

INTERNATIONAL STANDARD

NORME INTERNATIONALE

Digital audiobook file format and player requirements

Exigences relatives au format de fichier de livre audio et aux lecteurs



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2011 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester.

If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de la CEI ou du Comité national de la CEI du pays du demandeur.

Si vous avez des questions sur le copyright de la CEI ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de la CEI de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub

The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub

Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv

If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch

Tel.: +41 22 919 02 11

Fax: +41 22 919 03 00

A propos de la CEI

La Commission Electrotechnique Internationale (CEI) est la première organisation mondiale qui élabore et publie des normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications CEI

Le contenu technique des publications de la CEI est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

- Catalogue des publications de la CEI: www.iec.ch/searchpub/cur_fut-f.htm

Le Catalogue en-ligne de la CEI vous permet d'effectuer des recherches en utilisant différents critères (numéro de référence, texte, comité d'études,...). Il donne aussi des informations sur les projets et les publications retirées ou remplacées.

- Just Published CEI: www.iec.ch/online_news/justpub

Restez informé sur les nouvelles publications de la CEI. Just Published détaille deux fois par mois les nouvelles publications parues. Disponible en-ligne et aussi par email.

- Electropedia: www.electropedia.org

Le premier dictionnaire en ligne au monde de termes électroniques et électriques. Il contient plus de 20 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans les langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International en ligne.

- Service Clients: www.iec.ch/webstore/custserv/custserv_entry-f.htm

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions, visitez le FAQ du Service clients ou contactez-nous:

Email: csc@iec.ch

Tél.: +41 22 919 02 11

Fax: +41 22 919 03 00



IEC 62571

Edition 1.0 2011-03

INTERNATIONAL STANDARD

NORME INTERNATIONALE

Digital audiobook file format and player requirements

Exigences relatives au format de fichier de livre audio et aux lecteurs

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XE

ICS 33.160.60; 35.240.30

ISBN 978-2-88912-603-3

CONTENTS

FOREWORD.....	7
1 Scope.....	9
2 Normative references	9
3 General	9
4 MultiAudio audiobook extensions.....	11
4.1 Table of contents.....	11
4.2 Description	14
4.2.1 General	14
4.2.2 Tag.Identifier	14
4.2.3 Tag.OrdinalNumber	14
4.2.4 Tag.Reserved	14
4.2.5 Tag.Length	14
4.2.6 Data	14
5 Audiobook extendable structure (AB_ExtStruct)	15
5.1 General	15
5.2 Tag	15
5.3 Offset to extra data.....	15
5.4 Fixed length chunk (AB_FixedChunk).....	15
5.4.1 General	15
5.4.2 Example	16
5.5 Variable length chunk (AB_VariableChunk)	16
5.5.1 General	16
5.5.2 Example	17
5.6 Extra data	18
6 Audiobook header (AB_Header)	18
6.1 General	18
6.2 Tag	18
6.3 AB_Header fixed chunk	18
6.3.1 General	18
6.3.2 Tag.....	19
6.3.3 Audiobook specification major version	19
6.3.4 Audiobook specification minor version	19
6.3.5 Number of audiobooks (N_Bk).....	19
6.3.6 Offset to playlist indexes	19
6.4 AB_Header variable chunk	19
6.4.1 General	19
6.4.2 Tag.....	19
6.4.3 Playlist indexes	20
6.5 AB_Header extra data	20
7 Audiobook book (AB_Book).....	20
7.1 General	20
7.2 Tag	21
7.3 AB_Book fixed chunk	21
7.3.1 General	21
7.3.2 Tag.....	22

7.3.3	Text format	22
7.3.4	Flags	22
7.3.5	TOC_Playlist index	22
7.3.6	Offset to book title	22
7.3.7	Offset to author	22
7.3.8	Offset to narrator	22
7.3.9	Offset to copyright statement	22
7.3.10	Offset to audiobook UUID	22
7.3.11	Offset to book ISBN identifier	23
7.3.12	Offset to book ISSN identifier	23
7.3.13	Offset to book DOI identifier	23
7.3.14	Offset to book index number	23
7.3.15	Offset to book shelf number	23
7.3.16	Offset to book part number	23
7.3.17	Offset to other identifier	23
7.3.18	Offset to first audiobook level	23
7.3.19	Offset to playlist content location identifier	23
7.4	AB_Book variable chunk	24
7.4.1	General	24
7.4.2	Tag	24
7.4.3	Book title	24
7.4.4	Author	25
7.4.5	Narrator	25
7.4.6	Copyright Statement	25
7.4.7	Book UUID	25
7.4.8	Book ISBN Identifier	25
7.4.9	Book ISSN Identifier	25
7.4.10	Book DOI Identifier	26
7.4.11	Book index number	26
7.4.12	Book shelf number	26
7.4.13	Book part number	26
7.4.14	Other identifier	26
7.4.15	First audiobook level	26
7.4.16	Playlist content location identifier	27
7.5	AB_Book extra data	27
8	Audiobook level (AB_Level)	27
8.1	General	27
8.2	Tag	28
8.3	AB_Level fixed chunk	28
8.3.1	General	28
8.3.2	Tag	28
8.3.3	Text format	29
8.3.4	Flags	29
8.3.5	Number of sub-levels	29
8.3.6	Level number	29
8.3.7	TOC_TrackEntry index	29
8.3.8	Offset to sub-level chunk	29
8.3.9	Offset to level name	29
8.3.10	Offset to level description	29

8.3.11	Book time offset	29
8.3.12	Time start offset	29
8.3.13	Time duration	30
8.3.14	Offset to parent AB_Level.....	30
8.3.15	Offset to previous AB_Level	30
8.3.16	Offset to next AB_Level	30
8.4	AB_Level variable chunk	30
8.4.1	General	30
8.4.2	Tag.....	30
8.4.3	Level name.....	31
8.4.4	Level description	31
8.4.5	Sub-level chunk.....	31
8.5	AB_Level extra data	31
9	MPV audiobook profiles.....	31
9.1	General	31
9.2	XML schema namespace identifier	32
9.3	Use of MPV-Music profile identifier.....	32
9.4	Audiobook file	33
9.5	Audiobook profile metadata	33
9.6	MPV audiobook profile metadata	35
9.7	Dublin core metadata usage	39
9.8	Audiobook profile using MPV-Music playlists.....	41
9.8.1	General	41
9.8.2	Number of playlists per MPV file.....	42
9.8.3	Metadata usage.....	42
9.8.4	Background usage.....	43
9.8.5	Foreground usage	44
9.8.6	Related and rendition assets	44
9.9	Groups of assets in playlists.....	44
9.10	MPV-Audiobook schema details	47
9.10.1	XML schema definition	47
9.10.2	Audiobook metadata usage definitions.....	52
9.11	Audiobook level class identifiers.....	52
9.12	Audiobook BOOKMARK.AUB	54
9.13	MPV-Audiobook-Level metadata using <mpvp:Album>	61
9.14	<mpvm:AudioWithStills> Music and Stills Asset.....	61
9.15	MPV-Audiobook extensions to MPV-Core	64
9.15.1	Audiobook manifest file types and extensions.....	64
9.15.2	Audiobook manifest MIME media type	64
9.16	Media types and file formats.....	65
9.17	MPV-Audiobook and CEA-2003-C	65
9.18	CEA-2003-C binary to XML formats – Tag comparison	65
10	Conforming player and digital audiobook file format structure requirements.....	66
10.1	General	66
10.2	Player Requirements	66
10.3	Digital audiobook file format structure requirements	67
11	Player and digital audiobook file format structure functionality recommendations	67
11.1	General	67

