



BSI Standards Publication

A framework for digital preservation of cinematographic works — The Cinema Preservation Package

National foreword

This British Standard is the UK implementation of EN 17650:2022.

The UK participation in its preparation was entrusted to Technical Committee CPW/36, Cinematography.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2022
Published by BSI Standards Limited 2022

ISBN 978 0 539 12210 7

ICS 35.240.30; 37.060.99

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 September 2022.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

English Version

A framework for digital preservation of cinematographic works - The Cinema Preservation Package

Un cadre pour la conservation numérique des œuvres
cinématographiques - Le paquetage de conservation
cinéma

Ein Rahmenwerk für die digitale Erhaltung von
kinematografischen Werken - Das Cinema Preservation
Package

This European Standard was approved by CEN on 17 July 2022.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

Contents

Page

European foreword.....	6
Introduction	7
1 Scope	8
2 Normative references	8
3 Terms and definitions	10
4 Abbreviations	11
5 Syntax conventions used in this document.....	12
5.1 General.....	12
5.2 Expressions used to denote file or folder names.....	12
5.2.1 Composition of name with parts.....	12
5.2.2 Literal parts.....	12
5.2.3 Optional part.....	12
5.2.4 Alternative parts	12
5.2.5 Explicitly defined part	12
5.3 Expressions used to denote file or folder multiplicity	13
5.3.1 General.....	13
5.3.2 One occurrence or more	13
5.3.3 Zero occurrence or more.....	13
5.3.4 Optional file or folder	13
5.4 Typographic conventions.....	13
5.4.1 Monospaced fonts	13
5.4.2 Italic	13
5.5 Conventions used to denote XML content	13
5.5.1 General consideration	13
5.5.2 Hierarchy	13
5.5.3 Namespace.....	14
5.5.4 Subelements.....	14
5.5.5 Property.....	14
5.5.6 Conventions used in XML constraints list tables	14
6 Core structure.....	15
6.1 General description (informative).....	15
6.2 General constraints	15
6.3 Root Folder structure	15
6.3.1 Synthetic view (informative)	15
6.3.2 Root folder name.....	16
6.3.3 Root folder content.....	16
6.4 Subpackage folder structure.....	17
6.4.1 Synthetic view (informative)	17
6.4.2 List of existing subpackages.....	17
6.4.3 Subpackage UUID	18
6.4.4 Subpackage folder name	18
6.4.5 Subpackage folder content	18
6.5 Root Metadata folder structure	20
6.5.1 General description (informative).....	20

6.5.2	Synthetic view (informative).....	20
6.5.3	Root Metadata folder content	20
6.5.4	Descriptive metadata files.....	20
6.5.5	Provenance metadata files.....	20
6.6	Root Ancillary Data folder structure	21
6.6.1	General description (informative)	21
6.6.2	Synthetic view (informative).....	21
6.6.3	Root Ancillary Data folder content.....	21
6.6.4	Root Ancillary Data filename.....	21
6.7	Playlists folder structure	21
6.7.1	General description (informative)	21
6.7.2	Synthetic view (informative).....	21
6.7.3	Playlists folder content.....	22
6.7.4	Playlist UUID.....	22
6.7.5	Playlist filename.....	22
6.8	Root Checker Report folder structure	22
6.8.1	General description (informative)	22
6.8.2	Synthetic view	22
6.8.3	Root Checker Reports folder content	22
6.8.4	Content of the Report subfolder.....	23
6.8.5	Reports filename.....	23
6.9	Preservation Package UUID	23
6.9.1	UUID generation.....	23
6.9.2	UUID presentation.....	23
6.9.3	UUID uniqueness.....	23
7	Profiles	23
7.1	General description (informative)	23
7.2	Profile attribution	23
8	Content of Packing List files.....	24
8.1	Synthetic view (informative).....	24
8.2	Packing List.....	25
8.2.1	General description (informative)	25
8.2.2	File conformance	25
8.2.3	Common constraints on Packing List	25
8.3	Preservation Packing List.....	27
8.3.1	General description (informative)	27
8.3.2	Files to be referenced.....	27
8.3.3	Preservation Package Unique Identifier uniqueness.....	28
8.3.4	Specific constraints on the Preservation Packing List	28
8.4	Subpackage Packing List.....	35
8.4.1	General description (informative)	35
8.4.2	Files to be referenced.....	35
8.4.3	Subpackage Unique Identifier uniqueness.....	35
8.4.4	Constraints on the Subpackage Packing List	35
9	Specification of the subpackages	40
9.1	Image Package	40
9.1.1	General description	40
9.1.2	General provisions	40
9.1.3	Data folder content	40
9.1.4	Sequence homogeneity.....	40
9.1.5	Sequence continuity	40

9.1.6	Image filenames.....	40
9.1.7	Image numbering.....	40
9.1.8	Image Package standard conformance	40
9.1.9	Technical Metadata	41
9.2	Sound Package.....	42
9.2.1	General description.....	42
9.2.2	General provisions.....	42
9.2.3	Data folder content.....	42
9.2.4	Soundfield consistency.....	42
9.2.5	Homogeneity.....	42
9.2.6	Time representation consistency.....	42
9.2.7	Sound file names	42
9.2.8	Sound Package Standard conformance	42
9.2.9	Technical Metadata	43
9.3	Timed Text Package	43
9.3.1	General description.....	43
9.3.2	General provisions.....	43
9.3.3	Data folder content.....	43
9.3.4	Timed text filename	43
9.3.5	Technical Metadata	44
9.3.6	Constraints	44
9.3.7	Timed Text Package Standard conformance.....	44
9.4	Audiovisual Package	44
9.4.1	General description.....	44
9.4.2	General provisions.....	44
9.4.3	Data folder content.....	44
9.4.4	Audiovisual filename	44
9.4.5	Audiovisual Packages standard conformance.....	45
9.4.6	Technical Metadata	46
9.5	Componentized Package.....	46
9.5.1	General description (informative).....	46
9.5.2	General provisions.....	46
9.5.3	Data folder content.....	46
9.5.4	Componentized Packages standard conformance	47
9.5.5	Technical Metadata	47
9.6	Extra Package	47
9.6.1	General description.....	47
9.6.2	General provisions.....	47
9.6.3	Data folder content.....	48
9.6.4	Extra Package filename.....	48
9.6.5	Technical Metadata	48
10	Ancillary Data	48
10.1	General description (informative).....	48
10.2	Root Ancillary Data.....	48
10.2.1	General description (informative).....	48
10.2.2	File conformance.....	48
10.2.3	Constraints on text files	49
10.3	Subpackage Ancillary Data	49
10.3.1	General description (informative).....	49
10.3.2	File conformance.....	49
11	Metadata.....	49
11.1	General metadata files description	49

