.0 because most of the MPV Music Profile is applicable to Audiobooks, and leveraging the MPV Music profile enhances playback compatibility for audiobooks. MPV-Music includes schema and practices.

The Audiobook Profile (MPV-Audiobook) is an extension of MPVMusic and is fully compatible with it. Files implementing MPV-Audiobook profile should be usable in MPV-Music-aware applications and devices. This means, for example, that an MPV-Music playback application or device can read and playback CEA Audiobooks; however, a playback device that is only MPV-Music aware ignores the MPV-Audiobook profile, and the playback experience will be less full-featured than in an MPV-Audiobook-aware player.

9.1 XML Schema Namespace Identifier

CEA-2003-C defines a schema namespace identifier to be used to identify the XML Schema for audiobooks. See Table 19

Namespace Identifier	http://www.ce.org/CEA-2003-DTD/1.0/
Conventional Namespace Prefix	ab:

Table 19 Schema Namespace Identifier

This identifier shall be used in the namespace declaration value when defining namespace prefixes or declaring the namespace of an element.

9.2 Use of MPV-Music Profile Identifier

To be compatible with MPV-Music players, an audiobook MPV file shall also include the MPV Music Profile declaration in the Profiles section of the MPV manifest in Table 20. An example is included in Figure 1.

MPV Music Profile Identifier	http://ns.osta.org/mpv/music/1.0/
------------------------------	-----------------------------------

Table 20 MPV-Music Profile Identifier

```
<?xml version="1.0" encoding="UTF-8"?>
<file:Manifest
  xmlns:file="http://ns.osta.org/manifest/1.0/"
  xmlns:mpv="http://ns.osta.org/mpv/1.0/"
  xmlns:mpvp="http://ns.osta.org/mpv/presentation/1.0/"
  xmlns:mpvm="http://ns.osta.org/mpv/music/1.0/"
  xmlns:ab="http://ns.ce.org/audiobook/2003b/"
  xmlns:nmf="http://ns.osta.org/nmf/1.0/" >
  <nmf:Metadata>
    <ManifestProperties xmlns="http://ns.osta.org/manifest/1.0/">
      <ProfileBag>
        <Profile>http://ns.osta.org/mpv/basic/1.0/</Profile>
        <Profile>http://ns.osta.org/mpv/music/1.0/</Profile>
        <Profile>http://ns.ce.org/audiobook/2003b/</Profile>
      </ProfileBag>
    </ManifestProperties>
  </nmf:Metadata>

  ...

</file:Manifest>
```

Figure 1 Example of MPV Music Profile Identifier Usage

9.3 Audiobook File

“Audiobook file” means an MPV file that complies with the MPVMusic that also declares the MPV-Audiobook profile. An audiobook file shall also contain a single <mpvm:Album> element that defines the