11.2 Player functionality recommendations	67
11.3 Digital audiobook file format structure functionality recommendations	68
12 Digital audiobook file format structures and data constraints	68
13 Additional production options.....	68
13.1 ID3 tagging within MP3 files	68
13.2 M3U playlists.....	69
13.3 DAISY production options.....	69
14 Certification	69
Annex A (informative) Audiobook structure examples.....	70
Annex B (normative) Encoding types and identifiers	82
Annex C (informative) Use case and implementation examples: Spoken word audio	83
Bibliography.....	104
Figure A.1 – AB_Level structure organization in AB_Book variable length chunk area	74
Figure A.2 – View of an audiobook AB_Level hierarchy	76
Figure C.1 – Example 1 INDEX.AUB	88
Figure C.2 – Example 1 TOC.MUM	89
Figure C.3 – Example 2 INDEX.AUB	94
Figure C.4 – Example 3 INDEX.AUB	99
Figure C.5 – Example 4 INDEX.AUB	103
Table 1 – Table of contents data structure	12
Table 2 – Audiobook MultiAudio chunk format.....	14
Table 3 – AB_ExtStruct format.....	15
Table 4 – Example—AB_FixedChunk structure	16
Table 5 – Example—AB_FixedChunk structure revised	16
Table 6 – Example—AB_FixedChunk fields	17
Table 7 – Example—AB_VariableChunk data.....	17
Table 8 – AB_Header format.....	18
Table 9 – AB_Header fixed chunk format	18
Table 10 – AB_Header variable chunk format	19
Table 11 – AB_Book format	21
Table 12 – AB_Book fixed chunk format.....	21
Table 13 – AB_Book variable chunk format.....	24
Table 14 – AB_Level structure required field settings	26
Table 15 – Playlist content location identifier field settings.....	27
Table 16 – AB_Section format	28
Table 17 – AB_Level fixed chunk format	28
Table 18 – AB_Level variable chunk format	30
Table 19 – Schema namespace identifier.....	32
Table 20 – MPV-Music profile identifier.....	32
Table 21 – MPV audiobook profile properties.....	34
Table 22 – MPV audiobook profile metadata	35
Table 23 – Dublin Core metadata (DC-NMF).....	40

Table 24 – Metadata properties usage	42
Table 25 – CEA-2003-C binary and XML formats – Tag comparison	65
Table 26 – Examples of ID3 tags	69
Table A.1 – Sample table of contents	70
Table A.2 – Table of contents file structure	71
Table A.3 – File 1 TOC_TrackEntry structure	71
Table A.4 – File 2 TOC_TrackEntry structure	71
Table A.5 – File 3 TOC_TrackEntry structure	71
Table A.6 – File 4 TOC_TrackEntry structure	72
Table A.7 – Table of contents with file time offsets	72
Table A.8 – Example TOC_Playlist structure	72
Table A.9 – Example AB_Book structure	73
Table A.10 – AB_Level structure at Level 0 (audiobook)	77
Table A.11 – AB_Level at Level 1 (chapter)	77
Table A.12 – AB_Level at Level 1 (chapter)	78
Table A.13 – AB_Level at Level 1 (chapter)	78
Table A.14 – AB_Level at Level 2 (section)	79
Table A.15 – AB_Level at Level 2 (section)	79
Table A.16 – AB_Level at Level 2 (section)	80
Table A.17 – AB_Level at Level 2 (section)	80
Table A.18 – AB_Level at Level 3 (subsection)	81
Table A.19 – AB_Level at Level 3 (subsection)	81
Table B.1 – Hex representation and interpretation	82

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**DIGITAL AUDIOBOOK FILE FORMAT
AND PLAYER REQUIREMENTS**

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62571 has been prepared by technical area 10: Multimedia e-publishing and e-book, of IEC technical committee 100: Audio, video and multimedia systems and equipment.

This bilingual version (2011-07) replaces the English version.

The text of this standard is based on the following documents:

CDV	Report on voting
100/1543/CDV	100/1629/RVC

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

DIGITAL AUDIOBOOK FILE FORMAT AND PLAYER REQUIREMENTS

1 Scope

This International Standard 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 this International Standard, the term audiobook is defined as any audio file or collection of audio files of primarily spoken word content that are played in a linear or specified order. Therefore, spoken word audio with occasional music, a narration of newspaper articles, or other similar spoken word audio is assimilated to audiobooks in this standard.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 11578:1996, *Information technology – Open Systems Interconnection – Remote Procedure Call (RPC)*

ISO 639-1, *Codes for the representation of names of languages – Part 1: Alpha-2 code*

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes*

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

3 General

This International Standard defines the audiobook file format structure for digital audiobook media. Goals in creating this International Standard include compatibility with music industry and multimedia standards as well as effectively presenting and navigating an audiobook. This International Standard is a compilation standard that straddles early binary architectures represented by earlier versions of CEA-2003¹, and newer XML architectures represented by the Optical Storage Technology Association's (OSTA) MusicPhotoVideo™² MPV). A conforming playback system and conforming audiobook content provides a wonderful and highly functional reading and listening experience. The goal of this International Standard 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 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

¹ See Bibliography.

² MusicPhotoVideo is the trade name of a program supplied by the Optical Storage Technology Association. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the program named.

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. This International Standard "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 this International Standard, "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. This International Standard 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 Clause 11. In addition, conforming players and conforming audio content are encouraged, but not required, to conform to the recommendations of Clause 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 Clauses 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 standard.

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 1 Books, chapters and pages are used as level name examples; however, there is no naming requirement for levels in order to be compliant with this International Standard.

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 2 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, that is 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, Clauses 4 to 8 detail the MultiAudio audiobook extensions, headers, playlists and levels in the MultiAudio format. For a player that is capable of reading XML, Clause 10 discusses in detail the MPV schema for audiobooks in the MPV format. For players that read both binary files and XML schema, the XML should be used.