11.1.1	General description (informative)	49
11.1.2	Languages	49
11.1.3	Character encoding	49
11.2	Root Metadata	49
11.2.1	Descriptive metadata	49
11.2.2	Provenance metadata	55
11.3	Subpackage metadata	55
11.3.1	Technical metadata	55
11.3.2	Provenance metadata	78
12	Specification of the playlist	78
12.1	File content	78
12.2	File conformance	78
12.3	Constraints on the Playlist XML file	79
12.3.1	EssenceDescriptor elements	79
12.3.2	Resource elements	79
12.3.3	Sequence elements	79
12.4	File revision	79
13	Checker reports	79
13.1	General description (informative)	79
13.2	Constraints on the XML Checker Reports List files	79
13.2.1	General description (informative)	79
13.2.2	General constraints	79
13.2.3	Constraints on the checkerDescription element	80
13.2.4	Constraints on the checkerProcessDescription element	80
13.2.5	Constraints on the checkerFileDescription element	80
Annex A (normative)	XML Schema for the Packing List	82
Annex B (normative)	XML Schema for the Playlist	83
Annex C (normative)	XML Schema for the Checker Reports List	84
Annex D (normative)	Tables from ISO/IEC 21122-3:2022	86
D.1	Interlaced modes	86
D.2	Sampling structure	86
Annex E (informative)	Synthetic package view	87
Bibliography		88

European foreword

This document (EN 17650:2022) has been prepared by Technical Committee CEN/TC 457 “Digital preservation of cinematographic works”, the secretariat of which is held by DIN.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by February 2023, and conflicting national standards shall be withdrawn at the latest by February 2023.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN shall not be held responsible for identifying any or all such patent rights.

This document is complemented by CEN/TR 17862:2022.

Any feedback and questions on this document should be directed to the users’ national standards body. A complete listing of these bodies can be found on the CEN website.

According to the CEN-CENELEC Internal Regulations, the national standards organisations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and the United Kingdom.

Introduction

This document is part of a series of standards and technical recommendations for the digital preservation of cinematographic works. It gives European film archives and producers a guideline for storing and managing films in the digital age. The document references many existing formats and elements and serves as a super-format which includes other existing sub-formats like DCPs or IMPs for movies, XML-files for packing-lists or subtitles, AV files and metadata files. In addition, methods are defined to ensure data integrity and quality by calculating hash and fingerprinting-values.

This document reuses describing tools and archiving formats from the cinema archiving community as much as possible.

1 Scope

This document specifies the Cinema Preservation Package (CPP) to facilitate the digital preservation of cinematographic works. It specifies methods to describe the relationship of components of a cinematographic work and delivers a syntax to describe the package content. The document itself specifies the structure of the package and the constraints that are necessary to enable compliance and interoperability.

Versions of the content using different encoding formats can be preserved in a layered structure where the lowest level is describing the physical file. The files can carry data representing moving images, sound, metadata or ancillary information like quality control (QC) reports or film posters.

The Cinema Preservation Package also contains hash values on different levels to ensure data integrity and version control. The syntax for this description and the methods for the hash value generation are specified in this document. Various types of packages are described as reference for practical implementations.

The Cinema Preservation Package is designed to serve as a Submission Information Package (SIP) in an OAIS compliant preservation system, and it may be used as a self-contained exchange format between media archives (as Distribution Information Package DIP). Based on the requirements of an archive, it can also be used as complete Archival Information Package (AIP) if the constraints set out in this document for such use are adhered to.

A CPP does not necessarily contain a complete cinematographic work, it can also be used for the exchange of parts of a work.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

EN 15744, *Film identification — Minimum set of metadata for cinematographic works*

EBU Tech 3264-E, *Specification of the EBU Subtitling data exchange format*

EBU Tech 3285, *Specification of the Broadcast Wave Format (BWF)*, Geneva, May 2011

EBU Tech 3293¹, *EBU Core Metadata Set (EBUCore), Specification v. 1.10*, Geneva, April 2020

EBU Tech 3350, *EBU-TT, part 1 subtitling format definition*, Geneva, May 2017

ETSI TS 102 366, *Digital Audio Compression (AC-3, Enhanced AC-3) Standard*, September 2017

IEEE/Open Group 1003.1-2017, *IEEE Standard for Information Technology — Portable Operating System Interface (POSIX(TM)) Base Specifications, Issue 7*

IETF RFC 9043, *FFV1 Video Coding Format Versions 0, 1, and 3*

ISO 639-3:2007, *Codes for the representation of names of languages — Part 3: Alpha-3 code for comprehensive coverage of languages*

ISO 8601-1, *Date and time — Representations for information interchange — Part 1: Basic rules*

¹ Referred to as EBUCore throughout the text.

ISO 8601-2, *Date and time — Representations for information interchange — Part 2: Extensions*

ISO 12234-2, *Electronic still-picture imaging — Removable memory — Part 2: TIFF/EP image data format*

ISO 19005-1, *Document management — Electronic document file format for long-term preservation — Part 1: Use of PDF 1.4 (PDF/A-1)*

ISO 19005-2, *Document management — Electronic document file format for long-term preservation — Part 2: Use of ISO 32000-1 (PDF/A-2)*

ISO 19005-4, *Document management — Electronic document file format for long-term preservation — Part 4: Use of ISO 32000-2 (PDF/A-4)*

ISO 26429 (all parts), *Digital cinema (D-cinema) packaging*

ISO/IEC 9834-8, *Information technology — Procedures for the operation of object identifier registration authorities — Part 8: Generation of universally unique identifiers (UUIDs) and their use in object identifiers*

ISO/IEC 10646, *Information technology — Universal Coded Character Set (UCS)*

ISO/IEC 10918 (series), *Information technology — Digital compression and coding of continuous-tone still images*

ISO/IEC 13818-2, *Information technology — Generic coding of moving pictures and associated audio information — Part 2: Video*

ISO/IEC 13818-3, *Information technology — Generic coding of moving pictures and associated audio information — Part 3: Audio*

ISO/IEC 14496-3, *Information technology — Coding of audio-visual objects — Part 3: Audio*

ISO/IEC 14496-10, *Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding*

ISO/IEC 14496-12, *Information technology — Coding of audio-visual objects — Part 12: ISO base media file format*

ISO/IEC 14496-14, *Information technology — Coding of audio-visual objects — Part 14: MP4 file format*

ISO/IEC 15444-1, *Information technology — JPEG 2000 image coding system — Part 1: Core coding system*

ISO/IEC 15948, *Information technology — Computer graphics and image processing — Portable Network Graphics (PNG): Functional specification*

ISO/IEC 23091-3:2018, *Information technology — Coding-independent code points — Part 3: Audio*

ITU-T H.273:2021, *Coding-independent code points for video signal type identification*

ITU-R BS.1352-3, *File format for the exchange of audio programme materials with metadata on information technology media*

SMPTE ST 12-1, *SMPTE Standard — For Television — Time and Control Code*

SMPTE ST 268, *SMPTE Standard — File Format for Digital Moving-Picture Exchange (DPX)*

SMPTE ST 428-7, *SMPTE Standard — Digital Cinema Distribution Master — Subtitle*

SMPTE ST 429 (all parts), *SMPTE Standard - D-Cinema Packaging*

SMPTE ST 2019-1, *SMPTE Standard — VC-3 Picture Compression and Data Stream Format*

SMPTE ST 2065-4, *SMPTE Standard — ACES Image Container File Layout*

SMPTE ST 2067-3:2020, *SMPTE Standard — Interoperable Master Format — Composition Playlist*

SMPTE ST 2067-21, *SMPTE Standard — Interoperable Master Format — Application #2E*

SMPTE ST 2067-40, *SMPTE Standard — Interoperable Master Format — Application #4 Cinema Mezzanine*

SMPTE ST 2067-50, *SMPTE Standard — Interoperable Master Format — Application #5 ACES*

W3C - *TTML Profiles for Internet Media Subtitles and Captions 1.1*

Matroska MKV format specification. Available from: <https://www.matroska.org>

METS Metadata Encoding & Transmission Standard. Version 1.12.1. [online]. Library of Congress, 2019. Available from: <https://www.loc.gov/standards/mets/>

PREMIS Data Dictionary for Preservation Metadata. Version 3.0. [online]. Library of Congress, 2015. Available from: <https://www.loc.gov/standards/premis/>

SCHEMA P.R.E.M.I.S. 3.0 [online]. Library of Congress, 2015. Available from: <https://www.loc.gov/standards/premis/v3/premis-v3-0.xsd>

Xiph FLAC format specification. Available from: <https://xiph.org/flac/format.html>

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1

audio channel

sequence of data representing a single audio signal

3.2

audio track

part of a sound file containing one or more audio channels or multiple single audio channels files forming one sound sequence

3.3

cinema preservation package

package containing digital data of a cinematographic work

3.4**container**

file containing one or more data objects wrapped in a specific way

3.5**package**

set of files or folders belonging together

3.6**provenance**

metadata related to the origin, modification and the ownership history of a work

3.7**soundfield**

three-dimensional pressure distribution produced by transmitted sound energy

NOTE A soundfield can be represented as a discrete set of signals, monophonic or multichannel, or other representation.

3.8**sound sequence**

continuous group of sound samples related to a specific period of time

3.9**submission information package**

package sent from the data producer to the archive

NOTE The submission information package is specified in the ISO 14721:2012 OAIS Model.

3.10**subpackage**

set of data in a Cinema Preservation Package containing part of a cinematographic work with data and supplemental data separately stored in a folder

3.11**subpicture**

picture to be overlayed on the main picture

4 Abbreviations

ACES	Academy Color Encoding System
ASCII	American Standard Code for Information Interchange
BMFF	Base Media File format
CPP	Cinema Preservation Package
DCP	Digital Cinema Package
DPX	Digital Picture Exchange
FLAC	Free Lossless Audio Codec
IMF	Interoperable Master Format
JPEG	Joint Photographic Experts Group

LoC	Library of Congress, see https://www.loc.gov/
MPEG	Moving Pictures Experts Group
OAIS	Open Archive Information System
PCM	Pulse code modulation
PDF	Portable Document Format
PNG	Portable Network Graphics
SIP	Submission Information Package
TIFF	Tagged Image File Format
UUID	Universal Unique Identifier
XML	Extensible Markup Language

5 Syntax conventions used in this document

5.1 General

For the file and folder name descriptions the following conventions are used.

5.2 Expressions used to denote file or folder names

5.2.1 Composition of name with parts

A name may be composed of one or several parts.

5.2.2 Literal parts

A literal part appearing between quotes shall appear as is.

EXAMPLE “LiteralPart” designates a part which is exactly “LiteralPart”.