4 MultiAudio audiobook extensions

4.1 Table of contents

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

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

Table 1 – Table of contents data structure

Clause / Subclause	Description
4.2	Description
Table 2	Audiobook MultiAudio Chunk
4.2.2	Tag.Identifier
4.2.3	Tag.OrdinalNumber
4.2.4	Tag.Reserved
4.2.5	Tag.Length
4.2.6	Tag.Data
5	Audiobook Extendable Structure (AB_ExtStruct)
5.2	Tag
5.3	Offset to Extra Data
5.4	Fixed Length Chunk (AB_FixedChunk)
5.4.2	Example
5.5	Variable Length Chunk (AB_VariableChunk)
5.5.2	Example
5.6	Extra Data
6	Audiobook Header (AB_Header)
6.2	Tag
6.3	AB_Header Fixed Chunk
6.3.2	Tag
6.3.3	Audiobook Spec Major Version
6.3.4	Audiobook Spec Minor Version
6.3.5	Number of Audiobooks
6.3.6	Offset to Playlist Indexes
6.4	AB_Header Variable Chunk
6.4.2	Tag
6.4.3	Playlist Indexes
6.5	Extra Data
7	Audiobook (AB_Book)
7.2	Tag
7.3	AB_Book Fixed Chunk
7.3.2	Tag
7.3.3	Text Format
7.3.4	Flags
7.3.5	TOC_Playlist Index
7.3.6	Offset to Book Title
7.3.7	Offset to Author
7.3.8	Offset to Narrator
7.3.9	Offset to Copyright Statement
7.3.10	Offset to Audiobook UUIDs
7.3.11	Offset to Book ISBN Identifier

Table 1 (Continued)

Clause Subclause	Description
7.3.12	Offset to Book ISSN Identifier
7.3.13	Offset to Book DOI Identifier
7.3.14	Offset to Book Index Number
7.3.15	Offset to Book Shelf Number
7.3.16	Offset to Book Part Number
7.3.17	Offset to Other Identifier
7.3.18	Offset to First Audiobook Level
7.3.19	Offset to Playlist Content Location Identifier
7.4	AB_Book Variable Chunk
7.4.2	Tag
7.4.3	Book Title
7.4.4	Author
7.4.5	Narrator
7.4.6	Copyright Statement
7.4.7	Book UUID
7.4.8	Book ISBN Identifier
7.4.9	Book ISSN Identifier
7.4.10	Book DOI Identifier
7.4.11	Book Index Number
7.4.12	Book Shelf Number
7.4.13	Book Part Number
7.4.14	Other Identifier
7.4.15	First Audiobook Level
7.4.16	Playlist Content Location Identifier
7.5	Extra Data
8	Audiobook Level (AB_Level)
8.2	Tag
8.3	AB_Level Fixed Chunk
8.3.2	Tag
8.3.3	Text Format
8.3.4	Flags
8.3.5	Number of Sub-Levels
8.3.6	Level Number
8.3.7	First TOC_TrackEntry Index
8.3.8	Offset to Sub-Level Chunk
8.3.9	Offset to Level Name
8.3.10	Offset to Level Description
8.3.11	Book Time Offset
8.3.12	Time Start Offset
8.3.13	Time Duration
8.3.14	Offset to Parent AB_Level
8.3.15	Offset to Previous AB_Level
8.3.16	Offset to Next AB_Level
8.4	AB_Level Variable Chunk
8.4.2	Tag
8.4.3	Level Name
8.4.4	Level Description
8.4.5	Sub-Level Chunk
8.5	Extra Data

4.2 Description

4.2.1 General

All structures defined in CEA-2003-B were extensions of OSTA MultiAudio. These structures are retained in this International Standard. 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 Clause 5.

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

Table 2 – Audiobook MultiAudio chunk format

#	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

4.2.2 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.

4.2.3 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.

4.2.4 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.

4.2.5 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.

4.2.6 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)

5.1 General

The audiobook extendable structure defines a binary structure that allows easy extensibility. 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.

Table 3 – AB_ExtStruct format

#	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	ExtraData

5.2 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.

5.3 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.

5.4 Fixed length chunk (AB_FixedChunk)

5.4.1 General

The fixed length chunk is defined as a chunk structure, and provides for the extensibility 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 extensibility 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.

5.4.2 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.

Table 4 – Example—AB_FixedChunk structure

#	RBP	Length	Name	Contents
1	0	12	AB_FixedChunk Tag	Tag
2	12	4	Field1	Uint32
3	16	2	Field2	Uint16
4	18	2	Field3	Uint16

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.

Table 5 – Example—AB_FixedChunk structure revised

#	RBP	Length	Name	Contents
1	0	12	AB_FixedChunk Tag	Tag
2	12	4	Field1	Uint32
3	16	2	Field2	Uint16
4	18	2	Field3	Uint16
5	20	4	Field4	Uint32

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.

5.5 Variable length chunk (AB_VariableChunk)

5.5.1 General

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 is the 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 bytes.

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.

5.5.2 Example

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

Table 6 – Example—AB_FixedChunk fields

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

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

Table 7 – Example—AB_VariableChunk data

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

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.

5.6 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)

6.1 General

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 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 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.

Table 8 – AB_Header format

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

6.2 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.

6.3 AB_Header fixed chunk

6.3.1 General

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

Table 9 – AB_Header fixed chunk format

#	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

6.3.2 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. This International Standard specifies this length as 20 bytes.

6.3.3 Audiobook specification 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 specification.

For this International Standard, this field shall be set to the hex value 0001h, which designates that this is version 1.X and has a structure compliant with this International Standard.

NOTE Since this International Standard is backwards compatible with CEA-2003, CEA-2003-A, CEA-2003-B and ANSI/CEA-2003-C, version 1.0 is used for all versions of the standard.

6.3.4 Audiobook specification 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 specification.

For this International Standard, this field shall be set to the hex value 0001h, which designates that this is version X.1.

6.3.5 Number of audiobooks (N_Bk)

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

6.3.6 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 6.3.

6.4 AB_Header variable chunk

6.4.1 General

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

Table 10 – AB_Header variable chunk format

#	RBP	Length	Name	Contents
1	0	12	Tag	Tag
2	12	N_Bk * 2	Playlist Indexes	Uint16

6.4.2 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.

6.4.3 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 Clause 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 5.5 for more information about padding.

6.5 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)

7.1 General

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 this International Standard's 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 as shown in Table 11.

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 4.2 within the Extra Data field of a MultiAudio TOC_Playlist). The ordinal number of this Chunk is not defined.

Table 11 – AB_Book format

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

7.2 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.

7.3 AB_Book fixed chunk

7.3.1 General

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

Table 12 – AB_Book fixed chunk format

#	RBP	Length	Name	Contents
1	0	12	Tag	tag
2	12	2	Text Format	UInt16
3	14	2	Flags	UInt16
4	16	2	TOC_Playlist 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

7.3.2 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 this International Standard, the length is 48 bytes.

7.3.3 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.

7.3.4 Flags

This field shall be set to the value 0 for this International Standard.

7.3.5 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.

7.3.6 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 7.4.3.

7.3.7 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 7.4.4.

7.3.8 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 7.4.5.

7.3.9 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 7.4.6.

7.3.10 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 7.4.7.

7.3.11 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 7.4.8.

7.3.12 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 7.4.10.

7.3.13 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 7.4.10.

7.3.14 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 7.4.11.

7.3.15 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 7.4.12.

7.3.16 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 7.4.13.

7.3.17 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 7.4.14.

7.3.18 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 7.4.15.

7.3.19 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 7.4.16.

7.4 AB_Book variable chunk

7.4.1 General

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

Table 13 – AB_Book variable chunk format

#	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

7.4.2 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.

7.4.3 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 5.5 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.

7.4.4 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 5.5 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.

7.4.5 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 5.5 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.