5.2.3 Optional part

A part appearing between brackets shall appear zero or one time.

EXAMPLE [“OptionalPart”]“Radical” designates a name which is either “OptionalPartRadical” or “Radical”.

5.2.4 Alternative parts

Exactly one of the items in parentheses, separated by a vertical line ‘|’, shall appear in the result.

EXAMPLE (“Prefix1”|“Prefix2”|“Prefix3”)“Radical” designates a name which is either “Prefix1Radical”, “Prefix2Radical” or “Prefix3Radical”.

5.2.5 Explicitly defined part

<any text here> is a placeholder text and shall be replaced as specified by the prose.

EXAMPLE “Name_”<custom identifier> designates a name which could be “Name_0123456789”, supposing “0123456789” is defined as a valid custom identifier.

5.3 Expressions used to denote file or folder multiplicity

5.3.1 General

Some files or folders may appear several times in the parent folder. A trailing modifier indicates the allowed multiplicity.

5.3.2 One occurrence or more

A file or folder name with a trailing '+' shall appear one or more times in the parent folder.

EXAMPLE "File_"<number>+ designates a set of multiple files such as "File_0000", or "File_0000" and "File_0001" and "File_0002", etc., assuming the prose describes such a 4-digit numbering.

5.3.3 Zero occurrence or more

A file or folder name with a trailing '*' shall appear zero or more times in the parent folder.

EXAMPLE "File_"<number>* designates a set of multiple files such as no file, "File_0000", or "File_0000" and "File_0001" and "File_0002", etc., assuming the prose describes such a 4-digit numbering.

5.3.4 Optional file or folder

A file or folder name with a trailing '?' shall appear zero or one time in the parent folder.

EXAMPLE "OptionalFile"? designates no file, or exactly one file with name "OptionalFile".

5.4 Typographic conventions

5.4.1 Monospaced fonts

In the document, monospaced fonts are used for literal expression of textual content of a file, for instance as example quotation.

EXAMPLE <exampleXmlElement>example text</exampleXmlElement>

5.4.2 Italic

In the document, italic text is expected to be substituted as specified in the prose.

EXAMPLE *relative path to file* shall be the path from the root folder to the Descriptive Metadata file including the filename.

5.5 Conventions used to denote XML content

5.5.1 General consideration

Constraints on XML files are specified either as an XML Schema to apply, or as listed constraints for each element of a list of specific XML Nodes.

5.5.2 Hierarchy

To denote the hierarchical position of a node the XPath syntax is used.

EXAMPLE "/rootElement/subElement/subSubElement" denotes an element with name "<subSubElement>" which is a child of a "<subElement>" element, which itself is a child of the "<rootElement>" root element.

NOTE When an element in the path occurs more than one time, the XPath will be completed with discriminatory attributes if necessary. If the description applies to multiple elements, no discriminatory attribute will be added.

5.5.3 Namespace

XML elements may use a prefix to denote a namespace. The prefix indicated in the prose is only used for convenience, and does not denote a namespace. The prefix in the file may be different and this is suitable as long as the relation with the namespace is correctly indicated as a namespace attribute from a parent element.

EXAMPLE “/mainNs:rootElement/mainNs:subElement” the “<rootElement>” and the “<subElement>” belong to the namespace denoted by “mainNs” in the prose.

5.5.4 Subelements

The following are used:

- @attributeName: denotes the name of an attribute of the element
- #children: denotes the child elements
- #text: denotes the child text of the element

EXAMPLE

```
<mainNs:rootElement attribute1="example 1">
```

```
    <subElement1/>
```

```
    <subElement2>example 2</subElement2>
```

```
</mainNs:rootElement>
```

“/mainNs:rootElement/#children” denotes “subElement1” or “subElement2”

“/mainNs:rootElement/@attribute1” denotes “attribute1” with value “example 1”

“/mainNs:rootElement/subElement2/#text” denotes “example 2”

5.5.5 Property

If the XPath includes a part inside brackets, this is a constraint to properly address the right element.

EXAMPLE /mainNs:rootElement/subElement[@position='first'] denotes the <subElement> element which is a child of the root element <rootElement> and has an attribute “position” with value “first”.

5.5.6 Conventions used in XML constraints list tables

Prior to each table, the current XML element is reminded using the XPath syntax.

In the following sections, the presence of elements is specified using tables. In these tables the “Element Arity” column indicates the multiplicity allowed for an element:

- 1: the element is mandatory, exactly one element shall be present;
- 0..1: the element is optional, it shall appear zero or one time;
- 1..n: multiple mandatory. The element shall appear at least one time and may be repeated;
- 0..n: multiple optional. The element might not be present, or might be present one or multiple times.

For each element subelements are indicated. The “Subelement mandatory” column indicates if this subelement is mandatory. An ‘x’ in the corresponding cell indicates that the subelement shall be present, an ‘o’ in the same column indicates the subelement may be present. An attribute starts with the character ‘@’. #text denotes the child text, if mandatory, the element cannot be empty. #children indicates xml child elements.

Aiming to reduce the number of tables and sections in this document, when the semantic allows to do so consistently, subelements are sometimes documented in the same table when only few elements are described.

EXAMPLE ‘title/dc:title’ indicates that the subelement <title> contains itself a sub-subelement <dc:title>, meaning that the whole XML segment has to be added to the current element described.

```
<title typeLabel="titleType">
  <dc:title lang="fr"> title name</dc:title>
</title>
```

6 Core structure

6.1 General description (informative)

A Cinema Preservation Package is a set of files related to a cinematographic work.

The package is composed of subpackages, each of them including an atomic sequence of media, and replicating a similar structure with metadata and a Packing List file including checksums. Package related data and subpackages are located in subfolders of the root folder.

The Preservation Packing List file, at the root, lists the subfolders and checksums of the Subpackage Packing List files.

6.2 General constraints

The Cinema Preservation Package shall be a set of files organized as described in this document.

The stored items shall be related to the same work and variant.

Metadata files shall be located at the root of the file tree and in designated subdirectories.

6.3 Root Folder structure

6.3.1 Synthetic view (informative)

The root folder contains the following files and subfolders:

- <Preservation Package Instance name>;
- “preservationPackingList.xml”;
- “playlists”?;
- (“audiovisual”|“componentized”|“image”|“sound”|“timedText”|“extra”)“Package_”<uuid>+;
- “metadata”;
- “ancillaryData”?;
- “checkerReports”?.

6.3.2 Root folder name

The root folder name designated as <Preservation Package Instance name> should follow the pattern:
 <titleName>["_"<versionName>]["_"<work identifying system name>_"<work identifier>"]

Where:

- <titleName> should be the title of the work, using alphanumerical ASCII characters, with '-' as word separator;
- <versionName> if present should be the designation of the version of work, typically a variant, using alphanumerical ASCII characters, with '-' as word separator;
- <work identifying system name> should be a prefix of four alphabetical upper-case characters. If the identifying system is an archive private system, then the name should be 'PRIV';
- <work identifier> should be the identification of the work in the system denoted by the preceding prefix, using only alphanumerical ASCII characters and '-'.

NOTE The preservation facility can use the identification of the work from their local catalogues as work identifier.

EXAMPLE "Goal-The-dreams-begins_feature_ISAN_0000-0001-8947-0000-8-0000-0000-D".

6.3.3 Root folder content

6.3.3.1 Contained file

The root folder shall contain exactly one file.

The "preservationPackingList.xml" document shall be present and shall conform to provisions specified in 8.3.

6.3.3.2 Subpackage folders

One or more subpackage folders shall be present as children of the root folder. Each folder is a Subpackage and shall contain either an Image Package, a Sound Package, a Timed Text Package, an Audiovisual Package, a Componentized Package or an Extra Package, together with its related content.

Each subpackage folder shall conform to structure provisions as specified in 6.4 and conform to specific provisions:

- each Image Package folder shall conform to other provisions as specified in 9.1;
- each Sound Package folder shall conform to other provisions as specified in 9.2;
- each Timed Text Package folder shall conform to other provisions as specified in 9.3;
- each Audiovisual Package folder shall conform to other provisions as specified in 9.4;
- each Componentized Package folder shall conform to other provisions as specified in 9.5;
- each Extra Package folder shall conform to other provisions as specified in 9.6.

6.3.3.3 Playlists folder

One unique “playlists” folder may be present as a child of the root folder. This folder will be referred to as the Playlists folder in the rest of this document. This folder shall conform to structure provisions specified in 6.7 and other provisions in Clause 12.

6.3.3.4 Root Metadata folder

Exactly one “metadata” folder shall be present as a child of the root folder. This folder will be referred to as the Root Metadata folder in the rest of this document. This folder’s content shall conform to structure provisions specified in 6.5 and other provisions in 11.2.

6.3.3.5 Root Ancillary Data folder

One unique “ancillaryData” folder may be present as a child of the root folder. This folder will be referred to as the Root Ancillary Data folder in the rest of this document. This folder shall conform to structure provisions specified in 6.6 and other provisions in 10.2.

6.3.3.6 Root Checker Reports folder

One unique “checkerReports” folder may be present as a child of the root folder. This folder will be referred to as the Root Checker Reports folder in the rest of this document. This folder shall conform to structure provisions specified in 6.8 and other provisions in Clause 13.

6.4 Subpackage folder structure

6.4.1 Synthetic view (informative)

Subpackage folders contain the following files and subfolders:

- “packingList.xml”;
- (“sound”|“image”|“timedText”|“audiovisual”|“componentized”|“extra”)“Package_”<uuid>;
 - “data”;
 - <subpackage specific filenames>;
 - “metadata”;
 - “techMD_”<uuid>“_0000.xml”;
 - “digiprovMD_”<uuid>“_”<number>“.xml”*;
 - ancillaryData”?;
 - “ancillaryData_”<uuid>“_”<number>“.”<extension>.

6.4.2 List of existing subpackages

The Image Package, Sound Package, Timed Text Package, Audiovisual Package, Componentized Package, and Extra Package folders share common properties despite different behaviour related to the content described by the prefix. They will be designated as subpackages when referring to common properties.

6.4.3 Subpackage UUID

6.4.3.1 UUID generation

The Unique Universal Identifier shall conform to UUID version 4 according to ISO/IEC 9834-8.

6.4.3.2 UUID presentation

When a UUID appears in a filename, it shall be presented as the plain UUID character sequence, not as an URN.

EXAMPLE "imagePackage_dd779c52-0107-49d4-9a32-6b4ce4cd21f0".

6.4.3.3 UUID consistency

The same UUID shall be used as an abstract subpackage identifier within the Packing List file (see Clause 8), and in the subpackage folder name.

6.4.3.4 UUID uniqueness

Considering the content of the asset subfolders – files in the data and ancillaryData folder – in the subpackage folder as a unique set of files, a different set of files shall have a different UUID.

6.4.4 Subpackage folder name

The name of each subpackage folder shall follow the pattern: *<prefix>*"Package_"*<uuid>*.

Where:

- *<prefix>* shall be either "image" for an Image Package, "sound" for a Sound Package, "timedText" for a Timed Text Package, "audiovisual" for an Audiovisual Package, "componentized" for a Componentized Package, or "extra" for an Extra Package;
- *<uuid>* is the subpackage unique universal identifier of the subpackage as specified in 6.4.3.

6.4.5 Subpackage folder content

6.4.5.1 General

Each subpackage folder shall contain the following file and folders:

- the "packingList.xml" file shall be present and shall conform to provisions specified in 6.4.5.5;
- exactly one "data" folder shall be present. This folder will be referred to as the Subpackage Data folder;
- exactly one "metadata" folder shall be present. This folder will be referred to as the Subpackage Metadata folder;
- one "ancillaryData" folder may be present. This folder will be referred to as the Subpackage Ancillary folder.