7.4.6 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 5.5 for more information on padding.

7.4.7 Book UUID

A UUID format was defined in the Open Software Foundation's Distributed Computer Environment RPC standard also available as ISO/IEC 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.

7.4.8 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 5.5 for more information on padding.

7.4.9 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 5.5 for more information on padding.

7.4.10 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 5.5 for more information on padding.

7.4.11 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 5.5 for more information on padding.

7.4.12 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 5.5 for more information on padding.

7.4.13 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 5.5 for more information on padding.

7.4.14 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 5.5 for more information on padding.

7.4.15 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.

Table 14 – AB_Level structure required field settings

Subclause #	AB_Level field	Value
8.3.2	Tag.OrdinalNumber	0
8.3.6	Level Number	0
8.3.14	Offset to Parent AB_Level	0
8.3.11	Book Time Offset	0
8.3.12	Time Start Offset	0
8.3.13	Time Duration	Length of audiobook in milliseconds
8.3.15	Offset to Previous AB_Level	0
8.3.16	Offset to Next AB_Level	0

See Clause 8 for more information on the AB_Level structure.

7.4.16 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.

Table 15 – Playlist content location identifier field settings

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

7.5 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)

8.1 General

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 clause 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.

Table 16 – AB_Section format

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

8.2 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.

8.3 AB_Level fixed chunk

8.3.1 General

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

Table 17 – AB_Level fixed chunk format

#	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-Levels (N_Sc)	UInt16
5	18	2	Level Number	UInt16
6	20	2	First TOC_TrackEntry 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 AB_Level	UInt32
14	44	4	Offset to Previous AB_Level	UInt32
15	48	4	Offset to Next AB_Level	UInt32

8.3.2 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 this International Standard, the length is 52 bytes.

8.3.3 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.

8.3.4 Flags

Flags shall be set to the value 0 for this International Standard.

8.3.5 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.

8.3.6 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.

8.3.7 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.

8.3.8 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.

8.3.9 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 8.4.3.

8.3.10 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 8.4.4.

8.3.11 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.

8.3.12 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.

8.3.13 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.

8.3.14 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.

8.3.15 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.

8.3.16 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.

8.4 AB_Level variable chunk

8.4.1 General

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

Table 18 – AB_Level variable chunk format

#	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

8.4.2 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.

8.4.3 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 5.5 for more information on padding.

8.4.4 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 5.5 for more information on padding.

8.4.5 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.

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

8.5 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

9.1 General

This clause 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 MPV-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 MPV-Music 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.

9.2 XML schema namespace identifier

This International Standard defines a schema namespace identifier to be used to identify the XML schema for audiobooks, see Table 19.

Table 19 – Schema namespace identifier

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

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

9.3 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 given in Table 20.

Table 20 – MPV-Music profile identifier

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

An example of an MPV music profile identifier usage is given below.

```
<?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>
```

9.4 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 audiobook and contains a set of assets described within the `<mpv:AssetList>` element which are referenced directly or indirectly by the contents of the `<mpvm:Album>`.

9.5 Audiobook profile metadata

MPV-Audiobook extends and builds upon metadata defined in MPVMusic. MPV-Audiobook adds metadata specifically about audiobooks.

MPV-Audiobook metadata and MPV-Music metadata may be used as metadata assigned to individual MPV assets. In MPV-Audiobook, each MPV audio asset corresponds to a book chapter.

The metadata may also be used with the MPV presentation playlist (“element name `<mpvp:album>`”) to describe the metadata for the entire audiobook one time, instead of duplicating it for each audiobook chapter asset. The MPV presentation playlist is used to wrap a presentation layer on top of the individual audio assets. The MPV presentation album defines the audiobook as a whole.

MPV-Audiobook shall include the new properties defined in Table 21.

These properties are specific to audiobooks. These properties may be used within a `<nmf:Metadata>` element that is present on any MPV asset (such as `mpv:Audio`) or on the MPV presentation element `<mpvp:Album>`

Table 21 – MPV audiobook profile properties

Metadata	Audiobook Profile	Discussion
Book Title	nmf:Metadata ab:Properties ab:BookTitle	Primary metadata for the title of the book being read. When used on an asset, overrides any metadata value specified on mpvp:Album.
Book Author	nmf:Metadata ab:Properties ab:BookAuthor	Primary metadata for the author of the book being read. When used on an asset, overrides any metadata value specified on mpvp:Album.
Book Co-Author	nmf:Metadata ab:Properties BookCoauthor	Primary metadata for the co-author of the book being read. When used on an asset, overrides any metadata value specified on mpvp:Album.
Book Subtitle	nmf:Metadata ab:Properties BookSubtitle	Primary metadata for the Subtitle of the book being read. When used on an asset, overrides any metadata value specified on mpvp:Album.
Book Narrator	nmf:Metadata ab:Properties ab:BookNarrator	Primary metadata for the narrator of the book being read. When used on an asset, overrides any metadata value specified on mpvp:Album.
Book Genre	nmf:Metadata ab:Properties BookGenre	Primary metadata for the genre of the book being read. When used on an asset, overrides any metadata value specified on mpvp:Album.
Book UUID	nmf:Metadata ab:Properties ab:BookUUID	Primary metadata for book UUID.
Book ISBN Identifier	nmf:Metadata ab:Properties ab:BookISBN	Primary metadata for book ISBN.
Book ISSN Identifier	nmf:Metadata ab:Properties ab:BookISSN	Primary metadata for book ISSN.
Book DOI Identifier	nmf:Metadata ab:Properties ab:BookDOI	Primary metadata for book DOI.
Book Index Number	nmf:Metadata ab:Properties ab:BookIndexNumber	Primary metadata for book Book Index Number.
Book Shelf Number	nmf:Metadata ab:Properties ab:BookShelfNumber	Primary metadata for book Book Shelf Number.
Book Part Number	nmf:Metadata ab:Properties ab:BookPartNumber	Primary metadata for book Book Part Number.
Other Identifier	nmf:Metadata ab:Properties ab:OtherIdentifier	Primary metadata for any other book identifier.
Content Location Identifier	nmf:Metadata ab:Properties ab:ContentLocationIdentifier	Primary metadata for playlist content location identifier.

9.6 MPV audiobook profile metadata

As described in MPV-Music, a variety of metadata is available to describe audiobooks. Table 22 reviews those properties and gives an interpretation for audiobooks.

Table 22 – MPV audiobook profile metadata

Metadata	Metadata elements in MPV	Discussion regarding audiobooks
Audiobook asset Chapter	Sub-elements of the mpv:Audio asset	
Pathname	mpv:LastURL	One or more pathnames that should resolve to the chapter file. Each pathname is specific to a file system, so there can be different names for each file system on a CD.
Format	nmf:Metadata dc:Properties dc:format	Use MPV-provided list of well-known MIME types or vendor-specific extended types.
Title	nmf:Metadata dc:Properties dc:title	Primary metadata for the title of this audio asset. Such as: if a chapter, the chapter title; if a page, the page number.
Album Title	nmf:Metadata mpvm:MusicProperties mpvm:AlbumTitle	Not used as primary metadata for Audiobooks. SHOULD be populated with the value of ab:BookTitle for enhanced compatibility with MPV Music players.
Book Title	nmf:Metadata ab:Properties ab:BookTitle	Primary metadata for the title of the book being read. When used on an asset, overrides any metadata value specified on mpvp:Album.
Creator	nmf:Metadata dc:Properties dc:creator	Not used as primary metadata for Audiobooks. SHOULD be populated with the value of ab:BookAuthor for enhanced compatibility with MPV Music players.
Principal artist	nmf:Metadata mpvm:MusicProperties mpvm:PrincipalArtist	Not used as primary metadata for Audiobooks. SHOULD be populated with the value of ab:BookAuthor for enhanced compatibility with MPV Music players.
Book Author	nmf:Metadata ab:Properties ab:BookAuthor	Primary metadata for the author of the book being read. When used on an asset, overrides any metadata value specified on mpvp:Album.

Table 22 (continued)

Metadata	Metadata elements in MPV	Discussion regarding audiobooks
Contributor	nmf:Metadata dc:Properties dc:contributor	Not used as primary metadata for Audiobooks. SHOULD be populated with the value of ab:BookNarrator for enhanced compatibility with MPV Music players.
Book Narrator	nmf:Metadata ab:Properties ab:Narrator	Primary metadata for the narrator of the book being read. When used on an asset, overrides any metadata value specified on mpvp:Album.
Music by	nmf:Metadata mpvm:MusicProperties mpvm:MusicBy	Composer name(s) in free form plain text.
Lyrics by	nmf:Metadata mpvm:MusicProperties mpvm:LyricsBy	Not used.
Arranged by	nmf:Metadata mpvm:MusicProperties mpvm:ArrangedBy	Not used.
Produced By	nmf:Metadata mpvm:MusicProperties ProducedBy	Producer name(s) in free form plain text.
Genre	nmf:Metadata mpvm:MusicProperties mpvm:Genre	Predefined set of genres can be extended simply by using a new name.
Date Recorded	nmf:Metadata dcterms:Properties dcterms:created	recorded date-time, for example 2002-10-03T21:07:00Z
Identifier	nmf:Metadata dc:Properties dc:identifier	Not used as primary metadata for audiobooks. SHOULD be populated with the value of ab:BookISBN for enhanced compatibility with MPV players of all kinds.
Book UUID	nmf:Metadata ab:Properties ab:BookUUID	Primary metadata for book UUID.
Book ISBN Identifier	nmf:Metadata ab:Properties ab:BookISBN	Primary metadata for book ISBN.
Book ISSN Identifier	nmf:Metadata ab:Properties ab:BookISSN	Primary metadata for book ISSN.
Book DOI Identifier	nmf:Metadata ab:Properties ab:BookDOI	Primary metadata for book DOI.
Book Index Number	nmf:Metadata ab:Properties ab:BookIndexNumber	Primary metadata for book Book Index Number.
Book Shelf Number	nmf:Metadata ab:Properties ab:BookShelfNumber	Primary metadata for book Book Shelf Number.
Book Part Number	nmf:Metadata ab:Properties ab:BookPartNumber	Primary metadata for book Book Part Number.
Other Identifier	nmf:Metadata ab:Properties ab:OtherIdentifier	Primary metadata for any other book identifier.
Content Location Identifier	nmf:Metadata ab:Properties ab:LocationIdentifier	Primary metadata for playlist content location identifier.

Table 22 (continued)

Metadata	Metadata elements in MPV	Discussion regarding audiobooks
Description	nmf:Metadata dc:Properties dc:description	Commentary, message, description.
Keywords	nmf:Metadata dc:Properties dc:subject	Semicolon separated list of plain text keywords or phrases for search. Semi-colon can be used by escaping with backslash, for example \;
Copyright Statement	nmf:Metadata dc:Properties dc:rights	Free-form plain text copyright statement of rights.
Language	nmf:Metadata dc:Properties dc:language	"<language>_<territory>" where <language> is from [ISO 639-1] and <territory> is [ISO 3166-1] Example: "en_US"
Rendition using another encoder	mpv:Rendition mpv:renditionType="alt" mpv:AudioRef	Refers to an alternate encoding of the asset.
Rendition with another bit rate	mpv:Rendition mpv:renditionType="subsampling" mpv:AudioRef	Refers to another encoding of the asset at a different bit rate. Use mpvm:EncodedBitrate or interrogate the file for more information.
Encoded Bit rate	nmf:Metadata mpvm:MusicProperties mpvm:EncodedBitrate	Maximum bit rate in bps of the encoded asset. For variable bit rate-encoded assets, this is the maximum value; for constant bit rate encoded assets, it is the average value.
Asset artwork	mpv:Related mpv:relationship="urn:osta-org:mpv:music:artwork" mpv:StillRef	Not used.
Music video	mpv:Related mpv:relationship="urn:osta-org:mpv:music:video" mpv:VideoRef	Not used.
Online Info	nmf:Metadata mpvm:MusicProperties mpvm:MoreInfoURL	URL to follow to get more info about the audiobook.
Playing Time	nmf:Metadata mpvm:MusicProperties mpvm:PlayingTime	In seconds, duration. Decimal values are allowed, such as 134,58 s. Means 134 and 58/100 s.
Track Number	nmf:Metadata mpvm:MusicProperties mpvm:TrackNumber	Sequence order of the chapter in the original book. Starts with 1, not 0.
Number Tracks	nmf:Metadata mpvm:MusicProperties mpvm:NumTracks	Number of chapters in the original compilation. Starts with 1, not 0.
Set Number in Collection	nmf:Metadata mpvm:MusicProperties mpvm:SetNumber	Not used.
Number Sets in Collection	nmf:Metadata mpvm:MusicProperties mpvm:NumSets	Not used.

Table 22 (continued)

Metadata	Metadata elements in MPV	Discussion regarding audiobooks
PlayCount	nmf:Metadata mpvm:MusicProperties mpvm:PlayCount	Not used.
Lyrics	nmf:Metadata mpvm:MusicProperties mpvm:Lyrics	Not used.
Mood	nmf:Metadata mpvm:MusicProperties mpvm:Mood	Not used.
Tempo	nmf:Metadata mpvm:MusicProperties mpvm:Tempo	Not used.
Situation	nmf:Metadata mpvm:MusicProperties mpvm:Situation	Not used.
Medium	nmf:Metadata dcterms:Properties dcterms:medium	Not used.
Original file and filename	mpv:Related mpv:relationship="copyOf" mpv:AudioRef	Not used.
Extra Data	nmf:Metadata and mpv:Metadata	Any metadata that an application wishes to create can go into these containers.
Key-Value Pairs	nmf:Metadata mpvm:Properties mpvm:KeyValue mpvmkv:Key nmf:Metadata mpvm:Properties mpvm:KeyValue mpvmkv:Value	Not used.
Album ("Audiobook")	mpvp:Album	
Number of Entries	implicit – number of entries in the foreground	
Title	nmf:Metadata dc:Properties dc:title	Not used as primary metadata for Audiobooks. SHOULD be populated with the value of ab:BookTitle for enhanced compatibility with MPV players.
Book Title	nmf:Metadata ab:Properties ab:BookTitle	Title of the overall book.
Contributor	nmf:Metadata dc:Properties dc:contributor	Not used as primary metadata for Audiobooks. SHOULD be populated with the value of ab:BookNarrator for enhanced compatibility with MPV players.
Narrator	nmf:Metadata ab:Properties ab:BookNarrator	Primary audiobook metadata.
Creator	nmf:Metadata dc:Properties dc:creator	Not used as primary metadata for audiobooks. SHOULD be populated with the value of ab:BookAuthor for enhanced compatibility with MPV players.
Author	nmf:Metadata ab:Properties ab:BookAuthor	Primary audiobook metadata.