6.4.5.2 Subpackage Data folder

The Subpackage Data folder shall contain the data required for each type of subpackage and shall conform to prescriptions listed in Clause 9.

6.4.5.3 Subpackage Metadata folder

6.4.5.3.1 General description

The Subpackage Metadata folder shall contain the metadata required for each type of subpackage and shall conform to prescriptions listed in 11.3.

6.4.5.3.2 Contained files

The Subpackage Metadata folder shall contain zero or one technical metadata file as specified for each kind of subpackage. If present, the name of this file shall be “techMD_”<uuid>“_”<technical metadata file number>“.xml”, where

- <uuid> shall be replaced with the Subpackage UUID described in 6.4.3;
- <technical metadata file number> shall be replaced with “0000”.

NOTE The numbering is provisioned for future use, but only one file is present.

The Subpackage Metadata folder may contain one or more provenance metadata files. The name of these files shall be “digiprovMD_”<uuid>“_”<provenance metadata file number>“.xml”, where

- <uuid> shall be replaced with the Subpackage UUID described in 6.4.3;
- <provenance metadata file number> shall be replaced with the next subpackage provenance metadata file number string.

All subpackage provenance metadata filenames shall be consistent, and share the following numbering properties:

- the numbering shall start at 0;
- no distinct subpackage provenance metadata file shall share the same number within the same metadata data folder.
- the decimal number shall be as a string of 4 digits with leading zeros.
- this *provenance metadata file number* shall be incremented for each new subpackage provenance metadata file.

EXAMPLE The first subpackage provenance metadata filename will be “digiprovMD_dd779c52-0107-49d4-9a32-6b4ce4cd21f0_0000.xml”, the second one “digiprovMD_dd779c52-0107-49d4-9a32-6b4ce4cd21f0_0001.xml”, etc., supposing the Subpackage UUID is “dd779c52-0107-49d4-9a32-6b4ce4cd21f0”.

6.4.5.4 Subpackage Ancillary Data folder

6.4.5.4.1 General description

The Ancillary Data folder contained files shall conform to provisions listed in 10.3.

6.4.5.4.2 Contained files

The Ancillary Data folder shall contain one or more ancillary data files.

The name of the files shall be <ancillary filename>“.”<extension>, where:

- <ancillary filename> shall be the filename given by the user with alphanumerical ASCII characters, ‘-’ or ‘_’;

- *<extension>* shall be the file extension with alphanumerical ASCII characters.

6.4.5.5 “packingList.xml”

The “packingList.xml” document shall conform to provisions listed in 8.4. The entities references specified in the Packing List shall use the Subpackage UUID as specified in 6.4.3.

6.5 Root Metadata folder structure

6.5.1 General description (informative)

This folder is targeted to convey the root metadata files for the whole package.

6.5.2 Synthetic view (informative)

- “metadata”;
- “descMD_”*<preservation package UUID>*“_”*<root descriptive metadata number>*“*.xml*”+;
- “digiprovMD_”*<preservation package UUID>*“_”*<root provenance metadata number>*“*.xml*”*.

6.5.3 Root Metadata folder content

The Root Metadata folder shall contain at least one descriptive metadata file, and may contain additional provenance metadata files as specified in 11.2.

6.5.4 Descriptive metadata files

The Root Metadata folder shall contain one or many descriptive metadata files.

The name of these files shall be “descMD_”*<preservation package UUID>*“_”*<root descriptive metadata number>*“*.xml*”, where:

- *<preservation package uuid>* as specified in 6.9;
- *<root descriptive metadata number>* shall be replaced with the next root descriptive metadata number string.

All descriptive metadata filenames shall be consistent, and share the following numbering properties:

- the numbering shall start at 0;
- no distinct descriptive metadata file shall share the same number;
- the decimal number shall be a string of 4 digits with leading zeros;
- the *root descriptive metadata number* shall be incremented for each new descriptive metadata file.

EXAMPLE the first descriptive metadata filename will be “descMD_59c3adeb-e1c3-40a1-8dae-25fef73f15a2_0000.xml”, the second one “descMD_59c3adeb-e1c3-40a1-8dae-25fef73f15a2_0001.xml”, etc., supposing the Preservation Package UUID is “59c3adeb-e1c3-40a1-8dae-25fef73f15a2”.

6.5.5 Provenance metadata files

The Root Metadata folder may contain zero, one or many provenance metadata files.

The name of these files shall be “digiprovMD_”*<preservation package uuid>*“_”*<root provenance metadata number>*“*.xml*”, where:

- *<preservation package uuid>* is as specified in 6.9;
- *<root provenance metadata number>* shall be replaced with the next root provenance metadata number string.

All provenance metadata filenames shall be consistent, and share the following numbering properties:

- the numbering shall start at 0;
- no distinct provenance metadata file shall share the same number;
- the decimal number shall be a string of 4 digits with leading zeros;
- the *root provenance metadata number* shall be incremented for each new provenance metadata file.

EXAMPLE the first provenance metadata filename will be “digiprovMD_59c3adeb-e1c3-40a1-8dae-25fef73f15a2_0000.xml”, the second one “digiprovMD_59c3adeb-e1c3-40a1-8dae-25fef73f15a2_0001.xml”, etc., supposing the preservation UUID is “59c3adeb-e1c3-40a1-8dae-25fef73f15a2”.

6.6 Root Ancillary Data folder structure

6.6.1 General description (informative)

This folder is targeted to convey documents for human readers related to the whole preservation package.

6.6.2 Synthetic view (informative)

- “ancillaryData”;
- *<filename>+.*

6.6.3 Root Ancillary Data folder content

The Root Ancillary Data folder shall contain at least one root ancillary data file as specified in 10.2.