Table 22 (continued)

Album Audiobook	mpvp:Album	
Principal Artist	nmf:Metadata mpvm:MusicProperties mpvm:PrincipalArtist	Not used as primary metadata for audiobooks. SHOULD be populated with the value of ab:BookAuthor for enhanced compatibility with MPV Music players.
Description	nmf:Metadata dc:Properties dc:description	Free-form commentary, message, description in plain text.
Copyright statement	nmf:Metadata dc:Properties dc:rights	Free-form plain text of the Copyright Statement of rights.
Audiobook Content Items	mpvp:Album mpvp:Foreground	This content is played as the principal content of the audiobook playlist.
Audiobook accompaniment	mpvp:Album mpvp:Background	This content is played as background to the audiobook content in the foreground. For example, it could specify images or background music.
Audiobook Artwork	mpvp:Album mpv:Related mpv:relationship="urn:osta-org:mpv:music:artwork" mpv:StillRef	Artwork for the audiobook.
Identifier	nmf:Metadata dc:Properties dc:identifier	Not used as primary metadata for audiobooks. SHOULD be populated with the value of ab:BookISBN for enhanced compatibility with MPV Music players.
Book UUID	nmf:Metadata ab:Properties ab:BookUUID	Primary metadata for book UUID.
Book ISBN Identifier	nmf:Metadata ab:Properties ab:BookISBN	Primary metadata for book ISBN.
Book ISSN Identifier	nmf:Metadata ab:Properties ab:BookISSN	Primary metadata for book ISSN.
Book DOI Identifier	nmf:Metadata ab:Properties ab:BookDOI	Primary metadata for book DOI.
Book Index Number	nmf:Metadata ab:Properties ab:BookIndexNumber	Primary metadata for book Book Index Number.
Book Shelf Number	nmf:Metadata ab:Properties ab:BookShelfNumber	Primary metadata for book Book Shelf Number.
Book Part Number	nmf:Metadata ab:Properties ab:BookPartNumber	Primary metadata for book Book Part Number.
Other Identifier	nmf:Metadata ab:Properties ab:OtherIdentifier	Primary metadata for any other book identifier.
Content Location Identifier	nmf:Metadata ab:Properties ab:LocationIdentifier	Primary metadata for play-list content location identifier.
Key-Value Pairs	nmf:Metadata mpvm:Properties mpvm:KeyValue mpvmkv:Key nmf:Metadata mpvm:Properties mpvm:KeyValue mpvmkv:Value	Not used.
Extra Data	mpv:Metadata and nmf:Metadata anywhere	

9.7 Dublin core metadata usage

To promote interoperability, MPV-Core makes use of the Dublin Core metadata (DC-NMF) to represent essential metadata across all types of assets. The dc:title element is used to specify the title of an audiobook asset, as the title of an image or video asset.

In 9.6, the Dublin Core metadata elements were mixed into the overall set of audiobook meta-data properties and assets. To clarify usage of DC-NMF, Table 23 includes only the Dublin Core elements.

Table 23 – Dublin Core metadata (DC-NMF)

Audio asset Chapter, Section	mpv:Audio subelements	
Creator ("Author")	nmf:Metadata dc:Properties dc:creator	Not used as primary metadata for audiobooks. SHOULD be populated with the value of ab:BookAuthor for enhanced compatibility with MPV players.
Contributor ("Narrator")	nmf:Metadata dc:Properties dc:contributor	Not used as primary metadata for audiobooks. SHOULD be populated with the value of ab:BookNarrator for enhanced compatibility with MPV players.
Description	nmf:Metadata dc:Properties dc:description	Commentary, message, description in plain text.
Format	nmf:Metadata dc:Properties dc:format	Use MPV-provided list of well-known MIME types.
Identifier ("ISBN")	nmf:Metadata dc:Properties dc:identifier	Not used as primary metadata for audiobooks. SHOULD be populated with the value of ab:BookISBN for enhanced compatibility with MPV Music players.
Keywords	nmf:Metadata dc:Properties dc:subject	Semicolon separated list of plain text keywords or phrases for search. Semi-colon can be used by escaping with backslash, for example \;
Language	nmf:Metadata dc:Properties dc:language	"<language>_<territory>" where <language> is from [ISO 639-1] and <territory> is [ISO 3166-1] Example: "en_US"
Publisher	nmf:Metadata dc:Properties dc:publisher	Free-form plain text.
Rights ("Copyright Statement")	nmf:Metadata dc:Properties dc:rights	Free-form plain text copyright statement.
Title	nmf:Metadata dc:Properties dc:title	Free-form plain text of the title of the chapter.
Created Date	nmf:Metadata dcterms:Properties dcterms:created	Date-time audiobook performed or recorded.
Medium	nmf:Metadata dcterms:Properties dcterms:medium	Original audiobook release medium conformant to [ID3v240]

Table 23 (continued)

Album (audiobook) of spoken word	mpvp:Album subelements	Comments
Creator ("Author")	nmf:Metadata dc:Properties dc:creator	Not used as primary metadata for audiobooks. SHOULD be populated with the value of ab:BookAuthor for enhanced compatibility with MPV players.
Contributor ("Narrator")	nmf:Metadata dc:Properties dc:contributor	Not used as primary metadata for audiobooks. SHOULD be populated with the value of ab:BookNarrator for enhanced compatibility with MPV players.
Description	nmf:Metadata dc:Properties dc:description	Free-form commentary, message, description in plain text.
Identifier ("Book ISBN")	nmf:Metadata dc:Properties dc:identifier	Not used as primary metadata for audiobooks. SHOULD be populated with the value of ab:BookISBN for enhanced compatibility with MPV Music players.
Publisher	nmf:Metadata dc:Properties dc:publisher	Free-form plain text.
Rights ("Copyright Statement")	nmf:Metadata dc:Properties dc:rights	Free-form plain text of the copyright statement.
Title	nmf:Metadata dc:Properties dc:title	Free-form plain text of the title of the audiobook.

9.8 Audiobook profile using MPV-Music playlists

9.8.1 General

MPV-Audiobook uses MPVMusic playlists to organize audio assets and other content on a storage media. An MPV-Music playlist identifies the primary assets to be presented to the user, such as a set of audiobook chapters or sections. The MPV-Music playlist also specifies the sequence in which assets should be presented and other presentation characteristics.

MPV-Music playlists are implemented based on the MPV presentation profile specification [MPV-Pres]. The MPV-Music profile specifies the details of this usage for music content. This usage is maintained and refined for audiobook use.

An MPV-Music playlist uses the <mpvp:Album> element, which may contain metadata, foreground and background content, and related and rendition assets. An example is below.

```
<mpvp:Album>
```

```
  <mpv:Metadata> ... </mpv:Metadata>
```

```
  <nmf:Metadata> ... </nmf:Metadata>
```

```
  <mpvp:Background> ... </mpvp:Background>
```

```
  <mpvp:Foreground> ... </mpvp:Foreground>
```

<mpv:Rendition> ... </mpv:Rendition>

<mpv:Related> ... </mpv:Related>

</mpvp:Album>

MPV-Music anticipates that users will organize media files into folders/directories of their own choosing and assign long filenames. Content location and playback order may not be specified solely by requiring specific directory and filenames. Instead, MPV-Music provides an approach permitting the content to be located anywhere and with any name.

In addition, many related media assets may be present on storage media, such as CD, to enhance access and playback performance and to provide enhanced playback and printing experiences. The MPV playlist allows this content to be managed to present the user a simple high-level interaction with their content.

9.8.2 Number of playlists per MPV file

An MPV file conforming to the MPV-Audiobook shall contain one playlist (mpvm:Album element).

9.8.3 Metadata usage

Metadata placed on the mpvp:Album element defines the scope of the whole album or playlist, and metadata properties for this usage shall be as defined in Table 24. For example, the mpvm:PrincipalArtist element may be used with either a single music asset or on the mpvp:Album.

Table 24 – Metadata properties usage

Album Audiobook	mpvp:Album	
Number of Entries	implicit – number of entries in the foreground	
Title	nmf:Metadata dc:Properties dc:title	Not used as primary metadata for audiobooks. SHOULD be populated with the value of ab:BookTitle for enhanced compatibility with MPV players.
Book Title	nmf:Metadata ab:Properties ab:BookTitle	Title of the overall book from which the chapter came.
Book Subtitle	nmf:Metadata ab:Properties BookSubtitle	Subtitle of the overall book from which the chapter came.
Contributor	nmf:Metadata dc:Properties dc:contributor	Not used as primary metadata for audiobooks. SHOULD be populated with the value of ab:BookNarrator for enhanced compatibility with MPV players.
Narrator	nmf:Metadata ab:Properties ab:BookNarrator	Primary audiobook metadata.
Creator	nmf:Metadata dc:Properties dc:creator	Not used as primary metadata for audiobooks. SHOULD be populated with the value of ab:BookAuthor for enhanced compatibility with MPV players.
Author	nmf:Metadata ab:Properties ab:BookAuthor	Primary audiobook metadata.
Book Coauthor	nmf:Metadata ab:Properties BookCoauthor	Coauthor of the overall book from which the chapter came.

Table 24 (continued)

Album Audiobook	mpvp:Album	
Principal Artist	nmf:Metadata mpvm:MusicProperties mpvm:PrincipalArtist	Not used as primary metadata for audiobooks. SHOULD be populated with the value of ab:BookAuthor for enhanced compatibility with MPV Music players.
Book Genre	nmf:Metadata ab:Properties BookGenre	Genre of the overall book from which the chapter came.
Description	nmf:Metadata dc:Properties dc:description	Description summary of book.
Copyright statement	nmf:Metadata dc:Properties dc:rights	Free-form plain text of the Copyright Statement of rights.
Audiobook Content Items	mpvp:Album mpvp:Foreground	This content is played as the principal content of the audiobook playlist.
Audiobook accompaniment	mpvp:Album mpvp:Background	This content is played as background to the audiobook content in the foreground. For example, it could specify images or background music.
Audiobook Artwork	mpvp:Album mpv:Related mpv:relationship="urn:osta-org:mpv:music:artwork" mpv:StillRef	Artwork for the audiobook.
Identifier	nmf:Metadata dc:Properties dc:identifier	Not used as primary metadata for audiobooks. SHOULD be populated with the value of ab:BookISBN for enhanced compatibility with MPV Music players.
Book UUID	nmf:Metadata ab:Properties ab:BookUUID	
Book ISBN Identifier	nmf:Metadata ab:Properties ab:BookISBN	
Book ISSN Identifier	nmf:Metadata ab:Properties ab:BookISSN	
Book DOI Identifier	nmf:Metadata ab:Properties ab:BookDOI	
Book Index Number	nmf:Metadata ab:Properties ab:BookIndexNumber	
Book Shelf Number	nmf:Metadata ab:Properties ab:BookShelfNumber	
Book Part Number	nmf:Metadata ab:Properties ab:BookPartNumber	
Other Identifier	nmf:Metadata ab:Properties ab:OtherIdentifier	
Content Location Identifier	nmf:Metadata ab:Properties ab:LocationIdentifier	
Key-Value Pairs	nmf:Metadata mpvm:Properties mpvm:KeyValue mpvm kv:Key nmf:Metadata mpvm:Properties mpvm:KeyValue mpvm kv:Value	Not used.
Extra Data	mpv:Metadata and nmf:Metadata anywhere	

9.8.4 Background usage

The MPV-Audiobook may contain the mpvp:Background element to provide background presentation content to the spoken word in the foreground. It is applied at the discretion of the playback application or device. If it is a simple device or application with no visual presentation capabilities, the mpvp:Background content should be ignored.

If the application or device has visual presentation capabilities, the mpvp:Background content may be processed. The recommended content is a series of mpv:StillRef elements that refer

to still images that may be displayed behind the foreground content. This can be used in both browse/menu mode as well as playback mode. All the MPV-Pres properties may be applied, such as transitions. These are honored at the discretion of the playback application.

9.8.5 Foreground usage

An MPV-Audiobook shall include a foreground list mpvp:Foreground in mpvp:Album, which shall be identified using the mpvp:AudioRef element. mpvp:AudioRef defines the set of primary spoken word items in the playlist and their sequence. In addition to the referenced audio assets, the mpv:AssetList in the MPV file may also contain still image, video, and other kinds of assets. mpvp:Album explicitly identifies which assets in the mpv:AssetList are primary.

9.8.6 Related and rendition assets

mpvp:Album may identify renditions and related assets. Cover artwork may be specified using a mpv:Related element with an urn:osta-org:mpv:music:artwork relationship identifier.

9.9 Groups of assets in playlists

There may be situations in which a playlist has so many assets that it is inconvenient to navigate one item at a time. The playback application or device may choose to implement a FastForward or FastRewind function that jumps ahead by multiples of items.

MPV-generating applications may also choose to group assets within playlists. The playback device may choose to implement FastForward or FastRewind functions by jumping ahead by groups; group navigation may also be explicit, if desired by the playback device.

Groups are specified using standard MPV functionality. A group of assets to be played sequentially shall be identified using the mpv:Seq asset. When a playback application or device encounters a mpv:Seq asset, the application or device should play the contents sequentially.