6.6.4 Root Ancillary Data filename

The name of the files shall be *<root ancillary filename>“.”<extension>*, where:

- *<root ancillary filename>* shall be the filename given by the user with alphanumerical ASCII characters, ‘-’ or ‘_’;
- *<extension>* shall be the file extension with alphanumerical ASCII characters.

6.7 Playlists folder structure

6.7.1 General description (informative)

This optional folder is targeted to convey playlist files.

6.7.2 Synthetic view (informative)

- “playlists”;
- “playlist_”*<playlist_uuid>*“*.xml*”+.

6.7.3 Playlists folder content

One or more playlist file shall be present in the Playlists folder and shall conform to provisions specified in Clause 12.

6.7.4 Playlist UUID

6.7.4.1 UUID generation

The Unique Universal Identifier shall conform to UUID version 4 according to ISO/IEC 9834-8.

6.7.4.2 UUID presentation

When a UUID appears in a filename, it shall be presented as the plain UUID character sequence, not as an URN.

EXAMPLE "playlist_dd779c52-0107-49d4-9a32-6b4ce4cd21f0".

6.7.4.3 UUID uniqueness

The UUID shall be the UUID identification element of the playlist within the document, as described in 12.4.

6.7.5 Playlist filename

Each playlist filename shall conform to the pattern "playlist_<playlist UUID>.xml", where:

<playlist UUID> is the playlist UUID specified in 6.7.4.

EXAMPLE a playlist name could be: "playlist_ 9497a702-f3f2-4868-8e82-2535ec85bef8.xml".

6.8 Root Checker Report folder structure

6.8.1 General description (informative)

This folder contains quality control reports created after the generation of the subpackages.

6.8.2 Synthetic view

- "checkerReports";
 - "checkerReportsList.xml";
 - "reports_"<report process UUID>+;
 - <report UUID>"-result."<extension>+.

6.8.3 Root Checker Reports folder content

Exactly one Checker Report List file shall be present. The name of this file shall be "checkerReportsList.xml". This file shall comply with provisions in 13.2.

One or more Report subfolders shall be present.

The Report subfolder name shall be "reports_"<report process UUID>, where:

- <report process UUID> shall be an UUID representing the checker process.

6.8.4 Content of the Report subfolder

One or more report file shall be present.

6.8.5 Reports filename

The name of the report files shall be *<report UUID>“-result.”<extension>*, where:

- *<report UUID>* shall be a new UUID for each new report;
- *<extension>* shall be the file extension with alphanumerical ASCII characters.

6.9 Preservation Package UUID

6.9.1 UUID generation

The Unique Universal Identifier shall conform to UUID version 4 according to ISO/IEC 9834-8.

6.9.2 UUID presentation

When a UUID appears in a filename, it shall be presented as the plain UUID character sequence, not as an URN.

EXAMPLE “preservationPackage_65346dbc-c39f-4788-b87c-0193d59fff80”.

6.9.3 UUID uniqueness

The UUID shall be the UUID identification element of the Preservation Packing List within the document, as described in 8.3.3.

7 Profiles

7.1 General description (informative)

The Cinema Preservation Package usage is versatile in nature. Some use cases imply the list of acceptable formats stored in subpackages stored to remain open, like content exchange. On the other hand, long term preservation implies a strict selection of well-defined formats. For this reason different profiles are specified.

7.2 Profile attribution

The Cinema Preservation Package shall conform to one of the following profiles:

- the “Interim preservation profile”, to be used when the package content needs to stay flexible;
- the “Long term preservation profile”, to be used when a rigid selection of formats needs to be verified.

The “Interim preservation profile” shall be used as a default. All the recommended formats can be used. Although it is not recommended, other formats may be used as well.

A Cinema Preservation Package instance with the “Long term preservation profile” shall only contain sustainable formats and standards. Clause 9 specifies constraints on subpackages and defines the list of acceptable formats and applicable constraints to comply.

8 Content of Packing List files

8.1 Synthetic view (informative)

The synthetic view of the package is given in Figure 1.