When mpv:Seqs are used with a mpvp:Album, only the mpv:Seq assets shall be listed in the album. When the playback application or device reads mpv:Seq assets, then these assets shall play the mpv:Seq contents in sequence. See below for an example of MPV asset group's implementation.

```
<?xml version="1.0" encoding="UTF-8"?>
<file:Manifest
xmlns:file="http://ns.osta.org/manifest/1.0/"
xmlns:ab="http://ns.ce.org/audiobook/2003b/"
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:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcterms="http://http://purl.org/dc/terms/"
xmlns:nmf="http://ns.osta.org/nmf/1.0/"
  <nmf:Metadata>
    <ManifestProperties xmlns="http://ns.osta.org/manifest/1.0/">
```

```

    <ProfileBag>
      <Profile>http://ns.ce.org/audiobook/2003b/</Profile>
      <Profile>http://ns.osta.org/mpv/presentation/1.0/</Profile>
      <Profile>http://ns.osta.org/mpv/music/1.0/</Profile>
    </ProfileBag>
  </ManifestProperties>
  <Properties xmlns="http://purl.org/dc/terms/">
    <created>2003-01-29T15:51:41Z"</created>
    <modified>2003-01-29T15:51:41Z"/></modified>
  </Properties>
</nmf:Metadata>

<mpvp:Album>
  <nmf:Metadata>
    <dc:Properties>
      <dc:creator>Ima Smoothtalker</dc:creator>
      <dc:identifier>put an ISBN number here</dc:identifier>
      <dc:rights>Copyright (c)2003 HomeMade Press<dc:rights/>
      <dc:title>Life Along the Willamette</dc:title>
    </dc:Properties>
    <mpvm:MusicProperties>
      <mpvm:AlbumTitle>Life Along the
Willamette</mpvm:AlbumTitle>
      <mpvm:Genre>Spoken Word:Non-Fiction</mpvm:Genre>
    </mpvm:MusicProperties>
    <ab:AudiobookProperties>
      <ab:BookAuthor>Ura Writer</ab:BookAuthor>
      <ab:BookISBN>put an ISBN number here</ab:BookISBN>
      <ab:BookNarrator>Ima Smoothtalker</ab:BookNarrator>
      <ab:BookTitle>Life Along the Willamette</ab:BookTitle>
    </ab:AudiobookProperties>
  </nmf:Metadata>
  <mpvp:Foreground>
    <mpv:SeqRef mpv:idRef="IDSeq0001"/> <!-- chapter 1 -->

```

```

        <mpv:SeqRef mpv:idRef="IDSeq0002"/> <!-- chapter 2 -->
    </mpvp:Foreground>
</mpvp:Album>

<mpv:AssetList>
    <mpv:Seq mpv:id="IDSeq0001"> <!-- chapter 1 -->
        <mpv:AudioRef mpv:idRef="ID000001"/> <!-- page -->
        <mpv:AudioRef mpv:idRef="ID000002"/> <!-- page -->
        <mpv:AudioRef mpv:idRef="ID000003"/> <!-- page -->
    </mpv:Seq>
    <mpv:Seq mpv:id="IDSeq0002"> <!-- chapter 2 -->
        <mpv:AudioRef mpv:idRef="ID000004"/> <!-- page -->
        <mpv:AudioRef mpv:idRef="ID000005"/> <!-- page -->
        <mpv:AudioRef mpv:idRef="ID000006"/> <!-- page -->
    </mpv:Seq>
    <mpv:Audio mpv:id="ID000001">
        <nmf:Metadata>
            <Properties xmlns="http://purl.org/dc/elements/1.1">
                <contributor>Ima Smoothtalker</contributor>
                <creator>Ura Writer</creator>
                <format>audio/mpeg</format>
                <title>page 1</title>
            </Properties>
            <MusicProperties xmlns="http://ns.osta.org/mpv/music/1.0/">
                <AlbumTitle>Life Along the Willamette</AlbumTitle>
                <Genre>Spoken Word:Non-Fiction</Genre>
                <PlayingTime>3.48</PlayingTime>
            </MusicProperties>
        </nmf:Metadata>
        <mpv:LastURL>audio/page1.mp3</mpv:LastURL>
    </mpv:Audio>
    <mpv:Audio mpv:id="ID000002">

```

```

<nmf:Metadata>
  <Properties xmlns="http://purl.org/dc/elements/1.1">
    <contributor>Ima Smoothtalker</contributor>
    <creator>Ura Writer</creator>
    <format>audio/mpeg</format>
    <title>page 2</title>
  </Properties>
  <MusicProperties xmlns="http://ns.osta.org/mpv/music/1.0/">
    <AlbumTitle>Life Along the Willamette</AlbumTitle>
    <Genre>Spoken Word:Non-Fiction</Genre>
    <PlayingTime>3.24</PlayingTime>
  </MusicProperties>
</nmf:Metadata>
<mpv:LastURL>audio/page2.mp3</mpv:LastURL>
</mpv:Audio>
</mpv:AssetList>
</file:Manifest>

```

9.10 MPV-Audiobook schema details

9.10.1 XML schema definition

The XML Schema for an MPV-Audiobook shall be as specified below.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schematarget Namespace="http://ns.ce.org/audiobook/2003b/"
xmlns=http://ns.ce.org/audiobook/2003b/
xmlns:nmft="http://ns.osta.org/nmf/1.0/tools/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:nmf=http://ns.osta.org/nmf/1.0/
elementFormDefault="qualified" attributeFormDefault="qualified">
  <xs:import namespace="http://ns.osta.org/nmf/1.0/" schemaLocation="nmf.xsd"/>
  <xs:annotation>
    <xs:documentation>The CEA R6 WG11 Audiobook Properties
schema</xs:documentation>
  </xs:annotation>

```

```

<!--name for BySchemaProperties element: AudiobookProperties-->

<xs:element          name="AudiobookProperties"          type="BySchemaPropsType"
substitutionGroup="nmf:BySchemaPropsBase"/>

<!--top-level schema element type. Note that to be MPV NMF compliant, the order of
the elements MUST be alphabetical. -->

<xs:complexType name="BySchemaPropsType">
    <xs:complexContent>
        <xs:extension base="nmf:BySchemaPropsType">
            <xs:sequence>
                <xs:element ref="BookAuthor" minOccurs="0"/>
                <xs:element ref="BookDOI" minOccurs="0"/>
                <xs:element ref="BookIndexNumber" minOccurs="0"/>
                <xs:element ref="BookISBN" minOccurs="0"/>
                <xs:element ref="BookISSN" minOccurs="0"/>
                <xs:element ref="BookNarrator" minOccurs="0"/>
                <xs:element ref="BookPartNumber" minOccurs="0"/>
                <xs:element ref="BookShelfNumber" minOccurs="0"/>
                <xs:element ref="BookTitle" minOccurs="0"/>
                <xs:element ref="BookUUID" minOccurs="0"/>
                <xs:element ref="OtherIdentifier" minOccurs="0"/>
                <xs:element ref="LocationIdentifier" minOccurs="0"/>
                <xs:element ref="unitType" minOccurs="0"/>
                <xs:element ref="unitContext" minOccurs="0"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:element name="BookTitle" type="BookTitleType"/>
<xs:complexType name="BookTitleType">
    <xs:simpleContent>
        <xs:extension base="xs:string"/>
    </xs:simpleContent>
</xs:complexType>

```