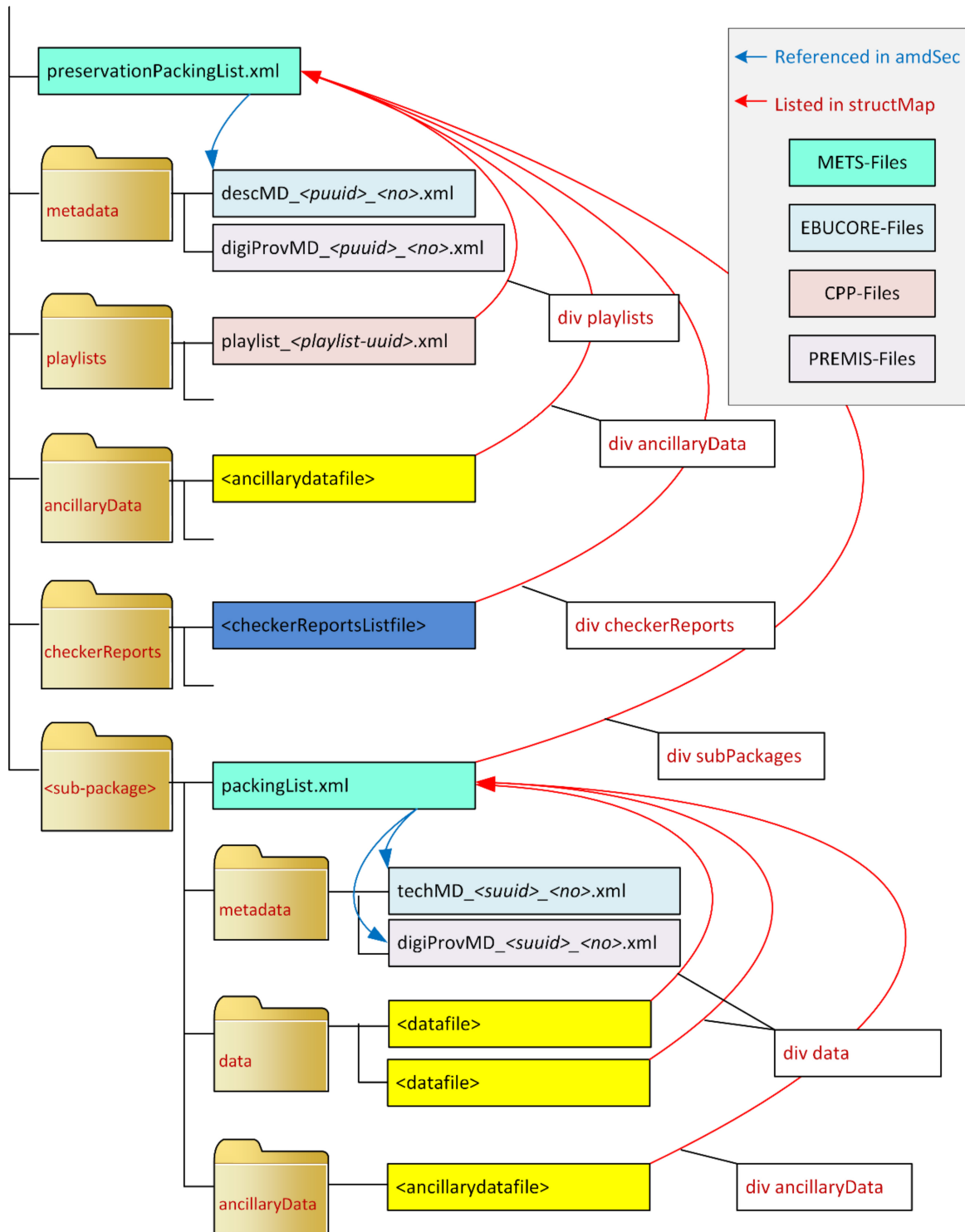


Figure 1 — File references inside Packing List

NOTE “amdSec”, “div” and “structMap” are elements of the METS Schema.

8.2 Packing List

8.2.1 General description (informative)

A Packing List is a METS document containing the following parts, under a root `<mets>` element:

- headers under the `<metsHdr>` element;
- metadata sections under `<dmdSec>` and `<amdSec>` elements;
- list of file pointers with integrity checks attributes under the `<fileSec>` element;
- definitions of abstract hierarchies under `<structMap>` elements.

The METS schema allows different elements dedicated to metadata. The metadata provisioned in this document are mapped to the adequate elements.

8.2.2 File conformance

The Packing List file shall conform to the XML standard and follow the METS XML Schema.

8.2.3 Common constraints on Packing List

8.2.3.1 Constraints on `<mets>` root element

XPath is: `"/mets:mets"`.

The `mets:mets` root element shall include the following namespaces:

- `"http://www.loc.gov/METS/";`
- `"http://www.w3.org/1999/xlink";`
- `"urn:cen.eu:en17650:2022:ns:packingList"` as specified in Annex A.

The following prose in this section assume that the used prefix for these namespaces are respectively "mets", "xlink" and "cpp", elements without namespaces belongs to the Mets schema.

Under the `<mets:mets>` root element, the elements in Table 1 shall be present.

Table 1 — Mandatory elements on the `<mets>` root element

Category	Topic	METS element	Element Arity	Attribute or subelement	Sub-element mandatory	value
Headers	Header data	metsHdr	1	@CREATEDATE	x	creation date in ISO 8601-1 format
				@cpp:packingList	x	packingListKind
				@cpp:packageProfile	o	packageProfile
				@cpp:packageFormatVersion	o	packageFormatVersion
				#children	x	See 8.2.3.2

Category	Topic	METS element	Element Arity	Attribute or subelement	Sub-element mandatory	value
Metadata	Descriptive metadata	dmdSec	0..n	@ID	x	<i>descriptiveMetadataId</i>
				#children	x	For Preservation Packing List see 8.3.4.2
	Provenance metadata	amdSec	1	@ID	x	<i>administrativeMetadataId</i>
				#children	x	For Preservation Packing List see 8.3.4.3, for Subpackage Packing List see 8.4.4.2
Files management	File section	fileSec	1	#children	x	For Preservation Packing List see 8.3.4.4, for Subpackage Packing List see 8.4.4.3
Abstract trees	Package physical description	structMap	1	@TYPE	x	"physical"
				#children	x	For Preservation Packing List see 8.3.4.9, for Subpackage Packing List see 8.4.4.6
	Package logical description	structMap	0..n	@TYPE	x	"logical"
				#children	x	For Preservation Packing List see 8.3.4.15.

See specific constraints for Preservation Packing List in 8.3.4.1 and for Subpackage Packing List in 8.4.4.1.

8.2.3.2 Constraints on the <metsHdr> element

XPath is: "/mets:mets/mets:metsHdr/".

The elements in Table 2 shall be present.

Table 2— Mandatory elements on the <metsHdr> element

Topic	METS element	Element Arity	Attribute or subelement	Subelement mandatory	value
Agent	agent	1	@ROLE	x	"CREATOR"
			@TYPE	x	"OTHER"
			#children	x	See Table 3

The elements in Table 3 shall be present.

Table 3 — Mandatory elements on the <agent> element

Topic	METS element	Element Arity	Attribute or subelement	Subelement mandatory	value
Name	name	1	#text	x	<i>packager agent name</i>

packager agent name shall be the name of the person or institution who produced the CPP package.

8.2.3.3 <fileGrp> elements common constraints

XPath is: "/mets:mets/mets:fileSec/mets:fileGrp/".

The element listed in Table 4 shall be repeated for each file belonging to the <fileGrp>.